Erika Andersson

är professor i fysik vid Heriot-Watt University i Edinburgh, där hon varit verksam sedan 2007. Hon forskar inom kvantinformation och kvantoptik. Efter doktorsexamen år 2000 vid KTH i Stockholm arbetade hon först vid Strathclyde University i Glasgow som Marie Curie-stipendiat och Royal Society Dorothy Hodgkin Fellow. Erika är även medredaktör inom kvantinformation för Physical Review A, en av de största vetenskapliga tidskrifterna inom atom- och molekylfysik samt optisk fysik.

Kvantmekaniska egenskaper, som superposition och sammanflätning, är inte bara märkliga och fascinerande. De kan vara användbara också. Här berättar Erika Andersson om hur dessa fenomen skulle kunna utnyttjas i framtidens datorer, om vilka vinster det skulle kunna innebära, och om de problem som först måste övervinnas.

Kvantdatorer – superdatorer i superposition

Forskningen inom kvantinformation och kvantteknologi har tagit fart på allvar under de senaste 20 åren. Sedan en tid tillbaka arbetar även företag som IBM, Intel och Google på att utveckla kvantdatorer, kvantsimulatorer och kvantalgoritmer. D-Wave, ett kanadensiskt företag, utvecklar en dator som sägs använda sig av ”quantum annealing”. IBM har till och med lanserat en liten kvantdator, ”The IBM Quantum Experience”, som vem som helst kan köra egna kvantprogram på över internet.

Vad är det då som gör kvantdatorer så intressanta? Och vad är de bra på? Tanken är att kvantdatorer skulle kunna använda sig av kvanteffekter som superposition och sammanflätning för att göra beräkningar som vanliga ”klassiska” datorer inte klarar av. Men kvantdatorer kommer antagligen aldrig att helt ersätta vanliga datorer. Istället är det så att det finns vissa specifika problem som kvantdatorer är bättre på. Möjliga användningsområden finns till exempel inom kryptografi, design av nya material och mediciner, maskinlärning, bioinformatik och logistik.

Hur fungerar en kvantdator?

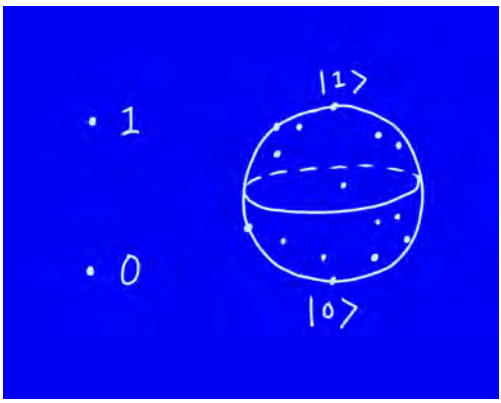
Den mest grundläggande skillnaden mellan en vanlig ”klassisk” dator och en kvantdator är hur information representeras. I en klassisk digital dator används *bitar* för att representera information. En bit kan anta antingen värdet 0 eller 1. Med logiska operationer kan man addera bitar till varandra eller multiplicera dem, och också utföra mer komplicerade beräkningar. Det är en hisnande tanke att det fortfarande är så här som våra allt mer kom-

plicerade datorer och smarttelefoner fungerar: inuti dem finns elektroniska kretsar som utför operationer med bitar, där nollor och ettor kan representeras exempelvis med låg eller hög spänning mellan delar av en elektronisk krets.

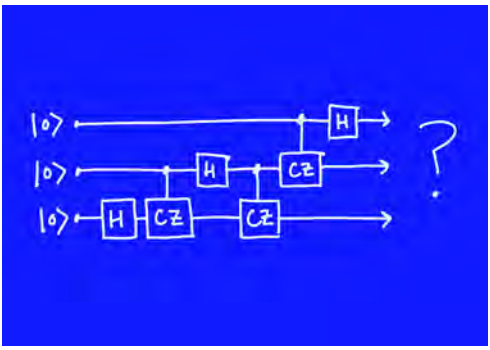
I en kvantdator representeras information istället med kvantsystem. En *kvantbit* är ett kvantmekaniskt tvånivåsystem; istället för ”1” och ”0” har vi två olika kvanttillstånd. Kvantbitar kan exempelvis representeras med hjälp av polarisationen hos enskilda fotoner, med energinivåer hos enskilda atomer eller med olika kvantiserade tillstånd hos en supraledande krets. En kvantbit måste inte heller nödvändigtvis bestå av ett enskilt kvantsystem – en supraledande krets, till exempel, kan involvera ett stort antal ledningselektroner och atomer och kan vara mikrometerstor. Kvantbiten måste dock bete sig som ett kvantiserat system. En och samma kvantdator kan även använda mer än en sorts kvantbitar, beroende på om en kvantbit just då förvaras i kvantdatorns minne, ingår i ett beräkningssteg, eller kanske är en ”flygande kvantbit” som förmedlar kvantlogiska operationer mellan andra kvantbitar.

