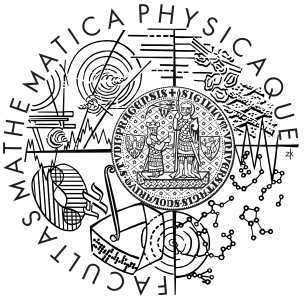
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Lukáš Mejdrech

Algoritmus pro kvantové hledání

Katedra Algebry

Vedoucí bakalářské práce: Mgr. Libor Barto, Ph.D.
Studijní program: Informatika, Programování

2007

Rád bych poděkoval vedoucímu své práce Mgr. Liboru Bartovi, Ph.D., s jehož laskavou pomocí a radou jsem byl schopen práci dokončit.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne Lukáš Mejdrech

Obsah

Název práce: Algoritmus pro kvantové hledání
Autor: Lukáš Mejdrech
Katedra: Katedra Algebry
Vedoucí bakalářské práce: Mgr. Libor Barto, Ph.D.
e-mail vedoucího: barto@karlin.mff.cuni.cz

Abstrakt: V předložené práci studujeme Groverův algoritmus, který slibuje na kvantovém počítači vyhledávat v nesetříděné databázi v čase úměrném odmocnině z počtu položek. Po popsání základních principů kvantových počítačů je uvedena jak původní Groverova formulace a důkaz algoritmu, tak i pozdější geometrický popis tohoto algoritmu. Také je poukázáno na vhodnost těchto popisů k výukovým účelům. Nakonec je zahrnut popis Groverova algoritmu v programovacím jazyce umožňujícím simulovat na klasických počítačích kvantové výpočty.

Klíčová slova: Grover, kvantový počítač, hledání

Title: Quantum search algorithm
Author: Lukáš Mejdrech
Department: Department of Algebra
Supervisor: Mgr. Libor Barto, Ph.D.
Supervisor's e-mail address: barto@karlin.mff.cuni.cz

Abstract: In this work we study Grover's algorithm for quantum computers. This algorithm promises to search in an unstructured database in time comparable with a square root of the number of objects. A description of basic quantum computer principles is followed by the original Grover's formulation and a proof of the algorithm, as well as a later geometrical description of the algorithm. We also mention their suitability for educational purposes. We also include a description of the Grover's algorithm in a programming language, which makes it possible to simulate a quantum computing on classical computers.

Keywords: Grover, quantum computer, search

Kapitola 1

Úvod

1.1 Několik slov o této práci

Tato práce by měla být stručným přehledem algoritmu pro kvantové vyhledávání spolu se základním popisem kvantového počítače. Kvantové počítače sice ještě nebyly vyvinuty, ale matematický model jejich fungování je již dobře popsán. Jedná se o zcela novou filozofii, která se opírá o intenzivní výzkum a jsou do ní vkládány velké naděje. Podle původních odhadů Feynmana z osmdesátých let by mohl být kvantový počítač přinejmenším stejně tak dobrý jako klasický, avšak koncem devadesátých let se dokonce objevily algoritmy, které ukázaly, že kvantové počítače mohou být rychlejší, někdy až exponenciálně rychlejší. Byly to Shorův algoritmus faktorizace čísel a Groverův vyhledávací algoritmus.

Groverův algoritmus, který je předmětem této práce, slibuje vyhledávat v nesetříděné databázi v čase úměrném odmocnině z počtu položek, zatímco klasické počítače stejný úkon provádí postupným čtením položek. Jedná se tedy o kvadratické zrychlení. V této práci je uvedena jak původní Groverova formulace a důkaz algoritmu, tak i pozdější geometrický popis tohoto algoritmu. Také bude poukázáno na vhodnost těchto popisů k výukovým účelům. V samostatné kapitole bude zahrnut popis Groverova algoritmu v programovacím jazyce umožňujícím simulovat na klasických počítačích kvantové výpočty.

Protože mnoho algoritmů potřebuje vyhledávat, mohla by se použitím tohoto algoritmu snížit i jejich časová náročnost. Bohužel jsou kvantové počítače teprve v počátcích a jejich reálná výroba, použití nebo dokonce provoz se potýká s velkým počtem fyzikálních problémů a omezení. Není jisté, zda se s kvantovým počítačem setkáme. Bylo již sice vyvinuto několik prototypů, ale jednalo se spíše o demonstraci než použití kvantové mechaniky [1]. Vývoj kvantových počítačů by mohl být stejně zajímavý jako byl kdysi vývoj klasických počítačů, které se nám od velkých monster v obrovských sálech dostaly na desky stolů a dokonce i do dlaní [2].

Dále v úvodu nastíním motivaci pro vznik úvah o kvantovém počítači a zmíním mezníky v kvantové fyzice, které umožnily jejich výzkum. Pak se již budu zabývat po-

[1] V únoru roku 2007 představila společnost D-Wave Systems, Inc. první prototyp komerčně využitelného kvantového počítače [12].

[2] I dnešní mobilní telefony a hudební přehrávače mají vetší výpočetní výkon než první počítače.

pisem matematického modelu kvantového počítače a Groverova vyhledávacího algoritmu. Nakonec zmíním zdrojový kód Groverova algoritmu pro simulátor kvantového počítače.

1.2 Myšlenka kvantového počítače

Klasické počítače jsou velmi komplikované a velmi mnoho dokáží. Stále se podle trochu upraveného Moorova zákona [7] každé dva roky zvyšuje výkon počítače na dvojnásobek [3]. Tím sice dostáváme velmi dobré vyhlídky do budoucna, ale na světě je mnoho problémů, k jejichž řešení potřebujeme trochu více.

Dnešní počítače pracují na modelu informace-operace. Mají nějakým způsobem uloženy informace a nějak je zpracovávají. Postupem času zbyl už jen jediný způsob práce s informacemi, práce s jedničkami a nulami [4]. Ty jsou uloženy jako písmena napsaná na papíře, bloky kovu, které zmagnetizujeme v discích a disketách, krystalky chemických sloučenin na kompaktních discích, jimž laserovým paprskem měníme odrazivost povrchu, a konečně jako drátky uvnitř procesoru, po kterých běhají elektrony ovládané změnami napětí. Všechny výše uvedené způsoby uchovávání informace spotřebovávají milióny a milióny částic. A přece nejjednodušší by bylo mít jednu částici na jednu jednotku informace, jedničku nebo nulu.

Na úrovni jednotlivých částic se již dostáváme do problémů s takzvanými kvantovými jevy. Pro svět, jak jej známe, platí takzvané deterministické fyzikální zákony. Pokud známe výchozí stav, umíme určit stav následný i předchozí. Ale pro jednotlivé částice toto tak úplně neplatí, vlastně téměř vůbec.

Zkoumání kvantových jevů přineslo výsledky na počátku dvacátého století, kdy přišli se svými závěry významní fyzikové. Prvním byl Max Planck s tvrzením, že energie je vyzařována v malých dávkách, takzvaných kvantech, namísto souvislého proudu [10]. S další a velmi podstatnou teorií přišel Albert Einstein, a to, že se světlo šíří také v dávkách [5] a ne jen ve vlnách [4], za kterou později dostal Nobelovu cenu. A konečně Luis de Broglie vyvodil naopak, že částice mají také vlnový charakter [2]. Tím byl uzavřen kruh duality, který znamená, že všechny částice mají také vlnové vlastnosti a naopak že všechny vlny mají také částicové vlastnosti. Spíše nám to říká, že ať vlny tak částice jsou všechno pouze kvanta energie, pro která platí stejné

[3] V původním znění:

> "The complexity for minimum component costs has increased at a rate of roughly a factor of two per year ... Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000. I believe that such a large circuit can be built on a single wafer." [7]

[4] Tyto dva stavy se vyjadřují ve dvojkové soustavě a jednotkou je bit. Nějakou dobu se experimentovalo i s vícestavovými systémy, z nichž se nejdéle uchytil třístavový systém, s jednotkou nazvanou trit. Nakonec se ale od vícestavových systémů upustilo, protože vše šlo vyjádřit ve dvojkové soustavě, se kterou se pracovalo jednodušeji.

[5] Na tyto dávky se opět může pohlížet jako na částice, později nazvané fotony.

principy. Nabízí se tedy otázka k budoucímu vývoji — pokud mají částice takovéto zvláštní vlastnosti, se kterými bychom se časem stejně museli naučit vypořádat, proč je dokonce nevyužít ke svému prospěchu?

1.3 Kvantová fyzika

Kvantová fyzika předpokládá, že každá částice má vlnové vlastnosti. Má takzvanou vlnovou funkci, která nám nedává přesné umístění částice, ale pravděpodobnost, s jakou se částice nachází v libovolné možné pozici. Protože částice je vlastně ve všech možných pozicích, říká se tomuto princip superpozice. Vlnovou mechaniku představil světu Schrödinger v roce 1926 [11], krátce po publikaci de Broglieho [2]. Vlnová funkce se vyvíjí v čase podle Schrödingerovy rovnice. Pokud je částice v kombinaci více stavů, pak je v této kombinaci i nadále. Jako pozorovatelé nemáme přístup k této vlnové funkci a jediné, co můžeme, je pozorovat jak se systém vyvinul. Protože mohou být kvantové systémy v superpozici stavů, není zřejmé a ani odvoditelné jaká pozorování dostaneme. Zamíchává se nám do nich pravděpodobnost stavů a tedy jistý druh náhody. To bylo důvodem obtížného pochopení samotnými fyziky a ještě mnohem obtížnějšího laiky.

Kvantové počítače jsou tedy pokusem o využití principu vlnové funkce a tím i superpozice stavů. Na rozdíl od klasických počítačů by tak mohly být v jednom okamžiku ve více stavech najednou. Díky této myšlence jsou zajímavé. Co musí klasické počítače počítat postupně, mohou kvantové počítače počítat najednou.

