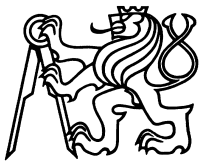
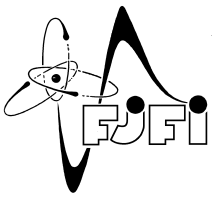


České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská



Kvantové počítání na současných kvantových počítačích

Quantum computation on contemporary quantum computers

Bakalářská práce

Autor: **Ludvík Cigna**

Vedoucí práce: **doc. Ing. Martin Štefaňák, Ph.D.**

Akademický rok: 2020/2021

- Zadání práce -

- Zadání práce (zadní strana) -

Poděkování:
Chtěl bych zde poděkovat především svému školiteli doc. Ing. Martinu Štefaňákovi, Ph.D. za pečlivost, ochotu, vstřícnost a odborné i lidské zázemí při vedení mé bakalářské práce.

Prohlášení:
Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd...) uvedené v přiloženém seznamu.

Nemám závažný důvod proti použití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 5. srpna 2021 podpis

Název práce:

Kvantové počítání na současných kvantových počítačích

Autor: Ludvík Cigna

Obor: Matematické inženýrství

Zaměření: Matematická fyzika

Druh práce: Bakalářská práce

Vedoucí práce: doc. Ing. Martin Štefaňák, Ph.D. Katedra fyziky, Fakulta jaderná a fyzikálně inženýrská, České vysoké učení technické v Praze

Abstrakt: Tato práce se zabývá problematikou kvantového počítání. To má potenciál překonat hranice efektivní řešitelnosti problémů stanovené klasickým pojetím informatiky. Začátek je věnován shrnutí základních konceptů kvantové informatiky, jako je qubit či kvantové brány. Dále se již pozornost obrací k jednotlivým nejdůležitějším algoritmům. Z těch vyhledávacích se jedná o Groverův algoritmus, u kterého se ukáže spojitost s kvantovou procházkou. Poté je čtenáři představena kvantová Fourierova transformace a z ní vycházející dobře známý Shorův algoritmus. V závěru jsou shrnuty možnosti a omezení současných NISQ (Noisy Intermediate-Scale Quantum) počítačů a možné budoucí aplikace.

Klíčová slova: Groverův algoritmus, kvantová brána, kvantové počítání, qubit, Shorův algoritmus

Title:

Quantum computation on contemporary quantum computers

Author: Ludvík Cigna

Abstract: This work deals with the issue of quantum computing. Quantum computing has the potential to overcome the limits of effective solvability of problems set by the classical approach to informatics. The beginning is devoted to a summary of basic concepts of quantum informatics, such as qubit or quantum gates. Then attention is paid to the most important algorithms. From the search algorithms, we will talk about the Grover's algorithm, in which we show the connection with the quantum walk. Then, the reader is introduced to the quantum Fourier transform and the well-known Shor's algorithm based on it. Finally, the possibilities and limitations of current NISQ (Noisy Intermediate-Scale Quantum) computers and possible future applications are summarized.

Key words: Grover's algorithm, quantum computing, quantum gate, qubit, Shor's algorithm

Obsah

Úvod

Ačkoli za největšího průkopníka samotného konceptu kvantového počítání bývá nejčastěji označován Richard Feynman, první úvahy na toto téma se začaly objevovat již v sedmdesátých letech minulého století. Za zásadní bývá v tomto směru považován článek Paula Benioffa z roku 1980, ve kterém dává do souvislosti proces výpočtu na Turingově stroji s vývojem stavu vzhledem k Hamiltoniánu [7]. Od té doby prošlo kvantové počítání značným vývojem především po teoretické stránce. Za milník, díky kterému se pojem *kvantový počítač* dostal do povědomí i širší veřejnosti, lze prohlásit objev Shorova algoritmu [26]. Ten ostatně i ve mně samotném probudil prvotní nadšení pro tuto velice zajímavou a perspektivní oblast kvantové informatiky.

V této práci shrnu základní pojmy a klíčové koncepty kvantového počítání a detailně představím tři zásadní kvantové algoritmy, na kterých lze ukázat možnosti kvantových počítačů sahající za možnosti těch klasických. Konkrétně se bude jednat o Deutsch-Jozsův algoritmus, Groverův vyhledávací algoritmus a Shorův algoritmus. Zde bude snaha klást důraz nejen na matematicky korektní odvození, ale také na vybudování intuice za daným problémem. Zároveň všude, kde to bude dávat smysl, bude teoretický výklad doplněn s pomocí veřejně dostupné sady pro vývoj softwaru určeného ke kvantovým výpočtům Qiskit o ukázky samotné implementace daných algoritmů. V případě, že budou obvody dostatečně jednoduché, využijeme k výpočtu reálná kvantová zařízení za pomoci platformy IBM Quantum. K většině ukázek však bude potřeba využít simulátor kvantového výpočtu. Nakonec bude diskutováno směřování kvantového počítání, jeho potenciální aplikace a na závěr nedávný úspěch dosažení kvantové nadřazenosti.

Cílem práce tedy bude vybudovat široké základy umožňující další zkoumání v této oblasti. Toho by mělo být dosaženo pomocí detailního popisu vybraných algoritmů a jevů v nich probíhajících. V rámci tohoto popisu bude vyvinuta snaha o poskytnutí různých pohledů na daný problém. Příkladem může být srovnání vyhledávání pomocí Groverova algoritmu a pomocí kvantové procházky na úplném grafu resp. na grafu typu hvězda. S takto vybudovanými základy může být další pozornost upřena na celou řadu dalších již konkrétnějších témat.

Kapitola 1

Základní koncepty

V této kapitole se seznámíme s pojmy a úvahami, které tvoří samotný základ kvantové informatiky. Hlavní oporou při souhrnu těchto konceptů nám bude [21], přesto však k jednotlivým úvahám budou mnohdy doplněny konkrétnější zdroje.

1.1 Qubit

Základním konceptem v kvantové informatice je qubit, který představuje analogii k bitu v klasické informatice. Jedná se tedy o reprezentaci informace. V rámci této práce budeme v drtivé většině případů s qubitem nakládat jako s matematickým objektem nezávislým na fyzikální realizaci. Stejně, jako se bit může nacházet ve stavu 0, nebo 1, tak i qubit má dva stavy - $|0\rangle$ a $|1\rangle$. Klíčovým rozdílem ovšem je, že qubit se na rozdíl od bitu může nacházet v lineární kombinaci těchto stavů, také nazývané jako superpozici. Ta má následující tvar:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{1.1}$$

Jak je tedy vidět, qubit můžeme považovat za vektor ve dvourozměrném Hilbertově prostoru (značíme $\mathcal{H}$) , kde kety $|0\rangle$ a $|1\rangle$ tvoří ortonormální výpočetní bázi. Obecně komplexní koeficienty α, β pak představují amplitudu pravděpodobnosti naměření $|0\rangle$ respektive $|1\rangle$ s pravděpodobností $|\alpha|^2$ respektive $|\beta|^2$. Platí tedy $|\alpha|^2 + |\beta|^2 = 1$. Toto lze geometricky interpretovat jako podmínku normalizace vektoru k 1.

S touto podmínkou lze rovnici (1.1) přepsat na:

$$|\psi\rangle = \exp(i\gamma)(\cos(\theta/2)\,|0\rangle + \exp(i\varphi)\sin(\theta/2)\,|1\rangle) \tag{1.2}$$

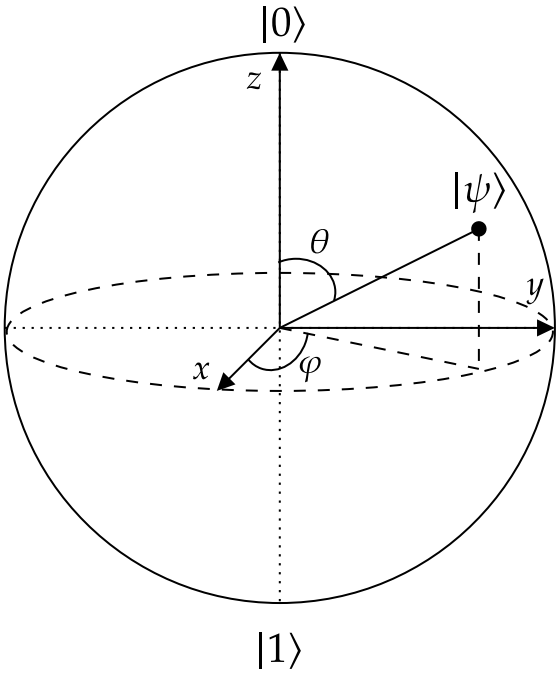
a vzhledem k tomu, že první exponenciála představuje tzv. globální fázi, která nemá žádný pozorovatelný význam, tak dostaneme tvar:

$$|\psi\rangle = \cos(\theta/2)\,|0\rangle + \exp(i\varphi)\sin(\theta/2)\,|1\rangle\,. \tag{1.3}$$

Takovéto vyjádření dobře slouží k vizualizaci qubitu na jednotkové sféře nazývané Blochova sféra znázorněné na obrázku (1.1).

Z postulátu kvantové mechaniky plyne, že po měření se qubit nachází buď ve stavu $|0\rangle$, nebo $|1\rangle$. To znamená, že měřením se zničí stav superpozice.

Kromě qubitu, jehož výpočetní bázi tvoří dva stavy, se můžeme ještě setkat s označením qutrir resp. qunit, jehož bázi tvoří tři resp. n stavů.



Obrázek 1.1: Blochova sféra

1.2 Kvantový registr

Síla kvantového počítání začne být zřejmá ve chvíli, když podíváme na systém více qubitů. Uvažujme nejprve systém n klasických bitů. Ten se bude nacházet v jednom z 2^n možných stavů. Na druhou stranu pro systém n qubitů bude Hilbertův prostor, ve kterém budou data ukládána, tenzorový součin Hilbertových prostorů jednotlivých qubitů, tj.

$$\mathcal{H} = \mathcal{H}_{n-1} \otimes \mathcal{H}_{n-2} \otimes ... \otimes \mathcal{H}_0 \tag{1.4}$$

s prvky

$$|\psi\rangle = |\psi\rangle_{n-1} \otimes |\psi\rangle_{n-2} \otimes ... \otimes |\psi\rangle_0 \,, \tag{1.5}$$

viz [17]. Celkový stav tohoto systému pak bude popsán 2^n amplitudami a do okamžiku měření bude držet informace o všech možných hodnotách výsledku měření najednou. Podívejme se na příkladu dvou qubitů, jak se tvoří a vypadají bazické stavy tohoto systému:

$$|00\rangle \equiv |0\rangle \otimes |0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}1 \cdot \begin{pmatrix}1\\0\end{pmatrix}\\ 0 \cdot \begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \qquad |01\rangle \equiv |0\rangle \otimes |1\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}1 \cdot \begin{pmatrix}0\\1\end{pmatrix}\\ 0 \cdot \begin{pmatrix}0\\1\end{pmatrix}\end{pmatrix} = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

$$|10\rangle \equiv |1\rangle \otimes |0\rangle = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0 \cdot \begin{pmatrix}1\\0\end{pmatrix}\\ 1 \cdot \begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix} = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \qquad |11\rangle \equiv |1\rangle \otimes |1\rangle = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0 \cdot \begin{pmatrix}0\\1\end{pmatrix}\\ 1 \cdot \begin{pmatrix}0\\1\end{pmatrix}\end{pmatrix} = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}.$$

Otázkou, zda je možné takovýmto způsobem získat všechny stavy z výsledného Hilbertova prostoru, se budeme zabývat v další sekci.

Již zde ovšem vidíme, že pro zápis stavu systému qubitů lze užít dvou různých zápisů. Mějme $x \in \{0, ..., 2^n - 1\}$ pro nějaké pevně dané $n \in \mathbb{N}_0$ a označme $N \equiv 2^n$. Binární rozvoj takového x je pak dán jako $x = x_1 2^{n-1} + x_2 2^{n-2} + ... + x_n 2^0$, kde $x_i \in \{0, 1\}$, $i \in \{1, ..., n\}$. První způsob, jak zapisovat stavy, je pomocí tohoto binárního rozvoje, tj. $(x+1)$. bazický stav zapíšeme jako $|x_1 x_2 ... x_n\rangle$. Druhý způsob zápisu je, že do ketu přímo vložíme, kolikátý bazický stav myslíme, tj. pro x zapíšeme $(x + 1)$. bazický stav pouze jako $|x\rangle$ a máme tím na mysli

$$|x\rangle = \begin{pmatrix} 0 & \dots & 0 & \underbrace{1}_{(x+1).\text{ pozice}} & 0 & \dots & 0 \end{pmatrix}^T . \tag{1.6}$$

Takto může identifikovat $|x\rangle \equiv |x_1 x_2 ... x_n\rangle$. Druhý způsob využijeme především při zápisu superpozice mnoha stavů. Jako příklad si ukažme, jak by se těmito dvěma způsoby zapsala rovnovážná superpozice všech bazických stavů:

$$\frac{1}{\sqrt{N}} \sum_{x_1=0}^{1} \sum_{x_2=0}^{1} ... \sum_{x_n=0}^{1} |x_1 x_2 ... x_n\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle . \tag{1.7}$$

Z kontextu by mělo být vždy jasné, který typ zápisu využíváme.

1.2.1 EPR páry a Bellova nerovnost

Vezměme nyní zcela obecně stavy

$$\left|\psi^{(1)}\right\rangle = \sum_{i=1}^{d_1} a_i \left|\psi_i^{(1)}\right\rangle \in \mathcal{H}^{(1)}$$

a

$$\left|\varphi^{(2)}\right\rangle = \sum_{j=1}^{d_2} b_j \left|\varphi_j^{(2)}\right\rangle \in \mathcal{H}^{(2)},$$

kde $\{\left|\psi_i^{(1)}\right\rangle\}$ je ON báze $\mathcal{H}^{(1)}$, $dim\mathcal{H}^{(1)} = d_1 < +\infty$ a $\{\left|\varphi_j^{(2)}\right\rangle\}$ je ON báze $\mathcal{H}^{(2)}$, $dim\mathcal{H}^{(2)} = d_2 < +\infty$. Máme tedy $d_1 + d_2$ koeficientů a tenzorový součin má tvar:

$$\left|\psi^{(1)}\right\rangle \otimes \left|\varphi^{(2)}\right\rangle = \sum_{i,j} a_i b_j \left|\psi_i^{(1)}\right\rangle \otimes \left|\varphi_j^{(2)}\right\rangle \in \mathcal{H} = \mathcal{H}^{(1)} \otimes \mathcal{H}^{(2)},$$

kde $\{\left|\psi_i^{(1)}\right\rangle \otimes \left|\varphi_j^{(2)}\right\rangle\}$ je ON báze $\mathcal{H}$. Takovéto stavy nazýváme separované. Vezměme nyní obecný stav $|\Psi\rangle$ z prostoru $\mathcal{H}$:

$$|\Psi\rangle = \sum_{i,j} c_{ij} \left|\psi_i^{(1)}\right\rangle \otimes \left|\varphi_j^{(2)}\right\rangle .$$

Je vidět, že počet koeficientů c_{ij} je $d_1 \cdot d_2$. Existují tedy i stavy, které nejsou separované. Takovým stavům říkáme provázané a implikují korelaci výsledků měření (viz [17]).

Podívejme se nyní na příklad dvou qubitů. Nechť první je

$$|\psi\rangle_1 = \alpha |0\rangle + \beta |1\rangle \in \mathcal{H}_1$$

a druhý

$$|\psi\rangle_2 = \gamma |0\rangle + \delta |1\rangle \in \mathcal{H}_2.$$

Pak tenzorovým součinem získáme stav

$$|\psi\rangle_1 \otimes |\psi\rangle_2 = \alpha\gamma\,|00\rangle + \alpha\delta\,|01\rangle + \beta\gamma\,|10\rangle + \beta\delta\,|11\rangle\,.$$

Uvažujme nyní stav

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{1.8}$$

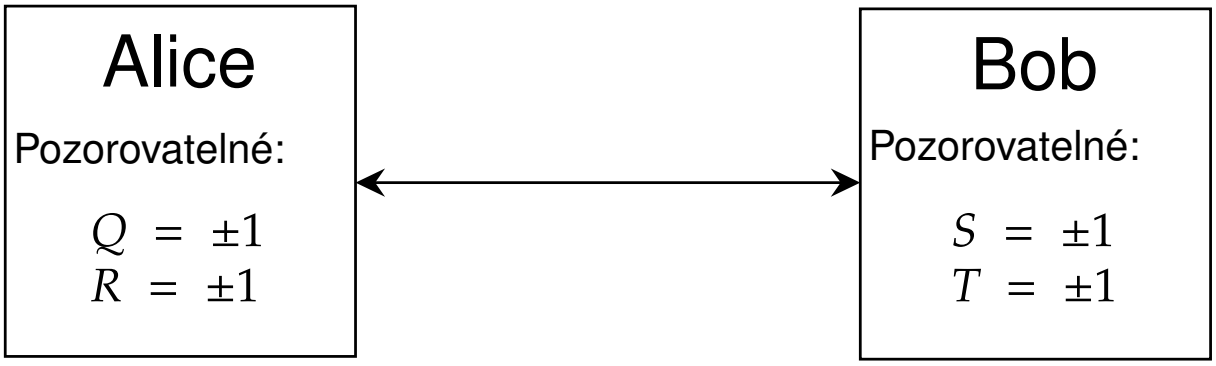
nazývaný Bellův stav nebo EPR (Einstein-Podolsky-Rosen) pár. Je snadné nahlédnout, že tento stav nemůže vzniknout tenzorovým součinem dvou samostatných qubitů. Potřebujeme totiž, aby $\alpha\delta = 0$ a zároveň $\alpha\gamma = \frac{1}{\sqrt{2}}$. Potom ovšem musí $\delta = 0$, což je spor s tím, že $\beta\delta = \frac{1}{\sqrt{2}}$. Jedná se tedy o provázaný stav a lze vypozorovat, že změřením prvního qubitu je s jistotou určena i hodnota druhého qubitu. Můžeme totiž na prvním qubitu s pravděpodobností $\frac{1}{2}$ naměřit hodnotu 0, což zanechá stav po měření $|\psi'\rangle = |00\rangle$ a stejně tak můžeme s pravděpodobností $\frac{1}{2}$ naměřit hodnotu 1, což zanechá stav po měření $|\psi'\rangle = |11\rangle$.

John Bell dokázal, že korelace měření na tomto stavu jsou silnější, než jaké kdy mohou existovat mezi klasickými systémy. To byl první krok k poznání, že kvantová mechanika může poskytnout možnosti zpracovaní informací za hranicemi klasického světa.

Einstein, Podolsky a Rosen se ve své práci (viz [12]) na příkladu EPR párů snažili ukázat, že kvantová mechanika je nekompletní teorie, že v ní chybí něco, co nazývali *element reality*. Dostatečnou podmínkou pro to, aby fyzikální vlastnost byla elementem reality, mělo podle nich být, aby bylo možné bezprostředně před měřením předpovědět hodnotu dané vlastnosti. Vzhledem k tomu, že u EPR párů je vždy po měření prvního qubitu s jistotou možné určit hodnotu druhého qubitu, koresponduje tato vlastnost s elementem reality. Měla by tedy být reprezentována v nějaké "kompletní"teorii. Kvantová mechanika však poskytuje pouze pravděpodobnosti možných výsledků měření, nikoliv nějaký fundamentální element reprezentující přesný výsledek měření.

Téměř třicet let po vydání EPR práce však přišel první návrh experimentu na ověření této hypotézy a později přišly výsledy, které daly za pravdu kvantové mechanice a jejich tvrzení tak vyvrátily. Asi nejzásednějším z nich byl experiment realizovný roku 1982 Alainem Aspectem, Phillipe Grangierem a Gérardem Rogerem využívající polarizaci fotonů [5]. Klíčovou roli v tomto experimentu hrála tzv. Bellova nerovnost (viz [6]). Tu získáme, budeme-li následující myšlenkový experiment analyzovat klasickým způsobem.

Schéma experimentu je znázorněno na obrázku (1.2).



Obrázek 1.2: Bellova nerovnost

Mějme dvě připravené částice (a mějme možnost je takto připravit opakovaně) a pošleme jednu Alici a druhou Bobovi. Alice má dva měřící aparáty a může si zvolit, kterou vlastnost bude měřit. Označme tyto vlastnosti P_Q a P_R. Pokaždé, když Alice dostane částici, zvolí si nějakým náhodným způsobem

(např. hod mincí), jakou vlastnost bude měřit. Pro jednoduchost předpokládejme, že možné výsledky obou vlastností jsou ±1. Nechť hodnota vlastnosti P_Q částice připadající Alici je Q a hodnota vlastnosti P_R je R. Tyto jsou vnímány jako *objektivní vlastnosti*, tj. měřením jsou pouze odhaleny.

Dále předpokládejme, že Bob je podobně schopen měřit vlastnosti P_S a P_T s hodnotami S, T nabývající ±1 a stejně jako Alice si měřenou vlastnost vybírá náhodným způsobem. Nechť měření Alice a Boba probíhají současně (resp. tak, že jsou tyto události prostorupodobné a neexistuje tedy mezi nimi kauzální souvislost). Výsledky měření Alice a Boba se tedy nemohou navzájem ovlivňovat.

Provedeme nyní pár jednoduchých úprav s těmito hodnotami.

$$QS + RS + RT - QT = (R + Q)S + (R - Q)T \tag{1.9}$$

Uvážíme-li, že R, $Q = \pm 1$, pak buď $(R + Q)S = 0$ nebo $(R - Q)T = 0$. Tak či onak jsou výrazy v rovnici (1.9) rovny ±2. Nechť *p(q, r, s, t)* je pravděpodobnost, že před měřením je systém ve stavu takovém, že $Q= q$, $R = r$, $S = s$, $T = t$. Jelikož tato pravděpodobnost závisí na způsobu přípravy částic a na experimentálním ruchu, budeme uvažovat střední hodnotu těchto hodnot. Platí:

$$\begin{aligned} \langle QS + RS + RT - QT \rangle &= \sum_{q,r,s,t} p(q,r,s,t)(qs + rs + rt - qt) \\ &\leq \sum_{q,r,s,t} 2p(q,r,s,t) \\ &= 2 \end{aligned} \tag{1.10}$$

a podobně

$$\begin{aligned} \langle QS + RS + RT - QT \rangle &= \sum_{q,r,s,t} p(q,r,s,t)qs + \sum_{q,r,s,t} p(q,r,s,t)rs \\ &+ \sum_{q,r,s,t} p(q,r,s,t)rt - \sum_{q,r,s,t} p(q,r,s,t)qt \\ &= \langle QS \rangle + \langle RS \rangle + \langle RT \rangle - \langle QT \rangle. \end{aligned} \tag{1.11}$$

Porovnáním nerovnic (1.10) a (1.11) získáme tzv. CHSH nerovnost (viz [8]):

$$\langle QS \rangle + \langle RS \rangle + \langle RT \rangle - \langle QT \rangle \leq 2, \tag{1.12}$$

která je součástí dříve zmíněného širšího systému nerovností obecně známého jako Bellovy nerovnosti.

Opakováním experimentu lze určit střední hodnoty na levé straně nerovnice. Nechť se po sérii experimentů Alice a Bob setkají a podívají se na ty experimenty, kdy Alice měřila vlastnost P_Q a Bob měřil vlastnost P_S. Vynásobením a zprůměrováním výsledků experimentu získají $\langle QS \rangle$. Stejně lze postupovat i u ostatních středních hodnot.

Uvažujme nyní opět v řeči kvantové mechaniky a připravme systém dvou qubitů ve stavu $|\psi\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$, přičemž Alici pošleme první a Bobovi druhý qubit. Nechť Alice měří pozorovatelné

$$Q = Z_1 \qquad\qquad R = X_1$$

a Bob pozorovatelné

$$S = \frac{-Z_2 - X_2}{\sqrt{2}} \qquad\qquad T = \frac{Z_2 - X_2}{\sqrt{2}},$$

kde X, Z označují Pauliho matice σ_1, σ_3 a indexy 1, 2 označují první resp. druhou částici. Zde by bylo dobré zdůvodnit volbu těchto pozorovatelných. To je nejsnazší ukázat na příkladu spinu, ovšem vzhledem k již zmíněnému experimentu [5] to naznačím i v případě polarizace fotonů.

V případě spinu vidíme, že Alice měří projekci spinu ve směru osy z (Q) a ve směru osy x (R). Jak interpretovat Bobovo měření spinu? Můžeme definovat obecně operátor projekce spinu do směru $\boldsymbol{n} = (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta)$ vztahem $S_{\boldsymbol{n}} = \boldsymbol{n}\cdot\boldsymbol{\sigma}$. Zřejmě pak S odpovídá směrovému vektoru $\boldsymbol{n}_S = \left(-1/\sqrt{2}, 0, -1/\sqrt{2}\right)$ a T směrovému vektoru $\boldsymbol{n}_T = \left(-1/\sqrt{2}, 0, 1/\sqrt{2}\right)$. Při volbě $\varphi = \pi$ tak dostaneme úhly $\theta_S = 3\pi/4 = 135°$ a $\theta_T = \pi/4 = 45°$. Měří se tedy projekce spinu do směru daného úhlem θ v rovině xz na Blochově sféře. Právě při těchto úhlech nastává největší rozkol mezi predikcemi plynoucími z kvantové mechaniky (které si níže explicitně spočítáme) a klasickými úvahami (rozkol detailněji popsán v [11]). Pro interpretaci polarizace fotonů ztotožňujeme horizontální a vertikální polarizaci s bazickými vektory, tj. $|\leftrightarrow\rangle \equiv |0\rangle$, $|\updownarrow\rangle \equiv |1\rangle$. Reálně jsou vůči sobě polarizace natočené o 90°, ačkoli při reprezentaci na Blochově sféře tomu odpovídají opačné vektory. Superpozice $|\nearrow\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ pak odpovídá polarizaci pootočené o 45° a $|\nwarrow\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ polarizaci pootočené o -45°. Je snadné nahlédnout, že onoho největšího konfliktu mezi kvantovou mechanikou a nerovností (1.12) se dosáhne, budou-li měřené polarizace vzájemně natočeny o 22.5° resp. o 67.5° (viz [5]).

Spočítejme tedy střední hodnotu pozorovatelné QS:

$$\langle QS\rangle = \langle\psi|\, Q\otimes S\,|\psi\rangle = \frac{1}{2\sqrt{2}}\begin{pmatrix}0 & 1 & -1 & 0\end{pmatrix}\begin{pmatrix}-1 & -1 & 0 & 0\\ -1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\\ 0 & 0 & 1 & -1\end{pmatrix}\begin{pmatrix}0\\1\\-1\\0\end{pmatrix} = \frac{1}{\sqrt{2}} \tag{1.13}$$

Analogickým způsobem lze ověřit, že platí také:

$$\langle RS\rangle = \frac{1}{\sqrt{2}} \qquad \langle RT\rangle = \frac{1}{\sqrt{2}} \qquad \langle QT\rangle = -\frac{1}{\sqrt{2}}$$

a tedy

$$\langle QS\rangle + \langle RS\rangle + \langle RT\rangle - \langle QT\rangle = 2\sqrt{2}. \tag{1.14}$$

To je v rozporu s výsledkem (1.12). Jak už jsem ovšem na začátku uvedl, experimentálně bylo ověřeno, že příroda se řídí zákony kvantové mechaniky a tím pádem se nepodřizuje Bellově nerovnosti odvozené klasickým uvažováním.

Na řádcích výše se měřily pozorovatelné (ať už spin či polarizace), a tedy příslušné operátory (zde matice 2×2) určitě musely být samosdružené. Byly ovšem také unitární, což nás dostává k další podkapitole.

1.3 Kvantové brány

Představili jsme si qubit jakožto nositele informace v kvantové informatice a spolu s tím i některé jeho vlastnosti vymykající se klasickému pojetí informatiky. Nyní potřebujeme nástroje, které nám umožní s qubity manipulovat. Využijeme zde opět paralely se světem klasické informatiky, kde s bity pracují logické brány a z nich složené logické obvody. Zde se tedy budeme zabývat kvantovými branami a kvantovými obvody. Zaměřme se nejprve na brány manipulující s jedním qubitem.

1.3.1 Jednoqubitové brány

Podívejme se na příklad klasického bitu. Jediná netriviální logická brána pracující s jedním bitem je tzv. NOT brána, která mění hodnotu bitu z 0 na 1 a naopak. Naproti tomu kvantových bran pracujících s

jedním qubitem je mnoho. Tyto brány definujeme podle toho, jak působí na bazické stavy qubitu. Začnu asi vcelku intuitivní představou o tom, jak by měla pracovat kvantová NOT brána, značme ji X:

$$X|0\rangle = |1\rangle \qquad\qquad X|1\rangle = |0\rangle\,.$$

Ve standardní bázi mají bazické stavy tvar $|0\rangle \equiv \binom{1}{0}$ a $|1\rangle \equiv \binom{0}{1}$. Brána X tedy musí být matice 2×2, která má ve standardní bázi tvar

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{1.15}$$

Ukažme její působení na obecný stav qubitu popsaný rovnící (1.1), který lze ve standardní bázi zapsat vektorově jako $|\psi\rangle = \binom{\alpha}{\beta}$:

$$X\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}. \tag{1.16}$$

Ve spojitosti s touto branou je vhodné zmínit ještě dvě další:

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{1.17}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{1.18}$$

Dohromady tyto matice představují tzv. Pauliho matice ve standardní bázi.

Můžeme se ptát, jaké 2×2 matice můžeme použít coby kvantové brány pro jeden qubit. Vzpomeňme si na normalizační podmínku $|\alpha|^2 + |\beta|^2 = 1$ pro obecný stav (1.1). Přirozeně budeme požadovat, aby po působení kvantové brány byl výsledný stav opět normalizován k 1. Ukazuje se, že vhodná a překvapivě jediná podmínka pro matici, aby mohla představovat kvantovou bránu, je, aby byla *unitární*. To znamená, že pro matici U představující kvantovou bránu pro jeden qubit musí platit $U^\dagger U = I$, kde I je 2×2 jednotková matice. Toto okamžitě implikuje reverzibilitu kvantových bran, jelikož inverzní matice k unitární matici je opět unitární matice.

Představme si ještě několik dalších význačných jednoqubitových bran. Začněme s jednou z vůbec nejdůležitějších bran v kvantové informatice, tzv. Hadamardova brána:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{1.19}$$

Vidíme, že tato brána působí na bazické stavy tak, že je nastaví do vyvážené superpozice, tj. u obou bazických stavů je po jejím působení stejná amplituda, přičemž stavu $|1\rangle$ ještě přidá relativní fázi π. Dále budou důležité brány

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \qquad\qquad T = \begin{pmatrix} 1 & 0 \\ 0 & \exp\left\{i\frac{\pi}{4}\right\} \end{pmatrix},$$

kde S je tzv. fázová brána, která stavu $|1\rangle$ přidá fázi $\frac{\pi}{2}$ a T je tzv. $\frac{\pi}{8}$ brána, která stavu $|1\rangle$ přidá fázi $\frac{\pi}{4}$. Je snadné nahlédnout, že S získáme složením dvou T.

Jako poslední bych v této podkapitole zmínil matice rotace kolem os *x, y, z*, které vzniknou exponencializací Pauliho matic:

$$R_x(\Theta) \equiv \exp\left\{-i\Theta\frac{X}{2}\right\} = \cos\frac{\Theta}{2}I - i\sin\frac{\Theta}{2}X = \begin{pmatrix} \cos\frac{\Theta}{2} & -i\sin\frac{\Theta}{2} \\ -i\sin\frac{\Theta}{2} & \cos\frac{\Theta}{2} \end{pmatrix} \tag{1.20}$$

$$R_y(\Theta) \equiv \exp\left\{-i\Theta\frac{Y}{2}\right\} = \cos\frac{\Theta}{2}I - i\sin\frac{\Theta}{2}Y = \begin{pmatrix}\cos\frac{\Theta}{2} & -\sin\frac{\Theta}{2}\\ \sin\frac{\Theta}{2} & \cos\frac{\Theta}{2}\end{pmatrix} \tag{1.21}$$

$$R_z(\Theta) \equiv \exp\left\{-i\Theta\frac{Z}{2}\right\} = \cos\frac{\Theta}{2}I - i\sin\frac{\Theta}{2}Z = \begin{pmatrix}\exp\left\{-i\frac{\Theta}{2}\right\} & 0\\ 0 & \exp\left\{i\frac{\Theta}{2}\right\}\end{pmatrix}. \tag{1.22}$$

S těmito maticemi se pojí známý $Z-Y$ rozklad, což je vlastně parametrizace rotací pomocí Eulerových úhlů. Pro spin 1/2 lze stav jednoznačně identifikovat se směrovým vektorem v $\mathbb{R}^3$ pomocí Blochovy sféry. Unitární transformace pak působí rotaci vektoru, kterou můžeme v $\mathbb{R}^3$ parametrizovat právě Eulerovými úhly určujícími rotace okolo pevných os z, y a z. Navíc může stavu přidat dodatečnou globální fázi. Toto tvrzení je shrnuto v následující větě, která má i jeden velmi důležitý důsledek.

