SHOROV ALGORITEM

LUKA PAPEŽ

Fakulteta za matematiko in fiziko
Univerza v Ljubljani

Shorov algoritem je kvantni algoritem za faktorizacijo števil. Članek začne s šifriranjem RSA, ki ga je mogoče razbiti s Shorovim algoritmom, kar služi kot motivacija za nadaljno obravnavo. Sledi predstavitev teoretičnega ozadja algoritma, vključno z osnovami kvantnih vezij, kvantno Fourierovo transformacijo in oceno faze. Ti koncepti so nato združeni v celovit opis delovanja Shorovega algoritma, njegova uporaba pa je ponazorjena s faktorizacijo števila 15.

SHOR'S ALGORITHM

Shor's algorithm is a quantum algorithm for integer factorization. As a motivating example, this article presents RSA encryption, which is known to be breakable using the Shor's algorithm. Then it introduces the necessary theoretical background, including the basics of quantum circuits, the Quantum Fourier Transform and phase estimation. These concepts are then combined into a complete explanation of Shor's Algorithm, which is applied to factorize the number 15.

1. Uvod

Razvoj kvantnih algoritmov predstavlja pomemben preboj v informacijski teoriji, pri čemer Shorov algoritem sodi med najpomembnejše mejnike na tem področju. To je algoritem za učinkovito faktorizacijo števil, ki ga je leta 1994 razvil Peter Shor [1]. Z njim se lahko demonstrira, kako lahko kvantni računalniki eksponentno pospešijo reševanje določenih problemov v primerjavi s klasičnimi pristopi.

V sklopu tega članka se bomo spoznali z delovanjem Shorovega algoritma in ostalimi kvantnimi načeli, ki so potrebna za njegovo delovanje. To vključuje osnove kvantnih vezij, kvantno Fourierovo transformacijo, ocenjevanje faze, iskanje modularnega reda in algoritem verižnih ulomkov.

Najbolj znana uporaba Shorovega algoritma je dešifriranje šifriranja RSA, ki se uporablja za avtentikacijske sisteme, dokazovanje izvirnosti dokumentov in spletno bančništvo. Kot motivacijo si oglejmo njegovo delovanje. Šifriranje RSA je primer asimetričnega šifriranja, kar pomeni, da temelji na načelu javnega in zasebnega ključa. Za tvorbo ključev uporabimo naslednje korake:

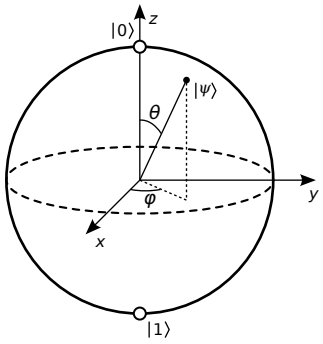
- Izberemo veliki praštevili p in q, ki ju uporabimo za izračun modula $n = p \cdot q$.
- Izračunamo Eulerjevo funkcijo $\varphi(n)$, ki pove število števil, manjših ali enakih n, ki z n nimajo skupnih deliteljev razen ena. Za prej uvedeno število $n = pq$ velja $\varphi(n) = (p-1)(q-1)$ [2].
- Izberemo javni eksponent e tako, da velja $1 < e < \varphi(n)$ in da je največji skupni delitelj e in $\varphi(n)$ enak 1, torej, da je $\gcd(e, \varphi(n)) = 1$. Preostane nam le še iskanje multiplikativnega inverza števila e. Z drugimi besedami, iščemo tak zasebni eksponent d, ki izpolni enačbo $d \cdot e \equiv 1 \pmod{\varphi(n)}$, kar storimo s pomočjo razširjenega Evklidovega algoritma [2].

Tako smo pridobili javni ključ, sestavljen iz modula n in javnega eksponenta e, ter zasebni ključ, ki vsebuje le zasebni eksponent d. Zdaj lahko naše sporočilo m zašifriramo s $c \equiv m^e \pmod{n}$ in dešifriramo s $c^d \equiv (m^e)^d \equiv m \pmod{n}$ [2]. Tu nastopi vloga Shorovega algoritma, saj je predpostavka varnosti šifriranja RSA to, da ne moremo učinkovito faktorizirati modula n in tako pridobiti njegovih prafaktorjev p in q, ki bi omogočila, da poiščemo zasebni eksponent d.

2. Kvantna vezja

2.1 Kubit

Preden se poglobimo v delovanje Shorovega algoritma, se moramo spoznati z nekaj osnovnimi pojmi kvantnega računalništva, ki temelji na uporabi kubita. Iz klasičnega sveta poznamo pojem bitov, ki zavzemajo stanji 0 ali 1. Kubitovo stanje pa tvori linearna kombinacija oziroma superpozicija stanj $|0\rangle$ in $|1\rangle$. Stanje kubita zapišemo s $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, kjer sta α in β kompleksni števili. Velja, da je $|\alpha|^2$ verjetnost, da izmerimo $|0\rangle$, in $|\beta|^2$ verjetnost, da izmerimo $|1\rangle$. Veljati mora tudi $|\alpha|^2 + |\beta|^2 = 1$, da je vsota verjetnosti enaka ena. Tako lahko stanje zapišemo tudi v drugačni obliki, in sicer kot $|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$. S tem vpeljemo dve spremenljivki: azimutski kot φ in polarni kot θ. Ta kota omogočita, da kubit vizualiziramo z Blochovo sfero (slika 1). Polarni kot θ je definiran tako, da stanje $|0\rangle$ ustreza kotu $\theta = 0$ in stanje $|1\rangle$ kotu $\theta = \pi$. Z azimutskim kotom φ pa opišemo rotacijo v ravnini, ki je pravokotna na os med stanjema $|0\rangle$ in $|1\rangle$ [4, str. 13–16].



