UNIVERSITAT DE BARCELONA
Facultat de Matemàtiques i Informàtica
Treball final del grau de Matemàtiques

El problema del subgrup amagat

Guillem Serrallonga Rosell

Director: Dr. Artur Travesa i Grau
Barcelona, 24 de gener de 2022

Abstract

The hidden subgroup problem is a theoretical formalism which encompasses some problems of great importance, like factoring, discrete logarithm and graph isomorphism. Here we study Shor's algorithm for prime factorization and its relationship with the hidden subgroup problem. We start with an introduction to the theoretical framework of the hidden subgroup problem and some particular cases of this. Next we introduce the necessary concepts of the quantum computation circuit model in order to understand Shor's algorithm. Finally, we study those quantic algorithms which allow us to construct Shor's factoring algorithm.

Resum

El problema del subgrup amagat és un formalisme teòric que engloba alguns problemes de gran importància, com els de factorització, de logaritme discret i d'isomorfisme de grafs. En aquest treball estudiem l'algoritme de factorització de Shor i quina relació té amb el problema del subgrup amagat. Comencem amb una introducció al marc teòric del problema del subgrup amagat i alguns casos particulars d'aquest. Tot seguit introduïm els conceptes necessaris del model de computació quàntica de circuits per a poder entendre l'algoritme de Shor. Finalment, estudiem els algoritmes quàntics que ens permeten construir l'algoritme de factorització de Shor.

Agraïments

En primer lloc, vull agrair al meu tutor Dr. Artur Travesa i Grau la seva dedicació i paciència al llarg de la realització d'aquest treball. També vull agrair-li les noves idees i els diferents punts de vista que m'ha aportat, així com diverses fonts d'informació rellevants. En segon lloc m'agradaria agrair als meus amics per haver-me recolzat durant tota l'etapa universitària. Per últim, vull donar les gràcies a la meva família per oferir-me l'oportunitat d'estudiar aquest grau i tot el seu suport.

2020 Mathematics Subject Classification: 11A51, 11Y16, 68Q12.

Contingut

Introducció

El problema de factorització de nombres enters és un dels més antics i importants de les matemàtiques. Fins ara, els mètodes per a resoldre'l requereixen d'una quantitat inviable de recursos, fins i tot per a factoritzar nombres no tan grans que, segons Gauss, aquests algoritmes "[...] fatiguen la paciència del calculador expert; [···]" [Ga-96, Article 329]. D'aquest fet se n'aprofiten alguns sistemes criptogràfics moderns, ja que trobar la solució és molt difícil, però comprovar-la és prou senzill.

Com aquest problema, el càlcul del logaritme discret també és molt difícil d'efectuar i, en canvi, comprovar-ne la solució es pot aconseguir de manera eficient.

De fet, aquests i alguns altres problemes, en què la solució queda amagada i és difícil de trobar, però verificar-la és prou senzill, es poden classificar dins del formalisme del problema del subgrup amagat.

En aquest treball veurem com s'expressen els problemes de factorització i del logaritme discret com a instàncies del problema del subgrup amagat. Ens centrarem a resoldre el problema de factorització, que és un cas particular fonamental del problema del subgrup amagat, i que permet estendre's a la resta de casos. Tot i així, la complexitat d'aquest problema només és polinòmica per a casos específics, en què el grup que utilitzen és abelià.

Cap algoritme clàssic no és capaç, de moment, de trobar una solució en temps polinòmic a cap dels casos del problema del subgrup amagat. Per a que alguns d'aquests casos siguin tractables computacionalment, haurem de considerar un model de computació quàntica; un model que encara està en desenvolupament i només s'ha pogut implementar físicament amb capacitats molt limitades.

El model de computació quàntica que estudiarem i farem servir és el model de computació quàntica de circuits, que és l'habitual per a introduir la computació quàntica i el que comparteix més similituds amb els models de computació clàssica actuals. Cal notar que també hi ha algoritmes quàntics, per a la resolució d'algun cas particular del problema del subgrup amagat, que es recolzen en altres models; el més conegut (després del de circuits) és el model adiabàtic (cf. [Ki-19]).

Veurem doncs quin és el marc teòric del model de computació quàntica de circuits, les seves característiques principals i com es construeixen alguns algoritmes. En particular, construirem l'algoritme de factorització que va proposar Peter Shor l'any 1994 (cf. [Sh-94]); i per fer-ho, veurem la implementació de la transformada quàntica de Fourier, que és un resultat necessari per a la implementació d'aquest. Seguidament veurem, a nivell teòric, com es pot adaptar aquest algoritme per poder resoldre el cas abelià del problema del subgrup amagat.

Capítol 1
Problema del subgrup amagat

1.1 Problema del subgrup amagat

Els problemes de factorització i del logaritme discret són problemes molt diferents, però tots dos formen part d'una família de problemes més general.

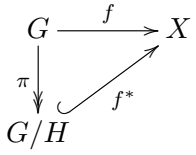
Intuïtivament, el problema de factorització consisteix a trobar els divisors primers d'un nombre enter donat. Aquests divisors estan barrejats dins del conjunt dels nombres menors que l'enter donat i no són fàcils de trobar. D'alguna manera podríem dir que estan ocults o amagats. Per al càlcul del logaritme discret ens trobem en un escenari similar. Les potències del generador queden desordenades o barrejades respecte l'ordre habitual, i això fa que trobar el valor de l'exponent que identifica un element qualsevol sigui tan difícil. En altres paraules, els exponents corresponents a les potències del generador queden amagats.

Anem a introduir ara un problema que generalitza aquesta propietat de buscar certs elements amagats, i que permet reduir aquests dos problemes comentats. Per a enunciar aquest problema, convé definir el concepte de separar classes laterals.

Definició 1.1.1 (Separa classes laterals)**.** Siguin G un grup, $H < G$ un subgrup i X un conjunt; es diu que una aplicació $f : G \to X$ **separa les classes laterals** (per la dreta) de G mòdul H si

$$\forall g, g' \in G, \quad f(g) = f(g') \iff gH = g'H.$$

És a dir, si f factoritza injectivament a través de la projecció en el conjunt quocient $G/H = \{gH : g \in G\}$:



Observació 1.1.2. Una condició equivalent de f per separar les classes laterals és que

$$\forall g, h \in G, \quad f(gh) = f(g) \iff h \in H,$$

ja que posant $h = g^{-1}g'$ tenim que $f(gh) = f(gg^{-1}g') = f(g')$. Per tant,

$$\begin{cases} f(g) = f(g') \iff f(gh) = f(g), \\ gH = g'H \iff h = g^{-1}g' \in H. \end{cases}$$

□

Problema 1.1.3 (Problema del subgrup amagat)**.** Siguin G un grup, $H < G$ un subgrup, X un conjunt finit i $f: G \to X$ una aplicació que separa les classes laterals de G mòdul H. Determinar un conjunt de generadors de H a partir d'avaluacions de f.

Distingirem els problemes que cauen dins del problema del subgrup amagat segons el tipus de grup G. En particular, parlarem del cas abelià per referir-nos als problemes en què G és un grup abelià i cas no abelià per als altres. A més, dins del cas abelià, serà de gran interès el cas en què G és cíclic; i el tractarem com a cas particular.

1.2 Factorització d'enters

El problema de descompondre un nombre enter qualsevol en els seus factors primers és un dels problemes més antics i importants de tota la història, estudiat pels matemàtics de totes les èpoques. Gauss ja s'hi referia així [Ga-96, Article 329]:

> "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, i que ha ocupat la diligència i la sagacitat dels geòmetres, tant els antics com els més moderns, és tant conegut que seria superflu parlar abundantment d'això. [...]"

Teorema 1.2.1 (Fonamental de l'Aritmètica)**.** *Sigui $N \in \mathbb{Z}$ un nombre enter no nul i no invertible, $n \notin \{0, 1, -1\}$. Existeixen $p_1, \ldots, p_r$ nombres primers positius i $\delta \in \{1, -1\}$ tals que $N = \delta p_1 \cdots p_r$. A més, δ i els p_i són únics llevat de l'ordre.*

El teorema fonamental de l'Aritmètica garanteix l'existència i la unicitat d'aquests factors primers, però no proporciona cap mètode per calcular-los numèricament. Al llarg d'aquest treball, direm *trobar* o *conèixer un nombre* per referir-nos a trobar o conèixer la seva expressió decimal (o binària) respectivament, i no només a una mera definició formal o propietat que determini el nombre.

Problema 1.2.2 (Factorització)**.** Sigui $N \in \mathbb{Z}$, $N \notin \{0, 1, -1\}$ un nombre enter conegut. Trobar els nombres primers positius $p_1, \ldots, p_r$ i $\delta \in \{1, -1\}$ tals que $N = \delta p_1 \cdots p_r$.

Com que δ queda determinat pel signe de N, podem assumir que $N > 1$. A més, podem reduir el problema de factoritzar qualsevol enter al problema de trobar un factor no trivial de qualsevol enter compost, i sovint serà aquest el problema al qual ens referirem quan parlem de factorització.

Per poder comparar els algoritmes i estimar la quantitat de temps d'execució que requereix cadascun d'ells, usarem la notació O-grossa per al càlcul estàndard de la complexitat. El valor respecte el qual es calcula la complexitat és la mida de l'*input* o de les dades del problema.

Es considera que un problema és tractable a nivell computacional si existeix algun algoritme que el resolgui de manera eficient; i es considera que un algoritme és eficient si la funció de complexitat creix, com a màxim, de forma polinòmica.

En el cas de la factorització d'un nombre N, $N > 1$, la mida del problema és el nombre de bits de N, $\log N$ (cf. [Co-93, §1.1.1]). Per tant, un algoritme de factorització serà eficient si la funció de complexitat és $O\Big((\log N)^k\Big) = O(poly\,(\log N))$.

Per exemple, l'algoritme de força bruta, que consisteix a dividir per tots els primers més petits que $\sqrt{N}$, té una complexitat $O\Big(\pi\Big(\sqrt{N}\Big)\Big) = O\Big(\frac{\sqrt{N}}{\log N}\Big) \supset O\Big(\sqrt[3]{N}\Big) = O\big(exp\,(\frac{1}{3}\log N)\big)$,

que és exponencial (i no polinòmica) i, per tant, no és eficient. Actualment, l'algoritme clàssic més eficient per a factoritzar un enter qualsevol és el garbell sobre cossos de nombres (NFS), amb un temps d'execució (cf. [Co-93, §10.5])

$$O\left(exp\left(\left(\sqrt[3]{\frac{64}{9}}+o(1)\right)(\log N)^{\frac{1}{3}}(\log\log N)^{\frac{2}{3}}\right)\right),$$

que és subexponencial. Tot i millorar la complexitat de l'algoritme de força bruta, no és suficient per poder-se considerar tractable a nivell computacional clàssic.

De fet, el rècord de factorització amb un algoritme genèric (que no utilitza cap propietat específica del nombre a factoritzar) és del nombre RSA-250, un nombre de 250 xifres decimals (829 bits) definit en el *RSA Factoring Challenge* (cf. [RSA-21]). Es va completar la computació el 28 de febrer de 2020, havent simulat 2700 anys de computació i utilitzant l'algoritme del garbell sobre cossos de nombres.

Actualment i segons el NIST (*National Institute of Standards and Technology*), els criptosistemes com RSA utilitzen nombres de 2048, 3072 o 4096 bits, és a dir, unes 617, 925 o 1234 xifres decimals respectivament (cf. [NIST-15]).

En computació clàssica no es coneix cap algoritme eficient de factorització. La pregunta natural ara és si la computació quàntica ens pot ajudar; i la resposta és que sí. Al 1994 Peter Shor va proposar un algoritme de factorització de complexitat polinòmica utilitzant un model de computació quàntica. Aquest algoritme, conegut com a algoritme de factorització de Shor o algoritme de Shor, es basa en el mateix principi matemàtic que l'algoritme $p-1$ de Pollard: el petit teorema de Fermat (cf. [Tr-98]). En particular, si p és un primer que divideix N, tots dos algoritmes parteixen d'un enter x invertible mòdul N, i busquen un exponent k que sigui un múltiple de l'ordre de x a $\left(\frac{\mathbb{Z}}{p\mathbb{Z}}\right)^*$ però que no ho sigui de l'ordre de x a $\left(\frac{\mathbb{Z}}{N\mathbb{Z}}\right)^*$; és a dir,

$$x^k \equiv 1 \pmod{p} \quad \text{i} \quad x^k \not\equiv 1 \pmod{N} \implies 1 < \operatorname{mcd}\left(x^k-1,\, N\right) < N.$$

Tots dos algoritmes són probabilístics, i no pas determinístics. Això vol dir que a vegades no troba cap solució (cap factor no trivial). Quan passa aquest fet, es diu que l'algoritme ha fracassat.

El mètode $p-1$ de Pollard consisteix a anar provant possibles valors de k i x en un cert ordre, fins que es trobi un factor de N. Si el factor trobat és N, aleshores l'algoritme ha fracassat. En canvi, l'algoritme de Shor utilitza la computació quàntica per calcular l'ordre r de l'element x a $\left(\frac{\mathbb{Z}}{N\mathbb{Z}}\right)^*$ triat a l'atzar. Si r és parell aleshores $k = r/2$ serà un bon candidat per a generar un factor no trivial de N. Els detalls de l'algoritme els veurem més endavant, a la secció 3.7.

L'objectiu ara és fer explícits quins grup G, subgrup H, conjunt X i funció f corresponen al problema de factorització per expressar-lo com a problema de subgrup amagat.

Observem primer que el problema de trobar l'ordre r d'un element b en un grup multiplicatiu M el podem expressar amb el formalisme de subgrup amagat com

$$G = \mathbb{Z}, \quad X = M, \quad f : k \mapsto b^k, \quad H = \langle r \rangle = r\mathbb{Z}.$$

Veiem que, efectivament, f separa les classes laterals de G mòdul H, ja que $\forall k, k' \in \mathbb{Z}$,

$$f(k) = f(k') \iff b^k = b^{k'} \in M \iff k \equiv k' \pmod{r} \iff k + r\mathbb{Z} = k' + r\mathbb{Z}.$$

Encara que $G = \mathbb{Z}$ sigui un grup infinit, a aquest cas se l'acostuma a anomenar cas cíclic. Això és degut al fet que si tenim un nombre m que sigui un múltiple de r, el formalisme de subgrup amagat també el podem expressar com

$$G = \frac{\mathbb{Z}}{m\mathbb{Z}}, \quad X = M, \quad f : k \mapsto b^k, \quad H = \langle r \rangle = \frac{r\mathbb{Z}}{m\mathbb{Z}}.$$

Cal notar que encara que existeixin altres generadors d'aquest subgrup, l'algoritme per a resoldre aquest problema permet trobar exactament el generador que ens interessa, r. Els detalls els veurem a la secció 3.6. A més a més, no és necessari conèixer el valor exacte de m; és suficient conèixer la seva existència i una cota.

El problema de factorització no té una traducció directa al formalisme de subgrup amagat, és a dir, els generadors de H no seran factors del nombre a factoritzar, sinó que seran nombres que ens ajudaran a trobar els factors. Per tant, com que la factorització es pot reduir a un problema que sí que admet el formalisme de manera directa (el problema de trobar l'ordre), aleshores s'utilitza aquesta reducció per a expressar la factorització com a problema de subgrup amagat.

Siguin $N \in \mathbb{Z}$ el nombre a factoritzar i x un enter invertible mòdul N. El problema de subgrup amagat associat s'expressa com

$$G = \mathbb{Z}, \quad X = \left(\frac{\mathbb{Z}}{N\mathbb{Z}}\right)^*, \quad f : k \mapsto x^k, \quad H = \langle r \rangle = r\mathbb{Z},$$

on r és l'ordre de x mòdul N. Si posem $m = \varphi(N)$, també el podem expressar com

$$G = \frac{\mathbb{Z}}{\varphi(N)\,\mathbb{Z}}, \quad X = \left(\frac{\mathbb{Z}}{N\mathbb{Z}}\right)^*, \quad f : k \mapsto x^k, \quad H = \langle r \rangle = \frac{r\mathbb{Z}}{\varphi(N)\,\mathbb{Z}},$$

on $\varphi(\cdot)$ és la funció phi d'Euler.

1.3 Logaritme discret

El problema de la factorització es pot reduir al problema del càlcul de l'ordre d'un element en el grup dels enters invertibles mòdul N, és a dir, trobar la primera potència (no nul·la) que doni 1 mòdul N. Una generalització natural d'aquest problema és el de calcular la primera potència que doni un valor donat. Aquest és el problema que definirem a continuació com a càlcul del logaritme discret.

Siguin b un element d'ordre finit d'un grup multiplicatiu qualsevol i m l'ordre de b. Considerem el subgrup cíclic C_b d'ordre m generat per b. Podem anomenar exponencial de base b a l'isomorfisme de grups $\frac{\mathbb{Z}}{m\mathbb{Z}} \to C_b$ definit per $k \mapsto b^k$. El seu invers és el logaritme discret en base b. Si $a = b^k$, escrivim $\log_b(a) = k$.

Problema 1.3.1 (Logaritme discret)**.** Sigui C_b un grup multiplicatiu cíclic d'ordre m generat per un element b conegut. Donat $a \in C_b$, calcular el valor de $\log_b(a)$.

En particular, si prenem $a = b^{-1}$, aleshores estem buscant quina potència de b és l'anterior al neutre. Per tant, estem calculant $m - 1$, que és l'ordre de b menys 1.

En quant a la complexitat dels algoritmes per a resoldre'l, l'algoritme de força bruta, que consisteix a calcular totes les potències de b; té una complexitat $O(m) = O(exp(\log m))$, i per tant, no és gens eficient. Actualment, l'algoritme clàssic més eficient per a calcular el

logaritme discret en un grup cíclic d'ordre potència d'un primer p qualsevol és el garbell sobre cossos de funcions, amb un temps d'execució subexponencial de (cf. [Ad,Hu-99])

$$O\left(exp\left(\left(\sqrt[3]{\frac{32}{9}}+o(1)\right)(\log m)^{\frac{1}{3}}(\log\log m)^{\frac{2}{3}}\right)\right).$$

Tot i així, si la potència de p és prou petita, aleshores l'algoritme del garbell sobre cossos de nombres és lleugerament més ràpid. Però segueix sent insuficient per poder-se considerar tractable a nivell computacional clàssic.