Kvantbitar i superposition

Den avgörande skillnaden mellan klassiska bitar och kvantbitar är att kvantsystem, och därmed kvantbitar, kan befinna sig i superpositionstillstånd. Förutom i sina bastillstånd, som i Dirac-notation kan skrivas $|0\rangle$ och $|1\rangle$ (och som vi kan tänka på som basvektorer), så kan en kvantbit befinna sig i en superposition av dessa, som vi kan skriva som $a|0\rangle + b|1\rangle$, där a och b är komplexa tal med $|a|^2 + |b|^2 = 1$. Vi kan tänka på detta tillstånd som en



Figur 1: En klassisk bit har enbart två möjliga tillstånd, 1 och 0 (vänster). Ett allmänt superpositionstillstånd hos en kvantbit, däremot, motsvaras av någon av punkterna på en sfär (höger).



Figur 2: I en kvantdator utförs beräkningar genom att man i en följd gör kvantoperationer på kvantbitarna i kvantdatorn. ”H” och ”CZ” står för två olika kvantoperationer: H verkar på en enskild kvantbit, och ger upphov till superpositionstillstånd, medan CZ verkar på två kvantbitar och resulterar i att dessa blir sammanflätade. Se vidare sidorutan om kvantoperationer.

tvådimensionell vektor med komplexa koefficienter a och b. Det är egentligen bara kvoten mellan koefficienterna a och b som spelar roll, vilket innebär att tillståndet kan beskrivas med ett enda komplext tal. Det visar sig därmed möjligt att representera kvantbitens tillstånd entydigt som en punkt på ytan av en sfär. Istället för bara de två diskreta lägena för en vanlig klassisk bit, så motsvaras de möjliga lägena för en kvantbit av sfärens alla punkter (se figur 1).

Kvantbitens bastillstånd $|0\rangle$ och $|1\rangle$ kan till exempel väljas som den horisontella respektive vertikala polarisationen hos en enskild foton. Men en foton med linjär polarisation kan förstås ha vilken riktning på sin polarisation som helst, inte bara horisontell eller vertikal. En allmän polarisationsriktning beskrivs då av superpositionen $a|0\rangle + b|1\rangle$. Genom att koefficienterna a och b är komplexa, kan vi även beskriva cirkulär och elliptisk polarisation på detta sätt.

Kvantkretsar och universella kvantoperationer

I en klassisk dator utförs beräkningar genom sekvenser av logiska operationer på bitar i datorns register. Man kan visa att det räcker att utföra logiska operationer på två bitar i taget för att kunna utföra vilken beräkning som helst, om operationerna bara kombineras på rätt sätt. På liknande sätt kan beräkningar med en kvantdator utföras genom att man i en följd gör kvantoperationer på datorns kvantbitar (se figur 2). Detta ger den så kallade ”kretsmodellen” för hur en kvantdator fungerar. Ur fysikalisk synvinkel åstadkommer operationerna tillsammans en önskad tidsutveckling för kvantbitarna i kvantdatorn, nämligen den som motsvarar

den kvantalgoritm man vill utföra. När man sedan läser av kvantbitarnas tillstånd i slutet av beräkningen, så ger mätresultatet informationen som man är ute efter.

En *universell* kvantdator kan utföra vilken kvantberäkning som helst. Det innebär att den ska kunna åstadkomma vilken tidsutveckling som helst, som är förenlig med fysikens lagar. Jämfört med det fåtal olika logiska operationer som kan utföras på två klassiska bitar, så kan ett register med kvantbitar växelverka på väldigt många olika sätt. Därför kunde man tro att det skulle behövas väldigt många olika sorters operationer för att styra kvantbitarna i en universell kvantdator. Men, som tur är, kan man visa att enbart genom att kombinera vissa basoperationer, kan man åstadkomma vilken operation – eller tidsutveckling – som helst på obegränsat många kvantbitar. Detta påminner om hur det i klassisk Boolesk algebra finns logiska operationer som är universella, så att vilken (klassisk) logisk beräkning som helst, på i princip hur många logiska bitar som helst, kan konstrueras med bara den sortens operationer.

Mer specifikt visar det sig att man, för att konstruera en universell kvantdator, bara behöver kunna göra två typer av operationer: dels måste man kunna göra vilken operation som helst på *enskilda* kvantbitar, dels måste man kunna låta två kvantbitar åt gången växelverka så att de försätts i ett *sammanflätat* tillstånd. Operationer på enskilda kvantbitar kan visualiseras som rotationer av vektorn som representerar kvantbitens tillstånd. Dessa är ofta relativt enkla att åstadkomma i praktiken. Exempelvis kan laserpulser användas för att kontrollera superpositionstillståndet hos en atom eller en jon, och vågplattor eller integrerad optik kan användas för att förändra polarisationen hos enskilda fotoner. Sammanflätning är ofta svårare att åstadkomma. Sammanflätade kvanttillstånd är också känsliga och förstörs lätt av omgivningens störningar. Detta är den huvudsakliga orsaken till att det är så svårt att bygga stora kvantdatorer. Samtidigt som man måste kunna kontrollera kvantbitarna, måste de vara utomordentligt väl isolerade från sin omgivning.

Några viktiga kvantoperationer

NOT

En NOT-operation inverterar en enskild kvantbit så att $|0\rangle$ blir $|1\rangle$, och vice versa. Denna operation ändrar därmed superpositionstillståndet $a|0\rangle + b|1\rangle$ till $b|0\rangle + a|1\rangle$. Till skillnad från en klassisk Boolesk NOT-operation, så finns det vissa kvanttillstånd som lämnas oförändrade av en NOT-operation. Dessa är $1/\sqrt{2}\,(|0\rangle + |1\rangle)$ och $1/\sqrt{2}\,(|0\rangle - |1\rangle)$. (Det senare tillståndet är effektivt detsamma som $1/\sqrt{2}\,(|1\rangle - |0\rangle)$, eftersom samma fasfaktor multiplicerad med alla komponenter i en tillståndsvektor inte ger någon detekterbar skillnad.)