Slika 1. Vizualizacija stanja $|\psi\rangle$ na Blochovi sferi. Vir: [3]

2.2 Kvantna vrata

Za transformacije stanj v kvantni mehaniki uporabljamo linearne operatorje, ki jih predstavljamo z matrikami. V kvantnem računalništvu linearne operatorje za transformacijo stanj kubitov imenujemo kvantna vrata. Za uporabo matrične oblike operatorjev, stanje kubita zapišemo kot vektor

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix}.$$

Ključna lastnost operatorjev je, da morajo ohranjati vsoto verjetnosti, kar pomeni, da mora biti vsak operator, ki predstavlja kvantna vrata, unitaren. To je ekvivalentno pogoju $U^\dagger U = I$, kjer † označuje hermitsko konjugiranje. Iz tega sledi, da za unitarne operatorje velja zveza $U^{-1} = U^\dagger$, kar pomeni, da je njihove inverze enostavno poiskati.

Primer kvantnih vrat so Paulijeva vrata X, ki služijo kot razširitev klasične operacije NOT. Ta vrata lahko konstruiramo tako, da zahtevamo klasični lastnosti $X|0\rangle = |1\rangle$ in $X|1\rangle = |0\rangle$, iz česar po linearnosti sledi $X(\alpha|0\rangle + \beta|1\rangle) = \beta|0\rangle + \alpha|1\rangle$. Kvantna operacija NOT torej zamenja amplitudi stanj $|0\rangle$ in $|1\rangle$. To linearno preslikavo lahko opišemo tudi z matriko na shemi 2, v kateri je razviden shematični prikaz vrat v vezju.

$$X = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \qquad\qquad |\psi\rangle \text{ —[} X \text{]— } X|\psi\rangle$$

Shema 2. Predstava Paulijevih X vrat kot matrike in v vezju.

Ena izmed najpomembnejših kvantnih vrat so Hadamardova vrata. Njihova pomembnost izhaja predvsem iz lastnosti, da ob delovanju na stanji $|0\rangle$ in $|1\rangle$ naredi superpozicijo, kjer sta verjetnosti za obe stanji enaki. Torej $H|0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ in $H|1\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. O ključnosti te operacije se prepričamo tudi v naslednjih poglavjih, kjer analiziramo delovanje Shorovega algoritma. Prikazana so na shemi 3 [4, str. 17–20].

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad |\psi\rangle \text{ —[} H \text{]— } H\,|\psi\rangle$$

Shema 3. Predstava Hadamardovih vrat kot matrike in v vezju.

Za nadaljnje razumevanje so ključna še fazna vrata. Ta delujejo tako, da koeficientu stanja $|1\rangle$ dodajo faktor faze $e^{i\phi}$, medtem ko stanje $|0\rangle$ ostane nespremenjeno. Spomnimo se tudi prej omenjene Blochove sfere, kjer spreminjanje faze spremeni azimutski kot kvantnega stanja. Ponavadi omenjamo tri fazna vrata: Paulijeva vrata Z, ki dodajo fazo $\phi = \pi$, vrata S z dodatkom faze $\phi = \pi/2$ in vrata T z dodatkom faze $\phi = \pi/4$. Opišemo jih z naslednjimi matrikami:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}.$$

Z nekaj znanja kompleksne analize med temi tremi vrati opazimo medsebojno odvisnost. Velja namreč $Z = S^2$ in $S = T^2$. Ta povezava vodi do posplošitve na rotacijska fazna vrata R_k, ki dodajo fazo $2\pi/2^k$:

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}. \tag{1}$$

S temi vrati opišemo vsa prejšnja vrata z vrednostmi $k = 1, 2, 3$ [4, str. 174, 218].

2.3 Kontrolna kvantna vrata

Ideja kontrolnih kvantnih vrat je, da imamo en kontrolni bit, ki običajno ostane nespremenjen, in drug bit, na katerega želimo delovati z danimi kvantnimi vrati le, ko je prvi bit v stanju 1. Najbolj osnovni primer takih vrat so kontrolna-NOT (CNOT) vrata (tabela 1b). Njihovo delovanje na drugem tarčnem bitu je analogno klasični operaciji XOR, katere delovanje je opisano v tabeli 1a.

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

(a) Delovanje operacije XOR.

Kontrola	Tarča	Kontrola	Tarča (po CNOT)
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

(b) Delovanje vrat CNOT.

Tabela 1. Primerjava operacije XOR in vrat CNOT.

Za opis teh vrat moramo najprej razširiti prostor na dva kubita $|\psi_1\rangle \otimes |\psi_2\rangle$. Tako ustvarimo štiri nova stanja $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ in lahko operator CNOT zapišemo v matrični in vezni obliki na shemi 4 [4, str. 177–178], [5].

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \psi_1 \text{ —●— } \psi_2 \text{ —⊕—}$$

Shema 4. Predstava vrat CNOT kot matrike in v vezju.

Obstaja tudi splošna implementacija kontrole na poljubnih vratih, ki je tu sicer ne navajamo, a se jo v implementaciji Shorovega algoritma pogosto uporabi.

3. Shorov algoritem

Shorov algoritem rešuje matematični problem iskanja faktorjev poljubnega števila, ki ga preoblikujemo v iskanje modularnega reda. Ključna zamisel algoritma je, da najprej na klasičen način obravnavamo posebne primere, ki jih je možno učinkovito rešiti. Primer iskanja reda nato prepustimo kvantnemu algoritmu. Podrobneje bomo potek Shorovega algoritma pojasnili, ko bomo razumeli njegove glavne komponente.