Věta 1 (Z-Y rozklad)**.** Nechť U je unitární operace na jednom qubitu. Pak existují reálná čísla $\alpha, \beta, \gamma, \delta$ tak, že

$$U = \exp\{i\alpha\}R_z(\beta)R_y(\gamma)R_z(\delta) \tag{1.23}$$

Důkaz. Vzhledem k tomu, že U je unitární, tak řádky a sloupce tohoto operátoru musí být ortonormální. Z toho plyne existence reálných čísel $\alpha, \beta, \gamma, \delta$ tak, že

$$U = \begin{pmatrix}e^{i(\alpha-\frac{\beta}{2}-\frac{\delta}{2})}\cos\frac{\gamma}{2} & -e^{i(\alpha-\frac{\beta}{2}+\frac{\delta}{2})}\sin\frac{\gamma}{2}\\ e^{i(\alpha+\frac{\beta}{2}-\frac{\delta}{2})}\sin\frac{\gamma}{2} & e^{i(\alpha+\frac{\beta}{2}+\frac{\delta}{2})}\cos\frac{\gamma}{2}\end{pmatrix}. \tag{1.24}$$

Rovnost (1.23) získáme roznásobením matic. □

Důsledek 1.1 (Z-Y rozklad)**.** Nechť U je unitární operátor působící na jeden qubit. Pak existují unitární operátory A, B, C působící na jeden qubit takové, že $ABC = I$ a $U = e^{i\alpha}AXBXC$, kde $e^{i\alpha}$ představuje nějakou globální fázi.

Důkaz. Ve větě 1 položme $A \equiv R_z(\beta)R_y(\frac{\gamma}{2})$, $B \equiv R_y(-\frac{\gamma}{2})R_z(-\frac{\delta+\beta}{2})$ a $C \equiv R_z(\frac{\delta-\beta}{2})$. Vzhledem k tomu, že přirozeně inverze rotace má tvar $R_i^{-1}(\varphi) = R_i(-\varphi); i \in \{x, y, z\}, \varphi \in \mathbb{R}$, tak se pouhým dosazením snadno přesvědčíme, že

$$ABC = I. \tag{1.25}$$

Prostým vynásobením Pauliho matic se lze snadno přesvečit, že platí rovnost

$$XYX = -Y, \tag{1.26}$$

ovšem lepší je na tento vztah nahlížet tak, že antikomutátor $\{X, Y\} = 0$ a zároveň platí $X^2 = I$. Z toho již plyne vztah

$$XR_y(\varphi)X = Xe^{-i\frac{\varphi}{2}Y}X = e^{i\frac{\varphi}{2}Y}X^2 = R_y(-\varphi) \tag{1.27}$$

Tohoto využijeme v následující rovnosti

$$XBX = XR_y(-\frac{\gamma}{2})XXR_z(-\frac{\delta+\beta}{2})X = R_y(\frac{\gamma}{2})R_z(\frac{\delta+\beta}{2}). \tag{1.28}$$

Dosazením již získáme

$$AXBXC = R_z(\beta)R_y(\gamma)R_z(\delta), \tag{1.29}$$

a tedy

$$U = \exp\{i\alpha\}R_z(\beta)R_y(\gamma)R_z(\delta). \tag{1.30}$$

□

1.3.2 Vícequbitové brány

Kvantové brány samozřejmě mohou působit na více qubitů najednou. Nejjednodušším a zároveň nejdůležitějším příkladem brány manipulující se dvěma qubity je tzv. *CNOT* brána

$$U_{CN} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{1.31}$$

působící na bazické stavy následovně:

$$U_{CN}\,|00\rangle = |00\rangle \qquad U_{CN}\,|01\rangle = |01\rangle \qquad U_{CN}\,|10\rangle = |11\rangle \qquad U_{CN}\,|11\rangle = |10\rangle\,.$$

Tuto skutečnost lze souhrnně vystihnout zápisem

$$U_{CN}\,|A, B\rangle = |A, B \oplus A\rangle\,, \tag{1.32}$$

kde $\oplus$ značí sčítání modulo 2.

Poslední brána, kterou zde zmíníme, je tříqubitová Toffoliho (zvaná také *CCNOT*) brána daná maticí

$$T_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{1.33}$$

Zde první dva qubity figurují jako kontrolní a hodnota třetího je změněna, pouze pokud jsou oba kontrolní ve stavu $|1\rangle$. Význam této brány je v tom, že v klasické informatice se pouze s její pomocí dají sestavit *univerzální reverzibilní* obvody. Pravdivostní tabulka Toffoliho brány je označena číslem (1.1). Podíváme-li se totiž na *NAND* bránu (pravdivostní tabulka (1.2)), která v klasické informatice funguje

Toffoli					
Vstup			Výstup		
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Tabulka 1.1: Pravdivostní tabulka klasické Toffoliho brány

jako univerzální (tj. každá Booleovská funkce na klasických bitech může být počítána pouze za pomoci složení NAND brán), tak snadno zjistíme, že není reverzibilní. Vzhledem k tomu, že kvantové brány

NAND		
Vstup		Výstup
0	0	1
0	1	1
1	0	1
1	1	0

Tabulka 1.2: Pravdivostní tabulka klasické NAND brány

musí být unitární, tak musí existovat jejich inverze a jsou tedy vždy reverzibilní. Může se tedy zdát, že ne každý klasický logický obvod může být simulován pomocí kvantového logického obvodu. Tento zdánlivý paradox řeší právě Toffoliho brána. Lze totiž ukázat, že každý klasický logický obvod může být nahrazen ekvivalentním obvodem obsahujícím pouze reverzibilní prvky za využití Toffoliho brány. Všimněme si, že uvažujeme-li první dva vstupní bity v Toffoliho bráně jako vstup do NAND brány a třetí bit je nastaven na 1, pak Toffoliho brána simuluje NAND bránu, přičemž výstup NAND brány je na třetím bitu. Její opětovnou aplikací pak získáme bity v původním stavu. Obdobně lze pomocí Toffoliho brány simulovat FANOUT brána, kterou už pro potřeby této práce nezmiňuji. Důsledkem těchto operací ovšem je, že tímto reverzibilním způsobem můžeme simulovat jakýkoliv prvek klasického logického obvodu. Z toho okamžitě plyne, že kvantové počítače jsou schopny všech výpočtů, kterých jsou schopny klasické počítače.

1.3.3 Univerzální kvantové brány

V této podkapitole se zaměříme na otázku univerzálnosti souboru kvantových bran. Soubor kvantových bran je v kvantové informatice univerzální, můžeme-li jakoukoli unitární operaci libovolně dobře aproximovat pomocí kvantového obvodu obsahujícího pouze brány z tohoto souboru. Budu se tedy snažit takovýto soubor najít. Při odvozování použiji např. některé důsledky numerické matematiky, jejichž dokazování je mimo rámec této práce.

Nejprve ukáži, že jakýkoli unitární operátor muže být vyjádřen *přesně* jako produkt unitárních operátorů, z nichž každý působí netriviálně pouze na podprostoru určeném dvěma stavy výpočetní báze. Vezměme tedy $d \times d$ unitární matici U působící na d-dimnezionálním Hilbertově prostoru. Hledáme unitární matice $U_1, ..., U_n$, $n \in \mathbb{N}$ takové, že každá z nich vznikne z $d \times d$ jednotkové matice změnou 2×2 submatice (která vznikla vynecháním řádků a sloupců se stejným indexem) tak, aby $U_n U_{n-1} ... U_1 U = I$. Z unitárnosti pak okamžitě bude platit $U_1^\dagger ... U_n^\dagger = U$. Tento proces je dobře známý a lze ukázat, že $n \leq \frac{d(d-1)}{2}$. Vezmeme-li v úvahu, že pro systém n qubitů má Hilbertův prostor dimenzi 2^n, pak každá unitární matice na tomto prostoru může být zapsána jako součin maximálně $2^n(2^n - 1)$ unitárních matic působících netriviálně pouze na podprostoru určeném lineárním obalem dvou stavů výpočetní báze.

Nyní naznačíme způsob, jakým lze ukázat, že za pomoci jednoqubitových bran spolu CNOT bránou lze implementovat libovolnou unitární operaci působící pouze na dva stavy výpočetní báze. Spolu s předchozím odstavcem pak dostaneme univerzálnost tohoto souboru. Mějme tedy unitární operaci U působící netrviálně pouze na dva stavy výpočetní báze, řekněme $|s\rangle$ a $|t\rangle$, kde $s = s_1...s_n, t = t_1...t_n$ jsou binární rozvoje těchto s a t. Nazvěme $\tilde{U}$ onu netriviální submatici U, která může být brána jako jednoqubitová brána. Za pomoci Grayova kódu (sekvenci změn bit po bitu) lze spojit tyto dva binární rozvoje, což odpovídá přechodu stavů $|g_1\rangle \to ... \to |g_m\rangle$, kde $|g_1\rangle = |s\rangle$ a $|g_m\rangle = |t\rangle$. Vidíme, že $m \leq n+1$. U implementujeme tak, že na stav $|g_{m-1}\rangle$ použijeme kontrolovanou $\tilde{U}$ operaci, přičemž cílový qubit se bude nacházet na onom jediném bitu, kde se binární rozvoje $|g_{m-1}\rangle$ a $|g_m\rangle$ liší. Podmíněn bude právě tím,

že na všech ostatních pozicích jsou tyto dva stavy stejné. Pak už stačí jen zpětně přejít $|g_{m-1}\rangle \rightarrow ... \rightarrow |g_1\rangle$. Tímto je implementace U dokončena.

Ukázali jsme, že CNOT spolu s jednoqubitovými branami tvoří univerzální soubor. Jak je to se složitostí? Lze ukázat, že pro každé prohození $|g_{k-1}\rangle \rightarrow |g_k\rangle$ a pro operaci $\tilde{U}$ je potřeba $O(n)$ jednoqubitových a CNOT bran. Těchto prohození je pro realizaci operace U působící pouze na dva stavy výpočetní báze potřeba $2(n-1)$, což nám dává složitost $O(n^2)$. V prvním odstavci této podkapitoly jsme dospěli k závěru, že jakákoliv unitární operace lze přepsat jako součin $2^n(2^n - 1)$ těchto bran působících netriviálně jen na podprostoru dvou stavů výpočetní báze. To už nám dává celkovou složitost $O(n^2 4^n)$.

Problém tohoto souboru je, že ne všechny jednoqubitové brány umíme implementovat způsobem odolným vůči chybám. Existují ovšem diskrétní soubory bran, s jejichž pomocí jsme schopni provést jakýkoli kvantový výpočet a které jsme schopni implementovat tak, aby byly odolné vůči chybám (odolnost vůči chybám a jejich opravování je obecně mimo rámec této práce a nebudeme se tomu již více věnovat). Vzhledem k tomu, že množina unitárních operací je spojitá, tak diskrétním souborem můžeme tyto operace pouze aproximovat. V tomto smyslu existují dva nejzásadnější univerzální soubory. První tzv. *standardní soubor* je tvořen Hadamardovou, fázovou, CNOT a $\frac{\pi}{8}$ bránou, kde fázová brána je zahrnuta kvůli toleranci vůči chybám (vzhledem k univerzalitě je irelevantní, když může být složena ze dvou $\frac{\pi}{8}$ bran). Druhý tvoří Hadamardova, fázová, CNOT a Toffoliho brána. Lze ukázat, že pouze $\frac{\pi}{8}$ brána spolu Hadamardou dokáží libovolně přesně aproximovat každou unitární operaci. Tento fakt má kořeny v tom, že T představuje rotaci okolo osy z o $\frac{\pi}{4}$ a THT rotaci okolo osy x o $\frac{\pi}{4}$. Spolu s CNOT bránou tak v návaznosti na předešlé odstavce představují univerzální soubor. Stejně tak lze ukázat i univerzálnost druhého zmíněného souboru.

Kapitola 2

Deutsch-Jozsův algoritmus

V této kapitole se postupně dobereme k prvnímu kvantovému algoritmu, na kterém lze alespoň v teoretické rovině demonstrovat kvantovou nadřazenost. Jedná se o Deutsch-Jozsův algoritmus, k jehož výsledné podobě povedou dva mezikroky. Nejprve si ukážeme, jak v sobě kvantový obvod, určený k výpočtu jisté funkce, dokáže do okamžiku měření udržovat výsledek pro všechny její možné vstupy. Poté na příkladu Deutschova algoritmu (což je vlastně jen zjednodušená verze Deutsch-Jozsova algoritmu) budeme prezentovat, jak pomocí interference amplitud pravděpodobnosti z obvodu získat potřebnou informaci. V poslední fázi tyto myšlenky zobecníme a výsledkem bude obvod schopný řešit daný problém efektivněji než by bylo možné klasickým způsobem. Kapitola bude shrnutím myšlenek z [21] a [31] a bude doplněna kódy a výpočty provedenými na kvantovém počítači IBMQ.

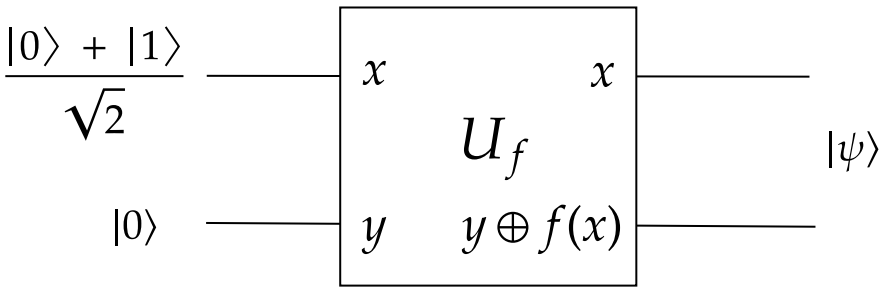
2.1 Paralelní kvantový výpočet

Ukažme tedy, jak dokáže kvantový počítač vyhodnotit určitou funkci pro více jejích vstupů najednou. Celý trik bude spočívat v tom, že qubity dostaneme do stavu superpozice, na němž vykonáme požadovanou akci. Uvažujme tedy funkci $f(x) : \{0, 1\} \to \{0, 1\}$. Je zřejmé, že takováto funkce může působit pouze čtyřmi různými způsoby:

$$\begin{aligned} f_1(0) &= 0 & f_1(1) &= 0, \\ f_2(0) &= 1 & f_2(1) &= 1, \\ f_3(0) &= 0 & f_3(1) &= 1, \\ f_4(0) &= 1 & f_4(1) &= 0. \end{aligned} \tag{2.1}$$

My zatím nevíme, o jakou z těchto funkcí se jedná. Tuto lze spočítat s využitím registru, který obsahuje dva qubity. Nechť je jejich počáteční stav $|x, y\rangle$. Pomocí vhodných bran lze systém transformovat do stavu $|x, y \oplus f(x)\rangle$, kde $\oplus$ značí sčítání modulo 2. V tomto případě slouží první qubit jako vstup funkce a říká se mu datový registr. Na druhém qubitu se projeví působení funkce a nazývá se cílový registr. Transformaci $|x, y\rangle \to |x, y \oplus f(x)\rangle$ nazvěme U_f a je snadné se přesvědčit o její unitárnosti. Ta pro nás bude zatím realizována jakousi černou skříňkou (pro kterou nadále budeme užívat anglický název *oracle*), o které víme jen to, že funguje uvedeným způsobem.

Snadno si rozmyslíme, že bude-li druhý qubit na začátku ve stavu $|0\rangle$, pak po aplikaci U_f v něm bude prostě uložen výsledek $f(x)$. Co kdybychom ovšem měli situaci jako na obrázku (2.1), kdy je první vstupní qubit ve stavu $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ (čehož se snadno dosáhne aplikací Hadamardovy brány na stav $|0\rangle$)?



Obrázek 2.1: Ukázka paralelního výpočtu pro obě vstupní hodnoty najednou

V tomto případě bude mít stav po působení U_f tvar

$$\frac{|0, f(0)\rangle + |1, f(1)\rangle}{\sqrt{2}}. \tag{2.2}$$

Z toho je zřetelné, že se nám v jistém slova smyslu povedlo spočítat funkci $f(x)$ pro oba její možné vstupy najednou.

Tento typ výpočtu lze zobecnit na libovolný počet bitů s využitím tzv. *Hadamardovy transformace* (také nazývané *Walsh-Hadamardova*). Můžeme opět uvažovat nějakou funkci $f(x) : \{0, 1, ..., 2^n - 1\} \to \{0, 1\}$ (toto lze vystihnout tím, že pro vstup nám stačí n bitů). Tentokrát budeme mít n qubitů v datovém registru a opět jeden v cílovém registru. Hadamardova transformace na n qubitů jednoduše znamená aplikaci Hadamardovy brány na každý z nich, značíme $H^{\otimes n}$. Při aplikaci na n qubitů ve stavu $|0\rangle$ dostáváme:

$$H^{\otimes n} \underbrace{|0\rangle ... |0\rangle}_{\text{n-krát}} = \underbrace{\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) ... \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)}_{\text{n-krát}} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle\,, \tag{2.3}$$

kde $|x\rangle$ prochází všechny bazické stavy. Vidíme, že tím získáváme rovnovážnou superpozici všech stavů výpočetní báze n qubitů. Je dobré zmínit i efektivnost této transformace, která na vytvoření 2^n stavů potřebuje pouze n bran.

Nyní máme tedy qubity v datovém registru v této superpozici, cílový qubit ve stavu $|0\rangle$ a necháme-li působit U_f, získáme stav

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle\,|f(x)\rangle\,. \tag{2.4}$$

Vidíme, že v funkce $f(x)$ je zde vyčíslena pro všechny své možné vstupní hodnoty. Může se zdát, že jsme takto získali opravdu mocný nástroj, s jehož pomocí jsme schopni vyhodnocovat funkci pro všechny její vstupy naráz. Naneštěstí z postulátu kvantové mechaniky plyne, že při měření se v závislosti na amplitudě pravděpodobnosti vybere pouze jeden stav odpovídající jen jedné vstupní hodnotě, přičemž ostatní stavy superpozice spolu s výsledky funkce pro jiné vstupy jsou nenávratně ztraceny. V našem případě mají všechny členy superpozice stejnou amplitudu (a tedy i stejnou pravděpodobnost naměření) a výsledný výběr je tedy zcela náhodný. Musíme tudíž najít způsob, jakým bychom ze superpozice získali nějakou další informaci než jen o jediné vstupní hodnotě. To nás dostává k další podkapitole.

2.2 Deutschův algoritmus

Předchozí poznatky využijeme při řešení problému, který roku 1985 ve své práci [10] představil David Deutsch a kde na výsledném algoritmu prezentoval, jak mohou kvantové počítače překonat ty klasické. Přesněji se budeme zabývat zjednodušenou a v jistém smyslu vylepšenou verzí tohoto algoritmu.

Uvažujme opět funkci $f(x) : \{0,1\} \to \{0,1\}$ a její možné podoby (2.1). O funkcích f_1 a f_2 řekneme, že jsou konstantní, zatímco f_3 a f_4 jsou vyvážené (tj. na výstupu mají stejný počet jedniček a nul). Vrátíme se tedy k případu, kdy datový registr má $n = 1$ qubitů. Naším úkolem bude určit, zda-li se jedná o funkci konstantní, či vyváženou.

Představme si nyní klasický případ. Abychom mohli s jistotou říct, že se jedná o ten, či onen typ funkce, museli bychom funkci za pomoci oracle vyčíslit pro obě možné vstupní hodnoty. My si zde ukážeme, jak o této globální vlastnosti dané funkce rozhodnout již při jediném použití.

Budeme opět uvažovat transformaci U_f s působením $|x, y\rangle \to |x, y \oplus f(x)\rangle$ jako v předešlé kapitole. Mějme tedy klasický počáteční stav

$$|\psi_0\rangle = |0\rangle\,|0\rangle \tag{2.5}$$

a postupujme krok po kroku. Nyní na druhý qubit použijeme X bránu, čímž ho dostaneme do stavu $|1\rangle$. Stav $|\psi_1\rangle$ získáme aplikací Hadamardovy brány na oba qubity:

$$|\psi_1\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = \frac{|0\rangle\,(|0\rangle - |1\rangle) + |1\rangle\,(|0\rangle - |1\rangle)}{2} \tag{2.6}$$

Nyní nechme zapůsobit černou skříňku U_f:

$$|\psi_2\rangle = U_f\,|\psi_1\rangle = \frac{|0\rangle\,(|f(0)\rangle - |1 \oplus f(0)\rangle) + |1\rangle\,(|f(1)\rangle - |1 \oplus f(1)\rangle)}{2}. \tag{2.7}$$

Pro představu ukažme, jak konkrétně vypadá tento stav pro jednotlivé funkce z (2.1):

$$\begin{aligned}
|\psi_{2,1}\rangle &= \frac{|0\rangle\,(|0\rangle - |1\rangle) + |1\rangle\,(|0\rangle - |1\rangle)}{2} = \frac{(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)}{2} \\
|\psi_{2,2}\rangle &= \frac{|0\rangle\,(|1\rangle - |0\rangle) + |1\rangle\,(|1\rangle - |0\rangle)}{2} = \frac{(-|0\rangle - |1\rangle)(|0\rangle - |1\rangle)}{2} \\
|\psi_{2,3}\rangle &= \frac{|0\rangle\,(|0\rangle - |1\rangle) + |1\rangle\,(|1\rangle - |0\rangle)}{2} = \frac{(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)}{2} \\
|\psi_{2,4}\rangle &= \frac{|0\rangle\,(|1\rangle - |0\rangle) + |1\rangle\,(|0\rangle - |1\rangle)}{2} = \frac{(-|0\rangle + |1\rangle)(|0\rangle - |1\rangle)}{2}.
\end{aligned}$$

Toto lze souhrnně zapsat jako

$$|\psi_2\rangle = \frac{\overbrace{(-1)^{f(0)}\,|0\rangle + (-1)^{f(1)}\,|1\rangle}^{\sum_x (-1)^{f(x)}|x\rangle}}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{2.8}$$

Vidíme, že působení U_f se vlastně přeneslo z druhého qubitu na první. Tomuto jevu se často říká *fázový zpětný ráz*. Obecně se jedná o jev, kdy při kontrolované operaci je vlastní hodnota přidaná samotnou operací předána na jiný qubit. Rychle si tuto vlastnost ilustrujeme na kontrolované T bráně, kde kontrolní qubit bude ve stavu $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ a cílový ve stavu $|1\rangle$.

$$CT\,|+1\rangle = \frac{|01\rangle + e^{i\pi/4}\,|11\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{i\pi/4}\,|1\rangle}{\sqrt{2}}\,|1\rangle \tag{2.9}$$

Vidíme, že fázový posun se promítl do kontrolního qubitu.

Vraťme se k našemu případu. Z (2.8) už je vidět, že je-li funkce f konstantní, bude

$$|\psi_2\rangle = \pm\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{2.10}$$

Naopak bude-li vyvážená, pak

$$|\psi_2\rangle = \pm\frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{2.11}$$

V tuto chvíli už druhý qubit není důležitý. Aplikací Hadamardovy brány dostaváme

$$|\psi_3\rangle = \pm\,|0\rangle\,\frac{|0\rangle - |1\rangle}{\sqrt{2}}, \tag{2.12}$$

respektive

$$|\psi_3\rangle = \pm\,|1\rangle\,\frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{2.13}$$

Zde již můžeme měřením prvního qubitu s jistotou určit, zda-li se jedná o funkci konstantní či vyváženou.Uvědomme si ještě, že $f(0) \oplus f(1) = 0$, pokud $f(0) = f(1)$ a pokud ne, pak je to rovno 1. Takto lze souhrnně zapsat

$$|\psi_3\rangle = \pm\,|f(0) \oplus f(1)\rangle\,\frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{2.14}$$

Kvantový obvod nám tedy pomohl určit globální vlastnost $f(0) \oplus f(1)$ funkce f při jejím jediném vyhodnocení.

Z předchozího postupu nemusí být na první pohled jasné, kde přesně hrála roli zmiňovaná interference amplitud. To je nejzřetelnější, podíváme-li se na první qubit v rovnici (2.8). Aplikací Hadamardovy brány na tento qubit získáme stav

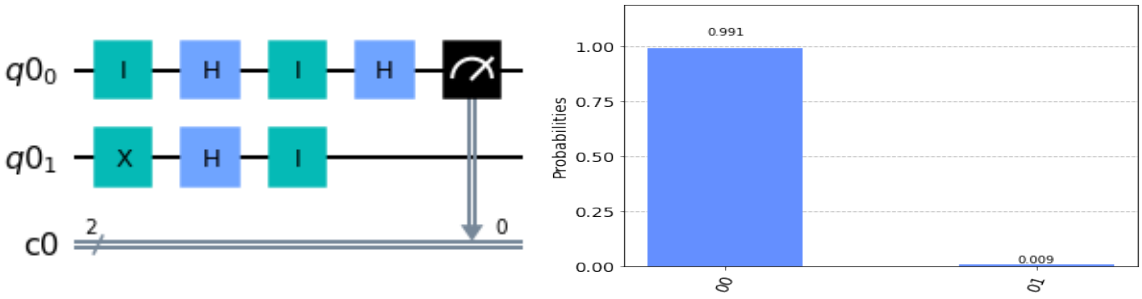
$$|\psi_4\rangle = \frac{((-1)^{f(0)} + (-1)^{f(1)})\,|0\rangle + ((-1)^{f(0)} - (-1)^{f(1)})\,|1\rangle}{2} \tag{2.15}$$

a spočtěme pravděpodobnost naměření qubitu ve stavu $|0\rangle$:

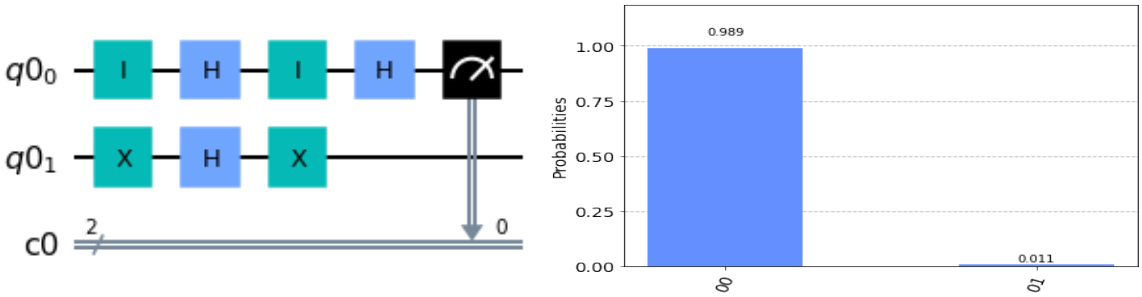
$$P(0) = \left|\frac{1}{2}((-1)^{f(0)} + (-1)^{f(1)})\right|^2. \tag{2.16}$$

Zde je zřetelná závislost na interferenci amplitud. Pro konstantní funkci budou mít amplitudy stejnou fázi a interferovat konstruktivně s výslednou pravděpodobností 1. Naopak u vyvážené funkce dojde k destruktivní interferenci s výslednou pravděpodobností rovnou nule, což již implikuje jistotu naměření hodnoty qubitu rovné 1.

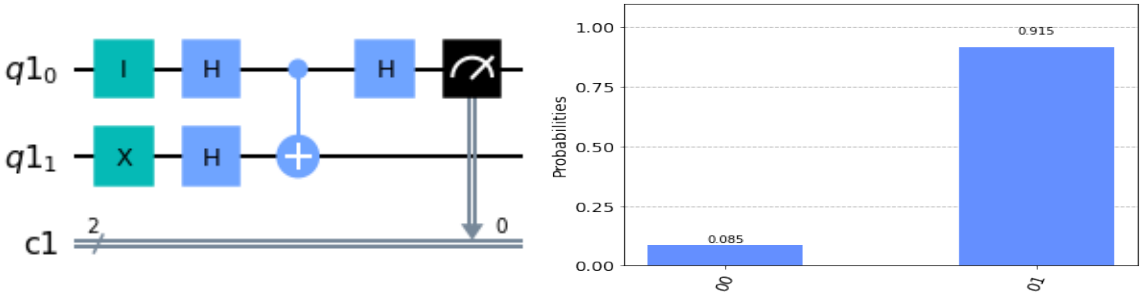
Na závěr této podkapitoly s pomocí platformy IBM Quantum a veřejně dostupné sady pro vývoj softwaru určeného ke kvantovým výpočtům Qiskit [1] poskytnu grafické znázornění kvantových obvodů pro Deutschův algoritmus pro jednotlivé funkce i s jejich statistikami měření. Tyto jsou znázorněné na obrázcích (2.2), (2.3), (2.4) a (2.5). Nutno přitom upozornit na dvě věci. První je, že Qiskit používá opačné řazení qubitů, než je tomu zde i ve většině literatury. Normálně se první prvek tenzorového součinu řadí doleva a poslední doprava. Qiskit používá řazení, kdy MSB (Most Significant Bit) je nalevo a LSB (Least Significant Bit) napravo, což odpovídá reprezentaci bitů v klasických počítačích a umožňuje tak snadnou konverzi řady bitů na čísla po měření. V histogramu se toto projeví tak, že onen požadovaný výsledek měření prvního qubitu je na druhém bitu, přičemž druhý qubit (v histogramu na první pozici) je stále považován ve stavu 0, protože jsme na něm neprováděli žádné měření. Druhou je, že současná kvantová zařízení nejsou dokonalá, proto se v malém množství ve statistikách objevují i výsledky odporující odvozeným teoretickým důsledkům.



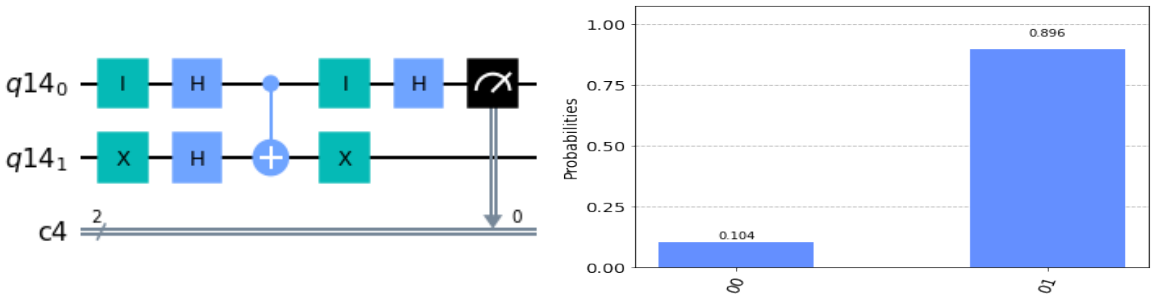
Obrázek 2.2: Deutschův algoritmus pro konstantní funkci f_1. U_f je zde realizována pouze pomocí identity na prvním i druhém qubitu.



Obrázek 2.3: Deutschův algoritmus pro konstantní funkci f_2. U_f je zde realizována pomocí identity na prvním a NOT brány na druhém qubitu.



Obrázek 2.4: Deutschův algoritmus pro vyváženou funkci f_3. U_f je zde realizována pomocí CNOT brány, kde první qubit je kontrolní a druhý cílový.