Hadamard

En Hadamard-operation förändrar en enskild kvantbit i bastillståndet $|0\rangle$ till superpositionen $1/\sqrt{2}\,(|0\rangle + |1\rangle)$. Bastillståndet $|1\rangle$ förändras till $1/\sqrt{2}\,(|0\rangle - |1\rangle)$. Detta leder till att om vi startar med ett helt register av kvantbitar, alla i tillståndet $|0\rangle$, och utför en Hadamard-operation på varje enskild kvantbit, så blir resultatet en superposition av alla möjliga bastillstånd. Detta är mycket användbart i kvantalgoritmer.

Som ett exempel, betrakta fallet med tre kvantbitar. Då får vi

$$H|0\rangle \otimes H|0\rangle \otimes H|0\rangle = \frac{1}{2\sqrt{2}}\big(|0\rangle + |1\rangle\big) \otimes \big(|0\rangle + |1\rangle\big) \otimes \big(|0\rangle + |1\rangle\big) =$$
$$= \frac{1}{2\sqrt{2}}\Big(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle\Big)$$

Här betyder tecknet för tensorprodukt $\otimes$ att två eller flera kvantbitar kombineras. Hadamard-operationen har fått sitt namn av att dess matrisrepresentation är proportionell mot en så kallad *Hadamard-matris*, vars element är +1 eller –1, och där samtliga rad- och kolumnvektorer är ortogonala.

Kontrollerad NOT

En ”kontrollerad NOT” eller CNOT-operation är ett exempel på en kvantoperation som kan åstadkomma sammanflätning av kvantbitar. Den verkar på två kvantbitar på så vis att om den första kvantbiten, ”kontrollkvantbiten”, är $|0\rangle$, så händer inget med den andra kvantbiten. Men om den första kvantbiten är $|1\rangle$, så inverteras den andra kvantbiten, dvs. om dess tillstånd är $|0\rangle$ så ändras det till $|1\rangle$, och vice versa. Det betyder att bastillstånden $|00\rangle$ och $|01\rangle$ förblir oförändrade, medan $|10\rangle$ och $|11\rangle$ blir $|11\rangle$ respektive $|10\rangle$.

Säg att vi startar med den första kvantbiten i superpositionstillståndet $1/\sqrt{2}\,(|0\rangle + |1\rangle)$ och den andra i bastillståndet $|0\rangle$, så att deras sammanlagda tillstånd kan skrivas

$$\frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big) \otimes |0\rangle = \frac{1}{\sqrt{2}}\big(|00\rangle + |10\rangle\big)$$

Om vi nu utför en CNOT-operation, så förändras kvantbitarnas tillstånd till

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

Detta tillstånd – till skillnad från det vi startade med – är sammanflätat. Det vill säga, det går *inte* att skriva det som en kombination av superpositionstillstånd för de individuella kvantbitarna,

$$\big(a|0\rangle + b|1\rangle\big) \otimes \big(c|0\rangle + d|1\rangle\big)$$

för några val av koefficienterna *a*, *b*, *c* och *d*. De två kvantbitarnas tillstånd kan efter CNOT-operationen endast beskrivas som en enhet.

Kontrollerad fasoperation, CZ

En annan ofta använd sammanflätande operation på två kvantbitar, ”kontrollerad-Z” eller CZ, innebär att $|11\rangle$ förändras till $-|11\rangle$, medan alla andra bastillstånd lämnas oförändrade.

Exempelvis får vi då

$$\begin{aligned}\mathrm{CZ}\frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big) \otimes \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big) &= \\ = \mathrm{CZ}\frac{1}{2}\Big(|00\rangle + |01\rangle + |10\rangle + |11\rangle\Big) &= \\ = \frac{1}{2}\Big(|00\rangle + |01\rangle + |10\rangle - |11\rangle\Big)\end{aligned}$$

som är ett sammanflätat tillstånd. Benämningen CZ kommer sig av att en så kallad Z-operation på en enskild kvantbit lämnar $|0\rangle$ oförändrat, medan $|1\rangle$ ändras till $-|1\rangle$. Dessa bastillstånd multipliceras alltså med olika fasfaktorer (+1 eller –1).

Sammanflätade kvanttillstånd

För att beskriva ett sammanflätat tillstånd räcker det inte att beskriva de enskilda kvantbitarnas tillstånd var för sig, utan tillståndet måste beskrivas som en enhet. Sådana tillstånd har ingen motsvarighet i klassisk fysik, och har gett upphov till mycket huvudbry bland fysiker och filosofer. Det välkända tankeexperimentet ”Schrödingers katt”, till exempel, innefattar ett sammanflätat tillstånd. Katten tycks här vara både levande (ifall en radioaktiv atom inte har sönderfallit) och död (ifall atomen har sönderfallit) på samma gång. Kattens och atomens tillstånd är sammanflätat.

