

Kvantinė kompiuterija

Tadas Paulauskas Julius Ruseckas

2022-12-21

Turinys

Pratarmė	7
Kaip skaityti šią knygą	7
Padėka	8
Apie autorius	8
Prisidėti prie knygos	8
Licencija	8
I Kvantinės kompiuterijos apžvalga	9
1.1 Įvadas	9
1.2 Kvantinės kompiuterijos pradmenys	13
1.3 Tiuringo mašina	15
1.4 Skaičiavimų ištekliai	16
1.5 Kvantiniai bitai	19
1.6 Kvantinės informacijos apdorojimas	22
1.7 Skaičiavimo procesas	25
1.8 Kvantinių kompiuterių charakteristikų palyginimas	27
1.9 Dekoherencijos trukmė ir loginių vartų tikslumas	28
II Matematinių įrankių rinkinys	31
2.1 Tiesinė algebra	31
2.2 Kompleksiniai skaičiai	35
2.3 Vidinė vektorių sandauga	37
2.4 Kubito reprezentacija Blocho sferoje	39
2.5 Tiesiniai operatoriai ir matricos	41
2.6 Unitariniai ir ermitiniai operatoriai	45
2.7 Diadinė operatorių dekompozicija	47
2.8 Matricos pėdsakas	49
2.9 Tenzorinė vektorių sandauga	50
2.10 Tenzorinė operatorių sandauga	51
2.11 Operatorių funkcijos	53
III Kvantinės mechanikos pagrindai	55
3.1 Kvantinės mechanikos postulatai	55
3.2 Kvantinis supynimas	60
3.3 Tankio operatorius	62
3.4 EPR paradoksas	66

IV Kvantiniai loginiai vartai ir grandinės	69
4.1 Vieno kubito loginiai vartai	69
4.2 Kvantinių grandinių lygibės ir atvirkštiniai loginiai vartai	72
4.3 Kubitų būsenų matavimas	74
4.4 Dviejų kubitų loginiai vartai <i>CNOT</i>	75
4.5 Tofoli loginiai vartai	77
4.6 SWAP ir Fredkin loginiai vartai	78
4.7 Bendro tipo sąlyginiai loginiai vartai <i>cU</i>	79
4.8 Bendro tipo būsenų matavimai	84
4.9 Universalių loginių vartų rinkinys	86
V Kvantinė informacija ir ryšiai	87
5.1 Kvantinės informacijos kopijavimas	87
5.2 Kvantinė teleportacija	88
5.3 Kvantinio supynimo sukeitimasis	91
5.4 Kvantinė kriptografija	93
5.5 Lokalios operacijos ir klasikiniai ryšiai	98
5.6 Belo nelygybė	100
VI Skaičiavimai kvantiniu kompiuteriu	105
6.1 Bazinių vektorių numeracija	105
6.2 Funkcinis skaičiavimas	106
6.3 Kvantinis paralelizmas	107
6.4 Duomenų kodavimo būdai	108
6.5 Doičo algoritmas	110
6.6 Kvantinė paieška ir Groverio algoritmas	111
6.7 Hadamardo ir SWAP testai	119
VII Furjė transformacija ir jos taikymai	123
7.1 Kvantinė Furjė transformacija	123
7.2 Furjė transformacijos realizavimas kvantinėje grandinėje	125
7.3 Funkcijos periodiškumo paieška	127
7.4 Kvantinis fazės nustatymo algoritmas	128
7.5 Tiesinių lygčių sprendimas HHL algoritmu	131
VIII Kvantinių sistemų modeliavimas ir mašininis mokymasis	135
8.1 Dinaminių sistemų modeliavimas	135
8.2 Erdvinės Šriodingerio lygties sprendimo algoritmas	138
8.3 Mašininis mokymasis	144
IX Kvantinių klaidų aptikimas ir taisymas	151
9.1 Klasikinės ir kvantinės klaidos	151
9.2 Kvantinis supynimas su aplinka ir klaidų atsiradimas	153
9.3 Bito apvertimo klaidos aptikimas ir taisymas	155
9.4 Fazės apvertimo klaidos aptikimas ir taisymas	158
9.5 Tolydžiosios klaidos	158
9.6 Bendrieji klaidų taisymo principai	159
9.7 Kvantinė Hamingo riba	160
9.8 Šoro 9 kubitų kodas	161
9.9 Kodų stabilizatoriai	164

<i>TURINYS</i>	5
9.10 Klaidoms atsparus skaičiavimas	167
9.11 Kvantinis tūris	170
A Debesyje pasiekiami kvantiniai kompiuteriai ir simuliatoriai	175

Pratarmė

Ši atviros prieigos knyga yra įvadas į kvantinę kompiuteriją. Ji skirta skaitytojams, kurie nėra anksčiau susipažinę su šia įdomia sritimi. Knygos skyriuose pasakojama, kaip informacija yra koduojama kvantiniuose bituose ir kaip ji apdorojama kvantiniame procesoriuje.

Idėja parengti knygą apie kvantinę kompiuteriją lietuvių kalba kilo *Covid-19* pandemijos metu. Dėl laboratorių uždarymo ir sulėtėjusių mokslinių tyrimų atsirado galimybė daugiau laiko skirti šiai sparčiai besivystančiai sričiai. Knyga prasidėjo veikiau kaip autorų sukauptų teorinių žinių susisteminas į vieną dokumentą asmeniniams darbams. Perorientuoti dokumentą į platesnei auditorijai skirtą knygą buvo nutarta atsižvelgiant į tai, kad jos rašymo metu dar neegzistavo nė viena lietuviška knyga apie kvantinę kompiuteriją. Tikimės, kad knyga padės geriau susipažinti su šia technologija ir paskatins ne vien platesnį susidomėjimą, bet ir jos plėtrą Lietuvoje.

Kvantinė kompiuterija apjungia keletą mokslo sričių – fiziką, kompiuterių mokslą, matematiką, inžinerinius mokslus. Knyga koncentruota į kompiuterių mokslo priartėjimą prie šios srities ir apibūdina informacijos apdorojimą kvantiniame procesoriuje abstrakčiu lygmeniu. To turėtų pakakti norint atsispirti nuo pradinio taško ir toliau plėsti žinias norima kryptimi.

Ši knyga buvo parašyta remiantis svarbiausiomis esamomis knygomis ir moksliniais straipsniais apie kvantinę kompiuteriją. Tarp naudotų knygų yra šios srities „biblija“ tapusi autorų Nielsen ir Chuang „Quantum Computation and Quantum Information“ (Nielsen ir Chuang 2000), Nakahara ir Ohmi „Quantum Computing – From Linear Algebra to Physical Realizations“ (Nakahara ir Ohmi 2008), ir kitos. Išsamus literatūros sąrašas yra pateiktas knygos pabaigoje. Dalis medžiagos taip pat naudota iš T.P. doktorantūros metu klausyto kurso „Quantum computing“, kurį dėstė profesorius Friedland Shmuel, užrašų.

Kaip skaityti šią knygą

Skaitytojams, kurie nėra turėję tikslinių mokslo žinių ir nori patenkinti savo smalsumą - rekomenduojame skaityti tik pirmąjį ir penktąjį skyrius. Pirmajame skyriuje glaučiai apžvelgiama kvantinė kompiuterija ir palyginama su klasikine kompiuterija. Kituose skyriuose iš esmės sugržtama prie pirmajame skyriuje minėtų savokų bei algoritmų ir jie parodomai išsamiau. Penktajame skyriuje apžvelgiami kvantinės informacijos aspektai, tokie kaip kriptografija ir kvantinės informacijos siuntimas pasitelkiant teleportacijos metodą. Šie taikymai perteikia kvantinių bitų elgsenos keistenybes ir jų siūlomus privalumus. Kituose skyriuose po visų matematinių išraiškų pateikti konceptualūs paaiškinimai. Tad net ir neskaitant lygčių įmanoma susidaryti bendrą idėją apie pateiktų kvantinių algoritmų veikimo principą.

Norint įsisavinti visą pateiktą knygoje medžiagą, rekomenduotina turėti bent bakalauro lygio

tiesinės algebro žinių. Kvantisinės mechanikos žinios, be abejo, praverstų, tačiau nėra būtinės. Reikalingi matematiniai įrankiai ir kvantisinės mechanikos taisyklės, taikomos kvantisinėje kompiuterijoje, yra pateikiamos šioje knygoje. Jeigu turite pakankamai žinių minėtose srityse, tada antrąjį skyrių, kuris skirtas matematikos žinioms atgaivinti, galite atsiversti esant poreikiui. Knyga nebuvė specialiai orientuojama į vadovėlio formatą, tačiau skyrių seką turi progresiją, kuri gali būti panaudota vieno semestro bakalauro ar magistrantūros studentų kursui.

Padėka

Autoriai dėkoja Vaidui Pačebutui, Arūnui Krotkui, Audriui Alkauskui, Maženai Mackoit Sinkevičienei ir Artūrui Pupšiui už paramą, teksto skaitymą ir vertingus patarimus rengiant šią knygą.

Iliustracijos – Jovita Jankauskienė

Kalbos redaktorė – Rozita Znamenskaitė (KTU leidykla „Technologija“)

Apie autorius

Tadas Paulauskas įgijo fizikos mokslų daktaro laipsnį Ilinojaus universitete Čikagoje. Vėliau atliko tyrimus Monašo universitete Melburne. Šios knygos rašymo metu dirba Fizinių ir technologijos mokslų centre (FTMC), Optoelektronikos skyriuje. Mokslinių tyrimų sritis apima puslaidininkų fiziką, nanotechnologijas, medžiagų inžineriją.

Julius Ruseckas įgijo fizikos mokslų daktaro laipsnį Vilniaus universitete. Vėliau dirbo mokslinių darbų Vilniaus universiteto Fizikos fakulteto teorinės fizikos ir astronomijos institute. Pagrindinė mokslinių tyrimų sritis – kvantinė optika ir labai šaltos atominės dujos. Šios knygos rašymo metu dirba Baltijos pažangių technologijų institute (BPTI).

Prisidėti prie knygos

Norite prisidėti prie šios atvirai prieinamos knygos turinio kūrimo, redagavimo ar apipavidalinimo? Visada ieškome gabiu žmonių ir tyrėjų, kurie papildytų knygos kūrėjų gretas!

Rašykite mums į el. paštą info@kvantinekompiuterija.lt.

Visą knygos turinį bookdown formate galite pasiekti GitHub paskyroje.

Licencija



Internete pateikiama knygos versija gali būti naudojama pagal Creative Commons licenciją. Priskyrimas - Nekomerčinis naudojimas - Analogiškas platinimas 4.0 Tarptautinė (CC BY-NC-SA 4.0)

I skyrius

Kvantinės kompiuterijos apžvalga

1.1 Įvadas

Šiandieniniai kompiuteriai, kuriuos čia vadinsime klasikiniais, atvėrė kelią į informacijos amžių. Puslaidininkinių tranzistorių išradimas XX a. viduryje ir jų pagrindu formuojami mikroprocesoriai leido nuosekliai didinti kompiuterių skaičiuojamąją galią bei kompiuterių miniatiūrizacijos lygi. Sunku nepervertinti kompiuterių vaidmens technologinėje ir socialinėje žmonijos raidoje – nuo naujų medžiagų ir medikamentų kūrimo, atliekamų skaičiavimų nustatant erdvėlaiivių trajektorijas, mašininiu mokymusi grindžiamo savaeigio transporto, genų inžinerijos iki socialinių tinklų, paremtų į tinklą sujungtais kompiuteriais.

Moore'o dėsnis (angl. *Moore's law*) yra artimai susietas su skaičiuojamosios galios didėjimu per pastaruosius 40 metų. Šis dėsnis teigia, kad tranzistorių skaičius integruotose mikroprocesorių grandinėse padvigubėja maždaug kas dvejus metus. Didesnis tranzistorių skaičius procesoriuose gali leisti atlikti sudėtingesnes skaičiavimų operacijas, suteikti daugiau greitai pasiekiamos atminties ir paralelizuoti skaičiavimus. Moore'o dėsnis savo ruožtu reiškia komponentų mažėjimą; puslaidininkinių procesorių komponentų dydžiai siekia 1–2 nanometrus. Dvieuose nanometruose galima išrikiuoti apie dešimt atomų, tad, be praktinės, matome ir fundamentinę ribą klasikinių kompiuterių komponentų dydžiui – vienas atomas. Tai nulems, jog netolimoje ateityje prireiks kitų būdų patenkinti didėjantiems skaičiavimų spartos poreikiams. Tačiau technologinei raidai svarbiu užduočių skaičiavimo galimybės jau ir dabar yra itin ribotos.

Mažėjant klasikinių kompiuterių komponentų dydžiui, tenka neišvengiamai atsižvelgti į **kvantinius dydžio efektus** (angl. *quantum size-effects*). Medžiagos apimtį sumažinus iki nanometro skalės, ima ryškėti krūvininkų energijos lygmenų persitvarkymai, taip pat didėja tikimybė krūvininkams pabėgti iš medžiagos **kvantinio tuneliavimo** principu (angl. *quantum tunneling*). Sie ir kiti kvantiniai efektai padaro puslaidininkinių tranzistorių veikimą nenuspėjamą, ir reikia papildomų priemonių norint išlaikyti jų funkcionavimą. Pažangesnių nanomedžiagų gamybos principų įsisavinimas veikiausiai leis klasikinius procesorius padaryti energetiškai efektyvesnius ir padidinti jų spartą. Tačiau yra ir kitas sprendimo būdas – keisti patį klasikinį skaičiavimo modelį į tokį, kuris paremtas kvantinėmis taisyklėmis.

Kvantinė kompiuterija yra nauja skaičiavimų paradigma. Tai nulemia fundamentalaus klasikinės informacijos paketo bito pakeitimas kvantiniu bitu, vadinančiuoju **kubitu** (angl. *qubit*). Šios technologijos privalumai kyla iš tipinių kvantinių efektų – kubito gebėjimo būti skirtingose būsenose

vienu metu, vadinamojoje būsenų **superpozicijoje** (angl. *state superposition*), **interferencijos efektu** (angl. *interference*) ir **kvantinio supynimo** (angl. *quantum entanglement*). Kvantiinis procesorius yra sudarytas iš daugelio kubitų, o bendroje visų kubitų būsenų superpozicijoje yra koduojama ir apdorojama informacija. Kadangi kvantinių būsenų transformacijos, kurioms vykstant atliekamas informacijos apdorojimas, veikia visas būsenas superpozicijoje vienu metu, kvantinis kompiuteris iš pagrindų pasižymi didelėmis paralelizavimo galimybėmis. Kubitų gali būti įvairios kvantinės fizinės sistemos, kuriose įmanoma tiksliai kontroliuoti ir nuskaityti dvi skirtinges būsenas. Tarp jų yra elektronų ir atomų branduolių sukininės būsenos, sklindančių fotonų poliarizacija, elektroniniai ir vibraciniai atomų lygmenys, elektromagnetinio lauko rezonansai uždaroje ertmėje, krūvininkų skaičius superlaidininkuose.

Įdomu paminėti, kad skaičiavimai, pagrįsti kvantinėmis taisyklemis, buvo pasiūlyti kur kas anksčiau, nei dydžio efektai klasikiniuose kompiuteriuose galėjo kelti rūpesčių, maždaug 1980 metais. Tada tai atrodė arti fantastikos ribų, nes kontroliuoti kvantinių sistemų būsenas ir apsaugoti jas pakankamai ilgai nuo sunykimo, norint spėti atlikti skaičiavimus, itin keblu. Šiuo požiūriu technologinėje raidoje kvantinė kompiuterija žymi pradžią, kai įspūdingu tikslumu gebama kontroliuoti fundamentalių gamtos ingredientų kitimą laike. Kvantinė kompiuterija yra tik vienas šios pažangos vaisius, tačiau potencialiai turintis esminę rolę paveikti žmonių kasdienybei.

Rašydami šią knygą esame vadinamojoje **triukšmingų tarpinės skalės kvantinių technologijų** etape (angl. *noisy intermediate-scale quantum*, trumpinys NISQ). NISQ kvantiniai procesoriai turi 50-100 kubitų skaičių, pasižymi sparčiai atsirandančiomis klaidomis ir nenaudoja klaidų taisymo algoritmų. Tai neleidžia atlikti ilgų ir tikslų praktinės svarbos skaičiavimų.

Didėjant kvantinių kompiuterių skaičiuojamajai galiai verta tikėtis proveržių srityse, kuriose reikalingi intensyvūs skaičiavimai. Šios sritys apima medikamentų ir technologinių medžiagų kūrimą, branduolinę energetiką, fundamentinius mokslinius tyrimus, dirbtinio intelekto tobulinimą, finansinių rinkų modeliavimą, logistiką, klimato kaitos ir kitų kompleksinių sistemų vyksmų modeliavimą.

Kvantiniai kompiuteriai veikiausiai neištums klasikinių įrenginių, puikiai atliekančių daugybę funkcijų, bet bus naudojami greta. To pavyzdys – **grafikos apdorojimo procesorius** (angl. *graphics processing unit*), integruotas daugumoje kompiuterių. Palyginus su bendro naudojimo kompiuterio procesoriu, grafikos procesoriaus elektroninė architektūra suteikia galimybes paralelizuoti ir itin paspartinti tam tikro tipo duomenų apdorojimą. Galima panašiai įsivaizduoti ir su klasikiniu kompiuteriu integruotą kvantinį procesorių, į kurį nukreipiamos užduotys, reikalaujančios jo suteikiamų privalumų, taip realizuojant hibridinį kvantinį-klasikinį skaičiavimo metodą.

Knygoje pristatomomi kvantinės kompiuterijos pagrindai vadinamiesiems **universaliesiems kvantiniams kompiuteriams** (angl. *universal quantum computers*). Universalumas čia reiškia, kad jie nėra specializuoti spręsti specifinius uždavinius, bet gali atlikti bet kokį suformuluotą skaičiavimą. Šiuo metu plačiausiai vystomas kvantinio skaičiavimo modelis yra **grįstas loginiaiš vartais** (angl. *gate-based model*); jis dar vadinamas **kvantinių grandinių modeliu** (angl. *quantum circuit model*). Kaip rodo pavadinimas, jų žemiausio lygmens programavimas yra pagrįstas kvantinių loginiaiš vartais, kurie atlieka kubitų būsenų transformacijas. Kvantiame procesoriuje pasitelkiama keletas skirtingu loginių vartų, sudarančių universalų rinkinį; jų kombinacijomis galima įvykdyti visus įmanomus skaičiavimus. Tokiu principu veikia ir klasikiniai kompiuterių procesoriai, kuriuose elektroninės loginės grandinės atlieka pateiktos dvejetainės skaičių sekos apdorojimą. Skaičiavimų universalumas leidžia atsiriboti nuo konkrečios fizinės sistemos, realizuojančios kubitus, ir susitelkti į skaičiavimo užduočių formulavimą bei programavimą. Skirtingi kvantiniai procesoriai gali naudoti skirtingo tipo loginius elementus; tačiau tai tik reiškia, kad

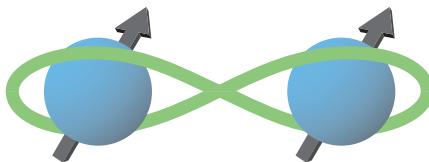
kvantinį algoritmą teks perkompliuoti naujais elementais – algoritmo pagrindas nekinta.