Kapitola 2

Princip fungování kvantového počítače

2.1 Kvantový bit

Kvantová varianta bitu v klasickém počítači jako nosiče informace je kvantový bit, zkráceně také qubit. Klasický bit nese vždy jednu jednotku informace, jedničku nebo nulu. Použijeme-li principy kvantové fyziky, může nést jedničku i nulu zároveň. Každou hodnotu s určitou pravděpodobností a to po celou dobu výpočtu. Hodnoty se uchovávají jako komplexní čísla a druhá mocnina pak odpovídá jejich pravděpodobnosti. Jen při měření výpočtu qubit kolabuje do jednoho ze základních stavů, tedy jedničky nebo nuly. Děje se tak podle pravděpodobnosti stavu. Nejsme tedy schopni zjistit jeho přesný stav, ale jen některé měřitelné hodnoty, kterými jsou právě základní stavy. Pro zápis stavu qubitu se používá komplexní sloupcový vektor o počtu řádek rovnému počtu různých stavů. Jeden qubit může být v základním výpočetním stavu, kdy označuje jedničku nebo nulu. Pak píšeme [1]

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.1}$$

Nebo může být qubit v obecném stavu ϕ, kterému odpovídají komplexní amplitudy ϕ_1 a ϕ_2

$$|\phi\rangle = \phi_1|0\rangle + \phi_2|1\rangle = \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}. \tag{2.2}$$

Jednotlivé amplitudy $\phi_{1,2}$ stavového vektoru $|\phi\rangle$ mohou být rozloženy na součin $\phi_j = |\phi_j| \cdot e^{i\theta_j}$ pro Eulerovo číslo e, $i = \sqrt{-1}$ a nějaké reálné číslo θ. Tomuto číslu se říká kvantová fáze. Fáze nemění pravděpodobnost základních stavů v kvantovém stavu, ale je významná při kvantové interferenci. To je schopnost kvantových stavů se spolu slučovat a docílit tak zesílení nebo potlačení svých amplitud, tedy pravděpodobnosti změření jejich hodnoty. Správným použitím operací tak můžeme zvýšit pravděpodobnost naměření u správných hodnot a naopak snížit u nesprávných.

[1] Podle takzvané bra-ket notace, kterou vytvořil Paul Dirac.

Základem počítače je jeho schopnost práce s informacemi-bity. Pro kvantový počítač tedy potřebujeme definovat operace s qubity. Již jsme definovali $|\phi\rangle$ ket vektor, sloupcový vektor o počtu řádek rovnému počtu různých stavů, tedy

$$|\phi\rangle = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_n \end{pmatrix}. \tag{2.3}$$

Dále definujeme $\langle\phi|$ bra vektor, řádkový vektor, transponovaný komplexně sdružený vektor k $|\phi\rangle$, tedy

$$\langle\phi| = |\phi\rangle^\dagger = \overline{|\phi\rangle^T} = \left(\overline{\phi_1}, \ldots, \overline{\phi_n}\right). \tag{2.4}$$

Budeme také používat standardní skalární součin $\langle\phi| \cdot |\psi\rangle$, zkráceně $\langle\phi|\psi\rangle$ [2]. Tedy

$$\langle\phi|\psi\rangle = \sum_{i=1}^{n} \overline{\phi_i}\psi_i. \tag{2.5}$$

Standardně definujeme i normu vektoru $\|\phi\|$, a to následovně

$$\|\phi\| = \sqrt{\langle\phi|\phi\rangle}. \tag{2.6}$$

Jak již bylo zmíněno výše, odpovídají jednotlivé amplitudy pravděpodobnosti, s jakou je qubit v daném stavu. Konkrétně je tato pravděpodobnost druhou mocninou amplitudy. Nezáleží tedy na fázi, ale na velikosti. Proto by bylo vhodné, aby součet druhých mocnin amplitud byl roven jedné, tedy aby celková velikost vektoru podle výše definované normy byla rovna jedné. Takovýmto vektorům říkáme normalizované nebo jednotkové a platí pro ně následující

$$\|\phi\| = 1 = 1^2 = \|\phi\|^2 = \left(\sqrt{\langle\phi|\phi\rangle}\right)^2 = \langle\phi|\phi\rangle, \tag{2.7}$$

pro dvoustavový qubit dostáváme

$$1 = \langle\phi|\phi\rangle = \left(\overline{\phi_1}\langle 0| + \overline{\phi_2}\langle 1|\right)\left(\phi_1|0\rangle + \phi_2|1\rangle\right) = \overline{\phi_1}\phi_1 + \overline{\phi_2}\phi_2 = |\phi_1|^2 + |\phi_2|^2. \tag{2.8}$$

Skalární součin dvou normalizovaných vektorů pak odpovídá kosinu úhlu, který spolu svírají

$$\langle\phi|\psi\rangle = \cos\left(\angle|\phi\rangle|\psi\rangle\right). \tag{2.9}$$

Formálně podobně jako skalární součin [3] můžeme definovat vnější součin dvou vektorů, kdy

$$|\phi\rangle\langle\psi| = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{pmatrix} \cdot \left(\overline{\psi_1}, \overline{\psi_2}, \ldots, \overline{\psi_n}\right) = \begin{pmatrix} \phi_1\overline{\psi_1} & \phi_1\overline{\psi_2} & \cdots & \phi_1\overline{\psi_n} \\ \phi_2\overline{\psi_1} & \phi_2\overline{\psi_2} & \cdots & \phi_2\overline{\psi_n} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_n\overline{\psi_1} & \phi_n\overline{\psi_2} & \cdots & \phi_n\overline{\psi_n} \end{pmatrix}. \tag{2.10}$$

[2] Tímto zápisem byly inspirovány názvy vektorů bra a ket, protože dohromady vypadají jako závorka, v angličtině braket.

[3] Také označovaný jako vnitřní součin.

2.2 Vektorový prostor

Pro kvantový stav se používá Hilbertův prostor $\mathbb{C}^n$. Jedná se o komplexní vektorový prostor s výše definovanými operacemi skalárního součinu $\langle\cdot|\cdot\rangle$ a normy $\|\cdot\|$. Dimenze je rovna počtu různých základních stavů. Pro jeden qubit tedy standardně 2, kdy se jedná o komplexní rovinu $\mathbb{C}^2$.

Ortonormální systém je taková množina vektorů dimenze n

$$B = \{|\beta_1\rangle, |\beta_2\rangle, \ldots, |\beta_n\rangle\}, \tag{2.11}$$

ve které jsou všechny vektory normované a navzájem kolmé, neboli skalární součin dvou různých vektorů je nula. Tvoří tak bázi Hilbertova prostoru $\mathbb{C}^n$ a každý vektor $|\phi\rangle$ z tohoto prostoru lze rozložit na složky podle jednotlivých vektorů tohoto systému,

$$|\phi\rangle = \sum_{i=1}^{n} \phi_i |\beta_i\rangle. \tag{2.12}$$

Pro dvoustavový qubit se běžně používá standardní báze 2.1 nebo Hadamardova báze

$$\begin{aligned} |+\rangle &= \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ |-\rangle &= \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}. \end{aligned} \tag{2.13}$$

O vektorovém prostoru V řekneme, že je direktním součtem vektorových podprostorů $V_1, V_2, \ldots, V_n$, a zapisujeme $V = V_1 \oplus V_2 \oplus \cdots \oplus V_n$, pokud $V_i \cap V_j = \{0\}$, $i \neq j$ a lineární obal $V_1, \ldots, V_n$ je V. Pokud $V = V_1 \oplus \ldots \oplus V_n$, pak se dá každý vektor $v \in V$ jednoznačně napsat jako $v = v_1 + v_2 + \cdots + v_n$, kde $v_i \in V_i$.

2.3 Kvantový registr

Nyní se budeme věnovat práci s více qubity najednou. Nejprve zavedeme tenzorový součin dvou vektorů z Hilbertových prostorů $\mathbb{C}^n$ a $\mathbb{C}^m$, $|\phi\rangle \otimes |\psi\rangle$. Budeme ho zapisovat zkráceně $|\phi\psi\rangle$, a definujeme jej

$$|\phi\rangle \otimes |\psi\rangle = |\phi\rangle|\psi\rangle = |\phi\psi\rangle = \{a_{(i-1)m+j} = \phi_i \psi_j\}_{i=1,j=1}^{n,m}, \tag{2.14}$$

názorněji

$$|\phi\rangle \otimes |\psi\rangle = (\phi_1\psi_1, \ldots, \phi_1\psi_m, \phi_2\psi_1, \ldots, \phi_2\psi_m, \cdots, \phi_n\psi_1, \ldots, \phi_n\psi_m)^T. \tag{2.15}$$

Takovýto vektor ovšem bude náležet do Hilbertova prostoru $\mathbb{C}^{nm}$, na který se podíváme blíže. Vezmeme-li Hilbertovy prostory V a W s ortonormálními bázemi $\{v_i\}_{i=1}^n$ a $\{w_j\}_{j=1}^m$, pak definujeme jejich tenzorový součin $S = V \otimes W$ jako Hilbertův prostor dimenze $n \cdot m$ s ortonormální bází $\{|v_i\rangle \otimes |w_j\rangle\}_{i=1,j=1}^{n,m}$. Pro zápis vektorů budeme používat konvenci podle definice 2.14, kdy postupně kombinujeme bázové vektory

z prvního prostoru se všemi bázovými vektory z druhého prostoru. Například pro kombinaci dvou qubitů dostaneme Hilbertův prostor $\mathbb{C}^4$ s bází

$$|0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = |00\rangle,$$

$$|01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \tag{2.16}$$

A stav dvou qubitů vyjádříme v tomto prostoru jako

$$|\phi\rangle \otimes |\psi\rangle = \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} \otimes \begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix} = \begin{pmatrix} \phi_1\psi_1 \\ \phi_1\psi_2 \\ \phi_2\psi_1 \\ \phi_2\psi_2 \end{pmatrix}. \tag{2.17}$$

Zjednodušíme značení zavedením tenzorové mocniny, nebo také řádu, Hilbertova prostoru $V^{\otimes 1} = V$ a $V^{\otimes(n+1)} = V \otimes V^{\otimes n}$. Pro výpočet s n dvoustavovými qubity se tedy budeme pohybovat v Hilbertově prostoru dimenze 2^n, prostor $(\mathbb{C}^2)^n$, který bude mít bázi z n-bitových řetězců $\{0,1\}^n$. Někdy o n qubitech mluvíme také jako o n-qubitovém registru. Při práci s n-qubitovým registrem pracujeme tedy současně s 2^n možnými stavy pouze na n qubitech. Tomuto se říká kvantový paralelizmus a je hlavním přínosem kvantových počítačů k jejich vysoké efektivitě.