Atomen kan befinna sig i ett superpositionstillstånd, så varför inte katten?

För att en kvantalgoritm skall vara mer effektiv än en klassisk algoritm, så är det ofta nödvändigt att sammanflätade kvanttillstånd uppstår under beräkningens gång. Det finns enstaka exempel på kvantalgoritmer där det inte uppstår sammanflätning, men också i dessa fall behövs superpositioner av de två bastillstånden hos enskilda kvantbitar.

Kvantparallellism

Superpositionstillstånd av många kvantbitar ger upphov till så kallad *kvantparallellism*. *N* klassiska bitar kan totalt befinna sig i 2^N lägen. Tre bitar kan till exempel ha 2^3 =8 olika lägen: 000, 001, 010, 011, 100, 101, 110 och 111. På samma sätt har *N* kvantbitar 2^N olika bastillstånd, när man kombinerar bastillstånden för individuella kvantbitar. Men skillnaden är att *N* kvantbitar kan befinna sig i vilken superposition som helst av dessa 2^N bastillstånd! Ett helt allmänt tillstånd för tre kvantbitar, till exempel, kan således skrivas

$$a_0|000\rangle + a_1|001\rangle + a_2|010\rangle + a_3|011\rangle + \\ + a_4|100\rangle + a_5|101\rangle + a_6|110\rangle + a_7|111\rangle$$

där de åtta koefficienterna a_0, a_1, … a_7 är komplexa tal. I kvantalgoritmer drar man nytta av att kvantbitarna kan befinna sig i sådana superpositionstillstånd. Det är detta som kallas kvantparallellism.

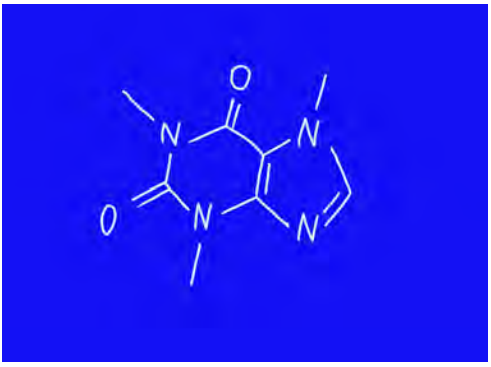
Figur 3: Kvantbitar kan befinna sig i sammanflätade tillstånd, där de inte kan beskrivas var för sig utan måste beskrivas som en enhet. Ett sammanflätat tillstånd för två kvantbitar är exempelvis en superposition av $|00\rangle$ *och* $|11\rangle$*. Antingen är båda kvantbitarna* $|0\rangle$*, eller så är båda* $|1\rangle$*. I en kvantdator uppstår sammanflätade superpositionstillstånd under beräkningens gång.*

Kvantsimulatorer

Att beskriva det sammanlagda superpositionstillståndet för N kvantbitar betyder att vi måste specificera alla koefficienter, en för varje bastillstånd. Antalet koefficienter är 2^N, ett antal som alltså växer exponentiellt med N. Av den här orsaken är det svårt att med en klassisk dator beräkna hur ett stort antal växelverkande kvantsystem beter sig tillsammans. För varje kvantmekaniskt tvånivåsystem vi lägger till, så dubbleras mängden minne som skulle behövas. Fler än några tiotal tvånivåsystem är det i dagens läge därför omöjligt att klara av även med de mest kraftfulla datorerna. Detta utgör ett hinder exempelvis för att räkna ut egenskaper hos stora molekyler eller olika material, och är en delorsak till att vi ännu inte riktigt vet hur fenomen som högtemperatursupraledning fungerar.

I en *kvantsimulator* används i stället andra kvantsystem för att representera det fysikaliska system man är intresserad av. Detta skulle kunna göra det enklare och billigare att hitta nya material med vissa egenskaper, till exempel för att utveckla nya mediciner.

Men varför gör vi inte i stället experiment direkt med de material eller molekyler som vi är intresserade av? Därför att detta kan vara både svårt och dyrt. Säg till exempel att man på kemisk väg lyckas tillverka en viss sorts ny molekyl, bara för att sedan upptäcka att den inte har de egenskaper man hade hoppats på. En kvantsimulator som förverkligar samma sorts fysik kan då vara mera praktisk. I ett verkligt material har man dessutom begränsade möjligheter att variera relevanta parametrar – som till exempel styrkan i växelverkan mellan materialets atomer – medan en kvantsimulator är flexibel, och kan användas till att undersöka många olika varianter av det material man är intresserad av. Om



Figur 4: En kvantdator skulle kunna användas för att effektivt och flexibelt simulera hur komplicerade kvantsystem, till exempel molekyler, beter sig. Detta är viktigt exempelvis för att utforma nya material och mediciner.

man till exempel vill se vad som händer ifall man byter ut en sorts atom mot en annan, så är detta mycket lättare om man använder en kvantsimulator i stället för det riktiga materialet.

En universell kvantdator, som kan förverkliga en godtycklig kvantalgoritm eller tidsutveckling av kvantbitarna i sitt register, kan givetvis också användas för kvantsimulering. Men en kvantsimulator kan också vara en kvantdator som inte är universell, utan i stället specialbyggd för en viss typ av problem. Relativt små sådana kvantsimulatorer skulle redan kunna slå klassiska datorer. Eftersom klassiska datorer på sin höjd klarar av att simulera ett tiotal sammankopplade tvånivåsystem, så kunde en kvantsimulator med bara några tiotal kvantbitar redan vara användbar.