Ankstyvieji kvantinės kompiuterijos tyrinėtojai turi pirmenybę apsirrantant su pasaulyu, kuriam reikia žaisti pagal kvantines taisykles. Vertinant iš fizikos perspektyvos, kubitas – pati paprasčiausia kvantinė sistema, kuri nėra triviali. Kvantiniame kompiuteryje matome situaciją, kai iš daugelio paprastų sistemų iškyla nauji kompleksiškumo reiškiniai. Kvantinis superpozicijos principas yra pagrindinis to šaltinės; visgi tai kitokios prigimties superpozicija, nei aptinkama klasikinėse sistemose. Pavyzdžiui, skirtinių stygų vibracijos dažniai muzikos instrumente nusako skirtinges būsenas. Spragtelėjus stygą pirštu, ji greičiausiai bus sužadinta virpēti vienu metu keletu skirtinių dažnių, o tai savo ruožtu nusakys stovinčiųjų bangų (virpesių) superpoziciją. Taip pat ir bangos, keliaujančios sužadintu vandens paviršiumi skirtingomis kryptimis, yra būsenų superpozicija bei demonstruoja tokius reiškinius kaip interferencija, kada persiklojusių bangų amplitudė vietomis padidėja ir sumažėja.

Norint pamatyti skirtumus tarp klasikinių ir kvantinių sistemų įsivaizduokime instrumento stygą, kuri gali vibrnuoti tik dviem skirtingais dažniais. Vadinsime juos $|0\rangle$ ir $|1\rangle$. Stygos būseną galime užrašyti kaip šių virpesių sumą: $|S\rangle = a|0\rangle + b|1\rangle$; čia a ir b nusako atitinkamą virpesių amplitudę, tai yra garsumą. Siekdami padaryti šį pavyzdį artimesnį kvantinėms sistemoms, tvirtinsime, kad stygą bet kuriuo metu negali nevibrnuoti bent vienu iš šių dviejų dažnių. Keturių stygų violončelėje jos bendrą būseną bet kuriuo metu galima nusakyti įvardijant keturių stygų būsenas – $|S_1\rangle$, $|S_2\rangle$, $|S_3\rangle$, $|S_4\rangle$, ir todėl iš viso aštuonių amplitudės $\{(a_1, b_1), \dots, (a_4, b_4)\}$. Tačiau, jeigu šios stygos būtų kvantinės sistemos taip pat su dviem skirtingomis būsenomis (kubitai), nusakyti violončelės būsenai reikėtų įvardyti šešiolika amplitudžių! Papildomos būsenos kvantinėje violončelės versijoje atsiranda dėl to, kad galima ne vien tos pačios stygos būsenų superpozicija, bet ir bendra skirtinių stygų būsenų superpozicija. Pavyzdžiui, dvi galimos klasikinės violončelės būsenos yra kai visos keturios stygos vibrnuoja tik $|0\rangle$ dažniu, nusakoma amplitudėmis $\{(a_1, 0), (a_2, 0), (a_3, 0), (a_4, 0)\}$ ir kai vienu metu visos stygos vibrnuoja tik $|1\rangle$ dažniu, nusakoma amplitudėmis $\{(0, b_1), (0, b_2), (0, b_3), (0, b_4)\}$. Jeigu šios stygos būtų kvantinės, tada galima ir šių klasikinių būsenų superpozicija. Tai reiškia, vienu metu visos stygos vibrnuoja tik $|0\rangle$ dažniu ir nevibrnuoja $|1\rangle$, taip pat vibrnuoja $|1\rangle$ dažniu ir nevibrnuoja $|0\rangle$. Bendrai tariant, jeigu klasikinių stygų skaičius yra n , tokiai sistemai nusakyti bet kuriuo metu reikia $2n$ amplitudžių vertęs koduojant informaciją, taps akivaizdu, jog kvantinės sistemos gali savyje laikyti eksponentiškai daugiau informacijos.

Eksponentinis skirtumas tarp klasikinių ir kvantinių būsenų skaičiaus atsiranda dėl to, kad kvantinėse sistemoje galimos **supintosios būsenos** (angl. *entangled states*). Minėta superpozicijos būsena yra kvantinio supynimo pavyzdys, leidžiantis keletui „stygų” elgtis kaip viena „superstyga”. Supintos sistemos pasižymi kvantinėmis koreliacijomis, kurių neįmanoma imituoti klasiskai ir paaiškinti klasikine tikimybų teorija. Jos elgiasi panašiai kaip vienas darinys, net jeigu yra atskirtos viena nuo kitos dideliais atstumais. Pavyzdžiui, dviejų klasikinių bitų būseną galima pakeisti į bet kurią kitą iš keturių galimų kombinacijų $\{00, 01, 10, 11\}$ tik keičiant kiekvieno individualaus bito būseną. Supintojoje 2 kubitų būsenoje pakanka kontroliuoti vieno pasirinkto kubito būseną norint akimirksniu pakeisti jų bendrą būseną į bet kurią iš analogiškai keturių skirtinių.

Supintosiose sistemoje (žr. 1.1 pav.) dalis informacijos yra laikoma kvantinėse koreliacijose tarp kubitų būsenų. To gerą pavyzdį pateikia fizikas Johnas Preskillas. Įsivaizduokime, kad turime dvi šimto puslapių dydžio knygas – vieną įprastą klasikinę, o kitą kvantinę, kurioje informacija yra laikoma kvantinėse būsenose. Perskaitę vieną įprastos knygos puslapį sužinome 1% visos joje pateikiamos informacijos, tad perskaitę šimtą puslapių žinome viską, kas joje yra. Tačiau



1.1 pav.: Dviejų kubitų kvantinis supynimas

skaitydami kvantinę knygą ir atversdami kiekvieną naują puslapį gauname nenuuspėjamą turinį. Perskaitę visą šimtą puslapiai turėsime tik mažą idėją, apie ką rašoma knygoje, nes eksponentiškai didesnė dalis informacijos yra laikoma ne atskiruose puslapiuose, bet koreliacijose tarp jų. Norint sužinoti visą informaciją, laikomą kvantinėje knygoje, reikia „skaityti“ visus knygos puslapius vienu metu. Tai yra klasikinėje terpéje neturintis analogijų procesas.

Informacija, kuri koduojama ir apdorojama kvantinėse sistemose, vadinama kvantine informacija. **Kvantinės informacijos teorija** (angl. *quantum information theory*) yra nauja informatikos šaka ir skiriasi nuo klasikinės informatikos keletu esminių principų. Didžioji dalis informacijos, esančios kvantinių būsenų erdvėje, nėra prieinama. Norint sužinoti anksčiau minėto kvantinio instrumento būseną esančioje superpozicijoje, galimi atsakymai yra: visos keturios stygos vibrusoja pirmuoju $|0\rangle$ arba antruoju $|1\rangle$ dažniu. Kitaip tariant, atskleisdami kvantinėse superpozicijos būsenose laikomą informaciją gauname klasikinę informaciją, o superpozicija yra sugriaujama. Neįmanoma nuspėti, kuris rezultatas bus aptiktas; žinomas tik jų tikimybės, kurios yra proporcingsos amplitudžių dydžiams. Wooterso ir Zureko **uždraustojo kopijavimo teorema** (angl. *no-cloning theorem*) nusako kitą esminį skirtumą tarp kvantinės ir klasikinės informacijos. Ši teorema teigia, kad kvantinės informacijos neįmanoma kopijuoti. Tai nereiškia, kad žinodami informacijos, tai yra kvantinės būsenos, paruošimo žingsnius negalime jos tiksliai atkartoti. Šis principas veikiau rodo, kad gavus kvantinės informacijos paketą su nežinomu turiniu neįmanoma sukurti šio turinio identiškos kopijos.

Nors šie skirtumai atrodytų kaip keblumai, panaikinantys minėtus kvantinių sistemų privalumus, tačiau dar ne viskas prarasta. Pirmiausiai galima atkreipti dėmesį, kad nors apdorojamos informacijos apimtis tam tikrose užduotyse gali būti astronominio dydžio, tačiau praktikoje dažnai prireikia tik salygiškai mažos jos dalies, kad sužinotume norimą atsakymą ar perteiktume duomenis žmogiškuoju pavidalu. Dėl eksponentinio būsenų skaičiaus perteikti net ir 300 kubitu laikomą kvantinę informaciją į klasikinės atminties išteklius būtų neįmanoma, tam prireikytu maždaug 10^{90} amplitudžių verčių. Šis skaičius yra didesnis nei stebimoje Visatoje esančių atomų skaičius.

Ekstravagantiškos kvantinių sistemų savybės pačios savaime neužtikrina, kad jomis pagrįstas skaičiavimas bus pranašesnis už klasikinius. Galima pateikti tris geras priežastis, kodėl kvantinis skaičiavimo modelis visgi demonstruoja pranašumą. Pirma (ir akivaizdžiausia) priežastis slypi kvantinių sistemų modeliavime. Dėl eksponentiškai didelės kvantinių būsenų erdvės nėra žinoma, kaip efektyviai jas modeliuoti klasikiniu kompiuteriu. Čia ištekliai auga eksponentiškai su sistemos dydžiu (kubitu skaičiumi), o štai modeliuojami kvantiniu kompiuteriu jie bendrai auga tiesiškai. Vienas pavyzdys, kuriame kvantinių sistemų modeliavimas galėtų suteikti globaliai svarbų proveržį, yra vadinamas **azoto fiksacijos procesu** (angl. *nitrogen fixation process*). Šiame procese stipriomis trigubomis kovalentinėmis jungtimis molekulėse N_2 susijungę azoto atomai yra atskiriami ir kartu su kitais elementais transformuojami į amoniaką (NH_3) bei kitus nitratus, naudojamus pramoninei trąšų gamybai. Efektyviausiam žinomam procesui, vadinamam

Haberio-Bošo procesu (angl. *Haber-Bosch process*), reikia itin didelių slėgių ir temperatūrų šiai transformacijai atlkti, ir todėl išeikvojama ypatingai daug energijos. Azoto fiksacija yra taip pat esminis procesas biologinėse sistemosose atliekant biosintezę. Biologinės sistemos tai atlieka itin efektyviai įprastinėmis kambario sąlygomis naudojant nitrogenazės fermentą. Šio organinio cheminio proceso supratimas galėtų padaryti trąšų gamybą energetiškai žymiai efektyvesnę. Tačiau jis yra neįkandamas klasikiniams superkompiuteriams dėl per didelio laisvės laipsnių skaičiaus net ir sąlygiškai mažoje molekulėje.

Antroji priežastis slypi sunkių ar neįveikiamų klasikinių užduočių skaičiavimuose kvantiniu kompiuteriu. Jau dabar yra atrasta ženklių greitesnių kvantinių algoritmų už geriausius žinomus klasikinius. Tarp gerai žinomų yra pirminių skaičių skaidymas, tiesinių lygių sprendimas, paieška duomenų bazėse ir jų sąrašas sparčiai auga.

Galiausiai **skaičiavimų sudėtingumo** (angl. *computational complexity*) teorijos argumentai teigia, kad kvantiniu kompiuteriu galima efektyviau testuoti reikšmes iš **koreliuotųjų tikimybinių funkcijų** (angl. *correlated probability distribution*). Tokio tipo skaičiavimais yra dažnai grindžiami pirmieji **kvantinės viršenybės** (angl. *quantum supremacy*) demonstravimai prieš klasikinius kompiuterius pasitelkiant mažą skaičių kubitų.

Esminiai kvantinės informacijos skirtumai taip pat atsispindi ir ryšiuose, kuriais siunčiama arba pasitelkiama kvantinė informacija. Kvantinis supynimas leidžia atlkti komunikacijų protokolus, neįmanomus naudojant vien klasikines būsenas. Tarp šių protokolų yra **tankusis kodavimas** (angl. *dense coding*), leidžiantis nusiųsti dvigubai daugiau bitų klasikinės informacijos su kiekvienu kubitu. **Kvantinė teleportacija** (angl. *quantum teleportation*) suteikia būdą persiųsti tolydžiosiose kubito būsenos amplitudėse koduojamą kvantinę informaciją naudojant tik du bitus informacijos. Ne mažiau svarbūs yra kvantinės informacijos principų taikymai kriptografijoje. Kvantiniai protokolai leidžia užtikrinti fizikos dėsniai apsaugotą komunikaciją. Itin didelio saugumo reikalaujančios komunikacijos ateityje veikiausiai kliausius kvantiniai ryšiai arba turės būti atnaujinamos naujais klasikiniais protokolais, kadangi šiandieniniai kriptografijos protokolai tampa įveikiami pažangiais kvantiniais kompiuteriais.

1.2 Kvantinės kompiuterijos pradmenys

Kvantinių sistemų egzistavimas ir pagrindiniai jų elgsenos principai buvo laipsniškai atskleisti 1900–1930 metų laikotarpiu ieškant išeities iš absurdų spėjimų, kuriuos buvo galima prieiti taikant tuometines fizikos teorijas. Viena tokų problemų buvo vadinama **ultravioletinės srities katastrofa** (angl. *ultraviolet catastrophe*), privedanti prie begalinių energijų. Šios anomalijos ištaisymas ir interpretacija fizikiniame modelyje įvedant elektromagnetinio lauko energijos kvantą, dabar vadinamą fotonu, buvo pirmieji žingsniai link kvantinės mechanikos atsiradimo. Kvantinė mechanika suteikia rinkinį taisyklių, vadinamų postulatais, kurios padeda sujungti abstraktųjų matematinį formalizmą su tuo, ką įmanoma objektyviai išmatuoti ir apibūdinti fizikiniame lygmenyje. Postulatai kvantinėje mechanikoje gali būti palyginami su aksiomomis matematinėse logikų sistemose, įvardijantys pradinius pasiūlymus ar principus, kurie savaime nėra įrodomi, tačiau pagal juos grindžiama likusi loginė sistema. Fizinių teorijų, pagrįstų kvantine mechanika, pavyzdžiai yra kvantinė elektrodinamika ir kvantinė chromodinamika, apibūdinančios šviesos (fotonų) sąveiką su elektros krūvų turinčia materija (elektronai, protonai, miuonai) ir branduolyje stipriasių sąveikas tarp kvarkų bei gluonų atitinkamai. Kitaip nei klasikinėje fizikoje, kvantinės sistemos būsena ir su ja susietos savybės, tokios kaip jos padėtis ir judėjimo kryptis, negali būti stebimi nepaveikiant pačios sistemos būsenos. Tai savo ruožtu apibūdina tarsi paslėptą pasaulį, kurio tiesiogiai pasiekti negalime. Kiek žinoma, gamta yra fundamentaliai kvantinė,

nors keistosios jos savybės atskleidžia tik esant specifinėms sąlygoms.

Pirmąsias idėjas panaudoti kvantines sistemos skaičiavimo procesuose galima aptikti jau devintojo dešimtmečio pradžioje. Tokie žymūs fizikai kaip Richardas Feynmanas, Paulas Benioffas, Yuri Maninas, Davidas Deutschas ir kiti atkreipė dėmesį, kad klasikiniai kompiuteriai modeliuoti kvantines sistemos bendoje situacijoje yra neefektyvu ir daugeliu atvejų tiesiog neįmanoma. Kompiuterio atminties ištekliai, reikalingi apibūdinti kvantinės sistemos būsenai, auga eksponentiškai kartu su sistemos dydžiu. Tai brėžia praktinę ribą ties santykinai mažo dydžio sistemomis bei riboto tipo skaičiavimais ir todėl tiesiogiai veikia fundamentinių tyrimų spartą bei technologinę raidą. Kvantinio kompiuterio idėja pastūmėjo to meto technologinių pasiekimų, pirmą kartą demonstruojantys individualių atominio lygmens sistemų kontrolę. Pavyzdžiui, galėjimas judinti ir sugrupuoti pavienius atomus medžiagos paviršiuje skenuojamuoju elektroniniu mikroskopu, jonų pagavimas elektromagnetinėse gaudyklėse, pavienių elektronų perkėlimas įvairiuose elektroniniuose dariniuose.

Šie pasiekimai veikiau suteikė stimulo tolimesniems tyrimams, kadangi mechaniskai kontroliuoti atominio lygmens sistemos nebūtinai reiškia, kad kontroliuojamos ir jų kvantinės būsenos. Informacijos kodavimas ir apdorojimas kvantinėse sistemoje yra pagristas jų būsenų **koherentiniu valdymu** (angl. *coherent control*). Fizikiniai terminai tai reiškia gebėjimą išlaikyti ir keisti santykines fazes tarp kvantinių sistemų. Kad geriau suprastume koherencijos principą, išsiaižduokime dviejų žmonių irkluojaną valtį. Norint ja plaukti irkluojotams tenka sukoordinuoti periodinį irklų pakėlimą iš vandens, nuleidimą ir atsistumimą nuo vandens. Jeigu jie tai idealiai sukoordinuoja, tada galime sakyti, kad jų santykinė fazė lygi nuliui. Koherentinis valdymas reiškia, kad jie gali tiksliai pasirinkti kokią tik pageidauja fazę ir ją išlaikyti. Tačiau jeigu šie irkluojotų atliekami judesiai nepastoviai kinta, tai lems nekoherentinį plaukimą ir tiesiog valties sukinėjimąsi vietoje. Viena iš esminių užduočių išlaikant kvantinių sistemų koherenciją – užtikrinti jų izoliavimą nuo nepageidaujamų sąveikų su aplinkos kvantinėmis sistemomis, kurios nenuuspėjamai keičia jų santykines fazes ir priveda prie **dekoherencijos** (angl. *decoherence*). Dekoherencija yra didžiausias kvantinių kompiuterių priešas, jos efektais pasireiškia atsitikimiai būsenų pokyčiai, sistemos triukšmu, ir dėl to atsirandančiomis skaičiavimų klaidomis. Žinoma, yra ir tokią kvantinių sistemų, kurios labai silpnai sąveikauja su kitomis sistemomis. Tačiau sąveikos yra reikalingos siekiant realizuoti logines operacijas ir skaičiavimų universalumą. Šie vienas kitam prieštaraujantys reikalavimai uždeda apribojimus sistemoms, kurios yra tinkamos realizuoti kubitus.