Jedna z nejpodivnějších vlastností kvantového stavu takovéhoto registru je možnost propletení stavů [4]. Výše zmíněnou metodou tenzorového součinu z navzájem nezávislých qubitů sestavíme kvantový stav. Každý je popsán samostatným Hilbertovým prostorem a dohromady vytvářejí jeden velký. Existují však fyzikální procesy, kterými dokážeme vytvořit i jiné stavy, které nelze vyjádřit jako tenzorový součin jednotlivých qubitů. O takovýchto qubitech říkáme, že jsou propletené. Znamená to, že jsou na sobě qubity nějakým způsobem závislé. Pokud změříme jeden qubit, nemusíme již měřit hodnotu druhého. Dopředu ale nevíme jaké hodnoty změříme. Měřením qubity opět kolabují do jednoho ze základních stavů. Této vlastnosti se využívá pro kvantovou teleportaci.

2.4 Vývoj stavu

Pokud máme nějaký počáteční stav systému, řídí se jeho další vývoj Schrödingerovou rovnicí [11]. Ta je deterministická, lineární a udává vývoj stavu v čase. K fungování kvantového počítače je zapotřebí pouze její zjednodušená verze. Zajímají nás jen námi chtěné operace nad kvantovým stavem v pořadí, které určíme.

[4] Někdy se používá i počeštěný anglický výraz proplétat, entangle.

Pracujeme s vektory vyjádřenými pomocí bázových vektorů vektorového prostoru a s operacemi, které se chovají lineárně, přesněji zachovávají operace sčítání a násobení komplexním číslem. Máme-li ortonormální bázi $\{v_i\}_{i=1}^n$ Hilbertova prostoru V dimenze n, pak stačí definovat operaci U pouze na tyto vektory a efekt na celkový kvantový stav plyne z linearity, neboli pro všechna i je $U : |v_i\rangle \rightarrow \sum_{j=1}^n u_{j,i}|v_j\rangle$, pro nějaké komplexní koeficienty $u_{i,j}$. Operaci U pak můžeme reprezentovat maticí o rozměrech $n \times n$

$$U = \{u_{i,j}\}_{i=1,j=1}^{n,n} = \begin{pmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ u_{2,1} & u_{2,2} & \cdots & u_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n,1} & u_{n,2} & \cdots & u_{n,n} \end{pmatrix}. \tag{2.18}$$

Stav systému $|\phi\rangle$ po provedení operace U lze tedy zapsat jako

$$|\phi_2\rangle = U|\phi_1\rangle. \tag{2.19}$$

Dále potřebujeme zachovávat normu vektoru, k tomu musí platit $U^{-1} = U^\dagger$, protože chceme

$$\begin{aligned} \||\phi\rangle\| &= \|U|\phi\rangle\| && (2.20) \\ \sqrt{\langle\phi|\phi\rangle} &= \sqrt{\langle U\phi|U\phi\rangle} \\ \langle\phi|\phi\rangle &= (U|\phi\rangle)^\dagger \cdot U|\phi\rangle \\ \langle\phi|I|\phi\rangle &= \langle\phi|U^\dagger \cdot U|\phi\rangle \\ I &= U^\dagger \cdot U. && (2.21) \end{aligned}$$

Tím také říkáme, že operaci lze obrátit použitím operace U^{-1}, která je dobře definovaná jako $U^{-1} = U^\dagger = \overline{U^T}$, komplexně sdružená transponovaná matice původní operace. Splnili jsme tak i poslední podmínku plynoucí ze Schrödingerovy rovnice, operace musí být reverzibilní, abychom mohli jejich průběh jednoznačně obrátit. Operacím, které splňují tyto vlastnosti říkáme unitární operátory.

Protože jsou operace vratné, nelze provádět operace známé a používané u klasických počítačů jako například ukládání čísel do paměti nebo mazání nepoužívaných registrů. Musíme je obejít vratnými operacemi, například bitovým součinem xor.

Pro popis algoritmu budeme pracovat s fázovým otočením

$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \tag{2.22}$$

které změní fáze stavu qubitu o daný úhel θ, s výběrovým otočením fáze

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{2.23}$$

které otočí fázi u jednoho ze základních stavů, a s Hadamardovou bránou

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{2.24}$$

Hadamardova brána má navíc velmi zajímavé vlastnosti. Je sama sobě inverzní, samo sdružená, neboli $H^2 = I$, a navíc převádí výpočetní bázi 2.1 na Hadamardovu 2.13 a naopak. Pro jednotlivé qubity se počítá jako

$$\begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) = |+\rangle \\ H|1\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) = |-\rangle. \end{aligned} \tag{2.25}$$

Tyto operace se používají na jeden qubit a tak je můžeme chtít použít v jednom kroku na více qubitů. Tím získáme kvantovou obdobu hradla u klasických počítačů. Formálně je rozšíříme do tenzorové mocniny. Máme-li n qubitů, pohybujeme se v Hilbertově prostoru tenzorové mocniny n, $(\mathbb{C}^2)^{\otimes n}$. Máme-li lineární operátor U_1 na V_1 a U_2 na V_2, pak tenzorovým součinem $U_1 \otimes U_2$ rozumíme operátor na $V_1 \otimes V_2$ definovaný na bázi takto $v_1 \otimes v_2 \to U_1 v_1 \otimes U_2 v_2$. Usnadníme si značení zavedením tenzorové mocniny operace pokud ji chceme použít na více qubitů současně

$$\begin{aligned} V^{\otimes n} &\xrightarrow{U^{\otimes n}} V^{\otimes n} \\ |v\rangle = |v_1, v_2, \ldots, v_n\rangle &\xrightarrow{U^{\otimes n}} (U|v_1\rangle) \otimes (U|v_2\rangle) \otimes \cdots \otimes (U|v_n\rangle) = \bigotimes_{i=1}^{n} U|v_i\rangle \end{aligned} \tag{2.26}$$

Pokud naopak potřebujeme provést operaci U pouze na některém qubitu, řekněme i, složíme tuto operaci s operacemi identity I a dostaneme tak

$$\begin{array}{ccccccccccc} & 1 & & & i-1 & & i & & i+1 & & n \\ U_i = & I & \otimes \cdots \otimes & & I & \otimes & U & \otimes & I & \otimes \cdots \otimes & I \end{array} = I^{\otimes(i-1)} \otimes U \otimes I^{\otimes(n-i)}. \tag{2.27}$$

Dále budeme používat takzvanou Walsh-Hadamardovu transformaci W, použití Hadamardovy brány na více qubitů. Dostaneme tak změnu kvantového stavu $|\phi\rangle$ n-stavového systému jako

$$\begin{aligned} |\phi\rangle = |\phi_1, \ldots, \phi_n\rangle &\xrightarrow{H^{\otimes n}} (H|\phi_1\rangle) \otimes \cdots \otimes (H|\phi_n\rangle) = \bigotimes_{i=1}^{n} H|\phi_i\rangle = \\ &= \frac{1}{\sqrt{2^n}} \bigotimes_{i=1}^{n} \sum_{\psi_i \in \{0,1\}} (-1)^{\psi_i \phi_i} |\psi_i\rangle = \\ &= \frac{1}{\sqrt{2^n}} \sum_{\psi \in \{0,1\}^n} \bigotimes_{i=1}^{n} (-1)^{\psi_i \phi_i} |\psi_i\rangle = \\ &= \frac{1}{\sqrt{2^n}} \sum_{\psi \in \{0,1\}^n} (-1)^{\sum_{i=1}^{n} \psi_i \phi_i} |\psi\rangle = \\ &= \frac{1}{\sqrt{2^n}} \sum_{\psi \in \{0,1\}^n} (-1)^{\phi \bullet \psi} |\psi\rangle, \end{aligned} \tag{2.28}$$

kde $\phi \bullet \psi$ je bitový skalární součin [5]

$$\phi \bullet \psi = \oplus_{i=1}^{n} \phi_i \psi_i. \tag{2.29}$$

2.5 Měření výsledku

V této práci si vystačíme pouze se základním způsobem měření kvantového stavu, projekcí [6]. Máme-li kvantový systém v Hilbertově prostoru V dimenze n s ortonormální bází $\{v_i\}_{i=1}^{n}$, můžeme tento prostor rozdělit na direktní součet m menších Hilbertových prostorů nižších dimenzí $V = S_1 \oplus \cdots \oplus S_m$ rozdělením bázových vektorů do m množin. Protože báze byla ortonormální, jsou i báze podprostorů ortonormální a podprostory jsou také navzájem ortogonální. Každý takovýto podprostor odpovídá možnému výsledku měření. Kvantový stav původního prostoru může být zapsán jako

$$|\phi\rangle = \sum_{i=1}^{m} \alpha_i |\phi_i\rangle, \tag{2.30}$$

kde α_i je komplexní číslo a $|\phi_i\rangle$ je normalizovaný kvantový stav odpovídající podprostoru S_i. Kvantový stav $|\phi_i\rangle$ vychází z původních hodnot kvantového stavu $|\phi\rangle$ příslušejících bázovým vektorům podprostoru S_i a nul na ostatních pozicích, jen je normovaný, konkrétně vydělený číslem α_i, které je jeho velikostí, normou. Toto rozložení stavu je jednoznačné díky tomu, že je rozklad prostoru direktní. Při měření získáme náhodně číslo $I \in \{1, \ldots, m\}$ s pravděpodobnostmi

$$P[I = i] = |\alpha_i|^2. \tag{2.31}$$

Po měření kvantový systém kolabuje do příslušného kvantového stavu $|\phi_i\rangle$. Měřením se tedy dozvíme do kterého podprostoru systém zkolaboval a zůstanou informace pouze o hodnotách příslušejících bázovým vektorům vybraným do množiny i.