Kvantalgoritmer

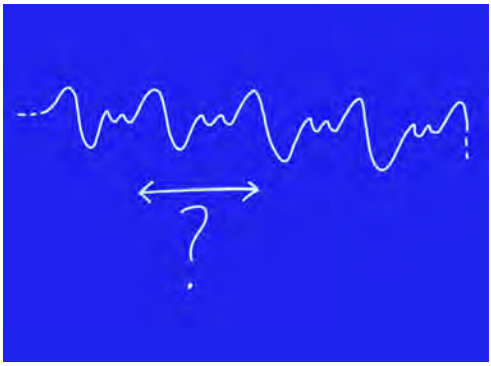
Kvantparallellism och superposition är viktiga beståndsdelar i kvantalgoritmer. Det ligger nära till hands att tänka sig att medan en klassisk dator måste utföra en beräkning 2^N gånger för att pröva alla möjliga kombinationer av N bitar indata, så kunde en kvantdator klara sig med att utföra beräkningen endast *en* gång, fast med en superposition av alla möjliga bastillstånd. Riktigt så enkelt är det inte. Även om kvantbitarna i en kvantdator under beräkningens gång kan befinna sig i sammanflätade superpositionstillstånd av exponentiellt många bastillstånd, så finns det en hake. Vid beräkningens slut måste slutresultatet på något sätt läsas ut ur kvantdatorn, och en kvantmätning kan ge bara *ett* resultat. Vi kan därför inte läsa ut exponentiellt mycket information ur kvantdatorn. Man kan faktiskt visa att endast en klassisk bit information kan läsas ut per kvantbit. För att inte gå miste om fördelen med kvantparallellism måste vi därför ha en smart metod att få ut användbar information ur kvantdatorn.

Det är här kvantalgoritmer kommer in i bilden. I allmänhet fungerar en kvantalgoritm så att kvantdatorn mer effektivt kan nå fram till sitt resultat, genom att fiffigt använda sig av superpositioner och sammanflätning. Det är inte lätt att hitta nya kvantalgoritmer, men ett stort antal finns redan för en mängd olika situationer. Ett par av de mest kända är Shors och Grovers algoritmer. Shors algoritm är en metod för att faktorisera tal, medan Grovers algoritm är ett effektivt sätt att söka igenom stora datamängder. Låt oss titta lite närmare på dessa.

Shors kvantalgoritm och ”kryptokalypsen”

Shors algoritm bygger på att kvantmekaniken medger ett effektivt sätt att upptäcka periodicitet i en funktion (figur 5). Faktorisering är viktigt eftersom den vanligaste metoden vid kryptering, så kallad RSA (uppkallad efter dess skapare Rivest, Shamir och Adleman), använder sig av det faktum att det är lätt att multiplicera tal, men betydligt svårare att hitta primtalsfaktorerna för ett givet stort tal – i varje fall med vanliga klassiska datorer. Dekryptering i RSA förutsätter att man känner till primtalsfaktorerna hos ett visst stort tal; primtalsfaktorerna är en *privat nyckel*. Kryptering å andra sidan svarar mot multiplikation, och vem som helst som känner till det stora talet (men inte dess primtalsfaktorer) kan kryptera meddelanden. Varje gång vi surfar på internet eller gör våra bankärenden med mobiltelefonen eller datorn, så använder vi kryptering, och ofta kryptering med just RSA. Vårt moderna samhälle är totalt beroende av effektiva och säkra krypteringsmetoder.

Ifall vi kan bygga en kvantdator som kan köra Shors algoritm för att hitta primtalsfaktorer, så är RSA är inte längre en säker krypteringsmetod. Detta skulle ha mycket stora följder och kallas ibland för ”kryptokalypsen”. Av den här anledningen har bland annat den amerikanska National Security Agency börjat rekommendera att krypteringsmetoder som bygger på RSA, liksom andra metoder som också är känsliga för dekryptering med kvantdatorer, så småningom tas ur bruk. Forskare har börjat utveckla nya krypteringsmetoder som tros vara säkra även i ett framtida samhälle med kvantdatorer. En möjlig dellösning baserar sig på *kvantkryptografi*, dvs. kvantmekaniska metoder för att distribuera hemliga nycklar som sedan kan användas för säkrare symmetrisk kryptering. Även om kvantkryptografi knappast kommer att vara



Figur 5: Shors algoritm kan faktorisera stora tal genom att en kvantdator effektivt kan hitta periodicitet. Faktorisering är viktigt eftersom vanligt förekommande metoder för kryptering bygger på att det är svårare att hitta primtalsfaktorerna i stora tal än att utföra multiplikation.

den enda lösningen, så utvecklar redan företag som IdQuantique i Genève, och Toshiba, utrustning för kvantkryptografi.