Kvantinė kompiuterija įgavo pagreitį ir sulaukė platesnės bendruomenės susidomėjimo dešimtojo dešimtmečio pradžioje. Peteris Šoras sukūrė algoritmą, kuriuo parodė, kad kvantiniu kompiuteriu įmanoma pasiekti eksponentinį pagreitinimą siekiant atliglioti itin didelės svarbos užduotis – pirminių skaičių faktorizavimą ir diskrečiojo logaritmo apskaičiavimą. Nėra žinomų efektyvių algoritmų spręsti šias problemas naudojant klasikinius kompiuterius, ir jos traktuojamos kaip neįveikiamos. Kadangi pirminių skaičių faktorizavimui yra pagrasta didžioji dalis globaliai taikomos kriptografijos, tai indikuja akivaizdžią praktinę svarbą ir kartu potencialiai egzistencinę krizę komunikacijų saugumui. Ne mažiau svarbūs buvo pirmieji kvantinių klaidų taisymo principai. Klaidų prevencijos ir taisymo algoritmai yra neatsiejama dalis praktinės svarbos skaičiavimuose.

Kubitais pagristas modelis buvo pastūmėtas itin sėkmingo skaitmeninio kompiuterio modelio, naudojančio dvejetainę sistemą. Vertinant iš praktinės pusės, kontroliuoti dvi būsenas sistemoje yra paprasčiau ir tai išlaiko skaičiavimų universalumą – sudėtingesnis modelis nėra būtinės norint atliglioti visus suformuluojamus skaičiavimus. Kubitais pagrįstuose skaičiavimo modeliuose pasirenkamos sistemos, kurios nusakomos dviam skirtingomis būsenomis arba kuriuose tarp daugelio galimų būsenų yra efektyviai kontroliuojamos tik dvi skaičiavimams atliglioti. Pavyzdžiui,

jonų gaudyklėmis (angl. *ion traps*) pagrįsti kvantiniai procesoriai išnaudoja dalį pagautų jonų elektroninių ir vibracinių energijos lygmenų. Ne vien išimtinai atominio lygmens sistemos demonstruoja kvantinius reiškinius. Makroskopinio skaičiaus elektronų bendrosios krūvio kvantinės būsenos, vadinamos **transmonais** (angl. *transmon*), stabilizuojasi superlaidininkuose esant itin mažoms temperatūroms ir yra naudojamos šios architektūros kvantiniuose kompiuteriuose. Kitos technologijos, galinčios tapti svarbios ateityje, yra grindžiamos fotonika ir atominiais defektais puslaidininkuose. Verta išskirti fotoninius iрenginius, kurie turi natūralų privalumą, nes gali būti paprasčiau integruojami į kvantinius tinklus išnaudojant egzistuojančius šviesolaidinius tinklus ir palydovus. Kvantiniai tinklai suteikia galimybes taikyti **išskaidytą skaičiavimų modelį** (angl. *distributed computing*) tarp ne itin galingų ankstyvosios raidos kvantinių kompiuterių, tačiau juos kartu sujungus galima efektyviai pasiekti didesnę skaičiavimų galią.

1.3 Tiuringo mašina

Siekiant tiksliau palyginti kvantinių ir klasikinių kompiuterių skaičiavimų efektyvumą, verta trumpam gりžti prie klasikinio skaičiavimo modelio. Šiandieninių kompiuterių konceptas, apimantis pagrindinius jų ingredientus bei skaičiavimų universalumą, buvo suformuluotas **Alano Tiuringo** (angl. *Alan Turing*) 1930 metais. Tiuringo darbas šioje srityje buvo paskatintas kitų siekių ir tik vėliau buvo suprasta atradimo svarba universalaus kompiuterio giminui. Tiuringas siekė atsakyti į žymaus matematiko **Davido Hilberto** (angl. *David Hilbert*) iškeltą klausimą ir iššūkį – surasti įrodymą: ar visada egzistuoja algoritmas, kuris naudodamas loginės sistemos aksiomas gali nuspresti, ar pateiktas teiginys yra teisingas arba neteisingas?

Prieš žygiuojant toliau priminsime, kad algoritmas – tai tiksliai nusakyta veiksmų seką ar instrukcijos, skirtos išspręsti užduotį. Su aritmetinio pobūdžio algoritmais susiduriamą jau vidurinėje mokykloje, pavyzdžiu, naudojant **Euklidu** (angl. *Euclid*) 2000 metų senumo algoritmus reikia atlikti dviejų skaičių dalybą su liekanomis arba didžiausio bendrojo daliklio paiešką. Algoritmu paskirtis gali būti įvairiausio pobūdžio, išskaitant duomenų apdorojimą ir automatizuotą sprendimų priėmimą. Algoritmus reikėtų atskirti nuo kompiuterio programų, kadangi algoritmai yra bendresnio pobūdžio instrukcijos ir egzistuoja nepriklausomai nuo programavimo kalbų. Skirtingos programavimo kalbos gali būti pavartojuamos tam pačiam algoritmui įvykdyti.

Hilbertas pats tikėjosi, kad visada galima rasti tokius algoritmus ir todėl iš principio įmanoma išspręsti visas problemas matematikoje. Įrodymas, kad matematikoje egzistuoja neišsprendžiamų problemų, dar tais pačiais metais buvo pateiktas **Kurto Gödelio** (angl. *Kurt Gödel*) jo dviejose žymiose **nepilnumo teoremore** (angl. *Gödel's incompleteness theorems*). Šios teoremos tvirtina, kad aksiomomis pagrįstose loginėse sistemoje visada bus galima rasti teiginį, kuriu neįmanoma nei įrodyti, nei paneigti. To nepavyks apeiti net ir įvedus papildomas aksiomas, t. y. išplečiant loginę sistemą, kadangi tokia sistema negalės užtikrinti savęs pačios nuoseklumo (užtikrinti neprieštaravimo pačiai sau).

Tiuringas formulavo Gödelio rezultatus į algoritmu kalbą, kuriuos gali įvykdyti jo apibūdinta Tiuringo mašina. Tai idealizuotas konceptualus kompiuteris, turintis visus esminius šių dienų kompiuterio komponentus: skaitymo ir rašymo iрenginius, atmintį, procesorių. Tiuringo mašina gali atlikti visus skaičiavimus, kuriuos gali atlikti žmogus, tik efektyviai turi neribotą kiekį popieriaus ir rašymo priemonių. A. Tiuringas kartu su A. Churchu pateikė formalų algoritmo apibūdinimą (angl. *Church-Turing thesis*):

Church-Tiuringo tezė: visos funkcijų klasės, apskaičiuojamos Tiuringo mašina, yra ekvivalentiškos klasei funkcijų, apskaičiuojamų naudojant algoritmą.

Kitaip tariant, Tiuringo mašina gali įvykdyti bet koki suformuluojamą algoritmą. Tai taikydamas Tiuringas parodė, kad egzistuoja kompiuteriu neišsprendžiamą algoritmą, vienas toks yra vadinamas **sustojimo problema** (angl. *halting problem*). Šiuolaikiniai programuojami kompiuteriai yra pagrįsti **fon Noimano architektūra** (angl. *von Neumann architecture*), pristatyta šio žymaus matematikoo (angl. *John von Neumann*) ir realizuota 1940–1950 metais. Fon Noimano architektūra atspindi realizuotą Tiuringo mašiną, naudojančią technologiskai praktiškesnius sprendimus ir aritmetinę loginę sistemą (Būlio algebrą). Skaičiavimo galios ir universalumo atžvilgiu programuojamų šiandieninių kompiuterių modelis yra ekvivalentiškas Tiuringo mašinai su ribota atmintimi. Šie bendri principai leidžia suprasti, kad kvantinis kompiuteris negalės atlikti to, ko negali atlikti universalii Tiuringo mašina, kuria galima modeliuoti ir kvantinius kompiuterius. Kvantių kompiuterių pranašumas atskleidžia kvantinių algoritmų efektyvumą.

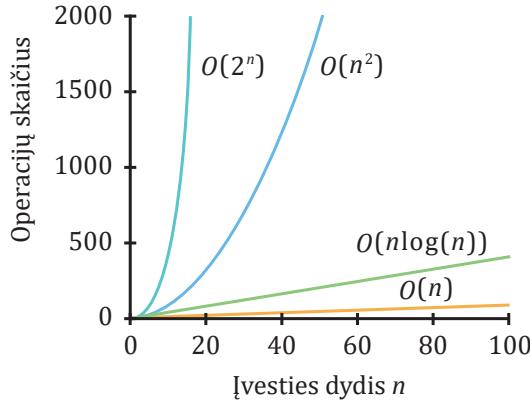
1.4 Skaičiavimų ištekliai

Algoritmų kompleksiškumo analizės sritis parodo problemai spręsti reikalingus skaičiavimo išteklius – bendrai laiko, erdvės (atminties) ir energijos išteklius. Tradiciškai kompiuterių moksle laikas ir erdvė buvo pagrindiniai dominantys ištekliai. Jie leidžia nustatyti, ar tam tikros klasės problemos gali būti išsprendžiamos per atitinkamą laiką esant ribotai kompiuterio atminčiai. Yra nemažai problemų, kurios traktuojamos kaip sunkios ar net neįveikiamos dėl joms išspręsti reikalingų nepasiekiamo dydžio išteklių. Pavyzdžiu, geriausias žinomas klasikinis algoritmas skaičiaus faktorizavimui į dviejų pirminių skaičių sandaugą reikalauja eksponentiškai augančių laiko išteklių su faktorizuojamojo skaičiaus dydžiu. Ši problema neturi efektyvaus skaičiavimų metodo ir nuo salygiškai mažos įvesties dydžio tampa laiko atžvilgiu neįveikiamą.

Operacijų skaičius yra tiesiogiai susietas su laiko ištekliais, reikalingais atlikti algoritmą. Kompiuterio procesoriaus dažnis ir jo architektūros bei algoritmo paralelizavimo galimybės yra pagrindiniai veiksnių, kurie atspindi, kiek operacijų per laiko vienetą galima atlikti, ir todėl nusako bendrą algoritmo vykdymo laiką. Tiksliai nusakyti laiko išteklius yra keblu, kadangi net nežymiai pakeitus skaičiavimo modelį ar kompiuterio architektūrą tai gali juos paveikti. Tačiau dažnai mus domina esminė algoritmo elgsena bei viršutinės operacijų skaičiaus ribos. Asimptotinė algoritmo elgsena didėjant įvesties dydžiu leidžia tai įvertinti. Imkime pavyzdį: palyginkime du skirtinges algoritmus, atliekančius tą pačią užduotį. Reikalingas operacijų skaičius, kuris priklauso nuo įvesties tikslumo ar dydžio, nusakyto n -bitais, pirmajame algoritme yra $f(n) = n^3 + 10n + 100$, o antrajame $g(n) = 1000n + \log(n)$. Šių funkcijų asimptotinė elgsena, kai $n \rightarrow \infty$ (artėja prie labai didelio skaičiaus), yra atitinkamai $O(n^3)$ ir $O(n)$. Simbolis O su skliausteliuose pažymėta funkcinė forma yra vartojamas įvardyti funkcijos viršutinei elgsenos ribai ir blogiausiam scenarijui. Algoritmas $g(n)$ su tiesine $O(n)$ elgsena yra akivaizdžiai efektyvesnis, kai $n \rightarrow \infty$, nepaisant didelio koeficiente šaliai n .

Vertinant laiko išteklius, riba yra standartiškai nubrėžiama tarp algoritmų, kurių ištekliai auga: 1) kaip polinomas ir 2) kuriuose resursai auga greičiau nei bet koks polinomas (sakoma „superpolinomiškai“). Problema yra lengva, traktuojama, arba įmanoma, jeigu egzistuoja algoritmas, kuriam reikalingi polinominiai laiko ištekliai. Palyginimui, problema yra vadinama sunkia, ne-traktuojama, arba neįmanoma, jeigu ištekliai auga superpolinomiškai. Antruoju atveju išprasta sakyti, kad yra reikalaujama eksponentinio dydžio išteklių. Tai mažas piktnaudžiavimas eksponentinės funkcijos apibrėžimu, nes, pavyzdžiu, $n^{\log n}$ auga greičiau nei bet kuris polinomas, bet lėčiau nei tikra eksponentinė funkcija. Aišku, galima įsivaizduoti ir polinomą su itin dideliu koeficientu (pvz., n^{1000}), kuriam gali prireikti daugiau operacijų nei kai kurioms eksponentėms (pvz., $2^{n/1000}$). Tačiau praktikoje algoritmų su dideliais koeficientais retai pasitaiko, ir istoriš-

kai polinominiai algoritmai pasitvirtina esantys efektyvesni. 1.2 pav. pateikiame kreives keleto dažnai aptinkamų kompleksiškumo klasės funkcijų, palyginančių, kaip auga operacijų skaičius didėjant įvesties dydžiui. Atkreipiame dėmesį, kad čia vertikalioji ašis užsibaigia tik ties 2000.



1.2 pav.: Keletas algoritmų sudėtingumo pavyzdžių. Nuo sparčiausiai iki lėčiausiai augančio operacijų skaičiaus: eksponentinė elgsena, polinominė, logtiesinė ir tiesinė

Galime palyginti laiko išteklius keletui kvantinių ir klasikinių algoritmų, skirtų atlikti tą pačią užduotį. Faktorizuojant pirminius skaičius geriausiam žinomam klasikiniam algoritmui reikalinga $O\left(e^{n^{\frac{1}{3}} \log(n)^{\frac{2}{3}}}\right)$ operacijų, o kvantiniam Šoro – $O(n^2 \log(n) \log(\log(n)))$. Šis eksponentinis pagreitinimas yra artimai susietas su Šoro algoritme taikoma **kvantine Furjė transformacija** (angl. *quantum Fourier transform*), kuriai reikalinga $O(n^2)$ operacijų. Klasikinei **diskrečiajai Furjė transformacijai** (angl. *fast Fourier transform*) reikalinga eksponentiškai daugiau operacijų, $O(n2^n)$. Taip pat galima paminėti nestruktūrizuotų duomenų bazių **Groverio kvantinės paieškos algoritmą** (angl. *Grover algorithm*), kuriam reikia $O(\sqrt{n})$ operacijų ir kuris suteikia kvadratinį pagreitinimą prieš klasikinę paiešką $O(n)$.

Algoritmui reikalingi erdvės ištekliai yra ne kas kita, kaip atminties ištekliai. Atminties išteklių esminis augimas didėjant įvesties dydžiui n yra taip pat nusakomas O simboliu. Laiko ir atminties ištekliai gali augti tokiu pačiu tempu, jeigu kiekviename algoritmo žingsnyje yra reikalaujama nauja atminties celė. Atminties ištekliai skiriasi nuo laiko, nes dažnai yra įmanoma atlaisvinoti nenaudojamos atminties išteklius. Néra įrodyta, kad kvantiniai algoritmai turi pranašumą atminties išteklių srityje.

Nors tai neatspindi algoritmų atminties sudėtingumo, tačiau šioje vietoje verta pabandyti lyginti klasikinių bei kvantinių kompiuterių atminties talpą. Informacija yra įrašoma kvantinio kompiuterio kubitų registre. Registras, sudarytas iš n kubitų, gali būti 2^n skirtingose būsenose vien metu, o kiekvieną būseną nusakančios amplitudės yra kompleksiniai skaičiai. Galime pažiūrėti, kiek informacijos baitais atitinką, pavyzdžiu, 50 kubitų registras klasikiniame kompiuteryje. Pirmiausiai atkreipiame dėmesį, kad nusakyti tolydžiai kintančią kubito amplitudę iš principo reikėtų begalinio tikslumo ir todėl formaliai begalinės informacijos kiekio. Tačiau praktiškai kvantiniame kompiuteryje tikslumas bus taip pat ribotas (dėl triukšmo, loginių vartų netikslumų), tad skaičiaus tikslumui apsiribosime **32 bitais** (angl. *single-precision floating-point number*). Nusakyti bendrai 50 kubitų sistemos būsenai reikia 2^{50} kompleksinių amplitudžių, kiekvienai iš jų reikia 2 realiųjų skaičių, kurių kiekvienas užima 32 bitus. Tad iš viso reikalaujama 8 petabaitų atminties. Tai yra artima darbinei atminčiai, kuria vienas iš pajėgiausių superkompiuterių (*IBM*

Summit) šios knygos rašymo metu geba pasiūlyti. Dėl eksponentinio atminties reikalavimų augimo kvantinio kompiuterio veikimo modeliavimas pradedant nuo maždaug 50–60 kubitų ribos tampa labai greitai nejveikiamas net ir sudėjus visas planetos atminties saugyklas. Šiuo požiūriu kvantinis kompiuteris gali pasiūlyti didesnę atminties talpą, negu kada nors bus įmanoma pasiekti klasikiniai įrenginiai. Čia svarbus priminimas, kad tai yra kvantinė informacija, saugoma būsenų superpozicijose. Šią informaciją galima efektyviai apdoroti, tačiau, kitaip nei klasikinės, jos visos tiesiogiai nuskaityti neįmanoma. Galime sakyti, kad kvantinė informacija yra paslėpta arba potenciali. Minėtame pavyzdyme pamatavę 50 kubitų gausime 50 bitų, o ne 8 petabajtų dvejetainių skaičių seką.

Energija yra kitas svarbus skaičiavimų išteklius ir gali būti susietas su laiko bei erdvės ištekliais, reikalingais atlikti algoritmą. Su vadinamaja **galios siena** (angl. *power wall*) klasikiniai kompiuteriai susidūrė dar 2000 metų pradžioje, kai buvo pasiekti maždaug 3,5 GHz procesorių dažniai. Kiekviename klasikinio procesoriaus cikle, kai tranzistorių loginės būsenos pasikeičia ($0 \rightarrow 1$, $1 \rightarrow 0$), yra išeikvojama energija ir išsklaidoma šilumos forma. Standartiniai **CMOS** (angl. *complementary metal-oxide-semiconductor*) tranzistorių technologija pagrįsti procesoriai itin pažengė mažinant energijos sąnaudas, daugiausia mažinant tranzistorių įtampą loginėms vertėms keisti. Norint toliau didinti klasikinių kompiuterių galią svarbu atrasti būdą, kaip tokiu pačiu greičiu ar sparčiau mažinti jų energijos sąnaudas. Nors šiuo metu klasikiniai procesoriai eikvoja energiją dėl neoptimalių technologijų, tačiau egzistuoja ir fundamentali riba, įvardijanti minimalią energijos kainą ištrinant informaciją. **Landauerio principas** (angl. *Landauer principle*) teigia, kad informacija ir energija yra artimai susietos:

Landauerio principas: kiekvieną kartą, kai ištrinamas vienas bitas informacijos, į aplinką yra išsklaidoma bent $k_B T \log(2)$ energijos.