Maticově lze projektivní měření vyjádřit jako takzvané projektory. Projektor A je samo sdružený unitární operátor, tedy $A^\dagger = A$, a navíc takový, že $A^2 = A$. Například pro podprostor tvořený pouze jedním bázovým vektorem v_i je to čtvercová matice A_i s jedinou nenulovou hodnotou $a_{i,i} = 1$. Každý podprostor S_i můžeme tedy reprezentovat projektorem do tohoto podprostoru P_i. Projektory můžeme skládat a dohromady dostáváme identitu

$$\sum_{i=1}^{m} P_i = I. \tag{2.32}$$

Projektor se chová tak, že některé amplitudy vektoru nuluje a ostatní nechá beze změny. Například pro dvou-qubitový systém $V = \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ můžeme chtít

[5] Jedná se vlastně o počet shodných jedničkových bitů obou vektorů, při umocnění -1 tedy o paritu.

[6] Pro další způsoby měření bych čtenáře odkázal na knihu Roberta Špalka [13].

projekci do podprostoru $V_1 = \{|00\rangle, |01\rangle\}$. Odpovídající projektor by se pak choval následovně

$$\begin{aligned} P_1 \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix} &= (A_{00} + A_{01}) \cdot \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \\ 0 \\ 0 \end{pmatrix} \\ &= P_1 \left(\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle\right) = \alpha|00\rangle + \beta|01\rangle. \end{aligned} \tag{2.33}$$

Obecně je projektor součet projektorů jednotlivých bázových vektorů cílového podprostoru. Je to tedy součet jednotlivých projektorů bázových vektorů podprostoru, které jsou tvořeny vnějším součinem bázového vektoru se sebou samým

$$P_i = \sum_j A_j = \sum_j |v_j\rangle\langle v_j|. \tag{2.34}$$

Pokud chceme změřit všechny qubity n-qubitového registru, použijeme měření podle množiny projektorů $\{|i\rangle\langle i| : i \in \{0,1\}^n\}$, tedy množinu projektorů do všech možných stavů. Nejjednoduššími projektory jsou identita a vnější součin stavového vektoru se sebou samým.

Měřením kvantového stavu $|\phi\rangle$ získáme výsledek i s pravděpodobností

$$\begin{aligned} P(i) &= \||P_i|\phi\rangle\|^2 = \left(\sqrt{\langle\phi|P_i^\dagger \cdot P_i|\phi\rangle}\right)^2 = \\ &= \langle\phi|P_i^\dagger \cdot P_i|\phi\rangle = \langle\phi|P_i \cdot P_i|\phi\rangle = \\ &= \langle\phi|P_i|\phi\rangle. \end{aligned} \tag{2.35}$$

Stav poté zkolabuje na

$$|\phi_i\rangle = \frac{P_i|\phi\rangle}{\sqrt{\langle\phi|P_i|\phi\rangle}}. \tag{2.36}$$

Původní hodnota $|\alpha_i|$ tedy odpovídá právě normě vektoru $P_i|\phi\rangle$ a stav $P_i|\phi\rangle = \alpha_i|\phi_i\rangle$.

Promítneme stavový vektor do podprostoru a normalizujeme jeho velikost. Zbude nám tak kvantový stav v Hilbertově prostoru S_i a ostatní superpozice jsou zničeny. Čím přesněji chceme zjistit hodnotu kvantového systému, na tím menší podprostory provedeme projekci a tím více kvantový systém zkolabuje. Pokud chceme přesnou hodnotu, systém kolabuje až na bázový vektor. Projekci můžeme také použít ke zpřesnění kvantového systému jen do určitého prostoru, který nás dále bude zajímat.

Kapitola 3

Groverův algoritmus pro kvantové hledání

Jedním z nejvýznamnějších algoritmů pro budoucí kvantové počítače je Groverův algoritmus sloužící k vyhledávání v nesetříděné databázi. Na kvantovém počítači by mohl být kvadraticky rychlejší než nejlepší klasické algoritmy. A protože vyhledávání se používá v mnoha dalších algoritmech jako pod úloha, jeho použitím bychom mohli zrychlit i je. Pro náročné úlohy by bylo i kvadratické zrychlení velmi významné. Algoritmus vymyslel a představil světu v roce 1996 Lov Kumar Grover [5].

3.1 Algoritmus

Nechť máme soubor $N = 2^n$ stavů, označených $S_1, S_2, \ldots, S_N$. Tyto stavy budou reprezentovány svými indexy jako n-bitové řetězce. Nechť máme jedinečný stav, označíme ho S_v, který splňuje podmínku $C(S_v) = 1$, zatímco pro všechny ostatní stavy S je podmínka $C(S) = 0$. Podmínka $C(S)$ je vyhodnotitelná v konstantním čase. Problémem je nalezení stavu S_v.

Vlastní algoritmus se skládá ze tří částí:

1. Nastavíme výchozí stav n-qubitového registru do superpozice všech možných stavů se stejnými amplitudami. Neboli do stavu $S = \left(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \ldots, \frac{1}{\sqrt{N}}\right)^T$.

2. Provedeme následující unitární operace $O\left(\sqrt{N}\right)$-krát. Grover počet opakování blíže neupřesnil, ale kratší zamyšlení nad ním bude v samostatné podkapitole 3.4:

   (a) Je-li systém ve stavu S a platí-li podmínka $C(S) = 1$, tedy právě ve stavu S_v, otočíme fázi.

   (b) Použijeme difuzní transformaci D definovanou maticí o rozměrech $N \times N$:

   $$D_{i,j} = \frac{2}{N}, \quad i \neq j, \quad \text{a} \quad D_{i,i} = -1 + \frac{2}{N}. \tag{3.1}$$

3. Změříme výsledný stav systému a dostaneme index v hledaného stavu.

3.2 Popis

Pracujeme s N možnými stavy, takže potřebujeme N-dimenzionální Hilbertův prostor. Pro ten potřebujeme $\log_2 N = n$ qubitů. Nastavení výchozího stavu systému se provede paralelním použitím Hadamardovy brány na všech n qubitů systému, které jsou na počátku nulové. Jak bylo popsáno dříve v rovnici 2.28, je tato operace v čase $O(\log N) = O(n)$.

Nyní probereme podrobně krok algoritmu. Jak bylo zmíněno v úvodu, hledaný stav neznáme a tak může vypadat trochu podivně, že otáčíme fázi jen u stavu dané vlastnosti, když nevíme, který to je. Na tuto operaci můžeme nahlížet jako na menší kvantový počítač, který se rozhodne zda fázi měnit či nechat, s tím, že nepoužije přesné měření stavu systému. Asi nejnázornější je vytvoření unitárního operátoru U_ω, který kvantový stav odpovídající indexu v hledaného stavu S_v otočí a všechny ostatní stavy ponechá beze změny, neboli

$$\begin{aligned} U_\omega|\omega\rangle &= -|\omega\rangle \\ U_\omega|\phi\rangle &= |\phi\rangle, \quad \phi \in \{0,1\}^n, \phi \neq \omega. \end{aligned} \tag{3.2}$$

Tento operátor v sobě bude mít již obsaženo otočení fáze podle operátoru 2.23. Můžeme ho vytvořit například pomocí prázdného registru, do kterého nejprve vložíme hledaný stav operací xor, O_ω, pak porovnáváme stavy s tímto registrem a při jejich rovnosti otočíme fázi a nakonec registr opět smažeme operací O_ω. Výpočet se ze složití o dvě operace a porovnání v čase $O(\log N)$, takže celková časová náročnost se nezmění. V další části bude dokázáno, že difuzní operátor D lze implementovat jako $D = WRW$, kde R je operátor fázového otočení a W je Walsh-Hadamardova transformace, opět definované jako matice o rozměrech $N \times N$

$$R_{i,j} = 0, \quad i \neq j, \quad R_{0,0} = 1, \quad R_{i,i} = -1, \quad i \neq 0 \tag{3.3}$$

$$W_{i,j} = 2^{-n/2}(-1)^{i \bullet j}, \quad i \bullet j \text{ je bitový skalární součin .} \tag{3.4}$$

Takovéto rozložení je vhodné pro vlastní implementaci algoritmu, protože se jedná o použití základních operací na málo qubitech. Podobně jako hradla v klasických počítačích tak můžeme zapojit tyto tři operace za sebe. Difuzní operátor může být přepsán také jako

$$D = -I + 2P, \tag{3.5}$$

kde I je identita a P je projektor s hodnotou $P_{i,j} = \frac{1}{N}$ pro všechna i, j. Snadno nahlédneme, že použitím projektoru P na vektor dostaneme vektor, jehož všechny složky budou rovny průměru složek vektoru původního. Navíc je zřejmé, že projektor P je roven vnějšímu součinu výchozího stavového vektoru

$$P = |\phi\rangle\langle\phi| = \left(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \ldots, \frac{1}{\sqrt{N}}\right) \cdot \begin{pmatrix} \frac{1}{\sqrt{N}} \\ \frac{1}{\sqrt{N}} \\ \vdots \\ \frac{1}{\sqrt{N}} \end{pmatrix} = \left\{\frac{1}{N}\right\}_{i=1,j=1}^{n,n}. \tag{3.6}$$

Nyní se budeme věnovat použití tohoto rozepsání difuzního operátoru na stavový vektor. Máme-li stavový vektor v a označíme-li průměr hodnot jeho složek A, můžeme použití difuzního operátoru zapsat po složkách jako

$$(Dv)_i = (-I + 2P)\, v_i = -v_i + 2Pv_i = A + (A - v_i)\,. \tag{3.7}$$

Překlopili jsme tedy složky vektoru v podle jejich aritmetického průměru. Pokud se opět vrátíme k našemu problému, máme stavový vektor, jehož složky mají hodnoty $O\left(\frac{1}{\sqrt{N}}\right)$ až na jedinou, kterou je náš hledaný stav. Ten má amplitudu dokonce zápornou. Po použití difuze se ostatní stavy příliš nezmění, jelikož aritmetický průměr bude také $O\left(\frac{1}{\sqrt{N}}\right)$, ale hledaný stav se překlopí o téměř dvojnásobek hodnoty průměru. To je přibližně o $\frac{2}{\sqrt{N}}$v každém opakování algoritmu. Tímto způsobem dostatečně zvýšíme amplitudu hledanému stavu a zvýšíme mu tak pravděpodobnost změření.