Hur allvarligt är kvanthotet mot RSA? En kvantdator som är tillräckligt stor för att kunna knäcka RSA, med så stora primtal som i praktiken kunde användas, kommer antagligen att behöva kunna hantera en biljon kvantbitar eller mer. Det är svårt att uppskatta antalet exakt, eftersom en stor del av kvantbitarna i en kvantdator används för felkorrigering, vilket innebär att det totala antalet kvantbitar beror på hur mycket felkorrigering som behövs. Ju noggrannare kvantbitarna i datorns register kan styras, desto mindre felkorrigering behövs. I vilket fall som helst skulle det krävas mycket stora kvantdatorer för att göra RSA osäkert. Men även om vi inte ännu på flera år eller kanske decennier lyckas bygga kvantdatorer som är tillräckligt stora för att knäcka RSA, är det viktigt att redan nu planera för vad som händer om eller när man lyckas. För det första kan det ta mycket lång tid att utveckla och byta ut krypteringsmetoder – man får nog räkna med åtminstone tio år. För det andra skulle man i efterhand kunna dekryptera dagens datatrafik med framtidens kvantdatorer. Så om en kvantdator som kan knäcka RSA kommer att finnas om tio eller tjugo år, börjar vi redan vara sent ute.

Grovers sökalgoritm

Grovers kvantalgoritm är en metod för sökning i ostrukturerad data. Som exempel, tänk på en gammaldags telefonkatalog med namnen alfabetiskt ordnade. Det är lätt att snabbt slå upp numret för en viss person som vi vet namnet på. Men om vi i stället vill veta vem som har ett visst telefonnummer, så måste vi börja söka igenom katalogen, namn för namn. I medeltal kommer vi att behöva gå igenom halva katalogen innan vi hittar rätt namn och nummer, och det kan råka sig så illa att det är det sista namnet som är det vi söker. Låt oss säga att katalogen har N namn. Vi har då alltså i medel kontrollerat $N/2$ namn. Med Grovers algoritm, däremot, skulle vi bara behöva ”slå upp” katalogen ett antal gånger som är proportionellt mot $\sqrt{N}$. Ju större katalogen är, desto större blir besparingen. Det som gör detta möjligt är att vi i kvantversionen av telefonkatalogen med hjälp av Grovers algoritm kan ”slå upp numret” för superpositioner av namn. Denna kvantparallellism är givetvis inte tillräcklig i sig. Vi måste även ha en metod att

på något vis av alla möjliga nummer få ut just det rätta. Utan att gå in på matematiska detaljer så kan Grovers algoritm förstås på ett geometriskt sätt. Tillståndet hos kvantbitarna i datorns register kan ses som en N-dimensionell vektor. Steg för steg roteras denna vektor mot ett sluttillstånd som representerar det namn vi söker.

Grovers algoritm är stokastisk i den meningen att slutresultatet inte alltid är rätt namn, eller, mer allmänt, rätt dataobjekt. Resultatet är det vi söker endast med hög sannolikhet. Detta är inget problem, eftersom det är lätt att i efterhand kontrollera att numret stämmer – gör det inte det, är det bara att köra algoritmen igen. I medeltal behöver vi fortfarande bara ”slå upp” katalogen ett antal gånger som är proportionellt mot $\sqrt{N}$. Många kvantalgoritmer – och även många klassiska algoritmer – är stokastiska i den här bemärkelsen.

Grovers algoritm ger inte kvantdatorer en lika stor fördel över klassiska datorer som Shors algoritm. Vinsten i antal steg som behövs är bara från N till $\sqrt{N}$, medan skillnaden för Shors algoritm är exponentiell. Men i gengäld kan Grovers algoritm användas allmänt för att snabba upp även andra klassiska sök- och optimeringsalgoritmer, eller som del i andra kvantalgoritmer. Till exempel kan metoden användas för att hitta det minsta elementet i en osorterad lista med tal, för att bestämma ifall ett nätverk är sammankopplat eller inte, eller för att hitta mönster i data, något som är användbart exempelvis inom textanalys eller bioinformatik.

Adiabatiska kvantdatorer

En kvantdator måste inte nödvändigtvis fungera så att man utför operationer i en följd på kvantbitar i ett register, som i ”kretsmodellen”. I en *mätningsbaserad* kvantdator utgår man istället från ett stort sammanflätat tillstånd bestående av många kvantbitar. Tillståndet ser likadant ut oberoende av vilken beräkning som ska utföras. Vilken beräkning som görs bestäms istället av hur man mäter – dvs. läser av – kvantbitarna (figur 7).

Ett annat alternativ som är speciellt värt att nämna är adiabatiska kvantdatorer. Här kodar man om problemet man är intresserad av på så vis att lösningen representeras av grundtillståndet för ett system av kvantbitar som växelverkar med varandra på ett komplicerat sätt – så komplicerat att det är svårt att

Deutsch-Jozsas algoritm – är funktionen konstant eller balanserad?

För att riktigt förklara hur Shors algoritm kan faktorisera stora tal genom att upptäcka periodiciteter, eller hur Grovers algoritm kan hitta rätt objekt i en stor databas, behöver man gå in på detaljerna i den matematik som beskriver algoritmerna. Ett enklare exempel, som ändå kan ge en antydan om hur kvantalgoritmer fungerar, är Deutsch-Jozsas algoritm. Denna kan avgöra om en *Boolesk funktion* är *balanserad* eller *konstant*. Låt mig förklara vad som menas med dessa begrepp.