A diferència del problema de factorització, el problema del càlcul del logaritme discret sí que el podem expressar amb el formalisme de subgrup amagat de manera directa. Siguin M un grup multiplicatiu qualsevol, $b \in M$ la base del logaritme, C_b el subgrup cíclic generat per b i $r = |C_b|$. Donat un element $A = b^k \in C_b$ volem calcular $k = \log_b(A)$. Posem

$$G = \mathbb{Z}\times\mathbb{Z}, \quad X = M, \quad f : (x,y) \mapsto b^x A^{-y} = b^{x-yk},$$
$$H = \langle (k,1),(r,0)\rangle = (k,1)\mathbb{Z} + (r,0)\mathbb{Z} = \{(x,y) : x = yk + \lambda r,\ \lambda \in \mathbb{Z}\}.$$

Efectivament, f separa les classes laterals de G mòdul H, perquè $\forall (x,y),(x',y') \in \mathbb{Z}\times\mathbb{Z}$,

$$f((x,y)+(x',y')) = f(x+x', y+y') = f(x,y) \iff b^{x+x'-(y+y')k} = b^{x-yk} \in M \iff$$
$$\iff x + x' - (y+y')k \equiv x - yk \pmod r \iff x' \equiv y'k \pmod r \iff (x',y') \in H.$$

De fet, podem expressar-ho de manera que H tingui un sol generador si posem

$$G = \frac{\mathbb{Z}}{r\mathbb{Z}} \times \frac{\mathbb{Z}}{r\mathbb{Z}}, \quad X = M, \quad f : (x,y) \mapsto b^x A^{-y}, \quad H = \langle (k,1)\rangle = \left\{(\ell k, \ell) : \ell \in \frac{\mathbb{Z}}{r\mathbb{Z}}\right\}.$$

Observem que els possibles generadors de H són de la forma $(\ell k, \ell) \in \frac{\mathbb{Z}}{r\mathbb{Z}} \times \frac{\mathbb{Z}}{r\mathbb{Z}}$. Per tant, si ℓ és invertible mòdul r, podem obtenir fàcilment $\log_b(A) = k = \ell k \cdot \ell^{-1}$.

1.4 Problemes de xarxes

En termes de seguretat informàtica, podem trobar el problema del subgrup amagat en els fonaments de molts sistemes criptogràfics actuals. Per exemple, els sistemes de clau pública com El Gamal i RSA, requereixen del coneixement de la clau privada per poder desxifrar el missatge, però per calcular-la a partir de la clau pública, és necessari resoldre un dels problemes de factorització o de logaritme discret. De la mateixa manera, protocols d'intercanvi segur de claus, com el de Diffie-Hellman, confien la seva seguretat en la dificultat de resoldre aquests problemes mencionats.

Però no tots els criptosistemes actuals basen la seva seguretat en els problemes de factorització o de logaritme discret. Hi ha problemes relacionats amb xarxes (subgrups discrets de rang n de $\mathbb{R}^n$), com els problemes del vector més curt en una xarxa o del vector més proper, que donen lloc a altres sistemes criptogràfics i protocols de signatura com RLWE-KEX i GGH respectivament.

És per aquest motiu que un algoritme eficient que resolgués algun d'aquests problemes comprometria la seguretat dels criptosistemes corresponents.

Veiem, doncs, quina relació tenen aquests dos problemes de xarxes amb el problema del subgrup amagat.

Problema 1.4.1 (Vector més curt en una xarxa)**.** Donada una xarxa de dimensió n, trobar el vector no nul més curt de la xarxa.

Una generalització d'aquest problema, però de complexitat equivalent, és la següent.

Problema 1.4.2 (Vector més proper en una xarxa)**.** Donada una xarxa de dimensió n i un vector $v \in \mathbb{R}^n$ qualsevol, trobar el vector de la xarxa més proper al vector v donat.

Malgrat que aquests dos problemes de xarxes no caben dins del formalisme del subgrup amagat, sí que hi tenen una forta relació (cf. [Re-04]). Bàsicament, si trobem un algoritme que resolgui el problema del subgrup amagat per al grup diedral d'ordre $2N$, $G = D_{2N}$, usant el mètode de mostreig de Fourier (*Fourier sampling*), també anomenat mostreig de classes laterals (*sampling cosets*); aleshores els problemes del vector més curt i més proper es poden reduir a un problema que es pot resoldre amb una modificació d'aquest algoritme. Aquesta modificació consisteix a saltar-se els primers passos. Si n és la dimensió de la xarxa i $M \in 2^{\mathcal{O}(n)}$ és un nombre natural prou gran, aleshores s'utilitza $N = (2M)^n$.

1.5 Isomorfisme de grafs

Problema 1.5.1 (Isomorfisme de grafs)**.** Donats dos grafs $\mathcal{G}_1$ i $\mathcal{G}_2$, determinar si són isomorfs o no ($\mathcal{G}_1 \simeq \mathcal{G}_2$ o $\mathcal{G}_1 \not\simeq \mathcal{G}_2$).

Podem suposar que $\mathcal{G}_1$ i $\mathcal{G}_2$ tenen el mateix nombre de vèrtexs i d'arestes, ja que és una condició necessària per a poder ser isomorfs. També podem suposar que són connexos, perquè sinó només cal comparar els components connexos de $\mathcal{G}_1$ amb els de $\mathcal{G}_2$.

Aquest problema redueix al problema de trobar el grup d'automorfismes d'un graf $\mathcal{G}$ de n vèrtexs, que es pot expressar com a problema de subgrup amagat posant

$$G = \mathbb{S}_n, \quad X = \{\pi(\mathcal{G}) : \pi \in \mathbb{S}_n\}, \quad f : \pi \mapsto \pi(\mathcal{G}), \quad H = \mathrm{Aut}(\mathcal{G}),$$

on $\pi(\mathcal{G})$ denota el graf resultant de permutar els vèrtexs de $\mathcal{G}$ amb la permutació $\pi \in \mathbb{S}_n$.

Un isomorfisme $\mathcal{G}_1 \simeq \mathcal{G}_2$ es pot expressar com un automorfisme de la unió disjunta dels dos grafs, $\mathcal{G}_1 \sqcup \mathcal{G}_2$. A més a més, aquest automorfisme no es pot expressar com a producte cartesià d'un automorfisme de $\mathcal{G}_1$ per un de $\mathcal{G}_2$. Com que $\mathcal{G}_1$ i $\mathcal{G}_2$ són connexos, tenim una caracterització de l'existència d'isomorfisme entre els dos grafs:

$$\mathcal{G}_1 \simeq \mathcal{G}_2 \iff \mathrm{Aut}(\mathcal{G}_1) \times \mathrm{Aut}(\mathcal{G}_2) \subsetneq \mathrm{Aut}(\mathcal{G}_1 \sqcup \mathcal{G}_2).$$

Per tant, el problema de l'isomorfisme de grafs redueix al problema de determinar el grup d'automorfismes dels grafs $\mathcal{G}_1$, $\mathcal{G}_2$ i $\mathcal{G}_1 \sqcup \mathcal{G}_2$.

El problema de l'isomorfisme de grafs té aplicacions en diverses àrees científiques.

Per exemple, algunes tècniques de reconeixement facial utilitzen grafs per identificar una cara. En l'àmbit de la química sovint s'usen tests d'isomorfismes de grafs per identificar compostos químics o generar grafs moleculars. Fins i tot, en electrònica permet verificar si un circuit imprès en una placa és el mateix que el model esquemàtic.

Hi ha tants problemes que es poden reduir al de l'isomorfisme de grafs, encara que la majoria siguin problemes purament de teoria de grafs, que s'ha creat una nova categoria de complexitat, GI, per poder estudiar i classificar aquests problemes. No se sap si GI és igual a P o NP, però quedaria determinat segons si el problema de l'isomorfisme de grafs fos P o NP-complet respectivament.

Capítol 2
Eines quàntiques

2.1 Qubits i notació quàntica

2.1.1 Bits i qubits

En computació clàssica es considera el bit com la unitat bàsica d'informació. Un bit admet només dos estats, que els representem matemàticament per 0 i 1. Així doncs, un bit és un element del conjunt $\{0, 1\}$.

Un bit quàntic o qubit és una unitat d'informació, anàleg al bit, però que admet molts més estats. El codifiquem com un vector unitari de $\mathbb{C}^2$. La norma que s'utilitza habitualment és la 2-norma o norma euclidiana.

Usarem la notació de Dirac, "$\langle\,|\,,|\,\rangle$", també anomenada bra-ket, ja que és la notació estàndard en mecànica quàntica. A l'expressió $|\delta\rangle$ se l'anomena ket, i a l'expressió $\langle\delta|$, bra (cf. [Mi,Ni-10, §1.2], [Pe-02, §3.9], [Wi-11, §1.3]). Representarem els qubits (vectors de $\mathbb{C}^2$) amb kets. Els bras correspondran a les formes corresponents al transposat i conjugat d'un vector (d'un ket). Així doncs, escriurem

$$|\psi\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix} \in \mathbb{C}^2, \qquad |\psi\rangle^\dagger = \langle\psi| = \begin{pmatrix}\bar{\alpha} & \bar{\beta}\end{pmatrix},$$

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

En treballar amb els vectors unitaris de $\mathbb{C}^2$, donarem nom als vectors de la base canònica, sovint anomenada base computacional:

$$|0\rangle := \begin{pmatrix}1\\ 0\end{pmatrix}, \qquad |1\rangle := \begin{pmatrix}0\\ 1\end{pmatrix}.$$

Per tant, podem expressar un vector (o qubit) qualsevol com a combinació lineal sobre els complexos de $|0\rangle$ i $|1\rangle$,

$$|\psi\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \alpha\,|0\rangle + \beta\,|1\rangle,$$

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

Notació 2.1.1. Anomenarem estat d'un qubit o superposició a una combinació lineal normalitzada i sobre els complexos respecte la base $\{|0\rangle, |1\rangle\}$.

2.1.2 Mesura d'un qubit

Malgrat que *a priori* pugui semblar que aquest model pot tractar una quantitat infinita d'informació amb un sol qubit, cal tenir present que el model està basat en sistemes físics reals, amb certes propietats i limitacions. L'acció de mesurar un qubit en modifica l'estat: només pot retornar un dels dos estats $|0\rangle$ o $|1\rangle$, i el qubit canvia a l'estat retornat. Es diu que en mesurar un qubit, aquest col·lapsa.

Això pot semblar poc intuïtiu, però volem que el model reflecteixi aquest fenomen. De la mateixa manera que els estats d'un bit poden correspondre a encès/apagat o a alt/baix voltatge, els estats d'un qubit corresponen a propietats de certs sistemes quàntics, que per la seva naturalesa quàntica, les seves propietats no estan determinades fins que es mesuren (i col·lapsen). Per exemple, podem utilitzar com a estats d'un qubit l'espín d'un electró, que es pot mesurar paral·lel a l'eix de mesura (*spin-up* o *spin-*$\frac{1}{2}$, $|\uparrow\rangle$) o anti-paral·lel (*spin-down* o *spin-*$\frac{3}{2}$, $|\downarrow\rangle$). També es podria utilitzar la polarització d'un fotó o qualsevol altra propietat quàntica amb dos possibles estats. Hi ha diferents opcions, i cadascuna aporta diferents avantatges i inconvenients (cf. [Mi,Ni-10, §7], [Un-95], [Wi-11, §1.3]).

Definició 2.1.2 (Mesurar)**. Mesurar un qubit** de la forma $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ consisteix a realitzar l'experiment aleatori d'una variable aleatòria que pren els valors $|0\rangle$ i $|1\rangle$ amb probabilitats $|\alpha|^2$ i $|\beta|^2$ respectivament; i sobreescriure o canviar l'estat del qubit amb el resultat de l'experiment.

Observem que l'acció de mesurar un qubit no és una operació reversible. És a dir, si volguéssim repetir l'experiment i tornar a mesurar el qubit, primer hauríem de construir altre cop l'estat que teníem.

Cal notar que hem definit l'acció de mesurar respecte la base computacional (o canònica). A nivell teòric es pot mesurar respecte una base (ortonormal) qualsevol, però no aporta cap benefici extra en quant a complexitat, i per tant, habitualment només es considera la mesura respecte la base computacional.

2.1.3 Sistemes de n qubits

Podem crear sistemes de més d'un qubit per així augmentar la quantitat d'informació que es pot tractar. La idea és la mateixa que quan augmentem el nombre de bits en el model clàssic, encara que el resultat és lleugerament diferent. En comptes de parlar del *conjunt d'estats*, parlarem de l'*espai d'estats*.

L'espai d'estats d'un sistema de n qubits és, per definició, el producte tensorial dels espais d'estats de cada qubit per separat (cf. [Wi-11, §1.4]). Per tant, l'espai d'estats és el conjunt dels vectors unitaris de $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2 = \left(\mathbb{C}^2\right)^{\otimes n} \cong \mathbb{C}^{2^n}$, on tenim una $\mathbb{C}$-base ortonormal induïda pels productes tensorials

$$\begin{array}{rcccl}
|0\rangle_n & := & |0\ldots000\rangle & := & |0\rangle \otimes \cdots \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle\,, \\
|1\rangle_n & := & |0\ldots001\rangle & := & |0\rangle \otimes \cdots \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle\,, \\
|2\rangle_n & := & |0\ldots010\rangle & := & |0\rangle \otimes \cdots \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle\,, \\
|3\rangle_n & := & |0\ldots011\rangle & := & |0\rangle \otimes \cdots \otimes |0\rangle \otimes |1\rangle \otimes |1\rangle\,, \\
 & \vdots & & \vdots & \\
|2^n-1\rangle_n & := & |1\ldots111\rangle & := & |1\rangle \otimes \cdots \otimes |1\rangle \otimes |1\rangle \otimes |1\rangle\,,
\end{array}$$

que és exactament la $\mathbb{C}$-base canònica de $\mathbb{C}^{2^n}$.

Per exemple, si $n=3$, el vector $|6\rangle_3$ viu a $\left(\mathbb{C}^2\right)^{\otimes 3} \cong \mathbb{C}^{2^3} = \mathbb{C}^8$ i és, per definició,

$$|6\rangle_3 = |110\rangle = |1\rangle \otimes |1\rangle \otimes |0\rangle = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix} 0 \begin{pmatrix}1\\0\end{pmatrix} \\ 1 \begin{pmatrix}1\\0\end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \begin{pmatrix}0\\1\end{pmatrix} \\ 1 \begin{pmatrix}0\\1\end{pmatrix} \end{pmatrix} = \begin{pmatrix}0\\0\\0\\0\\0\\0\\1\\0\end{pmatrix}.$$

Per tant, podem expressar un estat $|\psi\rangle_n \in \mathbb{C}^{2^n}$ qualsevol com

$$|\psi\rangle_n := \sum_{k=0}^{2^n-1} \alpha_k |k\rangle_n \in \mathbb{C}^{2^n}, \quad \forall k,\ \alpha_k \in \mathbb{C},\ \sum_{k=0}^{2^n-1} |\alpha_k|^2 = 1,$$

que sovint anomenarem registre o estat en superposició de n qubits.

Notació 2.1.3. Donats dos registres $|\psi\rangle_n$ i $|\varphi\rangle_m$ de n i m qubits respectivament, sovint escriurem $|\psi\rangle_n |\varphi\rangle_m := |\psi\rangle_n \otimes |\varphi\rangle_m$.

La mesura d'un registre de n qubits serà anàloga a la mesura de cada qubit individualment, sense importar l'ordre. El resultat serà un element de la $\mathbb{C}$-base de $\mathbb{C}^{2^n}$. Si mesurem el registre

$$|\psi\rangle_n = \sum_{k=0}^{2^n-1} \alpha_k |k\rangle_n = \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} \cdots \sum_{k_n=0}^{1} \alpha_{k_1 k_2 \ldots k_n} |k_1 k_2 \ldots k_n\rangle \in \mathbb{C}^{2^n},$$

on $k_1 k_2 \ldots k_n$ representa l'expressió binària de k; aleshores la probabilitat que cadascun dels qubits q_i col·lapsi a l'estat $|k_i\rangle$ és $|\alpha_{k_1 k_2 \ldots k_n}|^2$; és a dir, la probabilitat que $|\psi\rangle_n$ col·lapsi a $|k\rangle_n$ és $|\alpha_k|^2$.

Observació 2.1.4. Les variables aleatòries corresponents a mesurar cada qubit no són necessàriament independents entre elles. És a dir, al mesurar un qubit q_i, aquest col·lapsa i modifica el seu estat, i per tant, modifica l'estat de tot el registre sencer. Els coeficients corresponents a cadascun dels vectors de la base computacional queden modificats d'acord amb les probabilitats corresponents condicionades a haver mesurat el qubit q_i. És per aquest motiu que l'ordre de mesura és indiferent, ja que gràcies a la regla de Bayes, la probabilitat de mesurar tots els qubits alhora és la mateixa que mesurar-los un per un condicionats cadascun d'ells a haver mesurat els qubits anteriors (cf. [Pe-02, §2]).

La definició de l'espai d'estats d'un sistema de n qubits amb el producte tensorial (i no amb una suma directa) permet modelitzar el fenòmen de l'entrellaçament quàntic. Els estats de $\mathbb{C}^{2^n}$ que es poden escriure com a producte tensorial de vectors de $\mathbb{C}^2$ s'anomenen estats purs. Però no tots els estats es poden escriure d'aquesta manera. Els estats que no són purs s'anomenen estats entrellaçats.

Un exemple d'estat entrellaçat és l'estat de Bell $|\beta_{00}\rangle_2 := \frac{|00\rangle+|11\rangle}{\sqrt{2}}$ (cf. [Mi,Ni-10, §1.3.6]), ja que si fos pur, podríem escriure

$$\begin{pmatrix}\alpha\\\beta\end{pmatrix} \otimes \begin{pmatrix}\gamma\\\delta\end{pmatrix} = \begin{pmatrix}\alpha\gamma\\\alpha\delta\\\beta\gamma\\\beta\delta\end{pmatrix} = |\beta_{00}\rangle_2 = \frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix}1\\0\\0\\1\end{pmatrix},$$

que com que $\mathbb{C}$ no té divisors de zero, és una contradicció ($\alpha\gamma \neq 0 \neq \beta\delta$ i $\alpha\delta = 0 = \beta\gamma$). $\square$

2.2 Portes quàntiques

2.2.1 Operacions vàlides

Les accions que modifiquen els bits d'un model clàssic s'anomenen portes lògiques, i són aplicacions $\{0,1\}^n \to \{0,1\}^m$, sense cap mena de restricció. Però el seu anàleg al model quàntic de circuits, les portes quàntiques, sí que tindran certes restriccions.

Per començar, el nombre de qubits d'entrada haurà de ser el mateix que el de sortida, ja que les operacions s'han d'aplicar en un sistema de qubits tancat per tal de conservar els estats entrellaçats o de superposició sense col·lapsar. Això ja ens limita només a aplicacions $\mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$. Però, com que els qubits han de ser vectors unitaris, només podrem realitzar les operacions unitàries en $\mathbb{C}^{2^n}$.

Al tractar $\mathbb{C}^{2^n}$ com a $\mathbb{C}$-espai vectorial, tota transformació unitària estarà definida per una matriu unitària. Recordem que una matriu A és unitària si i només si $AA^\dagger = A^\dagger A = Id$, on $A^\dagger$ denota la matriu transposada i conjugada de A.