Pokud je pouze pro jediný možný stav S_v splněna podmínka $C\,(S_v) = 1$, pak výsledným stavem bude index v s pravděpodobností alespoň $\frac{1}{2}$.

3.3 Důkaz

Nejprve dokážeme, že může být zapsán jako posloupnost tří kvantových transformací, a poté teprve konvergenci ke správnému výsledku. Budeme dokazovat vlastnosti amplitud stavového vektoru po provedení opakování algoritmu. Výchozí amplitudy stavového vektoru označíme k_0 a l. Amplitudy po provedení pootočení fáze, neboli kroku 2a, $k = -k_0$ a l, které zůstává stejné. A po difuzní transformaci, neboli kroku 2b, k_1 a l_1.

Tvrzení 1 *Operátor D může být vyjádřen jako $D = WRW$, kde W je Walsh-Hadamardova transformace a R fázové otočení, definované maticemi o rozměrech $N \times N$*

$$R_{i,j} = 0, \quad i \neq j, \quad R_{0,0} = 1, \quad R_{i,i} = -1, \quad i \neq 0 \tag{3.8}$$

$$W_{i,j} = 2^{-n/2}\,(-1)^{i \bullet j}, \quad i \bullet j \text{ je bitový skalární součin}\,. \tag{3.9}$$

Důkaz Spočítáme WRW a ukážeme, že se rovná D. Matici R můžeme rozepsat jako $R = R_1 + R_2$, kde $R_1 = -I$ a I je identita. Matice R_2 má pouze jedinou hodnotu různou od 0 a to 2 na pozici $R_{2,00}$. Protože W je rozšířením Hadamardovy brány H, platí pro ní obdobné vlastnosti, $W^2 = I$, tedy $W^{-1} = W$. Proto

$$D_1 = WR_1W = W^{-1}R_1W = R_1W^{-1}W = R_1 = -I. \tag{3.10}$$

Dále se podíváme na $D_2 = WR_2W$ podle násobení matic. Protože má matice R_2 pouze jednu hodnotu nenulovou, je v matici součinu R_2W pouze první řádek nenulový a roven

$$(R_2W)_{0,j} = 2W_{0,j} = \frac{2}{2^{n/2}}\,(-1)^{0 \bullet j} = \frac{2}{2^{n/2}}. \tag{3.11}$$

Na všechny prvky matice celkového součinu se tedy použijí hodnoty z prvního sloupce Walsh-Hadamardovy transformace a prvního řádku součinu R_2W

$$D_{2,ij} = 2W_{i,0}W_{0,j} = \frac{2}{2^n}(-1)^{i\bullet 0+0\bullet j} = \frac{2}{2^n}(-1)^{0+0} = \frac{2}{2^n} = \frac{2}{N}. \tag{3.12}$$

Všechny prvky D_2 jsou tedy rovné $\frac{2}{N}$ a prostým součtem je již zřejmé, že

$$D = D_1 + D_2 = WR_1W + WR_2W = W(R_1 + R_2)W = WRW. \tag{3.13}$$

Tvrzení 2 *Mějme stavový vektor o N stavech s jednou amplitudou rovnou k a všemi ostatními rovnými l. Pak po provedení difuzního operátoru D dostaneme jednu amplitudu velikosti $k_1 = \left(\frac{2}{N} - 1\right)k + 2\frac{(N-1)}{N}l$ a ostatní $l_1 = \frac{2}{N}k + \frac{(N-2)}{N}l$.*

Důkaz Pokud se podíváme na definici difuzního operátoru 3.1 a z ní odvozené rovnice pro jednotlivé amplitudy 3.7, je vidět, že budeme překlápět přes průměr, ten označíme A. Stavový vektor označíme v a index hledaného stavu h, tedy $v_h = k$ a $v_i = l$, $i \neq h$. Je zřejmé, že

$$A = \sum_{i=1}^{N} v_i = \frac{(N-1)l + k}{N} = \frac{(N-1)l}{N} + \frac{k}{N} \tag{3.14}$$

$$\begin{aligned} v_h &= k_1 = -k + 2A = -k + 2\left(\frac{(N-1)l}{N} + \frac{k}{N}\right) = \\ &= \left(\frac{2}{N} - 1\right)k + 2\frac{(N-1)}{N}l \end{aligned} \tag{3.15}$$

$$\begin{aligned} v_i &= l_1 = -l + 2A = -l + 2\left(\frac{(N-1)l}{N} + \frac{k}{N}\right) = \frac{2}{N}k + 2\frac{(N-1)}{N}l - l = \\ &= \frac{2}{N}k + \frac{(N-2)}{N}l. \end{aligned} \tag{3.16}$$

Důsledek 2.1 *Mějme stavový vektor o $N \geq 8$ stavech s jednou amplitudou rovnou k a všemi ostatními rovnými l. Nechť k je záporné reálné číslo, l kladné reálné číslo a pro jejich poměr platí $\left|\frac{k}{l}\right| < \sqrt{N}$. Pak po provedení difuzního operátoru D budou obě čísla kladná.*

Důkaz Z Tvrzení 2 vyplývá, že $k_1 = \left(\frac{2}{N} - 1\right)k + 2\frac{(N-1)}{N}l$. Za předpokladu, že $N \geq 8$, je $\left(\frac{2}{N} - 1\right)$ záporné a po vynásobením s k, které je také záporné, dostáváme kladné číslo. K tomu navíc přičteme $2\frac{(N-1)}{N}l$, opět kladné číslo. k_1 je tedy kladné. Podobně také víme, že $l_1 = \frac{2}{N}k + \frac{(N-2)}{N}l$. Pokud by tedy byla splněna podmínka $\left|\frac{k}{l}\right| < \frac{(N-2)}{2}$, je i číslo l_1 kladné. Splňuje-li poměr $\left|\frac{k}{l}\right| < \sqrt{N}$, pak pro $N \geq 8$ splňuje též $\left|\frac{k}{l}\right| < \frac{(N-2)}{2}$. I l_1 je tedy kladné.

Důsledek 2.2 *Mějme stavový vektor o N stavech s amplitudou stavu S splňujícího podmínku $C(S) = 1$ rovnou k a všemi ostatními rovnými l. Pak po provedení difuzního operátoru D bude pro amplitudy k_1 a l_1 podle Tvrzení 2 platit $k_1^2 + (N-1)\,l_1^2 = k^2 + (N-1)\,l^2$.*

Z této rovnosti vyplývá, že celkový součet druhých mocnin amplitud zůstává stejný, dochází tedy pouze k přerozdělování. Pravděpodobnost systému se tedy od začátku stále rovná jedné.

Důkaz Z Tvrzení 1 víme, že difuzní operátor $D = WRW$, kde W i R jsou unitární. Operátor D je pak také unitární. Zachovává tedy normu a z definice normy 2.6 a skalárního součinu 2.5 dostáváme

$$\begin{aligned} \||\phi\rangle\| &= \|D|\phi\rangle\| \\ \sqrt{k^2 + (N-1)\,l^2} &= \sqrt{k_1^2 + (N-1)\,l_1^2} \\ k^2 + (N-1)\,l^2 &= k_1^2 + (N-1)\,l_1^2. \end{aligned} \tag{3.17}$$

Tvrzení 3 *Mějme stavový vektor o $N \geq 8$ stavech před krokem 2a algoritmu s amplitudou stavu S splňujícího podmínku $C(S) = 1$ rovnou k_0 a všemi ostatními rovnými l. Nechť navíc platí $0 < k_0 < \frac{1}{\sqrt{2}}$ a $l > 0$. Pak po provedení kroku 2a a 2b je změna k, kterou označíme Δk, zespoda omezená $\Delta k > \frac{1}{2\sqrt{N}}$ a l zůstane kladné.*

To znamená, že v každém opakování algoritmu se amplituda u hledaného stavu zvětší. Při dostatečném opakování se dostaneme s amplitudou k u hledaného stavu nad hranici $\frac{1}{\sqrt{2}}$. Jelikož druhá mocnina amplitudy odpovídá pravděpodobnosti změření kvantového systému v daném stavu, dostáváme pravděpodobnost změření hledaného stavu alespoň $\frac{1}{2}$.

Důkaz Užitím Tvrzení 2 dostáváme

$$k_1 = -\left(\frac{2}{N} - 1\right) k_0 + 2\frac{(N-1)}{N} l = \left(1 - \frac{2}{N}\right) k_0 + 2\left(1 - \frac{1}{N}\right) l \tag{3.18}$$

$$\Delta k = k_1 - k_0 = -\frac{2}{N} k_0 + 2\left(1 - \frac{1}{N}\right) l. \tag{3.19}$$

Z platnosti $0 < k_0 < \frac{1}{\sqrt{2}}$ a Důsledku 2.2 vyplývá

$$\begin{aligned} 1 = k^2 + (N-1)\,l^2 &< 2\,(N-1)\,l^2 \\ \frac{1}{2} < (N-1)\,l^2 &< Nl^2 \\ \frac{1}{2N} &< l^2 \\ |l| &> \frac{1}{\sqrt{2N}}. \end{aligned} \tag{3.20}$$

Jelikož je z předpokladu l kladné, je $l > \frac{1}{\sqrt{2N}}$. Po dosazení do změny k podle rovnice 3.19 a za předpokladu $N \geq 8$ dostáváme

$$\begin{aligned}
\Delta k &= -\frac{2}{N}k_0 + 2\left(1 - \frac{1}{N}\right)l > -\frac{2}{\sqrt{2}N} + \frac{2\left(1 - \frac{1}{N}\right)}{\sqrt{2N}} \\
&> -\frac{2}{\sqrt{2}N} + \frac{2N-2}{N\sqrt{2N}} > \frac{-2\sqrt{N} + 2N - 2}{N\sqrt{2N}} \\
&> \frac{N + \left(N - 2\sqrt{N} - 2\right)}{N\sqrt{2N}} > \frac{N}{N\sqrt{2N}} \\
&> \frac{1}{2\sqrt{N}}.
\end{aligned} \tag{3.21}$$

Pro důkaz $l_1 > 0$ si povšimneme, že $k < 0$ a $l > 0$. Z předpokladu máme $N \geq 8$ a z předchozího odstavce víme $l > \frac{1}{2\sqrt{N}}$ a z předpokladů $0 < k_0 < \frac{1}{\sqrt{2}}$. Nyní ukážeme

$$\left|\frac{k}{l}\right| = \frac{|k|}{|l|} < \frac{\frac{1}{2}}{\frac{1}{2\sqrt{N}}} = \frac{2\sqrt{N}}{2} = \sqrt{N}, \tag{3.22}$$

čímž splníme i poslední podmínku Důsledku 2.1 a díky němu bude l kladné.