Čia k_B yra Boltmano konstanta, T – kompiuterio aplinkos temperatūra. Tai suteikia užuominą, kad informacija néra vien abstraktus koncepcas, tačiau gali būti prilyginta kitiems fiziniams dydžiams. Atliekant skaičiavimus klasikiniame procesoriuje dalis informacijos yra visada ištrinama dėl juose naudojamų negrūtamųjų loginių vartų. Pavyzdžiui, plačiai naudojamuose loginiuose vartuose **NAND** (angl. *negated AND*) yra įvestis dviem bitams ir išvestis vienam bitui: $(0,0) \rightarrow 1$, $(0,1) \rightarrow 1$, $(1,0) \rightarrow 1$, $(1,1) \rightarrow 0$. Čia skliausteliuose nurodyta įvesties bitų vertė ir rodyklyte – išvesties bitas. Jeigu išvesties bito vertė yra 1, tada galimos trys skirtinės įvesties bitų kombinacijos. Todėl neįmanoma unikalai pasakyti, kokie buvo įvesties bitai iš trijų kombinacijų, ir dėl šios priežasties vienas bitas informacijos yra negrūtamai prarandas.

Landauerio ribą galima apeiti naudojant grižtamuosius loginius vartus ir neištrinant informacijos skaičiavimo procese. Tam atliekamas išprastas skaičiavimas, o pabaigoje registras yra grąžinamas į pradinę loginę būseną termodinamiškai grižtamuoju būdu. Klasikiniuose skaičiavimuose galima pasitelkti vien tik grižtamuosius loginius vartus, tačiau procesorių architektūra sudėtingėja, tam reikia papildomų bitų ir loginių vartų. O štai kvantiniame kompiuteryje skaičiavimai yra pagrįsti grižtamaisiais loginiais vartais, mat kvantinės fizikos dėsniai yra fundamentaliai grižtamieji. Kitaip tariant, kvantiniame kompiuteryje galima atsukti laiką atgal. Tam tereikia pritaikyti atvirkštinius loginius vartus atvirkštine tvarka. Tiesa, informacija kvantiniame procesoriuje yra trinama kubitų būsenų matavimo ir dekoherencijos procesų metu. Tačiau išsklaidoma Landauerio energija auga tiesiškai su kubitu skaičiumi n , o ne eksponentiškai su galimų būsenų skaičiumi 2^n . Tai bendrai yra gera žinia, kadangi kvantiniai kompiuteriai turi potencialą dominuoti ateityje atliekant intensyviausius skaičiavimus.

1.5 Kvantiniai bitai

Klasikinėje kompiuterijoje informacija yra koduojama naudojant „abécelę”, turinčią dvi skirtinges „raides”, standartiskai įvardijamas 0 ir 1, kurios nusako elementariausią loginę sistemos būseną. Mažiausias klasikinės informacijos vienetas **bitas** (angl. *binary digit*) atspindi šią dvejetainę būseną. Kadangi dvejetainių skaičių seka yra diskrečioji, joje koduojama informacija vadinama skaitmenine. Tai galima palyginti su informacija, laikoma analoginiu pavidalu, pavyzdžiui, vinilinėse plokštelėse, kuriose garso įrašas kinta tolydžiai su paviršiaus įspaudimo gyliu ir deformacijomis.

Bitas yra matematinis objektas, turintis dvejetainę savybę, ir nepriklauso nuo jį realizuojančių fizinės sistemų. Tai itin pravartu, kadangi naudojant vien abstrakčius matematinius objektus ir taikant su jais susietą operacijų logiką galima formuluoja skaičiavimo modelius ir testuoti informacines teorijas nesirūpinant dėl fizinio lygmens. Pagrindinis reikalavimas fiziniame lygmenyje yra tas, kad bitą realizuojantis darinys turėtų dvi būsenas, kurias būtų lengva atskirti ir keisti panorėjus. Fizinė sistema, realizuojanti skaitmeninę informaciją, neprivalo būti savaime diskreti. Tranzistoriai pagrįstose procesoriuose bitų vertės yra koduojamos elektros įtampa, kuri nusako tolydžiai kintantį fizinį dydį. Šis iš esmės analoginis signalas yra skaitmenizuojamas diskrečiosiomis elektroninių grandinių įtampomis $V = 0$ ir $V = V_0$, koduojančiomis bito būsenas 0 ir 1.

Kvantinis bitas (kubitas) taip pat gali būti realizuojamas skirtingomis fizinėmis sistemomis. Kubitą apibūdiname nusakydami jo būseną, tačiau tai yra iš esmės kitoks matematinis objektas nei klasikinis bitas. Kubito būsena yra įvardijama vektoriumi 2 dimensijų kompleksinėje vektorių erdvėje ir bendrai užrašoma taip:

$$|\psi\rangle = a|0\rangle + b|1\rangle. \quad (1.1)$$

Terminai „kvantinė būsena“ ir „vektorius“ dažnai vartojami pakaitomis ir reiškia tą patį, nes kvantinė būsena yra visiškai nusakoma vektoriumi. Skliausteliai $| \dots \rangle$ indikuoja, kad šis objektas yra vektorius. Kubito būseną žymime savo nuožiūra pasirinktu simboliu, šiuo atveju graikiška raide ψ (tariama *psi*). Vektorius $|\psi\rangle$ yra kitų dvių vektorių, $|0\rangle$ ir $|1\rangle$, vadinamų **skaičiuojamais baziniais vektoriais** (angl. *computational basis vectors*), kombinacija. Baziniai vektoriai atlieka panašią funkciją kaip ilguma ir platura geografinėje sistemoje nusakyti objekto padėciai Žemės paviršiuje. Santykinai su baziniais vektoriais yra įvardijama $|\psi\rangle$ orientacija vektorių erdvėje. Koeficientai a ir b , formaliai vadinami amplitudėmis, yra kompleksiniai skaičiai (turi realiąją ir menamąją dalis). Klasikinio bito būsenų 0 ir 1 kvantinis atitinkmuo būtų $|0\rangle$ ir $|1\rangle$ nustačius $b = 0$ ir $a = 0$ amplitudes, atitinkamai. Bendroje situacijoje, kubitas $|\psi\rangle$ yra šių būsenų superpozicijoje. Amplitudės gali kisti tolydžiai, tačiau yra tarpusavyje susietos taip, kad jų (kompleksinių) kvadratų suma susidėtų į vieną (sakoma, kad yra normuotos):

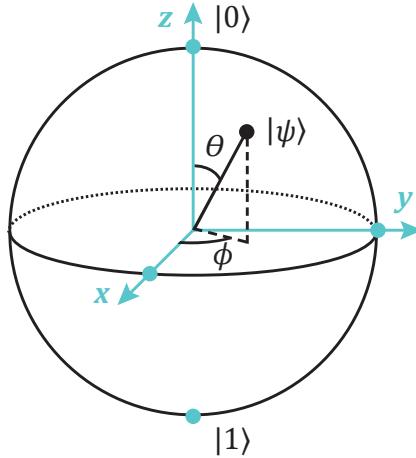
$$|a|^2 + |b|^2 = 1. \quad (1.2)$$

Šis reikalavimas reiškia, kad kubitų būsenos $|\psi\rangle$ yra visada nusakytos vienetinio ilgio vektoriumi. Vieno kubito būsenas galime iliustruoti grafiškai pasitelkdami vadinamąjį **Blocho sferą** (angl. *Bloch sphere*). Vieną kubitą perteikiame perrašę kompleksines amplitudes a ir b taip, kad jos būtų parametrizuotos kampais θ ir φ , nurodytais sferoje (žr. 1.3 pav.):

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi}\sin\left(\frac{\theta}{2}\right)|1\rangle. \quad (1.3)$$

Kubito būsena yra nusakoma Blocho vektoriumi, pavaizduotu nuo centro iki Blocho sferos paviršiaus. Matome, kad skirtinges kubito būsenos nusako skirtinges vektoriaus orientaciją išlaikant

vienetinį jo ilgi. Pavyzdžiu, būsenos $|0\rangle$ ir $|1\rangle$ yra Blocho vektoriaus pozicijos išilgai z ašies ir nukreiptos į viršų arba į apačią ($\theta = 0^\circ, 180^\circ$), atitinkamai. Sferos pusiaujas ($\theta = 90^\circ$) nusako visas $|0\rangle$ ir $|1\rangle$ būsenų lygias superpozicijas, kurios skiriasi viena nuo kitos tik santykiniu $e^{i\varphi}$ fazės nariu, parametrizuojamu azimutiniu kampu φ . Visi įmanomi vektoriai nuo centro iki bet kurio taško sferos paviršiuje įvardija visas skirtinges kubito būsenas, kurių iš esmės yra begalybė.



1.3 pav.: Blocho sfera

Siekdami sužinoti kubito būseną mes turime ji **pamatuoti** (angl. *state measurement*). Būsenų amplitudžių kvadratas nusako galimus matavimo rezultatus ir jų tikimybes. Matavimo procesą šiame skyriuje simboliškai rašysime M raide, prišlieta prie kubito būsenos $|\psi\rangle = a|0\rangle + b|1\rangle$. Atlikus kubito matavimą galima rasti $|0\rangle$ būseną su tikimybe $p = |a|^2$ arba $|1\rangle$ su tikimybe $p = |b|^2$:

$$M|\psi\rangle \rightarrow |0\rangle, \quad p = |a|^2, \quad (1.4)$$

$$M|\psi\rangle \rightarrow |1\rangle, \quad p = |b|^2. \quad (1.5)$$

Matavimas priverčia kubitą „pasirinkti“ vieną iš dviejų galimų būsenų, $|0\rangle$ arba $|1\rangle$, o superpozicijos būsena yra negrįjtamai prarandama. Kartais dar sakoma, kad superpozicijos būsena suvera (angl. *state collapse*). Amplitudžių kvadratų vienetinė suma atspindi tai, kad tikimybė p rasti $|0\rangle$ arba $|1\rangle$ būseną turi susidėti į 1. Čia naudojame trupmenas vietoj procentų nusakyti tikimybėms, pavyzdžiu, $p = 0.75$ nusako 75% tikimybę.

Siekdami pabrėžti statistinį matavimų aspektą, imkime pavyzdį $|\psi\rangle$ būsenos, kurioje amplitudės yra $a = i\sqrt{0.7}$ ir $b = \sqrt{0.3}$. Paruošus daugybę kubitų identiškoje $|\psi\rangle$ būsenoje ir visiems atlikus būsenų matavimus, 0.7 visų kartų bus rasta būsena $|0\rangle$ bei 0.3 visų kartų $|1\rangle$. Panašiai kaip metant monetą – nors ir žinome, kad tikimybė gauti vieną iš dviejų rezultatų turėtų būti $p = 0.5$, tačiau norint šį statistinį faktą realizuoti reikia kartoti monetos metimą begalę kartų. Trupmenė sekoje ši statistika gali drastiškai skirtis nuo $p = 0.5$, kadangi atskiri monetos metimo rezultatai nėra priklausomi vienas nuo kito.

Net ir paruošus begalę kubitų identiškose būsenose ir visus pamatavus, amplitudžių a ir b įvardyti taip nepavyks. Žinosime tik jų (kompleksinius) kvadratus $|a|^2$ ir $|b|^2$, nusakančius tikimybes rasti $|0\rangle$ ir $|1\rangle$ būsenas. Mat kubito amplitudės a ir b yra kompleksiniai skaičiai, kuriuos bendrai

užrašome $z = |z|e^{i\varphi}$. Čia $|z|$ yra kompleksinio skaičiaus z modulis, o narys $e^{i\varphi}$ nusako santykinę fazę. Santykinė fazė tarp kubitų būsenų $|0\rangle$ ir $|1\rangle$ yra svarbi ir turi fiziškai stebimą efektą, tačiau informacija apie ją kompleksiniame kvadrate yra prarandama.

Tad galime klausti, jeigu neįmanoma atskleisti bendros kubito superpozicijos būsenos, kaip žinoti, kad visas šis reikalus nėra susigalvota matematinė iliuzija? Nors matavimas pakeičia kubito būseną atsitiktinai, tačiau iki matavimo būsenas kvantiniame kompiuterioje galime keisti deterministiškai. Būsenų transformacijose atsitiktinumą nėra, todėl pradėjė nuo, pavyzdžiu, $|1\rangle$ būsenos, galime patikrinti, ar po sekos transformacijų matavimų tikimybės atitinka spėjimus, nusakomus $|a|^2$ ir $|b|^2$. **Kvantinių būsenų tomografija** (angl. *quantum state tomography*) yra vienas metodas, leidžiantis unikaliai nusakyti būseną. Šis metodas reikalauja paruošti identišką kvantinę būseną daugybę kartų ir atliliki sumanai parinktus skirtingo tipo būsenų matavimus. Kvantinių būsenų tomografija nėra įprastai taikoma kvantinėje kompiuterijoje, tačiau leidžia atliliki diagnostines užduotis. Vis dėlto, turint tik vieną kubitą nežinomoje superpozicijos būsenoje, atskleisti amplitudžių a ir b , kitaip tariant, pasakyti, kokia yra būsena $|\psi\rangle$ prieš matavimą – iš esmės neįmanoma.

Kiek žinoma, kvantiniuose matavimuose stebimas atsitiktinumas yra vienintelis **tikras atsitiktinis procesas gamtoje** (angl. *true quantum randomness*). Vis dar nėra suprastos to ištakos – kvantinėje fizikoje šis faktas tiesiog pateikiamas kaip postulatas. Vienas iš bandymų tai paaškinti traktuojas viskā, iškaitant matavimo įrenginį ir patį eksperimentuotoją, kvantiniu lygmeniu. Tada visas procesas turėtų būti unitarinis ir be atsitiktinumų. Kažkas panašaus į matavimo procesą galėtų atsirasti dėl mikroskopinio dydžio daiktų sparčios dekoherencijos. Nepaisant atlirkštinti darbų šioje srityje, nėra sutarimo, ar taip įmanoma įrodyti matuojant stebimą atsitiktinumą. Tad nors kvantinė mechanika ir turi interpretacinių klaustukų, tačiau gali tiksliau nei bet kuri kita teorija apibūdinti stebimą pasaulį. Tuo mes ir vadovaujamės kvantinės kompiuterijos taikymuose nesigilindami į filosofinius aspektus. Šis pragmatinis principas yra Davido N. Mermino vadinamas „**stylėk ir skaičiuok**“ (angl. *shut up and calculate*). Gebėjimas beprecedentiškai tiksliai kontroliuoti kvantines sistemas ir besivystantiesi informatikos mokslas galbūt ateityje leis geriau suprasti keistąsias šios teorijos savybes.

Dabar aptarsime, kaip perteikti būsenas, sudarytas iš daugiau nei vieno kubito. Tam pirmiausiai imkime du klasikinius bitus. Du bitai vienu metu gali būti vienoje iš keturių skirtingų būsenų $\{00, 01, 10, 11\}$. Vadovaujantis kvantinės superpozicijos principu, bendra 2 kubitų būsena $|\psi\rangle$ yra visų šių klasikinių bitų būsenų superpozicija:

$$|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle. \quad (1.6)$$

Vektorius $|\psi\rangle$ yra nusakomas įvardijant keturias amplitudes, prišlietas prie $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ 2 kubitų skaičiuojamųjų bazinių vektorių. Pirmo kubito k_1 būsena yra įvardijama kairiuoju skaičiumi, o antro k_2 – dešiniuoju, $|k_1 k_2\rangle$. Formaliai $|k_1 k_2\rangle$ nusako vektorių $|k_1\rangle$ ir $|k_2\rangle$ tensorinę sandaugą, žymimą $|k_1 k_2\rangle \equiv |k_1\rangle \otimes |k_2\rangle$, ir yra apibrėžtas 4 dimensijų kompleksinėje vektorių erdvėje. Keičiant amplitudes yra keičiama $|\psi\rangle$ vektoriaus orientacija šioje vektorių erdvėje. Amplitudės tarpusavyje vėlgi susietos taip, kad jų (kompleksinių) kvadratų suma susidėtų į vienetą:

$$|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1. \quad (1.7)$$

Matavimo $M|\psi\rangle$ pabaigoje gauname du bitus informacijos, susietus su viena iš keturių galimų

būsenų. Tikimybė rasti bet kurią iš jų vėlgi nusakoma tos būsenos amplitudės kvadratu:

$$M|\psi\rangle \rightarrow |00\rangle, \quad p = |a|^2, \quad (1.8)$$

$$M|\psi\rangle \rightarrow |01\rangle, \quad p = |b|^2, \quad (1.9)$$

$$M|\psi\rangle \rightarrow |10\rangle, \quad p = |c|^2, \quad (1.10)$$

$$M|\psi\rangle \rightarrow |11\rangle, \quad p = |d|^2. \quad (1.11)$$

Galiausiai kvantinio registro, sudaryto iš n kubitų, būsena yra įvardijama visų 2^n bazinių vektorių kombinacija ir nusakoma 2^n dimensijų vektorių erdvėje. Ši superpozicijos būsena glaustai užrašoma pasitelkiant sumos simbolį:

$$|\psi\rangle = \sum_{x=0}^{2^n-1} c_x |x\rangle. \quad (1.12)$$

Čia c_x yra $|x\rangle$ bazinio vektoriaus amplitudė, o amplitudžių kvadratų suma tenkina $\sum_x |c_x|^2 = 1$. Kiekvienas bazinis vektorius $|x\rangle$ yra formaliai pavienių n kubitų tensorinė sandauga $|x\rangle \equiv |k_1\rangle \otimes |k_2\rangle \otimes \dots \otimes |k_n\rangle$, $k_i \in \{0, 1\}$. Atlikę visų n kubitų būsenų matavimus rasime n bitų seką, nusakomą vienu iš galimų superpozicijoje esančių būsenų $|x\rangle$ su tikimybe $|c_x|^2$.

1.6 Kvantinės informacijos apdorojimas

Kompiuterių mokslas teigia, kad pasitelkiant vos keletą skirtinį elementarių loginių elementų ir sujungiant juos kartu galima realizuoti visus įmanomus skaičiavimus. Klasikiniame procesoriuje dvejetainė informacija yra apdorojama pasitelkiant loginius vartus. Skirtingi loginiai vartai turi tam tikrą skaičių įvesties ir išvesties jungčių, į kurias galime išsibaigduoti ateinant laidus. Laidais tarp loginių vartų keliaujantis elektros signalas dvielę skirtinį verčią įtampos pavidalu nusako keliaujančius bitus. Loginiai vartai su atkeliavusiais bitais gali atlikti skirtinias elementarijas bitų transformacijas. Pavyzdžiui, loginiai vartai, kurie turi po vieną įvesties ir išvesties bitą, gali būti tik dvielę rūšių. Pirma iš jų palieka išvesties bito vertę tokią pačią, kaip ir įvesties $0 \rightarrow 0$, $1 \rightarrow 1$ ir nusako trivialų identitetą, dar vadinama **tapatybės transformacija** (angl. *identity*). Antrojo tipo vartai išvestyje apverčia įvesties bito vertę $0 \rightarrow 1$, $1 \rightarrow 0$ ir yra žinomi kaip *NOT*. Kiti elementarūs loginiai vartai *AND*, *OR*, *NAND* turi įvesti dviem bitams ir vieną išvesties bitą. Norint atlikti visus įmanomus klasikinius skaičiavimus, tai yra pasiekti skaičiavimų universalumą, pakanka vien *AND* ir *NOT*, arba vien tik *NAND* loginį vartą, iš kurių kombinacijų konstruojamos loginės grandinės. Dvejetainis elektros signalas, praėjęs tokią programuojamą loginę grandinę, yra pakeičiamas kita seka taip atliekant dvejetainių skaičių aritmetiką ir logines operacijas. Likusią kompiuterio įrangą galima vadinti išoriniais elementais, kurie padeda pateikti informaciją į procesoriaus loginių operacijų elementą ir iš jo paimti bei perteikti apdorotą informaciją.