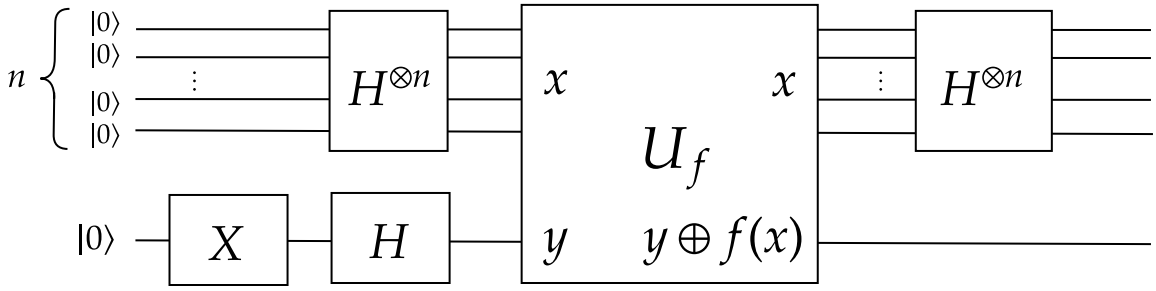


Obrázek 2.5: Deutschův algoritmus pro vyváženou funkci f_4. U_f je zde realizována pomocí CNOT brány, kde první qubit je kontrolní a druhý cílový a následné NOT brány na druhém qubitu.

2.3 Deutsch-Jozsův algoritmus

Obecnější problém a kvantový algoritmus tento problém řešící představili David Deutsch a Richard Jozsa ve své práci [9] roku 1992. My zde budeme opět prezentovat poněkud upravenou verzi tohoto problému.

Uvažujme opět funkci $f : \{0, ..., 2^n - 1\} \to \{0, 1\}$ zmíněnou v první sekci této kapitoly. O funkci víme, že je buď konstantní, nebo vyvážená. Naším úkolem bude rozhodnout o této vlastnosti s využitím předchozích poznatků. Poznamenejme, že k tomu, abychom klasickým výpočtem s jistotou určili tuto vlastnost, tak bychom potřebovali v nejhorším případě (kdy bychom měřili stále jen samé jedničky nebo naopak samé nuly) vyhodnotit funkci pro $2^{n-1} + 1$ hodnot. Diagram Deutsch-Joszova algoritmu je na obrázku (2.6).



Obrázek 2.6: Diagram Deutsch-Jozsova algoritmu.

Začneme tedy náš kvantový výpočet s n qubity v datovém registru a jedním qubitem v cílovém registru, všechny nechť jsou ve stavu $|0\rangle$. Máme tedy klasický počáteční stav

$$|\psi_0\rangle = \underbrace{|0\rangle ... |0\rangle}_{\text{n-krát}} \otimes |0\rangle \,. \tag{2.17}$$

Na datový registr použijeme Hadamardovu transformaci (2.3), zatímco na cílový qubit aplikujeme nejdříve NOT a pak Hadamardovu bránu. Stav registru poté bude

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \tag{2.18}$$

kde celé číslo $x \in \{0, ..., 2^n - 1\}$ identifikuji s jeho binárním rozvojem $x \equiv x_{n-1}...x_1x_0$, $x_i \in \{0, 1\}$. Mějme opět oracle U_f působící stejně jako v předchozí kapitole, tj. $U_f : |x, y\rangle \to |x, y \oplus f(x)\rangle$ a nechme ji působit na $|\psi_1\rangle$:

$$|\psi_2\rangle = U_f |\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes \frac{|f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}} \tag{2.19}$$

Podívejme se nyní vztah (2.8) a jakým způsobem jsme k němu dospěli. Zcela analogickými úvahami lze za využití fázového zpětného rázu předešlý vztah přepsat do tvaru

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{2.20}$$

V tuto chvíli již není poslední qubit důležitý. Provedeme dva pomocné výpočty. Nejprve se snadno přesvědčíme, že působení Hadamardovy brány na jeden qubit ve stavu $|x\rangle$ pro $x \in \{0, 1\}$ lze psát takto:

$$H |x\rangle = \frac{1}{\sqrt{2}} \sum_{z=0}^{1} (-1)^{xz} |z\rangle . \tag{2.21}$$

Potom Hadamardova transformace na n qubitů je

$$\begin{aligned} H^{\otimes n} |x_{n-1}, ..., x_0\rangle &= \frac{1}{\sqrt{2^n}} \sum_{z_{n-1}=0}^{1} (-1)^{x_{n-1}z_{n-1}} |z_{n-1}\rangle \otimes ... \otimes \sum_{z_0=0}^{1} (-1)^{x_0z_0} |z_0\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{z_{n-1}=0}^{1} ... \sum_{z_0=0}^{1} (-1)^{x_{n-1}z_{n-1}+...+x_0z_0} |z_{n-1}, ..., z_0\rangle , \end{aligned} \tag{2.22}$$

což lze s pomocí bitového skalárního součinu $x \cdot z$ velmi elegantně zapsat jako

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} (-1)^{x \cdot z} |z\rangle , \tag{2.23}$$

kde opět bereme x i z jako binární rozvoje. Pustíme-li tedy Hadamardovu transformaci na prvních n qubitů stavu $|\psi_2\rangle$, získáme

$$\begin{aligned} |\psi_3\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \left(\frac{\sum_{z=0}^{2^n-1} (-1)^{x \cdot z} |z\rangle}{\sqrt{2^n}} \right) \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \\ &= \frac{1}{2^n} \sum_{z=0}^{2^n-1} \left(\sum_{x=0}^{2^n-1} (-1)^{f(x)+x \cdot z} \right) |z\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \end{aligned} \tag{2.24}$$

Zde opět začíná být zřejmá závislost na interferenci amplitud. Prozkoumejme pravděpodobnost naměření počátečního stavu datového registru, tedy stavu $|0...0\rangle$:

$$P(0...0) = \left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \right|^2 . \tag{2.25}$$

Nyní uvažme možnosti, kdy je funkce f konstantní a kdy vyvážená. Je-li konstantní, tak amplituda u tohoto stavu je díky konstruktivní interferenci ± 1 (a vzhledem k normalizaci stavu k jedné musí být

ostatní amplitudy rovny nule), a tedy pravděpodobnost je rovna jedné. Je-li naopak funkce vyvážená, tak pro přesně polovinu členů v sumě (2.25) bude $f(x)$ rovno 1 a pro polovinu 0. Členy se tak díky destruktivní interferenci vyruší a pravděpodobnost naměření tohoto stavu je rovna 0. Pokud tedy na datovém registru naměříme alespoň jeden qubit ve stavu $|1\rangle$, tak můžeme s jistotou říci, že se jedná o vyváženou funkci. Přitom jsme využili pouze jedné iterace vyhodnocení této funkce a dosáhli jsme tedy exponenciálního snížení počtu použití oracle.

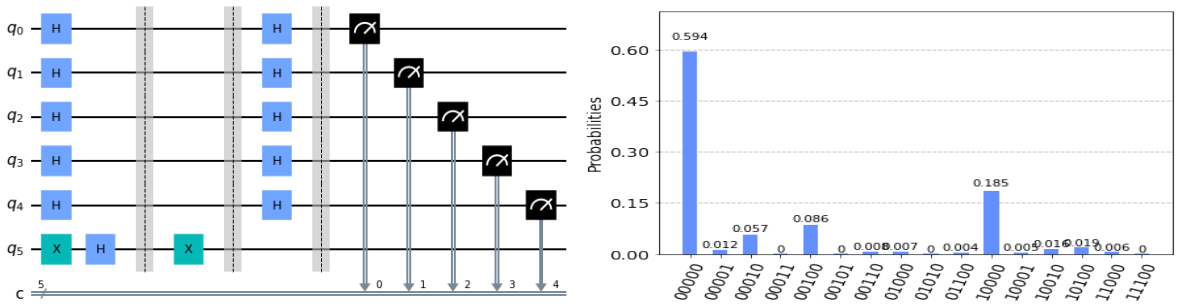
Nutno však zmínit, že klasickým pravděpodobnostním algoritmem také nalezneme řešení v konstantním čase, tj. $O(1)$. V nejlepším případě, použijeme-li oracle dvakrát a dostaneme různé výsledky, pak víme, že je funkce vyvážená. Dostáváme-li však stále stejné výstupy, tak i v tomto případě můžeme již po pár vyhodnoceních s velmi slušnou pravděpodobností rozhodnout o konstantnosti funkce. Snadno nahlédneme, že pravděpodobnost, že je funkce konstantní, lze zapsat jako funkci počtu vyhodnocených vstupů k:

$$P_{konst}(k) = 1 - \frac{1}{2^k} \tag{2.26}$$

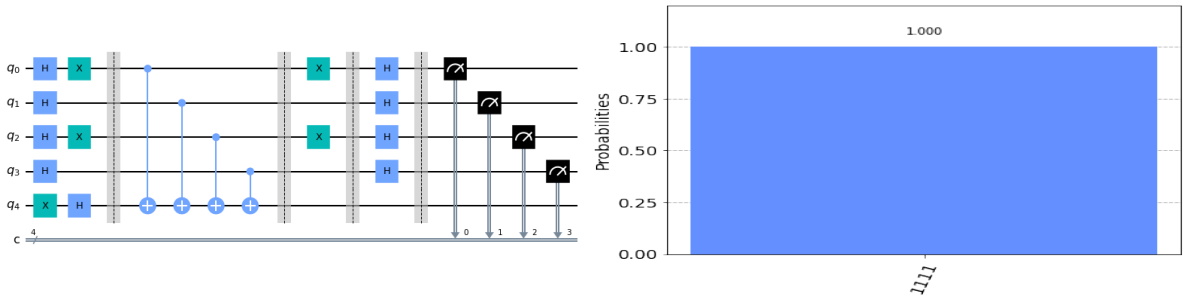
pro $1 \leq k \leq 2^{n-1}$. Vidíme tedy, že počet použití černé skříňky zde nezávisí na délce vstupu n, pouze na toleranci chyb.

Nakonec se opět podívejme na implementaci pomocí Qiskit a IBMQ. Udáme zde jeden příklad pro konstantní a jeden pro vyváženou funkci. Implementace oracle pro konstantní funkci je jednoduchá. Vzhledem k tomu, že její působení se na posledním qubitu projevuje jako $|y \oplus f(x)\rangle$, tak snadno nahlédneme, že jedná-li se o $f(x) = 0$, pak na cílový qubit nemusíme nijak měnit, stačí aplikovat identitu. Naopak pokud jde o $f(x) = 1$, pak stačí na poslední qubit aplikovat X bránu. Obvod s implementací takovéto funkce a pěti vstupními qubity je znázorněn na obrázku (2.7).

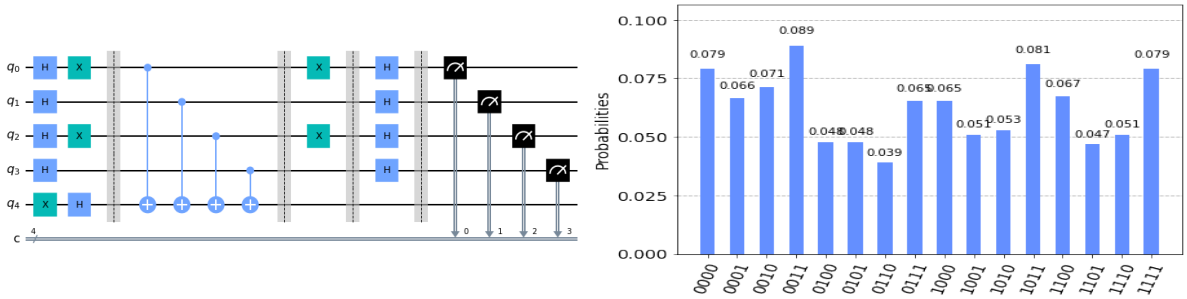
Implementací vyvážené funkce je na druhou stranu mnoho. Jedním ze způsobů, jak zajistit vyváženost funkce, je, že každý qubit v datovém registru použijeme jako kontrolní qubit pro CNOT bránu, kde cílový qubit je vždy onen jediný qubit v cílovém registru. To nám vzhledem k rovnovážné superpozici stavů zajišťuje vyváženost. Měnit působení funkce způsobem, který zachovává její vyváženost, lze s tímto přístupem tak, že na vybrané qubity ještě před aplikací CNOT bran pustíme X brány. Toto samozřejmě musíme následně opětovnou aplikací X bran na tyto qubity zvrátit. Příklad obvodu s vyváženou funkcí se čtyřmi vstupními qubity je na obrázcích (2.8) a (2.9), kde jsme byli vzhledem k nedokonalostem reálného zařízení nuceni použít také simulátor.



Obrázek 2.7: Deutschův algoritmus pro konstantní funkci $f(x) = 1$ s pětiqubitovým datovým registrem. U_f je zde realizována pomocí NOT brány na cílovém qubitu.



Obrázek 2.8: Deutschův algoritmus pro vyváženou funkci s čtyřqubitovým datovým registrem. Pro tento příklad byl vzhledem k příliš velkým nepřesnostem reálného zařízení použit simulátor zobrazující ideální případ, kdy nelze naměřit stav $|0\rangle$ na všech vstupních qubitech.



Obrázek 2.9: Deutschův algoritmus pro vyváženou funkci s čtyřqubitovým datovým registrem. Zde bylo použito reálné zařízení, ovšem vzhledem k nedokonalostem byl nejednou naměřen i stav se všemi vstupními qubity ve stavu $|0\rangle$ a výsledky jsou tedy nereprezentativní.

Kapitola 3

Groverův vyhledávací algoritmus

Deutsch-Joszův algoritmus nám posloužil jako první příklad úlohy, kterou kvantový počítač zvládne vyřešit efektivněji než klasický. Naneštěstí dodnes není znám zásadní příklad úlohy, při které by se nám tato schopnost rozlišit konstantní a vyváženou funkci hodila. V této kapitole však již rozebereme algoritmus s potenciálem reálného využití. Jedná se o kvantový vyhledávací algoritmus také známý jako Groverův algoritmus, jenž byl prvně představen L. K. Groverem roku 1996 [14].

S pomocí [21], [13] a [1] nejprve představím obecný algoritmus, jeho součásti a geometrickou interpretaci. Dále budu diskutovat určení počtu řešení daného problému, popř. samotnou existenci řešení. Nakonec s využitím [18], [27], [2], [25], [24] a [30] budu ilustrovat ekvivalenci Groverova algoritmu s vyhledáváním pomocí kvantové procházky na úplném grafu a na grafu typu hvězda.

3.1 Princip algoritmu

Představme si dva problémy. První nechť je takový, že máme najít nejkratší cestu mezi mnoha městy, kde počet všech možných cest je N. Druhý je najít v databázi o celkovém počtu N prvků prvek s požadovanou vlastností. Oba tyto problémy mohou mít více než jedno řešení a jsou ekvivalentní v tom smyslu, že v klasickém případě je potřeba $O(N)$ (resp. $O(\frac{N}{M})$ pro více možných řešení, kde M je počet řešení) operací na určení řešení. V prvním problému prostě procházíme cesty a udržujeme informaci o tom, která je nejkratší. Ve druhém porovnáváme vlastnost jednotlivých prvků s požadovanou vlastností. My ukážeme, že na kvantovém počítači jsme schopni nalézt požadované řešení již po $O(\sqrt{N})$ (resp. $O(\sqrt{\frac{N}{M}})$) krocích. K tomu využijeme tzv. *zesílení amplitudy*.

K obecné prezentaci algoritmu budeme opět uvažovat oracle se schopností rozpoznat a označit řešení problému, přijde-li jako vstup. Nechť tedy celkový počet prohledávaných prvků je $N = 2^n$ (případně lze vždy doplnit prázdnými vstupy tak, aby se jednalo o mocninu 2) a celkový počet řešení (který v tuto chvíli považujeme za známý) je M, $1 \leq M \leq N$. Tyto prvky si oindexujme indexy 0 až $N - 1$, což znamená, že je můžeme uložit pomocí n bitů. Dále již budeme uvažovat jen tyto indexy, kde tedy pro $x \in N$ platí, že x je řešení $\iff x \in M$ a naopak x není řešení $\iff x \in N \setminus M$. Označení řešení může být reprezentováno funkcí

$$f(x) = \begin{cases} 0, & \text{pro } x \in N \setminus M \\ 1, & \text{pro } x \in M. \end{cases} \tag{3.1}$$

Náš oracle reprezentovaný unitárním operátorem O schopný rozpoznat řešení tedy bude pracovat následovně:

$$|x\rangle\,|q\rangle \xrightarrow{O} |x\rangle\,|q \oplus f(x)\rangle\,, \tag{3.2}$$

kde $|x\rangle$ je stejně jako v Deutsch-Joszově algoritmu datový registr a $|q\rangle$ cílový qubit, do kterého je propsáno působení O. Inspirováni minulou kapitolou se podívejme, jak bude oracle působit, připravíme-li si cílový qubit do stavu $(|0\rangle - |1\rangle)/\sqrt{2}$:

$$O|x\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = (-1)^{f(x)}|x\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right). \tag{3.3}$$

Vzhledem k tomu, že stav cílového qubitu je nezměněn a zůstane nezměněn i během celého procesu, můžeme ho nadále z našich úvah vynechat a akci oracle uvažovat pouze jako

$$O|x\rangle = (-1)^{f(x)}|x\rangle. \tag{3.4}$$

Vidíme tedy, že řešením daného problému oracle posouvá fázi o π.

Podívejme se pro zajímavost, jak by takovýto operátor vypadal, pokud bychom měli celkový prostor osmi prvků a hledané řešení by byl jen jeden prvek s indexem např. 5, v binární soustavě zapsaný jako 101. Tento operátor tedy musí šestému bazickému stavu přidat fázi a ostatní nechat nezměněné. Snadno domyslíme, že takovýto operátor bude mít tvar

$$O_5 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.5}$$

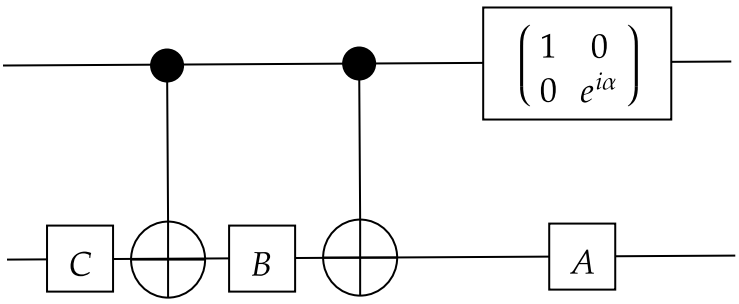
Pozastavme se ještě nad samotným zavedením našeho oracle. Může se totiž zdát, že se zde bavíme o zařízení, které již zná řešení tohoto problému a celá dosavadní diskuze by v tuto chvíli postrádala smysl. Ukazuje se však, že je zásadní rozdíl mezi tím znát řešení a umět řešení rozlišit. Vezměme jako příklad faktorizaci, kdy chceme najít rozklad většího čísla na součin dvou prvočísel. Dostaneme-li číslo, pak podílem původního většího čísla a tohoto čísla snadno zjistíme, zda-li se jedná o námi hledané prvočíslo a pokud ano, tak tím získáme i druhé hledané prvočíslo. Ovšem těchto pokusů o dělení bude úměrně s odmocninou původního velkého čísla (samozřejmě existují mnohem efektivnější algoritmy, toto slouží jen pro ukázku). Klasické dělení je ireverzibilní operace, ovšem jak už jsme si dříve řekli, každou ireverzibilní operaci na klasickém počítači lze efektivně implementovat pomocí reverzibilního obvodu, který lze okamžitě převést na kvantový obvod. Tou zásadní myšlenkou tedy je, že i když neznáme prvočíselné faktory původního čísla, tak dokážeme zkonstruovat oracle, který je rozpozná, přijdou-li mu jako vstup. Navíc, jak si dále ukážeme, s použitím Groverova algoritmu bude potřebný počet dotazů na oracle úměrný pouze čtvrté odmocnině původního čísla. Konkrétní implementací oracle se tedy pro tuto chvíli nebudeme zabývat a několik ukázek naprogramovaných pomocí Qiskit poskytneme později.

Nyní se již zaměřme na jednotlivé kroky celého algoritmu. Budeme tedy operovat na n qubitovém registru, přičemž je možné, že pro implementaci oracle budou potřeba další pracovní qubity. Naším úkolem zjevně bude najít řešení problému za použití co nejmenšího počtu dotazů na oracle. Vycházíme ze standardního počátečního stavu

$$|\psi_0\rangle = \underbrace{|00...0\rangle}_{n}. \tag{3.6}$$

V prvním kroku dostaneme stav Hadamardovou transformací do vyvážené superpozice

$$|\psi_1\rangle = H^{\otimes n}|\psi_0\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle. \tag{3.7}$$



Obrázek 3.1: Způsob implementace libovolné kontrolované unitární operace za pomoci *CNOT* a jednoqubitových bran.

Tento krok lze interpretovat tak, že na začátku netušíme, jaký index je náš hledaný, a tedy každý pokus o uhodnutí tohoto indexu je stejně dobrý. Poté již přichází na řadu fáze *zesílení amplitudy* prováděná pomocí tzv. *Groverových iterací*. Cílem této fáze je při každé této iteraci zvětšit amplitudu u označeného řešení (pomocí oracle), což vzhledem k normalizaci okamžitě implikuje zmenšení amplitud ostatních indexů. Takto po jistém počtu iterací dostaneme řešení problému téměř s jistotou.

Každá Groverova iterace lze popsat pomocí čtyř kroků:

1. Aplikace orale
2. Hadamardova transformace $H^{\otimes n}$
3. Posun fáze o π u každého stavu výpočetní báze kromě stavu $\overbrace{|00...0\rangle}^{n}$, tj. $|x\rangle \rightarrow (-1)^{\delta_{x0}+1} |x\rangle$
4. Hadamardova transformace $H^{\otimes n}$

Implementaci Hadamardovy transformace již známe a ukázali jsme si též, že tato transformace je velmi efektivní. Podívejme se nyní na fázový posun ve třetím kroku.

Jedná se o podmíněnou operaci. My si zde ukážeme, jakým způsobem lze implementovat libovolná kontrolovaná unitární operace c-U, která na jeden qubit působí pouze jako U, a to jen za pomoci jednoqubitových bran a *CNOT* brány. Nejprve aplikujeme fázový posun $\exp\{i\alpha\}$ na cílový qubit kontrolovaný kontrolním qubitem. Matice takového operátoru by měla tvar

$$U_{phase1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\alpha} & 0 \\ 0 & 0 & 0 & e^{i\alpha} \end{pmatrix}. \tag{3.8}$$

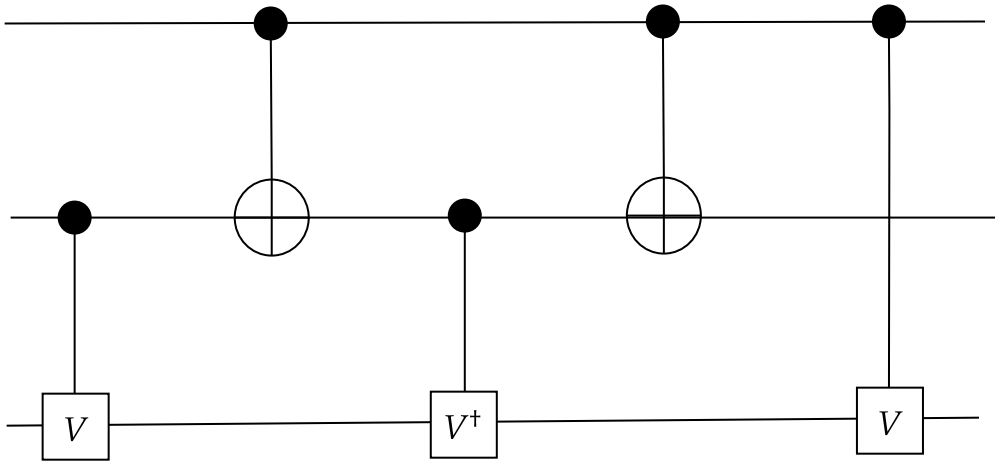
Všimněme si ovšem, že úplně stejného efektu lze dsáhnout aplikací jednoqubitové brány ve tvaru

$$U_{phase2} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix}. \tag{3.9}$$

Nyní využijeme důsledku (1.1) z první kapitoly. Způsob, jakým rozklad z důsledku použijeme, je znázorněn na obrázku (3.1). Vidíme, že je-li první qubit ve stavu $|1\rangle$, pak se aplikuje U na druhý qubit. Je-li naopak první qubit ve stavu $|0\rangle$, pak se vzhledem k vlastnosti $ABC = I$ nic nestane.

Toto lze zobecnit na případ, kdy chceme mít operaci kontrolovanou více qubity, tj pro n kontrolních qubitů chceme aplikovat operaci $C^n(U)$. Působení takové operace lze zapsat jako

$$C^n(U) |c_1 c_2 ... c_n\rangle |\psi\rangle = |c_1 c_2 ... c_n\rangle U^{c_1 c_2 ... c_n} |\psi\rangle, \tag{3.10}$$



Obrázek 3.2: Znázornění implementace $C^2(U)$, kde $V^2 = U$.

kde $c_1c_2...c_n$ v exponentu je součin bitů. $|\psi\rangle$ přitom může být stav jednoho či více qubitů podle toho, jak je U definována. Nechť V je unitární operátor takový, že $V^2 = U$, pak se snadno přesvědčíme, že pro dva kontrolní a jeden cílový qubit lze $C^2(U)$ implementovat tak, jak je znázorněno na obrázku (3.2).

Všimněme si, že při volbě $V \equiv \frac{(1-i)(I+iX)}{2}$ dostaneme $C^2(X)$, neboli Toffoliho bránu. S tou už můžeme implementovat libovolnou $C^n(U)$ následovně. K n kontrolním qubitům vezmene $n-1$ pracovních qubitů ve stavu $|0\rangle$. Nejdříve aplikujeme Toffoliho bránu na první pracovní qubit, kde první dva kontrolní qubity poslouží jako kontrolní. Tím vlastně dostaneme první pracovní qubit do stavu $|c_1 \cdot c_2\rangle$. Následně použijeme Toffoliho bránu na druhý pracovní qubit, přičemž jako kontrolní qubity použijeme třetí kontrolní a první pracovní. Takto dostaneme druhý pracovní qubit do stavu $|c_1 \cdot c_2 \cdot c_3\rangle$. Takto postupujeme dále, až poslední pracovní qubit dostaneme do stavu $|c_1 \cdot c_2 \cdot ... \cdot c_n\rangle$. Nakonec vykonáme kontrolovanou U operaci s posledním pracovním qubitem jako kontrolním. K dovršení stačí už jen reverzně aplikovat Toffoliho brány tak, abychom pracovní qubity vrátili do stavu $|0\rangle$.

Vraťme se k našemu konkrétnímu případu, kde chceme každému bazickému stavu kromě $|00...0\rangle$ přidat fázi (-1). Na to nám stačí si uvědomit dvě skutečnosti. Určitě vhodnou branou místo obecné brány U bude pro tento fázový posun brána Z. Druhou věcí je, že stejně tak jako můžeme kontrolovat podle stavu $|1\rangle$, tak můžeme kontrolovat podle stavu $|0\rangle$. Stačí před kontrolovanou operací aplikovat na kontrolní qubit bránu X a poté její opětovnou aplikací její působení zvrátit. Zde požadované operace tedy dosáhneme tak, že nejprve na všechny kontrolní qubity aplikujeme X bránu, poté využijeme předešlého procesu, kde těsně před použitím kontrolované Z aplikujeme na poslední pracovní qubit X bránu a nakonec účinky Toffoliho bran a X bran na pracovních a kontrolních qubitech reverzně zvrátíme.

Podívejme se nyní, jak bude vypadat operátor takového fázového posunu. Snadno se přesvědčíme, že musí jít o matici ve tvaru $diag(1, -1, ..., -1)$, která lze zapsat jako $(2\,|00...0\rangle\langle 00...0| - I)$. S tímto vyjádřením a s uvědoměním, že $(H^{\otimes n})^{-1} = (H^{\otimes n})^{\dagger} = H^{\otimes n}$, pak můžeme kroky 2, 3 a 4 souhrnně přepsat jako

$$\begin{aligned} H^{\otimes n}(2\,|00...0\rangle\langle 00...0| - I)H^{\otimes n} &= 2(H^{\otimes n})^{\dagger}\,|00...0\rangle\langle 00...0|\,H^{\otimes n} - H^{\otimes n}IH^{\otimes n} \\ &= 2\,|\psi\rangle\langle\psi| - I, \end{aligned} \tag{3.11}$$

kde $|\psi\rangle$ je vyvážená superpozice všech bazických stavů. Celkově tedy dostáváme Groverovu iteraci jako $G = (2\,|\psi\rangle\langle\psi| - I)O$.

3.2 Geometrická interpretace

Předchozí kapitola nám sice vysvětlila postup a možnosti realizace, avšak neposkytla nám příliš velkou intuici ohledně fungování algoritmu. Dále si detailně vysvětlíme, co se odehrává při Groverově iteraci. Podívejme se nejprve, jak působí člen $(2\,|\psi\rangle\langle\psi| - I)$ na obecný stav:

$$\begin{aligned}(2\,|\psi\rangle\langle\psi| - I)\sum_k \alpha_k\,|k\rangle &= \left(\frac{2}{N}\sum_x |x\rangle\sum_y \langle y| - I\right)\sum_k \alpha_k\,|k\rangle \\ &= \frac{2}{N}\sum_x |x\rangle\sum_k \alpha_k - \sum_k \alpha_k\,|k\rangle \\ &= 2\langle\alpha\rangle\sum_x |x\rangle - \sum_k \alpha_k\,|k\rangle \\ &= \sum_k (2\langle\alpha\rangle - \alpha_k)\,|k\rangle\,.\end{aligned} \tag{3.12}$$

Zde jsme v poslední rovnosti jen přejmenovali sčítací index a $\langle\alpha\rangle$ značí střední hodnotu koeficientů α_k. Z toho je zřejmé, proč se $2\,|\psi\rangle\langle\psi| - I$ také říká operace inverze kolem střední hodnoty.

S připomenutím, že N je celkový počet indexů a M je počet indexů představujících řešení, zaveďme dva normalizované stavy

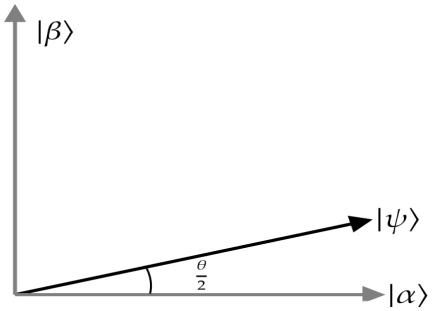
$$|\alpha\rangle \equiv \frac{1}{\sqrt{N-M}}\sum_p |p\rangle \tag{3.13}$$

$$|\beta\rangle \equiv \frac{1}{\sqrt{M}}\sum_s |s\rangle\,, \tag{3.14}$$

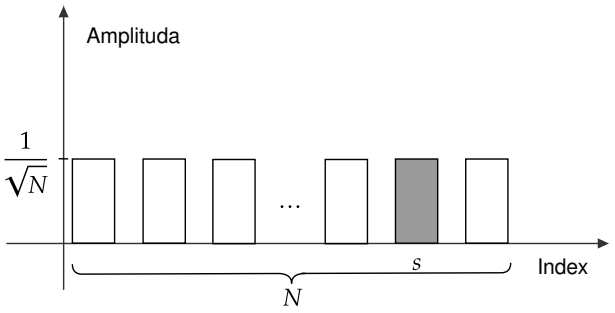
kde p prochází přes všechny bazické stavy nepředstavující řešení a s naopak přes všechny stavy řešící problém. Jinými slovy je $|\beta\rangle$ vyvážená superpozice všech řešení a naopak $|\alpha\rangle$ vyvážená superpozice všech stavů, které problém neřeší. Naši původní rovnovážnou superpozici všech bazických stavů lze v řeči těchto stavů zapsat jako

$$|\psi\rangle = \sqrt{\frac{N-M}{N}}\,|\alpha\rangle + \sqrt{\frac{M}{N}}\,|\beta\rangle\,. \tag{3.15}$$

Toto nám dává vyjádření superpozice všech stavů v dvourozměrném prostoru tvořeném lineárním obalem dvou ortonormálních stavů představujících řešení či neřešení problému. Dá se nahlédnout, že v souladu s přirozenou intuicí platí, že čím máme více řešení (tj. čím větší M), tím více bude posílena amplituda u řešení $|\beta\rangle$. Jsme ve 2D, tedy určitě můžeme provést identifikaci $|\psi\rangle \equiv \cos\theta/2\,|\alpha\rangle + \sin\theta/2\,|\beta\rangle$, a tedy $\cos\theta/2 = \sqrt{(N-M)/N}$ a $\sin\theta/2 = \sqrt{M/N}$. Graficky je toto znázorněné na obrázku (3.3a) a na obrázku (3.3b) jsou amplitudy vyváženého stavu, jejichž vývoj bude dále velmi důležitý.