Definició 2.2.1 (Porta quàntica)**.** Una **porta quàntica** que actua sobre un registre de n qubits és una transformació unitària en $\mathbb{C}^{2^n}$.

Notació 2.2.2. Sovint representem les portes quàntiques de n qubits per les matrius complexes unitàries $2^n \times 2^n$. Com que un registre $|\psi\rangle_n$ qualsevol de n qubits el representem com a vector unitari de $\mathbb{C}^{2^n}$, escrivim $U|\psi\rangle_n := U(|\psi\rangle_n)$, per a cada transformació unitària $U : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$.

Observació 2.2.3. Una conseqüència immediata és que totes les operacions amb portes quàntiques són reversibles, ja que tota transformació unitària té inversa.

En computació clàssica existeixen portes lògiques que no són reversibles. Per tant, *a priori* semblaria que la computació quàntica no podria simular totes les operacions clàssiques. Però s'ha trobat la manera de realitzar aquests càlculs de manera reversible usant registres auxiliars (cf. [Be-73]). Això ho explicarem amb més detall al final de la propera secció.

2.2.2 Portes quàntiques principals

Recordem, doncs, que les bases computacionals de $\mathbb{C}^2$ i $\mathbb{C}^{2^2}$ són

$$\left\{|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}\right\} \text{ i } \left\{|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, |10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\right\}$$

respectivament, i així també podrem definir matricialment les transformacions bàsiques.

Veiem primer les portes quàntiques bàsiques o principals que actuen sobre un sol qubit, i que per tant estan definides a $\mathbb{C}^2$ per matrius complexes unitàries 2×2. A més, dibuixarem l'esquema del circuit quàntic corresponent a cadascuna de les portes.

Identitat (I): La identitat no modifica l'estat del qubit. El seu esquema del circuit és

$$|\delta\rangle \text{ —} \boxed{I} \text{— } |\delta\rangle, \quad \text{o simplement} \qquad |\delta\rangle \text{ ——— } |\delta\rangle.$$

Correspon a la transformació $\begin{cases}|0\rangle \mapsto |0\rangle\,,\\ |1\rangle \mapsto |1\rangle\,,\end{cases}$ de matriu $I = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$.

Negació (NOT, X): La negació intercanvia els estats de la base. Equival a la porta lògica clàssica de la negació. El seu esquema del circuit és

$$|\delta\rangle \text{ —}\boxed{X}\text{— } X|\delta\rangle, \quad \text{o bé} \qquad |\delta\rangle \text{ —}\oplus\text{— } X|\delta\rangle.$$

Correspon a la transformació $\begin{cases} |0\rangle \mapsto |1\rangle\,, \\ |1\rangle \mapsto |0\rangle\,, \end{cases}$ de matriu $NOT = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

Hadamard (H): La porta d'Hadamard envia els estats de la base a la suma i la diferència normalitzades. El seu esquema del circuit és

$$|\delta\rangle \text{ —}\boxed{H}\text{— } H|\delta\rangle.$$

Correspon a la transformació $\begin{cases} |0\rangle \mapsto |+\rangle := \dfrac{|0\rangle + |1\rangle}{\sqrt{2}}, \\ |1\rangle \mapsto |-\rangle := \dfrac{|0\rangle - |1\rangle}{\sqrt{2}}, \end{cases}$ de matriu $H = \dfrac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.

Porta de desplaçament de fase ($R(\phi)$): La porta de desplaçament de fase no modifica les probabilitats de la mesura, però desfasa el coeficient de $|1\rangle$. El seu esquema del circuit és

$$|\delta\rangle \text{ —}\boxed{R(\phi)}\text{— } R(\phi)|\delta\rangle.$$

Correspon a la transformació $\begin{cases} |0\rangle \mapsto |0\rangle\,, \\ |1\rangle \mapsto e^{i\phi}\,|1\rangle\,, \end{cases}$ de matriu $R(\phi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$.

Ara que hem vist les portes quàntiques d'un sol qubit més rellevants, passem a veure com es construeixen portes de dos qubits. Començarem pel cas general del concepte de porta controlada, i després donarem alguns exemples més concrets de portes controlades que també es consideren portes quàntiques bàsiques.

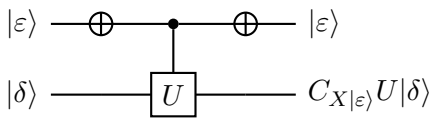
Portes controlades (CU): La majoria de portes quàntiques principals que actuen sobre sistemes de dos qubits són les portes controlades, que es creen a partir d'una porta d'un sol qubit U. El primer qubit és el de control i al segon s'aplica la transformació U si el qubit de control és $|1\rangle$ i la identitat si és $|0\rangle$. El seu esquema del circuit és

$$\begin{array}{ll} |\varepsilon\rangle & \text{———●———} \;\; |\varepsilon\rangle \\ |\delta\rangle & \text{—}\boxed{U}\text{—} \;\; C_\varepsilon U|\delta\rangle. \end{array}$$

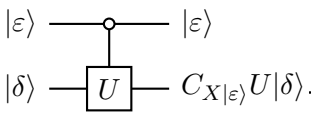
Correspon a la transformació $\begin{cases} |00\rangle = |0\rangle\,|0\rangle \mapsto |0\rangle\,|0\rangle\,, \\ |01\rangle = |0\rangle\,|1\rangle \mapsto |0\rangle\,|1\rangle\,, \\ |10\rangle = |1\rangle\,|0\rangle \mapsto |1\rangle\,U\,|0\rangle\,, \\ |11\rangle = |1\rangle\,|1\rangle \mapsto |1\rangle\,U\,|1\rangle\,, \end{cases}$ i si escrivim $U = \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix}$,

aleshores CU correspon a la matriu $CU = \begin{pmatrix} I & 0 \\ 0 & U \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{pmatrix}$.

A vegades, però, ens interessarà controlar una porta però en comptes d'aplicar U si el qubit de control és $|1\rangle$, aplicar-la si aquest és $|0\rangle$. El seu esquema del circuit és

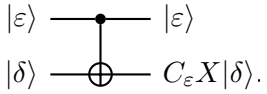


o equivalentment



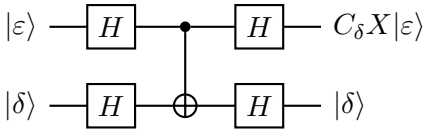
Correspon a la transformació $\begin{cases} |00\rangle = |0\rangle\,|0\rangle \mapsto |0\rangle\,U\,|0\rangle\,, \\ |01\rangle = |0\rangle\,|1\rangle \mapsto |0\rangle\,U\,|1\rangle\,, \\ |10\rangle = |1\rangle\,|0\rangle \mapsto |1\rangle\,|0\rangle\,, \\ |11\rangle = |1\rangle\,|1\rangle \mapsto |1\rangle\,|1\rangle\,, \end{cases}$ de matriu $CU = \begin{pmatrix} U & 0 \\ 0 & I \end{pmatrix}$.

Negació controlada (CNOT, CX): La negació controlada intercanvia els estats $|10\rangle$ i $|11\rangle$. El seu esquema del circuit és

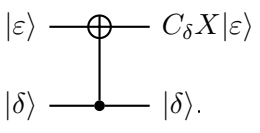


Correspon a la transformació $\begin{cases} |00\rangle \mapsto |00\rangle\,, \\ |01\rangle \mapsto |01\rangle\,, \\ |10\rangle \mapsto |11\rangle\,, \\ |11\rangle \mapsto |10\rangle\,, \end{cases}$ de matriu $CNOT = CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$.

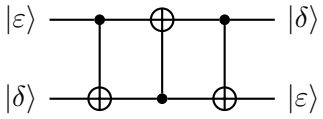
Per obtenir la negació controlada, però que el qubit de control sigui el segon en comptes del primer, podem aplicar portes Hadamard de la manera següent:



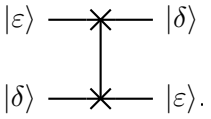
o equivalentment



SWAP: La porta Swap intercanvia els dos qubits, que a efectes pràctics és equivalent a intercanviar els estats $|01\rangle$ i $|10\rangle$. Aquesta porta es pot obtenir a partir de la negació controlada, aplicada tres vegades consecutives però alternant el qubit de control. El seu esquema del circuit és

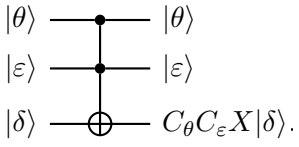


o equivalentment



Correspon a la transformació $\begin{cases} |00\rangle \mapsto |00\rangle\,, \\ |01\rangle \mapsto |10\rangle\,, \\ |10\rangle \mapsto |01\rangle\,, \\ |11\rangle \mapsto |11\rangle\,, \end{cases}$ de matriu SWAP $= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$.

Toffoli (CCNOT, CCX, TOFF): La porta Toffoli actua sobre tres qubits i és la porta controlada de la negació controlada, és a dir, la negació doblement controlada. Per tant només intercanvia els estats de la base del tercer qubit quan els dos primers estan en estat $|1\rangle$ i deixa tot igual en cas contrari, que a efectes pràctics és equivalent a intercanviar només els estats $|110\rangle$ i $|111\rangle$. El seu esquema del circuit és



$$\text{Correspon a la transformació } \begin{cases} |000\rangle \mapsto |000\rangle\,, \\ |001\rangle \mapsto |001\rangle\,, \\ |010\rangle \mapsto |010\rangle\,, \\ |011\rangle \mapsto |011\rangle\,, \\ |100\rangle \mapsto |100\rangle\,, \\ |101\rangle \mapsto |101\rangle\,, \\ |110\rangle \mapsto |111\rangle\,, \\ |111\rangle \mapsto |110\rangle\,, \end{cases} \quad \text{de matriu } TOFF = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

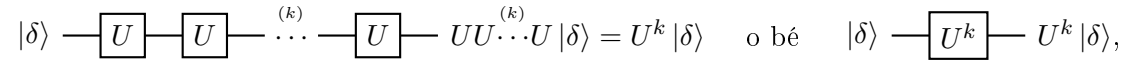
Mesura: La mesura no és una porta quàntica, però convé tenir present el seu símbol per als esquemes de circuit:

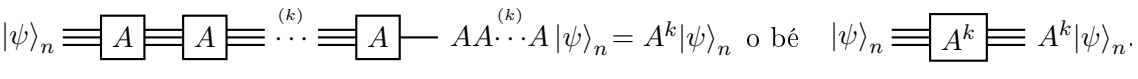
2.2.3 Combinació de portes quàntiques

Veiem ara les dues maneres de combinar portes quàntiques: Producte matricial i producte tensorial. Si a més sabem com calcular la inversa de les portes bàsiques i d'una combinació de portes, aleshores podrem construir qualsevol circuit quàntic a partir de les portes quàntiques bàsiques que acabem de veure (cf. [EGQC-95]). En particular, com construir operacions no reversibles amb portes quàntiques.

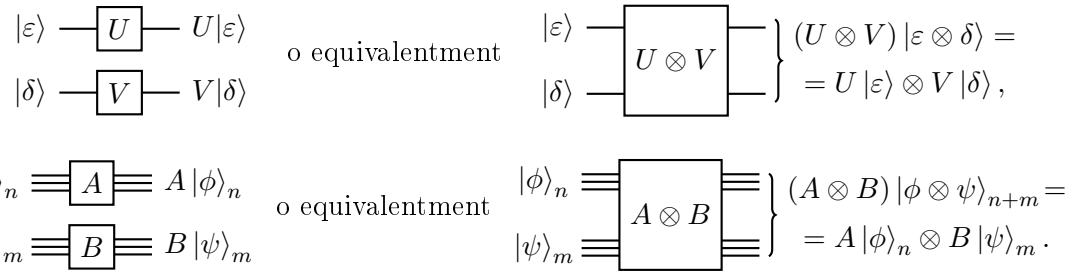
Producte matricial de portes ($U \cdot V$, $A \cdot B$): El producte matricial de dues portes U i V tals que cadascuna actua sobre un sol qubit correspon a aplicar-les una a continuació de l'altra, i de la mateixa manera, el producte matricial de dues portes A i B tals que cadascuna actua sobre un registre de n qubits correspon a concatenar-les. Ho expressem amb els esquemes de circuit

Definirem les potències d'una porta com el producte matricial iterat. Si U i A són portes que actuen sobre un qubit i un registre de n qubits respectivament, les potències k-èsimes s'escriuran amb els esquemes de circuit

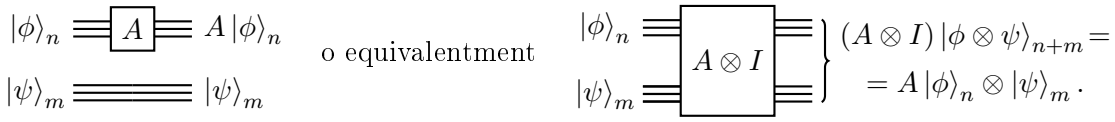




Producte tensorial de portes ($U \otimes V$, $A \otimes B$): El producte tensorial de dues portes U i V tals que cadascuna actua sobre un sol qubit correspon a aplicar per separat cada porta al qubit corresponent, i en general, el producte tensorial de dues portes A i B tals que cadascuna actua sobre registres de n i m qubits respectivament correspon a aplicar per separat cada porta al registre corresponent. Ho expressem amb els esquemes de circuit



El cas particular en què la porta B és la identitat permet aplicar portes quàntiques en un subconjunt de qubits sense afectar la resta, és a dir, aplicar una porta A en un registre sense modificar l'altre registre. Ho expressem amb l'esquema de circuit



Portes inverses: Hem vist que les portes quàntiques són reversibles. Ens preguntem ara com calcular les inverses de les portes bàsiques. Observem que en ser transformacions unitàries, la seva inversa és la transposada conjugada, $U^{-1} = U^\dagger$.

Com que les matrius corresponents són simètriques, transposar no canvia res; i com que la majoria són reals, en aquests casos conjugar tampoc les afectarà.

$$\left\{\begin{array}{ll} I^{-1} & = I \\ X^{-1} & = X \\ H^{-1} & = H \end{array}\right\} \text{simètriques reals.}$$

$$\left\{\begin{array}{ll} R(\phi)^{-1} & = R(-\phi) \\ CR(\phi)^{-1} & = CR(-\phi) \end{array}\right\} \text{simètriques, } \overline{e^{i\phi}} = e^{-i\phi}.$$

$$\left\{\begin{array}{ll} CX^{-1} & = CX \\ \text{SWAP}^{-1} & = \text{SWAP} \\ CCX^{-1} & = CCX \end{array}\right\} \text{simètriques reals.}$$

Les portes controlades CU per a qualsevol porta U veiem que

$$CU^{-1} = CU^\dagger = \begin{pmatrix} I & 0 \\ 0 & U \end{pmatrix}^\dagger = \begin{pmatrix} I & 0 \\ 0 & U^\dagger \end{pmatrix} = C\Big(U^\dagger\Big) = C\big(U^{-1}\big)\,.$$

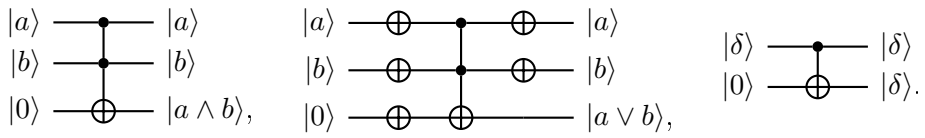
La inversa d'un producte tensorial és el producte de les inverses per separat, ja que cadascuna s'aplica a un registre diferent. La transposada conjugada d'un producte matricial és el producte de les transposades conjugades canviades d'ordre. Amb aquestes dues propietats ja podrem invertir qualsevol circuit, perquè ja coneixem les inverses de les portes bàsiques i la resta de portes es poden obtenir a partir de les bàsiques. Siguin A i B dues portes quàntiques,

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}, \qquad (AB)^{-1} = B^{-1}A^{-1}.$$

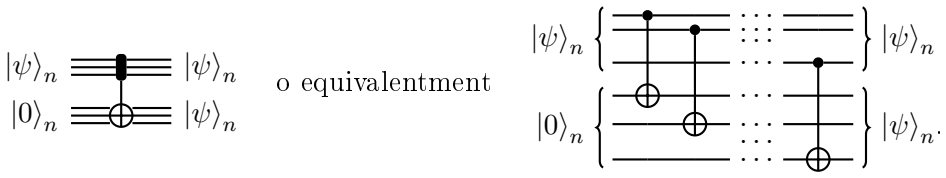
Portes no reversibles: Les portes lògiques clàssiques AND, OR i NOT són suficients per generar qualsevol operació lògica clàssica, amb la condició que es puguin duplicar bits. La porta lògica NOT és equivalent a la porta quàntica NOT, però les portes AND i OR no són reversibles i *a priori* no es poden implementar. A més, tampoc es poden duplicar qubits, ja que estem treballant en un sistema tancat per poder mantenir els estats quàntics (cf. [Be-73], [Mi,Ni-10], [Sh-97], [Wi-11]).

La clau està en usar qubits extra o auxiliars els quals inicialitzem a un estat determinat. D'aquesta manera, encara que les operacions no siguin reversibles, aquests qubits extra portaran la informació necessària per desfer aquestes operacions. A més, en comptes de duplicar qubits, podem començar amb un qubit extra i en aquest hi haurà una còpia del qubit que volem duplicar. D'aquesta manera mantenim el nombre de qubits d'entrada i de sortida.

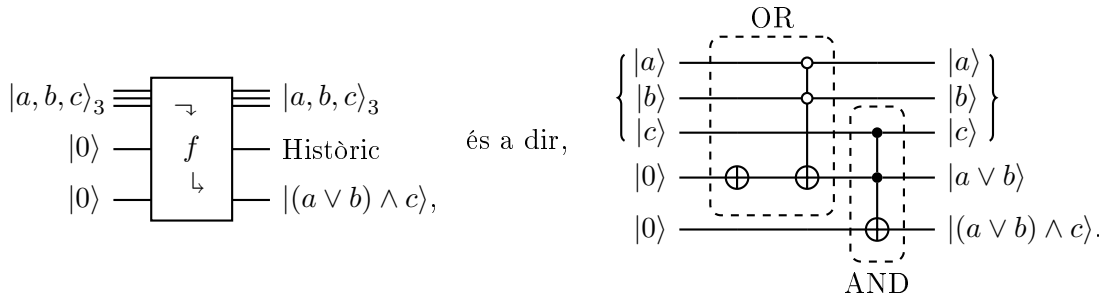
Posem $a, b \in \{0,1\}$, i calculem $\mathrm{AND}(a,b) = a \wedge b$, $\mathrm{OR}(a,b) = a \vee b$, i $\delta \mapsto (\delta, \delta)$.