3.1 Uvod v iskanje reda

Delovanje algoritma za iskanje reda temelji na operatorju U, katerega lastne vrednosti vsebujejo informacijo o redu. Tak operator namreč omogoči, da problem rešimo s kvantnim iskanjem lastnih vrednosti. Najprej pojasnimo pojem reda: red števila x po modulu N je najmanjše pozitivno celo število r, ki zadošča enačbi $x^r \equiv 1 \pmod{N}$, pri čemer sta x in N prav tako pozitivni celi števili. Operator U za problem iskanja reda vpeljemo kot

$$U\left|y\right\rangle = \begin{cases} \left|xy \bmod N\right\rangle, & y < N, \\ \left|y\right\rangle, & y \geq N, \end{cases}$$

za $y \in \{0,1\}^L$, kjer je L število potrebnih kubitov za stanje $\left|y\right\rangle$. Za naš problem pa se lahko omejimo na vrednosti $y < N$. Motivacija za to definicijo izhaja iz lastnosti r-te potence operatorja. Recimo, da je λ neka lastna vrednost operatorja U, ki ji pripada lastni vektor $\left|y\right\rangle$:

$$U^r\left|y\right\rangle = \left|x^r y \bmod N\right\rangle = \left|y\right\rangle = \lambda^r\left|y\right\rangle.$$

Iz te enakosti sledi, da mora veljati $\lambda^r = 1$. Rešitve te enačbe so $\lambda = \exp\left[(2\pi i s)/r\right]$ za vsa cela števila s, kjer je $0 \leq s < r$. Ta lastnost potrjuje, da lastne vrednosti operatorja U dejansko kodirajo iskani red r, kar je bil naš cilj pri vpeljavi operatorja.

Pokažimo še, da so vektorji

$$\left|u_s\right\rangle := \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left[\frac{-2\pi i s k}{r}\right] \left|x^k \bmod N\right\rangle \tag{2}$$

lastni vektorji operatorja U in s tem dokažimo, da so vse rešitve prejšnje enačbe $\lambda^r = 1$ tudi lastne vrednosti operatorja U. Za potrditev tega na vektorje $\left|u_s\right\rangle$ delujemo z operatorjem U in dobimo stanje

$$U\left|u_s\right\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left[\frac{-2\pi i s k}{r}\right] \left|x^{k+1} \bmod N\right\rangle.$$

Pokazati moramo torej, da je to stanje sorazmerno prvotnemu vektorju $\left|u_s\right\rangle$. Opazimo, da imamo eno odvečno stanje $\left|x^r \bmod N\right\rangle$ in eno manjkajoče stanje $\left|x^0 \bmod N\right\rangle$ v primerjavi z definicijo vektorja $\left|u_s\right\rangle$

$$\begin{aligned} U\left|u_s\right\rangle = \exp\left[\frac{2\pi i s}{r}\right] \frac{1}{\sqrt{r}} \sum_{k=1}^{r-1} \exp\left[\frac{-2\pi i s k}{r}\right] \left|x^k \bmod N\right\rangle + \\ + \exp\left[\frac{-2\pi i s (r-1)}{r}\right] \left|x^r \bmod N\right\rangle. \end{aligned}$$

Velja pa tudi ciklična lastnost $\left|x^r \bmod N\right\rangle = \left|1\right\rangle = \left|x^0 \bmod N\right\rangle$, ki omogoči, da lahko stanje zapišemo kot

$$U\left|u_s\right\rangle = \exp\left[\frac{2\pi i s}{r}\right] \left|u_s\right\rangle.$$

S tem smo natančno opredelili operator U, lastne vrednosti $\exp(2\pi is/r)$, ki mu pripadajo, in zakaj je njegova definicija smiselna za iskanje reda. Nadaljujemo z obravnavo kvantne Fourierove transformacije, ki je ključni korak algoritma ocenjevanja faze in omogoča učinkovito iskanje lastnih vrednosti [4, str. 226-227].

3.1.1 Kvantna Fourierova transformacija

Fourierova transformacija je integralska transformacija, ki omogoča pretvorbo v frekvenčno domeno. Za primer signalov uporabljamo diskretno Fourierovo transformacijo, ki transformira vektor kompleksnih točk $[x_0, x_1, ..., x_{N-1}]^T$ v drug kompleksen vektor $[y_0, y_1, ..., y_{N-1}]^T$ po naslednjem predpisu:

$$y_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} x_k.$$

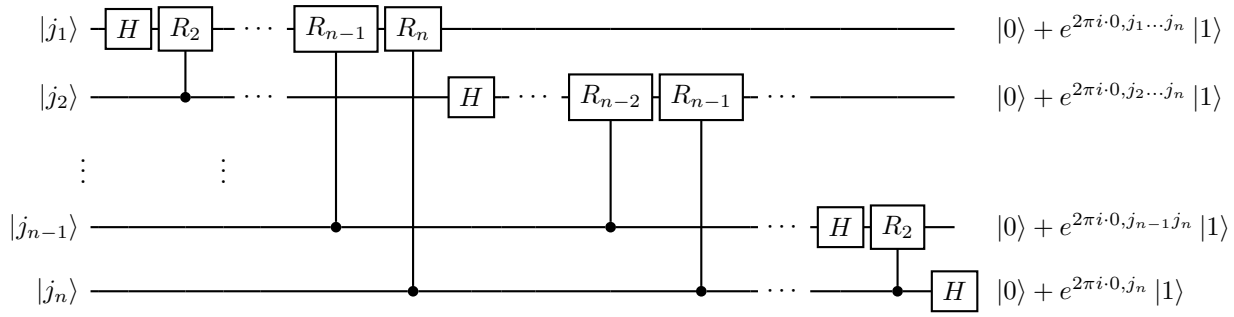
Kvantna Fourierova trasformacija analogno transformira kvantna stanja namesto vektorjev. Za potrebe članka zadošča definicija na ortonormirani bazi, kjer ortonormirano stanje označimo z $|j\rangle$ [4, str. 217]:

$$|j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle . \tag{3}$$

Iz definicije lahko razberemo, da je operacija linearna in unitarna, kar pove, da jo lahko predstavimo kot kvantna vrata. Želimo jih sestaviti iz že poznanih preprostejših kvantnih vrat, a k temu s trenutnim predpisom težko pristopimo. Zato uvedemo alternativni zapis binarnih ulomkov $0.j_l j_{l+1} ... j_n := j_l/2 + j_{l+1}/2^2 + \cdots + j_n/2^{l-n-1}$ in z nekaj algebre, ki je na voljo v dodatku, izpeljemo zvezo ekvivalentno, enačbi (3) v produktni reprezentaciji:

$$|j_1, j_2, \ldots, j_n\rangle \to \frac{\left(|0\rangle + e^{2\pi i 0.j_n} |1\rangle\right) \left(|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle\right) \cdots \left(|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_n} |1\rangle\right)}{2^{n/2}}. \tag{4}$$