3.4 Počet opakování

Kromě podmíněného otočení fáze v kroku 2a, který je závislý na charakteru prohledávané databáze, nezmiňuje Grover počet opakování. Jak již bylo řečeno, je počet opakování Groverova algoritmu $O\left(\sqrt{N}\right)$. Existuje však také spodní omezení počtu opakování, které je $\Omega\left(\sqrt{N}\right)$ [1]. Z toho vyplývá, že algoritmus je optimální. Postupem času se objevilo několik tvrzení o přesném počtu opakování, ale z tohoto původního popisu nemůžeme vyvodit přesnou hodnotu. Máme pouze horní odhad, jelikož v každém opakování máme spodní odhad změny amplitudy, kterou chceme dostat nad určitou hranici. Na počátku jsou všechny stavy v rovnocenné superpozici, tedy $\frac{1}{\sqrt{N}}$, a výslednou amplitudu chceme alespoň $\frac{1}{\sqrt{2}}$. Každým opakováním se amplituda zvětší alespoň o $\frac{1}{2\sqrt{N}}$, takže nejvýše po

$$r = \frac{\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{N}}}{\frac{1}{2\sqrt{N}}} = \frac{\frac{\sqrt{N}-\sqrt{2}}{\sqrt{2}\sqrt{N}}}{\frac{1}{2\sqrt{N}}} = \frac{\sqrt{N}-\sqrt{2}}{\sqrt{2}\sqrt{N}} \cdot \frac{2\sqrt{N}}{1} = \sqrt{2}\sqrt{N} - 2 \tag{3.23}$$

opakováních dostáváme amplitudu větší než $\frac{1}{\sqrt{2}}$. Pokud by změřený stav nebyl hledaný, spustíme algoritmus znovu. Pravděpodobnost chyby by nám tak kvadraticky klesala.

Dalším nepopsaným případem je příliš mnoho opakování. Intuitivně je vidět, že pokud bude amplituda u hledaného stavu příliš velká, vyjde nám průměr všech amplitud záporný a amplitudy ostatních stavů se tak překlopí do záporných hodnot. Amplituda u hledaného stavu se zase začne zmenšovat o přibližně dvojnásobek

hodnoty průměru. Dalšími opakováními by se amplitudy opět vyrovnaly a měli bychom výchozí stavový vektor s opačnou fází. Jak je vidět, přesný počet opakování je stejně důležitý jako algoritmus samotný. Tomuto problému se lze ale jednoduše vyhnout tak, že zvolíme $\lambda > 1$ a vyzkoušíme postupně počty opakování $\lambda, \lambda^2, \lambda^3, \ldots, \sqrt{2}\sqrt{N} - 2$. Stále bychom zůstali v časové složitosti $O\left(\sqrt{N}\right)$.

3.5 Geometrický popis

Protože Groverův algoritmus je velmi zajímavý pro budoucí možné použití, zabývalo se jím nezávisle na sobě mnoho lidí a vytvořilo se tak více náhledů na jeho princip. Dalším popisem je geometrická abstrakce publikovaná například v knize Roberta Špalka [13], kterou popíši dále v této podkapitole, při které nepočítáme s hodnotami amplitud, ale s úhly. Vlastní algoritmus je stejný, jen se o něm jinak přemýšlí a jinak se dokazuje.

Algoritmus vyhledává mezi N stavy, které jsme použili jako báze stavového prostoru. Na začátku máme opět kvantový stav $|\phi\rangle$ v rovnocenné superpozici všech stavů. Počáteční vektor je vlastně ve směru tělesové úhlopříčky N-rozměrné krychle. Skalární součin, který také odpovídá kosinu úhlu, počátečního vektoru a libovolného bázového vektoru, tedy i hledaného směru, je $\frac{1}{\sqrt{N}}$. Stavový vektor je tedy odkloněn od bázových vektorů o stejný úhel, pojmenujeme ho $\alpha = \frac{\pi}{2} - \theta$

$$\cos\alpha = \cos\left(\frac{\pi}{2} - \theta\right) = \sin\theta = \frac{1}{\sqrt{N}}. \tag{3.24}$$

Budeme se tedy snažit zvětšit původní úhel θ z hodnoty $\arcsin\frac{1}{\sqrt{N}}$ na $\frac{\pi}{2}$, kdy bychom měli největší pravděpodobnost změření správného výsledku $\sin^2\theta = 1$.

Tvrzení 4 *Kvantový stav leží po celou dobu výpočtu ve dvoudimenzionálním reálném podprostoru V s bázovými vektory*

$$\begin{aligned} |\phi_0\rangle &= \frac{1}{\sqrt{N-1}} \sum_{i:x_i=0} |i\rangle \\ |\phi_1\rangle &= |i\rangle,\ \textit{kde}\ x_i = 1 \end{aligned} \tag{3.25}$$

Nadále tak budeme počítat se superpozicí pouze dvou vektorů.

Důkaz Náš výchozí kvantový stav můžeme s použitím úhlu θ a rovnice 3.24 rozepsat jako

$$\begin{aligned} |\phi\rangle &= \sqrt{\frac{N-1}{N}}|\phi_0\rangle + \sqrt{\frac{1}{N}}|\phi_1\rangle = \left(1 - \frac{1}{N}\right)|\phi_0\rangle + \frac{1}{\sqrt{N}}|\phi_1\rangle = \\ &= \cos\theta|\phi_0\rangle + \sin\theta|\phi_1\rangle, \end{aligned} \tag{3.26}$$

z čehož je patrné, že leží v prostoru V.

Nyní probereme krok algoritmu. Nejprve operátor U_ω otočení fáze u hledaného vektoru. Ten zobrazuje $|\phi_0\rangle \to |\phi_0\rangle$ a $|\phi_1\rangle \to -|\phi_1\rangle$, zachovává tedy prostor V. Operátor má vzhledem k bázi ϕ_0 a ϕ_1 prostoru V matici

$$U_\omega = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{3.27}$$

Dále se podíváme na difuzní operátor D použitý na vektor z prostoru V. Použijeme Tvrzení 1 z původního důkazu o rozložení operátoru D na součin WRW, definici Walsh-Hadamardovy transformace 2.28 a definici počátečního stavu algoritmu

$$\begin{aligned}
|i\rangle \overset{W=H^{\otimes n}}{\longrightarrow} & \frac{1}{\sqrt{2^n}} \sum_{j\in\{0,1\}^n} (-1)^{i\bullet j} |j\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} (-1)^{i\bullet j} |j\rangle \\
\overset{R}{\longrightarrow} & \frac{1}{\sqrt{N}} (-1)^{i\bullet 0} |0\rangle - \frac{1}{\sqrt{N}} \sum_{j=1}^{N-1} (-1)^{i\bullet j} |j\rangle = \\
= & -\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} (-1)^{i\bullet j} |j\rangle + \frac{2}{\sqrt{N}} |0\rangle \\
\overset{W=H^{\otimes n}}{\longrightarrow} & -|i\rangle + \frac{2}{\sqrt{N}} \cdot \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} (-1)^{0\bullet j} |j\rangle = -|i\rangle + \frac{2}{\sqrt{N}} \cdot \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle = \\
= & -|i\rangle + \frac{2}{\sqrt{N}} \cdot |\phi\rangle = -|i\rangle + 2|\phi\rangle \cdot \langle\phi|i\rangle = \\
= & -|i\rangle + \frac{2}{\sqrt{N}} \cdot |\phi\rangle,
\end{aligned} \tag{3.28}$$

kde $i \bullet j$ je bitový skalární součin. Difuzní operátor opět zachovává prostor V a vzhledem k bázi ϕ_0 a ϕ_1 prostoru V ho lze zapsat jako

$$D|i\rangle = -|i\rangle + 2|\phi\rangle \cdot \langle\phi|i\rangle = -|i\rangle + 2|\phi\rangle\langle\phi| \cdot |i\rangle \tag{3.29}$$

$$\begin{aligned}
D = & -I + 2|\phi\rangle\langle\phi| = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} + 2 \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \cdot (\cos\theta, \sin\theta) = \\
= & \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} + 2 \begin{pmatrix} \cos^2\theta & \sin\theta\cos\theta \\ \sin\theta\cos\theta & \sin^2\theta \end{pmatrix} = \\
= & \begin{pmatrix} 2\cos^2\theta - 1 & 2\sin\theta\cos\theta \\ 2\sin\theta\cos\theta & 2\sin^2\theta - 1 \end{pmatrix} = \\
= & \begin{pmatrix} 2\cos^2\theta - \cos^2\theta - \sin^2\theta & \sin 2\theta \\ \sin 2\theta & 2\sin^2\theta - \cos^2\theta - \sin^2\theta \end{pmatrix} = \\
= & \begin{pmatrix} \cos^2\theta - \sin^2\theta & \sin 2\theta \\ \sin 2\theta & \sin^2\theta - \cos^2\theta \end{pmatrix} = \\
= & \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix}.
\end{aligned} \tag{3.30}$$

Pokud se podíváme na použití difuzního operátoru na bázové vektory, dostáváme

$$\begin{aligned} |\phi_0\rangle &\rightarrow \cos 2\theta|\phi_0\rangle + \sin 2\theta|\phi_1\rangle \\ |\phi_1\rangle &\rightarrow \sin 2\theta|\phi_0\rangle - \cos 2\theta|\phi_1\rangle. \end{aligned} \tag{3.31}$$

Máme tedy operátor podmíněného otočení fáze a difuzní operátor vyjádřeny maticemi o rozměrech 2×2 zachovávající prostor V.