Observem que podem copiar un registre sencer de n qubits a un d'auxiliar també de n qubits i inicialitzat a l'estat $|0\rangle_n$ simplement aplicant la porta CX qubit a qubit. Ho escriurem amb l'esquema de circuit

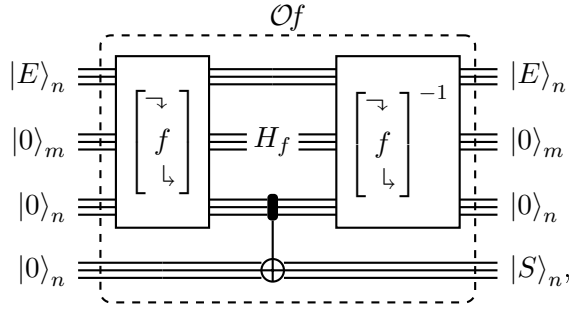


En aquestes operacions que acabem de veure (AND, OR i Copiar), tots els qubits auxiliars corresponen a la sortida, és a dir, es transformen en la sortida directament a partir de l'entrada. Però també ens podem trobar que necessitem fer càlculs entremig, i per poder-los emmagatzemar, com a mínim temporalment, utilitzarem un altre registre de qubits auxiliars inicialitzats també a cadascun a $|0\rangle$. Aquest nou registre l'anomenem històric. Per exemple, si volem calcular $f : |a,b,c\rangle_3 \mapsto |(a \vee b) \wedge c\rangle$, farem



Amb aquestes eines podem calcular qualsevol operació no reversible. Ara bé, tenim un registre addicional que no conté cap informació útil, però en comptes de llençar-lo, potser ens interessa esborrar aquests qubits històrics, per poder-los reaprofitar com a qubits auxiliars un altre cop. Fins i tot, també ens pot interessar netejar els qubits de l'entrada, i així quedar-nos només amb els de sortida.

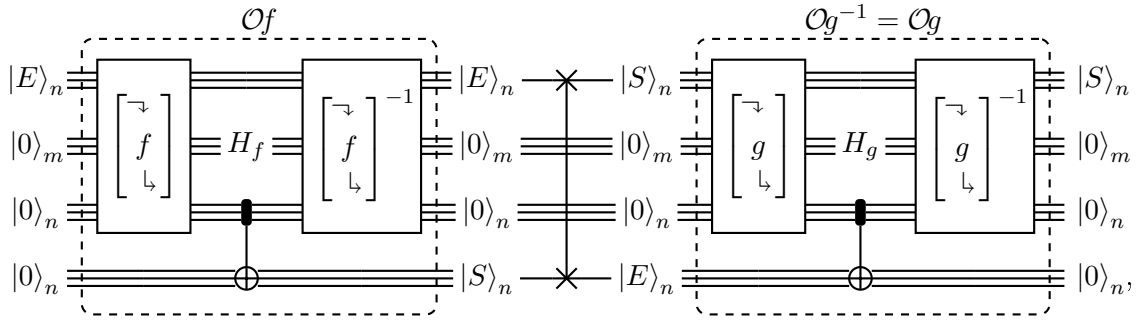
Per esborrar els qubits històrics, podem aplicar la funció i la seva inversa seguides, i entremig, copiar el resultat en un altre registre. Direm que la implementació quàntica de la funció f és aquest conjunt d'operacions, i ho denotem per $\mathcal{O}f$.



on $|E\rangle_n$ és l'entrada, $|S\rangle_n$ la sortida i H_f l'històric de f.

Si a més volem esborrar també l'entrada, ho podem fer només si l'entrada es pot calcular a partir de la sortida, és a dir, si f és bijectiva. En aquest cas, l'entrada i la sortida tindran el mateix nombre de qubits i, per tant, podrem escriure la sortida al primer registre (al registre de l'entrada).

Sigui f la funció que volem calcular i $g = f^{-1}$ la seva inversa (encara que generi històrics diferents). La idea és trobar la implementació $\mathcal{O}f$ de f i la implementació $\mathcal{O}g$ de g. Si apliquem $\mathcal{O}f$, canviem el registre de l'entrada pel de la sortida, i apliquem $\mathcal{O}g^{-1}$, que resulta ser exactament $\mathcal{O}g$; aleshores obtindrem només la sortida en el primer registre.



on el SWAP es fa qubit a qubit, similar a com hem copiat un registre sencer.

Notem que els històrics de f i de g poden no coincidir amb el nombre de qubits. Això no és cap problema, ja que podem afegir tants qubits extres com necessitem per l'històric que n'ocupi més, i ignorar els sobrants quan apliquem l'altra funció.

Per tant, ara ja podem construir una porta quàntica que realitzi una operació clàssica qualsevol.

Capítol 3

Algoritmes

3.1 Transformada quàntica de Fourier

La primera part de l'algoritme de Shor s'encarrega de trobar l'ordre r d'un element en el grup dels invertibles mòdul N.

La idea és construir, en un registre de n qubits, una superposició r-periòdica respecte els coeficients de la base canònica, $\{|0\rangle_n, |1\rangle_n, |2\rangle_n, \ldots, |2^n-1\rangle_n\}$, i utilitzar la transformada de Fourier per extreure'n informació suficient per a calcular r utilitzant un algoritme clàssic de fraccions continuades.

Notació 3.1.1. Sigui $x \in \mathbb{C}$, escriurem $e(x) := e^{2\pi i x} \in \mathbb{C}$. Observem que $e(x)$ és una funció periòdica de període 1 i $\forall m \in \mathbb{Z}$, $e(m) = 1$. A més, notem que $e\left(\frac{k}{N}\right)$ és la potència k-èsima d'una arrel primitiva N-èsima de la unitat.

Definició 3.1.2 (Transformada quàntica de Fourier)**.** Anomenem **transformada quàntica de Fourier** de n qubits a l'aplicació $\mathcal{F}_n : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$ definida per

$$|j\rangle_n \mapsto \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e\left(\frac{jk}{2^n}\right) |k\rangle_n .$$

L'objectiu ara és trobar una expressió d'aquesta superposició separada per qubits, i així facilitar la implementació en un circuit quàntic.

Notació 3.1.3. Escriurem $0.b_1 b_2 \ldots b_m := b_1 2^{-1} + b_2 2^{-2} + \cdots + b_m 2^{-m}$, on $b_i \in \{0, 1\}$.

Observem que $j2^{-s} - 0.j_{n-s+1} \ldots j_n = j_1 2^{n-1-s} + \cdots + j_{n-s} 2^{n-(n-s)-s} \in \mathbb{Z}$, i per tant, per la periodicitat de $e(x)$, obtenim que $e(j2^{-s}) = e(0.j_{n-s+1} \ldots j_n)$.

Proposició 3.1.4. *La transformada quàntica de Fourier també es pot definir pels estats dels diferents qubits individualment:*

$$|j\rangle_n = |j_1 j_2 \ldots j_n\rangle \mapsto \bigotimes_{\ell=1}^{n} \frac{|0\rangle + e(0.j_{n-\ell+1} \ldots j_n)\,|1\rangle}{\sqrt{2}}.$$

Demostració. Veiem que totes dues aplicacions envien $|j\rangle_n$ a la mateixa superposició.

$$\begin{aligned}
\frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e\left(\frac{jk}{2^n}\right)|k\rangle_n &= \frac{1}{\sqrt{2^n}}\sum_{k_1=0}^{1}\cdots\sum_{k_n=0}^{1} e\left(j\sum_{\ell=1}^{n} k_\ell 2^{n-\ell}\cdot 2^{-n}\right)|k_1\ldots k_n\rangle \\
&= \frac{1}{\sqrt{2^n}}\sum_{k_1=0}^{1}\cdots\sum_{k_n=0}^{1}\bigotimes_{\ell=1}^{n} e\left(jk_\ell 2^{-\ell}\right)|k_\ell\rangle \\
&= \frac{1}{\sqrt{2^n}}\bigotimes_{\ell=1}^{n}\left(e(0)\,|0\rangle + e\left(j2^{-\ell}\right)|1\rangle\right) \\
&= \bigotimes_{\ell=1}^{n}\frac{|0\rangle + e(0.j_{n-\ell+1}\ldots j_n)\,|1\rangle}{\sqrt{2}}.
\end{aligned}$$

□

3.2 Implementació de la transformada quàntica de Fourier

Observem de l'expressió separada per qubits que el qubit ℓ-èsim només depèn dels últims ℓ dígits de j. És a dir, $|j_\ell\rangle$ només es fa servir en el càlcul dels $n-\ell+1$ últims qubit. Aquest fet permet calcular cada qubit individualment i en ordre invers, emmagatzemant-los també en ordre invers. Al finalitzar els càlculs només caldrà invertir l'ordre dels qubits.

La superposició corresponent al ℓ-èsim qubit és

$$\frac{|0\rangle + e(0.j_{n-\ell+1}\ldots j_n)\,|1\rangle}{\sqrt{2}} = \frac{|0\rangle + e\left(j_{n-\ell+1}2^{-1}\right)\cdots e\left(j_n 2^{-\ell}\right)|1\rangle}{\sqrt{2}},$$

però, com que anirem emmagatzemant els resultats en ordre invers per només sobreescriure els valors que ja no necessitarem, la superposició del ℓ-èsim qubit abans d'invertir l'ordre és

$$\frac{|0\rangle + e(0.j_\ell\ldots j_n)\,|1\rangle}{\sqrt{2}} = \frac{|0\rangle + e\left(j_\ell 2^{-1}\right)\cdots e\left(j_n 2^{-(n-\ell+1)}\right)|1\rangle}{\sqrt{2}}.$$

Per tant, hem de trobar una combinació de portes que construeixin aquesta superposició.

Reescrivim la porta de desplaçament de fase per $R_k := R\left(\frac{2\pi}{2^k}\right) = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e\left(2^{-k}\right) \end{pmatrix}$.

Per a un qubit qualsevol $|\delta\rangle \in \{|0\rangle, |1\rangle\}$, Hadamard actua com

$$H(|\delta\rangle) = \frac{1}{2^{1/2}}\left(|0\rangle + e\left(\delta 2^{-1}\right)|1\rangle\right) = \begin{cases} \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right), & \text{si } \delta = 0, \\ \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right), & \text{si } \delta = 1, \end{cases}$$

i la porta de desplaçament de fase, controlada per un qubit $|\varepsilon\rangle \in \{|0\rangle, |1\rangle\}$, actua com

$$\begin{cases} C_\varepsilon R_k(|0\rangle) = |0\rangle, \\ C_\varepsilon R_k(|1\rangle) = e\left(\varepsilon 2^{-k}\right)|1\rangle = \begin{cases} |1\rangle, & \text{si } \varepsilon = 0, \\ e\left(2^{-k}\right)|1\rangle, & \text{si } \varepsilon = 1. \end{cases} \end{cases}$$

Observem que aplicar Hadamard al ℓ-èsim qubit genera la superposició

$$\frac{|0\rangle + e\left(j_\ell 2^{-1}\right)|1\rangle}{\sqrt{2}}.$$

Si apliquem ara la porta de desplaçament R_2 controlada pel $(\ell+1)$-èsim qubit, obtenim la superposició

$$\frac{|0\rangle + e\big(j_\ell 2^{-1}\big)\, e\big(j_{\ell+1}2^{-2}\big)\,|1\rangle}{\sqrt{2}}.$$

Seguim aplicant successivament les portes R_k per a cada $k = 3, 4, \ldots, n-\ell+1$ controlades pel $(\ell+k-1)$-èsim qubit respectivament, i obtenim la superposició

$$\frac{|0\rangle + e\big(j_\ell 2^{-1}\big)\, e\big(j_{\ell+1}2^{-2}\big)\cdots e\big(j_n 2^{-(n-\ell+1)}\big)\,|1\rangle}{\sqrt{2}},$$

que és el que volíem trobar.

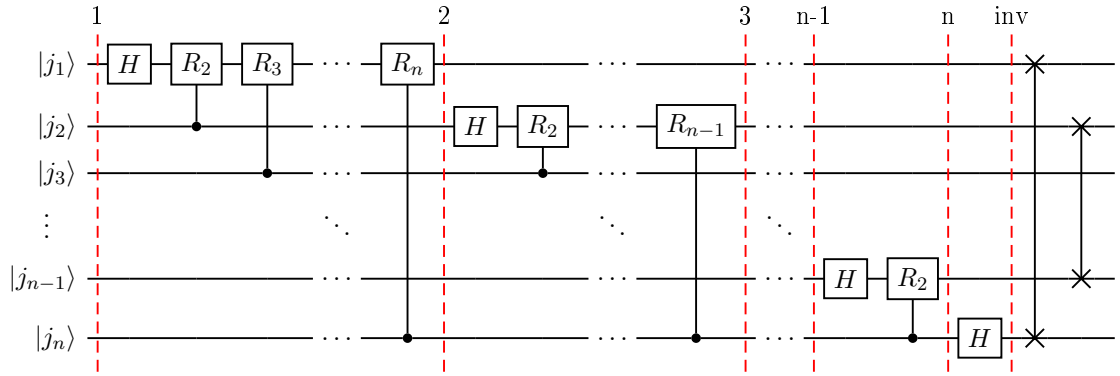
Així doncs, partint de l'estat

$$|j\rangle_n = |j_1 j_2 \ldots j_n\rangle = |j_1\rangle \otimes |j_2\rangle \otimes \cdots \otimes |j_n\rangle\,,$$

si apliquem per ordre a cada qubit ℓ la porta Hadamard seguida de les portes de desplaçament de fase $R_2, \ldots, R_{n-\ell+1}$ controlades pels qubits $\ell+1, \ldots, n$ respectivament, obtenim la superposició

$$\frac{|0\rangle + e(0.j_1 j_2 j_3 \ldots j_n)\,|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e(0.j_2 j_3 \ldots j_n)\,|1\rangle}{\sqrt{2}} \otimes \cdots \otimes \frac{|0\rangle + e(0.j_n)\,|1\rangle}{\sqrt{2}},$$

i per tant, només ens faltarà intercanviar l'ordre dels qubits per a obtenir la transformada quàntica de Fourier. Per fer-ho podem aplicar la porta SWAP $\left\lfloor \frac{n}{2} \right\rfloor$ vegades, intercanviant els qubits ℓ i $n-\ell$, per a tot $\ell \leq \frac{n}{2}$.



La quantitat de portes Hadamard és d'una per qubit, de desplaçament de fase és de $n-\ell$ per a cada qubit ℓ i al final efectuem $\left\lfloor \frac{n}{2} \right\rfloor$ portes SWAP. Per tant, el total de portes quàntiques bàsiques és

$$\sum_{\ell=1}^{n} 1 + \sum_{\ell=1}^{n}(n-\ell) + \left\lfloor \frac{n}{2} \right\rfloor = n + \frac{n(n-1+n-n)}{2} + \left\lfloor \frac{n}{2} \right\rfloor \in O\left(n^2\right).$$

Podem concloure doncs que la transformada quàntica de Fourier de n qubits, $\mathcal{F}_n$, es pot implementar usant $O(n^2)$ portes quàntiques elementals (cf. [Co-94]).

3.3 Estimació de fase

El problema de l'estimació de fase proporcionarà un algoritme que és la clau per a entendre l'algoritme de Shor.

Suposem que tenim, en un sistema de m qubits, un operador unitari U i un vector propi $|u\rangle_m$ de valor propi $e(\varphi) = e^{2\pi i\varphi}$, per a algun valor desconegut de φ el qual volem estimar. Podem suposar que $0 \leq \varphi < 1$, per la periodicitat de $e(x)$.

Cal notar que el valor de φ només el podrem trobar de manera exacta si $\varphi 2^t \in \mathbb{N}$ per a algun t. Això és degut al fet que en computació es treballa en binari, i per tant, l'estimació de φ només la podem aconseguir en base 2. Això significa que el valor buscat serà una aproximació $\widetilde{\varphi}$ de φ tal que $\widetilde{\varphi}2^t \in \mathbb{N}$.

Podem interpretar la transformada quàntica de Fourier com una aplicació que envia una codificació de les arrels 2^t-èsimes de la unitat a una superposició amb les seves potències com a coeficients.

$$e\left(\frac{j}{2^n}\right) \rightsquigarrow |j\rangle_n \mapsto \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e\left(\frac{jk}{2^n}\right) |k\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e\left(\frac{j}{2^n}\right)^k |k\rangle_n ,$$

$$e(\widetilde{\varphi}) \rightsquigarrow \left|\widetilde{\varphi}2^t\right\rangle_t \mapsto \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^n-1} e\left(\frac{k\widetilde{\varphi}2^t}{2^t}\right) |k\rangle_t = \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} e(\widetilde{\varphi})^k |k\rangle_t .$$

Si d'alguna manera aconseguim construir la superposició de la dreta, aplicant la inversa de la transformada (quàntica) de Fourier obtindríem la fase (o angle) de l'arrel en qüestió.

Suposem que sabem construir l'estat $|u\rangle_m$ i computar les operacions CU^{2^η} per a $0 \leq \eta < t$, on $t \in O(m)$ i

$$U^{2^\eta} |u\rangle_m = e(\varphi)^{2^\eta} |u\rangle_m = e^{2\pi i 2^\eta \varphi} |u\rangle_m = e(2^\eta\varphi) |u\rangle_m .$$

L'objectiu és construir la superposició

$$\frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} e(k\varphi) |k\rangle_t = \bigotimes_{\ell=1}^{t} \frac{|0\rangle + e\big(2^{t-\ell}\varphi\big) |1\rangle}{\sqrt{2}}.$$

Per fer-ho, utilitzarem dos registres: El primer consistirà de t qubits inicialitzats cadascun a $|0\rangle$, i al segon construirem l'estat $|u\rangle_m$ amb m qubits. Al primer registre és on construirem la superposició desitjada. El valor de t el triarem més endavant, i servirà per a determinar la precisió de l'aproximació i la probabilitat d'èxit de l'algoritme.

Veiem primer quin és el resultat d'aplicar la porta $C_+U^{2^\eta}$ a $|u\rangle$, és a dir, controlada per un qubit inicialitzat en l'estat $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$.

$$C_+U^{2^\eta} \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |u\rangle_m\right) = \frac{|0\rangle \otimes |u\rangle_m + |1\rangle \otimes U^{2^\eta} |u\rangle_m}{\sqrt{2}} =$$

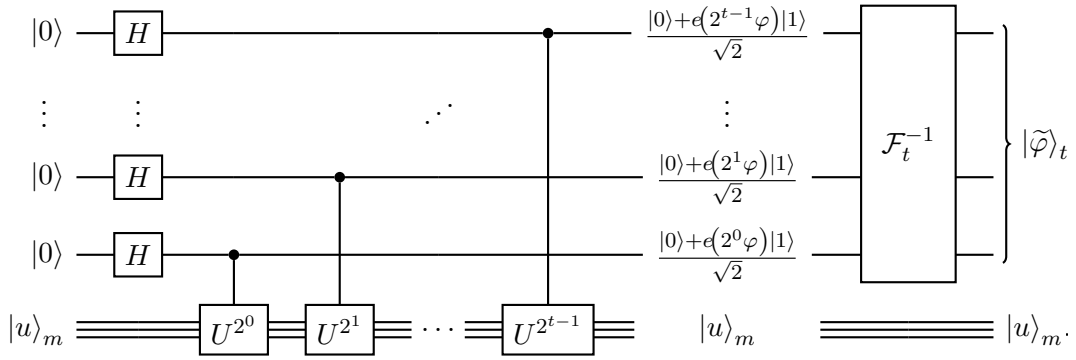
$$= \frac{|0\rangle \otimes |u\rangle_m + |1\rangle \otimes e(2^\eta\varphi) |u\rangle_m}{\sqrt{2}} = \frac{|0\rangle + e(2^\eta\varphi) |1\rangle}{\sqrt{2}} \otimes |u\rangle_m .$$

Observem que el qubit de control manté les probabilitats però queda desplaçat de fase i, en canvi, el registre $|u\rangle_m$ queda invariant. De fet, el qubit de control coincideix amb l'estat desitjat del qubit ℓ-èsim quan $\eta = t - \ell$.