Ta oblika olajša konstrukcijo vezja, saj vsako posamično stanje kubita zapišemo v enem oklepaju in zaradi tega vidimo točne transformacije vhodnih kubitov. Zavoljo boljše preglednosti je normalizacijski faktor v preostanku tega poglavja izpuščen. Opazimo, da za potrebne transformacije $|j_l\rangle \to |0\rangle + e^{2\pi i 0.j_l \ldots j_n} |1\rangle$, kjer je $0 < l \leq n$, v fazo dodamo tudi informacije o stanju drugih $n - l$ kubitov. Iz tega lahko razberemo, da za vsak kubit $|j_l\rangle$ uporabimo ena enokubitna vrata, ki podajo informacijo o stanju $|j_l\rangle$, in $n - l$ dvokubitnih vrat, ki podajo informacijo o stanju drugih kubitov $|j_k\rangle$, kjer je $l < k \leq n$.



Shema 5. Vezje kvantne Fourierove transformacije.

Kubit $|j_n\rangle$ moramo torej preobraziti po predpisu $|j_n\rangle \to |0\rangle + e^{2\pi i 0.j_n} |1\rangle$. Za to transformacijo potrebujemo le enokubitna Hadamardova vrata iz sheme 3. Kubit ima namreč na začetku dve možni

stanji $|0\rangle$ ali $|1\rangle$, ki se transformirata v $|0\rangle + |1\rangle$ in $|0\rangle - |1\rangle$. Transformirani stanji lahko zapišemo tudi kot $|0\rangle + e^{2\pi i 0/2} |1\rangle$ ter $|0\rangle + e^{2\pi i 1/2} |1\rangle$. To je natanko to, kar želimo, saj je prej uvedeni zapis $0.j_n$ lahko enak le 0 ali 1/2.

V naslednji transformaciji $|j_{n-1}\rangle \to |0\rangle + e^{2\pi i 0.j_{n-1}j_n} |1\rangle$ na enak način uporabimo Hadamardova vrata in tako preidemo v stanje $|0\rangle + e^{2\pi i 0.j_{n-1}} |1\rangle$. V fazo moramo dodati le še informacijo o kubitu $|j_n\rangle$. To storimo s pomočjo rotacijskih faznih vrat R_k iz enačbe (1). V tem primeru potrebujemo fazna vrata z vrednostjo $k = 2$, saj želimo dodati fazni faktor $e^{2\pi i j_n/4}$. Stanje kubita $|j_n\rangle$ upoštevamo s funkcijo kontrole. Ko velja $|j_{n-1}\rangle = |1\rangle$, želimo namreč fazo dodati, v nasprotnem primeru, ko je $|j_{n-1}\rangle = |0\rangle$, pa tega ne želimo storiti. Iz poglavja o kontrolnih vratih vemo, da natanko to stori kontrolni kubit. Kot kontrolni kubit na vrata R_2 torej povežemo kubit $|j_n\rangle$ in vrata spremenimo v dvokubitna. Za preostale transformacije ponovimo ekvivalenten proces tako, da za vsak kubit $|j_l\rangle$ oziroma „binarno decimalko“ dodamo kontrolna rotacijska fazna vrata s primerno vrednostjo k, kot je razvidno iz sheme 5 [4, str. 218–221].

3.1.2 Ocenjevanje faze

V kontekstu članka je ocenjevanje faze le korak algoritma za iskanje reda, a je to v resnici samostojen problem, s katerim poiščemo lastno vrednost znanega operatorja. Pogosto se pojavlja tudi v drugih algoritmih, kot so iskanje periode in diskretno logaritmiranje.

Recimo, da imamo podan splošen unitarni operator U z lastnim vektorjem $|u\rangle$. Iz unitarnosti vemo, da velja $U|u\rangle = e^{2\pi i\varphi}|u\rangle$, kjer se zadošča omejiti na φ med nič in ena. Problem, ki ga rešujemo z ocenjevanjem faze, je iskanje vrednosti neznanke φ.

K razlagi algoritma ocenjevanja faze pristopimo s pomočjo produktne oblike kvantne Fourierove transformacije na enačbi (4). Namreč, če $U|u\rangle$ preobrazimo v tako obliko, lahko uporabimo inverzno kvantno Fourierovo transformacijo, da dobimo stanje, ki opisuje približek neznanke φ.

Prvi korak v tem procesu je, da vzamemo t kubitov v stanju $|0\rangle$ in na njih uporabimo Hadamardova vrata iz sheme 3. Tako je vseh t kubitov v stanju $(|0\rangle + |1\rangle)/\sqrt{2}$. Z določitvijo števila kubitov t tudi določimo natančnost približka φ. S tem namreč omejimo število „binarnih decimalk“, ki jih dobimo iz inverzne kvantne Fourierove transformacije. Od tod naprej ponovno izpuščamo normalizacijske faktorje.