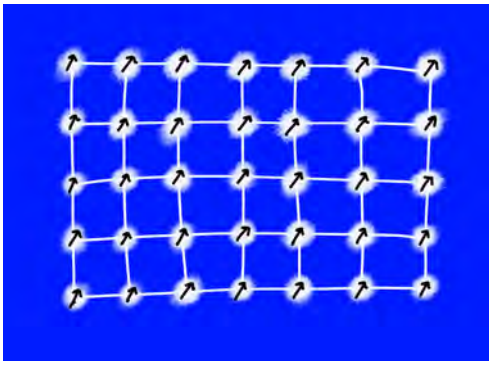
En Boolesk funktion är en funktion där både variablerna (dvs. ingångsvärdena) och funktionsvärdet är ettor och nollor. Detta kan tyckas abstrakt, men Booleska funktioner kan användas för att beskriva en mängd fenomen, exempelvis hur nätverk är sammankopplade, om ett tal är ett primtal eller inte, eller hur utgången av ett val beror på hur väljarna röstar. Algoritmer som effektivt kan avgöra om en Boolesk funktion har en viss egenskap eller inte är därför viktiga ur beräkningssynpunkt.

	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
x=0	0	0	1	1
x=1	0	1	0	1

Figur 6: Tabell över de fyra olika möjliga Booleska funktionerna med en binär funktionsvariabel x och ett binärt funktionsvärde. Funktionerna $f_2(x)$ och $f_3(x)$ är balanserade, medan $f_1(x)$ och $f_4(x)$ är konstanta.

Den enklaste Booleska funktionen har en binär funktionsvariabel, som alltså kan vara 0 eller 1, och ett binärt funktionsvärde, som alltså också kan vara 0 eller 1. Det finns då bara fyra olika möjliga sådana Booleska funktioner, se tabellen i figur 6. Två av dessa har olika funktionsvärden för variablerna 0 och 1; dessa sägs vara *balanserade*. De andra två möjliga Booleska funktionerna har samma funktionsvärde för båda variablerna; dessa kallas *konstanta*. Om vi vill veta om en okänd Boolesk funktion är balanserad eller konstant, måste vi – om vi använder en klassisk dator – först se efter vad funktionsvärdet är för variabelvärdet 0, sedan vad det är för variabelvärdet 1, och slutligen jämföra dessa. För detta behövs således två funktionsevalueringar.

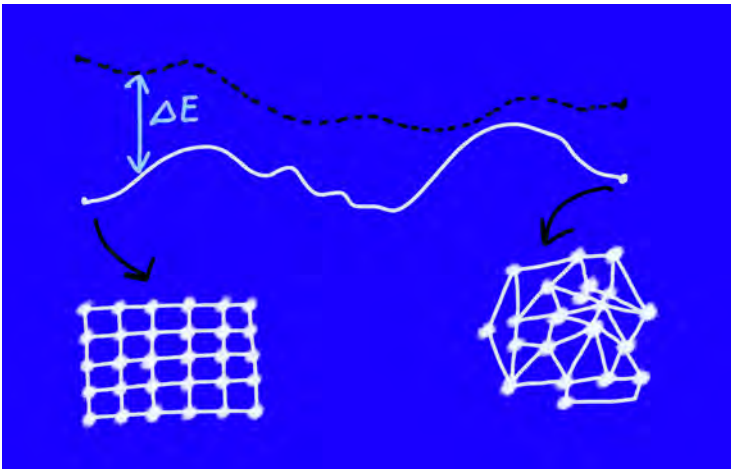
Ändå kan informationen om huruvida funktionen är balanserad eller konstant ändå beskrivas enbart med en bit information. Deutsch-Jozsas algoritm klarar av att extrahera just den informationen med endast *en* funktionsevaluering, genom att den kvantbit som representerar funktionsvariabeln kan befinna sig i ett superpositionstillstånd. Å andra sidan säger kvantalgoritmen bara just detta – om funktionen är konstant eller balanserad – och ger ingen information om de individuella funktionsvärdena för ingångsvärdena 0 och 1. För en Boolesk funktion av endast en variabel är besparingen liten. Algoritmen kan dock generaliseras, så att vi kan få veta om en Boolesk funktion av N variabler är balanserad eller konstant med endast *en* funktionsevaluering, oberoende av N. Med klassiska metoder ökar antalet funktionsevalueringar linjärt med N.



Figur 7: I en mätningsbaserad kvantdator skapas först ett nätverk av sammanflätade kvantbitar. Efter detta utförs beräkningen genom att man mäter de enskilda kvantbitarnas tillstånd en efter en. Nätverket ser från början alltid likadant ut, och vilken kvantalgoritm som utförs bestäms av exakt hur mätningarna görs.

klassiskt beräkna vad det kvantmekaniska grundtillståndet är. Sedan startar man från en enklare situation med ett enkelt grundtillstånd, och ställer långsamt om växelverkningarna så att man till slut har det mer komplicerade mönstret av växelverkningar. Ifall omställningen görs tillräckligt långsamt förblir kvantbitarna i grundtillståndet igenom hela proceduren, och man kan sedan göra mätningar på det svårberäknade grundtillståndet (figur 8).