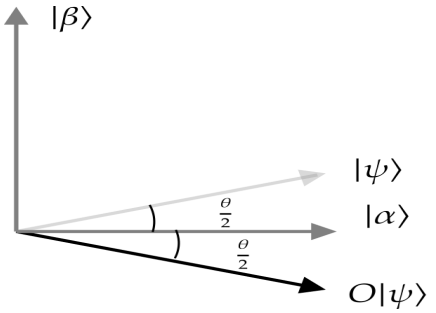
(a) Rovnovážná superpozice $|\psi\rangle$ v rovině $|\alpha\rangle, |\beta\rangle$



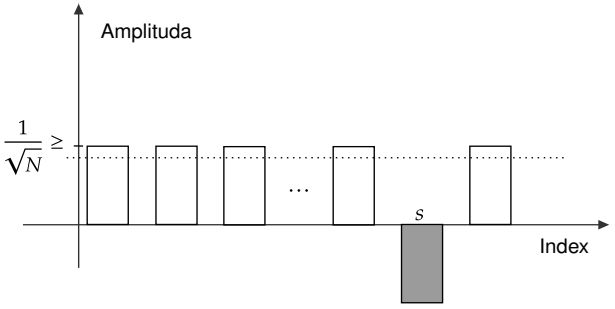
(b) Amplitudy vyváženého stavu $|\psi\rangle$. s znázorňuje index řešení, přičemž zde pro jednoduchost uvažujeme jen jedno řešení.

Obrázek 3.3

Aplikujme nyní Groverovu iteraci $G = (2|\psi\rangle\langle\psi| - I)O$. Nejprve sledujme působení oracle O. O přidává fázi stavům, které představují řešení, což jsou ty stavy, které v naší rovině udávají průmět do osy určené stavem $|\beta\rangle$. Působení O je tedy zrcadlení okolo stavu $|\alpha\rangle$ znázorněné na obrázku (3.4).



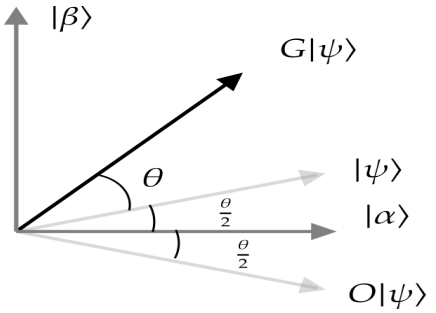
(a) Působení O má význam zrcadlení okolo stavu $|\alpha\rangle$.



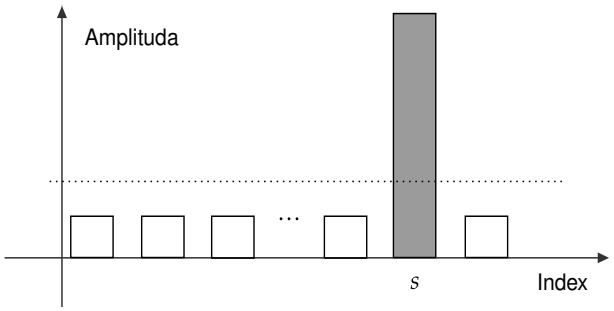
(b) Amplituda řešení (zde opět uvažuji jen jedno) dostane fázový posun (-1). Tečkovaná čára představuje nový průměr amplitud.

Obrázek 3.4

K dokončení iterace zbývá aplikovat $2|\psi\rangle\langle\psi| - I$. To je zrcadlení podle rovnovážného stavu $|\psi\rangle$. S tím svírá stav $O|\psi\rangle$ úhel θ. Vidíme tedy, že složením těchto dvou zrcadlení získáváme rotaci o úhel θ vstříc stavu $|\beta\rangle$, což je znázorněno na obrázku (3.5a). Abychom viděli, co operace $2|\psi\rangle\langle\psi| - I$ dělá s amplitudou, využijeme vztahu (3.12) odvozeného na začátku kapitoly. Jak je znázorněno na obrázku (3.4b), aplikování O nám snížilo celkový průměr amplitud vyznačený tečkovanou čarou, okolo které my teď zrcadlíme amplitudu našeho řešícího stavu. Výsledek je znázorněn na obrázku (3.5b).



(a) Celkové působení Groverovy iterace G.



(b) Aplikací $2|\psi\rangle\langle\psi| - I$ se výrazně zvětšila amplituda u stavu, který představuje řešení.

Obrázek 3.5

Z obrázku (3.5a) nám plynou dvě skutečnosti. Jednou je, že opakovanou aplikací $G^k|\psi\rangle$ zůstaneme v rovině určené lineárním obalem stavů $|\alpha\rangle$ a $|\beta\rangle$ (lineární obal stavů $|\alpha\rangle$, $|\beta\rangle$ je invariantní podprostor Groverovy iterace G). Druhou je již zmíněný úhel rotace θ a z něho plynoucí vztah

$$G|\psi\rangle = \cos\frac{3}{2}\theta|\alpha\rangle + \sin\frac{3}{2}\theta|\beta\rangle\,. \tag{3.16}$$

Takto opakovanou aplikací G získáme

$$G^k|\psi\rangle = \cos\frac{2k+1}{2}\theta|\alpha\rangle + \sin\frac{2k+1}{2}\theta|\beta\rangle\,. \tag{3.17}$$

3.3 Složitost

Ukázali jsme si možnost realizace a princip fungování Groverova algoritmu. Nyní je potřeba ukázat onu slíbenou redukci složitosti na $O(\sqrt{N})$ (resp. na $O\left(\sqrt{\frac{N}{M}}\right)$ pro více řešení). Zjevně naším cílem bude postupně dorotovat stav $|\psi\rangle$ až ke stavu $|\beta\rangle$. Připomeňme, že počáteční stav je ve tvaru (3.15), kde amplituda u stavu $|\beta\rangle$ je $\sin\theta/2 = \sqrt{M/N}$. Úhel, o který potřebujeme otočit $|\psi\rangle$, je tedy

$$\frac{\pi}{2} - \frac{\theta}{2} = \frac{\pi}{2} - \arcsin\sqrt{\frac{M}{N}} = \arccos\sqrt{\frac{M}{N}}. \tag{3.18}$$

To nám již dává počet iterací potřebný pro přiblížení se $|\beta\rangle$:

$$K = \mathrm{CI}\left(\frac{\arccos\sqrt{M/N}}{\theta}\right) = \mathrm{CI}\left(\frac{\pi}{2\theta} - \frac{1}{2}\right), \tag{3.19}$$

kde $\mathrm{CI}(x)$ značí nejbližší celé číslo číslu x, přičemž poloviny zde zaokrouhlujeme dolů. Takto bude odklon našeho stavu od stavu $|\beta\rangle$ menší nebo roven $\frac{\pi}{4}$, tudíž řešení problému naměříme s pravděpodobností větší nebo rovnou $\frac{1}{2}$. Stejného výsledku bychom se možná elegantnějším způsobem dobrali, pokud bychom si uvědomili, že ve vyjádření (3.17) chceme maximalizovat amplitudu u stavu $|\beta\rangle$. To nám dává vztah

$$\frac{(2k+1)\theta}{2} = \frac{\pi}{2}, \tag{3.20}$$

a tedy

$$K = \mathrm{CI}(k) = \mathrm{CI}\left(\frac{\pi}{2\theta} - \frac{1}{2}\right). \tag{3.21}$$

Z rovnice (3.19) je vidět, že K závisí na počtu řešení M, jehož znalost předpokládáme. Později si ukážeme, že se můžeme zbavit i tohoto požadavku na znalost počtu řešení.

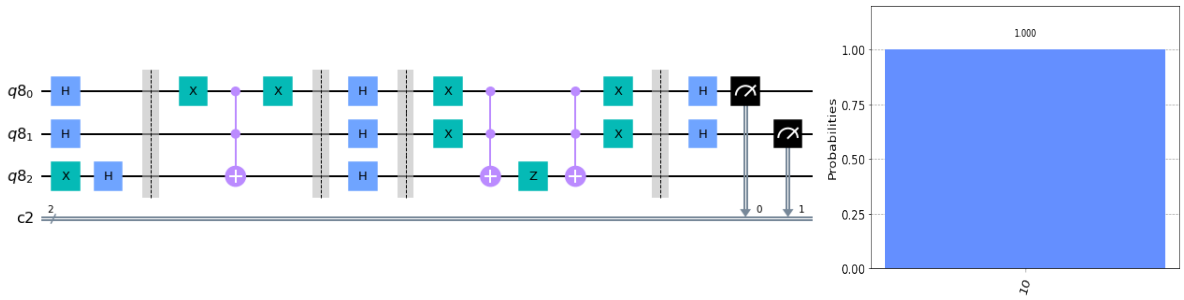
Zkusme nyní odvodit snazší vzorec vystihující chování K. Je zřejmé, že $K \leq \lceil \frac{\pi}{2\theta} \rceil$. Udělejme nyní vcelku přirozený předpoklad $M \leq \frac{N}{2}$. Potom

$$\frac{\theta}{2} \geq \sin\frac{\theta}{2} = \sqrt{\frac{M}{N}}, \tag{3.22}$$

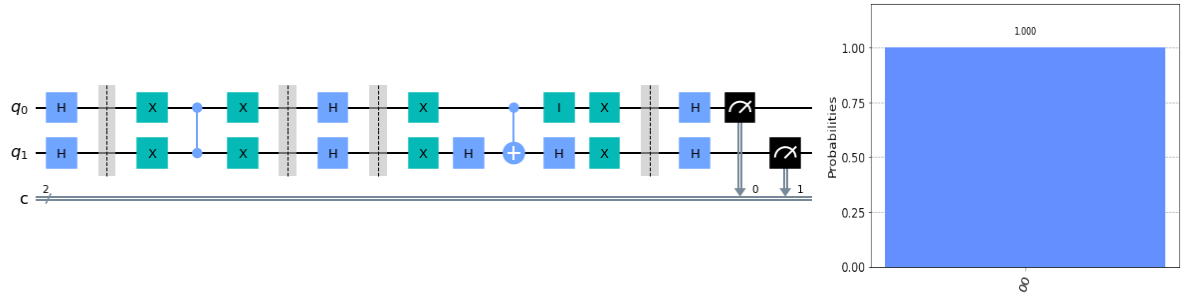
z čehož okamžitě plyne horní hranice pro počet iterací

$$K = \left\lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rceil. \tag{3.23}$$

Je-li $M \geq \frac{N}{2}$, pak nám proti přirozené intuici stoupá počet potřebných iterací. Na to lze nahlížet dvěma způsoby. V prvním případě zkusím náhodně vybrat jeden index a s pravděpodobností $\geq \frac{1}{2}$ to bude řešení. Druhý přístup je, že původní soubor rozšíříme o N položek, která nejsou řešení. To vyžaduje přidání jednoho qubitu. Potom však určitě platí $M \leq 2N$ a můžeme použít předešlý postup. Celkově tedy můžeme kapitolu uzavřít s tím, že potřebný počet použití oracle je $O(\sqrt{\frac{N}{M}})$, což je kvadratické zlepšení oproti klasickému $O(\frac{N}{M})$. Dva možné způsoby implementace Groverova algoritmu pro jedno řešení jsou na obrázcích (3.6) a (3.7).



Obrázek 3.6: Obvod realizující Groverův algoritmus pro $N = 4$ a s oracle označujícím řešení $|01\rangle$ (vzpomeňme, že Qiskit má obrácené pořadí bitů).



Obrázek 3.7: Obvod realizující Groverův algoritmus pro $N = 4$ s pomocí pouze dvou qubitů a s oracle označujícím řešení $|00\rangle$.

3.4 Určení počtu řešení

V předchozích kapitolách jsme počet řešení M pokládali za daný a např. odvození počtu potřebných Groverových iterací na M záviselo. Ne vždy je ovšem počet řešení předem jasný. V této kapitole si ukážeme metodu zvanou *kvantové počítání*, díky které jsme schopni odhadnout počet řešení rychleji, než je tomu možné na klasickém počítači. Budeme-li tedy schopni efektivně určit počet řešení, pak budeme schopni dané řešení také efektivně najít. Navíc nám také umožní rozhodnout o samotné existenci řešení. K tomu, abychom byli schopni tuto metodu použít, ovšem potřebujeme tzv. *algoritmus pro odhad fáze*, který je detailněji popsaný v následující kapitole 4 - *Shorův algoritmus*. Budeme se tedy odvolávat na důsledky odvozené později.

Kvantové počítání využívá algoritmus pro odhad fáze k určení vlastních hodnot Groverovy iterace G. Vzpomeňme, že lineární obal stavů $|\alpha\rangle$ a $|\beta\rangle$ tvoří invariantní podprostor G. Zaveďme operátor G_{ef}, což je restrikce Groverovy iterace na tento podprostor. Vzhledem k tomu, že Groverova iterace je rotace v této rovině o úhel θ, bude G_{ef} reprezentovat matice 2×2 tvaru

$$G_{ef} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{3.24}$$

Označme $|a\rangle$, $|b\rangle$ vlastní vektory G_{ef}. Ty budou určitě ležet v tomto podprostoru. Spočítejme nyní vlastní čísla tohoto operátoru.

$$\begin{aligned} & \det\begin{pmatrix} \cos\theta - \lambda & -\sin\theta \\ \sin\theta & \cos\theta - \lambda \end{pmatrix} = 0 \\ \iff & \cos^2\theta - 2\lambda\cos\theta + \lambda^2 + \sin^2\theta = 0 \\ \iff & (\lambda - \cos\theta)^2 + \sin^2\theta = 0 \\ \iff & \lambda = \cos\theta \pm i\sin\theta = e^{\pm i\theta}, \end{aligned} \tag{3.25}$$

což dává vlastní stav

$$|a\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -i \end{pmatrix} \tag{3.26}$$

k vlastnímu číslu $e^{i\theta}$ a vlastní stav

$$|b\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ i \end{pmatrix} \tag{3.27}$$

k vlastnímu číslu $e^{-i\theta}$. Pro jednoduchost dále předpokládejme rozšířený vyhledávací prostor na $2N$ prvků ze závěru předešlé kapitoly.

Přikročme nyní k odhadu fáze. Vzhledem k tomu, že detailní popis tohoto algoritmu je v kapitole (4.2), tak raději než přesný způsob realizace zde zmíníme nároky a velikost chyby. Naším cílem bude určit θ s přesností na m bitů s pravděpodobností úspěchu přinejmenším $1 - \epsilon$. Na to bude potřeba celkově registr s $t + n + 1$ qubity, kde $t \equiv m + \lceil \log(2 + 1/2\epsilon) \rceil$ (viz (4.20)). Prvních t qubitů (registr 1) je potřeba pro odhad fáze a $n + 1$ (registr 2) pro uskutečnění samotné Groverovy iterace. Na začátku dostaneme oba registry do vyvážené superpozice pomocí Hadamardovy transformace. Poté provádíme příslušný počet Groverových iterací postupně kontrolovaných všemi qubity v prvním registru. Nakonec aplikujeme *inverzní kvantovou Fourierovu tranformaci* (kvantová Fourierova transformace je detailně popsaná v kapitole (4.1)) na první registr. Jak si později ukážeme, měřením prvního registru získáme θ resp. $2\pi - \theta$ s přesností $|\Delta\theta| \leq 2^{-m}$ a s pravděpodobností nejméně $1 - \epsilon$. Snadno pak již z rovnice

$$\sin^2(\frac{\theta}{2}) = \frac{M}{2N} \tag{3.28}$$

dostaneme odhad pro počet řešení M.

Pro účely výsledné implementace je nutno zmínit, že na konci algoritmu pro odhad fáze měříme *hodnotu* rovnou $2^t\varphi$ pro vlastní hodnotu $e^{2\pi i\varphi}$. Naše θ tedy bude rovna

$$\theta = hodnota \cdot \frac{2\pi}{2^t}. \tag{3.29}$$

Zabývejme se nyní chybou ΔM při tomto odhadu. Zjevně

$$\frac{|\Delta M|}{2N} = \left|\sin^2\left(\frac{\theta+\Delta\theta}{2}\right) - \sin^2\left(\frac{\theta}{2}\right)\right| = \left(\sin\left(\frac{\theta+\Delta\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)\right)\left|\sin\left(\frac{\theta+\Delta\theta}{2}\right) - \sin\left(\frac{\theta}{2}\right)\right| \tag{3.30}$$

a uvědomíme-li si, že jednak

$$\left|\sin\left(\frac{\theta+\Delta\theta}{2}\right) - \sin\left(\frac{\theta}{2}\right)\right| \leq \left|\frac{\Delta\theta}{2}\right| \tag{3.31}$$

a jednak

$$\left|\sin\left(\frac{\theta+\Delta\theta}{2}\right)\right| < \sin\left(\frac{\theta}{2}\right) + \left|\frac{\Delta\theta}{2}\right|, \tag{3.32}$$

potom

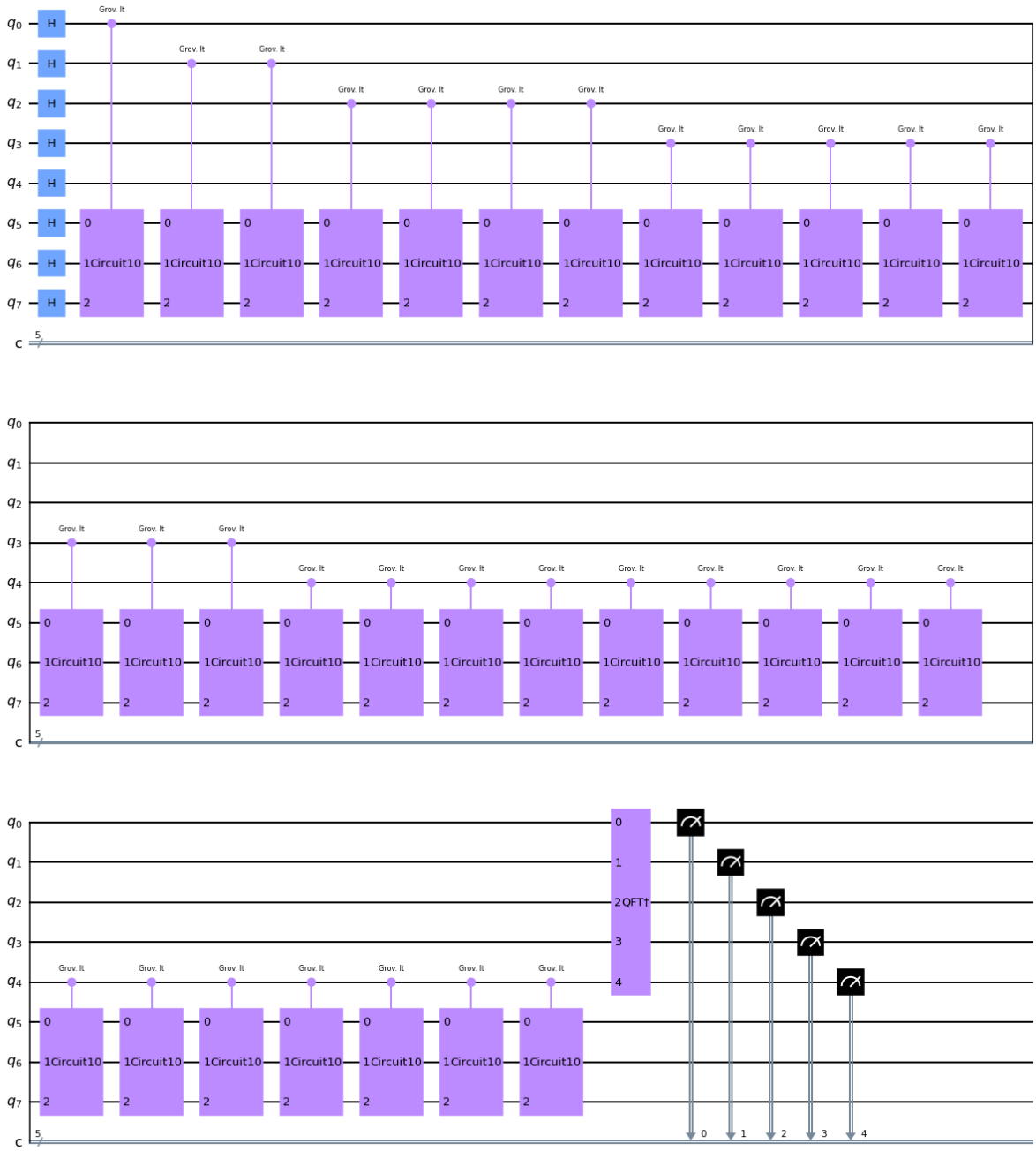
$$\left|\frac{\Delta M}{2N}\right| < \left(2\sin\left(\frac{\theta}{2}\right) + \left|\frac{\Delta\theta}{2}\right|\right)\left|\frac{\Delta\theta}{2}\right|. \tag{3.33}$$

S využitím rovnosti (3.28) a vztahu $|\Delta\theta| \leq 2^{-m}$ pak již dostáváme odhad

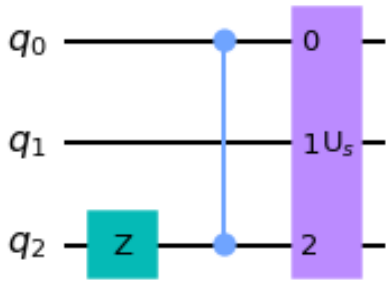
$$|\Delta M| < \left(\sqrt{2MN} + \frac{N}{2^{m+1}}\right)2^{-m}. \tag{3.34}$$

Volbou např. $m = \lceil n/2\rceil + 1$ a $\epsilon = 1/6$ se můžeme se znalostí fungování algoritmu pro odhad fáze přesvědčit, že počet potřebných Groverových iterací (a tedy dotazů na oracle) je $\Theta(\sqrt{N})$, kde $\Theta(g(n))$ znamená stejné asymptotické chování jako nějaká funkce $g(n)$.

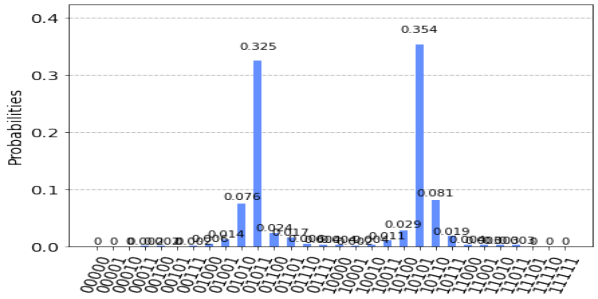
Viděli jsme tedy, že kvantové počítání slouží jednak k samotnému oveření existence řešení a poté k určení počtu těchto řešení. Znalost toho počtu je pak zásadní k určení počtu Groverových iterací v Groverově algoritmu. Ukázka implementace s oracle označující dvě řešení z prostoru osmi prvků je na obrázcích (3.8) a (3.9).



Obrázek 3.8: Ukázka celkového algoritmu pro $t = 5$ a $n = 3$. Vzpomeňme, že díky obrácenému pořadí qubitů v Qiskit bychom museli obrátit pořadí qubitů v prvním registru, abychom dostali později přesně odvozený algoritmus pro odhad fáze. Jednotlivé Groverovy iterace jsou rozebrány na obrázku (3.9a).



(a) Detail Groverovy iterace. U_s zde implementuje kroky 2), 3), 4) z kapitoly popisující obecně Groverovu iteraci. Je důležité zmínit, že jsme zde použili obvyklý postup, a to implementaci ve skutečnosti $-U_s$. Ta při Groverově algoritmu přidává bezvýznamnou globální fázi, ovšem zde je Groverova iterace kontrolovaná, takže význam blíže vysvětlený u obr. (3.9b) mít bude. Oracle zde označuje dvě řešení - $|001\rangle$ a $|011\rangle$.



(b) Výsledky měření kvantového počítání. Vidíme, že dvě hodnoty mají znatelně nejvyšší zastoupení - 11 a 21. Ty odpovídají $e^{i\theta}$ a $e^{-i\theta}$. θ pak získáme s pomocí rovnice (3.29) a M s pomocí rovnice (3.28). Ovšem jak je zmíněno na obr. (3.9a), U_s je implementována jako $-U_s$, a tedy jsme vlastně prohledávali přes stavy, které nepředstavují řešení. Výsledný počet řešení je pak tedy $N - M \approx 1.8$, což odpovídá našemu oracle označující dvě řešení.

Obrázek 3.9

3.5 Vyhledávání pomocí kvantových procházek

V této části si vysvětlíme základní myšlenky a vlastnosti kvantových procházek. Následně ukážeme ekvivalenci Groverova algoritmu a kvantové procházky na úplném grafu. Nakonec pro určitost použijeme k ukázání ekvivalence jiný typ grafu, konkrétně graf typu hvězda.

3.5.1 Kvantová procházka

Představme si tedy v rychlém shrnutí koncept kvantových procházek, přičemž nejlepší bude začít s tzv. *klasickou náhodnou procházkou*. Ta se nejčastěji prezentuje na modelu opilce, který pro jednoduchost může chodit pouze po přímce. Ten ujde 1 krok délky a za τ sekund, přičemž pravděpodobnost úkroku doprava i doleva je $\frac{1}{2}$. Jaká bude pravděpodobnost nalezení opilce v pozici x po čase t? Za tuto dobu vykoná opilec $n = \frac{t}{\tau}$ kroků. Vzhledem k tomu, že v každé pozici má opilec dvě možnosti, kam se vydat, bude celkový počet možných trajektorií roven 2^n. Nás pak zajímají ty trajektorie, které vedou do bodu x. Ty jsou dány kombinací s kroků doprava a $n - s$ kroků doleva tak, že $x = as - a(n - s)$, kterých je $\binom{n}{s}$. Pravděpodobnost je pak dána

$$P(x,t) = \frac{1}{2^n}\binom{n}{s} = \frac{1}{2^{\frac{t}{\tau}}}\binom{\frac{t}{\tau}}{\frac{t}{2\tau} + \frac{x}{2a}}, \tag{3.35}$$

což lze pro velký počet kroků aproximovat Gaussovským rozdělením

$$P(x,t) = \sqrt{\frac{2\tau}{\pi t}}\exp\left(\frac{-\tau x^2}{2a^2 t}\right). \tag{3.36}$$

Z toho vidíme, že střední kvadratická odchylka

$$\sigma = \frac{a}{\sqrt{\tau}}\sqrt{t} \tag{3.37}$$

je úměrná $\sqrt{t}$.

Přejděme nyní ke kvantové analogii tohoto procesu. Kvantová částice se při každém „kroku"nerozhoduje, zda-li půjde doleva, či doprava, nýbrž se díky vlnovým vlastnostem dostane do interferenci podléhajícího stavu superpozice. Nechť tedy pro $v \in \mathbb{Z}$ označuje $|v\rangle$ částici na pozici v. Obecný stav pak zapíšeme jako

$$|\psi\rangle = \sum_v \alpha_v |v\rangle . \tag{3.38}$$

Částice začíná na pozici 0 a v každém kroku se může dostat do pozice $v-1$ resp. $v+1$ se stejnou pravděpodobností. Transformace má tedy tvar

$$|v\rangle \to \frac{|v-1\rangle + |v+1\rangle}{\sqrt{2}}. \tag{3.39}$$

Takto ovšem vývoj zavést nemůžeme. Snadno se přesvědčíme, že již při druhém kroku by byl součet pravděpodobností větší než 1. Trochu obecněji se na to můžeme nahlédnout tak, že operátor popisující vývoj musí být unitární. Představme si, že by tedy působil uvedeným (ačkoli vzhledem k amplitudám obecnějším) způsobem

$$|v\rangle \to \alpha |v-1\rangle + \beta |v+1\rangle \tag{3.40}$$

pro nějaké komplexní amplitudy α, β. Pak stav $|v-1\rangle$ posune do stavu $\alpha|v-2\rangle + \beta|v\rangle$ a stav $|v+1\rangle$ do stavu $\alpha|v\rangle + \beta|v+2\rangle$. Tyto původně ortogonální stavy by po působení libovolného unitárního operátoru měly zůstat ortogonální. To zde ovšem nastane pouze pokud jeden z koeficientů α, β je roven 0. Takový vývoj by pak pro naše účely neměl valný význam.

Tento problém lze vyřešit přidáním dalšího stupně volnosti, tzv. *prostor mince*. Ten reprezentuje nějaký vnitřní stav částice a určuje, jakým směrem se částice vydá. Bude určen dvěma stavy, nazvěme je $|R\rangle$ a $|L\rangle$. Celkový Hilbertův prostor pak bude $\mathcal{H} = \mathcal{H}_P \otimes \mathcal{H}_C$, kde $\mathcal{H}_P$ určuje pozici částice a $\mathcal{H}_C$ je onen prostor mince. Obecný stav částice popisované těmito dvěma veličinami bude

$$|\psi\rangle = \sum_v (\psi_{v,L}|v,L\rangle + \psi_{v,R}|v,R\rangle). \tag{3.41}$$

Zaměřme se nyní na vývoj částice. Každý krok bude vyjádřen evolučním operátorem U sestávajícím ze dvou transformací

$$U = S(I_p \otimes C). \tag{3.42}$$

Zde $I_p \otimes C$ je operátor působící netriviálně pouze na prostoru mince a S působí na celý Hilbertův prostor. Operátor C je přitom přirozené zavést tak, aby

$$|v,L\rangle \to \frac{|v,L\rangle + |v,R\rangle}{\sqrt{2}} \qquad |v,R\rangle \to \frac{|v,L\rangle - |v,R\rangle}{\sqrt{2}}, \tag{3.43}$$

čímž definujeme tzv. Hadmardovu minci $C \equiv H$. Translační operátor S následně po hodu mincí vykoná posun pozice podmíněný stavem mince způsobem

$$|v,L\rangle \to |v-1,L\rangle \qquad |v,R\rangle \to |v+1,R\rangle . \tag{3.44}$$

Takto definovaný operátor můžeme zapsat jako

$$S = \sum_v (|v-1\rangle\langle v| \otimes |L\rangle\langle L| + |v+1\rangle\langle v| \otimes |R\rangle\langle R|) \tag{3.45}$$

Zvolme nyní počáteční stav $|0, L\rangle$ a udělejme tři kroky procházky. Dostaneme tak stav

$$U^3 |0, L\rangle = \frac{1}{\sqrt{8}}(|-3, L\rangle + 2|-1, L\rangle + |-1, R\rangle - |-1, L\rangle + |3, R\rangle), \tag{3.46}$$

na kterém naměříme pravděpodobnosti

$$P(-3) = \frac{1}{8} \qquad P(-1) = \frac{5}{8} \qquad P(1) = \frac{1}{8} \qquad P(3) = \frac{1}{8}. \tag{3.47}$$

To je rozdílné od klasické náhodné procházky, kdy by byl poměr pravděpodobností symetrický podle středu 1 : 3 : 3 : 1. Vidíme zde asymetrii, která je při více krocích čím dál zřetelnější. Je to důsledek stavu mince na počátku. Lze ukázat, že následkem toho je, že střední kvadratická odchylka roste úměrně s počtem kroků (tedy s časem), tj. $\sigma \sim t$. To vlastně znamená kvadraticky rychlejší šíření částice v prostředí – zde přímka.

Zobecněním takovéto procházky již získáme potenciální využití tohoto přístupu. Jedna z vůbec nejdůležitějších aplikací je pro vyhledávání v nesetříděné databázi. Princip superpozice zde může být využit k řízení pohybu podle více výsledků naráz a interference pomáhá k tomu, aby se pohyb nezdržoval v okolí počáteční pozice.

3.5.2 Groverův algoritmus jako kvantová procházka na úplném grafu

Jak jsme jsme již uvedli, kvantová procházka může být využita k vyhledávání v databázi. Oproti Groverovu algoritmu mají kvantové procházky tu výhodu, že databáze zde může být reprezentována různými grafy, které mohou být snadněji implementovány. Na takovém grafu odpovídá každý vrchol položce v databázi a hledaná položka je reprezentována označeným vrcholem. Grafy mohou být různého typu a podle toho se budou lišit užité evoluční operátory. Povede-li například z daného vrcholu N hran, pak by prostor mince na tomto vrcholu měl mít N bazických stavů. Naším cílem samozřejmě bude, až po K krocích změříme pozici „chodce", aby s velkou pravděpodobností ukazoval na označený vrchol představující řešení. Ukazuje se, že optimálnost vyhledávání pomocí kvantové procházky je velice silně spjata se strukturou grafu. Bylo dokázáno, že optimálnosti (tj. nalezení vrcholu v $O(\sqrt{N})$ krocích) lze dosáhnout např. na hyperkrychli nebo na mřížce dimenze větší než 2.