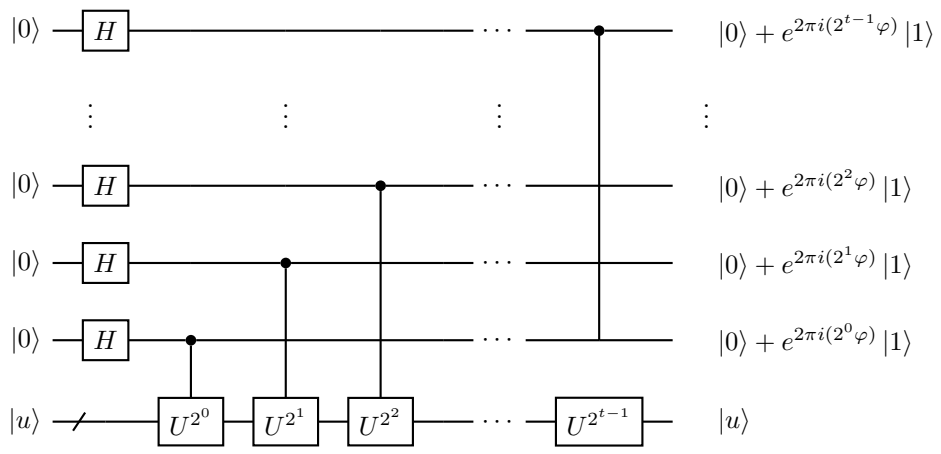
Stanjem moramo dodati še fazo s pravimi „binarnimi decimalkami“ φ. Spomnimo se uvedenega zapisa $\varphi = \varphi_1/2 + \varphi_2/2^2 + \cdots + \varphi_n/2^n = 0.\varphi_1\varphi_2\ldots\varphi_n$. Opazimo, da „binarne decimalke“ pomikamo v levo z množenjem z dve, oziroma s kvadriranjem, ko je φ v eksponentu $e^{2\pi i\varphi}$. Prva „binarna decimalka“ se tako pomakne med cela števila in se o njej izgubi informacija, saj je $e^{2\pi ik}$ za vsako celo število k enak 1. Torej, če na stanje $|u\rangle$ delujemo z U 2^j-krat, dobimo $U^{2^j}|u\rangle = e^{2\pi i 2^j\varphi}|u\rangle$. To lahko iz prejšnjih ugotovitev ekvivalentno zapišemo z $e^{2\pi i 0.\varphi_{j+1}\varphi_{j+2}\ldots\varphi_n}|u\rangle$.

Naslednji korak je prenos faze v kubite, na katere smo delovali s Hadamardovimi vrati. Podobno kot pri konstrukciji vezja kvantne Fourierove transformacije to dosežemo tako, da s kontrolo povežemo kubit in operator U^{2^j}. Kontrolna vrata se aktivirajo le za stanje $|1\rangle$. Kubiti pa so trenutno v stanju $|0\rangle + |1\rangle$. Sledi končno stanje kubita $|0\rangle + e^{2\pi i 2^j\varphi}|1\rangle$. Vse, kar preostane, je, da to storimo za vsak kubit s primerno potenco operatorja, kar je razvidno iz sheme 6 [4, str. 221–223]. Uspeli smo pripraviti stanje, iz katerega lahko z inverzno kvantno Fourierovo transformacijo pridobimo željeni približek za φ. Poznamo vezje kvantne Fourierove transformacije iz sheme 5. S hermitskim konjugiranjem vezje preobrazimo v vezje inverzne kvantne Fourierove transformacije. Nato ga uporabimo na trenutnem stanju t kubitov:

$$\frac{\left(|0\rangle + e^{2\pi i 0.\varphi_t}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.\varphi_{t-1}\varphi_t}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.\varphi_1\varphi_2\ldots\varphi_t}|1\rangle\right)}{2^{t/2}} \tag{5}$$

in dobimo stanje približka φ:

$$|\varphi_1\varphi_2...\varphi_t\rangle\,.$$



Shema 6. Del vezja ocenjevanja faze, ki pripravi t kubitov na inverzno kvantno Fourierovo transformacijo.

Preden se vrnemo na reševanje problema iskanja reda, zapišimo še matematično zvezo za transformacijo v stanje (5). Po uporabi Hadamardovih vrat na vseh t kubitih z začetnim stanjem $|0\rangle$, lahko zapišemo novo stanje kot $2^{-t/2}\sum_{j=0}^{2^t-1}|j\rangle$, pri čemer j predstavlja vrednosti, ki pokrivajo vse možne binarne kombinacije t kubitov. Naslednji korak transformacije je delovati z vsemi kontrolnimi operatorji U^{2^k} na lastno stanje $|u\rangle$. Funkcijo kontrole v kvantnem vezju lahko opišemo s potenciranjem operatorja na vrednost stanja posameznega kubita:

$$\begin{aligned}|j\rangle\,|u\rangle \to |j\rangle\,((U^{2^0})^{j_0}(U^{2^1})^{j_1}\dots(U^{2^{n-1}})^{j_{n-1}})\,|u\rangle = \\ = |j\rangle\,U^{j_0 2^0+j_1+\dots+j_{n-1}2^{n-1}} = |j\rangle\,U^j\,|u\rangle\,.\end{aligned}$$

V tej enačbi j_i predstavljajo binarne komponente števila j in hkrati vrednosti posameznih stanj kubitov. S takim pristopom opazimo, da potenca U ustreza vrednosti j. S to ugotovitvijo pridemo do željene oblike zapisa transformacije t kubitov v stanju $|0\rangle$ in lastnega stanja $|u\rangle$ operatorja U v stanje (5):

$$|0\rangle\,|0\rangle\dots|0\rangle\,|u\rangle \to \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle\,U^j\,|u\rangle\,. \tag{6}$$

3.2 Iskanje reda

Iskanje faze smo obravnavali podrobneje, saj smo želeli poiskati lastno vrednost $U\,|u\rangle = e^{i\varphi}\,|u\rangle$, v katero smo zakodirali red z operatorjem $U = |xy \pmod N\rangle$. Algoritem za ocenjevanje faze to omogoča, če izpolnimo dva pogoja:

- Učinkovito implementiramo operator U^{2^j} za poljubni j.
- Pripravimo lastno stanje $|u_s\rangle$ iz enačbe (2).