Norint suprasti pagrindinius kvantinės informacijos apdorojimo principus galima taip pat apsiriboti kvantinio procesoriaus veikimo funkcijomis. Kaip ir klasikinis procesorius, kvantinis procesorius vykdo pateiktas logines grandines, sudarytas iš loginių vartų sekų. Vieno kubito atveju kvantinių loginių vartų efektą galima iliustruoti kaip Blocho vektoriaus krypties keitimą Blocho sferoje. Kitaip nei klasikiniame procesoriuje, kuriame vieno bito galimos transformacijos yra tik tapatybės transformacija ir *NOT* loginiai vartai, skirtinį 1 kubito transformaciją iš principo yra begalybė. Tai yra dėl to, kad amplitudės, nusakančios kubito būseną, gali kisti tolydžiai, tad Blocho vektoriaus pasukimas apie tam tikrą Blocho sferos ašį skirtiniais kampais formaliai nusako skirtinges loginius vartus. Kaip ir klasikiniame, kvantiniame procesoriuje galima rasti universalų

rinkinį loginių vartų, kurių kombinacijos leidžia atlikti visas įmanomas kvantinio registro būsenų transformacijas ir todėl galiausiai – visus suformuluojamus algoritmus.

Kubitų loginiai vartai yra apibūdinami matematiniais objektais, kurie vadinami tiesiniais operatoriais. Šie operatoriai turi svarbią savybę – veikdami bendrą kvantinę būseną, nusakyta vektoriumi $|\psi\rangle$, jie keičia tik šio vektoriaus orientaciją vektorių erdvėje, tačiau nekeičia vektoriaus ilgio. Ši savybė vadinama **unitarumu** (angl. *unitarity*) ir užtikrina, kad loginiai vartai nepažeidžia tikimybinio būsenų matavimo principo – visų galimų matavimo rezultatų tikimybės p susidės iš 1. Norint sužinoti kvantinių loginių vartų efektą bendrai superpozicijos būsenai pakanka žinoti, kaip atitinkamas operatorius U veikia kiekvieną skaičiuojamąjį vektorių atskirai:

$$U|\psi\rangle = U(a|0\rangle + b|1\rangle) = a(U|0\rangle) + b(U|1\rangle). \quad (1.13)$$

Skaičius (amplitudes) a ir b iškéléme už skliaustelių norėdami aiškiau parodyti, kad operatoriai veikia vektorius ir kiekvieną jų superpozicijoje vienu metu. Pavyzdžiui, kvantiniai loginiai vartai *NOT*, žymimi simboliu X ir veikiantys superpozicijos būseną $|\psi\rangle$, turi šį efektą:

$$X|\psi\rangle = a(X|0\rangle) + b(X|1\rangle) = a|1\rangle + b|0\rangle. \quad (1.14)$$

Jų efektas baziniams vektoriams yra $X|0\rangle = |1\rangle$ ir $X|1\rangle = |0\rangle$. Matome elementariausią kvantinio paralelizmo pavyzdį, kai pritaikius vieną loginę operaciją ji yra ivertinama visuose superpozicijos nariuose vienu metu. Klasikiniame kompiuteryje, norint apskaičiuoti funkcijos $f(x)$ vertes su reikšmėmis x , reikia $f(x)$ išvertinti su kiekviena x reikšme atskirai. Tad jeigu funkcijos reikšmes x koduojame skirtingais kubitų skaičiuojamaisiais vektoriais, superpozicijos principas ir tiesinis operatorių veikimas leidžia išvertinti vienu funkcijos f iškvietimu visas reikšmes tuo pačiu metu.

Kvantinės būsenos ir jas transformuojantys operatoriai yra realizuojami tiesinėje algebroje stulpeliniais vektoriais ir matricomis, atitinkamai. Vieno kubito būsena gali būti išreikšta bet kuriuo iš šių būdų:

$$|\psi\rangle = a|0\rangle + b|1\rangle = a \begin{bmatrix} 1 \\ 0 \end{bmatrix} + b \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}. \quad (1.15)$$

Visi operatoriai, keičiantys 1 kubito būseną, yra realizuojami (2×2) dydžio matricomis. Pavyzdžiui, kvantinių *NOT* bei vadinamųjų **Hadamardo loginių vartų** (angl. *Hadamard*, trumpinys H) veiksmai kubito būsenai $|1\rangle$ atrodo taip:

$$X|1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle, \quad (1.16)$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \quad (1.17)$$

Iš pateiktų pavyzdžių matyti, kad H ir X vartai yra sau atvirkštiniai. Atlikus du veiksmus vieną po kito, pavyzdžiui, HH , kubito būsena nepakinta. Iš tiesų visi loginiai vartai kvantinėje kompiuterijoje turi sau atvirkštinius loginius vartus arba yra patys sau atvirkštiniai. Pasukus būseną nusakantį vektorių tam tikra kryptimi, atvirkštiniai loginiai vartai atsuka šį vektorių atgal atbuline kryptimi išlaikydamai jo ilgį.

Tiesinėje algebroje n kubitų būseną nusakantis vektorius yra stulpelis su 2^n elementų, o loginiai vartai yra $(2^n \times 2^n)$ dydžio unitarinės matricos. Jos nusako n kubitu būsenos transformaciją – vektoriaus posūkį 2^n dimensijų vektorių erdvėje. Tačiau norint atlikti visas įmanomas n kubitu būsenų transformacijas nepakanka, kad kiekvienam kubitui atskirai veiktu 1 kubito loginiai vartai. Tai pirmiausiai matome todėl, kad didžioji dalis 2^n skirtų superpozicijos būsenų yra

supintos. Supynimas yra nelokalus efektas, jo neįmanoma nei įvesti, nei panaikinti atliekant kvantines transformacijas su pavieniais kubitais. Nelokalios transformacijos yra reikalingos supintosioms būsenoms ir taip atlikti bendresnes n kubitų būsenų transformacijas. Kaip tik galėjimas panaudoti visą eksponentiškai didelę būsenų erdvę suteikia kvantiniams kompiuteriams pranašumą prieš klasikinius įrenginius.

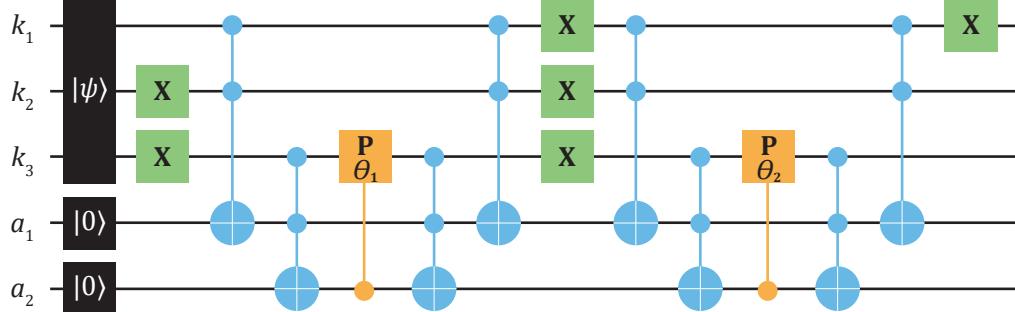
Salyginių 2 kubitų loginiai vartai $CNOT$ (angl. *controlled NOT*, trumpinys cX) kartu su bendro tipo 1 kubito vartais leidžia realizuoti vieną galimą universalų rinkinį. Vartai cX apverčia antrojo kubito būseną, jeigu pirmojo kubito būsena yra $|1\rangle$, ir palieka ją nepakeistą, jeigu pirmasis yra $|0\rangle$. Fiziškai tai apibūdina sąveikas tarp dviejų kubitų. Vienas pavyzdys cX , veikiančių 2-kubitų būseną $|11\rangle$ tiesinėje algebroje:

$$cX|11\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = |10\rangle. \quad (1.18)$$

Tačiau kvantinėje kompiuterijoje galime pritaikyti cX , kai pirmas kubitas yra $|0\rangle$ ir $|1\rangle$ būsenų superpozicijoje. Tai nusako klasikinėje kompiuterijoje analogo neturinčią deterministinę operaciją – kvantinį supynimą:

$$cX(a|0\rangle + b|1\rangle) \otimes |0\rangle = a|00\rangle + b|11\rangle. \quad (1.19)$$

Kubito būsenų transformacijas galima itin paprastai ir intuityviai iliustruoti. Kiekvienam kubitiui yra priskiriama atskira grandis – horizontali linija, o loginių vartų sekos yra išrikiuojamos iš kairės į dešinę ir pavaizduoja laikę atliekamas būsenų transformacijas. Skaičiavimo pradžioje kubitai yra standartiniškai inicijuojami (angl. *reset*) į pradines $|0\rangle$ būsenas. Tokios diagramos yra vadinamos **kvantinėmis grandinėmis** (angl. *quantum circuits*).



1.4 pav.: Kvantinės grandinės pavyzdys

1.4 pav. kvantinėje grandinėje matome 1 kubito vadinamuosius Pauli- X vartus (žali); loginiai vartai, veikiantys du ar daugiau kubitų, turi vertikalių kubito grandines jungiančią liniją. Tarp jų yra 2 kubito fazės vartai (oranžiniai) ir 3 kubito vadinamieji Tofoli loginiai vartai (mėlyni). Parodyta kvantinė grandinė pritaiko jau paruoštai trijų kubito (k_1, k_2, k_3) būsenai $|\psi\rangle$ santykinę fazę. Kubitai a_1 ir a_2 pradinėse $|0\rangle$ būsenose čia atlieka juodraščio funkciją, į kurį užrašomi tarpiniai skaičiavimo rezultatai. Nuo jų būsenų priklauso, kuriam iš skaičiuojamųjų vektorių, sudarančių $|\psi\rangle$, bus pritaikytos fazės θ_1 ir θ_2 .

Kvantiniai loginiai vartai, veikiantys atskirus kubitus, gali būti atliekami vienu metu (paraleliai), jeigu kvantinio procesoriaus architektūra tai leidžia. Kvantinės **grandinės gylis** d (angl. *circuit*

depth) nusako ilgiausią loginių vartų seką grandinėje nuo pradžios iki algoritmo pabaigos, kuri nebegali būti padaroma trumpesnė didesniu paralelizavimu. Parodytos grandinės gylis yra $d = 13$.

1.7 Skaičiavimo procesas

Apibendrinus, norint atlikti skaičiavimą kvantiniame procesoriuje reikalingi trys pagrindiniai elementai:

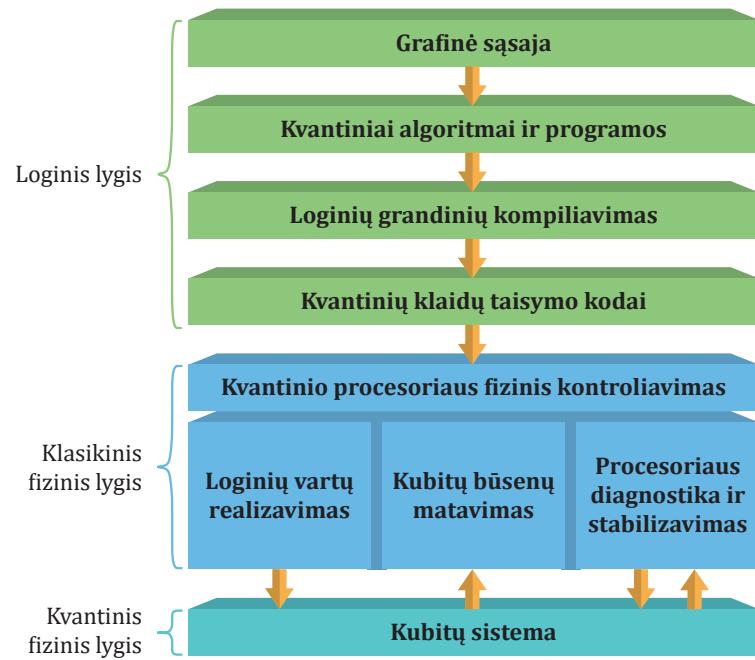
1. *kvantinių bitų registras. Registras – tai kubitų rinkinys, kuriame koduojama ir apdorojama informacija. Registrą galima inicializuoti į mums žinomą pradine būseną, standartiskai $|0\rangle$ visiems kubitams;*
2. *universalių kvantinių loginių vartų rinkinys. Loginiai vartai atlieka registro būsenų keitimo operacijas, leidžiančias koduoti ir apdoroti informaciją;*
3. *kubitų būsenų matavimai. Norint nuskaityti registre apdorotą informaciją, kubitų būsenos turi būti išmatuojamos.*

Šie trys elementai yra traktuojami, kaip esantys loginiame lygmenyje. Kubitų skaičius bei skaičiavimo klaidų efektyvus nebuvinamas loginiame lygmenyje nebūtinai atspindi fizinio procesoriaus kubitų skaičių bei veikimo netikslumus. Fiziniame lygmenyje skaičiavimo klaidos gali atsirasti dėl kubitų būsenų dekoherencijos, netikslių loginių vartų atlikimo bei matavimo procedūrų klaidų. Loginis lygmuo atspindi fizinio lygmens procesoriaus funkcionavimą su jau atliktais **klaidų taisymo kodais** (angl. *error correction codes*). Klaidoms atspari skaičiavimo teorija teigia, kad jeigu klaidų atsiradimo dažnis yra mažesnis nei tam tikra riba, tada įmanoma transformuoti kvantines grandines į klaidoms atsparias, kuriose klaidos efektyviai aptinkamos ir ištaisomos. Tai leistų atlikti neribotos trukmės kvantinius skaičiavimus.

Kvantinių klaidų taisyme, panašiai kaip ir klasikinėje skaitmeninėje kompiuterijoje bei ryšiuose, pasitelkiama **perteiklinė informacija** (angl. *redundant information*). Vietoj vieno fizinio kubito, informacija yra koduojama kelete fizinių kubitų, kurie formuoja vadinamąjį loginį kubitą (angl. *logical qubit*). Loginiai vartai yra pakeičiami loginių vartų seką, kurią pritaikius fiziniams kubitams efektyviai atliekama norima loginė operacija su loginiais kubitais. Esminiai skirtumai nuo klasikinių klaidų taisymo atsiranda dėl to, kad kvantinės taisyklės neleidžia kopijuoti būsenų, o tiesioginis būsenų matavimas sugriauna superpoziciją ir joje saugomą informaciją. Kvantinių klaidų aptikimas ir taisymas yra įmanomas pasitelkiant vėlgi supintąsių kvantines būsenas.

Dėl tolygiai galinčių kisti būsenų kvantinis kompiuteris iš pirmo žvilgsnio atrodytų analoginius įrenginys. Analoginiai klasikiniai kompiuteriai neprigijo praktikoje, nes jų klaidas taisyti itin keblu. Tačiau kvantinėje kompiuterijoje klaidos gali būti efektyviai skaitmenizuojamos, pakanka sugebėti taisyti tik tris skirtingo tipo klaidas norint ištaisyti bendriausio tipo (tolydžiai kintančias) klaidas. Tad klaidų taisymo skaitmeninimas bei išmatuotų būsenų skaitmeninis pavidalas rodo, kad kvantinis kompiuteris yra veikiau dualus analoginis–skaitmeninis įrenginys, ir galbūt tiksliausiai – tiesiog kvantinis.

Siekdami lengviau įsivaizduoti kvantinio skaičiavimo procesą, pateikiame jį visą sluoksniu (angl. *full quantum stack*) iliustracijoje 1.5 pav. Aukščiausiaame loginiame lygmenyje kvantinio kompiuterio programavimą galima atlikti įvairiomis programavimo kalbomis, pavyzdžiui, *Python*, *C++*, ar specializuotose kvantinio programavimo aplinkose. Šiame lygmenyje yra pasitelkiami jau suformuluoti kvantiniai algoritmai ar įvairūs jų moduliai, kurie taip pat gali pasitelkti tarpinius klasikinius skaičiavimus.



1.5 pav.: Kvantinio skaičiavimo procesas

Programos yra toliau nukreipiamos į žemesnį kvantinių loginių grandinių lygį, kuriame algoritmai yra išreiškiami loginiais vartais. Kvantinio loginio lygmens programavimas yra nauja paradigma. Nors 10–20 kubitų dydžio procesoriui pateikiamo algoritmo loginių vartų sekas galima iš principo sudėlioti ir optimizuoti „rankomis“, tačiau to tampa nebeįmanoma padaryti procesoriui, turinčiam 1000 kubitų. Todėl neišvengiamai teks automatizuoti kvantinių grandinių kompiliavimą ir optimizavimą. Artimuoju kvantinių kompiuterių raidos laikotarpiu tai gali lemti, ar tam tikrus algoritmus apskritai įmanoma atlikti. Pavyzdžiu, Šoro algoritmą galima realizuoti renkantis panaudoti daugiau kubitų ir mažiau loginių operacijų, arba atvirkščiai. Ribotas kubitų skaičius, tarpusavio kubitų jungčių topologija, ribotas loginių vartų tikslumas bei jų atlikimo trukmė yra vieni iš pagrindinių faktorių, į kuriuos turėtų būti atsižvelgiama kvantinių grandinių kompilavime. Kvantinis loginis lygmuo į tai neatsižvelgia ir todėl turi būti kompiliuojamas pasitelkiant kvantinių kladų taisymo kodus. Šie kodai priklausys nuo kvantinio procesoriaus architektūros ir galbūt specifinės skaičiavimo užduoties.

Loginės operacijos yra toliau išreiškiamos mašinine kalba, kuri įvardija, kokie fiziniai veiksmai (lazerių impulsai, grandinės įtampos keitimai ir t.t.) yra atliekami kontroliuoti kvantiniam procesoriui. Kvantiniame fiziniame lygmenyje yra fizinių kubitų sistema. Ji izoliuota nuo išorinių sąveikų ir veikiama tik per klasikinę–kvantinę ribą atlikti unitarinėms kvantinių būsenų transformacijoms bei matavimams. Pasitelkiama klasikinė kompiuterija atlikti kvantinio procesoriaus veikimo diagnostikai, papildomiems sistemą stabilizuojantiems ir kaidas mažinantiems protokolams.