Mluvme nyní formálněji. Mějme graf $G = (V, E)$, kde každý vrchol $v \in V$ ukládá proměnnou $a_v \in \{0, 1\}$. Zde 1 označuje hledané řešení, tj. označený vrchol a 0 neoznačený vrchol. Celkový Hilbertův prostor evolučního operátoru pro zcela obecný graf bude

$$\mathcal{H} = \bigoplus_{v \in V} \mathcal{H}_v, \quad \mathcal{H}_v = [|v, u\rangle \,|(v, u) \in E]_\lambda \,.$$

To lze chápat tak, že každé orientované hraně (v, u) (v je počáteční vrchol, u koncový) přiřadím bazický stav. První vrchol určuje pozici chodce, druhý kam směřuje. $\mathcal{H}_v$ chápu jako lokální Hilbertův prostor ve vrcholu v. Obecný evoluční operátor procházky je tvaru

$$U = S \cdot C,$$

kde minci reprezentuje operátor tvaru

$$C = \bigoplus_{v \in V} C_v.$$

Zde C_v je unitární operátor na $\mathcal{H}_v$, tj. nemění první index v ketu.

Operátor posunutí S lze volit různě. Jedna z možností, která funguje na libovolném grafu, je „flip-flop"

$$S = \sum_{(v,u)\in E} |u,v\rangle\langle v,u| .$$

To odpovídá dynamice, kdy částice skočí z vrcholu v do vrcholu u a současně se "otočí stav mince", tj. částice bude směřovat zpět do vrcholu v, ze kterého přišla.

Uvažujme dále neorientovaný *d-regulární* graf G s N vrcholy. Pro d-regulární graf je dimenze všech lokálních prostorů stejná a rovna d. Očíslováním sousedních vrcholů v jako u_i, $i = 1,\ldots,d$ pak můžeme identifikovat bazické stavy způsobem

$$|v,u_i\rangle \equiv |v\rangle \otimes |i\rangle ,$$

kde kety $|i\rangle$ budou představovat bazické stavy mince. Pro d-regulární graf pak bude platit

$$\mathcal{H} = \bigoplus_{v\in V} \mathcal{H}_v \equiv \mathcal{H}_p \otimes \mathcal{H}_c,$$

kde prostor mince bude $\mathcal{H}_c = [|1\rangle_c , ..., |d\rangle_c]_\lambda$ a prostor vrcholů $\mathcal{H}_p = [|1\rangle_p , ..., |N\rangle_p]_\lambda$.

Náš algoritmus bude složen z K unitárních operací

$$\left|\psi_{final}\right\rangle = U_K U_{K-1} ... U_1 |\psi_{start}\rangle , \tag{3.48}$$

kde každá U_i je buď dotaz, nebo lokální transformace. Zde $|\psi_{start}\rangle$ je nějaký fixní počáteční stav. My budeme uvažovat tzv. Z-lokální transformace. Transformace je Z-lokální, když pro libovolné $v \in V$ a $|\psi\rangle \in \mathcal{H}_c$ je stav $U_i(|v\rangle \otimes |\psi\rangle)$ z podprostoru $\mathcal{H}_{\Gamma(v)} \otimes \mathcal{H}_c$, kde $\mathcal{H}_{\Gamma(v)} \subset\subset \mathcal{H}_p$ je dán lineárním obalem stavu $|v\rangle$ a stavů $|v_{ad}\rangle$ reprezentující vrcholy sousedící s v. Algoritmus skončil úspěšně, pokud na stavu $\left|\psi_{final}\right\rangle$ naměříme na části odpovídající prostoru $\mathcal{H}_p$ stav $|m\rangle$ takový, že $a_m = 1$.

Kvantová procházka bude tedy opět složena z evolučních operátorů, kde tentokrát

$$S\,|v,i\rangle = |u_i, \pi(i)\rangle . \tag{3.49}$$

Zde $i = 1, ..., d$ a v, u_i jsou vrcholy spojené hranou označenou indexem i na straně v. π je permutace bazických stavů prostoru mince. Operátor mince C zde má tvar

$$C = 2\,|\psi\rangle_c \langle\psi| - I_c, \tag{3.50}$$

kde $|\psi\rangle$ má tvar vyvážené superpozice

$$|\psi\rangle_c = \frac{1}{\sqrt{d}} \sum_{i=1}^{d} |i\rangle_c . \tag{3.51}$$

Takováto mince ovšem zachází se všemi vrcholy stejně a nemá schopnost rozlišit vrchol představující řešení. Pro označený vrchol m tedy zavedu nový operátor mince C' rozšířený na celý Hilbertův prostor způsobem

$$C' = I_p \otimes C - |m\rangle\langle m| \otimes (C + I_c) = I_p \otimes C - 2\,|m\rangle\langle m| \otimes |\psi\rangle_c \langle\psi| . \tag{3.52}$$

Takovýto operátor pak daný řešící vrchol označí fázovým posunem o π. Evoluční operátor pak bude mít tvar

$$U' = S\,C' = U - 2\,|v,\psi_c\rangle\langle v,\psi_c| . \tag{3.53}$$

Vyhledávací algoritmus pomocí kvantové procházky pro graf s N vrcholy pak lze rozdělit do čtyř kroků:

1. Inicializace do vyvážené superpozice $|\psi\rangle = \frac{1}{\sqrt{Nd}} \sum_{m=1}^{N} \sum_{i=1}^{d} |m, i\rangle$ pomocí Hadamardovy transformace
2. k-krát aplikace U'
3. Změření registru, který odpovídá pozici
4. Kontrola, že naměřený vrchol je onen označený hledaný vrchol.

Nyní se již konečně podívejme, jakým způsobem lze na Groverův algoritmus pro N položek nahlížet jako na vyhledávání pomocí kvantové procházky na úplném grafu. Graf nazveme úplným, jsou-li každé jeho dva vrcholy spojeny hranou. My zde budeme uvažovat i hranu představující cyklus sám na sebe. Z každého vrcholu tedy povede N hran , a tudíž $dim\mathcal{H}_p = dim\mathcal{H}_c = N$. Translační operátor zde bude působit jako

$$S\,|v, i\rangle = |i, v\rangle\,. \tag{3.54}$$

Operátor mince zde pro označený stav $|m\rangle$ bude mít tvar

$$C' = (I - 2\,|m\rangle\,\langle m|) \otimes C \tag{3.55}$$

Zde je nutno se pozastavit a podívat se na tuto volbu z hlediska souvislosti s Groverovým algoritmem. Operátor

$$C = 2\,|\psi\rangle_c\,\langle\psi| - I_c \tag{3.56}$$

je přesně náš operátor zrcadlení okolo střední hodnoty představující kroky 2), 3) a 4) z popisu Groverovy iterace. Na druhou stranu operátor

$$O \coloneqq I_p - 2\,|m\rangle\,\langle m| \tag{3.57}$$

přesně odpovídá fázovému posunu o π provedeném pomocí oracle na stavu, který představuje řešení. Můžeme tedy ztotožnit operátor mince s Groverovou iterací $C' \equiv G$. Celkově tak lze psát $U' = S \cdot G$ a podívejme se na jeho působení na počáteční stav

$$|\psi_0\rangle = \frac{1}{N} \sum_{m=1}^{N} \sum_{i=1}^{N} |m, i\rangle \coloneqq |\psi\rangle_p \otimes |\psi\rangle_c\,, \tag{3.58}$$

kde $|\psi\rangle_p$ resp. $|\psi\rangle_c$ jsou vyvážené superpozice na $\mathcal{H}_p$ resp. $\mathcal{H}_c$.

$$U'\,|\psi_0\rangle = S \cdot G\,|\psi_0\rangle = S \cdot (O \otimes C)\,|\psi\rangle_p \otimes |\psi\rangle_c = S\,(O\,|\psi\rangle_p \otimes C\,|\psi\rangle_c) = C\,|\psi\rangle_c \otimes O\,|\psi\rangle_p\,, \tag{3.59}$$

což nám už samo o sobě může dát jistou intuici o průběhu algoritmu, ale udělejme ještě pomocné výpočty.

$$G \cdot U'\,|\psi_0\rangle = (O \otimes C)(C\,|\psi\rangle_c \otimes O\,|\psi\rangle_p) = (O \cdot C)\,|\psi\rangle_c \otimes (C \cdot O)\,|\psi\rangle_p \tag{3.60}$$

a konečně

$$U'^2 = S \cdot (O \cdot C)\,|\psi\rangle_c \otimes (C \cdot O)\,|\psi\rangle_p = (C \cdot O)\,|\psi\rangle_p \otimes (O \cdot C)\,|\psi\rangle_c\,. \tag{3.61}$$

Z toho vidíme, že kvantová procházka na úplném grafu je vlastně Groverův algoritmus aplikovaný jak na $\mathcal{H}_p$, tak na $\mathcal{H}_c$. Toto automaticky vyžaduje nutný počet užití oracle vynásobit 2 oproti klasickému Groverovu algoritmu a ze vztahu (3.23) tak po dosazení $M = 1$ dostáváme

$$K = \left\lceil \frac{\pi}{2}\sqrt{N} \right\rceil. \tag{3.62}$$

3.5.3 Vyhledávání na grafu typu hvězda

Pojďme si nyní v závěrečné sekci této kapitoly ilustrovat vyhledávání pomocí kvantové procházky na grafu typu hvězda. Pokusíme se trochu ozřejmit tvorbu evolučního operátoru a také se detailněji zaměříme na složitost procesu.

Hvězda je bipartitní graf, tj. jeho vrcholy můžeme rozdělit na dvě disjunktní množiny tak, že žádné dva vrcholy ze stejné množiny nejsou spojeny hranou. V tomto konkrétním případě tvoří jednu množinu pouze jediný středový vrchol. Ten je N hranami spojen s N vnějšími vrcholy, které tvoří druhou množinu a z nichž jeden je označen. Označme středový vrchol indexem 0. Pak Hilbertův prostor vrcholů bude $\mathcal{H}_p = [|0\rangle, ..., |N\rangle]_\lambda$. Zásadní změnou oproti předchozímu příkladu ovšem je, že musíme s Hilbertovým prostorem a operátorem mince zacházet opatrněji. Konkrétně musí být různě definován pro středový a pro okrajové vrcholy.

Uvažujeme opět Z-lokální transformace. Z toho plyne povaha pohybu částice po grafu a z té zase plynou příslušné operátory mince. Na vnějších vrcholech nemá částice kam jinam jít než na středový vrchol. Stav mince tedy bude jednorozměrný a budeme ho značit $|0\rangle_c$. Naopak na středovém vrcholu může částice skočit na jakýkoli z vnějších, tudíž prostor bude N-dimenzionální s bazickými stavy označenými jako $|i\rangle_c$ korespondujícími s vrcholy. Celkový Hilbertův prostor tedy bude

$$\mathcal{H} = [|1,0\rangle, ..., |N,0\rangle, |0,1\rangle, ..., |0,N\rangle]_\lambda, \tag{3.63}$$

kde opět dodržujeme notaci, že první píšeme stav pozice a druhý stav mince.
Evoluční operátor U (oproti předchozí sekci už pro lehčí zápis vynechávám čárku) bude opět dán vztahem

$$U = S \cdot C. \tag{3.64}$$

Translační operátor S odpovídající skákání z vnějších vrcholů na středový a opačně musí mít tedy dvě části odpovídající těmto dvěma situacím. Snadno nahlédneme, že jeho přesný tvar bude

$$S = \sum_{j=1}^{N} (|j,0\rangle\langle 0,j| + |0,j\rangle\langle j,0|). \tag{3.65}$$

Operátor mince se bude na jednodimenzionálním prostoru vnějších vrcholů chovat jako identita, tedy až na označený vrchol, řekněme opět $|m\rangle_p$, kterému přidá fázi π. Na N-dimenzionálním prostoru středového vrcholu bude působit jako

$$C_0 = 2|\psi\rangle_c\langle\psi| - I_c \tag{3.66}$$

(kde $|\psi\rangle_c$ je teď vyvážená superpozice N stavů), tedy znovu jako kroky 2), 3) a 4) Groverovy iterace. Celkově tak dostáváme operátor mince jako

$$C = \underbrace{\overbrace{(I_N - 2|m\rangle_p\langle m|)}^{\text{oracle } O \text{ pro } |m\rangle} \otimes |0\rangle_c\langle 0|}_{\text{vněší}} + \underbrace{|0\rangle_c\langle 0| \otimes C_0}_{\text{středový}}. \tag{3.67}$$

Evoluční operátor tak má tvar

$$U = S \cdot C = \sum_{i=1}^{N} (|0,i\rangle\langle i,0| - 2|0,m\rangle\langle m,0|) + \frac{2}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}|i,0\rangle\langle 0,j| - \sum_{i=1}^{N}|i,0\rangle\langle 0,i|. \tag{3.68}$$

Vzhledem k tomu, že hvězda je bipartitní graf a kvantová procházka je tedy střídání skoků z vnějších vrcholů na středový, bude nás zajímat operátor dvou skoků, tedy U^2. Rozdělíme si, jak působí na

označený a neoznačený vrchol.

$$U^2 |i, 0\rangle = \frac{2}{N} \sum_{j \neq i} |j, 0\rangle - \frac{N-2}{N} |i, 0\rangle \text{ pro } i \neq m \tag{3.69}$$

$$U^2 |m, 0\rangle = \frac{N-2}{N} |m, 0\rangle - \frac{2(N-1)}{N} \sum_{i \neq m} |i, 0\rangle \tag{3.70}$$

Obdobně ke vztahům (3.13) a (3.14) bude $|m, 0\rangle$ stav představující řešení a

$$|\alpha, 0\rangle := \frac{1}{\sqrt{N-1}} \sum_{i \neq m} |i, 0\rangle \tag{3.71}$$

rovnovážná superpozice stavů neřešících problém. Vidíme, že lineární obal stavů $|m, 0\rangle$ a $|\alpha, 0\rangle$ opět tvoří invariantní podprostor U^2. Vyváženou superpozici všech stavů potom můžeme napsat jako

$$|\psi\rangle = \sqrt{\frac{N-1}{N}} |\alpha, 0\rangle + \sqrt{\frac{1}{N}} |m, 0\rangle \tag{3.72}$$

obdobně jako v (3.15) pro $M = 1$. Vidíme tedy ekvivalenci dvou kroků kvantové procházky a Groverovy iterace. Uvědomíme-li si tedy, že počet dotazů na oracle bude oproti Groverovu algoritmu dvojnásobný, získáme opět vztah pro počet kroků (3.62).

Kapitola 4

Shorův algoritmus

V této kapitole se podíváme na asi nejznámější kvantový algoritmus, který nejlépe demonstruje možnosti kvantových počítačů přesahující možnosti těch klasických. Jedná se o tzv. Shorův algoritmus [26], který ve svém důsledku umožňuje efektivně provést prvočíselný rozklad. Díky této schopnosti má Shorův algoritmus potenciál prolomení RSA, čímž dostal celé odvětví kvantové informatiky do povědomí širší veřejnosti.

Základem celého algoritmu je tzv. *kvantová Fourierova transformace*, na kterou se detailně zaměříme v první sekci. Jedná se vlastně o kvantovou verzi diskrétní Fourierovy transformace pro amplitudy stavů. Fourierova transformace je samozřejmě i mimo kvantovou mechaniku velmi mocný nástroj, nám však její kvantová verze odemkne cestu k *algoritmu pro odhad fáze*. S jeho pomocí pak již můžeme vyřešit problém hledání periody funkce nebo zde přesněji problém hledání multiplikativního řádu celého čísla modulo N. Za pomoci trochy teorie grup a čísel si pak ukážeme ekvivalenci nalezení řešení tohoto problému s faktorizací. Při odvozování budeme vycházet z [21], [11], [1], [20] a [32].

4.1 Kvantová Fourierova transformace

Diskrétní Fourierova transformace vezme vektor komplexních čísel $(x_0, ..., x_{N-1})$ a jejím výstupem je opět vektor komplexních čísel $(y_0, ..., y_{N-1})$ takových, že

$$y_k \equiv \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{\frac{i2\pi jk}{N}}. \tag{4.1}$$

Kvantová Fourierova transformace na stavy ortonormální báze $|0\rangle, ..., |N-1\rangle$ je definována vztahem

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{i2\pi jk}{N}} |k\rangle \tag{4.2}$$

a na obecný stav lze působení popsat jako

$$\sum_{j=0}^{N-1} x_j |j\rangle \rightarrow \sum_{k=0}^{N-1} y_k |k\rangle, \tag{4.3}$$

kde amplitudy y_k jsou dány (4.1). Vidíme tedy, že kvantová Fourierova transformace působí jen na amplitudy stavů a její operátor lze zapsat jako

$$U_{QFT} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} e^{\frac{i2\pi xy}{N}} |y\rangle \langle x|. \tag{4.4}$$

Jedná se vlastně o přechod z výpočetní báze do Fourierovy báze. Pro intuici je dobré si uvědomit, že Hadamardova brána je operátor kvantové Fourierovy transformace pro 1 qubit a převádí z báze $(|0\rangle, |1\rangle)$ do báze $(|+\rangle, |-\rangle)$, kde $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$. Bazické stavy jsou zde při interpretaci na Blochově sféře dány kladnou či zápornou projekcí do osy x. Máme-li n qubitů a budeme-li postupně procházet bazické stavy $|x\rangle, x \in \{0, ..., N-1\}$, kde $N = 2^n$, můžeme si všimnout, že každý bude měnit svou hodnotu s jinou frekvencí (poslední se bude měnit při každém zvýšení bazického stavu, zatímco první se bude měnit nejpomaleji). Ve Fourierově bázi jsou čísla ukládána pomocí různých rotací okolo osy z v rovině xy. Úhel, o který budou otočeny jednotlivé qubity okolo osy z je dán právě vztahem (4.2). Opět zde platí, že budeme-li procházet všechny bazické stavy, pak při jejich vyjádření ve Fourierově bázi budou qubity rotovat každý s jinou frekvencí (zde naopak poslední qubit bude mít nejnižší frekvenci rotace a první nejvyšší).

Dále bude výhodné psát bazický stav $|j\rangle$ pomocí bitové reprezentace $j = j_1...j_n \equiv j_1 2^{n-1} + ...j_n 2^0$. Zavedeme notaci *binárních zlomků* jako $0.j_l...j_m \equiv \frac{j_l}{2} + ... + \frac{j_m}{2^{m-l+1}}$. Takto můžeme vztah (4.2) zapsat jako

$$
\begin{aligned}
U_{QFT}|j\rangle &= \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{\frac{i2\pi jk}{2^n}} |k\rangle \text{, dále využiji } k = k_1...j_n \equiv k_1 2^{n-1} + ...k_n 2^0 \\
&= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} ... \sum_{k_n=0}^{1} e^{i2\pi j(\sum_{l=1}^{n} \frac{k_l}{2^l})} |k_1...k_n\rangle \\
&= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} ... \sum_{k_n=0}^{1} \prod_{l=1}^{n} \left(e^{\frac{i2\pi jk_l}{2^l}}\right) |k_1...k_n\rangle \text{, dále platí } \prod_{l=1}^{n} \left(e^{\frac{i2\pi jk_l}{2^l}}\right) |k_1...k_n\rangle = \bigotimes_{l=1}^{n} e^{\frac{i2\pi jk_l}{2^l}|k_l\rangle} \\
&= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} ... \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} e^{\frac{i2\pi jk_l}{2^l}|k_l\rangle} \\
&= \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \left(\sum_{k_l=0}^{1} e^{i2\pi j\frac{k_l}{2^l}} \right), \text{ dále } \sum_{k_l=0}^{1} e^{i2\pi j\frac{k_l}{2^l}} = |0\rangle + e^{i2\pi\frac{j}{2^l}}|1\rangle \\
&= \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \left(|0\rangle + e^{i2\pi\frac{j}{2^l}}|1\rangle \right), \text{ nakonec } e^{i2\pi\frac{j}{2^l}} = \underbrace{e^{i2\pi\frac{j_1 2^{n-1}+...+j_{n-l}2^l}{2^l}}}_{=1} e^{i2\pi 0.j_{n-l+1}...j_n} = e^{i2\pi 0.j_{n-l+1}...j_n} \\
&= \frac{(|0\rangle + e^{i2\pi 0.j_n}|1\rangle) \otimes (|0\rangle + e^{i2\pi 0.j_{n-1}j_n}|1\rangle) \otimes ... \otimes (|0\rangle + e^{i2\pi 0.j_1...j_n}|1\rangle)}{2^{n/2}}
\end{aligned}
\tag{4.5}
$$

Poslední řádek předešlé rovnice nám poskytuje vyjádření, podle kterého můžeme zkonstruovat efektivní kvantový obvod pro výpočet Fourierovy transformace.

Ke konstrukci daného obvodu si nejprve uvědomíme, že působení Hadamardovy brány na jednoqubitový stav $|j\rangle$ lze zapsat jako

$$H|j\rangle = \frac{|0\rangle + e^{i2\pi\frac{j}{2}}|1\rangle}{\sqrt{2}}. \tag{4.6}$$

Toho využijeme při zápisu aplikace Hadamardovy brány na první qubit vícequbitového stavu $|j_1...j_n\rangle$:

$$H|j_1...j_n\rangle = \frac{|0\rangle + e^{i2\pi 0.j_1}|1\rangle}{\sqrt{2}}|j_2...j_n\rangle. \tag{4.7}$$

Dále definuji kontrolovanou rotaci

$$CR_k := \begin{pmatrix} I & 0 \\ 0 & R_k \end{pmatrix}, \tag{4.8}$$

kde

$$R_k \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & e^{i2\pi/2^k} \end{pmatrix}. \tag{4.9}$$

Obecně lze působení CR_k na dvouqubitový stav $|j_k j_l\rangle$, kde první qubit poslouží jako kontrolní a druhý jako cílový, zapsat jako

$$CR_k \, |0 j_l\rangle = |0 j_l\rangle \tag{4.10}$$

resp.

$$CR_k \, |1 j_l\rangle = e^{i2\pi \frac{j_l}{2^k}} \, |1 j_l\rangle \, . \tag{4.11}$$

Z toho je vidět, že aplikujeme-li CR_2 na náš první qubit ve stavu (4.7) s druhým qubitem jako kontrolním, dostaneme stav

$$\frac{|0\rangle + e^{i2\pi 0.j_1 j_2} \, |1\rangle}{\sqrt{2}} \, |j_2 ... j_n\rangle \, . \tag{4.12}$$

Budeme-li analogicky pokračovat v aplikování bran CR_3 až CR_n na první qubit s kontrolními qubity j_3 až j_n, získáme vztah

$$CR_n CR_{n-1} ... CR_2 H \, |j_1 j_2 ... j_n\rangle = \frac{|0\rangle + e^{i2\pi 0.j_1 j_2 ... j_n} \, |1\rangle}{\sqrt{2}} \, |j_2 ... j_n\rangle \, . \tag{4.13}$$

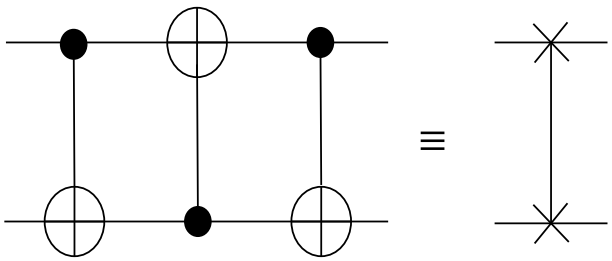
Analogický proces uděláme na druhém qubitu, tentokrát však pro každou rotaci CR_k bude hrát roli kontrolního qubitu qubit s indexem $k + 1$. Celkově tak aplikovaných rotací bude $n - 2$ a výsledný stav bude

$$\frac{(|0\rangle + e^{i2\pi 0.j_1 j_2 ... j_n} \, |1\rangle) \otimes (|0\rangle + e^{i2\pi 0.j_2 j_3 ... j_n} \, |1\rangle)}{2} \, |j_3 ... j_n\rangle \, . \tag{4.14}$$

Zopakujeme-li to pro všechny qubity (přičemž na poslední se aplikuje pouze Hadamardova brána), získáme stav

$$\frac{(|0\rangle + e^{i2\pi 0.j_1 j_2 ... j_n} \, |1\rangle) \otimes (|0\rangle + e^{i2\pi 0.j_2 j_3 ... j_n} \, |1\rangle) \otimes ... \otimes (|0\rangle + e^{i2\pi 0.j_n} \, |1\rangle)}{2^{n/2}}. \tag{4.15}$$

K získání finálního stavu z (4.5) nám již stačí otočit pořadí qubitů, což lze snadno udělat pomocí tzv. *swap operace*, jejíž implementace je znázorněna na obrázku (4.1).



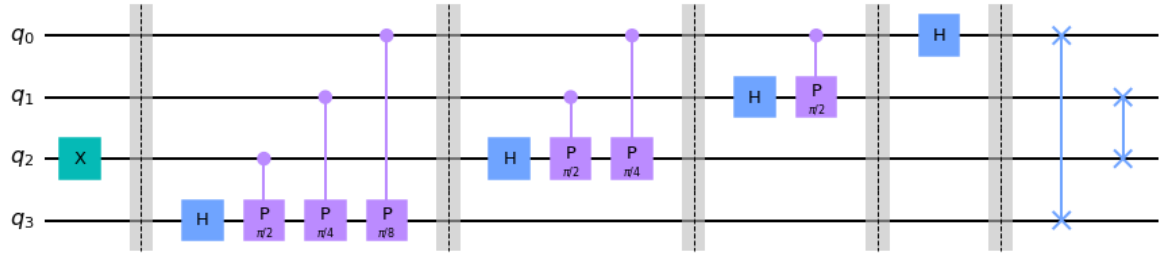
Obrázek 4.1: Znázornění implementace *swap* brány prohazující dva qubity pomocí *CNOT* bran.

Z konstrukce vidíme dvě věci. Jednak že kvantová Fourierova transformace je skutečně unitární operace (všechny použité brány byly unitární) a jednak to, že k její konstrukci potřebujeme $\Theta(n^2)$ bran.

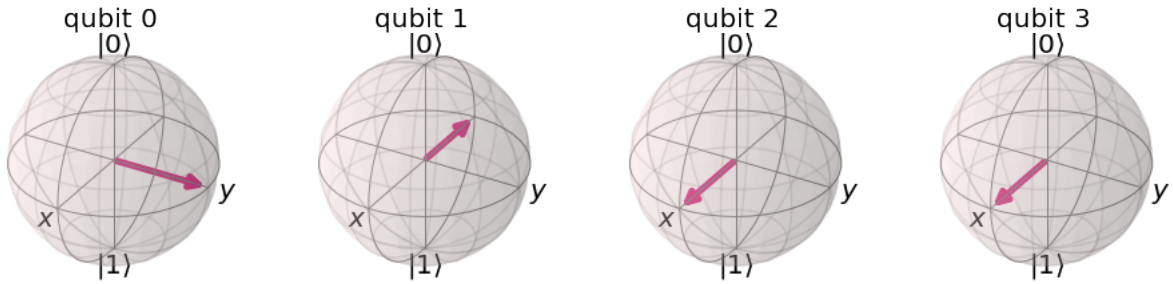
Ukázka implementace kvantové Fourierovy transformace pro 4 qubity v Qiskit je znázorněna na obrázku (4.2) a výsledné qubity zakódované ve Fourierově bázi na obrázku (4.3). K implementaci CR_k se zde používá brána $CP(\theta)$ definovaná jako

$$CP(\theta) \coloneqq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\theta} \end{pmatrix}, \tag{4.16}$$

kde tedy v našem případě $\theta = 2\pi/2^k = \pi/2^{k-1}$.



Obrázek 4.2: Implementace kvantové Fourierovy transformace aplikované na stav $|j\rangle = |0100\rangle$ v Qiskit. Opět díky obrácenému pořadí qubitů v Qiskit je nutné obrázek horizontálně převrátit, aby přesně odpovídal odvozování v teorii.



Obrázek 4.3: Stav $|0100\rangle$ zakódovaný Fourierovou transformací do Fourierovy báze. Opět vzhledem k odvozované teorii musíme pořadí qubitů oproti obrázku otočit.

4.2 Algoritmus pro odhad fáze

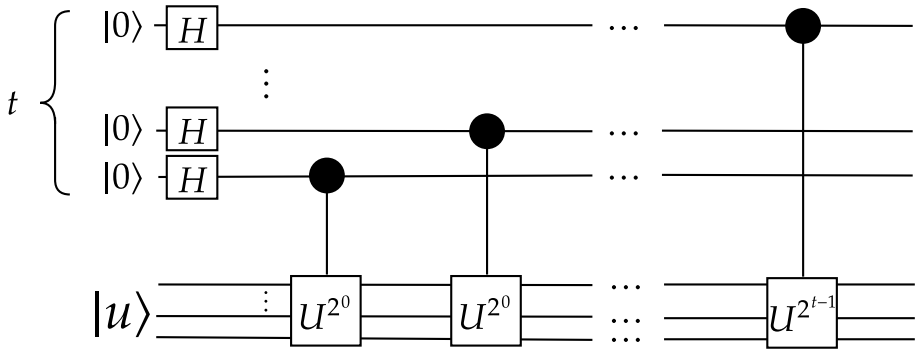
Mějme unitární operátor U a jeho vlastní stav $|u\rangle$ příslušný vlastní hodnotě $e^{2\pi i\varphi}$, kde φ neznáme. Unitárnost U nám říká, že norma vlastních hodnot bude vždy 1. Naším úkolem je φ určit s co největší přesností. K tomu přesně slouží algoritmus pro odhad fáze. Předpokládáme zde, že máme oracle schopný produkovat stav $|u\rangle$ a operaci CU^{2^j}, což implikuje využití tohoto konceptu v rámci jiných algoritmů (viz např. *kvantové počítání*) než jako algoritmus sám o sobě.

Obvod realizující tento algoritmus vyžaduje dva registry. První obsahující počet qubitů t ve stavu $|0\rangle^{\otimes t}$ (vhodný počet qubitů t si odvodíme později) a druhý ve stavu $|u\rangle$ s počtem qubitů odpovídajícím stavu $|u\rangle$.

První fáze algoritmu je zobrazena na obrázku (4.4). Využívá se zde (stejně jako v Deutsch-Joszově algoritmu) *fázového zpětného rázu* (viz ukázka pro T bránu (2.9)). V našem případě nám fázový zpětný ráz přenese fázi U do Fourierovy báze na t qubitů v prvním registru. V předešlé sekci jsme si naznačili, že poslední qubit udělá jednu celou rotaci ve Fourierově bázi při počítání od 0 do 2^t – při reprezentaci nějakého $j \in \{0, .., 2^t - 1\}$ se tento qubit pootočí o $j/2^t$ celých otáček okolo osy z. Každý další qubit se pak otáčí o dvojnásobek toho předchozího. Přesně toho my nyní využijeme.

Chceme-li zakódovat fázi φ vlastní hodnoty $e^{2\pi i\varphi}$ do Fourierovy báze, pak qubity prvního registru použijeme jako kontrolní. Na to, abychom fázovým zpětným rázem přenesli vlastní hodnotu na poslední qubit, nám stačí jediné použití CU s posledním qubitem prvního registru jako kontrolním. Předposlední

qubit prvního registru pak pro zakódování do Fourierovy báze musíme otočit dvakrát tolik, tedy použijeme CU^2 s tímto qubitem jako kontrolním. Takto musíme vždy zdvojnásobovat počet užití CU pro každý další qubit prvního registru. Toto je znázorněno na obrázku (4.4).



Obrázek 4.4: První fáze algoritmu pro odhad fáze.