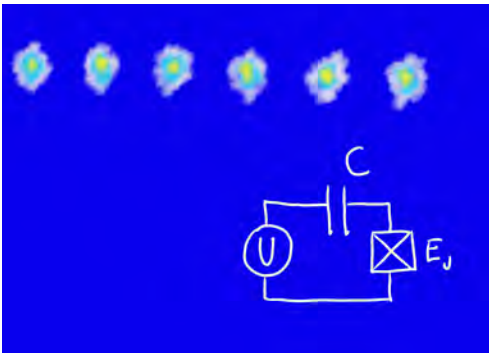
Huruvida denna adiabatiska metod är effektiv eller inte beror på hur långsamt omställningen måste ske. Detta bestäms av hur stor energiskillnaden är mellan grundtillståndet och det första exciterade tillståndet. Ju mindre skillnaden är, desto långsammare måste omställningen ske för att kvantdatorn ska förbli i grundtillståndet. Blir energiskillnaden för liten för ett visst beräkningsproblem, så ger kanske inte metoden någon fördel jämfört med en klassisk beräkning. Datorerna som kanadensiska D-Wave Systems har byggt av supraledande kretsar fungerar ungefär så här, men med den skillnaden att datorn inte befinner sig i grundtillståndet, utan i ett termiskt tillstånd, visserligen kallt, men inte vid absoluta nollpunkten. Metoden kallas ”quantum annealing” och har likheter med den klassiska optimeringsmetoden ”simulated annealing”. Det är ännu oklart om den här typen av dator verkligen kan ge upphov till samma fördelar som en adiabatisk kvantdator som följer grundtillståndet, och huruvida D-Waves maskiner faktiskt är att betrakta som kvantdatorer är omdebatterat. I vilket fall som helst är de inte universella kvantdatorer – de kan till exempel inte användas för att faktorisera tal med Shors algoritm.



Figur 8: En adiabatisk kvantdator fungerar genom att lösningen till det problem man är intresserad av kodas om så att det representeras av grundtillståndet för ett system av kvantbitar som växelverkar på ett komplicerat sätt. Sedan startar man från en enklare situation, och ställer gradvis om växelverkningarna. För att metoden skall vara effektiv måste energiskillnaden ΔE mellan grundtillståndet och det första exciterade tillståndet förbli tillräckligt stor under hela processen.

Hur långt har vi kommit med att bygga kvantdatorer?

För tillfället är det många forskargrupper världen över som på olika sätt arbetar med att konstruera kvantdatorer. Vi har redan nämnt att kvantbitar kan realiseras med polarisationen hos fotoner, eller med energinivåer hos atomer eller joner, eller med supraledande kretsar (figur 9). Ytterligare ett alternativ är att använda syntetiska diamanter med utplacerade orenheter som ger upphov till kvant-



Figur 9: Kvantbitar kan realiseras exempelvis med hjälp av joner som hålls fångade med elektriska och magnetiska fält, eller med supraledande kretsar.

bitens energinivåer. Alla metoder har sina för- och nackdelar, och det finns ännu inte någon klar vinnare. Svårigheten med att bygga en kvantdator består i att kunna tillverka tillräckligt många kvantbitar som kan kontrolleras väl. Men samtidigt som vi måste kunna styra kvantbitarna – vilket betyder att de måste växelverka både med varandra och med sin omgivning – måste de också isoleras så bra som möjligt från omgivningen. Detta är svårt, eftersom kvantsystem är mycket känsliga för störningar.

Den här artikeln har fokuserat på de generella principerna för hur kvantdatorer fungerar, snarare än på hur långt forskarna har kommit, eller exakt vilka experimentella tekniker som är heta för tillfället. Teknikerna utvecklas och förbättras hela tiden. Nya kvantalgoritmer utvecklas visserligen också, men principerna för hur en kvantdator fungerar – med superposition, sammanflätning och kvantparallellism – förblir desamma.

I framtiden kan vi komma att få se kombinationer av klassiska datorer och kvantdatorer, men än så länge är de experimentella kvantdatorerna inte särskilt kraftfulla. Exempelvis kan kvantdatorer som använder sig av joner fångade i elektriska och magnetiska fält typiskt ha mellan 5 och 10 joner. De beräkningar man har demonstrerat kan vi oftast lika gärna klara av med huvudräkning. Det är svårt att säga hur länge det kommer att dröja innan vi har mer användbara kvantdatorer, men många forskare tror att vi inom fem till tio år kommer att kunna bygga medelstora kvantdatorer med några tiotals kvantbitar. Dessa kan börja slå vanliga klassiska datorer när det gäller kvantsimulering, exempelvis inom kvantkemi, för att räkna ut egenskaper hos stora molekyler. De senaste åren har omfattande forskningsanslag vikts för kvantteknologi – exempelvis en biljon euro inom EU-flaggskeppet för kvantteknologi. Det blir därmed alltmer sannolikt att kvantdatorer faktiskt kan komma att bli verklighet. ❖

För vidare läsning

Neil Savage, *Quantum Computers Compete for "Supremacy"*, Scientific American, 5 juli 2017 (https://www.scientificamerican.com/article/quantum-computers-compete-for-supremacy/)

Abigail Beal, *Inside the weird world of quantum computers*, Wired, 23 mars 2017 (http://www.wired.co.uk/article/quantum-computing-explained)

Maria Schuld, *A quantum boost for machine learning*, Physics World, vol. **30**, nr 3 (DOI: 10.1088/2058-7058/30/3/35).

IBM Quantum Experience – testa att köra en riktig kvantdator! (https://research.ibm.com/ibm-q/)