Per tant, si inicialitzem cada qubit del primer registre a l'estat $|+\rangle$ i apliquem les portes $C_+U^{2^{t-\ell}}$ usant el ℓ-èsim qubit del primer registre com a qubit de control, obtindrem la superposició desitjada al primer registre i l'estat $|u\rangle_m$ al segon. Apliquem la inversa de la transformada de Fourier al primer registre per a obtenir la fase al mesurar, i mesurem i ignorem el segon registre perquè no aporta cap informació útil.

Per a construir els estats $|+\rangle$ podem inicialitzar cada qubit del primer registre a l'estat $|0\rangle$ i aplicar la porta Hadamard a cadascun d'ells.

Així doncs, l'esquema del circuit de l'algoritme de l'estimació de fase és el següent:



La quantitat de portes Hadamard és t, una per qubit del primer registre; de CU^{2^η} n'hi ha també t, una per qubit (de control) del primer registre, i finalment, una inversa de la transformada quàntica de Fourier de t qubits, que gasta $O(t^2)$ portes bàsiques. Considerem que les portes U^{2^η} tenen una complexitat $O(c(m))$, és a dir, que es poden implementar amb $O(c(m))$ portes elementals. Per tant, el total de portes quàntiques elementals és

$$t + t \cdot O(c(m)) + O(t^2) \in O(t^2 + t \cdot c(m)) = \begin{cases} O(t^2) & \text{si } O(c(m)) \subset O(t), \\ O(t \cdot c(m)) & \text{si } O(t) \subset O(c(m)). \end{cases}$$

3.4 Precisió i probabilitat d'èxit

Ara que ja sabem com implementar l'algoritme, només falta determinar t per a controlar la precisió de $\widetilde{\varphi}$ i la probabilitat de trobar tal aproximació.

Comencem fent el càlcul formal de la inversa de la transformada quàntica de Fourier. Així podrem calcular exactament la superposició que s'obté abans de mesurar. Veiem que la matriu corresponent a $\mathcal{F}_n$ és simètrica, i per tant, per a trobar la seva inversa simplement haurem de conjugar.

$$\mathcal{F}_n^{-1} = \mathcal{F}_n^\dagger = \frac{1}{\sqrt{2^n}} \left(e\left(\frac{jk}{2^n}\right) \right)_{j,k}^\dagger = \frac{1}{\sqrt{2^n}} \left(\overline{e\left(\frac{jk}{2^n}\right)} \right)_{k,j} = \frac{1}{\sqrt{2^n}} \left(e\left(\frac{-jk}{2^n}\right) \right)_{j,k}$$

$$\Longrightarrow \mathcal{F}_n^{-1} : |j\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e\left(\frac{-jk}{2^n}\right) |k\rangle \,.$$

Apliquem doncs la transformada $\mathcal{F}_t^{-1}$ al primer registre després d'haver aplicat el conjunt de portes H i CU^{2^η}.

$$\mathcal{F}_t^{-1}\left(\frac{1}{2^{t/2}}\sum_{k=0}^{2^t-1} e(k\varphi)\,|k\rangle\right) = \frac{1}{2^t}\sum_{k=0}^{2^t-1}\sum_{\ell=0}^{2^t-1} e\left(k\varphi - \frac{k\ell}{2^t}\right)|\ell\rangle\,.$$

L'objectiu és calcular quin valor de t és necessari per a garantir una *alta* probabilitat de mesurar una *bona* aproximació. Considerarem que una probabilitat és *alta* si aquesta és superior a $1-\varepsilon$, per al valor de $\varepsilon \geq 0$ que es vulgui imposar; i també que un valor és una *bona* aproximació si la precisió és de 2^{-d}, per a algun nombre natural $d \in \mathbb{N}$ que representa la quantitat de dígits (binaris) de precisió que vulguem.

És a dir, si el valor mesurat és $\ell = \widetilde{\varphi}2^t$, aleshores considerem que és una *bona* aproximació si es compleix que $\left|\varphi - \ell 2^{-t}\right| = |\varphi - \widetilde{\varphi}| < 2^{-d}$, que és equivalent a $\left|\varphi 2^t - \ell\right| < 2^{t-d}$; per a algun representant de $\varphi \in \frac{\mathbb{R}}{\mathbb{Z}}$. Per aquest motiu convé triar un representant adient per a realitzar els càlculs. Podem fixar $0 \leq \varphi < 1$ i considerar els representants de ℓ mòdul 2^t centrats a $\varphi 2^t$.

Denotem per α_ℓ al coeficient de $\left|\ell \pmod{2^t}\right\rangle$ de la superposició del primer registre abans de mesurar.

$$\alpha_\ell := \frac{1}{2^t}\sum_{k=0}^{2^t-1} e\left(k\varphi - \frac{k\ell}{2^t}\right) = \frac{1}{2^t}\sum_{k=0}^{2^t-1} e\left(\varphi - \frac{\ell}{2^t}\right)^k = \frac{1}{2^t}\cdot\frac{1-e\left(\varphi-\frac{\ell}{2^t}\right)^{2^t}}{1-e\left(\varphi-\frac{\ell}{2^t}\right)} = \frac{1}{2^t}\cdot\frac{1-e\left(\varphi 2^t-\ell\right)}{1-e(\varphi-\ell 2^{-t})}.$$

La probabilitat de mesurar $\left|\ell \pmod{2^t}\right\rangle$ és $|\alpha_\ell|^2$. Per a acotar la probabilitat de fracàs haurem d'acotar $|\alpha_\ell|$ per als valors ℓ tals que $\left|\varphi 2^t - \ell\right| < 2^{t-d}$. Ens aprofitarem de dues propietats geomètriques:

$$\begin{cases} \forall\theta \in \mathbb{R},\ \left|1-e^{i\theta}\right| \leq 2, \\ \forall\theta \in \mathbb{R},\ -\pi \leq \theta \leq \pi \implies \left|1-e^{i\theta}\right| \geq \frac{2}{\pi}\,|\theta|\,. \end{cases}$$

Com que els valors ℓ són representants mòdul 2^t centrats a $\varphi 2^t$, aleshores es compleix que $-2^{t-1} \leq \varphi 2^t - \ell \leq 2^{t-1}$, que implica que $-\pi \leq 2\pi(\varphi - \ell 2^{-t}) \leq \pi$. Per tant, podem aplicar la primera propietat al numerador i la segona al denominador.

$$|\alpha_\ell| = \frac{1}{2^t}\cdot\frac{\left|1-e\left(\varphi 2^t-\ell\right)\right|}{|1-e(\varphi-\ell 2^{-t})|} \leq \frac{1}{2^t}\cdot\frac{2}{\frac{2}{\pi}\,|2\pi(\varphi-\ell 2^{-t})|} = \frac{1}{2\cdot|\varphi 2^t-\ell|}.$$

Així doncs, calculem una cota de la probabilitat de fracàs ε.

$$\begin{aligned}
\varepsilon &= P\left(\ell : \left|\varphi 2^t - \ell\right| \geq 2^{t-d}\right) = \sum_{\ell:|\varphi 2^t-\ell|\geq 2^{t-d}} |\alpha_\ell|^2 = \\
&= \sum_{\ell=\lfloor\varphi 2^t\rfloor-2^{t-1}+1}^{\lfloor\varphi 2^t\rfloor-2^{t-d}} |\alpha_\ell|^2 + \sum_{\ell=\lceil\varphi 2^t\rceil+2^{t-d}}^{\lfloor\varphi 2^t\rfloor+2^{t-1}} |\alpha_\ell|^2 \leq \\
&\leq \sum_{\ell=2^{t-d}}^{2^{t-1}-1} \left|\alpha_{\lfloor\varphi 2^t\rfloor-\ell}\right|^2 + \sum_{\ell=2^{t-d}}^{2^{t-1}} \left|\alpha_{\lceil\varphi 2^t\rceil+\ell}\right|^2 = \\
&= \sum_{\ell=2^{t-d}}^{2^{t-1}-1} \frac{1}{2^2\cdot|\varphi 2^t-(\lfloor\varphi 2^t\rfloor-\ell)|^2} + \sum_{\ell=2^{t-d}}^{2^{t-1}} \frac{1}{2^2\cdot|\varphi 2^t-(\lceil\varphi 2^t\rceil+\ell)|^2}.
\end{aligned}$$

Com que $\forall x \in \mathbb{R}$, $-\ell \leq -1 < x - \lceil x \rceil \leq 0 \leq x - \lfloor x \rfloor < 1 \leq \ell$, aleshores es compleix que

$$\begin{cases} \left|\varphi 2^t - \left\lfloor \varphi 2^t \right\rfloor + \ell\right| \geq |0 + \ell| = |\ell|\,, \\ \left|\varphi 2^t - \left\lceil \varphi 2^t \right\rceil - \ell\right| \geq |0 - \ell| = |\ell|\,, \end{cases}$$

i com que $\frac{1}{\ell^2}$ és una funció decreixent per a $\ell > 0$, podem acotar les sumes per una integral.

$$\varepsilon \leq \sum_{\ell=2^{t-d}}^{2^{t-1}-1} \frac{1}{2^2 \cdot |\ell|^2} + \sum_{\ell=2^{t-d}}^{2^{t-1}} \frac{1}{2^2 \cdot |\ell|^2} \leq \frac{2}{2^2} \int_{2^{t-d}-1}^{2^{t-1}} \frac{d\ell}{\ell^2} < \frac{1}{2} \int_{2^{t-d}-1}^{\infty} \frac{d\ell}{\ell^2} = \frac{1}{2\,(2^{t-d}-1)}.$$

Hem calculat una cota de la probabilitat de fracàs en funció de la precisió d i la quantitat de qubits t usats. Si aïllem t, podem concloure que si apliquem l'algoritme amb

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

qubits al primer registre, obtindrem una aproximació de φ amb una precisió de 2^{-d} i una probabilitat d'èxit superior a $1 - \varepsilon > 1 - \frac{1}{2\left(2^{t-d}-1\right)}$.

3.5 Fraccions continuades

El mètode de fraccions continuades permet trobar dos enters coprimers tals que el seu quocient sigui una aproximació d'un nombre (real) conegut. Definim primer què és una fracció continuada i estudiem quines propietats té.

Definició 3.5.1 (Fraccions continuades)**.** Donat un $x \in \mathbb{R}$, la **fracció continuada** de x és la successió entera $\{a_k\}_k \subset \mathbb{Z}$ tal que

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots + \cfrac{1}{a_k + \ddots}}}}$$

que podem definir per $a_k := \lfloor x_k \rfloor$, on $\{x_k\}_k$ és la successió real donada per

$$\begin{cases} x_0 := x, \\ x_{k+1} := \dfrac{1}{x_k - \lfloor x_k \rfloor} \end{cases} \quad \forall k \in \mathbb{N}, \text{ fins que } x_k = \lfloor x_k \rfloor\,.$$

Notació 3.5.2. Per a tot $x \in \mathbb{R}$ escriurem $[a_0, a_1, \ldots, a_k, \ldots] := x =: [a_0, a_1, \ldots, a_k, x_{k+1}]$.

Definició 3.5.3 (Convergents)**.** Sigui $x = [a_0, a_1, \ldots, a_k, \ldots]$. El **$k$-èsim convergent** de x és $[a_0, a_1, \ldots, a_k]$.

Veiem ara alguns resultats que trobem demostrats a [Ha,Wr-79, Chap. X] i [Tr-98, VIII §4].

Proposició 3.5.4 ([Ha,Wr-79, Th.149])**.** *Siguin $x \in \mathbb{R}$ i $[a_0, \ldots, a_k, \ldots] = x$ la seva fracció continuada. Siguin les successions enteres $\{b_k\}_k$ i $\{c_k\}_k$ definides per*

$$\begin{cases} b_0 := a_0, & b_1 := a_0 a_1 + 1, & \ldots, & b_{k+2} := a_{k+2} b_{k+1} + b_k, \\ c_0 := 1, & c_1 := a_1, & \ldots, & c_{k+2} := a_{k+2} c_{k+1} + c_k. \end{cases}$$

Aleshores, el k-èsim convergent és $[a_0, a_1, \ldots, a_k] = \dfrac{b_k}{c_k}$ i b_k i c_k són coprimers entre si. □

Proposició 3.5.5 ([Ha,Wr-79, Th.150,154]). *Siguin $x \in \mathbb{R}$ i $\{b_k/c_k\}_k$ els convergents de x. Aleshores*

$$\frac{1}{c_k c_{k+1}} = \left| \frac{b_k}{c_k} - \frac{b_{k+1}}{c_{k+1}} \right|$$

i x es troba sempre entremig de dos convergents consecutius qualssevol. □

Proposició 3.5.6 ([Ha,Wr-79, Th.184]). *Siguin $x \in \mathbb{R}$, $p, q \in \mathbb{Z}$ tals que*

$$\left| \frac{p}{q} - x \right| \leq \frac{1}{2q^2}.$$

Aleshores p/q és un convergent de la fracció continuada de x. □

Proposició 3.5.7. *Siguin $x \in \mathbb{R}$, $p, q \in \mathbb{Z}$, $m, d \in \mathbb{N}$ tals que $q < 2^m$, $d = 2m + 1$ i*

$$\left| \frac{p}{q} - x \right| \leq \frac{1}{2^d}.$$

Aleshores p/q és el k-èsim convergent de x tal que $c_k < 2^m < c_{k+1}$.

Demostració. Per la proposició 3.5.6 tenim que p/q és un convergent, ja que

$$\left| \frac{p}{q} - x \right| \leq \frac{1}{2^d} = \frac{1}{2\,(2^m)^2} \leq \frac{1}{2q^2}.$$

Si posem $p/q = b_k/c_k$, per la proposició 3.5.5 tenim que

$$\frac{1}{c_k c_{k+1}} = \left| \frac{b_k}{c_k} - \frac{b_{k+1}}{c_{k+1}} \right| = \left| \frac{b_k}{c_k} - x \right| + \left| \frac{b_{k+1}}{c_{k+1}} - x \right| \leq 2 \left| \frac{b_k}{c_k} - x \right| \leq 2\frac{1}{2^d} < \frac{1}{2^m c_k},$$

i per tant, $c_k < 2^m < c_{k+1}$. Com que la successió $\{c_k\}_k$ és creixent, b_k/c_k és l'únic convergent que ho compleix. □

El mètode de fraccions continuades permet trobar un nombre racional $p/q \in \mathbb{Q}$ a partir d'una aproximació $x \in \mathbb{R}$ si aquest compleix les condicions de la proposició 3.5.7. El mètode consisteix a anar calculant els termes a_k de la fracció continuada de x i alhora també calcular els termes b_k i c_k, és a dir, els convergents de x. Quan s'arriba a calcular un denominador tal que $2^m < c_{k+1}$, aleshores la proposició 3.5.7 ens garanteix que el k-èsim convergent b_k/c_k és el nombre racional p/q buscat.

Observem que tal com hem definit la successió $\{c_k\}_k$, és immediat que c_k és més gran que el k-èsim nombre de Fibonacci $F(k)$. A més, tenim una fórmula que ens permet calcular l'índex k a partir del nombre $F(k)$:

$$k = \left\lfloor \log_\varphi \left(F(k) \cdot \sqrt{5} + \frac{1}{2} \right) \right\rfloor.$$

Per tant, una cota superior de k en funció de $c_k = q$ és

$$k \leq \left\lfloor \log_\varphi \left(q \cdot \sqrt{5} + \frac{1}{2} \right) \right\rfloor \in O(\log\, q)\,.$$

Com que cada pas té una complexitat clàssica $O\Big((\log\, q)^2\Big)$, ja que s'utilitza l'algoritme d'Euclides (cf. [Co-93, §1.3]), podem garantir que el mètode de fraccions continuades té una complexitat clàssica $O\Big((\log\, q)^3\Big)$.

3.6 Cerca de l'ordre d'un element en un grup abelià finit

La part quàntica de l'algoritme de factorització de Shor consisteix a trobar l'ordre d'un element en el grup $\left(\frac{\mathbb{Z}}{N\mathbb{Z}}\right)^*$, on N és el nombre a factoritzar.

Donats un grup multiplicatiu M i l'element $b \in M$ del qual volem calcular l'ordre r, suposem que tenim una manera de codificar els elements de M en la base canònica d'un sistema de $m = O\left(log\,|M|\right)$ qubits; i que podem operar-hi a partir de matrius unitàries. Escriurem $|h\rangle_m$ per referir-nos a l'estat corresponent a la codificació de $h \in M$.

Definim l'aplicació $U_b : |h\rangle \mapsto |bh\rangle$. Com que $b \in M$ té invers b^{-1} (M és un grup), aleshores $|h\rangle \mapsto \left|b^{-1}h\right\rangle$ és la inversa de U_b, i per tant U_b es podrà implementar en un sistema quàntic per ser unitària.

L'estratègia a seguir serà buscar vectors propis de U_b amb valors propis de la forma $e(\varphi)$ tals que r sigui fàcil de calcular a partir de φ, i així poder usar l'algoritme d'estimació de fase per a trobar una aproximació de l'ordre r.

Definim els estats

$$|u_s\rangle_m := \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e\left(\frac{-sk}{r}\right) \left|b^k\right\rangle_m$$

i veiem que són vectors propis de U_b, ja que $b^0 = 1 = b^r$, i $e\left(\frac{-s\cdot r}{r}\right) = 1 = e\left(\frac{-s\cdot 0}{r}\right)$.

$$U_b\,|u_s\rangle_m = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e\left(\frac{-sk}{r}\right) U_b \left|b^k\right\rangle_m = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e\left(\frac{-sk}{r}\right) \left|b^{k+1}\right\rangle_m =$$

$$= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e\left(\frac{-s(k-1)}{r}\right) \left|b^k\right\rangle_m = e\left(\frac{s}{r}\right) \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e\left(\frac{-sk}{r}\right) \left|b^k\right\rangle_m = e\left(\frac{s}{r}\right) |u_s\rangle_m\,,$$

i per tant, són de valor propi $e\left(\frac{s}{r}\right)$.