Zapišme to nyní matematicky korektně. Začínáme ve stavu $|0\rangle^{\otimes t} \otimes |u\rangle$. Aplikací Hadamardovy transformace na první registr získáme stav

$$\frac{|0\rangle + |1\rangle}{2^{t/2}} |u\rangle . \tag{4.17}$$

Po aplikaci výše popsaného procesu bude stav prvního registru

$$\frac{1}{2^{t/2}} \left(|0\rangle + e^{2\pi i 2^{t-1}\varphi} |1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i 2^{t-2}\varphi} |1\rangle\right) \otimes ... \otimes \left(|0\rangle + e^{2\pi i 2^{0}\varphi} |1\rangle\right) = \frac{1}{2^{t/2}} \sum_{k=0}^{2^t-1} e^{2\pi i \varphi k} |k\rangle \tag{4.18}$$

Nyní předpokládejme, že φ lze zapsat přesně pomocí t bitů a zapišme ho jako $\varphi = 0.\varphi_1 ... \varphi_t$. Stav (4.18) pak můžeme zapsat jako

$$\frac{1}{2^{t/2}} \left(|0\rangle + e^{2\pi i 0.\varphi_t} |1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i 0.\varphi_{t-1}\varphi_t} |1\rangle\right) \otimes ... \otimes \left(|0\rangle + e^{2\pi i 0.\varphi_1\varphi_2...\varphi_t} |1\rangle\right). \tag{4.19}$$

Druhou fází celého algoritmu je nyní už ze zjevných důvodů aplikace inverzní Fourierovy transformace. Porovnáme-li totiž stav (4.19) s výsledkem vztahu (4.5), pak je jasné, že po aplikaci inverzní Fourierovy transformace bude výsledný stav $|\varphi_1\varphi_2...\varphi_t\rangle$. Měřením ve výpočetní bázi pak dostaneme přesně φ. Implementace celkového algoritmu v rámci *kvantového počítání* v Qiskit je na obrázku (3.8), kde již místo obecné unitární operace používáme Groverovu iteraci.

V předešlém postupu jsme uvažovali φ, které šlo přesně zapsat pomocí binárního rozvoje t bitů. To ovšem není vždy možné. Budeme tedy muset φ aproximovat. Velikost našeho prvního registru pak bude záviset na dvou požadavcích: na kolik cifer přesně chceme mít odhad φ a s jak velkou pravděpodobností chceme, aby algoritmus skončil úspěšně. Analýzou těchto požadavků lze ukázat, že chceme-li φ aproximovat s přesností na n bitů a pravděpodobností úspěchu přinejmenším $1 - \epsilon$, pak vhodnou volbou počtu bitů v prvním registru bude

$$t = n + \left\lceil \log\left(2 + \frac{1}{2\epsilon}\right)\right\rceil . \tag{4.20}$$

Může se také stát, že nebudeme schopni připravit vlastní stav $|u\rangle$. Libovolný stav $|\psi\rangle$ však můžeme rozepsat do báze vlastních stavů U jako $|\psi\rangle = \sum_u c_u |u\rangle$. Nechť vlastní stav $|u\rangle$ přísluší vlastní hodnotě $e^{2\pi i \varphi_u}$. Není těžké si uvědomit, že výstup algoritmu pro odhad fáze pak bude blízký stavu $\sum_u c_u |\tilde{\varphi_u}\rangle |u\rangle$, kde $|\tilde{\varphi_u}\rangle$ je aproximace φ_u. Pravděpodobnost naměření stavu $|\tilde{\varphi_u}\rangle$ je potom $|c_u|^2$. Celkově tak při volbě t podle (4.20) dostáváme pravděpodobnost naměření φ_u s přesností na n bitů přinejmenším $|c_u|^2(1 - \epsilon)$.

4.3 Matematický aparát

Jak již bylo naznačeno, k odvození celkového algoritmu bude potřeba se ponořit alespoň do základů teorie grup a hlavně teorie čísel, a to především za účelem ukázání ekvivalence nalezení multiplikativního řádu celého čísla modulo N a faktorizace. Tato sekce tedy bude strukturována v klasickém matematickém pojetí a předpokládá se znalost základních pojmů teorie množin, popř. alespoň malý vhled do modulární aritmetiky.

Definice 1 (Grupa)**.** Grupa je uspořádaná čtveřice $(G,*, \iota, e)$, kde G je množina, $*$ je binární operace na G, ι je unární operace na G a $e \in G$ (nulární operace), které splňují

1. $\forall x, y \in G, x * y \in G$
2. $\forall x, y, z \in G,\ x * (y * z) = (x * y) * z,$
3. $\forall x \in G, e * x = x * e = x,$
4. $\forall x \in G, x * \iota(x) = \iota(x) * x = e,$

Pokud navíc platí $\forall x, y \in G, x * y = y * x$, potom říkáme, že grupa G je *komutativní* nebo *abelovská*.

Poznámka. Při použití multiplikativního resp. aditivního zápisu se $\iota(x)$ nazývá inverzní resp. opačný prvek a místo $\iota(x)$ píšeme x^{-1} resp. $-x$.

Definice 2 (Homomorfismus grup)**.** Buďte $(G,*, \iota, e)$ a $(G',*', \iota', e')$ grupy. Zobrazení $f : G \to G'$ je homomorfismus, jestliže

$$(\forall a, b \in G)(f(a * b) = f(a) *' f(b)).$$

Definice 3 (Řád grupy)**.** Buď G grupa. Počet prvků grupy označíme $|G|$ a nazveme ho *řádem grupy*.

Definice 4 (Generátor grupy)**.** Buď G grupa. O množině $\{a_1, ..., a_n\} \subset G$ řekneme, že generuje grupu G, jestliže každý prvek grupy G lze zapsat jako produkt prvků obsažených v této množině při dané grupové operaci $*$. Značíme $G = \langle a_1, ..., a_n \rangle$.

Definice 5 (Podgrupa)**.** Buďte (G, *, ι, e) grupa a $H \subset G$. Nechť je splněno:

1. $e \in H$,
2. $(\forall a \in H)(\iota(a) \in H)$,
3. $(\forall a, b \in H)(a * b \in H)$.

Potom $(H, *, \iota, e)$ s operacemi zúženými na H je grupa. Říkáme, že H je *podgrupou* G.

Definice 6 (Řád prvku grupy)**.** Pro $a \in G$ řád podgrupy $\langle a \rangle_G$ nazýváme *řádem prvku* a.

Definice 7. Řekneme, že grupa G je *cyklická*, jestliže je generovaná jedním prvkem $a \in G$, tj. $G = \langle a \rangle$.

Poznámka. Ve vztahu $r = m \bmod n$ se r nazývá zbytkem při dělení m dělitelem n, přičemž se pohybujeme v celých číslech.
Je-li m dělitelné n beze zbytku, píšeme $n|m$.
Mají-li dvě čísla $a, b \in \mathbb{Z}$ stejný zbytek při dělení $n \in \mathbb{N}$, říkáme, že jsou kongruentní modulo n a píšeme $a \equiv b (\bmod n)$.

Poznámka. Pro $n \in \mathbb{N}$ je

$$\mathbb{Z}_n := (\{0, ..., n-1\}, +\text{mod } n, ', 0)$$

abelovská grupa. Symbol $\mathbb{Z}_n$ budeme používat i pro označení samotné množiny zbytků při dělení n $\{0, ..., n-1\}$. Opačný prvek r' k prvku r je jednoznačně určen a je roven $r' = n - r$ pro $1 \leq r \leq n-1$, $0' = 0$.

Definice 8 (Nesoudělná čísla)**.** Čísla, která mají jediného společného dělitele - číslo 1, nazýváme nesoudělná.

Poznámka. Definujme $\mathbb{Z}_n^*$ jako množinu všech elementů $\mathbb{Z}_n$ takových, že mají inverze modulo n, tj. elementů $\mathbb{Z}_n$ nesoudělných s n. Lze ověřit, že $\mathbb{Z}_n^*$ tvoří grupu velikosti $\varphi(n)$, kde $\varphi(n)$ je tzv. *Eulerova* φ *funkce*, jejíž hodnotou je počet kladných celých čísel menších než n nesoudělných s n. Je-li n mocninou nějakého lichého prvočísla p, $n = p^\alpha$, pak lze ukázat, že $\mathbb{Z}_{p^\alpha}^*$ tvoří cyklickou grupu.

Poznámka. Uveďme si bez důkazu jeden známý fakt. Nechť $a, b \in \mathbb{N}$ jsou nesoudělná čísla. Potom $(\forall m \in \mathbb{Z})(a|mb \implies a|m)$.

Lemma 1. Buďte $n \in \mathbb{N}$ a $a_1, a_2, ..., a_n \in \mathbb{N}$ po dvou nesoudělná čísla. Potom platí

$$(\forall m \in \mathbb{Z})(a_1|m \wedge a_2|m \wedge ... \wedge a_n|m \implies a_1 a_2 ... a_n|m).$$

Důkaz. Budeme postupovat matematickou indukcí podle n. Pro $n = 1$ tvrzení triviálně platí. Pro $n = 2$ vyjdeme z předchozí poznámky. $a_1|m \wedge a_2|m$ znamená, že $m = ka_1 = la_2$ pro jistá $k, l \in \mathbb{Z}$. Potom $a_2|a_1 k$ a podle poznámky $a_2|k$. To znamená, že $k = ja_2$ pro jisté $j \in \mathbb{Z}$ a $m = ka_1 = ja_1 a_2$, čili $a_1 a_2|m$ (to také implikuje, že pokud prvočíslo p splňuje $p|kl$ pro jistá $k, l \in \mathbb{Z}$, pak $p|k$ nebo $p|l$ a p nazýváme prvočinitel). Při obecném kroku $n - 1 \to n$ pro $n \geq 2$ vezmeme v úvahu, že čísla $a_1 ... a_{n-1}$ a a_n jsou nesoudělná. Kdyby byla soudělná, pak by nutně měla společného prvočíselného dělitele p. Pak by platilo $p|a_j$ pro jisté $j \in \{1, ..., n-1\}$ a současně $p|a_n$, což je spor s předpokladem. Z indukčního předpokladu tak máme $a_1 ... a_n|m$ a navíc $a_n|m$. Z toho již plyne $a_1 ... a_n|m$. □

Věta 2 (Čínská věta o zbytcích)**.** Buďte $n \in \mathbb{N}$ a $m_1, ..., m_n \in \mathbb{N}$ po dvou nesoudělná čísla. Položme

$$M := m_1 m_2 ... m_n.$$

Potom zobrazení

$$\varphi : \mathbb{Z}_M \to \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times ... \times \mathbb{Z}_{m_n} : k \to (k \bmod m_1, k \bmod m_2, ..., k \bmod m_n)$$

je izomorfismus. Platí tedy

$$\mathbb{Z}_M \simeq \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times ... \times \mathbb{Z}_{m_n},$$

kde $\simeq$ značí, že grupy jsou izomorfní.

Poznámka. Při stejném značení a předpokladech lze předešlou větu zapsat také následovně.
Pro libovolnou n-tici zbytků $(f_1, f_2, ..., f_n) \in \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times ... \times \mathbb{Z}_{m_n}$ existuje právě jedno řešení $k \in \mathbb{Z}_M$ soustavy rovnic

$$k \equiv f_1 (\text{mod } m_1), k \equiv f_2 (\text{mod } m_2), ..., k \equiv f_n (\text{mod } m_n).$$

Důkaz. Je snadné si ukázat, že φ je skutečně homomorfismus grup. Ukážeme, že φ je monomorfismus. Prvek $k \in \mathbb{Z}_M$ patří do Kerφ, právě když

$$k \equiv 0(\text{mod } m_1) \wedge k \equiv 0(\text{mod } m_2) \wedge ... \wedge k \equiv 0(\text{mod } m_n),$$

což lze ekvivalentně přepsat jako $m_1|k \wedge m_2|k \wedge ... \wedge m_n|k$. Podle lemma 1 pak platí $M|k$. Jelikož však $0 \leq k \leq M$, tak nutně $k = 0$, tedy φ je injektivní. Z injektivity ovšem plyne i surjetivita, neboť

$$|\mathbb{Z}_M| = m_1 m_2 ... m_n = \left|\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times ... \times \mathbb{Z}_{m_n}\right|$$

a φ je tedy izomorfismus. □

Definice 9 (Řád modulo N)**.** Buď $N \in \mathbb{N}$, $x \in \{1, ..., N-1\}$ a nechť N a x jsou nesoudělná. *Řádem* x modulo N nazveme nejmenší kladné celé číslo r takové, že $x^r = 1(\text{mod } N)$. Značíme $r = ord_N(x)$.

Poznámka. Obecněji lze říct, že řád x modulo N je řád prvku x v multiplikativní grupě tvořené prvky s multiplikativní inverzí v okruhu celých čísel modulo N – viz zavedení $\mathbb{Z}_N^*$. My si však vystačíme s výše uvedenou definicí.

Poznámka. Dále budeme předpokládat znalost Eukleidova algoritmu na hledání největšího společného dělitele (značíme nsd). Ten převádí hledání nsd dvou původně velkých čísel na hledání nsd dvou menších čísel pomocí celočíselného dělení. Lze ukázat, že pro nalezení nsd čísel a, b takových, že obě tato čísla mohou být reprezentována maximálně L bity, je potřeba $O(L^3)$ operací.

Věta 3. Nechť N je složené číslo reprezentovatelné L bity a x je netriviální řešení rovnice $x^2 = 1(\text{mod } N)$ pro $x \in \{1, ..., N\}$, tj. $x \neq \pm 1(\text{mod } N)$. Pak alespoň jeden z dvojice nsd$(x-1, N)$, nsd$(x+1, N)$ je netriviální faktor N, který můžeme spočítat za použití $O(L^3)$ operací.

Důkaz. Jelikož $x^2 = 1(\text{mod } N)$, tak $N|(x+1)(x-1)$, a tedy N musí mít společný faktor s jedním z těchto dvou činitelů. Vzhledem k tomu, že z předpokladů plyne $1 < x < N-1$, tak $x - 1 < x + 1 < N$, a tedy společným faktorem nemůže být samotné N. Za pomoci Eukleidova algoritmu pak můžeme s použitím $O(L^3)$ operací vypočítat nsd$(x-1, N)$ a nsd$(x+1, N)$, čímž získáme netriviální faktor N. □

Lemma 2. Buď p liché prvočíslo. Nechť 2^d je největší mocnina 2 taková, že $2^d|\varphi(p^\alpha)$. Pak s pravděpodobností přesně $\frac{1}{2}$ bude 2^d dělit řád modulo p^α prvku náhodně vybraného z grupy $\mathbb{Z}_{p^\alpha}^*$.

Důkaz. Vzhledem k tomu, že p je prvočíslo, tak si snadno uvědomíme, že $\varphi(p^\alpha) = p^{\alpha-1}(p-1)$, což je sudé číslo, a tedy $d \geq 1$. Jak jsme již dříve uvedli, $\mathbb{Z}_{p^\alpha}^*$ je cyklická grupa, což implikuje existenci g takového, že $\langle g \rangle = \mathbb{Z}_{p^\alpha}^*$. To znamená, že každý prvek této grupy lze zapsat ve formě $g^k(\text{mod } p^\alpha)$ pro nějaké $k \in \{1, ..., \varphi(p^\alpha)\}$. Nechť r je řád g^k modulo p^α. Uvažujme dále dva případy. Pro k liché z toho, že $g^{kr} = 1(\text{mod } p^\alpha)$, dostáváme, že $\varphi(p^\alpha)|kr$ a z lichosti k tedy $2^d|r$. Pro k sudé platí

$$g^{k\varphi(p^\alpha)/2} = (g^{\varphi(p^\alpha)})^{k/2} = 1^{k/2} = 1(\text{mod } p^\alpha).$$

Proto $r|\varphi(p^\alpha)/2$ z čehož vyplývá, že 2^d nedělí r. Prvky $\mathbb{Z}_{p^\alpha}^*$ tedy můžou být rozděleny na dvě stejně velké množiny podle toho, jestli při jejich vyjádření jako g^k je k sudé nebo liché. Z toho již plyne pravděpodobnost $\frac{1}{2}$, že 2^d dělí řád r prvku náhodně vybraného z $\mathbb{Z}_{p^\alpha}^*$. □

Věta 4. Nechť $N = p_1^{\alpha_1} ... p_m^{\alpha_m}$ je prvočíselný rozklad lichého složeného přirozeného čísla. Nechť x je vybráno zcela náhodně ze $\mathbb{Z}_N^*$ a nechť r je řád x modulo N. Pak pravděpodobnost, že r je sudé a $x^{r/2} \neq -1(\text{mod } N)$ je větší nebo rovna $1 - \frac{1}{2^m}$.

Důkaz. Ekvivalentně ukážeme, že pravděpodobnost, že r je liché nebo $x^{r/2} = -1(\mathrm{mod}\, N)$ je menší nebo rovna $\frac{1}{2^m}$. Z Čínské věty o zbytcích plyne, že zcela náhodný výběr prvku x z $\mathbb{Z}_N^*$ je ekvivalentní nezávislému zcela náhodnému výběru prvku x_j z $\mathbb{Z}^*_{p_j^{\alpha_j}}$ s požadavkem $x = x_j(\mathrm{mod}\, p_j^{\alpha_j})$ pro každé j. Nechť r_j je řád x_j modulo $p_j^{\alpha_j}$, 2^{d_j} je největší mocnina 2 tak, že $2^{d_j}|r_j$ a nechť 2^d je největší mocnina 2 tak, že $2^d|r$. Ukážeme, že aby bylo r liché nebo aby $x^{r/2} = -1(\mathrm{mod}\, N)$, tak je nutné, aby d_j mělo stejnou hodnotu pro všechny j. Z lemma 2 pak plyne, že pravděpodobnost tohoto jevu je $\frac{1}{2^m}$.
Opět si situaci rozdělíme na dva případy. Pro r liché snadno vidíme, že $r_j|r$ pro každé j. To znamená, že r_j je liché, a tedy $d_j = 0$ pro všechna j. Pro r sudé platí $x^{r/2} = -1(\mathrm{mod}\, N)$. Pak ovšem také $x^{r/2} = -1(\mathrm{mod}\, p_j^{\alpha_j})$. Z toho plyne, že r_j nedělí $\frac{r}{2}$ a jelikož $r_j|r$, tak nutně $d_j = d$ pro všechna j. □

Definice 10 (Řetězový zlomek)**.** Buď $N \in \mathbb{N}_0$. Konečným řetězovým zlomkem délky N rozumíme výraz tvaru

$$[a_0, ..., a_N] \equiv a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ldots + \cfrac{1}{a_N}}}}.$$

Definujeme n-tý konvergent pro $0 \leq n \leq N$ tohoto zlomku jako $[a_0, ..., a_n]$.

Poznámka. Algoritmus řetězových zlomů slouží především k tomu, abychom byli schopni zapsat reálná čísla pouze pomocí celých čísel. Celý postup spočívá v rozdělení čísla na celou a zlomkovou část, invertování zlomkové části a opakování postupu pro takto vyjádřený zlomek. Nejlépe se to ukáže na příkladu

$$\frac{31}{13} = 2 + \frac{\mathbf{5}}{13} = 2 + \frac{1}{\frac{13}{5}} = 2 + \frac{1}{2 + \frac{\mathbf{3}}{5}} = 2 + \frac{1}{2 + \frac{1}{\frac{5}{3}}} = 2 + \frac{1}{2 + \frac{1}{1 + \frac{\mathbf{2}}{3}}} = 2 + \frac{1}{2 + \frac{1}{1 + \frac{\mathbf{1}}{\frac{3}{2}}}} = 2 + \frac{1}{2 + \frac{1}{1 + \frac{1}{1 + \frac{\mathbf{1}}{2}}}}.$$

Tučně jsme vyznačili čitatele, aby bylo vidět, že klesají (5, 3, 2, 1).

Věta 5. Buď $x \geq 1$ racionální číslo. Pak x má reprezentaci ve formě řetězového zlomku, $x = [a_0, ...a_N]$, která může být nalezena algoritmem pro řetězové zlomky.

Důkaz. Větu jsme v podstatě dokázali v předešlé poznámce. Vzhledem k tomu, že posloupnost (zvýrazněných) čitatelů je klesající, tak je jasné, že pro libovolné racionální číslo algoritmus skončí po konečném počtu kroků. □

Věta 6. Buď $a_0, ..., a_N$ posloupnost přirozených čísel. Pak

$$[a_0, ..., a_n] = \frac{p_n}{q_n},$$

kde p_n, q_n jsou reálná čísla definovaná rekurzivně následovně: $p_0 \equiv a_0$, $q_0 \equiv 1$, $p_1 \equiv 1 + a_0a_1$, $q_1 \equiv a_1$ a pro $2 \leq n \leq N$ platí

$$p_n \equiv a_n p_{n-1} + p_{n-2}$$

$$q_n \equiv a_n q_{n-1} + q_{n-2}.$$

Jsou-li a_j kladná celá čísla, pak i p_j a q_j jsou kladná celá čísla.

Důkaz. Důkaz není složitý, v rámci této práce ho však uvádět nebudeme a může být nalezen v [21]. □

Poznámka. Z předchozí věty můžeme odvodit, kolik algoritmus vyžaduje operací. Uvažme $x = \frac{p}{q} > 1$, kde p,q jsou nesoudělná. Nechť $a_0, ..., a_N$ jsou kladná celá čísla. Z definice pak p_n i q_n tvoří rostoucí posloupnosti a platí $p_n \geq 2p_{n-2}$, $q_n \geq 2q_{n-2}$. Z toho již $2^{\lfloor N/2 \rfloor} \leq q \leq p$, a tedy potřebných hodnot na vyjádření x jakožto řetězového zlomku je $N = O(\log(p))$.
Pro x racionální a p, q vyjádřitelné pomocí L bitů pak algoritmus vyžaduje $O(L)$ operací na rozdělení a invertování, z nichž každá vyžaduje $O(L^2)$ operací na základní aritmetiku. Celkově tedy máme složitost $O(L^3)$.

Věta 7. Buď x racionální číslo a nechť $\frac{p}{q}$ je racionální číslo takové, že

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

Pak $\frac{p}{q}$ je konvergentem řetězového zlomku pro x.

Důkaz. Buď $\frac{p}{q} = [a_0, ..., a_n]$ řetězový zlomek a nechť p_j, q_j jsou definovány jako ve větě 6 tak, že $\frac{p_n}{q_n} = \frac{p}{q}$. Nechť δ je definována vztahem

$$x \equiv \frac{p_n}{q_n} + \frac{\delta}{2q^2},$$

tedy $|\delta| \leq 1$. Definujme dále λ jako

$$\lambda \equiv 2\left(\frac{q_n p_{n-1} - p_n q_{n-1}}{\delta} = \frac{q_{n-1}}{q_n} \right).$$

Pak

$$x = \frac{\lambda p_n + p_{n-1}}{\lambda q_n + q_{n-1}},$$

což znamená, že $x = [a_0, ..., a_n, \lambda]$. Vzhledem k tomu, že indukcí podle n lze dokázat vztah $q_n p_{n-1} - p_n q_{n-1} = (-1)^n$, tak při volbě sudého n dostáváme

$$\lambda = \frac{2}{\delta} - \frac{q_{n-1}}{q_n} > 1,$$

kde nerovnost plyne z toho, že posloupnost q_n je rostoucí. λ je tedy racionální číslo větší než 1 a můžeme ho zapsat jako řetězový zlomek $\lambda = [b_0, ..., b_m]$. Tím dostaneme řetězový zlomek $x = [a_0, ..., a_n, b_0, ..., b_m]$ s $\frac{p}{q}$ jakožto n-tým konvergentem. □

4.4 Hledání řádu modulo N

Zkoumejme nyní už samotný algoritmus na hledání řádu prvku modulo N. Opět počet bitů potřebný pro specifikaci N je $L \equiv \lceil \log(N) \rceil$. Jedná se o problém, k němuž není v klasické informatice znám efektivní algoritmus. My si ukážeme řešení tohoto problému s využitím nástrojů, které již máme k dispozici. Poté se podíváme, co přesně se v průběhu algoritmu odehrává. Nakonec v další sekci využijeme odvozený matematický aparát, abychom tento algoritmus zahrnuli do celkového algoritmu pro faktorizaci.

Připomeneme definici, že pro přirozená čísla x a N, $x < N$, nazveme řádem x modulo N nejmenší přirozené číslo r takové, že $x^r = 1 (\text{mod } N)$. Řešením tohoto problému bude zprvu možná vcelku neintuitivně použít algoritmus pro odhad fáze na unitární operátor, jehož působení lze zapsat jako

$$U |y\rangle \equiv |xy (\text{mod } N)\rangle , \tag{4.21}$$

kde $|y\rangle$ představuje nějaký bazický stav, $y \in \{0, ..., N-1\}$. Abychom pochopili, k čemu je tento operátor dobrý, tak se podíváme na jeho vlastní stavy. Mějme na počátku registr ve stavu $|1\rangle$. Snadno nahlédneme, že při postupných aplikacích U na tento stav vždy násobíme daný stav $x(\mathrm{mod}\, N)$. Z definice řádu a operátoru U pak vyplývá, že po r aplikacích U dostanu opět stav $|1\rangle$, tj $U^r|1\rangle = |1\rangle$. Už zde si můžeme povšimnout periodicity působení U. Vzhledem k tomu, že stavy, které se při této postupné aplikaci vyskytnou, tvoří cyklus, tak stav jejich rovnovážné superpozice

$$|u_0\rangle \equiv \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \left|x^k \mathrm{mod}\, N\right\rangle \tag{4.22}$$

bude určitě vlastní stav U, tj. $U|u_0\rangle = |u_0\rangle$. Tento stav přísluší vlastní hodnotě 1 a žádnou zajímavou informaci z něj nedostaneme. My potřebuje do vlastní hodnoty dostat závislost na r. Konkrétně se nám bude hodit, budeme-li mít vlastní stav, u kterého fáze u jednotlivých bazických stavů budou různé a úměrné indexu daného stavu:

$$|u_1\rangle \equiv \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i k}{r}} \left|x^k \mathrm{mod}\, N\right\rangle. \tag{4.23}$$

Snadno ověříme, že potom $U|u_1\rangle = e^{\frac{2\pi i}{r}}|u_1\rangle$. Všimneme si, že díky zakomponování r jsou fázové rozdíly mezi těmito r stavy výpočetní báze stejné. Nakonec ještě provedeme zobecnění vlastního stavu tak, že fázi těchto stavů vynásobíme celým číslem s, $0 \leq s \leq r-1$:

$$|u_s\rangle \equiv \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i k s}{r}} \left|x^k \mathrm{mod}\, N\right\rangle. \tag{4.24}$$

Opět platí, že se jedná o vlastní stavy U, jelikož

$$U|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i k s}{r}} \left|x^{k+1} \mathrm{mod}\, N\right\rangle = e^{-\frac{2\pi i s}{r}}|u_s\rangle. \tag{4.25}$$

Zdůvodníme si tento poslední krok. Určitě budeme chtít použít algoritmus pro odhad fáze na zjištění r. S tím ovšem přichází dva požadavky - schopnost efektivně implementovat CU^{2^j} pro jakékoliv celé číslo j a schopnost připravit vlastní stav U s netriviální vlastní hodnotou nebo superpozici takovýchto vlastních stavů. První požadavek splníme níže vysvětlenou *modulární exponencializací*. Druhý požadavek by ovšem na první pohled vynucoval znalost r, což nedává smysl. Zde přesně využijeme vyjádření (4.24). Všimneme si, že sečteme-li tyto stavy přes všechny $s \in \{0, ..., r-1\}$, pak různé fáze vyruší všechny bazické stavy kromě $|1\rangle$, tj. platí

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle. \tag{4.26}$$

Stav $|1\rangle$ je tedy superpozicí stavů $|u_s\rangle$ a algoritmus pro odhad fáze proto odhadne fázi $\varphi \approx s/r$, kde s bude náhodně vybrané celé číslo v rozmezí $0 \leq s \leq r-1$. Zvolíme-li v algoritmu pro odhad fáze první registr o velikosti $t = 2L + 1 + \lceil \log\left(2 + \frac{1}{2\epsilon}\right) \rceil$ qubitů a druhý registr bude ve stavu $|1\rangle$, pak pro každé s naměříme s pravděpodobností $(1-\epsilon)/r$ hodnotu $\varphi \approx s/r$ s přesností na $2L+1$ bitů.

Nyní již zbývá z φ určeného na $2L+1$ bitů získat r. Jinými slovy získat zlomek nejbližší φ. K tomu přesně využijeme nám již známý algoritmus řetězových zlomků. Podle věty (7), kam dosadíme $p \equiv s$, $q \equiv r$ a $x \equiv \varphi$, pak totiž dostaneme, že s/r je konvergentem řetězového zlomku pro φ a podle poznámky předcházející této větě může být spočítán za použití $O(L^3)$ operací. Abychom ověřili, že jsou splněny podmínky věty, tak si uvědomíme, že $r \leq N \leq 2^L$, a tedy $|s/r - \varphi| \leq 2^{-2L-1} \leq 1/2r^2$, kde

první nerovnost vyplývá z aproximace φ na $2L+1$ bitů. Po algoritmu řetězových zlomků tedy dostaneme nesoudělná čísla r', s' taková, že $s'/r' = s/r$. Zde r' bude náš kandidát na řád a jeho správnost lze snadno ověřit.

Zaměřme se nyní na druhý požadavek algoritmu pro odhad fáze – schopnost efektivní implementace CU^{2^j}. Proces modulární exponencializace zde nebudeme popisovat do detailu (konstrukce může být nalezena např. v [28]), spíše pouze ukážeme, že lze pomocí reverzibilního výpočtu provést požadovanou transformaci. Ona transformace má tvar

$$\begin{aligned} |z\rangle |y\rangle &\rightarrow |z\rangle U^{z_t 2^{t-1}} ... U^{z_1 2^0} |y\rangle \\ &= |z\rangle \left| x^{z_t 2^{t-1}} \cdot ... \cdot x^{z_1 2^0} y (\mathrm{mod}\ N) \right\rangle \\ &= |z\rangle \left| x^z y (\mathrm{mod}\ N) \right\rangle \end{aligned} \tag{4.27}$$

a vidíme tedy, že při algoritmu pro odhad fáze dochází k násobení druhého registru tzv. *modulární exponenciálou* $x^z (\mathrm{mod}\ N)$, kde z určuje stav prvního registru. Toho lze nejsnadněji dosáhnout tak, že na třetím registru se znalostí reverzibilních operací spočítáme $x^z (\mathrm{mod}\ N)$, čímž poté vynásobíme stav druhého registru.

Proces tedy rozdělíme na dvě fáze. V první fázi pomocí modulárního násobení postupně spočítáme

$$x (\mathrm{mod}\ N) \rightarrow x^2 (\mathrm{mod}\ N) \rightarrow x^4 (\mathrm{mod}\ N) \rightarrow ... \rightarrow x^{2^j} (\mathrm{mod}\ N), \tag{4.28}$$

kde j jde až do $t-1$. Uvážíme-li, že $t = 2L + 1 + \lceil \log\left(2 + \frac{1}{2\epsilon}\right) \rceil = O(L)$ a že každá tato mocnící operace stojí $O(L^2)$ operací základní aritmetiky, pak pro první fázi dostáváme celkově $O(L^3)$ operací.

Ve druhé fázi využijeme toho, že

$$x^z \mathrm{mod}\ N = \left[(x^{z_t 2^{t-1}} \mathrm{mod}\ N)(x^{z_{t-1} 2^{t-2}} \mathrm{mod}\ N) ... (x^{z_1 2^0} \mathrm{mod}\ N) \right] \mathrm{mod}\ N. \tag{4.29}$$

Z toho vidíme, že musíme provést $t-1$ modulárních násobení, každé vyžadující $O(L^2)$ operací. Celkově tak opět získáváme $O(L^3)$ operací (existují efektivnější algoritmy, ale nám tato složitost postačuje). Účinky na našem třetím pomocném registru je pak nutné samozřejmě reverzně zvrátit. Jsme tedy schopni sestavit reverzibilní obvod efektivně vykonávající operaci $(z, y) \rightarrow (z, x^z y (\mathrm{mod}\ N))$. To již implikuje možnost sestavení kvantového obvodu implementujícího transformaci (4.27) s pomocí $O(L^3)$ operací.

Nyní již máme vše připravené a zbývá se podívat na případy, kdy může celý algoritmus na hledání řádu selhat. Samozřejmě to může nastat, selže-li algoritmus pro odhad fáze. To ovšem nastane s pravděpodobností ϵ, kterou můžeme udělat zanedbatelnou, aniž bychom utrpěli na efektivitě algoritmu. Větší problém může způsobit to, že s a r mohou mít společný faktor. Pak totiž výstupem algoritmu řetězových zlomků bude r' faktor r, místo r samotného. Možností, jak se s tímto problémem vypořádat je více, avšak my si zde ukážeme pouze tu nejvýhodnější z hlediska nutnosti použití pouze konstantního počtu pokusů opakování algoritmu.