Tvrzení 5 *V každém opakování algoritmu se stavový vektor pootočí o úhel* $2\theta = 2 \arcsin \frac{1}{\sqrt{N}}$.

Takto se po čase dostaneme s úhlem stavového vektoru blízko k $\frac{\pi}{2}$, kde je již velká pravděpodobnost změření hledaného stavu. Z tohoto je také zřejmé, že pokud budeme dále pokračovat, budeme se od hledaného stavu opět vzdalovat.

Důkaz Celé jedno opakování Groverova algoritmu G můžeme zapsat jako

$$G = D \cdot U_\omega = \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{pmatrix}. \tag{3.32}$$

Jedná se tedy o otočení fází vektoru o úhel 2θ podle operátoru otočení 2.22 v každém opakování algoritmu. Opakujeme-li tento postup t-krát, dostáváme otočení fází o úhel $2t\theta$, tedy $G^t = \begin{pmatrix} \cos 2t\theta & -\sin 2t\theta \\ \sin 2t\theta & \cos 2t\theta \end{pmatrix}$. Použitím na původní stavový vektor

$$G^t|\phi\rangle = \cos\left((2t+1)\,\theta\right)|\phi_0\rangle + \sin\left((2t+1)\,\theta\right)|\phi_1\rangle. \tag{3.33}$$

3.6 Počet opakování podruhé

Nyní se podíváme na přesný počet opakování. Algoritmus neustále pootáčí stavový vektor o úhel 2θ. Po $\frac{2\pi}{2\theta} = \frac{\pi}{\theta}$ opakováních jsme opět na začátku s výchozím stavovým vektorem. Mezitím ještě projdeme stavem, kdy po $\frac{\pi}{2\theta}$ opakováních budeme mít stavový vektor právě záporný výchozí. Kolik opakování je tedy nejvhodnější? Vezmeme-li původní úhel α a $\theta = \arcsin \frac{1}{\sqrt{N}}$, pak dostáváme pro počet opakování r

$$\alpha = \frac{\pi}{2} - \theta = 2r\theta \tag{3.34}$$

$$r = \frac{\frac{\pi}{\theta} - 2}{4} = \frac{\pi}{4\theta} - \frac{1}{2}. \tag{3.35}$$

Na konci algoritmu bude vzdálenost stavového vektoru a hledaného směru $O(\theta)$. Pravděpodobnost správné odpovědi tedy bude $O(\cos^2\theta)$. Pro špatné je to doplněk k jedné, tedy $O(1 - \cos^2\theta) = O\left(\sin^2\theta\right)$. Nyní se odvoláme na linearitu sinu pro velmi nízká čísla a dostáváme

$$r \approx \frac{\pi}{4}\sqrt{N} - \frac{1}{2} \tag{3.36}$$

a navíc pravděpodobnost špatné odpovědi algoritmu bude $O\left(\frac{1}{N}\right)$, což je pro velká N malé. Pokud bychom přesto obdrželi špatnou odpověď, spustíme algoritmus znovu. Tím nám pravděpodobnost chyby bude opět kvadraticky klesat.

Pokud bychom spustili algoritmus s $r = \frac{\pi}{4\theta} - \frac{1}{2}$ opakováními, kde $\theta = \arcsin\frac{1}{\sqrt{N}}$, dostaneme přesně výsledný směr. Úhel mezi stavovým vektorem a hledaným směrem by byl nulový, pravděpodobnost změření hledaného stavu tedy jedna [1]. Počet opakování si můžeme spočítat dopředu a pak spustit algoritmus. Počet opakování ale musí být celé číslo. Pokud není, pak při použití $\lfloor r \rfloor$ nebo $\lceil r \rceil$ opakování nedostaneme přesně hledaný vektor. Můžeme si pomoci tak, že provedeme $\lfloor r \rfloor$ opakování a jedno takzvané pomalé opakování. V něm změníme původní algoritmus tak, že budeme v kroku 2a násobit kvantovou fázi u hledaného stavu $e^{i\theta_1}$ místo -1 a v kroku 2b provedeme násobení $e^{i\theta_2}$ místo -1, tedy operátor D se bude rovnat $e^{i\theta_2}I + 2P$. Takovéto dva úhly nám mohou dát libovolné pootočení o úhel, který nám po $\lfloor r \rfloor$ opakováních chyběl a docílit tak přesného natočení k hledanému směru.

Jak je vidět, dává nám tento popis daleko více informací o počtu opakování než původní Groverův. Dokonce nám umožňuje provést přesný algoritmus, který nám vrátí výsledek s pravděpodobností rovnou jedné.

3.7 Rozšíření

Na začátku kapitoly popsaný Groverův algoritmus hledá v databázi, kde pouze jeden stav splňuje hledanou podmínku. Co dělat v případě, že takových stavů je v databázi více, řekněme k? Můžeme použít stejný algoritmus, jen upravíme počet opakování na

$$r \approx \frac{\pi}{4}\sqrt{\frac{N}{k}} - \frac{1}{2}. \tag{3.37}$$

A to z důvodu, že se stavový vektor natáčí k více směrům najednou, přičemž se opět musíme zastavit při natočení k nim co možná nejbližším. Jednoduchý náhled nám dá opět geometrický popis, kde se nám změní počáteční úhel z $\arcsin\frac{1}{\sqrt{N}}$ na $\arcsin\sqrt{\frac{k}{N}}$. Dále vše plyne z počtu opakování pro tento případ. Pokud neznáme počet vyhovujících stavů, můžeme algoritmus postupně pouštět s $\frac{\pi}{4}\sqrt{N}, \frac{\pi}{4}\sqrt{\frac{N}{2}}, \frac{\pi}{4}\sqrt{\frac{N}{4}}, \ldots$ opakováními. Pro nějaké k algoritmus hledaný stav najde s dostatečnou pravděpodobností. Pak již stačí nalezený stav označit, abychom ho dále nenacházeli, a spustit algoritmus znovu, tentokráte s menším k. Takovýchto pokusů uděláme nanejvýš $\frac{\pi}{4}\sqrt{N}\left(1 + \frac{1}{\sqrt{2}} + \frac{1}{2} + \cdots\right)$, což je stále $O\left(\sqrt{N}\right)$.

[1] Proto se této úpravě říká exaktní nebo také přesný algoritmus.

Kapitola 4

Popis algoritmu na simulaci kvantového počítače

4.1 QCL

Praktické využití kvantových počítačů je ještě otázkou budoucnosti, ale dobře definované základy jejich výpočetních možností dovolují vznik programovacích technik a jazyků. Jeden takový programovací jazyk a jeho simulátor pro použití na klasických počítačích, zatím pouze pro Linux, vytvořil Bernhard Ömer [8]. Jmenuje se QCL, neboli Quantum Computer Language, a podobá se programovacímu jazyku C s přidanými typy pro qubity a pro ně vestavěnými funkcemi. Ömerovo podání Groverova algoritmu bych rád představil v této kapitole.

Program dostane jako vstup pořadové číslo n hledané položky. Bude hledat toto číslo mezi $N = 2^{\lfloor \log_2 n \rfloor + 1}$ položkami. Použije $\log_2 N$ qubitů, na kterých bude reprezentovat pořadová čísla stavů. Výsledkem bude pořadové číslo hledaného stavu uložené bitově na těchto qubitech.

Zdrojový kód popisovaný níže je součástí programového balíku simulátoru QCL. Simulátor stačí stáhnout z domovské stránky

`<http://tph.tuwien.ac.at/%7Eoemer/qcl.html>`,

rozbalit a nechat přeložit příkazem `make`. Spustitelný program se bude jmenovat `qcl` a bude vytvořen v adresáři simulátoru. Po spuštění stačí načíst zdrojový kód Groverova algoritmu příkazem `include "lib/grover.qcl"` a spouštět funkci `grover()` s hledaným číslem.

4.2 Zdrojový kód

Protože dotazovací funkce na splnění podmínky není důležitá, ale bylo by vhodné, aby byla kvantová, byla vytvořena funkce, která vrací právě hledané číslo n. Tato funkce vezme qubitový registr a qubitům na místech, kde má v binárním zápise číslo n nulu, neguje hodnotu. Tak se původní stav n stane stavem $N - 1$ reprezentovaným

samými jedničkami. Ten je pak označen kontrolním qubitem. Nakonec se qubitový registr vrátí do původní polohy, jen s tím, že stav n je označen.

```
qufunct query( qureg x, quvoid f, int n){
 int i;
 for i = 0 to #x - 1 {      // x -> NOT ( x XOR n )
  if( not bit( n, i )){ Not( x[ i ] ); }
 }
 CNot( f, x );              // negace f pokud x = 1111...
 for i = 0 to #x - 1 {      // x <- NOT ( x XOR n )
  if( not bit( n, i )){ ! Not( x[ i ] ); }
 }
}
```

Nás zajímá celý operátor podmíněného otočení fáze. Ten můžeme vyjádřit jako posloupnost spočítání podmínek, kdy je hledaný stav označen příznakem, podmíněné otočení fáze označeného stavu, a opětovné vrácení výpočtu podmínek, odstranění označení. Otočíme tak fázi právě hledanému stavu.

```
query( q, f, n );           // vypočítání C( q )
CPhase( pi, f );            // otočení fáze stavu n
! query( q, f, n );         // odstranění C( q )
```