Per a construir algun d'aquests estats $|u_s\rangle_m$ hauríem de conèixer el valor de r, que és precisament el que volem calcular. Ara bé, podem adaptar l'algoritme de l'estimació de fase per a una combinació lineal de vectors propis.

Observem que $\forall s, k$, $e\left(\frac{-sk}{r}\right)$ és una arrel r-èsima de la unitat, i per tant, fixada una k, la suma de totes les potències s-èsimes de $e\left(\frac{-k}{r}\right)$ s'anul·la excepte si $e\left(\frac{-k}{r}\right) = 1$, és a dir, excepte si $k = 0$.

$$\sum_{s=0}^{r-1} e\left(\frac{-sk}{r}\right) = r \cdot \delta_{k,0} = \begin{cases} \sum_{s=0}^{r-1} 1 = r, & \text{si } k = 0, \\ 0, & \text{si } k \neq 0, \end{cases}$$

on $\delta_{i,j}$ és la delta de Kronecker.

Amb això veiem que la suma de tots els estats $|u_s\rangle_m$ que hem definit és igual a $|1\rangle_m$, que és fàcil d'implementar per ser simplement la codificació del neutre:

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle_m = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e\left(\frac{-sk}{r}\right) \left|b^k\right\rangle_m = \sum_{k=0}^{r-1} \frac{1}{r} \left(\sum_{s=0}^{r-1} e\left(\frac{-sk}{r}\right)\right) \left|b^k\right\rangle_m =$$

$$= \sum_{k=0}^{r-1} \frac{1}{r} r\delta_{k,0} \left|b^k\right\rangle_m = \sum_{k=0}^{r-1} \delta_{k,0} \left|b^k\right\rangle_m = \left|b^0\right\rangle_m = |1\rangle_m\,.$$

Per acabar d'adaptar l'algoritme d'estimació de fase, veiem com afecta la seqüència de portes Hadamard i $CU_b^{2^\eta}$, i quina superposició queda després d'aplicar la inversa de la transformada quàntica de Fourier.

Comencem amb l'estat inicial $|0\rangle_t \otimes |1\rangle_m$, on el primer registre consta de t qubits per a algun t prou gran (a determinar) i el segon registre usa m qubits, els necessaris per a fer les codificacions dels elements de M. Després d'aplicar les portes Hadamard queda l'estat

$$\frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle_t \otimes |1\rangle_m .$$

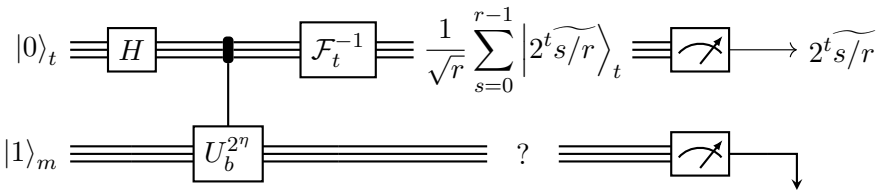
El conjunt de les portes $CU_b^{2^\eta}$, en l'ordre descrit a l'apartat anterior, actua als dos registres com $|k\rangle_t |h\rangle_m \mapsto |k\rangle_t U_b^k |h\rangle_m$, i per tant obtenim la superposició

$$\frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle_t \otimes U_b^k |1\rangle_m = \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle_t \otimes U_b^k \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle_m = \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle_t \otimes \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} U_b^k |u_s\rangle_m =$$

$$= \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle_t \otimes \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e\Big(\frac{s}{r}\Big)^k |u_s\rangle_m = \frac{1}{\sqrt{2^t r}} \sum_{k=0}^{2^t-1} \sum_{s=0}^{r-1} e\left(\frac{sk}{r}\right) |k\rangle_t \otimes |u_s\rangle_m .$$

En aplicar la inversa de la transformada quàntica de Fourier obtenim una superposició quasi equiprobable d'aproximacions dels diferents valors $\frac{s}{r}$ per a tot s.

Si mesurem, obtindrem una aproximació de $\frac{s}{r}$ per a algun s, que si resulta ser coprimer amb r, podem calcular aquest valor usant el mètode de fraccions continuades.



De fet, ja no és necessari usar la inversa de la transformada quàntica de Fourier. Aplicant la transformada sense invertir obtenim el mateix resultat. Això és degut al fet que la diferència entre la inversa i l'original és només un signe i al fet que la superposició que tenim és la suma de tots els valors que identifiquen $\frac{s}{r}$ mòdul $\mathbb{Z}$. Per tant, els conjunts dels valors amb alta probabilitat de ser mesurats (en haver aplicat la inversa i l'original respectivament) són el mateix:

$$\left\{ \frac{s}{r} \pmod 1 : s \in \mathbb{Z}, 0 \le s < r \right\} = \left\{ \frac{-s}{r} \pmod 1 : s \in \mathbb{Z}, 0 \le s < r \right\} .$$

Ara toca determinar t segons la precisió d necessària i la probabilitat d'èxit $1 - \varepsilon$ mínima d'obtenir un resultat amb tal precisió.

Per a trobar els valors r i s a partir de l'aproximació $\widetilde{s/r}$ podem aplicar el mètode de fraccions continuades descrit a la secció 3.5; i per tant, necessitem una precisió $d = 2m+1$, on m és tal que $r < 2^m$. Podem triar $m = \lceil \log |M| \rceil$, ja que $r < |M| < 2^m$. La complexitat de fraccions continuades en aquest cas és $O\Big((\log r)^3\Big) = O\big(m^3\big)$.

Ara bé, el mètode de fraccions continuades retorna la fracció irreductible del convergent corresponent. Per tant, si s i r tenen algun factor en comú, obtindrem dos nombres coprimers s' i r' tals que $s/r = s'/r'$, i per tant, $s' \mid s$ i $r' \mid r$.

Obtindrem $r' = r$ amb probabilitat $\frac{\varphi(r)}{r}$, on $\varphi(\cdot)$ és la funció phi d'Euler. Però en cas que obtinguem un valor $r' < r$, aleshores podem repetir l'algoritme usant $b^{r'}$ en comptes de b, ja que aleshores la probabilitat de trobar l'ordre de $b^{r'}$, r/r', serà $\frac{\varphi(r/r')}{r/r'} \geq \frac{\varphi(r)}{r}$. Ara bé, necessitem el valor r' per a repetir l'algoritme, cosa que impedeix executar-lo en paral·lel.

Tanmateix, podem aplicar l'algoritme k vegades utilitzant b i obtenir $\widetilde{s_1/r}, \widetilde{s_2/r}, \ldots, \widetilde{s_k/r}$, amb els seus valors corresponents $s'_1, \ldots, s'_k; r'_1, \ldots, r'_k$ tals que $\forall i, \frac{s'_i}{r'_i} = \frac{s_i}{r}$ i $\operatorname{mcd}(s'_i, r'_i) = 1$. Aleshores podem calcular $r = \operatorname{mcm}(r'_1, \ldots, r'_k)$ si i només si $\operatorname{mcd}(r, s_1, \ldots, s_k) = 1$.

Calculem, doncs, una cota de la probabilitat de fracàs ϵ en funció de k. Notem que per a tot primer p hi ha $\left\lfloor \frac{r}{p} \right\rfloor$ múltiples de p entre 0 i $r-1$.

$$\begin{aligned} \epsilon &= P\left(\operatorname{mcd}(r, s_1, \ldots, s_k) > 1\right) = \sum_{p \text{ primer}} P\left(p \mid r\right) \prod_{i=1}^{k} P\left(p \mid s_i\right) \leq \\ &\leq \sum_{p \text{ primer}} \prod_{i=1}^{k} \frac{r/p}{r} = \sum_{p \text{ primer}} \frac{1}{p^k} < \sum_{\ell=2}^{\infty} \frac{1}{\ell^k} = \sum_{\ell=1}^{\infty} \frac{1}{\ell^k} - 1 = \zeta(k) - 1, \end{aligned}$$

on $\zeta(\cdot)$ denota la funció zeta de Riemann.

Per tant, la probabilitat d'obtenir r a partir de k repeticions de l'algoritme és com a mínim de $1 - \epsilon > 2 - \zeta(k)$, que no depèn de la mida del problema; és a dir, té una probabilitat d'èxit de $O(1)$. Alguns valors de $1 - \epsilon$ corresponents a valors de k són:

$k =$	2	3	4	5	$\cdots$	7	$\cdots$	10	$\cdots$
$1 - \epsilon >$	0.355066	0.797943	0.917677	0.963072	$\cdots$	0.991651	$\cdots$	0.999005	$\cdots$

Així doncs, l'algoritme del càlcul de l'ordre en un grup M té una complexitat polinòmica respecte del nombre de bits de $|M|$ i una probabilitat d'èxit constant. Més concretament, l'algoritme per trobar l'ordre d'un element en un grup M utilitza $t \in O(\log|M|)$ qubits i $O\Big(\left(\log|M|\right)^3\Big)$ portes quàntiques bàsiques, la part clàssica té una complexitat $O\Big(\left(\log|M|\right)^3\Big)$ i la probabilitat d'èxit de l'algoritme és constant, $(1-\varepsilon)(1-\epsilon) \in O(1)$.

3.7 Factorització (algoritme de Shor)

Ara que ja sabem calcular l'ordre d'un element d'un grup abelià finit, anem a veure com implementar-ho per a factoritzar un enter $N \in \mathbb{N}$. Com hem comentat a la secció 1.2, el problema de factorització redueix al problema de trobar un factor no trivial de qualsevol enter compost (menor o igual que N), i és aquest el problema que realment resoldrem. Veurem que la complexitat és polinòmica respecte el nombre de bits de N.

El grup que usarem serà el grup dels invertibles mòdul N, on N és el nombre a factoritzar. Posem $n > 0$ al menor enter tal que $N < 2^n$, és a dir, $n = \lceil \log_2 N \rceil$, i serà aquest el nombre de qubits suficients per a codificar els elements de $\left(\frac{\mathbb{Z}}{N\mathbb{Z}}\right)^*$, ja que $\varphi(n) < N$. Per cada classe mòdul N triem l'enter representant $m \in \mathbb{Z}$ tal que $0 \leq m < N$, i la codificació serà $[m] \rightsquigarrow |m\rangle_n := |m \pmod N\rangle_n$.

Així doncs, comencem triant un nombre a l'atzar del conjunt $\{2, 3, \ldots, N-1\}$, amb distribució uniforme. Si no fos invertible mòdul N, vol dir que comparteix un factor no trivial amb N i el podem calcular mitjançant l'algoritme d'Euclides. Però poques vegades tindrem tanta sort.

Un cop seleccionat aquest enter, x, calculem l'ordre r mitjançant l'algoritme descrit a la secció anterior:

Usarem dos registres inicialitzats a $|0\rangle_t \otimes |1\rangle_n$; el primer de t qubits, $t > 2n+1$ i determinat segons la probabilitat d'èxit desitjada, i el segon de n qubits.

Definim l'aplicació $U_x : |h\rangle_n \mapsto |hx\rangle_n$, i ens interessa poder implementar el conjunt de les portes $CU_x^{2^\eta}$ que actua en els dos registres com $|k\rangle_t |h\rangle_n \mapsto |k\rangle_t U_x^k |h\rangle_n = |k\rangle_t \left|hx^k\right\rangle_n$.

Observem que podem expressar qualsevol nombre $k \in \mathbb{Z}$ entre 0 i $2^t - 1$ com a suma d'un múltiple de r i un nombre entre 0 i $r-1$ de manera única. Així doncs, escriurem $\forall k \in \mathbb{Z},\ 0 \leq k < 2^t,\ k = ar + b$ per a alguns $a, b \in \mathbb{Z}$ tals que $0 \leq a < 2^t/r$ i $0 \leq b < r$.

Després d'aplicar les portes Hadamard al primer registre, apliquem les portes $CU_x^{2^\eta}$, i obtenim la superposició

$$CU_x^{2^\eta}\left(H|0\rangle_t \otimes |1\rangle_n\right) = CU_x^{2^\eta}\left(\frac{1}{\sqrt{2^t}}\sum_{k=0}^{2^t-1}|k\rangle_t \otimes |1\rangle_n\right) = \frac{1}{\sqrt{2^t}}\sum_{k=0}^{2^t-1}|k\rangle_t \otimes U_x^k|1\rangle =$$

$$= \frac{1}{\sqrt{2^t}}\sum_{k=0}^{2^t-1}|k\rangle_t \left|x^k\right\rangle_n = \frac{1}{\sqrt{2^t}}\sum_{a=0}^{\left\lfloor\frac{2^t}{r}\right\rfloor-1}\sum_{\substack{b=0\\ ar+b<2^t}}^{r-1}|ar+b\rangle_t \left|x^{ar+b}\right\rangle_n = \frac{1}{\sqrt{2^t}}\sum_{b=0}^{r-1}\sum_{\substack{a=0\\ ar+b<2^t}}^{\left\lfloor\frac{2^t}{r}\right\rfloor-1}|ar+b\rangle_t \left|x^b\right\rangle_n.$$

El segon registre ja ha fet la seva feina. Com hem comentat abans, podem mesurar-lo i ignorar-ne el resultat, ja que no aporta cap informació útil, o bé podem continuar sense mesurar-lo. Els efectes són els mateixos, però els càlculs se simplifiquen més si mesurem, ja que el valor de b queda determinat i no cal escriure el primer sumatori. Tanmateix, seguirem efectuant els càlculs com si no haguéssim mesurat i veurem que realment el valor de b no interfereix en el resultat.

En aplicar la transformada quàntica de Fourier obtenim la següent superposició:

$$\frac{1}{\sqrt{2^t}}\sum_{b=0}^{r-1}\sum_{\substack{a=0\\ ar+b<2^t}}^{\left\lfloor\frac{2^t}{r}\right\rfloor-1}\mathcal{F}_t|ar+b\rangle_t \left|x^b\right\rangle_n = \frac{1}{\sqrt{2^t}}\sum_{b=0}^{r-1}\sum_{\substack{a=0\\ ar+b<2^t}}^{\left\lfloor\frac{2^t}{r}\right\rfloor-1}\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}e\left(\frac{(ar+b)j}{2^t}\right)|j\rangle_t \left|x^b\right\rangle_n =$$

$$= \frac{1}{2^t}\sum_{b=0}^{r-1}\sum_{j=0}^{2^t-1}e\left(\frac{bj}{2^t}\right)\underbrace{\left[\sum_{a=0}^{\left\lfloor\frac{2^t-b}{r}\right\rfloor-1}e\left(\frac{aj}{2^t/r}\right)\right]}_{\delta(j)}|j\rangle_t \left|x^b\right\rangle_n.$$

Observem que la part de dins dels claudàtors, $\delta(j)$, és molt proper a una suma d'1s quan j s'acosta a un múltiple de $2^t/r$, i molt proper a zero si no. És a dir,

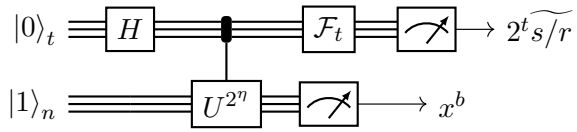
$$\delta(j) = \sum_{a=0}^{\left\lfloor\frac{2^t-b}{r}\right\rfloor-1}e\left(\frac{aj}{2^t/r}\right) \approx \begin{cases}\left\lfloor\frac{2^t-b}{r}\right\rfloor \approx \frac{2^t}{r} & \text{si } \exists s \in \mathbb{N}, j \approx s\frac{2^t}{r},\\ 0 & \text{si } \nexists s \in \mathbb{N}, j \approx s\frac{2^t}{r}.\end{cases}$$

Per tant, abans de mesurar el primer registre tenim la superposició

$$\frac{1}{2^t}\sum_{b=0}^{r-1}\sum_{j=0}^{2^t-1} e\left(\frac{bj}{2^t}\right)\delta(j)\left|j\right\rangle_t\left|x^b\right\rangle_n \approx \frac{1}{2^t}\sum_{b=0}^{r-1}\sum_{s=0}^{r-1} e\left(\frac{bs\frac{2^t}{r}}{2^t}\right)\delta\left(s\frac{2^t}{r}\right)\left|\widetilde{s\frac{2^t}{r}}\right\rangle_t\left|x^b\right\rangle_n \approx$$

$$\approx \frac{1}{2^t}\sum_{b=0}^{r-1}\sum_{s=0}^{r-1} e\left(\frac{bs}{r}\right)\frac{2^t}{r}\left|\widetilde{\frac{s2^t}{r}}\right\rangle_t\left|x^b\right\rangle_n = \frac{1}{r}\sum_{b=0}^{r-1}\sum_{s=0}^{r-1} e\left(\frac{bs}{r}\right)\left|\widetilde{\frac{s2^t}{r}}\right\rangle_t\left|x^b\right\rangle_n.$$

És a dir, tindrem una superposició quasi equiprobable d'aproximacions dels diferents valors $\frac{s2^t}{r}$ per a tot s. Si mesurem un valor $\widetilde{\frac{s}{r}2^t}$, podem aplicar fraccions continuades amb el nombre mesurat dividit per 2^t, i així obtindrem una aproximació de s/r per a algun s. De fet, tot aquest procés l'hauríem de repetir tantes vegades com fos necessari per a garantir poder trobar el valor de r tal i com hem vist al final de la secció 3.6.



Per a trobar un factor no trivial de N amb èxit, interessa que r sigui parell. Per tant, si r surt senar haurem de tornar a escollir un altre enter x a l'atzar (de manera uniforme). En altres paraules, l'algoritme ha fracassat i hem de tornar a començar.

Un cop tenim $x^r \equiv 1 \pmod N$ amb r parell, observem que

$$N \mid \left(x^{r/2}\right)^2 - 1 \iff N \mid \left(x^{r/2}+1\right)\left(x^{r/2}-1\right).$$

Si N no divideix cap d'aquests factors, aleshores $\operatorname{mcd}\left(\left(x^{r/2}+1\right), N\right)$ i $\operatorname{mcd}\left(\left(x^{r/2}-1\right), N\right)$ són divisors no trivials de N, i els podem trobar fàcilment amb l'algoritme d'Euclides.

$$\begin{cases} N \mid \left(x^{r/2}+1\right) \iff x^{r/2} \equiv -1 \pmod N \implies \text{ no podem factoritzar.} \\ N \mid \left(x^{r/2}-1\right) \iff x^{r/2} \equiv 1 \pmod N \implies r \text{ no és l'ordre de } x. \text{ Cas "impossible".} \end{cases}$$

Hem vist a la secció anterior (3.6) que la complexitat de la part quàntica és $O\left((\log N)^3\right)$; els càlculs un cop donat r són també $O\left((\log N)^3\right)$, ja que estem utilitzant l'algoritme d'Euclides (cf. [Co-93, §1.3]) per trobar els factors no trivials (en cas que existeixin), i l'exponenciació binària (cf. [Co-93, §1.2]) és $O\left((\log N \cdot \log\log N)^2\right)$.