1.8 Kvantinių kompiuterių charakteristikų palyginimas

Atsižvelgdami į kuriamų kvantinių kompiuterių neidealias veikimo charakteristikas norime geriau suprasti, kaip galima įvertinti jų gebėjimą įvykdyti pateiktą algoritmą, ir tarpusavyje palyginti skirtingus įrenginius. Vienas iš pagrindinių rodiklių palyginti skirtingų superkompiuterių skaičiuojamajai galiai yra **flopai** (angl. *floating-point operations per second*, trumpinys FLOPS). Flopais matuojama skaičiuojamoji galia nusako, kiek aritmetinių operacijų skaičių per sekundę sugeba atlikti superkompiuteriai. Pagrindiniai fiziniai rodikliai, turintys įtakos flopams, yra naudojamų procesorių dažniai ir procesorių lustų skaičius. Antrasis rodiklis nusako gebėjimą paralelizuoti skaičiavimus, tai leidžia atlikti daugiau loginių operacijų per tą patį procesoriaus laiko ciklą.

Dominančiam kvantiniam algoritmui atlikti egzistuoja minimalūs kubitų bei loginių vartų skaičiaus reikalavimai. Jeigu kaidos neribotų kvantinių kompiuterių veikimo, jų skaičiavimo galia taip pat galiausiai atsiremtų į loginių operacijų skaičių per laiko vienetą. Imant konkretų pavyzdį, galima įvertinti Šoro algoritmui reikalingus laiko išteklius siekiant įveikti $n = 1024$ bitų RSA kriptografija užšifruotą turinį. Teoriniai skaičiavimai rodo, kad tam reikia maždaug $2n$ loginių kubitų, o sudėtingumas yra nulemtas $O(n^3 \log n)$ skaičiumi Tofoli loginių vartų. Tad reikėtų maždaug 2050 kubitų ir bent maždaug 10^{10} loginių vartų. Loginių vartų atlikimo trukmė skiriasi tarp fizinių kvantinio procesoriaus realizacijų, tačiau bendrai yra žymiai ilgesnė (veikia apytikriai MHz dažniu) nei klasikinių procesorių (veikiančiu GHz dažniu). Pavyzdžiu, transmonais pagrįstuose IBM procesoriuose šios knygos rašymo metu 2 kubitų *cX* loginiai vartai, ribojantys bendrą loginių operacijų atlikimo trukmę, užtrunka maždaug 200 nanosekundžių. O štai Jonų gardelėmis pagrįstuose procesoriuose *IonQ* 2 kubitų loginiai vartai atliekami žymiai lėčiau, apytikriai 200 mikrosekundžių per operaciją. Darant prielaidas, kad tiek užtrunka atlikti ir 3 kubitų Tofoli loginius vartus bei kad kvantineis grandinės gylis abiejose architektūrose yra toks pat kaip ir loginių vartų skaičius, apytikriai 10^{10} loginių vartų užstruktū apytikriai 1 ir apytikria 1 $\times 10^2$ valandų,

atitinkamai.

Aktreipiame dėmesį, kad čia naudojome loginių kubitų skaičių. Vienam loginiam kubitui realiuoti gali prieikti 10^3 fizinių kubitų, loginių vartų skaičius didės taip pat. Vis dėlto tai yra veikiau pesimistinis įvertinimas – kvantinės technologijos itin sparčiai tobulėja ir tikėtina, kad atsiras sumanėsni būdai, kaip sumažinti kvantinių skaičiavimų išteklius.

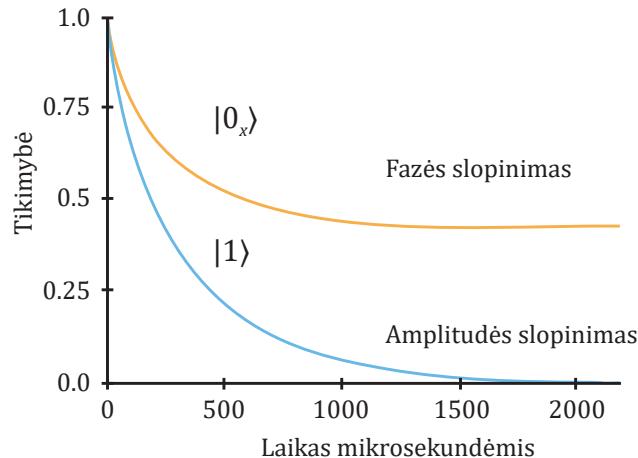
1.9 Dekoherencijos trukmė ir loginių vartų tikslumas

Artimuoju kvantinių kompiuterių raidos laikotarpiu yra svarbiau atsižvelgti į skaičiavimų trukmę ir tikslumą ribojančius faktorius. Elementariausiai fiziniai rodikliai, nusakantys kvantinio procesoriaus gebėjimą atlikti skaičiavimus, yra kubitų dekoherencijos trukmė ir loginių vartų tikslumas. Dekoherencija yra procesas, kurio metu nekontroliuojamos kubitų sąveikos su kitomis kompiuterio vidinėmis ar išorinėmis kvantinėmis sistemomis nebeleidžia demonstruoti interferencinių savybių. Mūsų požiūriu, nežinant apie įvykusias sąveikas atrodys, kad laikui bėgant kubitų būsenos tampa atsitsiktinės, praranda gebėjimą išlaikyti superpozicijas ir demonstruoti supynimą su kitais procesoriaus kubitalis. Sakoma, kad kubitų būsenos sunyksta.

Standartiškai yra išskiriamos dvi dekoherencijos trukmės T_1 ir T_2 . Tikimybė, kad kubitas, paruoštas į būseną $|1\rangle$ atskaitos laike $t = 0$, joje išliks praėjus laiko intervalui t , paprastai seka eksponentinę gesimo funkciją $e^{-\frac{t}{T_1}}$. Dekoherencijos laiko konstanta T_1 nusako vadinamąjį **amplitudės slopinimo** (angl. *amplitude-damping*), arba **energijos relaksacijos** (angl. *energy relaxation*) laiką. Istoriskai kubitų sistemose, pagrįstose superlaidžiomis grandinėmis ir jonų gaudyklėmis, kubitų būsenos $|0\rangle$ ir $|1\rangle$ nusakomas žemiausio ir gretimo aukštesnio energijos lygmens kvantinės sistemos būsenomis. Energijos relaksacija šiame kontekste, $|1\rangle \rightarrow |0\rangle$, yra procesas, kuriamė aukštesnio energijos lygmens būseną savaime ar stimuliuojama iš išorės sugrįžta į žemesnį lygi. Pavyzdžiui, praėjus laikui $t = T_1$ tikimybė pamatavas kubitą vis dar jį rasti $|1\rangle$ būsenoje yra $p = 0.37$.

Dekoherencijos trukmės konstanta T_2 nusako **fazės slopinimo** (angl. *phase-damping*) trukmę, dar vadinamą **depoliarizacijos** trukmę (angl. *depolarization*). Šis rodiklis įvardija kubitų, esančių būsenų superpozicijoje, gebėjimą išlaikyti santykinę fazę. Kaip ir laiko konstantą T_1 , T_2 galima nustatyti atlikus seriją matavimų naudojant skirtingus laiko intervalus, po kurių išmatuojamos būsenos. Fazės slopinimo konstantai rasti kubitas yra paruošiamas į superpozicijos būseną $|0_x\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$ atskaitos laike $t = 0$. Praėjus laiko intervalui t yra pritaikoma Hadamardo transformacija ir išmatuojama būsena. Jeigu įvyko fazės klaida, kubito būsena pasikeičia taip: $|0_x\rangle \rightarrow |1_x\rangle$, tad atlikus H vartus bus rasta būsena $|1\rangle$, o ne $|0\rangle$. Tikimybė rasti $|0\rangle$ būseną seka eksponentinę gesimo funkciją $e^{-\frac{t}{T_2}+1}$. Po pakankamai ilgo laiko $t \gg T_2$ tikimybės rasti $|0\rangle$ arba $|1\rangle$ būseną susivienodina. Abiejų minėtų procesų gesimo funkcijos yra parodytos 1.6 pav.

Individualių kvantinių loginių vartų tikslumą (angl. *gate fidelity*) galima įvertinti palyginus, kiek būsenos, kurioms buvo atliekami šie loginiai vartai, yra artimos būsenoms, kurioms buvo atlikti idealiai veikiantys loginiai vartai. Idealiai veikiančių loginių vartų efektas kvantinei būsenai gali būti apskaičiuotas popieriaus lape arba klasikiniu kompiuteriniu modeliavimu. Gerai funkcionuojančių vartų tikslumas neturėtų priklausyti nuo pateiktos kvantinės būsenos, tad dažnai atliekant šį testą yra suvidurkinamas loginių vartų efektas daugeliui skirtinį pradinių būsenų. Loginių vartų netikslumą įvedamos klaidos sumuoja su atliekamų vartų skaičiumi, tad kuo ilgesnis algoritmas, tuo tikslesni turėtų būti loginiai vartai. Norint sėkmingai vykdyti kladų taisymo algoritmus ir pasiekti kladoms atsparią skaičiavimų ribą, reikalingas tam tikras minimalus logi-



1.6 pav.: Fazės ir amplitudės slopinimas. Parodytos kreivės nusako, kokia tikimybė rasti kubitą pradinėje nurodytoje būsenoje bėgant laikui. Iliustracijai panaudota $T_1 = T_2 = 300$ mikrosekundžių dekoherencijos trukmė

nių vartų tikslumas. Iprastai kvantiniame procesoriuje tikslumą riboja 2 kubitų loginiai vartai, kuriuos yra fiziškai sudėtingiau realizuoti negu 1 kubito loginius vartus.

II skyrius

Matematiniai įrankiai rinkinys

2.1 Tiesinė algebra

Tiesinė algebra yra kvantinės kompiuterijos matematinė kalba. Šiame skyriuje pateikiame jos pagrindinius konceptus bei kitus matematinius įrankius, kurių naudojimą bus galima aptikti įvairiose knygos vietose. Tai jokiui būdu nėra pilnutinis tiesinės algebrros išdėstymas – išsamesnį pristatymą, esant poreikiui, galima rasti matematikos ir fizikos sričių literatūros šaltiniuose, kurių dalis yra pateikta knygos pabaigoje.

Pagrindiniai objektai tiesinėje algebroje yra vektorių erdvės, kurių elementai – vektoriai. Kvantinėje kompiuterijoje aptinkamos vien baigtinio dimensijų skaičiaus vektorių erdvės. Taip yra todėl, kad naudojamas baigtinis n skaičius kubitų, pavienių kubitų būsena yra nusakoma vektoriumi 2 dimensijų vektorių erdvėje, o jų bendra būsena 2^n dimensijose. Baigtinio dimensijų skaičiaus sistemas yra lengviau analizuoti – išvengiama matematinijų komplikacijų, dažnai sutinkamų kitose kvantinėse sistemoje su begaliniu dimensijų skaičiumi ir reikalaujančių papildomo žinių bagažo. Tad, šiuo požiūriu, koncentruojantis vien į baigtinio dimensijų skaičiaus sistemas galima žymiai greičiau įsisavinti esminius konceptus.

Vektoriai yra aptinkami įvairiausiose srityse ir dažnai apibūdinami kaip rodyklytės erdvėje, turinčios tam tikrą ilgį bei orientaciją. Kvantineje kompiuterijoje tik specifinėse situacijose tokia vektorių vaizdinė reprezentacija yra įmanoma, pavyzdžiui, perteikiant vieno kubito būseną vektoriumi Blocho sferoje. Panašios vizualizacijos, be abejo, yra naudingos įgauti pradinę intuiciją apie vektorius ir jų transformacijas. Vis dėlto yra tikslingiau galvoti apie vektorius tiesiog kaip apie abstrakčius objektus, turinčius nurodytas savybes. Jos nusako, kaip galima sudėti vektorius ir sudauginti juos su skaičiais. Vektorių vizualizacijos keblumas kvantineje kompiuterijoje atsiranda dėl to, kad šie vektoriai yra apibréžti kompleksinėje **vektorių erdvėje** (angl. *complex vector space*), o ne įprastinėje realiųjų skaičių **Euklido erdvėje** (angl. *Euclidean vector space*). Be to, kvantinės būsenos yra formaliai nusakomas **spinduliais** (angl. *rays*), nors ir įprasta sakyti, kad vektoriais. Spindulys yra grupė vektorių, kurie skiriasi tarp savęs tik globalia faze (čia reikia atskirti nuo santykinės fazės). Pavyzdžiui, kvantine būsena, nusakyta $|v\rangle$ arba $-|v\rangle$ vektoriais, yra fiziškai identiška, nors tai ir reikštų skirtingus euklidinius vektorius, orientuotus antiparaleliai.

Kvantineje kompiuterijoje kompleksinių vektorių baigtinio dydžio d dimensijų erdvę žymėsime simboliu V^d . Viršuje užrašome erdvės dimensiją d (natūralusis teigiamasis skaičius), jeigu yra

poreikis specifinėje situacijoje ją įvardyti. Vektorių erdvės elementai yra vektoriai, kuriuos visada galime išreikšti sugrupuotų kompleksinių skaičių stulpeliu:

$$|v\rangle = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}. \quad (2.1)$$

Skliausteliai, turintys formą $| \dots \rangle$ yra naudojami įvardyti kad šis objektas yra vektorius stulpelis. Skaičius žymime indeksuotomis mažosiomis raidėmis z_i . Norint glaustai parodyti, kad vektorius $|v\rangle$ priklauso tam tikrai vektorių erdvei V , rašome $|v\rangle \in V$. Simbolis \in nusako, kad kairėje esantis objektas yra vienas iš dešinėje esančio objekto elementų.

Elementų z_i skaičius vektoriuje nusako vektorių erdvės dimensijų skaičių. Vektorius galima sudėti tik tuo atveju, jeigu jie priklauso tai pačiai vektorių erdvei. Apsistojant ties viena reprezentatyvia vektorių erdve V , joje yra apibrėžtos vektorių sudėties operacijos, tenkinančios šias savybes:

$$|v_1\rangle + |v_2\rangle = |v_2\rangle + |v_1\rangle; \quad (2.2)$$

$$|v_1\rangle + (|v_2\rangle + |v_3\rangle) = (|v_1\rangle + |v_2\rangle) + |v_3\rangle. \quad (2.3)$$

Tai bendrai parodo, kad eiliškumas vektorių sudėčiai nėra svarbus. Vektorių erdvėje taip pat egzistuoja nulinis vektorius, analogiškas nuliniam skaičiui. Kvatinėje kompiuterijoje $|0\rangle$ yra jau paskirtas kitkam, tad nusakyti nuliniam vektoriui vartojanamas simbolis 0. Nulinio vektoriaus efektas apibūdinamas $|v\rangle + 0 = |v\rangle$, iš to gauname $|v\rangle - |v\rangle = 0$.

Kompleksinio skaičiaus z_i daugyba su vektoriais tenkina šias savybes:

$$z(|v_1\rangle + |v_2\rangle) = z|v_1\rangle + z|v_2\rangle; \quad (2.4)$$

$$(z_1 + z_2)|v\rangle = z_1|v\rangle + z_2|v\rangle; \quad (2.5)$$

$$(z_1 z_2)|v\rangle = z_1(z_2|v\rangle). \quad (2.6)$$

Atkreipiame dėmesį, kad sandaugose praleidžiame daugybos simbolį, tad du vienas šalia kito raidėmis nusakyti skaičiai (ar skaičius su vektoriumi) reiškia, kad jie yra sudauginami. Skaičių daugybai nesvarbu eiliškumas $z_1 z_2 = z_2 z_1$. Taip pat galima daugyba su nuliniu skaičiumi, deja, taip pat žymimiu 0, $0|v\rangle = 0$. Tai yra visos elementariosios ir itin intuityvios aritmetinės operacijos. Šių aritmetinių operacijų metu gautas vektorius vėlgi priklauso tai pačiai vektorių erdvei. Kvatinėse sistemoje yra apibrėžta vadinamoji dviejų vektorių vidinė sandauga, žymima $\langle v|u\rangle$, kurios rezultatas yra skaičius. Prie vidinės sandaugos šiame skyriuje dar sugrižime.

Toliau primename vektorių, išreikštų sugrupuotų skaičių stulpeliais, aritmetiką. Imant kaip pavyzdį trijų elementų vektorius, sudėtis nusakoma:

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} z_4 \\ z_5 \\ z_6 \end{bmatrix} = \begin{bmatrix} z_1 + z_4 \\ z_2 + z_5 \\ z_3 + z_6 \end{bmatrix}. \quad (2.7)$$

Čia kiekvienas elementas z_i yra kompleksinis skaičius. Vektoriaus ir skaičiaus g sandauga:

$$g \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} g z_1 \\ g z_2 \\ g z_3 \end{bmatrix}. \quad (2.8)$$

Kiti du svarbūs konceptai yra vektorių erdvę dengiantis vektorių rinkinys (angl. *spanning set*) ir vektorių **tiesinė nepriklausomybė** (angl. *linear independence*). Vektorių erdvę V dengiantis rinkinys $\{|v_1\rangle, |v_2\rangle, |v_3\rangle \dots\}$ yra toks vektorių rinkinys, kurių tiesinėmis kombinacijomis galima išreikšti bet kokį vektorių $|v\rangle$, esantį V . Tai yra:

$$|v\rangle = \sum_i z_i |v_i\rangle. \quad (2.9)$$

Tiesinė vektorių nepriklausomybė leidžia formaliai nusakyti vektorių erdvės dimensiją ir tuo pačiu rasti mažiausią skaičių vektorių, dengiančių vektorių erdvę V . Vektoriai $\{|v_1\rangle, |v_2\rangle, \dots, |v_n\rangle\}$ yra tiesiškai priklausomi, jeigu bent vienas vektorius šiame rinkinyje, sakysime $|v_2\rangle$, gali būti išreikštas likusių $n-1$ vektorių tiesinėmis kombinacijomis:

$$z_1 |v_1\rangle + z_3 |v_2\rangle + \dots + z_n |v_n\rangle = |v_2\rangle. \quad (2.10)$$

Tiesiškai nepriklausomų vektorių rinkinyje nė vieno iš jų neįmanoma išreikšti kitų likusiųjų sumą. Formaliai, tiesiškai priklausomas vektorių rinkinys tenkina šią lygybę:

$$z_1 |v_1\rangle + z_2 |v_2\rangle + \dots + z_n |v_n\rangle = 0. \quad (2.11)$$

Čia suma lygi nuliniam vektoriui ir joje egzistuoja skaičiai z_i , kurie ne visi lygūs nuliui $z_i \neq 0$. O štai tiesiškai nepriklausomame vektorių rinkinyje ši lygybė gali būti tenkinama tik tuo atveju, jeigu visi skaičiai $z_i = 0$.