Myšlenka je zde taková, že zopakujeme algoritmus pro odhad fáze a následný algoritmus řetězových zlomků dvakrát, přičemž obdržíme výsledky r'_1, s'_1 a r'_2, s'_2. Nebudou-li mít s'_1 a s'_2 žádný společný faktor, pak výsledné r mohu získat jako nejmenší společný násobek r_1 a r_2. Přitom pravděpodobnost, že s'_1 a s'_2 nemají společný faktor, lze vyjádřit jako

$$1 - \sum_q p(q|s'_1) p(q|s'_2), \tag{4.30}$$

kde q prochází prvočísla a $p(x|y)$ značí pravděpodobnost, že x dělí y. Vzhledem k tomu, že bude-li q dělit s'_1, pak určitě bude dělit i s_1 a s. Abychom tedy shora ohraničili $p(q|s'_1)$, tak nám stačí shora

ohraničit $p(q|s_1)$. Přitom s_1 je vybráno zcela náhodně od 0 do $r-1$. Určitě platí $p(q|s_1) \leq 1/q$, proto $p(q|s_1') \leq 1/q$ a stejně tak $p(q|s_2') \leq 1/q$. Dosazením do (4.30) tak dostáváme, že s_1' a s_2' jsou nesoudělná s pravděpodobností větší nebo rovnou

$$1 - \sum_q \frac{1}{q^2}. \tag{4.31}$$

Tento výraz bychom ještě rádi nějak ohraničili. Způsobů je více, tak vybereme ten, který se snadno ukazuje. Nejprve pro $x \geq 2$ ukážeme, že

$$\int_x^{x+1} \frac{1}{y^2} dy \geq \frac{2}{3x^2}. \tag{4.32}$$

Platí totiž

$$\int_x^{x+1} \frac{1}{y^2} dy = \frac{1}{x(x+1)} \tag{4.33}$$

Snadno ověříme, že

$$\frac{1}{x(x+1)} \geq \frac{2}{3x^2} \iff \frac{x}{x+1} \geq \frac{2}{3} \tag{4.34}$$

pro $x \geq 2$. Z toho již plyne (4.32). Nyní již snadno ověříme

$$\sum_q \frac{1}{q^2} \leq \sum_{i=2}^{+\infty} \frac{1}{i^2} \leq \frac{3}{2} \sum_{i=2}^{+\infty} \int_i^{i+1} \frac{1}{y^2} dy = \frac{3}{2} \int_2^{+\infty} \frac{1}{y^2} dy = \frac{3}{4}, \tag{4.35}$$

a tedy

$$1 - \sum_q p(q|s_1')p(q|s_2') \geq 1 - \sum_q \frac{1}{q^2} \geq \frac{1}{4}. \tag{4.36}$$

Vidíme, že pravděpodobnost získání správného r je minimálně $\frac{1}{4}$.

Spočtěme nyní celkovou náročnost algoritmu. Vzhledem k tomu, že počáteční Hadamardova transformace vyžaduje $O(L)$ bran a inverzní kvantová Fourierova transformace $O(L^2)$ bran, tak zásadní přírůstek vychází z modulární exponencializace. Ta vyžaduje $O(L^3)$ bran, stejně jako algoritmus řetězových zlomků. Nakonec právě díky námi použité metodě na získání r z r' potřebuje celý tento proces pouze konstantní počet pokusů, proto i celkové nároky jsou řádu $O(L^3)$.

Tímto již máme celý algoritmus na hledání řádu odvozený. Než však přikročíme k faktorizaci, ještě by bylo dobré v krátkosti ukázat trochu jiný pohled na to, co se odehrává během tohoto algoritmu.

Jak už jsme zdůvodnili, do algoritmu pro odhad fáze vstupujeme ve stavu

$$|\psi_0\rangle = \underbrace{|00...0\rangle}_{t} \underbrace{|00...01\rangle}_{L}, \tag{4.37}$$

přitom stav druhého registru je vlastně superpozicí vlastních stavů U. Po vykonání Hadamardovy transformace na prvním registru máme

$$|\psi_1\rangle = \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle |00...01\rangle . \tag{4.38}$$

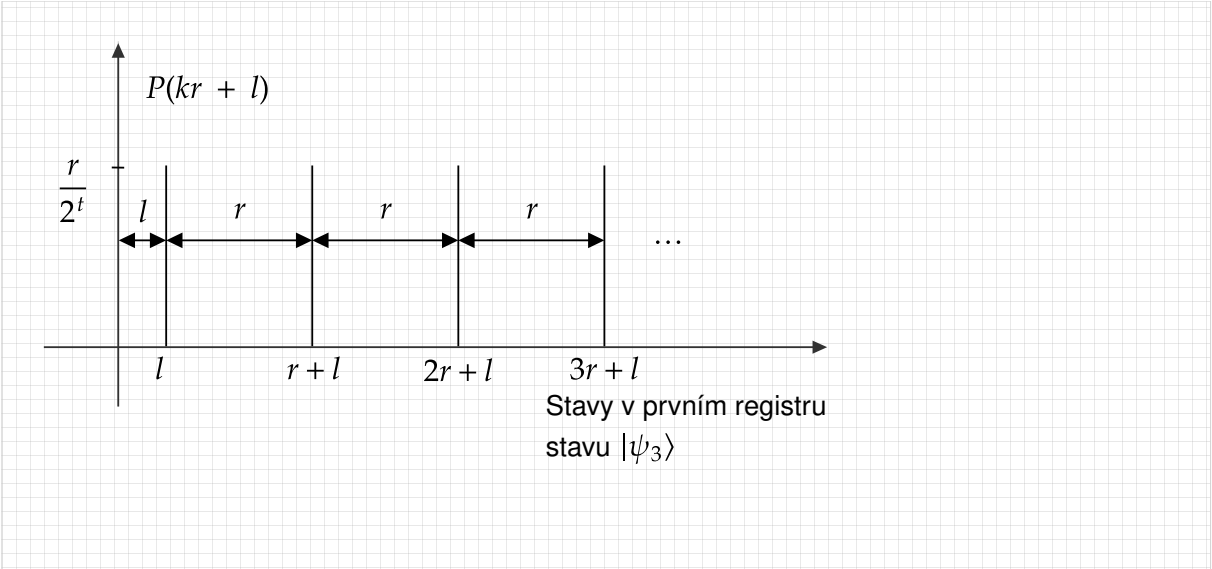
Po modulární exponencializaci dostaneme

$$|\psi_2\rangle = \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} |k\rangle \left|x^k \bmod N\right\rangle . \tag{4.39}$$

Zde je dobré se pozastavit. Vidíme, že stav $|\psi_2\rangle$ je vlastně superpozice všech mocnin x modulo N. Co kdybychom nyní změřili druhý registr? To, co jsme vlastně na začátku této sekce odvodili, je, že funkce $f(k) = x^k \bmod N$ je periodická s periodou r. Uvažujme nyní pro jednoduchost speciální případ, kdy r je mocnina 2 (pak r dělí 2^t). Řekněme, že na druhém registru naměříme stav $|w\rangle$. Pak vlastně kolapsem vlnové funkce v prvním registru zůstane superpozice jen těch hodnot, které odpovídají hodnotě $|w\rangle$ v druhém registru. První z těchto hodnot, l, bude dána tak, že l je nejmenší kladné celé číslo splňující $x^l \bmod N = w$. Tomuto číslu se říká *posuv* a je vidět, že na základě kolapsu po měření může nabývat náhodné hodnoty od 0 do $N-1$. Ostatní členy budou pak jednoznačně dány periodou. Snadno nahlédneme, že počet členů v superpozici na prvním registru odpovídajících naměření $|w\rangle$ na druhém registru bude $2^t/r$. Stav po měření tak můžeme zapsat jako

$$|\psi_3\rangle = \sqrt{\frac{r}{2^t}} \sum_{k=0}^{2^t/r-1} |kr + l\rangle\,|w\rangle\,. \tag{4.40}$$

Problém je zde právě v náhodnosti posuvu l, díky němuž nemůžeme opakováním algoritmu a měřením prvního registru zjistit periodu r. Distribuci pravděpodobnosti znázorňuje obrázek (4.5).



Obrázek 4.5: Rozložení pravděpodobností na prvním registru stavu $|\psi_3\rangle$.

Tento problém přesně řeší inverzní kvantová Fourierova transformace následující po modulární exponencializaci. Ta je definována vztahem (4.2) až na to, že do exponenciály dáme znaménko -.
Aplikujeme-li ji totiž na první registr stavu $|\psi_3\rangle$, dostaneme

$$\begin{aligned} |\psi_4\rangle &= \sqrt{\frac{r}{2^t}} \sum_{k=0}^{2^t/r-1} \left(\frac{2}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} e^{-i2\pi j(kr+l)/2^t}\,|j\rangle \right) |w\rangle \\ &= \frac{1}{\sqrt{r}} \left(\sum_{j=0}^{2^t-1} \left[\frac{r}{2^t} \sum_{k=0}^{2^t/r-1} e^{\frac{-i2\pi jk}{2^t/r}} \right] e^{-i2\pi jl/2^t}\,|j\rangle \right) |w\rangle\,. \end{aligned} \tag{4.41}$$

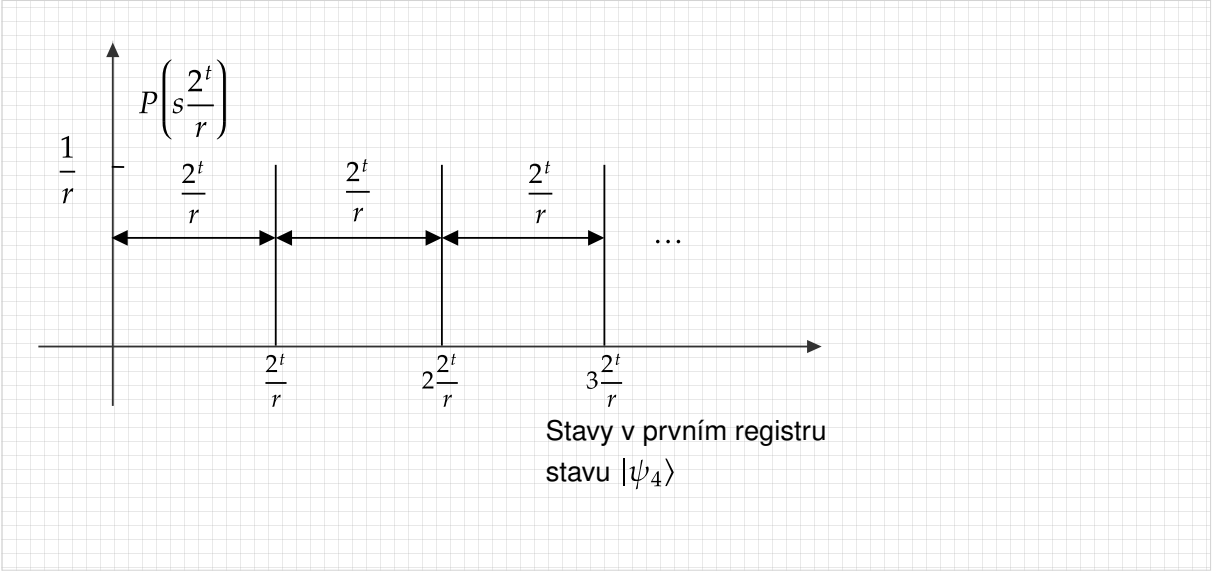
Zde můžeme využít obecné identity

$$\frac{1}{N} \sum_{j=0}^{N-1} e^{i2\pi jk/N} = \begin{cases} 1 & \text{když } k \text{ je násobek } N \\ 0 & \text{jinak (fáze se díky rovnoměrnému uspořádání vyruší).} \end{cases} \tag{4.42}$$

U nás to znamená, že výraz v hranatých závorkách bude nenulový pouze pro $j = s2^t/r$, kde $s \in \{0, ..., r-1\}$. Výsledný stav tak lze zapsat jako

$$|\psi_4\rangle = \frac{1}{\sqrt{r}} \left(\sum_{s=0}^{r-1} e^{-i2\pi \frac{s}{r} l} \left| \frac{s2^t}{r} \right\rangle \right) |w\rangle \,. \tag{4.43}$$

Z toho je vidět, že jsme inverzní kvantovou Fourierovou transformací odstranili škodlivý posuv. Výsledné rozložení pravděpodobnosti je znázorněno na obrázku (4.6).



Obrázek 4.6: Rozložení pravděpodobností na prvním registru stavu $|\psi_4\rangle$.

Po změření prvního registru tedy získáme hodnotu $s2^t/r$ a po vydělení 2^t hledáme s/r. Opět pokud provedeme tento proces dvakrát, dostaneme hodnoty s'_1/r'_1 a s'_2/r'_2. Jsou-li s'_1, s'_2 nesoudělná, dostaneme r. Tím se přesně dostáváme k již vysvětlenému postupu (viz. odvození (4.36)), kdy víme, že tento proces potřebuje pouze konstantní počet pokusů.

Při odvozování tohoto pohledu na celý algoritmus jsme provedli měření druhého registru, které jsme původně nedělali. Je tedy nutné říct, že zde měření nebylo nezbytně nutné a bylo použito pouze k ulehčení vysvětlování. Bez tohoto měření by inverzní kvantová Fourierova transformace byla provedena prostě při superpozici stavů v druhém registru a výstupem by byla superpozice všech možných stavů konzistentních se stavy v superpozici na druhém registru

$$\left|\psi'_4\right\rangle = \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} \left(\frac{1}{\sqrt{r}} \left[\sum_{s=0}^{r-1} e^{-i2\pi \frac{s}{r} j} \left| \frac{s2^t}{r} \right\rangle \right] \right) \left| x^j \bmod N \right\rangle \,. \tag{4.44}$$

Vidíme, že to na měření na prvním registru nemá vliv.

V případě, že by r nebylo mocninou 2, tj. nedělilo by beze zbytku 2^t, pak lze ukázat, že rozložení pravděpodobnosti by bylo velmi mírně „rozmazané"v okolí hodnot $s2^t/r$. Toto rozmazání však nemá podstatný vliv na efektivitu uvedeného postupu.

4.5 Faktorizace

Jak jsme již avizovali, algoritmus na hledání řádu modulo N nám pomůže při hledání prvočíselného rozkladu složeného čísla N. Ve skutečnosti je hledání řádu jediná část faktorizačního algoritmu, kterou je nutno řešit kvantovým výpočtem. Na ostatní části existují efektivní klasické algoritmy.

Vzhledem k našemu již připravenému matematickému aparátu bude nyní velmi snadné ukázat spojitost mezi hledáním řádu a faktorizací. Než k tomu však přistoupíme, ukážeme si jeden z dílčích klasických algoritmů používaných při faktorizaci. Jedná se o algoritmus, který nám pomůže rozeznat, zda-li N není perfektní mocnina.

Nechť N lze zapsat pomocí L bitů. Naším úkolem je určit, jestli

$$N = a^b \tag{4.45}$$

pro nějaké přirozená čísla $a \geq 1$, $b \geq 2$. Předpokládejme nyní, že (4.45) platí a vynechme triviální případ, kdy $a = 1$ (pak nutně $N = 1$). Pak jistě platí $a^b = N \geq 2^L$, a tedy $b \geq L$ (již uvažujeme $a \geq 2$).

Je známý fakt, že násobení lze spočítat s počtem operací menším než $O(L^{1,5})$ (např. Toom-Cook násobení) a logaritmus lze spočítat se složitostí $O(M(L)L^{0,5})$, kde $M(L)$ značí složitost násobení. Výpočet logaritmu tak vyžaduje $O(L^2)$ operací. Ještě připomeneme, že dělení má také složitost $O(L^2)$. S těmito znalostmi dostáváme, že výpočet $x \equiv y/b$, kde $y = \log_2 N$, má složitost $O(L^2)$. Stejně jako u logaritmu dostáváme, že i exponencializace je řádu $O(L^2)$, a tedy i výpočet 2^x. Poté spočteme tomuto číslu nejbližší celá čísla $u_1 = \lfloor 2^x \rfloor$, $u_2 = u_1 + 1$. Nyní umocníme u_1^b, u_2^b (algoritmus využívající opakované mocnění na druhou vyžaduje $O(L^2)$) a zkontrolujeme, jestli se některé z těchto čísel rovná N.

Tento proces opakujeme pro všechna b a víme, že $b \leq L$. Celkově tak tento algoritmus na rozpoznávání perfektních mocnin vyžaduje $O(L^3)$ operací. Spolu s tímto algoritmem budeme při faktorizaci ještě využívat velmi dobře známý Eukleidův algoritmus na hledání nejmenšího společného dělitele, na který jsme již narazili v sekci Matematický aparát. Ten budeme potřebovat, jelikož při hledání řádu x modulo N mají být x a N nesoudělná a nakonec také při hledání $\mathrm{nsd}(x^{r/2} \pm 1, N)$.

Nyní se již podívejme, jak nám hledání řádu pomůže najít prvočíselný rozklad. V prvním kroku si ukážeme, že můžeme najít faktor čísla N, jestliže můžeme najít netriviální řešení rovnice $x^2 = 1 (\mathrm{mod}\ N)$. Přesně to nám říká věta (3). Najdeme-li tedy takovéto x, pak alespoň jedno z čísel $\mathrm{nsd}(x + 1, N)$, $\mathrm{nsd}(x - 1, N)$ bude netriviální faktor N.

Ve druhém kroku si vlastně ukážeme, že u náhodně vybraného y nesoudělného s N je docela velká šance, že bude splňovat podmínky věty (3). Jinými slovy ukážeme, že pro takové y je dosti pravděpodobné, že bude mít sudý řád r a že bude platit $y^{r/2} \neq 1 (\mathrm{mod}\ N)$. Z toho již potom vyplyne, že $x \equiv y^{r/2} (\mathrm{mod}\ N)$ bude netriviální řešení rovnice $x^2 = 1 (\mathrm{mod}\ N)$. Toto tvrzení je shrnuté ve větě (4).

Tím máme Shorův algoritmus završen a jeho shrnutí pro složené číslo N je v následujícím seznamu:

1. Zkontrolujeme sudost. Pokud je N sudé, vrátíme faktor 2.
2. Zkontrolujeme, jestli neplatí $N = a^b$. Pokud ano, vrátíme faktor a.
3. Zcela náhodně vybereme $x \in \{1, ..., N - 1\}$ a Eukleidovým algoritmem zkontrolujeme, že $\mathrm{nsd}(x, N) = 1$. Pokud $\mathrm{nsd}(x, N) > 1$, pak vrátíme faktor $\mathrm{nsd}(x, N)$.
4. Použijeme algoritmus na hledání řádu r, abychom našli řád x modulo N.
5. Pokud je r liché nebo $x^{r/2} = -1 (\mathrm{mod}\ N)$, vrátíme se ke kroku 3. Naopak pokud jsou splněny požadavky věty (3), spočítáme $\mathrm{nsd}(x^{r/2} + 1, N)$ a $\mathrm{nsd}(x^{r/2} - 1, N)$.
6. Otestujeme, jestli alespoň jedno z těchto čísel je netriviálním faktorem N. Pokud není, vracíme se ke kroku 3.

Kapitola 5

Potenciál a první úspěchy NISQ počítačů

V této závěrečné kapitole s využitím [22] shrneme cíle a potenciál kvantového počítání v blízké budoucnosti a nastíníme překážky, které je nutné překonat během vývoje univerzálního kvantového počítače odolného vůči chybám. Nakonec se podíváme na nedávný úspěch, kdy se na kvantovém počítači společnosti Google podařilo při specifické úloze dosáhnout kvantové nadřazenosti [3].

5.1 Proč je kvantové počítání složité

Základ této práce tvoří tři algoritmy, na nichž se demonstrují možnosti kvantových počítačů přesahující možnosti počítačů klasických. Přitom jeden z těchto algoritmů umožňuje efektivně faktorizovat složená čísla, což implikuje možnost prolomení RSA – dnes asi nejpoužívanější šifry s veřejným klíčem. Přesto je tento systém dále hojně používán největšími institucemi i vládami po celém světě. Důvod je prostý. Ač existují teoretické podklady k těmto algoritmům, tak k sestrojení kvantového počítače schopného vykonávat tyto úlohy na požadované úrovni máme ještě daleko.

Proč je tedy fyzická realizace tak náročná? Důvod tkví ve fundamentální vlastnosti kvantového světa - kvantový systém nemůže být pozorován, aniž bychom způsobili kolaps vlnových funkcí a tím ztratili veškeré výhody kvantového výpočtu. Tato náchylnost k narušení procesu vynucuje takřka dokonalou izolaci kvantového systému od okolního světa. Současně ovšem ke zpracování informace potřebujeme, aby spolu qubity silně interagovaly. Tento proces navíc potřebujeme kontrolovat a nakonec musíme být schopni konečný stav qubitů vyčíst. Ukazuje se, že splnit všechny tyto podmínky dohromady je velice náročný úkol.

Zatím dosti vzdáleným cílem je, že budeme schopni zvyšovat výkon kvantových počítačů a přitom udržovat věrnost výpočtů za pomoci kvantové korekce chyb. Základní myšlenkou tohoto přístupu je, že chceme-li uchránit kvantový systém, pak bychom ho měli zakódovat do vysoce provázaného stavu. Prostředí interagující pouze s částí tohoto systému pak není schopné „spatřit"zakódovanou informaci, a tudíž není schopné stav zničit. Problémem kvantové korekce chyb ovšem je, že vyžaduje velké množství dodatečných qubitů. Proto si budeme muset na kvantové počítače využívající tuto technologii ještě nějakou dobu počkat.

5.2 Vstupujeme do NISQ éry

I když je kvantové počítání odolné vůči chybám stále hudbou dosti vzdálené budoucnosti, vstupujeme nyní do nové éry kvantových technologií popsané zkratkou *NISQ - Noisy Intermediate-Scale Quantum*. Zde „intermediate-scale"odkazuje na velikost kvantových počítačů pohybující se od 50 do několika sto-

vek qubitů. Přitom 50 qubitů lze považovat za důležitý milník, protože tím se dostáváme za možnosti dnešních nejvýkonnějších superpočítačů. „Noisy"zde odkazuje na nedokonalou kontrolu nad qubity a fakt, že šum bude hrát u zařízení vyvinutých v nadcházejících letech klíčovou limitující roli.

Přesto však NISQ technologie vzbuzují alespoň ve vědecké obci velké nadšení, především díky možnosti zkoumání fyziky mnoha provázaných částic. Nejsme si jisti, zda-li budou mít již tyto technologie přímé využití i ve světě byznysu, nicméně je třeba je brát jako první kroky k technologiím dostupným v budoucnosti. Tyto budoucí technologie mají potenciál zásadních dopadů na celou společnost, ačkoli nikdo nemůže s jistotou říct, kdy se na tuto úroveň dostaneme.

Počet qubitů jsme uvedli jako hlavní měřítko toho, jak náročné je simulovat kvantový počítač na klasickém počítači. Pro provedení kvantového výpočtu to však není jediný atribut, který nás zajímá. Důležitá je také *kvalita* qubitů a přesnost, s jakou jsme schopni aplikovat kvantové brány. V současné době se u dvouqubitových bran při realizaci pomocí zachycených iontů nebo supravodivých obvodů dosahuje poměru chyb přinejlepším 0.1%. Naivně tak lze očekávat, že u těchto nedokonalých zařízení nebudeme schopni spolehlivě uskutečnit obvod s více než 1000 branami. Toto pak lze řešit právě kvantovou korekcí chyb. Jak již však bylo zmíněno, to vynucuje přidání mnoha dodatečných qubitů. Proto když se bavíme o NISQ počítačích, máme na mysli právě nedokonalá zařízení nechráněná kvantovou korekcí chyb.

Dalším z aspektů, na které je třeba při konstrukci kvantových počítačů koukat, je rychlost, s jakou můžeme vykonat kvantovou bránu. Zde je dobré zmínit, že supravodivé obvody jsou takřka tisíckrát rychlejší než kvantové procesory využívající iontové pasti. Nakonec je nutné qubity spolehlivě změřit. Pravděpodobnost chyby při měření se u supravodivých obvodů pohybuje okolo 1%. V neposlední řadě pak také hraje roli naše schopnost spolehlivě qubity připravit.

5.3 Potenciální oblasti využití v dohledné době

Ptáme-li se, v jaké oblasti a kdy budou mít kvantové počítače široké využití, musíme si nejprve položit otázku: Kdy budou kvantové počítače schopné vyřešit klíčové problémy rychleji než klasické počítače a o jaké problémy se jedná? Když mluvíme o zrychlení, máme tím na mysli porovnání s nejlepším možným klasickým algoritmem běžícím na nejlepším aktuálním hardwaru. Do toho je nutno započítat rychlost, s jakou se výkon klasických počítačů zvětšuje.

Čistě z akademického hlediska je dosažení kvantové nadřazenosti nehledě na úlohu, při které se jí dosáhne, samo o sobě velmi zajímavý a důležitý milník. Nicméně z komerčního hlediska je nutné hledět na užitečnost dané úlohy. Kvantové počítače příštích několika let budou zařízení určená k velmi speciálním účelům a kvantová nadřazenost bude pro investory zajímavá ne díky své vnitřní důležitosti, nýbrž jako krok k pozdějším hodnotnějším aplikacím. Nastiňme si nyní konkrétní oblasti, kde by kvantové počítače mohly mít uplatnění.

5.3.1 Kvantové optimalizátory

Ačkoli se neočekává, že by kvantové počítače byly schopné efektivně řešit NP-těžké problémy jako například kombinatorické optimalizační problémy, lze si představit, že budou schopné najít lepší přibližné řešení, popř. budou schopné najít přibližné řešení rychleji. Existují i takové problémy, pro které samotné hledání přibližného řešení patří mezi NP-těžké problémy, chceme-li mít výsledek dostatečně přesný. V těchto případech také neočekáváme, že by kvantové počítače našly přibližné řešení efektivně. Ovšem jsou i takové problémy, u nichž je velká propast mezi aproximací získanou aktuálně nejlepším algoritmem a hranicí NP-těžkosti. V některých z těchto případů by nebylo překvapivé, kdyby kvantové počítače dokázaly najít přibližné řešení lépe a rychleji, což by mohlo mít velmi užitečné aplikace.

Nicméně musíme uznat, že i kdyby se takováto potenciální aplikace našla, tak není jisté, že by NISQ technologie byla dostačující k průkazné demonstraci. Co se ovšem rozvíjí jako obecně uznávané schéma pro řešení optimalizačních problémů na kvantových zařízeních blízké budoucnosti, je hybridní kvantově-klasický algoritmus. Zde kvantový procesor připraví n-qubitový stav, který je následně změřen. Klasický optimalizátor poté zpracuje výsledky měření a instruuje kvantový procesor, jak pozměnit připravovaný stav. Tento proces se mnohokrát zopakuje, až pomalu zkonverguje do kvantového stavu, ze kterého je možné získat přibližné řešení. Při aplikaci na klasické kombinatorické problémy se tato procedura nazývá *QAOA - Quantum Approximate Optimization Algorithm*. Naopak při aplikaci na kvantové problémy se tomuto přístupu říká *VQE - Variational Quantum Eigensolver*.

5.3.2 Kvantové žíhání

I když se zde bavíme o milníku 50 až 100 qubitů, tak ve skutečnosti již existuje zařízení pracující s 2000 qubity nazývané D-Wave 2000Q. Nejedná se však o zařízení založené na kvantových obvodech, nýbrž o tzv. *kvantový žíhací počítač*. Tyto stroje řeší optimalizační problémy jinak než spuštěním obvodu a je nutné dodat, že je často řeší úspěšně.

Není však zatím prokázáno, že by kvantové žíhací počítače dokázaly urychlit řešení oproti nejlepším klasickým počítačům. Kvantové žíhání je vlastně nedokonalá verze toho, čemu se obecně říká *kvantové adiabatické počítání*, pro které existuje (v případě dokonalých qubitů) teoretický argument naznačující, že je stejně dobré jako kvantové počítání založené na kvantových obvodech. Ten je však založen na potřebě přidání mnoha qubitů. Navíc nemáme (na rozdíl od počítačů založených na kvantových obvodech) teoretický argument dokládající škálovatelnost kvantových žíhacích počítačů.

Zatím byly tyto stroje aplikovány především na případy, kdy je pro klasické počítače relativně snadné simulovat tento proces. V dohledné době však mají přijít kvantové žíhací počítače, které nebudou moci být simulovány klasickými počítači a které mají větší potenciál urychlení daných problémů.

5.3.3 Kvantové hluboké učení

Strojové učení je v poslední době velmi rychle se rozvíjející obor a je tedy přirozené se zamýšlet nad jeho kombinací s kvantovými technologiemi. Omezme nejprve naše úvahy nad kvantovým hlubokým učením na tzv. *omezený Boltzmannův stroj*. Ten může být považován za spinový systém v tepelné rovnováze při nízké avšak nenulové teplotě s mnoha skrytými vrstvami spinu oddělujícími vstup a výstup. Systém může mít miliony vazebných parametrů optimalizovaných během tréninkové fáze pro dosažení požadované společné distribuce pravděpodobnosti pro vstup a výstup. Tréninková fáze může být pod dohledem a tato síť může být naučena například k rozpoznaní označené sady obrázků. Naopak, není-li pod dohledem, pak může být síť využita k rozpoznávání vzorců v neoznačených datových souborech. Může se ukázat, že kvantová analogie tohoto přístroje, kde by spiny byly qubity namísto bitů, by měla oproti klasické síti výhody. Mohla by se dát kupříkladu snadněji vytrénovat k jistým účelům. Zatím to ovšem s jistotou nevíme.

Díky konceptu *QRAM – Quantum Random Access Memory* však máme důvod k mírně optimistickému postoji k využití kvantového hlubokého učení. QRAM je vlastně reprezentace velkého souboru dat zakódováním vektoru s N komponentami do $\log N$ qubitů. Tento přístup s sebou samozřejmě nese mnoho úskalí jako např. náročnost tohoto zakódování nebo fakt, že z výsledného měření získáme maximálně $\log N$ bitů. Tyto překážky vyvstávají, když chceme učit kvantovou síť o korelacích v klasických datech. Bylo by tedy možná lepší uvažovat o případech, kdy vstupem i výstupem je kvantový stav. Je tedy dost možné že nejlepší využití by kvantové sítě našly v kvantových úlohách jako např. kontrola komplexního kvantového stavu.

5.3.4 Kvantová inverze matic

Existuje tzv. HHL algoritmus [16], který má na vstupu reprezentaci velké, dostatečně řídké a dobře podmíněné $N \times N$ matice A a vektor s N složkami $|b\rangle$ zakódovaný v QRAM pomocí $\log N$ qubitů. Výstupem algoritmu je stav $\left|A^{-1}b\right\rangle$ a jeho složitost je $O(\log N)$, což je exponenciální urychlení oproti klasické inverzi matic.

Jsou zde však podmínky, které musí matice splňovat a navíc nesmíme zapomenout na cenu zakódování vstupních dat do QRAM, která může veškeré urychlení anulovat. Toto by šlo teoreticky vyřešit výpočtem $|b\rangle$ přímo v kvantovém počítači místo načítáním ho z databáze.

Důvodem, proč si myslíme, že je tento algoritmus velmi silný, je, že řeší tzv. *BQP-kompletní* problém. To znamená, že každý problém, který lze efektivně řešit na kvantovém počítači, může být přetvořen speciální instanci této inverze matic. Tato inverze matic pak lze využít např. pro vyřešení rovnic elektromagnetismu v komplexní trojrozměrné geometrii za účelem optimalizování výkonu antény.

Lze předpokládat, že HHL algoritmus bude mít jednou velký dopad, ale pravděpodobně to nebude vzhledem k náročnosti v NISQ éře.

5.3.5 Kvantové doporučovací systémy

Nedávno byl představen kvantový algoritmus [19] umožňující exponenciální zrychlení při úloze poskytování vysoce hodnotných doporučení oproti nejlepšímu tou dobou známému klasickému algoritmu. Nedlouho poté byl však představen klasický algoritmus inspirovaný tímto algoritmem řešící tuto úlohu ve stejném čase.