Prvi pogoj izpolnimo z uporabo algoritma hitrega modularnega potenciranja [4, str. 228], na katerega se v tem članku ne osredotočamo. Za izpolnitev drugega pogoja opazimo lastnost lastnih vektorjev

operatorja U:

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle \,. \tag{7}$$

V tej enačbi opazimo vsoto geometrijske vrste, iz česar sledi, da je vsota po s vseh členov lastnega stanja $|u_s\rangle$, razen tistih z vrednostjo $k = 0$ (k definiran v enačbi (2)), enaka nič. Tako v vsoti ostane le r členov s stanjem $|1\rangle$, kar se s koeficienti izvrednoti v stanje $|1\rangle$.

Ta lastnost omogoča, da v algoritmu ocenjevanja faze nadomestimo lastno stanje z enostavnejšim stanjem $|1\rangle$. Vendar se pri meritvi rezultata zaplete, saj je vrednost faze s/r odvisna od lastnega stanja $|u_s\rangle$. Namesto enega lastnega stanja $|u_s\rangle$ pa smo uporabili stanje $|1\rangle$, ki je po enačbi (7) superpozicija vseh lastnih stanj $|u_s\rangle$. Težavo rešimo tako, da opravimo meritev na stanju $|1\rangle$, ki nam vrne eno izmed lastnih stanj $|u_s\rangle$, kar določi vrednost s z enakomerno verjetnostjo $1/r$ iz množice $\{0, 1, \ldots, r-1\}$. Tako je vrednost s/r določena na t „binarnih decimalk“ [4, str. 226–228].

Iz števila s/r potem izluščimo vrednost reda r z algoritmom verižnih ulomkov. V kontekstu Shorovega algoritma algoritem verižnih ulomkov služi za pretvorbo racionalnega števila v obliko ulomka. Prvi korak algoritma je, da neko število a zapišemo v naslednji obliki:

$$a = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots + \frac{1}{a_M}}}} := [a_0, \ldots, a_M]. \tag{8}$$

S pomočjo tega zapisa definiramo m-ti konvergent kot $[a_0, ..., a_m]$ za $0 \leq m \leq M$, ki je m-ti približek ulomka števila. Drugi konvergent, torej $m = 2$, lahko zapišemo kot [4, str. 230]:

$$[a_0, a_1, a_2] := a_0 + \frac{1}{a_1 + \frac{1}{a_2}} = a_0 + \frac{a_2}{a_1 a_2 + 1} = \frac{a_0 a_1 a_2 + a_0 + a_2}{a_1 a_2 + 1}.$$

Za boljše razumevanje izvedimo algoritem na številu 2,8125. Najprej ga prevedemo na obliko iz enačbe (8):

$$2{,}8125 = 2 + 0{,}8125 = 2 + \frac{1}{1 + 0{,}230769\ldots} = 2 + \frac{1}{1 + \frac{1}{4 + 0{,}333333\ldots}} = 2 + \frac{1}{1 + \frac{1}{4 + \frac{1}{3}}}.$$

Alternativna oblika števila 2,8125 je torej $[2, 1, 4, 3]$. Ta postopek vrne tri približke ulomka vrednosti 2,8125: $\frac{2}{1}$, $\frac{3}{1}$, $\frac{14}{5}$, na koncu pa še ulomek s točno vrednostjo $\frac{45}{16}$.

Naravno vprašanje, ki sledi, je, kaj se zgodi v primeru, ko imata števili s in r skupne faktorje. Algoritem verižnih ulomkov takrat najde r', ki je faktor reda r. Da r' ni red, hitro preverimo preko definicije reda $x^r \equiv 1 \pmod{N}$. Nastali problem rešimo tako, da algoritem iskanja reda še enkrat poženemo na številu $x^{r'}$. Opazimo namreč, da je red števila $x^{r'}$ po modulu N enak r/r', saj velja $(x^{r'})^{r/r'} = x^r \equiv 1 \pmod{N}$ [4, str. 231].

Povzemimo vse korake algoritma za iskanje reda [4, str. 232]:

1. **Inicializacija:**
   Pripravimo sistem s t kubiti v osnovnem stanju $|0\rangle$ in še eno dodatno stanje $|1\rangle$.

2. **Ocenjevanje faze:**

   - Na t kubitov v začetnem stanju $|0\rangle$ najprej delujemo s Hadamardovimi vrati. Nato še s kontrolnimi vrati U^{2^j}, pri čemer dodatni kubit $|1\rangle$ predstavlja superpozicijo lastnih stanj operatorja U, kot v enačbi (7). Tako dobimo superpozicijo stanj, katerih splošna oblika je v enačbi (5). Točen postopek je prikazan na shemi 6.

- Na trenutnem stanju uporabimo inverzno kvantno Fourierovo transformacijo in dobimo stanje $\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |s/r\rangle \, |u_s\rangle$.

3. **Meritev:**
   Izvedemo meritev in pridobimo s/r, z naključno vrednostjo $0 \leq s < r$.

4. **Algoritem verižnih ulomkov:**
   S pomočjo algoritma verižnih ulomkov izračunamo rezultat, red r.