Galima parodyti, kad bet kurie du skirtinių tiesiškai nepriklausomų vektorių rinkiniai, dengiantys tą pačią vektorių erdvę V , turi vienodą skaičių vektorių. Vektoriai, priklausantys tokiam rinkiniui, yra vadinami **bazinių vektorių** (angl. *basis vectors*), o jų skaičius rinkinyje formaliai nusako vektorių erdvės V dimensiją. Pavyzdžiui, imkime 3 dimensijų erdvės bazinių vektorių rinkinį:

$$|v_1\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad |v_2\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad |v_3\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (2.12)$$

Matome, kad bet kokį vektorių $|v\rangle$ šioje erdvėje galime išreikšti jų sumą:

$$|v\rangle = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = a \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + b \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + c \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (2.13)$$

Taip pat akivaizdu, kad dviejų bazinių vektorių, pavyzdžiui $|v_1\rangle$ ir $|v_3\rangle$, nepakaktų išreikšti visus įmanomus vektorius šioje erdvėje. Keturi baziniai vektoriai 3 dimensijose būtų perteklius, kadangi vieną iš jų visada galime išreikšti kitų sumą. Egzistuoja begalė skirtinių bazinių vektorių rinkinių. Pavyzdžiui, kitas rinkinys:

$$|v_1\rangle = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix}, \quad |v_2\rangle = \begin{bmatrix} 2 \\ 5 \\ 3 \end{bmatrix}, \quad |v_3\rangle = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \quad (2.14)$$

Siekdami patikrinti šių vektorių tiesinę nepriklausomybę išspręstume trijų lygčių sistemą:

$$\begin{aligned} 2a + 2b + c &= 0, \\ -a + 5b + c &= 0, \\ 3b + c &= 0. \end{aligned} \quad (2.15)$$

Sistemą išsprendę rastume, kad vieninteliai skaičiai, tenkinantys lygybę, yra $a = b = c = 0$. Vienas svarbus skirtumas tarp šio bazinio vektorių rinkinio ir parodyto anksčiau yra tai, kad anksčiau rinkinyje visi vektoriai tarpusavyje sudaro stačiuosius kampus. Kompleksinėje vektorių erdvėje statmenumo konceptas yra vadinamas ortogonalumu (angl. *orthogonality*). Dviejų vektorių ortogonalumą, kaip matysime vėliau, galima nustatyti atlikus jų vidinę sandaugą, $\langle v_m | v_n \rangle$. Jeigu du (nenuliniai) vektoriai yra ortogonalieji, jų vidinė sandauga visada bus skaičius, lygus nuliui.

Kvantinėje kompiuterijoje 2 dimensijų kompleksinių vektorių erdvės V^2 nusako individualių kubitų būsenų erdvę. Bendra vieno kubito būsena yra vektorius:

$$|\psi\rangle = a|0\rangle + b|1\rangle. \quad (2.16)$$

Čia $|0\rangle$ ir $|1\rangle$ yra standartiškai naudojami baziniai vektoriai, dar vadinami skaičiuojamaisiais baziniais vektoriais. Išreikškus juos stulpeliniu vektoriumi:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (2.17)$$

Galima lengvai patikrinti, kad šis dviejų vektorių rinkinys $\{|0\rangle, |1\rangle\}$ yra tiesiškai nepriklausomas ir todėl jų kombinacijomis galima išreikšti bet kokį kitą vektorių $|v\rangle$ šioje 2 dimensijų erdvėje keičiant koeficientus a ir b :

$$|v\rangle = a \begin{bmatrix} 1 \\ 0 \end{bmatrix} + b \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}. \quad (2.18)$$

Kvantinėje mechanikoje koeficientai turi tenkinti lygybę $|a|^2 + |b|^2 = 1$. Nors šis bazinių vektorių rinkinys kvantinėje kompiuterijoje naudojamas standartiškai, tačiau skirtingų bazinių rinkinių yra begalė. Kitas dažnai aptinkamas ortogonalus 1 kubito būsenas nusakantis rinkinys yra šis:

$$|0_x\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad |1_x\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \quad (2.19)$$

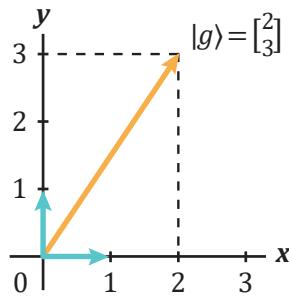
Palyginę su $\{|0\rangle, |1\rangle\}$ matome, kad $\{|0_x\rangle, |1_x\rangle\}$ rinkinyje bazinius vektorius galime išreikšti sudėjė bei atėmę pirmųjų elementus, atitinkamai, naudojant koeficientus $a = b = 1/\sqrt{2}$:

$$|0_x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |1_x\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \quad (2.20)$$

Dviejų bazinių vektorių suma kvantinėje kompiuterijoje yra vadinama būsenų superpozicija. Tiesinės algebras požiūriu, tai tiesiog atspindi vieną galimą būdą išreikšti vektorių sąlygiškai su kitais vienais.

Užbaigdami šią dalį atitrūksime nuo kompleksinių skaičių ir kvantinių sistemų. Kompleksinių skaičių vartojimas komplikuoja vektorių, kaip matematinių objektų, iliustravimą. Siekiant su sidarysti intuiciją yra palankiau sugrįžti prie realiosios Euklido vektorių erdvės, kurioje galima vektorius pavaizduoti rodyklyte. Euklido erdvėje vektorių daugyba yra galima tik su realaisiais skaičiais. Realieji skaičiai yra vartojami nusakant kasdienius dydžius, tokius kaip atstumas, aukštis, valiutos kiekis saskaitoje ir panašiai. Vektorius nusako dydį, bet dar pateikia ir informaciją apie kryptį. Paprastas pavyzdys būtų greičio vektorius, suteikiantis informaciją apie greitį bei judėjimo kryptį. Imkime kaip pavyzdį greitumo vektorių $|g\rangle$, nusakantį greitį 2 dimensijose (plokštumoje):

$$|g\rangle = \begin{bmatrix} 2 \\ 3 \end{bmatrix}. \quad (2.21)$$



2.1 pav.: Vektoriaus pavyzdys plokštumoje. Joje priskirta stačiakampė x – y kordinacijų sistema; vienetiniai vektoriai pažymėti žalia spalva

Galime žvelgti į šiuos du skaičius stulpelyje, kaip suteikiančius koordinates (x , y). Naudojant statmeną x – y kordinacijų sistemą, šis vektorius pavaizduotas 2.1 pav.

Norėdami nubrėžti $|g\rangle$ vektorių, einame 2 žingsnius x ašimi į dešinę ir 3 y ašimi į viršų. Šiuo atveju nėra svarbu, ar vektorius prasideda nuo 0, ar kitur, svarbu vektoriaus orientacija ir ilgis. Vektoriaus ilgis $|g\rangle$ nusakys šiame pavyzdyje greitį, kurį rodytų spidometras. Taikydami Pitagoro teoremą randame $|g\rangle$ ilgi $\sqrt{2^2 + 3^2} = \sqrt{13}$. Vektoriaus iliustracijoje matome $|0\rangle$ ir $|1\rangle$, pažymėtus rodyklytėmis, kurie atlieka, analogiškai su kubitais, bazinių vektorių vaidmenį. Jie yra statmeni (ortogonalieji) vienas kito atžvilgiu ir vienetinio ilgio. Vektorių $|g\rangle$ išreiškiame jais taip:

$$|g\rangle = 2|0\rangle + 3|1\rangle = 2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 3 \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (2.22)$$

Visi įmanomi vektoriai šioje plokštumoje gali būti sukonstruoti naudojant šiuos bazinius vektorius bei keičiant koeficientus a ir b . Bazinių vektorių rinkinys, analogiškas minėtam kubitų rinkiniui $\{|0_x\rangle, |1_x\rangle\}$, šioje plokštumoje būtų gautas pasukus $|0\rangle$ ir $|1\rangle$ kartu pagal laikrodžio rodyklę 45° laipsnių kampu išlaikant tarp jų statujį kampą.

2.2 Kompleksiniai skaičiai

Šie skaičiai turi realiąjį ir menamąjį dalis ir yra bendrai išreiškiami $z = a + bi$. Čia a ir b yra realieji skaičiai, kompleksinio skaičiaus realioji (Re) ir menamoji (Im) dalys yra atitinkamai a ir b . Tai dar gali būti rašoma $\text{Re}(z) = a$, $\text{Im}(z) = b$. Skaičių b dauginantis raide i žymimą narys yra menamasis vienetas, turintis savybę:

$$i^2 = -1. \quad (2.23)$$

Tad į realiųjų skaičių galime žiūrėti kaip į kompleksinių skaičių, kuriame menamoji dalis $b = 0$. Atliekant dviejų kompleksinių skaičių arba kompleksinio ir realiojo skaičiaus sudėtis, realiosios ir menamosios dalys yra sudedamos tarpusavyje atskirai:

$$(a + bi) + (c + di) = (a + c) + (b + d)i. \quad (2.24)$$

Taikydami menamojo vieneto savybę bei kompleksinių skaičių sudėtį, dviejų kompleksinių skaičių sandaugą randame:

$$(a + bi)(c + di) = (ac - bd) + (ad + bc)i. \quad (2.25)$$

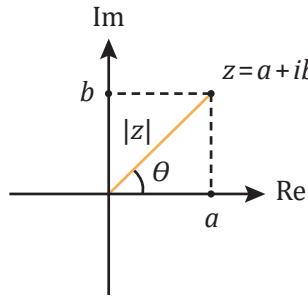
Atliekant skaičiavimus kvantinėje kompiuterijoje dažnai naudojamas vadinamasis kompleksinis skaičiaus jungimas, dar įvardijamas kaip **konjugacija** (angl. *complex conjugation*). Apačioje parodytas ryšys tarp kompleksinio skaičiaus z ir jo kompleksinės jungties, kuri yra žymima su žvaigždute z^* :

$$z = a + bi \rightarrow z^* = a - bi. \quad (2.26)$$

Kompleksinis jungimas apverčia menamosios dalies ženklą. Pavyzdžiui, kvantinės būsenos kompleksinės amplitudės z kvadratas yra apskaičiuojamas naudojant kompleksinę jungtį:

$$|z|^2 = zz^* = (a + bi)(a - bi) = a^2 + b^2. \quad (2.27)$$

Čia $|z|$ reiškia šio skaičiaus modulį ir matome, kad zz^* visada yra realusis skaičius. Kompleksinius skaičius galime išreikšti grafiškai plokštumoje, kurioje horizontalioji ir vertikalioji ašys nusako realiąją (Re) ir menamąją (Im) dalis, atitinkamai.



2.2 pav.: Kompleksinio skaičiaus pavaizdavimas stačiakampėje Re–Im koordinačių sistemoje pateikiant koordinates (a, b) . Naudojant polines koordinates, pateikiamas spindulio ilgis ir kampus $(|z|, \theta)$

Viršuje parodyti du būdai išreikšti kompleksiniams skaičiui. Galime nusakyti kompleksinį skaičių (a, b) koordinatėmis Re–Im koordinačių sistemoje arba išreikšti z polinėje koordinačių sistemoje. Nubrėžę spindulį nuo koordinačių centro iki (a, b) taško, spindulys sudaro θ kampą su Re ašimi. Kadangi spindulio ilgis yra $|z| = \sqrt{a^2 + b^2}$, koordinates (a, b) galime išreikšti $(a = |z| \cos(\theta), b = |z| \sin(\theta))$. Tad polinėje koordinačių sistemoje nusakomi du parametrai, spindulio ilgis ir kampus $(|z|, \theta)$.

Jeigu imsime kompleksinį skaičių, kurio spindulio ilgis $|z| = 1$, Oilerio formulė (angl. *Euler formula*) mums rodo:

$$e^{i\theta} = \cos(\theta) + i \sin(\theta). \quad (2.28)$$

Oilerio formulėje e yra natūraliojo logaritmo pagrindas, kurio reikšmė $e \sim 2.718 \dots$. Akivaizdu, kad Oilerio funkcijos modulis $|e^{i\theta}| = 1$. Bet kokį kompleksinį skaičių z galime išreikšti taip:

$$z = |z|e^{i\theta}. \quad (2.29)$$

Taip išreikšto skaičiaus z kompleksinė jungtis bei jo modulio kvadratas, minėti viršuje, gaunami atliekant konjugaciją eksponentėje:

$$z^* = |z|e^{-i\theta}; \quad (2.30)$$

$$|z|^2 = |z||z|e^{-i\theta}e^{i\theta} = |z|^2. \quad (2.31)$$

Viršuje pritaikėme eksponenčių daugybos formulę, $e^{ia}e^{ib} = e^{i(a+b)}$ bei $e^0 = 1$.

2.3 Vidinė vektorių sandauga

Šioje dalyje apibūdiname kvantinėje kompiuterijoje dažnai aptinkamą **vidinę dviejų vektorių sandaugą** (angl. *inner product*), dar vadinama **skaliarine sandauga** (angl. *scalar product*). Vektorius $|\psi\rangle$ su tokio tipo skliausteliais yra vadinamas *ket* ir yra asocijuojamas su stulpeliniu vektoriumi. Vektorius eilutė yra žymimas apsuktais skliausteliais, $\langle\psi|$ ir vadinamas *bra*. Iš bet koks *ket* vektoriaus galime padaryti *bra* vektorių dviem žingsniais. Pavyzdžiui, turime *ket* su keturiais elementais:

$$|\psi\rangle = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}. \quad (2.32)$$

Visi $|\psi\rangle$ vektoriaus elementai yra bendrai kompleksiniai skaičiai. Pirmame žingsnyje vektorius eilutė yra gaunamas atlikus vadinamąją **transpoziciją** (angl. *transposition*), žymimą T raide virš vektoriaus $|\psi\rangle^T$. Elementai stulpelyje iš viršaus į apačią yra pergrupuojami eilutėje iš kairės į dešinę:

$$|\psi\rangle^T = [a_1 \ a_2 \ a_3 \ a_4]. \quad (2.33)$$

Galiausiai, *bra* vektorius yra gaunamas atlikus transponuotajam vektoriui kiekvieno jo elemento kompleksinę jungtį $(|\psi\rangle^T)^* = |\psi\rangle^\dagger = \langle\psi|$. Ši dviguba operacija yra vadinama ermitine jungtimi, kuriai nurodyti vartojamas durklo formos ženklas \dagger . Tad galiausiai randame:

$$\langle\psi|^\dagger = \langle\psi| = [a_1^* \ a_2^* \ a_3^* \ a_4^*]. \quad (2.34)$$

Bra vektoriai yra **dualūs** *ket* vektoriams (angl. *dual vector*) – kiekvienas *bra* turi vieną sau atitinkantį *ket*. Formaliai, jeigu $|\psi\rangle$ yra V vektorių erdvės elementas, tai jam dualus $\langle\psi|$ vektorius yra dualios \bar{V} vektorių erdvės elementas. Kadangi *ket* ir *bra* yra skirtinę erdvę elementai, jų tarpusavyje sudėti negalima, tai yra neapibrėžta operacija. Tačiau visos minėtos operacijos *bra* vektorių erdvėje \bar{V} yra identiškos *ket* V erdvėi.

Vidinė dviejų vektorių sandauga yra atliekama tarp *bra* ir *ket* vektorių, tiesinėje algebroje – tarp vektoriaus eilutės ir stulpelio. Imkime kaip pirmą pavyzdį vidinę $|\psi\rangle$ vektoriaus sandaugą su sau dualiu vektoriumi, $\langle\psi|\psi\rangle$. Rašant vektoriaus elementais, vidinė sandauga formaliai išreiškiama:

$$\begin{aligned} \langle\psi|\psi\rangle &= [a_1^* \ a_2^* \ a_3^* \ a_4^*] \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \\ &= a_1^* a_1 + a_2^* a_2 + a_3^* a_3 + a_4^* a_4 \\ &= |a_1|^2 + |a_2|^2 + |a_3|^2 + |a_4|^2. \end{aligned} \quad (2.35)$$

Matome, kad atlikus vidinę vektoriaus sandaugą su sau dualiu vektoriumi rezultatas visada bus realusis neneigiamasis skaičius, $\langle\psi|\psi\rangle \geq 0$. Šis skaičius nusako kompleksinių kvadratų sumą, kurią dar galime identifikuoti kaip $|\psi\rangle$ vektoriaus ilgio kvadratą. Kvintinės būsenos yra nusakomos normuotaisiais vektoriais, tai yra, turinčiais vienetinį vektoriaus ilgį. Tad normuotojo vektoriaus vidinė sandauga su savo dualiuoju vektoriumi visada lygi vienetui, $\langle\psi|\psi\rangle = 1$. Galime bet kurį vektorių padaryti normuotuoju, jeigu jis toks nėra, padalindami jį iš skaičiaus, nusakančio vektoriaus ilgi $\sqrt{\langle\psi|\psi\rangle}$:

$$\hat{|\psi\rangle} = |\psi\rangle / \sqrt{\langle\psi|\psi\rangle}. \quad (2.36)$$

Dviejų skirtinį tos pačios vektorių erdvės vektorių $|\psi\rangle$ ir $|\phi\rangle$ vidinė sandauga $\langle\psi|\phi\rangle$ yra:

$$\langle\psi|\phi\rangle = [a_1^* a_2^* a_3^* a_4^*] \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = a_1^* b_1 + a_2^* b_2 + a_3^* b_3 + a_4^* b_4. \quad (2.37)$$

Dviejų vektorių, turinčių n elementų, vidinė sandauga $\langle\psi|\phi\rangle$ yra glaustai užrašoma:

$$\langle\psi|\phi\rangle = \sum_{i=1}^n a_i^* b_i. \quad (2.38)$$

Ket ir *bra* eiliškumas (kuris vektorius rašomas kairėje ir kuris – dešinėje) vidinėje sandaugoje gali būti svarbus, nes sudauginus skirtinį vektorius gaunamas skaičius yra bendrai kompleksinis. Skirtumas tarp jų eiliškumo slypi gauto skaičiaus kompleksiniame jungime. Ši principą galima išreikšti taip:

$$\langle\psi|\phi\rangle = (\langle\phi|\psi\rangle)^*. \quad (2.39)$$

Skaičius vidinėje sandaugoje galime visada iškelti už jos:

$$\langle\psi|(z_1|\phi_1\rangle + z_2|\phi_2\rangle) = z_1\langle\psi|\phi_1\rangle + z_2\langle\psi|\phi_2\rangle. \quad (2.40)$$

Vidinės sandaugos modulio kvadratas, $|\langle\psi|\phi\rangle|^2$, dar vadinamas kompleksiniu kvadratu, naudojant vektorių simboliką yra:

$$|\langle\psi|\phi\rangle|^2 = \langle\psi|\phi\rangle\langle\psi|\phi\rangle^* = \langle\psi|\phi\rangle\langle\phi|\psi\rangle = \langle\phi|\psi\rangle\langle\psi|\phi\rangle = |\langle\phi|\psi\rangle|^2. \quad (2.41)$$

Viršuje matome dviejų kompleksinių skaičių sandaugą, kuri yra gaunama iš dviejų vidinių vektorių sandaugų $\langle\psi|\phi\rangle$ ir $\langle\phi|\psi\rangle$. Kadangi šie du nariai nusako skaičius, juos galime pergrupuoti, kaip pageidaujama, nekeičiant rezultato; tai ir parodyta viršuje. Kompleksiniame kvadrate vektorių eiliškumas nėra svarbus.