La probabilitat de fracàs ξ, un cop ja hem trobat l'ordre r, és la probabilitat de triar un $x \in (\mathbb{Z}/N\mathbb{Z})^*$ a l'atzar (uniformement) que tingui ordre r parell amb $x^{r/2} \not\equiv -1 \pmod N$.

Proposició 3.7.1. *Sigui m la quantitat de nombres primers diferents que divideixen N.*

$$\xi = P\left(r \textit{ senar o } r \textit{ parell amb } x^{r/2} \not\equiv -1 \,(\textit{mod } N)\right) \leq \frac{1}{2^m}.$$ □

Així, la probabilitat d'èxit és més gran que $1 - \frac{1}{2^m}$. Com que N és compost, el nombre de primers diferents que divideixen N és com a mínim $m = 2$, i per tant, la probabilitat d'èxit és com a mínim de $1 - \frac{1}{2^2} = \frac{3}{4}$; és a dir, constant.

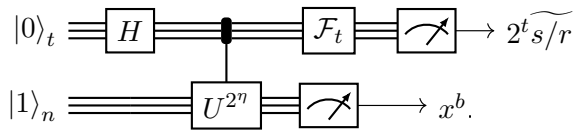
Podem concloure, doncs, que l'algoritme de Shor es pot implementar utilitzant $O(\log N)$ qubits i $O\left((\log N)^3\right)$ portes quàntiques elementals, una complexitat clàssica $O\left((\log N)^3\right)$ i una probabilitat d'èxit $O(1)$, és a dir, constant (cf. [Jo-01], [Lo-04]).

3.8 Interpretació de l'algoritme de Shor

Hem vist que la part quàntica de l'algoritme de Shor resol el problema de subgrup amagat induït per al problema de factorització. Recordem que és

$$G = \mathbb{Z}, \quad X = \left(\frac{\mathbb{Z}}{N\mathbb{Z}}\right)^*, \quad f : k \mapsto x^k, \quad H = \langle r \rangle = r\mathbb{Z},$$

per a algun $x \in X$, r l'ordre de x. Anem a veure com la part quàntica de Shor usa aquests elements, i així poder generalitzar aquest algoritme per a qualsevol grup abelià finit. Recordem que el circuit és



Al segon registre hi codifiquem els elements de $X = \left(\frac{\mathbb{Z}}{N\mathbb{Z}}\right)^*$.

Al primer registre, però, no hi codifiquem $G = \mathbb{Z}$ de manera directa. Tenim la base computacional $\left\{|0\rangle_t, \ldots, \left|2^t - 1\right\rangle_t\right\}$, que codifica de manera canònica els elements de $\frac{\mathbb{Z}}{2^t\mathbb{Z}}$. Realment, no estem resolent el problema del subgrup amagat per a $G = \mathbb{Z}$, sinó que estem fent un intent de reducció mòdul 2^t:

$$G = \frac{\mathbb{Z}}{2^t\mathbb{Z}}, \quad H = \langle r \rangle = \frac{r\mathbb{Z}}{2^t\mathbb{Z}}, \quad \frac{G}{H} \text{ "=" } \frac{\mathbb{Z}}{r\mathbb{Z}}.$$

Per aquest motiu, quan $r \mid 2^t$ obtenim un resultat exacte, i quan no, obtenim un valor aproximat i necessitem un valor de t *prou gran* per mitigar l'error, ja que l'expressió $\frac{r\mathbb{Z}}{2^t\mathbb{Z}}$ perd significat.

Les portes Hadamard simplement creen la superposició corresponent a la suma de tots els vectors de la base, que si considerem que codifiquen els elements de G, llavors tenim una combinació lineal de tots els elements de $G = \frac{\mathbb{Z}}{2^t\mathbb{Z}}$:

$$\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle_t .$$

Apliquem les portes CU^{2^η} amb el neutre codificat al segon registre. El motiu pel qual l'inicialitzem al neutre és per a obtenir-hi escrit (al segon registre) el resultat d'aplicar la funció f al primer registre; és a dir,

$$CU^{2^\eta} : |k\rangle_t |1\rangle_n \mapsto |k\rangle_t \left|x^k\right\rangle_n = |k\rangle_t |f(k)\rangle_n .$$

Observem que l'algoritme és independent de la implementació quàntica de f. Simplement ens interessa aplicar f, és a dir, aplicar un operador $\widetilde{\mathcal{O}f}$ que generi l'estat $|k\rangle_t |f(k)\rangle_n$ a partir de l'estat $|k\rangle_t |\phi\rangle_n$, per a algun valor ϕ constant, conegut i fàcil de calcular.

La transformada de Fourier s'aplica només al primer registre, per tant, tracta els elements de $G = \frac{\mathbb{Z}}{2^t\mathbb{Z}}$. Per veure què significa la transformada de Fourier en un grup abelià finit qualsevol, haurem de definir què són els caràcters d'un grup abelià finit (cf. [Se-67]).

Notació 3.8.1. Tot grup abelià finit G és isomorf a una suma directa de grups cíclics finits, és a dir, existeixen $G_1, \ldots, G_M$ grups cíclics finits tals que $G \simeq G_1 \oplus \cdots \oplus G_M$. Encara que els G_i no siguin únics, els podem deixar fixats. A cada $g \in G$ li correspon, per la bijecció, un element $(g_1, \ldots, g_M) \in G_1 \oplus \cdots \oplus G_M$. Abusarem una mica de la notació i escriurem els elements de G com $g = (g_1, \ldots, g_M) \in G$.

Definició 3.8.2 (Caràcter)**.** Sigui $(G,+)$ un grup abelià finit. Un **caràcter** de G és un homomorfisme de grups entre G i els complexos invertibles, $\chi : G \to \mathbb{C}^*$.

Observació 3.8.3. Sigui χ un caràcter de G, Aleshores $\forall g, g' \in G$, $\chi(g+g') = \chi(g)\chi(g')$.

Proposició 3.8.4. *Les següents afirmacions són certes:*

1. *$Hom\,(G, \mathbb{C}^*)$ és un grup amb el producte $(\chi \cdot \chi')(g) = \chi(g) \cdot \chi'(g)$.*
2. *Sigui G un grup cíclic finit. Existeix un isomorfisme no canònic $Hom\,(G, \mathbb{C}^*) \cong G$.*
3. *Siguin $G_1, \ldots, G_M$ grups abelians. Aleshores $Hom\,(\bigoplus_i G_i, \mathbb{C}^*) \cong \prod_i Hom\,(G_i, \mathbb{C}^*)$.*
4. *Sigui G un grup abelià finit. Existeix un isomorfisme no canònic $Hom\,(G, \mathbb{C}^*) \cong G$.*

□

Definició 3.8.5 (Subgrup ortogonal)**.** Siguin G un grup finit, $H < G$ un subgrup. Definim el **subgrup ortogonal** de H com

$$H^\perp := \{\chi : \chi(h) = 1 \forall h \in H\} < \mathrm{Hom}(G, \mathbb{C}^*).$$

Els N caràcters de $G = \frac{\mathbb{Z}}{N\mathbb{Z}}$ són exactament les aplicacions definides per $\chi_k(j) = e\left(\frac{jk}{N}\right)$, i els $N = N_1 \cdots N_M$ caràcters de $G \simeq \frac{\mathbb{Z}}{N_1\mathbb{Z}} \oplus \cdots \oplus \frac{\mathbb{Z}}{N_M\mathbb{Z}}$ són els productes dels caràcters de cada $\frac{\mathbb{Z}}{N_i\mathbb{Z}}$, és a dir, els homomorfismes definits per

$$\chi_k(j) = \chi_{(k_1,\ldots,k_M)}((j_1,\ldots,j_M)) = \prod_{i=1}^{M} e\left(\frac{j_i k_i}{N_i}\right) = e\left(\sum_{i=1}^{M} \frac{j_i k_i}{N_i}\right).$$

Aquesta definició fixa un isomorfisme (no canònic) entre G i $\mathrm{Hom}(G, \mathbb{C}^*)$. A més a més, una propietat immediata és que $\forall k, j \in G,\ \chi_k(j) = \chi_j(k)$.

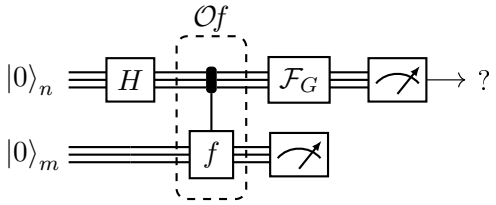
Notació 3.8.6. Denotarem per $|g\rangle_n$ la codificació de $g \in G$ amb $n = O(\log |G|)$ qubits i $|x\rangle_m$ la codificació de $x \in X$ amb $m = \lceil \log_2 |X| \rceil$ qubits.

Definició 3.8.7 (Transformada quàntica de Fourier)**.** Sigui G un grup finit. Definim la **transformada quàntica de Fourier** de G com l'aplicació

$$\mathcal{F}_G : |s\rangle_n \mapsto |\chi_s\rangle_n := \frac{1}{\sqrt{|G|}} \sum_{g \in G} \chi_s(g)\, |g\rangle_n .$$

Observació 3.8.8. Si $G = \frac{\mathbb{Z}}{2^n\mathbb{Z}}$ és un grup cíclic d'ordre potència de 2, aleshores la transformada de Fourier $\mathcal{F}_G$ coincideix amb la transformada $\mathcal{F}_n$ definida a la secció 3.1. A més, si $G = G_1 \oplus \cdots \oplus G_M$, aleshores $\mathcal{F}_G = \mathcal{F}_{G_1} \otimes \cdots \otimes \mathcal{F}_{G_M}$.

El circuit quàntic de l'algoritme de Shor amb la notació de subgrup amagat és



on $\mathcal{O}f$ és un operador tal que $\mathcal{O}f : |g\rangle_n |0\rangle_m \mapsto |g\rangle_n |f(g)\rangle_m$. Si en comptes de $\mathcal{O}f$ tenim un operador $\widetilde{\mathcal{O}f} : |g\rangle_n |\phi\rangle_m \mapsto |g\rangle_n |f(g)\rangle_m$, per a algun $|\phi\rangle_m$ constant, conegut i fàcil de construir; aleshores considerem la composició $\mathcal{O}f = \widetilde{\mathcal{O}f} \circ (|0\rangle_m \mapsto |\phi\rangle_m)$.

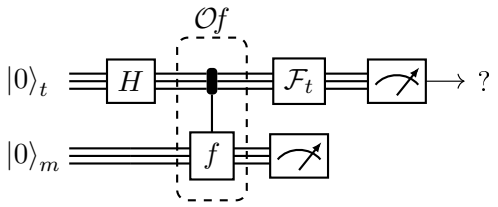
3.9 Implementació del cas cíclic

Veiem que l'algoritme de Shor permet realment resoldre qualsevol cas cíclic de subgrup amagat i estudiem també quina és la probabilitat d'èxit.

Posem doncs, $G = \frac{\mathbb{Z}}{N\mathbb{Z}}$, $H = \langle r \rangle = \frac{r\mathbb{Z}}{N\mathbb{Z}}$, $m = \lceil \log_2 |X| \rceil$ i f la funció que separa les classes laterals de G mòdul H. Els únics subgrups d'un grup cíclic finit són els subgrups generats per un sol element i, com que $\langle r \rangle = \langle \operatorname{mcd}(r, N) \rangle$, podem definir $H = \langle r \rangle$ amb $r \mid N$. Usarem t qubits al primer registre i m al segon, on $N < 2^t$, per a algun valor de t que ja acabarem de fixar després en funció de N i de la probabilitat d'èxit. Els kets de la base computacional identificaran els elements de G de manera que $|k\rangle_t \rightsquigarrow k \pmod N \in G$.

La transformada de Fourier $\mathcal{F}_G$ no la podem implementar de manera exacta. Per tant, utilitzarem l'aproximació $\mathcal{F}_G \approx \mathcal{F}_t$.

La part quàntica de l'algoritme, doncs, correspondrà al següent circuit quàntic:



Apliquem l'algoritme partint de l'estat inicial $|0\rangle_t \otimes |0\rangle_m$ i ens aturem abans de mesurar el primer registre. Denotem per Obs_2 l'acció de mesurar el registre 2.

$$\mathcal{F}_t\left(Obs_2\left(\mathcal{O}f\left(\left(H\,|0\rangle_t\right)\otimes|0\rangle_m\right)\right)\right) = \mathcal{F}_t\left(Obs_2\left(\frac{1}{\sqrt{2^t}}\sum_{b=0}^{r-1}\sum_{a=0}^{\left\lfloor\frac{2^t-b}{r}\right\rfloor-1}|ar+b\rangle_t\,|f(b)\rangle_m\right)\right) =$$

$$= \mathcal{F}_t\left(\frac{1}{\sqrt{\left\lfloor\frac{2^t-b}{r}\right\rfloor}}\sum_{a=0}^{\left\lfloor\frac{2^t-b}{r}\right\rfloor-1}|ar+b\rangle_t\right) = \frac{1}{\sqrt{\left\lfloor\frac{2^t-b}{r}\right\rfloor}}\sum_{a=0}^{\left\lfloor\frac{2^t-b}{r}\right\rfloor-1}\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}e\left(\frac{(ar+b)j}{2^t}\right)|j\rangle_t =$$

$$= \sum_{j=0}^{2^t-1}\frac{e\left(\frac{bj}{2^t}\right)}{\sqrt{2^t\left\lfloor\frac{2^t-b}{r}\right\rfloor}}\underbrace{\left[\sum_{a=0}^{\left\lfloor\frac{2^t-b}{r}\right\rfloor-1}e\left(\frac{j}{2^t/r}\right)^a\right]}_{\delta(j)}|j\rangle_t = \sum_{j=0}^{2^t-1}\frac{e\left(\frac{bj}{2^t}\right)}{\sqrt{2^t\left\lfloor\frac{2^t-b}{r}\right\rfloor}}\delta(j)\,|j\rangle_t = \sum_{j=0}^{2^t-1}\alpha_j\,|j\rangle_t\,,$$

on α_j és el coeficient de $|j\rangle_t$, i el valor de b queda fixat al mesurar.

Si j és aproximadament un múltiple de $\frac{2^t}{r}$, $j \approx s\frac{2^t}{r}$, aleshores podem trobar els valors de s i r amb el mètode de fraccions continuades. A més, hem vist a la secció 3.5 que aquesta aproximació ha de tenir una precisió de $\frac{1}{2r^2}$. Això vol dir que al mesurar j, s'ha de complir que $\left|\frac{j}{2^t} - \frac{s}{r}\right| \leq \frac{1}{2r^2}$; sinó, l'algoritme haurà fracassat.

La probabilitat de fracàs és

$$\varepsilon = P\left(\text{Mesurar } j \text{ tal que } \left|\frac{j}{2^t} - \frac{s}{r}\right| \geq \frac{1}{2r^2}\right) = \sum_{\substack{j=0 \\ \left|\frac{j}{2^t}-\frac{s}{r}\right|\geq\frac{1}{2r^2}}}^{2^t-1}|\alpha_j|^2\,.$$

Veiem ràpidament com acotar aquests coeficients α_j, ja que els càlculs són similars als de la secció 3.4. Escrivim $j = \ell + k\frac{2^t}{r}$, $k \in \mathbb{Z}$, amb representant $-\frac{2^t}{2r} \leq \ell \leq \frac{2^t}{2r}$, i per tant, la condició de fracàs serà $|\ell| \geq \frac{2^t}{2r^2}$. Així, $e\left(\frac{j}{2^t/r}\right) = e\left(\frac{\ell}{2^t/r}\right) e\left(\frac{k2^t/r}{2^t/r}\right) = e\left(\frac{\ell r}{2^t}\right)$.

$$|\delta(j)| = \left| \sum_{a=0}^{\left\lfloor \frac{2^t-b}{r} \right\rfloor - 1} e\left(\frac{\ell r}{2^t}\right)^a \right| = \frac{\left| e\left(\frac{\ell r}{2^t}\left\lfloor \frac{2^t-b}{r} \right\rfloor\right) - 1 \right|}{\left| e\left(\frac{\ell r}{2^t}\right) - 1 \right|} \leq \frac{2}{\frac{2}{\pi}\left|2\pi\frac{\ell r}{2^t}\right|} = \frac{2^t}{2\,|\ell|\,r}.$$

Ja ho tenim tot preparat per calcular ε en funció del nombre de qubits usats, t, i en funció també de r.

$$\varepsilon = \sum_{\substack{j=0 \\ |\frac{j}{2^t} - \frac{s}{r}| \geq \frac{1}{2r^2}}}^{2^t-1} |\alpha_j|^2 = \sum_{\substack{j=0 \\ |\frac{j}{2^t} - \frac{s}{r}| \geq \frac{1}{2r^2}}}^{2^t-1} \left| \frac{e\left(\frac{bj}{2^t}\right)}{\sqrt{2^t\left\lfloor \frac{2^t-b}{r} \right\rfloor}} \delta(j) \right|^2 \leq r \cdot \sum_{\substack{\ell=\left\lceil \frac{-2^t}{2r} \right\rceil \\ |\ell| \geq \frac{2^t}{2r^2}}}^{\left\lfloor \frac{2^t}{2r} \right\rfloor} \frac{1}{2^t\left(\frac{2^t}{r} - 2\right)} \left(\frac{2^t}{2\,|\ell|\,r}\right)^2 \leq$$

$$\leq 2r \cdot \sum_{\ell=\left\lceil \frac{2^t}{2r^2} \right\rceil}^{\left\lfloor \frac{2^t}{2r} \right\rfloor} \frac{1}{2^t \frac{2^t}{r}\left(1 - \frac{2r}{2^t}\right)} \cdot \frac{\left(2^t\right)^2}{2^2\ell^2 r^2} = \frac{1}{1 - \frac{2r}{2^t}} \sum_{\ell=\left\lceil \frac{2^t}{2r^2} \right\rceil}^{\left\lfloor \frac{2^t}{2r} \right\rfloor} \frac{1}{2\ell^2} < \frac{1}{1 - \frac{2r}{2^t}} \int_{\frac{2^t}{2r^2}-1}^{\infty} \frac{1}{2\ell^2} d\ell =$$

$$= \frac{1}{1 - \frac{2r}{2^t}} \cdot \frac{1}{2\left(\frac{2^t}{2r^2} - 1\right)} = \frac{1}{2\left(\frac{2^t}{2r^2} - \frac{1}{r} + \frac{2r}{2^t} - 1\right)} \leq \frac{1}{2\left(\frac{2^t}{2r^2} - 1 + 0 - 1\right)} = \frac{1}{2\left(\frac{2^t}{2r^2} - 2\right)}.$$

Si posem n tal que $r < 2^n$, aleshores

$$\varepsilon < \frac{1}{2\left(\frac{2^t}{2r^2} - 2\right)} \leq \frac{1}{2\left(\frac{2^t}{2^{2n+1}} - 2\right)} \implies t = 2n + 1 + \left\lceil \log_2\left(2 + \frac{1}{2\varepsilon}\right) \right\rceil.$$

Com que r és desconegut, ens haurem de conformar amb una cota. De fet, com pitjor sigui la cota, la probabilitat de fracàs real serà més baixa, ja que estarem obligats a utilitzar més qubits. En estar als enters mòdul N, una cota trivial i suficient és $r \leq N \leq 2^{\lceil \log_2(N) \rceil}$; és a dir, $n = \lceil \log_2(N) \rceil$.