I tak však stojí tento problém (nebo spíše jeho zjednodušená verze) za bližší představení. Mějme m zákazníků a n produktů a nechť P je $m \times n$ preferenční matice taková, že $P_{ai} = 1$, pokud se zákazníkovi a produkt i líbí a naopak $P_{ai} = 0$, pokud se zákazníkovi a produkt i nelíbí. V normálním světě se pohybujeme v řádech $m \approx 10^8$, $n \approx 10^6$, ale hodnost matice k bývá docela malá, pohybující se okolo $k \approx 100$. To implikuje pouze omezený počet typů zákazníků, a proto jakmile známe pouze pár preferencí nového zákazníka, tak již dokážeme doporučit další produkty.

Celý algoritmus má dvě fáze. V první (offline) se vytvoří aproximace preferenční matice P s nízkou hodností a ve druhé (online) nový zákazník udá některé jeho preference a systém mu jako výstup vrátí doporučení produktu, který se zákazníkovi bude s vysokou pravděpodobností líbit. Právě onu druhou fázi dokázal kvantový algoritmus urychlit oproti tehdy nejlepšímu klasickému algoritmu.

5.3.6 Kvantové simulace

Již Richard Feynman tvrdil, že simulace vysoce provázaných kvantových systémů je oblast, kde kvantové počítače budou mít rozhodující výhodu nad klasickými. Domníváme se, že simulovat vysoce provázanou hmotu je těžký výpočetní problém. V této oblasti by kvantové počítače v dlouhodobém horizontu mohly mít obrovský dopad na celý svět. Kvantové simulace mohou mít zásadní dopad např. při vývoji léků nebo při tvorbě nových materiálů umožňujících kupříkladu efektivnější přenos energie či zlepšení procesu získávání solární energie. Aktuálně si nejspíše ani neumíme představit na jaké všechny oblasti mohou mít kvantové simulace dopad. Opět je ale nutné podotknout, že se zde pravděpodobně nebavíme o cílech dosažitelných v NISQ éře. Kvantové algoritmy určené k simulaci velkých molekul či exotických materiálů totiž budou příliš náročné, než aby je bylo možné aplikovat bez kvantové korekce chyb.

Oblast obzvláště náročná pro klasické počítače je simulace kvantové dynamiky, tj. předpověď časového vývoje vysoce provázaného kvantového stavu. Zde mají kvantové počítače nespornou výhodu a v tomto směru se očekávají výsledky již od NISQ technologií. V této souvislosti je dobré zmínit velmi

rychlý pokrok v teorii chaosu v šedesátých a sedmdesátých letech poté, co bylo možné chaotické dynamické systémy simulovat pomocí klasických počítačů. Lze si představit, že podobný vývoj může přijít i v našem porozumění kvantovému chaosu poté, co budeme schopni simulovat chaotické kvantové systémy.

Již nyní si však můžeme uvést velmi aktuální úspěch Google AI Quantum v oblasti kvantové chemie [4]. S využitím VQE simulovali vazebnou energii vodíkových řetězců délky až 12 a dále proběhla simulace dvou způsobů izomerizace diazénu. Zásadní roli v tomto experimentu hrály metody určené ke zmenšení chyb ve výpočtu. Celkově lze shrnout, že cílem experimentu bylo na kvantovém počítači implementovat tzv. Hartree-Fockovu metodu, což je způsob získání co nejpřesnější orbitalové funkce za předpokladu, že každý elektron cítí průměrný potenciál generovaný všemi ostatními elektrony.

Je známým faktem, že je možné vyjádřit operaci rotace báze aplikovanou na kvantový stav jako časovou evoluci generovanou hamiltoniánem neinteragujících fermionů. Máme-li tedy výpočetní bázi $\varphi_p(r)$ tvořenou stavy $|\eta\rangle = a_\eta^\dagger ... a_1^\dagger |0\rangle$ (kde $a_p^b\dagger$ resp. a_p jsou kreační resp. anihilační operátory tzv. základní orbitalové funkce), pak antisymetrický součinnový stav $|\Psi_\kappa\rangle$ v nové bázi $\tilde{\varphi_p}(r)$ dané jako

$$\tilde{\varphi_p}(r) = \sum_{q=1}^{N} [e^\kappa]_{pq} \varphi_p(r), \tag{5.1}$$

kde κ je $N \times N$ antihermitovská matice, můžeme vyjádřit jako

$$|\Psi_\kappa\rangle = U_\kappa |\eta\rangle , \tag{5.2}$$

kde

$$U_\kappa = \exp\left\{ \sum_{p,q=1}^{N} \kappa_{pq} a_p^\dagger a_p \right\}. \tag{5.3}$$

Takovýto stav se nazývá Slaterův determinant. U_κ lze implementovat pomocí tzv. Givensových rotací, které můžeme sestavit za pomoci dvou $\sqrt{iSWAP}$ bran a tří R_z bran. Hartree-Fockův stav pak získáme postupnou optimalizací parametrů κ pro minimalizaci energie. Přitom počáteční κ je určeno klasickým řešením Hartree-Fockových rovnic. Tato idealizovaná implementace VQE takto umožňuje zmírnění koherentních chyb. Aplikujeme tedy U_κ na $|\eta\rangle$ pomocí kvantového počítače a následně optimalizujeme κ s využitím klasických algoritmů.

Řetězec N vodíků byl simulován pomocí N qubitů (které díky omezujícím podmínkám simulovali 2N spinů). Konkrétně byly simulovány řetezce délky 6, 8, 10 a 12. Pro simulaci izomerace diazénu bylo pak použito 10 qubitů. Cílem zde pak bylo vyřešit energetický rozdíl mezi dvěma tranzitními stavy dvou konkurenčních mechanismů - přechod z cis konfigurace do trans konfigurace buď přechodem v rovině nebo mimo rovinu. To vyžadovalo přesnost okolo 40 milihartree (hartree jednotka energie úměrná elektrické potenciální energii atomu vodíku v základním stavu, přibližně rovna 27 eV).

Na naměřených datech byly použity dvě metody na zmírnění chyb – dodatečný výběr na počtu částic a projekce čistého stavu (bližší specifikace těchto metod je mimo rámec této práce). Výsledkem je, že pro 6 a 8 qubitů dosáhla data po VQE tzv. chemické přesnosti a i pro 12 qubitů byla data velmi blízká očekávaným hodnotám.

Simulace dvou způsobů izomerace diazénu byl vůbec první případ, kdy byl mechanismus chemické reakce modelován na kvantovém počítači. Data optimalizovaná pomocí VQE simulovala devět bodů podél reakční koordináty pro rotaci vodíku okolo dusíku v rovině a mimo rovinu. Opět zde k určení počátečních parametrů sloužilo klasické řešení Hartree-Fockových rovnic. VQE pak poskytlo jedno-částicové redukované matice hustoty s přesností větší než 0.98 (za využití již zmíněných metod ke zmírnění chyb).

Ač se při tomto experimentu nedosáhlo kvantové nadřazenosti (všechny výpočty by byly proveditelné na současných klasických počítačích), tak jednak demonstroval prozatím největší simulaci kvan-

tové chemie (do této doby bylo využito maximálně 6 qubitů) a dále poskytl velmi užitečné přístupy pro další bádání v této oblasti.

5.4 Další kvantové technologie

Kromě kvantových počítačů umožňujících efektivnější řešení jistých úloh existují i jiné kvantové technologie, které nelze od kvantového počítání tak úplně oddělit. Mnohdy je u nich potřeba překonat podobné technologické výzvy, jindy zase potřeba jejich realizace plyne přímo z existence kvantových počítačů a následné nedostatečnosti současných technologií.

Příkladem druhého uvedeného důvodu může být kryptografie odolná vůči kvantovým počítačům. Současné kryptosystémy s veřejným klíčem budou s příchodem dostatečně výkonných kvantových počítačů zastaralé. Bude tedy třeba nahradit tyto kryptosystémy novými, které dokážou čelit útoku pomocí kvantového počítače. Takové systémy mohou být samy založeny na kvantových principech.

Navazujícím příkladem jsou kvantové sítě. Ty mohou sloužit kupříkladu k distribuci kvantového provázaní, a tedy i tajného klíče. Globální kvantová síť však může být užita i k dalším účelům jako třeba sdílení informací mezi kvantovými zařízeními.

Za zmínku také stojí fundamentální schopnost kvantových zařízení generovat náhodnost. Tato vlastnost kvantové fyziky může být využita na vytvoření řetězce prokazatelně náhodných bitů i v případě, že nedůvěřujeme zařízení použitému pro tento účel. Na to stačí předpokládat, že prostorupodobně oddělené strany spolu nemohou komunikovat. Prokazatelná náhodnost má mnoho potenciálních využití počínaje zabezpečením komunikačních protokolů, přes objektivní statistický výběr až po simulace Monte Carlo.

Jako další bychom zde zmínili kvantové snímání. Kvantová zařízení mohou snímat slabé síly s větší citlivostí a lepším prostorovým rozlišením než jiné technologie. Proto kvantové snímání může mít již v blízké době velice zásadní dopady např. v medicíně.

Nakonec se blíže podívejme na úlohu, která se ukazuje jako vhodná pro demonstraci kvantové nadvlády a která lze realizovat na specifických kvantových zařízeních odlišných od univerzálních kvantových počítačů – zde konkrétně na fotonickém kvantovém počítači. Jedná se vzorkování bosonů. Při vzorkování bosonů se k výpočtu pravděpodobnostního rozdělení využívá maticová funkce zvaná permanent. Ta není efektivně řešitelná na klasickém počítači. Problém se vzorkováním bosonů ve své základní podobě je, že nejsme schopni dostatečně kvalitně připravit jednofotonové stavy světla.

Teoretické řešení tohoto problému poskytli Dr. Craig Hamilton a prof. Igor Jex z ČVUT FJFI spolu s partnery z University v Paderbornu. Ti ve své práci [15] ukázali, že i při použití stlačených Gaussovských stavů světla zůstane úloha výpočetně extrémně náročná. K výpočtu pravděpodobnostního rozdělení se zde místo permanentu využívá jiná maticová funkce – tzv. Hafnián.

Tohoto výsledku se dále využívá v práci [23], kde se ke vzorkování uvažují prahové detektory bez rozlišení počtu fotonů. Zde k popisu pravděpodobnosti naměření daného výstupu zavádějí maticovou funkci zvanou Torontonián, který je v jistém smyslu analogií Hafniánu.

Toho již využívají v experimentu nedávno uskutečněném výzkumným týmem Čínské univerzity vědy a technologie [29]. Jednalo se o celkově druhé prokázání kvantové nadvlády hned po experimentu společnosti Google popsaném v poslední sekci. Realizován byl na fotonickém kvantovém počítači zvaném Jiuzhang. Experiment spočíval ve vyslání 25 dvoumódových stlačených stavů (ekvivaleních 50 jednomódovým stlačeným stavům) do stomódového speciálně upraveného interferometru. Vzorkování výstupu se pak provádělo na 100 vysoce efektivních jednofotonových detektorech. Jiuzhang generoval až 76 výstupních fotonových „kliknutí", což dává výstupní stavový prostor dimenze až 10^{30}. Pro nejlepší klasické superpočítače je nemožné ukládat amplitudy pravděpodobnosti v takto rozsáhlém prostoru. Celkově byla zde dosažená vzorkovací frekvence rychlejší než dosud nejlepší simulační strategie a superpočítače o faktor úměrný 10^{14}.

5.5 Cesta za škálovatelností

Jak již bylo zmíněno, v NISQ éře několika následujících let není reálné dosáhnout kvantového počítače odolného vůči chybám, který by byl schopný řešit těžké problémy. Stěžejním prvkem ke škálovatelnosti kvantových počítačů je naše schopnost implementovat kvantovou korekci chyb. Proto bude v následujících letech klíčové zaměřit se mimo jiné také na vývoj lepších metod a hardwaru pro implementaci této technologie. Očekává se, že během několika příštích let bude poprvé laboratorně demonstrována vylepšená kontrola qubitu podléhajícímu kvantové korekci. Mimo to by také měly být stále vyvíjeny techniky na zmírnění šumu.

Nicméně, abychom byli schopni spustit algoritmy využívající tisíce chráněných qubitů, tak budeme potřebovat celkový počet qubitů pohybující se v řádech milionů. Je tedy jasné, že na reálnou možnost faktorizovat tisíce bitů dlouhé číslo si budeme muset ještě počkat. Překonání propasti mezi stovkami a miliony qubitů bude vyžadovat ještě hodně času a úsilí, ale panuje obecné přesvědčení, že se na tuto metu dostaneme.

Do té doby budeme však muset během NISQ éry vyřešit mnoho základních věcí. Jednou z nich je razantně snížit chybovost kvantových bran. S přesnějšími branami budou kvantové počítače i bez kvantové korekce chyb schopné spustit delší obvody, a tedy i náročnější algoritmy. Navíc s příchodem kvantové korekce nebude potřeba tolik dodatečných qubitů. Tento přístup se snaží aplikovat například společnost Microsoft ve svém programu zaměřeném na topologické kvantové počítání.

Celkově můžeme shrnout, že při cestě za plně škálovatelnými kvantovými počítači, které budou odolné vůči chybám, musíme překonat důležité výzvy. Vzhledem k tomu, jak dlouhá je před námi cesta, tak každý nový objev či inovace může mít podstatný dopad na směřování tohoto oboru.

5.6 Dosažení kvantové nadřazenosti

Na závěr si zde ukážeme nedávný úspěch společnosti Google demonstrující experimentální realizaci kvantové nadřazenosti pro specifickou výpočetní úlohu. Bylo toho dosaženo pomocí kvantového procesoru Sycamore schopného pracovat s 53 programovatelnými supravodivými qubity. Výzkumníci tvrdí, že tato úloha by na současném nejlepším klasickém superpočítači trvala přibližně deset tisíc let.

Představme si nyní tuto úlohu. Jedná se o výpočetní úkol vzorkování výstupu pseudonáhodného kvantového obvodu. Jde vlastně o vytvoření kvantového chaosu. Výstupem vzorkování je soubor řetězců bitů. Tyto řetězce mají díky kvantové interferenci rozdělení pravděpodobnosti připomínající skvrnitý vzor intenzity produkované interferencí světla v laserovém rozptylu. V důsledku toho mají některé řetězce mnohem větší pravděpodobnost výskytu v souboru než jiné. Právě výpočet tohoto rozdělení pravděpodobnosti je úloha, jejíž náročnost roste pro klasické počítače exponenciálně s přibývajícím počtem qubitů a bran aplikovaných v rámci obvodu.

Pro verifikaci správného fungování procesoru se zde využívá metoda zvaná *cross-entropy benchmarking*. Při té se porovnává četnost experimentálně naměřených řetězců s pravděpodobností vypočtenou pro ideální obvod pomocí simulace na klasickém počítači. Pro daný obvod vypočteme jeho přesnost danou touto metodou, $\mathcal{F}_{XEB}$, následovně. Vezmeme naměřené řetězce $\{x_i\}$ a spočteme střední hodnotu (klasickou simulací vypočtených) pravděpodobností výskytu řetězce x_i, $P(x_i)$, přes všechny naměřené řetězce. Vzorec pro výpočet je

$$\mathcal{F}_{XEB} = 2^n \langle P(x_i) \rangle_i - 1, \tag{5.4}$$

kde n je počet qubitů. Z toho je vidět, že nenastanou-li v kvantovém obvodu žádné chyby, bude $\mathcal{F}_{XEB} = 1$. Naopak vzorkování přes rovnovážné rozdělení pravděpodobnosti dá $\mathcal{F}_{XEB} = 0$. Hodnoty mezi 0 a 1 pak odpovídají pravděpodobnostem výskytu chyby během vykonávání obvodu.

Z nutnosti výpočtu $p(x_i)$ na klasickém počítači je jasné, že tento přístup nemůže fungovat v „režimu nadřazenosti", tj. pro dostatečně velký počet qubitů a dostatečně mnoho bran. V tomto případě se využívá metod zjednodušení či rozdělení obvodu tak, aby bylo možné provést simulace na klasickém počítači. Tyto metody poskytují kvantitativní odhad přesnosti práce procesoru i při plném vytížení.

Cílem tedy je dosáhnout dostatečně velké hodnoty $\mathcal{F}_{XEB}$ pro obvod s dostatečně mnoha qubity a dostatečnou hloubkou (počtem bran), kdy výpočet této úlohy na klasickém počítači by byl neúnosně náročný. Tento úkol je velmi náročný z důvodu, že každá chyba během výpočtu může posunout onen zmíněný skvrnitý vzor, což vyústí v $\mathcal{F}_{XEB}$ blízké nule. Proto základem úspěchu je procesor schopný vykonávat tento program s dostatečně malou chybovostí.

Podívejme se proto pouze v rychlosti, jakým způsobem v Googlu sestavili jejich procesor pojmenovaný Sycamore. Ten je složen z dvojrozměrného pole transmonů (typ fyzikální realizace qubitu využívající supravodivosti), kde je každý laditelně spojen se svými čtyřmi sousedy. Transmony mohou být považovány za nelineární supravodivé rezonátory o frekvenci 5-7 GHz. Qubit je pak zakódován jako dva nejnižší vlastní stavy rezonančního obvodu. Spojení mezi qubity je provedeno nastavitelnou spojkou umožňující ladit spojení qubit-qubit od 0 (vypnuto) do 40 MHz. Vzhledem k tomu, že jeden qubit nefungoval, tak Sycamore využíval zmíněných 53 qubitů a 83 spojek.

Největší náhodný kvantový obvod v experimentu byl tedy pro 53 qubitů a sestával z 1113 jednoqubitových bran, 430 dvojqubitových bran a měření pro každý qubit. Přesnost tohoto obvodu byla určena na 0,2 %. S touto přesností potřebujeme řádově pár milionů měření vzhledem k tomu, že nejistota u $\mathcal{F}_{XEB}$ je $1/\sqrt{N_s}$, kde N_s je počet vzorků.

Jak se provádí klasický výpočet? Do 43 qubitů se využívá Schrödingerův algoritmus simulující vývoj celého kvantového stavu. K tomu byl využit superpočítač v německém Jülichu. Nad 43 qubitů už přichází nedostatek RAM na uložení kvantového stavu. Pro větší počet qubitů se využívá hybridní Schrödingerův-Feynmanův algoritmus, který byl spuštěn na datacentrech Googlu. Při tom se vypočítávají amplitudy pravděpodobnosti jednotlivých řetězců. Během procesu je vlastně obvod rozdělen na dvě části qubitů, z nichž každá je efektivně simulována pomocí Schrödingerovy metody. Poté jsou spojeny za využití přístupu podobného Feynmanovu dráhovému integrálu. Tato metoda, přestože je přívětivější k paměti, se stává exponenciálně výpočetně náročnější s přibývající hloubkou obvodu díky exponenciálnímu růstu počtu drah s počtem bran spojujících tyto části.

K odhadu klasické výpočetní náročnosti v režimu nadřazenosti se spouštějí části simulací kvantového obvodu jak na Summitu (současný nejvýkonnější superpočítač), tak i na clusterech Googlu a celková výpočetní cena je pak extrapolována. Na Summitu nelze z důvodu nedostatku paměti simulovat obvody s velkou hloubkou. Při výrazném omezení nároků byl konečný časový odhad pro vzorkování tří milionů bitových řetězců s přesností 1 % stanoven na jeden rok.

Na Google Cloud serverech bylo odhadnuto, že bez těchto omezení a při přesnosti 0.1 % by se cena vyšplhala na 50 bilionů hodin při jednom jádře a spotřebovalo by se okolo petawatthodiny energie. Pro srovnání vzorkování obvodu 3000000krát vezme kvantovému procesoru 600 sekund, kde je čas vzorkování navíc limitován hardwarovou komunikací. Ve skutečnosti je čistý čas vyžadovaný procesorem pouze 30 sekund.

Zde je nutné dodat, že krátce po publikování výsledků experimentu byl odhad 10 000 let napaden vědci z IBM. Ti ve zkratce tvrdí, že přístup, kdy místo Schrödingerovy metody je díky nedostatku RAM využita Schrödingerova-Feynmanova metoda, není nutný. Jejich přístup by se dal shrnout tak, že místo ukládání celého stavového vektoru s 2^{53} amplitudami do RAM lze tyto hodnoty ukládat na disku a načítat je z něj. Takto se potřebný čas zkrátí na 2,5 dne a navíc lze dosáhnout lepší přesnosti než u kvantového procesoru. I přesto je však důvod se domnívat, že tento experiment Googlu byl důležitým úspěchem na cestě ke kvantové nadřazenosti.

Závěr

V této práci jsme se zaměřili na problematiku kvantového počítání jako celku a detailně si popsali zásadní kvantové algoritmy. V první kapitole jsme představili samotné stavební kameny této oblasti - qubity a kvantové brány. V rámci celé práce jsme s těmito pojmy pracovali jakožto s abstraktními objekty nezávislými na způsobu realizace. Brali jsme tedy v úvahu idealizovaný případ a nezapočítávali do našich myšlenek v reálném světě nezanedbatelný šum. Ukázali jsme si, jak s těmito základními koncepty pracovat a jak s jejich pomocí dosáhnout libovolné unitární transformace. S takto vybudovaným aparátem jsme mohli přistoupit ke konstrukci složitějších obvodů. Přitom jsme se zaměřili na ty algoritmy, které dokážou jistou úlohu řešit efektivněji než nejlepší známé klasické algoritmy.

Prvním z této řady byl díky své snadné konstrukci Deutsch-Jozsův algoritmus. Ten mnohdy spolu se superhustým kódováním a kvantovou teleportací slouží jakožto vstupní bod do kvantových algoritmů. Velmi názorně na něm lze demonstrovat sílu paralelního výpočtu a finální účinky konstruktivní resp. destruktivní interference. Jak jsme ale již zmínili, stále není jasné, jestli vůbec existuje nějaké reálné využití tohoto algoritmu.

Tím jsme se dostali k prvnímu algoritmu s potenciálem reálného využití. Groverův vyhledávací algoritmus představuje velmi elegantní způsob vyhledávání v nesetříděné databázi. Byl zde kladen opravdu velký důraz na geometrickou interpretaci tohoto algoritmu. Jelikož optimální počet Groverových iterací k nalezení řešení silně závisel na samotném počtu řešení, ukázali jsme si také metodu, jak tento počet zjistit - *kvantové počítání*. Nakonec byl představen koncept kvantových procházek a v případě úplného grafu a grafu typu hvězda ukázána ekvivalence s Groverovým algoritmem.

Ve čtvrté kapitole byl popsán Shorův faktorizační algoritmus. K tomu bylo nutné si připravit patřičný matematický aparát. Ten jsme následně využili především k ukázání spojitosti mezi hledáním řádu a samotnou faktorizací. Klíčovou roli při hledání řádu hrál algoritmus pro odhad fáze, jehož základem je kvantová Fourierova transformace. Tento algoritmus má více aplikací a my jsme ho využili krom Shorova algoritmu také v rámci *kvantového počítání*. Mimoto byl prezentován také druhý pohled na problém hledání řádu, který graficky lépe znázorňoval úlohu kvantové Fourierovy transformace. S takto připraveným matematickým aparátem a odvozeným algoritmem na hledání řádu již bylo snadné dát dohromady celkový faktorizační algoritmus.

Závěrem práce bylo shrnutí současné fáze vývoje kvantové informatiky a její další ubírání. Přitom byly nastíněny možné budoucí aplikace kvantových počítačů a překážky, které bude nutné v následujících letech překonat. Mimoto jsme ve stručnosti představili další kvantové technologie s velkým potenciálem. Posledním zmíněným tématem bylo nedávné dosažení kvantové nadřazenosti na stroji společnosti Google při velmi specifické úloze. Osobně se domnívám, že ač se nejednalo o takové urychlení, jaké bylo původně prezentováno výzkumníky Googlu, tak to byl velký úspěch a krok kupředu v oblasti kvantové informatiky.

Ačkoli práce byla ze své podstaty rešerše, při samotném programování byl jistý prostor pro vlastní iniciativu. Při hledání optimálního způsobu podání daného problému jsem se opíral také o grafické znázornění ať už schématické či jako výstup z Qiskitu. Vzhledem k tomu, že práce není koncipována

primárně jako programátorská, tak nejsou její součástí samotné zdrojové kódy, ale jen jejich výstupy. Celkově se tak, myslím, podařilo vytvořit ucelený obrázek o fungování nejzásadnějších kvantových algoritmů a o odvětví kvantového počítání jako takovém.

Samotný výběr tohoto tématu byl motivován cílem vybudovat si co nejširší základy, na kterých bude možno stavět v dalším bádání. Do budoucna se tímto otevírá mnoho možností, jak na toto téma navázat. Mezi tyto možnosti zcela určitě patří např. fyzická realizace kvantových počítačů, kvantová korekce chyb, topologické kvantové počítání, kvantová teorie informace a další. Vzhledem k rostoucímu zájmu o toto odvětví i ze strany soukromého sektoru bude značná poptávka po nových přístupech a inovativních myšlenkách. Přitom každá taková myšlenka může mít zásadní vliv na vývoj celého odvětví.

Literatura

[1] Abraham, H. et al. *Qiskit: An Open-source Framework for Quantum Computing [počítačový program]*. Ver. 0.23.6. [citováno 25. 1. 2021]. Framework pro vývoj software určený pro kvantové počítání. 2019. DOI: 10.5281/zenodo.2562110. URL: https://qiskit.org.

[2] Ambainis, A., Kempe, J. a Rivosh, A. "Coins Make Quantum Walks Faster". In: *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms* (břez. 2004). DOI: 10.1145/1070432.1070590.

[3] Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J., Barends, R., Biswas, R., Boixo, S., Brandao, F., Buell, D., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B. a Martinis, J. "Quantum supremacy using a programmable superconducting processor". In: *Nature* 574.7779 (říj. 2019), s. 505–510. DOI: 10.1038/s41586-019-1666-5.

[4] Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Boixo, S., Broughton, M., Buckley, B. B. a al., et. "Hartree-Fock on a superconducting qubit quantum computer". In: *Science* 369.6507 (srp. 2020), 1084–1089. ISSN: 1095-9203. DOI: 10.1126/science.abb9811. URL: http://dx.doi.org/10.1126/science.abb9811.

[5] Aspect, A., Grangier, P. a Roger, G. "Experimental Realization of Einstein-Podolsky-Rosen-Bohm Gedankenexperiment: A New Violation of Bell's Inequalities". In: *Phys. Rev. Lett.* 49 (čvc 1982), s. 91–94. DOI: 10.1103/PhysRevLett.49.91.

[6] Bell, J. "On the Einstein Podolsky Rosen paradox". In: *Physics Physique Fizika* 1.3 (lis. 1964), s. 195–200. DOI: 10.1103/PhysicsPhysiqueFizika.1.195.

[7] Benioff, P. "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines". In: *Journal of Statistical Physics* 22.5 (květ. 1980), s. 563–591. DOI: 10.1007/BF01011339.

[8] Clauser, J., Horne, M., Shimony, A. a Holt, R. "Proposed Experiment to Test Local Hidden-Variable Theories". In: *Phys. Rev. Lett.* 23.15 (říj. 1969), s. 880–884. DOI: 10.1103/PhysRevLett.23.880.

[9] Deutsch, D. a Jozsa, R. "Rapid Solution of Problems by Quantum Computation". In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907 (pros. 1992), s. 553–558. DOI: 10.1098/rspa.1992.0167.

[10] Deutsch, D. "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 400.1818 (čvc 1985), s. 97–117. DOI: 10.1098/rspa.1985.0070.

[11] Dušek, M. *Koncepční otázky kvantové teorie*. 1. vydání. Univerzita Palackého, 2002, s. 121–230. ISBN: 80-244-0449-4.

[12] Einstein, A., Podolsky, B. a Rosen, N. "Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?" In: *Physical Review* 47.10 (led. 1935), s. 777–780. DOI: 10.1103/PhysRev.48.696.

[13] Figgatt, C., Maslov, D., Landsman, K., Linke, N., Debnath, S. a Monroe, C. "Complete 3-Qubit Grover Search on a Programmable Quantum Computer". In: *Nature Communications* 8.1 (pros. 2017), s. 1–9. DOI: 10.1038/s41467-017-01904-7.

[14] Grover, L. "Fast quantum mechanical algorithm for database search". In: *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing* (červ. 1996), s. 212–219. DOI: 10.1145/237814.237866.

[15] Hamilton, C. S., Kruse, R., Sansoni, L., Barkhofen, S., Silberhorn, C. a Jex, I. "Gaussian Boson Sampling". In: *Physical Review Letters* 119.17 (říj. 2017). ISSN: 1079-7114. DOI: 10.1103/physrevlett.119.170501. URL: http://dx.doi.org/10.1103/PhysRevLett.119.170501.

[16] Harrow, A., Hassidim, A. a Lloyd, S. "Quantum Algorithm for Linear Systems of Equations". In: *Physical review letters* 103.15 (říj. 2009), s. 150502. DOI: 10.1103/PhysRevLett.103.150502.

[17] Hlavatý, L. a Štefaňák, M. *Slabikář kvantové mechaniky*. [online]. [citováno 15. 2. 2021]. Naposledy aktualizováno 18. 9. 2018. URL: https://physics.fjfi.cvut.cz/files/predmety/02KVAN/02KVAN.

[18] Kempe, J. "Quantum random walks: An introductory overview". In: *Contemporary Physics* 44.4 (břez. 2003), s. 307–327. DOI: 10.1080/0010715103100011077 6.

[19] Kerenidis, I. a Prakash, A. "Quantum Recommendation Systems". In: *arXiv preprint arXiv:1603.08675* (břez. 2016).

[20] Lavor, C., Manssur, L. a Portugal, R. "Shor's Algorithm for Factoring Large Integers". In: *arXiv preprint quant-ph/0303175* (dub. 2003).

[21] Nielsen, M. A. a Chuang, I. L. *Quantum computation and quantum information*. 10th anniversary ed. Cambridge ; New York: Cambridge University Press, 2010. ISBN: 9781107002173.

[22] Preskill, J. "Quantum Computing in the NISQ era and beyond". In: *Quantum* 2 (led. 2018), s. 79. DOI: 10.22331/q-2018-08-06-79.

[23] Quesada, N., Arrazola, J. M. a Killoran, N. "Gaussian boson sampling using threshold detectors". In: *Physical Review A* 98.6 (pros. 2018). ISSN: 2469-9934. DOI: 10.1103/physreva.98.062322. URL: http://dx.doi.org/10.1103/PhysRevA.98.062322.

[24] Reitzner, D., Hillery, M., Feldman, E. a Bužek, V. "Quantum Searches on Highly Symmetric Graphs". In: *Physical Review A* 79.1 (červ. 2008), s. 012323. DOI: 10.1103/PhysRevA.79.012323.

[25] Reitzner, D., Nagaj, D. a Bužek, V. "Quantum Walks". In: *Acta Physica Slovaca* 61 (čvc 2012). DOI: 10.2478/v10155-011-0006-6.

[26] Shor, P. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, s. 124–134. DOI: 10.1109/SFCS.1994.365700.

[27] Štefaňák, M. a Jex, I. "Kvantové procházky". In: *Rozhledy matematicko-fyzikální* 90.1 (2015), s. 22–30.

[28] Vedral, V., Barenco, A. a Ekert, A. "Quantum Networks for Elementary Arithmetic Operations". In: *Physical Review A* 54.1 (lis. 1995), s. 147. DOI: 10.1103/PhysRevA.54.147.

[29] Zhong, H.-S. et al. “Quantum computational advantage using photons”. In: *Science* 370.6523 (2020), s. 1460–1463. ISSN: 0036-8075. DOI: 10.1126/science.abe8770. eprint: https://science.sciencemag.org/content/370/6523/1460.full.pdf. URL: https://science.sciencemag.org/content/370/6523/1460.

[30] Štefaňák, M. “Quantum Walks”. Habilitační práce. Praha: České vysoké učení technické v Praze. Fakulta jaderná a fyzikálně inženýrská, 2017, s. 58–67.

[31] Štefaňák, M. *Úvod do kvantové teorie - kapitola 10*. [online]. [citováno 8. 4. 2021]. Naposledy aktualizováno 19. 5. 2020. URL: https://physics.fjfi.cvut.cz/files/predmety/02UKT/kap_10.pdf.

[32] Šťovíček, P. *Úvod do obecné algebry s prvky teorie množin*. 1. vydání. ČVUT, 2021, s. 55–89. ISBN: 978-80-01-06813-7.