3.3 Faktorizacija

Shorov algoritem poenostavimo z izrekom, ki pravi, da če za sestavljeno število N poiščemo celo število x, ki je rešitev enačbe $x^2 \equiv 1 \pmod{N}$ v intervalu $1 \leq x \leq N$ in ne velja $x \equiv 1 \pmod{N}$, niti $x \equiv N-1 \equiv -1 \pmod{N}$, potem lahko uporabimo koncept največjega skupnega delitelja (gcd), saj je vsaj eden izmed $\gcd(x-1, N)$ in $\gcd(x+1, N)$ netrivialni faktor števila N [4, str. 233].

Iz ideje, ki sledi iz izreka, iščemo število x v intervalu $1 \leq x < N$, ki ima sod red r po modulu N. S to zahtevo lahko zapišemo definicijo reda v obliki, ki zadošča izreku $(x^{r/2})^2 \equiv 1 \pmod{N}$. Preveriti moramo le še, da velja $x^{r/2} \not\equiv -1 \pmod{N}$. Neenakost $x^{r/2} \not\equiv 1 \pmod{N}$ je namreč že izpolnjena, saj bi v nasprotnem primeru red bil delitelj $r/2$, kar r ni. V primeru, ko so vsi pogoji izpolnjeni, je tako vsaj eden izmed $\gcd(x^{r/2}-1, N)$ in $\gcd(x^{r/2}+1, N)$ faktor števila N [4, str. 233].

S točnimi koraki algoritma določimo še vse posebne primere, ki jih je hitreje izračunati klasično [4, str. 233–234]:

Klasični del algoritma:

1. **Sodost:**
   Če je N sod, takoj vrnemo 2 kot enega izmed faktorjev.

2. **Preverjanje popolne potence:**
   Preverimo, ali obstajata celi števili $a \geq 1$ in $b \geq 2$, za kateri velja $N = a^b$. Če obstajata, vrnemo a kot faktor.

Kvantni del algoritma:

3. **Izbira števila za iskanje reda:**
   Naključno izberemo število x v intervalu $1 \leq x \leq N$. Izračunamo $\gcd(x, n)$; če je ta večji od 1, smo že našli netrivialni faktor.

4. **Iskanje reda:**
   Uporabimo kvantni algoritem za iskanje reda r števila x po modulu N.

5. **Preverjanje pogojev in izračun faktorja:**
   Če je $r/2$ sod in $x^{r/2} \not\equiv -1 \pmod{N}$, izračunamo $\gcd(x^{r/2}-1, N)$ in $\gcd(x^{r/2}+1, N)$. Nato preverimo, da je vsaj eden izmed njiju faktor števila N, ki ga v tem primeru vrnemo. V primeru, ko pogoji niso izpolnjeni, algoritem ne najde netrivialnega faktorja.

V zadnjem koraku omenimo tudi možnost neuspeha algoritma. Eden izmed možnih razlogov je, da algoritmu kot vhod podamo praštevilo N. V tem primeru algoritmu ne bo uspelo najti nobenega faktorja, ker ti ne obstajajo. Druga možnost je, da z naključno izbiro izberemo x, za katerega red po modulu N ne obstaja ali pa je ta lih. V takem primeru moramo algoritem ponoviti z drugo vrednostjo x. Točka neuspeha lahko izvira tudi iz premajhne natančnosti ocene faze, kar algoritmu verižnih ulomkov onemogoči najdbo prave vrednosti reda.

3.3.1 Faktorizacija števila 15

Za primer delovanja Shorovega algoritma smo izbrali število 15. Vsa manjša števila namreč učinkovito faktoriziramo brez uporabe algoritma za iskanje reda. Predpostavimo, da izberemo število $x = 7$. Ta izbira očitno ni faktor števila 15, zato nadaljujemo z iskanjem reda r števila 7 po modulu 15. S to izbiro smo definirali operator $U\,|y\rangle = |7y \bmod 15\rangle$.

Naslednji korak je algoritem ocenjevanja faze. Za naš primer vzamemo $t = 3$ kubite v stanju $|0\rangle$ in stanje $|1\rangle$. Uporabimo transformacijo iz enačbe (6) in jo razpišimo za naš primer:

$$\begin{aligned}\frac{1}{\sqrt{2^t}}\sum_{k=0}^{2^t-1}|k\rangle\,U^k\,|1\rangle &= \frac{1}{\sqrt{2^3}}\sum_{k=0}^{7}|k\rangle\left|7^k \bmod 15\right\rangle = \\ &= \frac{1}{\sqrt{2^3}}\big[(|000\rangle + |100\rangle)\,|1\rangle + (|001\rangle + |011\rangle)\,|7\rangle + \\ &\quad + (|010\rangle + |110\rangle)\,|4\rangle + (|011\rangle + |100\rangle)\,|13\rangle\big].\end{aligned}$$

Nato na t kubitih, ki so začeli v stanju $|0\rangle$, uporabimo inverzno Fourierovo transformacijo. Na točen postopek pretvorbe se tu ne bomo osredotočali, a preko nje dobimo stanje

$$\begin{aligned}\frac{1}{2}\Big[\frac{1}{2}\,|000\rangle\,\big(\,|1\rangle + |7\rangle + |4\rangle + |13\rangle\,\big) + \frac{1}{2}\,|010\rangle\,\big(\,|1\rangle - i\,|7\rangle - |4\rangle + i\,|13\rangle\,\big) + \\ + \frac{1}{2}\,|100\rangle\,\big(\,|1\rangle - |7\rangle + |4\rangle - |13\rangle\,\big) + \frac{1}{2}\,|110\rangle\,\big(\,|1\rangle + i\,|7\rangle - |4\rangle - i\,|13\rangle\,\big)\Big].\end{aligned} \tag{9}$$

Na tej točki bi v dejanski izvedbi algoritma izvedli meritev na stanju $|1\rangle$ in z meritvijo bi bil določen naključen lastni vektor $|u_s\rangle$ operatorja U in s tem vrednost s. Mi pa v enačbi (9) prepoznamo lastne vektorje znotraj oklepajev in zapišemo v lepšo obliko:

$$\frac{1}{\sqrt{4}}(|000\rangle\,|u_0\rangle + |010\rangle\,|u_1\rangle + |100\rangle\,|u_2\rangle + |110\rangle\,|u_3\rangle).$$

Poljubno izberimo stanje za $s = 1$, torej $s/r = 0.\varphi_1\varphi_2\varphi_2 = 0.010 = 0.25 = 0 + 1/4$. Torej je $r = 4$ res red in je sod, zato preverimo še pogoj $x^{r/2} \not\equiv -1 \pmod N$. Izračunamo še $\gcd(7^{4/2} - 1, 15) = 3$ in $\gcd(7^{4/2} + 1, 15) = 5$. Oba največja skupna delitelja sta tudi faktorja in s tem je algoritem zaključen [6].