Segons quina probabilitat d'èxit hem de garantir, triem la quantitat $p = \left\lceil \log_2\left(2 + \frac{1}{2\varepsilon}\right) \right\rceil$ suficient. Si utilitzem $t = 2n + 1 + p$ qubits, aleshores

$p =$	2	3	4	5	6	$\cdots$	9	$\cdots$
$\varepsilon <$	0.25	0.0833333	0.0357143	0.0166667	0.00806452	$\cdots$	0.000980392	$\cdots$

Així doncs, ja hem determinat t.

Un cop hem implementat l'algoritme, haurem mesurat un enter $j \approx s\frac{2^t}{r}$. Comencem a calcular els convergents de la fracció continuada de $\frac{j}{2^t}$ fins a obtenir un convergent de denominador més gran que N. Aleshores, l'anterior convergent, $\frac{s'}{r'}$ és exactament la fracció irreductible de $\frac{s}{r}$. Per tant, r' és un divisor de r.

Repetim l'algoritme per a obtenir altres divisors de r. Calculant el mínim comú múltiple d'aquests, obtindrem r amb probabilitat $(1 - \epsilon) \in O(1)$. A la secció 3.6 hem calculat una cota de ϵ en funció del nombre de divisors obtinguts.

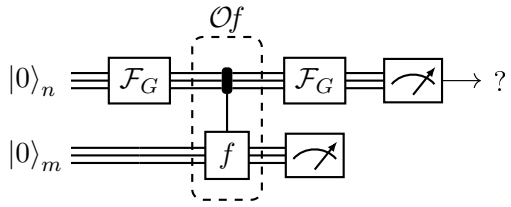
3.10 Algoritme per al cas abelià general

Veiem que la generalització de l'algoritme de Shor també permet resoldre a nivell teòric qualsevol cas abelià finit de subgrup amagat.

Considerem la codificació dels elements de G i de X amb n i m qubits respectivament i denotem per $\mathcal{O}f$ a un operador quàntic de f tal que $|g\rangle_n |0\rangle_m \mapsto |g\rangle_n |f(g)\rangle_m$. A més a més, per a construir la superposició de tots els elements de G sovint s'utilitza el formalisme de la transformada de Fourier $\mathcal{F}_G$ en comptes de les portes Hadamard, perquè el caràcter associat al neutre de G, χ_0, és el caràcter trivial, i per tant,

$$\mathcal{F}_G |0\rangle_n = |\chi_0\rangle_n = \frac{1}{\sqrt{|G|}} \sum_{g\in G} \chi_0(g) |g\rangle_n = \frac{1}{\sqrt{|G|}} \sum_{g\in G} |g\rangle_n .$$

En realitat només es poden usar les portes Hadamard si la codificació dels elements de G amb els vectors de la base computacional és bijectiva. Per tant, el circuit quàntic serà



Comencem doncs amb l'estat inicial $|0\rangle_n \otimes |0\rangle_m$, i després d'aplicar la transformada de Fourier al primer registre, obtenim una superposició de tots els elements de G, és a dir,

$$\frac{1}{\sqrt{|G|}} \sum_{g\in G} |g\rangle_n |0\rangle_m .$$

Ara implementem f via l'oracle donat, és a dir, $\mathcal{O}f$. Això resulta en la superposició

$$\frac{1}{\sqrt{|G|}} \sum_{g\in G} |g\rangle_n |f(g)\rangle_m = \frac{1}{\sqrt{|G|\,|H|}} \sum_{s\in S} \frac{1}{\sqrt{|H|}} \sum_{h\in H} |s+h\rangle_n |f(s)\rangle_m ,$$

on S denota un conjunt de representants de les classes laterals de G mòdul H. La igualtat és certa perquè $\forall h \in H$, $f(s+h) = f(s)$. Així doncs, podem interpretar que al primer registre hi ha una superposició de les classes laterals $s+H$ per a tot $s \in S$.

Observem que l'acció de mesurar el segon registre simplement determina el valor de $s \in S$; ja que si fos $f(s) = f(s')$ per a $s, s' \in S$ amb $s \neq s'$, com que f separa les classes laterals de G mòdul H, aleshores $s+H = s'+H$, i això contradiu el fet que S sigui un conjunt de representants de les classes laterals de G mòdul H.

Per tant, després de mesurar el segon registre, del qual ignorem el resultat ja que no ens aporta cap informació útil, obtenim l'estat

$$\frac{1}{\sqrt{|H|}} \sum_{h\in H} |s+h\rangle_n$$

en el primer registre, que és la superposició dels elements de la classe lateral $s+H$, per a algun s determinat però desconegut.

Si apliquem ara la transformada quàntica de Fourier $\mathcal{F}_G$, obtenim l'estat

$$|\Psi_s\rangle_n := \frac{1}{\sqrt{|H|}} \sum_{h\in H} |\chi_{s+h}\rangle_n = \frac{1}{\sqrt{|H|}} \sum_{h\in H} \frac{1}{\sqrt{|G|}} \sum_{g\in G} \chi_{s+h}(g)\,|g\rangle_n\,.$$

Observem que $\chi_{s+h}(g) = \chi_g(s+h) = \chi_g(s)\chi_g(h)$, i per tant, podem reescriure l'estat obtingut com

$$|\Psi_s\rangle_n = \frac{1}{\sqrt{|H|}} \sum_{h\in H} \frac{1}{\sqrt{|G|}} \sum_{g\in G} \chi_g(s)\chi_g(h)\,|g\rangle_n = \frac{1}{\sqrt{|H|\,|G|}} \sum_{g\in G} \chi_g(s) \left(\sum_{h\in H} \chi_g(h)\right) |g\rangle_n\,.$$

Observació 3.10.1. Un caràcter χ de G restringit a H és un caràcter de H. Si $\chi \in H^\perp$, aleshores χ és constant igual a 1 per ortogonalitat. Si $\chi \notin H^\perp$, aleshores χ anul·la la suma sobre H; ja que existeix un $h' \in H$ tal que $\chi(h') \neq 1$ i $\chi(h')^\varrho = 1$, on ϱ és l'ordre de h', i aleshores la suma de totes les potències de $\chi(h')$ és zero. És a dir,

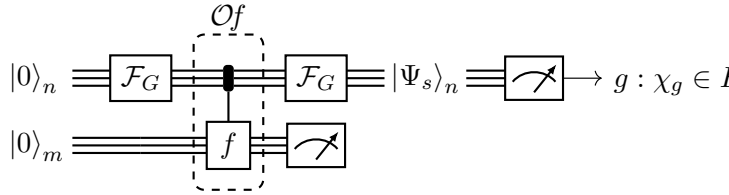
$$\sum_{h\in H} \chi(h) = \begin{cases} \displaystyle\sum_{h\in H} 1 = |H| & \text{si } \chi \in H^\perp, \\ \displaystyle\sum_{k\in K} \sum_{j\in\langle h'\rangle} \chi(k+j) = \sum_{k\in K} \chi(k) \sum_{a=0}^{\varrho-1} \chi(h')^a = \sum_{k\in K} \chi(k)\cdot 0 = 0 & \text{si } \chi \notin H^\perp, \end{cases}$$

on K és un conjunt de representants de les classes laterals de H mòdul $\langle h'\rangle$.

Per tant,

$$|\Psi_s\rangle_n = \frac{1}{\sqrt{|H|\,|G|}} \sum_{g:\chi_g\in H^\perp} \chi_s(g)\,|H|\,|g\rangle_n = \frac{1}{\sqrt{|G|\,/\,|H|}} \sum_{g:\chi_g\in H^\perp} \chi_s(g)\,|g\rangle_n\,.$$

Com que $\forall s, g \in G$, $|\chi_s(g)|^2 = 1$, l'estat $|\Psi_s\rangle_n$ és una superposició equiprobable de cadascun dels $g \in G$ tals que $\chi_g \in H^\perp$. En altres paraules, els únics valors possibles a mesurar són les etiquetes de $H^\perp$ (respecte l'isomorfisme no canònic $G \simeq \mathrm{Hom}(G, \mathbb{C}^*)$ fixat a la secció 3.8). Així doncs, després d'aplicar l'algoritme i mesurar l'estat $|\Psi_s\rangle_n$ obtindrem una etiqueta de $H^\perp$.



Sigui $g = (g_1, \ldots, g_M) \in G$ l'etiqueta de $\chi_g \in H^\perp$, sabem, per la definició d'ortogonal, que $\forall h \in H$, $\chi_g(h) = 1$. Per tant, podem trobar una equació a partir de la definició del caràcter χ_g que es compleix per a tot $h = (h_1, \ldots, h_M) \in H$.

$$1 = \chi_g(h) = \chi_{(g_1,\ldots,g_M)}((h_1, \ldots, h_M)) = e\left(\sum_{i=1}^{M} \frac{g_i h_i}{N_i}\right) \implies \sum_{i=1}^{M} \frac{g_i h_i}{N_i} = 0 \pmod 1.$$

D'aquesta manera, amb $O(\log|G|)$ mesures es poden trobar prou equacions per determinar un conjunt de generadors de H (cf. [Lo-04, §3.5]).

Ara bé, la implementació de $\mathcal{F}_G = \mathcal{F}_{G_1} \otimes \cdots \otimes \mathcal{F}_{G_M}$ només la podem dur a terme de manera exacta i eficient si cadascun dels G_i és un grup cíclic d'ordre potència de 2; per exemple, $G = \left(\frac{\mathbb{Z}}{2\mathbb{Z}}\right)^{\oplus n}$ per a resoldre el problema de Simon (cf. [Br,Ho-97]). Per tant, a la pràctica, els càlculs clàssics posteriors a mesurar s'han de modificar per tal de trobar etiquetes de l'ortogonal, és a dir, elements de $\left\{g \in G : \chi_g \in H^\perp\right\}$, a partir de valors aproximats.

3.11 Altres grups i resultats

Per a grups no abelians, la transformada de Fourier definida per caràcters a la secció 3.8 no serveix. S'ha d'usar una definició per representacions (cf. [M,M,T-99]), i això complica els càlculs. A més, la part clàssica que s'encarrega de trobar generadors del subgrup a partir dels elements mesurats té una complexitat més gran que polinòmica respecte del nombre de bits de $|G|$.

Alguns dels resultats més rellevants en aquesta línia de generalització de Shor corresponen al cas del diedral (cf. [Ku-05], [Re-04]), en què s'ha aconseguit igualar la complexitat dels millors algoritmes clàssics que resolen aquest cas. Per tant, la computació quàntica encara no permet trencar els criptosistemes que confien la seva seguretat en problemes del cas diedral, és a dir, problemes relacionats amb xarxes. És per aquest motiu que a aquests criptosistemes se'ls anomena algoritmes postquàntics, fent referència al fet que teòricament seguiran sent segurs després del desenvolupament dels computadors quàntics més avançats.

A part de l'algoritme generalitzat de Shor, conegut també com a mostreig de classes laterals (*Coset sampling*) o mostreig de Fourier (*Sampling Fourier*), hi ha altres aproximacions per a resoldre el problema del subgrup amagat per al cas no abelià. S'ha aconseguit algun resultat teòric d'un algoritme de complexitat polinòmica, però que requereix de temps exponencial per a ser executat (cf. [E,H,K-04]). Consisteix a aplicar un nombre polinòmic d'operacions unitàries que actuen com a tests de pertinença d'un subgrup. El problema és que per a construir aquestes portes es necessita una quantitat exponencial de portes quàntiques bàsiques o elementals, i això fa que el temps d'execució sigui tan elevat en comparació a la complexitat teòrica de l'algoritme.

L'algoritme de Shor, però, no és l'únic algoritme quàntic de factorització conegut. Malgrat que a nivell teòric sigui l'únic que s'ha demostrat que té una complexitat polinòmica, no és pas el tipus d'algoritme que bat rècords de factorització en un ordinador quàntic. Shor està basat en el model de computació quàntica de circuits, mentre que els que baten rècords es basen en el model de computació quàntica adiabàtica (cf. [Ki-19]). Experimentalment sembla que l'algoritme de factorització adiabàtic sigui més eficient en temps i en nombre de qubits que el de Shor, encara que no s'hagi pogut demostrar a nivell teòric. Tot i així, l'algoritme adiabàtic de factorització no es pot estendre de manera natural a la resta de problemes del subgrup amagat, ja que es resol directament el problema de trobar un factor no trivial, i no pas el de trobar l'ordre d'un element. De fet, alguns dels problemes que es poden resoldre amb la computació adiabàtica actual són problemes d'optimització.

Referències

[Ad,Hu-99] Leonard M. Adleman, Ming-Deh A. Huang: Function Field Sieve Method for Discrete Logarithms over Finite Fields. *Information and Computation* **151**, n. 1-2 (1999), p. 5-16. DOI: 10.1006/inco.1998.2761

[Be-73] Charles H. Bennett: Logical reversibility of computation. *IBM Journal of Research and Development* **17**, n. 6 (1973), p. 525-532. DOI: 10.1147/rd.176.0525

[Br,Ho-97] Gilles Brassard, Peter Hoyer: An Exact Quantum Polynomial-Time Algorithm for Simon's Problem. *Fifth Israeli Symposium on Theory of Computing and Systems (ISTCS'97)* (1997), p. 12-23. DOI: 10.1109/ISTCS.1997.595153

[Ca-08] Alberto Cámara: Algoritmes quàntics de factorització. *Universitat de Barcelona, Facultat de Matemàtiques (Treball de Fi de Màster)* (2008).

[Co-93] Henri Cohen: *A Course in Computational Algebraic Number Theory.* Springer-Verlag, Berlin Heidelberg New York, 1993. ISBN: 3-540-55640-0, ISBN: 0-387-55640-0

[Co-94] Don Coppersmith: An Approximate Fourier Transform Useful in Quantum Factoring. *IBM Research Report* RC 19642 (1994).

[E,H,K-04] Mark Ettinger, Peter Hoyer, Emanuel Knill: The quantum query complexity of the hidden subgroup problem is polynomial. *Information Processing Letters* **91**, n. 1 (2004), p. 43-48. DOI: 10.1016/j.ipl.2004.01.024

[EGQC-95] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John Smolin, Harald Weinfurter: Elementary gates for quantum computation. *Physical Review A* **52**, n. 5 (1995), p. 3457–3467. DOI: 10.1103/PhysRevA.52.3457

[Ga-96] Carl F. Gauss: *Disquisicions aritmètiques.* Traducció de Griselda Pascual Xufré (original del 1801). Societat Catalana de Matemàtiques (Institut d'Estudis Catalans), Barcelona, 1996. ISBN: 84-7283-313-5

[Ha,Wr-79] Godfrey H. Hardy, Edward M. Wright: *An introduction to the Theory of Numbers. 5a edició* (edició original, 1938), Clarendon Press, Oxford, 1979. ISBN: 0-19-853171-0

[Jo-01] Richard Jozsa: Quantum factoring, discrete logarithms and the hidden subgroup problem. *Computing in Science & Engineering* **3**, n. 2 (2001), p. 34-43. DOI: 10.1109/5992.909000

[Ki-19] Tien D. Kieu: A Factorisation Algorithm in Adiabatic Quantum Computation. *Journal of Physics Communications* **3**, n. 025014 (2019). DOI: 10.1088/2399-6528/ab060d

[Ku-05] Greg Kuperberg: A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM J. Computing* **35**, n. 1 (2005), p. 170-188. DOI: 10.1137/S0097539703436345

[Lo-04] Chris Lomont: The Hidden Subgroup Problem - Review and Open Problems. *Cornell University* arXiv:quant-ph/0411037 (2004).

[M,M,T-99] Markus Püschel, Martin Rötteler, Thomas Beth: Fast Quantum Fourier Transforms for a Class of Non-abelian Groups. *Proceedings 13th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (AAECC'99)* (1999), p. 148-159.

[Mi,Ni-10] Michael A. Nielsen, Isaac L. Chuang: *Quantum computation and quantum information. 10th Anniversary Edition* (edició original, 2000), University Press, Cambridge, 2010. ISBN: 978-1-107-00217-3

[NIST-15] Elaine Barker, Quynh Dang: Recommendation for Key Management. Part 3: Application-Specific Key Management Guidance. *National Institute of Standards and Technology (NIST)* Special Publication 800-57 Part 3 Revision 1 (2015). DOI: 10.6028/NIST.SP.800-57pt3r1

[Pe-02] Asher Peres: *Quantum Theory: Concepts and Methods.* Kluwer Academic Publishers, New York, Boston, Dordrecht, London, Moscow, 2002. ISBN 0-792-33632-1, e-ISBN: 0-306-47120-5

[Re-04] Oded Regev: Quantum computation and lattice problems. *SIAM J. Computing* **33**, n. 3 (2004), p. 738-760. DOI: 10.1137/S0097539703440678

[RSA-21] Wikipedia: RSA Factoring Challenge. *Wikimedia Foundation* (15 nov. 2021). `https://en.wikipedia.org/wiki/RSA_Factoring_Challenge`

[Se-67] Jean-Pierre Serre: *Representaciones lineales de los grupos finitos.* Traducció de Sebastián Xambò (original del 1967). Ediciones Omega, Barcelona, 1970.

[Sh-94] Peter W. Shor: Algorithms for quantum computation: discrete logarithms and factoring: *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (1994), p. 124–134. DOI: 10.1109/SFCS.1994.365700

[Sh-97] Peter W. Shor: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Computing* **26**, n. 5 (1997), p. 1484–1509. DOI: 10.1137/S0097539795293172

[Tr-98] Artur Travesa: *Aritmètica.* Edicions Universitat de Barcelona, Barcelona, 1998. ISBN: 84-8338-031-5

[Un-95] William G. Unruh: Maintaining coherence in Quantum Computers. *Physical Review A* **51**, n. 2 (1995), p. 992-997. DOI: 10.1103/PhysRevA.51.992

[Wi-11] Colin P. Williams: *Explorations in quantum computing. 2a edició* (edició original, 1998), Springer-Verlag, London, 2011. ISBN: 978-1-84628-886-9, e-ISBN: 978-1-84628-887-6 DOI: 10.1007/978-1-84628-887-6