Dále se podíváme na difuzní operátor, který je implementován jako $-D = -WRW = W\left(-R\right)W$ podle vztahu 3.3 a 3.4. Protože znaménko stavového vektoru v jednotlivých opakováních nehraje roli, takovéto zjednodušení nevadí. Na qubitovém registru se provede nejprve Walsh-Hadamardova transformace. Pak se kvantový stav neguje, aby se stav $|0\rangle$ stal stavem $|N-1\rangle$, tedy řetězec samých jedniček, aby se dalo použít funkce `CPhase()`, která u něj otočí fázi. Nakonec se stav neguje zpět a provede se Walsh-Hadamardova transformace. Otáčíme tedy fázi pouze u prvního stavu, u ostatních zůstává stejná. V definici R je tomu naopak.

```
operator diffuse( qureg q ){
 H( q );                    // Walsh-Hadamardova transformace
 Not( q );                  // negace q
 CPhase( pi, q );           // otočení fáze q stavu 1111...
 ! Not( q );                // negace q zpět
 ! H( q );                  // zpětná Walsh-Hadamardova transformace
}
```

Nyní se již můžeme podívat na celkovou funkci na hledání podle Groverova algoritmu. Vstupem je hledané pořadové číslo n a výstupem nalezené pořadové číslo n. Počet kroků je snížen na $\lceil \frac{\pi}{8}\sqrt{N} \rceil$, což podle rovnice 3.34 postačuje k dosažení pravděpodobnosti změření hledaného stavu $\frac{1}{2}$. Následuje opakování kroku algoritmu složeného z počítání podmínek, podmíněného otáčení a difuzního operátoru. Nakonec změříme výsledný stav a pokud je hledaným pořadovým číslem, pak algoritmus končí, jinak se pustí znovu od začátku.

```
procedure grover( int n ){
 int l = floor( log( n, 2 )) + 1;
                              // počet qubitů
 int m = ceil( pi / 8 * sqrt( 2 ^ l ));
                              // počet opakování
 int x;
 int i;
 qureg q[ l ];
 qureg f[ 1 ];
 {
  reset;
  H( q );                     // příprava superpozice
                              // Walsh-Hadamardovou transformací
  for i = 1 to m {            // hlavní smyčka
   query( q, f, n );          // vypočítání C( q )
   CPhase( pi, f );           // otočení fáze stavu n
   ! query( q, f, n );        // odstranění C( q )
   diffuse( q );              // difuzní operátor
  }
  measure q, x;               // měření
  print "measured", x;
 } until x == n;
}
```

Kapitola 5

Závěr

5.1 Srovnání

Ukázali jsme, že Groverův algoritmus by mohl být kvadraticky rychlejší než hledací algoritmy na klasických počítačích. Dosahuje dokonce optimální rychlosti algoritmu pro hledání v nesetříděné databázi, které by se dalo na kvantovém počítači dosáhnout. Pokud nemáme o databázi žádné další informace, na kterých by se dalo postavit efektivnější prohledávání než postupné, je tedy tento algoritmus nejlepším možným. Není sice jednoduché vytvořit kvantovou funkci pro porovnávání hodnot podle podmínky, ale algoritmus samotný je jednoduchý pro výpočet.

Původní Groverův popis a důkaz jsou založeny na vlastnostech čísel, konkrétně průměrování, a jsou spíše technické. Navíc nám neříkají mnoho o praktickém použití algoritmu, hlavně o počtu opakování. Oproti tomu geometrický popis, který přestože vychází z naprosto stejného zápisu algoritmu, je k nám velmi štědrý. Jednak dává srozumitelný a přehledný náhled v podobě přibližujících se vektorů ve dvou dimenzích, jednak obsahuje dostatek informací pro zjištění počtu opakování. Navíc jde tak daleko, že podle něj můžeme vytvořit přesný hledací algoritmus, ve kterém nám nehraje roli náhoda a máme jistotu nalezení správné odpovědi již napoprvé. Tento popis významně doplňuje původní Groverův, který byl určen spíše k ukázání, že je problém hledání řešitelný. Geometrický popis nám již říká jak. Na obou popisech jsou velmi názorně a elegantně předvedeny všechny užitečné myšlenky kvantového počítání. Pro výukové účely algoritmu bych dal přednost názornějšímu geometrickému popisu.

Algoritmus obsahuje právě základy kvantového počítání a tak jej lze bez obtíží využívat i na nejjednodušších kvantových počítačích. Bude tedy s největší pravděpodobností k použití již s první generací kvantových počítačů. Pro ty budou nejspíše vytvořeny vlastní programovací jazyky podobné prvnímu assembleru pro klasické počítače, ale použití rozsáhlejších jazyků nakonec převládne.

5.2 Použití

Budoucí použití Groverova algoritmu na kvantovém počítači je především dáno kvadratickým zrychlením vyhledávání. Při běhu algoritmů na klasických počítačích

můžeme sice vědět další informace o databázi, které mohou pomoci při hledání, ale toto takzvané předtřídění spotřebuje také určitý čas. Proto by mohlo být výhodnější vše přenechat kvantovému počítači.

Groverův algoritmus se téměř vždy spojuje s myšlenkou prohledávání databáze. Tento algoritmus ale také umožňuje na základě počítání hodnot původní funkce simulovat funkci inverzní, neboť ve své podstatě je vlastně hledáním inverzní funkce.

Princip Groverova algoritmu byl již použit pro jiné algoritmy, například pro odhadnutí hodnoty průměru nebo mediánu, pro urychlení řešení NP-úplných problémů, pro hledání největší bipartitní a nebipartitní části grafu, nebo pro hledání toků v sítích.

Literatura

[1] BENNETT, C.H.; BERNSTEIN, E.; BRASSARD, G.; VAZIRANI, U. The strengths and weaknesses of quantum computation. In *SIAM Journal on Computing*, 1997, 26 (5) : 1510–1523.

[2] BROGLIE, L.de. *Researches on the quantum theory*. PhD thesis. Paris, 1924.

[3] DOBŠÍČEK, M. *Úvod do kvantového počítání* [on-line texty k přednášce], Praha, 2005. [citováno 2006-12-21]. ČVUT Praha. Fakulta elektrotechnická. Katedra počítačů. Dostupné z World Wide Web: <http://www.scycore.com/courses/index.html>.

[4] EINSTEIN, A. On a heuristic viewpoint concerning the production and transformation of light. In *Annalen der Physik*. 17. Leipzig, 1905, 132–148.

[5] GROVER, L.K. A Fast Quantum Mechanical Algorithm for Database Search. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC)*. Philadelphia, 1996, 212–219. Dostupný také z World Wide Web: <http://arxiv.org/pdf/quant-ph/9605043>.

[6] KUPČA, V. *Teorie a perspektiva kvantových počítačů*. Praha, 2001. ČVUT Praha. Fakulta elektrotechnická. Dostupný také z World Wide Web: http://cml.fsv.cvut.cz/%7Ekupca/qc/>.

[7] MOORE, G.E. Cramming more components onto integrated circuits. In *Electronics Magazine*. 19. 4. 1965, vol. 38, no. 8. Dostupný z World Wide Web: <ftp://download.intel.com/museum/Moores%5FLaw/Articles-Press%5FReleases/Gordon%5FMoore%5F1965%5FArticle.pdf>.

[8] ÖMER, B. *Quantum Programming in QCL* [on-line]. Vienna, 2000. Document version of 2000-01-20 [citováno 2006-12-21]. Institute of Information Systems, Technical University of Vienna. Dostupný z World Wide Web: <http://tph.tuwien.ac.at/%7Eoemer/doc/quprog.pdf>.

[9] PERRY, R.T. *The Temple of Quantum Computing* [on-line]. Document version 1.1 of 2006-04-29 [citováno 2006-12-21]. Dostupný z World Wide Web: <http://www.toqc.com/TOQCv1%5F1.pdf>.

[10] PLANCK, M. On the law of distribution of energy in the normal spectrum. In *Annalen der Physik*. 4. Leipzig, 1901, 553–563.

[11] SCHRÖDINGER, E. Quantisierung als Eigenwertproblem. In *Annalen der Physik*. 79. Leipzig, 1926, 361–376.

[12] Verlagsgruppe Georg von Holtzbrinck GmbH. *Scientific American* [on-line magazine]. Document version of 2007-02-13 [citováno 2007-04-29]. First "Commercial" Quantum Computer Solves Sudoku Puzzles. Dostupný z World Wide Web: <http://www.sciam.com/article.cfm?articleID=BD4EFAA8-E7F2-99DF-372B272D3E271363>.

[13] ŠPALEK, R. *Quantum Algorithms, Lower Bounds and Time-Space Tradeoffs*, Amsterdam, 2006, Institute for Logic, Language & Computation, Universiteit van Amsterdam, ISBN-10: 90-5776-155-6.

[14] University of Cambridge. Centre for Quantum Computation. *Quantiki* [on-line encyklopedie]. Cambridge, 2006 [citováno 2006-12-21]. Introduction to Quantum Theory. Dostupný z World Wide Web: <http://www.quantiki.org/wiki/index.php/Introduction%5Fto%5FQuantum%5FTheory>.

[15] University of Cambridge. Centre for Quantum Computation. *Quantiki* [on-line encyklopedie]. Cambridge, 2005 [citováno 2006-12-21]. What is Quantum Computation. Dostupný z World Wide Web: <http://www.quantiki.org/wiki/index.php/What%5Fis%5FQuantum%5FComputation%3F>.

[16] University of Cambridge. Centre for Quantum Computation. *Quantiki* [on-line encyklopedie]. Cambridge, 2006 [citováno 2006-12-21]. Grover's search algorithm. Dostupný z World Wide Web: <http://www.quantiki.org/wiki/index.php/Grover%27s%5Fsearch%5Falgorithm>.

[17] Wikimedia Foundation. *Wikipedia* [on-line encyklopedie]. 2006 [citováno 2006-12-21]. Quantum Computer. Dostupný z World Wide Web: <http://en.wikipedia.org/wiki/Quantum%5Fcomputer>.