4. Praktične omejitve

Čeprav je teorija Shorovega algoritma dobro razvita, je praktična izvedba še vedno precej oddaljena. Edini števili, ki ju trenutno lahko konsistentno faktoriziramo z uporabno kvantnega algoritma, sta 15 in 21, pri čemer je napredek v smeri faktoriziranja večjih števil počasen. Število 15 je bilo prvič faktorizirano že leta 2001, 21 pa šele leta 2012. V zadnjih letih so bili tudi uspešni primeri faktoriziranja večjih števil, a ti niso zanesljivi [7]. Glavni razlog za to omejitev je v visokem številu za izvedbo potrebnih kubitov. Čeprav trenutna največja implementacija razpolaga s 1180 kubiti, so ti še vedno močno izpostavljeni šumu, kar povzroča visok delež napak [8]. Zaradi teh težav šifriranje RSA, ki smo ga v uvodu predstavili kot motivacijo za faktoriziranje števil, še ni ogroženo, saj trenutni kvantni računalniki niso sposobni kompleksnosti, potrebne za razbitje kriptografskih ključev.

5. Zaključek

Shorov algoritem dokazuje, da kvantni računalniki lahko rešujejo določene probleme bistveno hitreje kot klasični algoritmi. V tem članku smo obravnavali njegovo poenostavitev na iskanje reda, ki

ga je mogoče poiskati s pomočjo ocene faze in kvantne Fourierove transformacije. Kljub nekaterim praktičnim omejitvam ostaja Shorov algoritem pomemben mejnik v razvoju kvantnega računalništva ter ponuja vpogled v potencial kvantnih tehnologij za reševanje problemov, ki so trenutno nerešljivi.

6. Dodatek

6.1 Izpeljava produktne reprezentacije kvantne Fourierove transformacije

Izpeljavo produktne reprezentacije kvantne Fourierove transformacije začnemo z definicijo iz poglavja o kvantni Fourierovi transformaciji:

$$|j\rangle \to \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i j k/2^n} |k\rangle .$$

Prvi korak je, da vsoto iz definicije nadomestimo z n vsotami, kjer vsaka obsega vse možne vrednosti posameznega kubita. V eksponentu pa število k zapišemo z vsoto, ki je ekvivalentna zapisu števila k z „binarnimi decimalkami“. Ta zapis omogoči pretvorbo v zapis s tenzorskim produktom

$$\frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi i j (\sum_{l=1}^{n} k_l 2^{-l})} |k_1 \ldots k_n\rangle = \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} e^{2\pi i j k_l 2^{-l}} |k_l\rangle .$$

Za izvedbo naslednjega koraka potrebujemo multilinearnost tenzorskega produkta, ki jo za $n = 2$ zapišemo kot:

$$(a+b) \otimes (c+d) = a \otimes c + a \otimes d + b \otimes c + b \otimes d.$$

S to lastnostjo preobrazimo enačbo (10) v

$$\frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \left(\sum_{k_l=0}^{1} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \right) .$$

Iz slednje nato razpišemo vsoto in tenzorski produkt v končno stanje

$$\frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \left(|0\rangle + e^{2\pi i j 2^{-l}} |1\rangle \right) =$$
$$= \frac{\left(|0\rangle + e^{2\pi i 0.j_n} |1\rangle\right) \left(|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle\right) \ldots \left(|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_n} |1\rangle\right)}{2^{n/2}}.$$

Tako smo izpeljali obliko kvantne Fourierove transformacije, ki jo uporabimo za konstrukcijo vezja in za razlago postopka ocenjevanja faze [4, str. 218].

LITERATURA

[1] Wikipedia contributors, *Shor's algorithm - Wikipedia*, https://en.wikipedia.org/wiki/Shor%27s_algorithm, dostopano: 25.03.2025.

[2] Wikipedia contributors, *RSA(cryptosystem) - Wikipedia*, https://en.wikipedia.org/wiki/RSA_(cryptosystem), dostopano: 25.03.2025.

[3] Smite-Meister, *Bloch sphere, a geometrical representation of a two-level quantum system - Wikimedia*, https://commons.wikimedia.org/wiki/File:Bloch_sphere.svg, dostopano: 15.06.2025.

[4] M. A. Nielsen in I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press (2010).

[5] A. P. Young, *Quantum Gates - University of California, Santa Cruz*, https://young.physics.ucsc.edu/150/gates.pdf, dostopano: 25.03.2025.

[6] Elucyda, *Quantum Order-Finding Example N=15 Part 3: Quantum algorithm for a=7 - Youtube*, https://www.youtube.com/watch?v=vFsEe_XbR3o, dostopano: 25.03.2025.

[7] Wikipedia contributors, *Integer Factorization Records - Wikipedia*, https://en.wikipedia.org/wiki/Integer_factorization_records, dostopano: 25.03.2025.

[8] Wikipedia contributors, *List of Quantum Processors - Wikipedia*, https://en.wikipedia.org/wiki/List_of_quantum_processors, dostopano: 25.03.2025.