Ortogonalijų vektorių normuotumas gali būti glaustai užrašytas taip: $\langle v_i|v_j\rangle = \delta_{ij}$. Čia simbolis δ_{ij} vadinamas **Kronekerio delta funkcija** (angl. *Kronecker delta function*), kuri $\delta_{ij} = 0$, jeigu $i \neq j$ (pavyzdžiui, baziniai vektoriai vidinėje sandaugoje skiriasi) bei $\delta_{ij} = 1$, jeigu $i = j$. Imkime vektorių $|v\rangle$, išreikštą ortogonalaisiais baziniais vektoriais $\{|v_i\rangle\}$ su atitinkamais koeficientais z_i :

$$|v\rangle = \sum_i z_i |v_i\rangle. \quad (2.42)$$

Naudodami δ_{ij} , bet kuri i -tajį koeficientą z_i galime rasti atlikę vektoriaus $|v\rangle$ vidinę sandaugą su atitinkamu *bra* $\langle v_i|$:

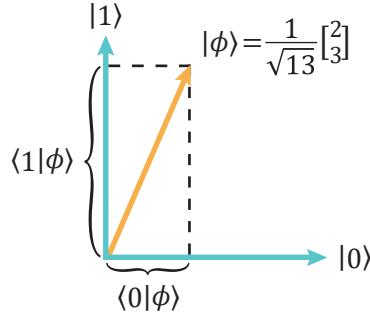
$$z_i = \langle v_i|v\rangle. \quad (2.43)$$

Vidinės sandaugos geometrinę interpretaciją galime lengviau pamatyti naudodami 2 dimensijų Euklido vektorių erdvę. Imkime $|\phi\rangle = a|0\rangle + b|1\rangle$ normuotąjį vektorių, išreikštą $|0\rangle$ ir $|1\rangle$ baziniais vektoriais, ir konkretų a ir b koeficientų pavyzdį:

$$|\phi\rangle = \frac{2}{\sqrt{13}}|0\rangle + \frac{3}{\sqrt{13}}|1\rangle = \frac{1}{\sqrt{13}} \begin{bmatrix} 2 \\ 3 \end{bmatrix}. \quad (2.44)$$

Atlikę šio vektoriaus vidinę sandaugą su $|0\rangle$ vektoriumi rasime:

$$\langle 0|\phi\rangle = \frac{2}{\sqrt{13}}\langle 0|0\rangle + \frac{3}{\sqrt{13}}\langle 0|1\rangle = \frac{2}{\sqrt{13}}. \quad (2.45)$$



2.3 pav.: Vektoriaus $|\phi\rangle$ dekompozicija į statmenus vienetinius (bazinius) vektorius $|0\rangle$ ir $|1\rangle$. Vidinės sandaugos čia nusako $|\phi\rangle$ vektoriaus projekcijas (arba persiklojimą) į atitinkamus bazinius vektorius

Viršuje $\langle 0|0\rangle = 1$ ir $\langle 0|1\rangle = 0$, naudojant bazinių vektorių ortogonalumą ir normuotumą, apibendrinta minėta delta funkcija $\langle v_i | v_j \rangle = \delta_{ij}$. Galime šią vidinę sandaugą iliustruoti 2.3 pav.

Nubrėžę brūkšniuotą statmeną liniją matome, kad vidinė sandauga $\langle 0|\phi\rangle$ indikuoja $|\phi\rangle$ vektoriaus projekcijos dydį (arba persiklojimą) santykinai su $|0\rangle$ vektoriumi. Tai atspindi koeficientas šalia $|0\rangle$ vektoriaus. Analogiskai randama projekcija į $|1\rangle$ bazinį vektorių vidinėje sandaugoje $\langle 1|\phi\rangle$. Net ir kompleksinėje vektorių erdvėje yra teisinga sakyti, kad $\langle 0|\phi\rangle$ parodo, kokį komponentą $|\phi\rangle$ turi $|0\rangle$ atžvilgiu.

Norėdami įvertinti, kiek du normuotieji kompleksiniai vektoriai persikloja, galime apskaičiuoti vidinės sandaugos modulį

$$|\langle \psi | \phi \rangle| = \sqrt{\langle \psi | \phi \rangle \langle \phi | \psi \rangle} = \cos(\theta). \quad (2.46)$$

θ nusako kampą tarp vektorių $|\phi\rangle$ ir $|\psi\rangle$. Verta atkreipti dėmesį, kad kampus čia yra apibrėžtas $0 \leq \theta \leq \pi/2$, kadangi modulis visada grąžina teigiamąjį skaičių. Matome, kad dviejų vienodų vektorių vidinės sandaugos modulis yra 1, o ortogonalinių ($\theta = \pi/2$), žinoma, 0.

2.4 Kubito reprezentacija Blocho sferoje

Vieno kubito būsenas $|\psi\rangle = a|0\rangle + b|1\rangle$ įmanoma išreišksti geometriškai naudojant vadinamąją Blocho sferos reprezentaciją. Kubitas šioje reprezentacijoje pavaizduojamas kaip orientuotas vektorius realioje 3 dimensijų erdvės Blocho sferoje, prasidedantis nuo sferos centro ir užsibaigiantis jos paviršiuje. Tokį kubito būsenos pavaizdavimą galime rasti pirmiausiai išreiškė kompleksinius skaičius a ir b Oilerio formule:

$$|\psi\rangle = |a|e^{i\phi_1}|0\rangle + |b|e^{i\phi_2}|1\rangle. \quad (2.47)$$

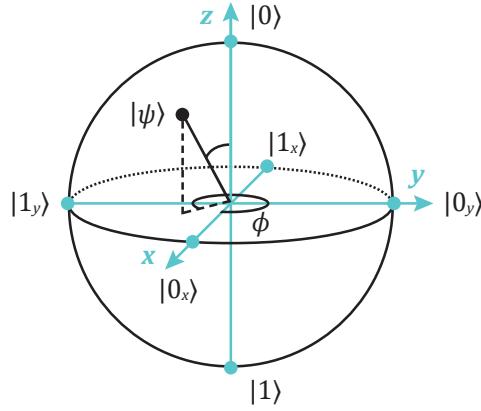
Kadangi $e^{i\phi}$ narių modulis yra $|e^{i\phi}| = 1$ ir amplitudės susideda $|a|^2 + |b|^2 = 1$, galime atlikti keitimą $|a| + |b| = \cos(\alpha) + \sin(\alpha)$. Kampams paprastai yra leidžiama kisti nuo 0 iki 2π apskrantant visą ratą. Tačiau $|a|$ ir $|b|$ yra teigiamieji skaičiai, tad išsaugodami šį reikalavimą turime apibrėžti α kampą $0 \leq \alpha \leq \pi/2$. Konvenciskai yra taikomas keitimas $\alpha = \theta/2$ ir kampas θ apibrėžiamas $0 \leq \theta \leq \pi$, tad amplitudės tampa $|a| + |b| = \cos(\theta/2) + \sin(\theta/2)$. Iškeldami $e^{i\phi_1}$ narij ir pervadindami $\phi_2 - \phi_1 \equiv \phi$, gauname:

$$|\psi\rangle = e^{i\phi_1} (\cos(\theta/2)|0\rangle + \sin(\theta/2)e^{i\phi}|1\rangle). \quad (2.48)$$

Kadangi globali kvantinės būsenos fazė neturi fizinės įtakos, galime panaikinti narį $e^{i\phi_1}$. Taip prieiname prie kubito būsenos Blocho reprezentacijos:

$$|\psi\rangle = \cos(\theta/2)|0\rangle + \sin(\theta/2)e^{i\phi}|1\rangle. \quad (2.49)$$

Iš pirmo žvilgsnio atrodytu, kad turint dvi kompleksines amplitudes a ir b reikia bendrai keturių realiujų skaičių nusakyti bendrai kubito būsenai. Tačiau dėl reikalavimo amplitudžių kvadratui susidėti į vienetą ir įtakos nedarančios globalios fazės pakanka tik dviejų realiujų skaičių. Pirmasis parametras $0 \leq \theta \leq \pi$ nusako ilgumos kampą, kurį šioje reprezentacijoje Blocho vektorius sudaro su Blochos sferos z ašimi. Antrasis parametras $0 \leq \phi < 2\pi$, nusako azimutinį kampą, kurį šio vektoriaus projekcija sudaro $x-y$ plokštumoje (sferos pusiaujoje) skaičiuojant nuo teigiamosios x ašies. Blocho vektorius, tolydžiai keičiant šiuos du parametrus, apibūdina visus įmanomus taškus Blocho sferos paviršiuje.



2.4 pav.: Bendra kubito būsena $|\psi\rangle$ nusakoma Blocho vektoriumi, pažymėtu nuo Blocho sferos centro iki jos paviršiaus. Išilgai x , y ir z ašių pažymėtos dažnai algoritmose pasitelkiamas būsenos

Patikrinkime keletą orientacijų, kad pamatytume, kaip Blocho sferoje pavaizduojamos skirtinges kubito būsenos. Jeigu imsime kampą $\theta = 0$, tada $\cos(0/2) = 1$, $\sin(0/2) = 0$ ir randame, kad į $+z$ orientuotas Blocho vektorius nusako $|0\rangle$ būseną. Jeigu imsime $\theta = \pi$, tada $\cos(\pi/2) = 0$, $\sin(\pi/2) = 1$ ir randame, kad į $-z$ orientuotas vektorius nusako $|1\rangle$ būseną. Šiame kubito parametrizavime $|0\rangle$ ir $|1\rangle$ būsenos, kurios, kaip žinome, yra ortogonalios, vaizduojamos kaip antiparaleliai orientuoti Blocho vektoriai. Tai gali šiek tiek klaidinti ir ši skirtumą, atsirandantį perteikiant 1 kubito kompleksinę vektorių erdvę realioje sferoje, tiesiog reikia prisiminti. Tad jeigu z ašys nusako $|0\rangle$ ir $|1\rangle$ būsenas, kokios būsenos yra išilgai x ir y ašių? Šias būsenas vadinsime atitinkamai $|0_x\rangle$ ir $|1_x\rangle$ bei $|0_y\rangle$ ir $|1_y\rangle$. Išilgai x ašies turime $\theta = \pi/2$, o imdami $\phi = 0$ ir $\phi = \pi$ randame $|0_x\rangle$ ir $|1_x\rangle$:

$$|0_x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |1_x\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \quad (2.50)$$

Išilgai y ašies $\theta = \pi/2$ bei $\phi = \pi/2$ ir $\phi = 3\pi/2$, randame $|0_y\rangle$ ir $|1_y\rangle$:

$$|0_y\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle), \quad |1_y\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle). \quad (2.51)$$

Šios būsenos, kaip ir visos kitos $x-y$ plokštumoje, yra lygios $|0\rangle$ ir $|1\rangle$ būsenų superpozicijos, besiskiriančios viena nuo kitos santykinėmis ϕ fazėmis. Galima patikrinti jų ortogonalumą, $\langle 0_x|1_x\rangle = 0$, $\langle 0_y|1_y\rangle = 0$.

Trumpam grįžkime prie to, kodėl Blocho sferos reprezentacijoje atsiranda pusiniai kampai $\theta/2$. Kvanticiai 1 kubito loginiai vartai yra visiškai nusakomi unitariniais operatoriais, priklausančiais **SU(2) grupei** (angl. *special unitary group*), kurie atlieka vektoriaus rotaciją 2 kompleksinių dimensijų erdvėje. Tačiau Blocho sferoje vartų efektą iliustruojame kaip vektoriaus rotacijas 3 dimensijų Euklido erdvėje, kurias apibūdina operatoriai, priklausantys **SO(3) grupei** (angl. *special orthogonal group*). Tarp šių grupių yra atsitiktinė sąsaja, ledžianti atlikti SU(2) elemento vizualizaciją, tačiau SU(2) grupė dvigubai dengia SO(3) grupę. Tai reiškia, kad du SU(2) elementus, besiskiriančius tik π faze, galime perteikti tuo pačiu elementu SO(3) grupėje, išsaugodami visą likusią grupės struktūrą. Dėl to rotacija θ kampu Blocho sferoje nusako $\theta/2$ rotaciją kompleksinėje vektorių erdvėje. Turime apeiti Blocho sferą aplink du kartus ($\theta = 4\pi$) siekdam i sugržti į tą pačią būseną.

2.5 Tiesiniai operatoriai ir matricos

Kvanticiai loginiai vartai matematikoje yra nusakomi objektais, kurie vadinami **tiesiniais operatoriais** (angl. *linear operator*). Operatoriai, veikdami vektoriais nusakytas kvantines būsenas, apibūdina šių vektorių transformacijas. Pavyzdžiu, 1 kubito Blocho sferos reprezentacijoje operatoriaus veiksmas yra keisti Blocho vektoriaus orientaciją sferoje. Formaliai, operatorius A (vartojame didžiašias raides žymėti operatoriams) transformuoja vektorių $|v\rangle \in V$ į kitą vektorių $|u\rangle$, priklausantį tai pačiai vektorių erdvėi, $|u\rangle \in V$:

$$A|v\rangle = |u\rangle. \quad (2.52)$$

Operatoriaus veiksmas *ket* vektoriui yra užrašomas panašiai kaip jų tarpusavio sandauga, kurioje operatorius stovi kairėje *ket* pusėje. Toliau pateikiame elementarias operacijas tarp operatorių ir vektorių, operatorių ir operatorių.

Skaičių, dauginantį vektorių, galima visada iškelti už operatoriaus ir vektoriaus:

$$A(z|v\rangle) = zA|v\rangle. \quad (2.53)$$

Visi loginiai vartai kvantinėje kompiuterijoje yra tiesiniai operatoriai. Operatoriaus tiesiškumo savybė jam veikiant bet kokius du (ar daugiau) vektorius:

$$A(|v\rangle + |u\rangle) = A|v\rangle + A|u\rangle. \quad (2.54)$$

Tai parodo, kad operatorius A veikia atskirai kiekvieną vektorių sumoje tiesiniu būdu. Kadangi bet kokį vektorių V erdvėje galima išreikšti pasirinktais baziniais vektoriais, norint nustatyti $A|\psi\rangle$ pakanka žinoti, kaip tiesinis operatorius veikia pasirinktus bazinius vektorius.

Du operatoriai A ir B yra lygūs ($A = B$), jeigu bet kokiam vektoriui galioja sąlygos $|v\rangle \in V$, $A|v\rangle = B|v\rangle$. Dviejų tiesinių operatorių suma nusako kitą tiesinį operatorių $A + B = C$:

$$C|v\rangle = (A + B)|v\rangle = A|v\rangle + B|v\rangle. \quad (2.55)$$

Operatorių sudėtyje eiliškumas nėra svarbus, $A + B = B + A$. Vienas iš paprasčiausių operatorių yra **identitetas** (angl. *identity*), žymimas simboliu I ir dar vadinamas **vienetiniu operatoriumi**. Identitetas, veikdamas vektorių, jo nekeičia $I|v\rangle = |v\rangle$; tai yra analogiška vektoriaus

sandaugai su skaičiumi 1. Taip pat egzistuoja ir nulinis operatorius, vadinsime jį N , $N|v\rangle = 0$. Dviejų operatorių sandauga $AB = D$ nusako kitą operatorių:

$$AB|v\rangle = D|v\rangle. \quad (2.56)$$

Kairėje lygties dalyje operatorius B veikia vektorių $|v\rangle$, toliau operatorius A veikia gautą vektorių $B|v\rangle$. Eilišumas operatorių sandaugoje bendrai yra svarbus, nes kvantinėje kompiuterijoje dažnai aptinkami operatoriai, kuriems $AB \neq BA$. Tik itin specifinėse situacijose galima aptikti, kai eilišumas nėra svarbus, $AB = BA$. Kai dviejų ar daugiau operatorių sandaugos eilišumas nėra svarbus, tokie operatoriai yra vadinti **tarpusavyje komutatyviais** (angl. *commutative operators*). Dviejų operatorių komutatyvumas yra standartiskai užrašomas įdedant juos į skliaustelius, kurios forma išskleidus reiškia:

$$[A, B] = AB - BA. \quad (2.57)$$

Operatoriai A ir B yra komutatyvūs, jeigu $[A, B] = 0$ ir nekomutatyvūs, jeigu $[A, B] \neq 0$. Kita savybė, dviejų operatorių **antikomutatyvumas** (angl. *anticommutative*), yra operacija, apibrėžta lenktiniais skliausteliais:

$$\{A, B\} = AB + BA. \quad (2.58)$$

Du operatoriai yra antikomutatyvieji, jeigu $\{A, B\} = 0$. Sukeitus antikomutatyvius operatorius vietomis skliausteliuose, jų sandaugoje atsiranda minuso ženklas, nes $AB = -BA$. Šios dvi operatorių klasės aptinkamos specifinėse kvantinių skaičiavimų užduotyse, kaip bus aptarta vėliau.

Abstraktus operatorius kvantinėje kompiuterijoje gali būti visada realizuojamas tiesinėje algebroje matricos forma – sugrupuotų skaičių lentele. Tai yra vadintama operatoriaus **vaizdavimas matrica** (angl. *matrix representation*). Abstrakti operatoriaus forma ir matricos forma yra ekvivalentiškos, tad šiuos terminus dažnai vartosime pakaitomis. Verta atsiminti, kad jei norime realizuoti operatorių matricos formą, reikia pasirinkti tam tikrą vektorių erdvę V , kuriame veikia operatorius A , bazinių vektorių rinkinį. Mat operatorius gali būti išreikštas su skirtingais bazinių vektorių rinkiniais. Jeigu nėra nurodoma kitaip, operatoriai standartiskai išreiškiami skaičiuojamųjų vektorių bazėje $\{|0\rangle, |1\rangle\}$.

Išsamiau panaigrinėkime, kaip tiesiniai operatoriai (loginiai vartai) yra išreiškiami matricų forma, kuo išsiskiria tokiai matricų savybės ir jų aritmetika. Matrica, transformuojanti vektorių n dimensijų vektorių erdvę, yra kvadratinės formos kompleksinių skaičių lentelė, turinti $(n \times n)$ elementų (sakoma: n stulpelių ir n eilucių). Kadangi n kubitų vektorių erdvė yra 2^n dimensių, kvantinėje kompiuterijoje paprastai aptinkame tik kvadratinės matricas su lyginiu skaičiumi eilucių ir stulpelių. Pavyzdžiu, visos vieno kubito būsenas nusakančios transformacijos yra išreiškiamos (2×2) dydžio matricomis. Toliau pateikiame pagrindines aritmetines operacijas tarp matricų, iliustracijai naudodami (3×3) matricų dydį.

Matricų sudėtis galima tik tarp tokio paties dydžio matricų. Dviejų operatorių A ir B , išreikštų matricomis, sudėtis:

$$\begin{aligned} A + B &= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \\ &= \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & a_{13} + b_{13} \\ a_{21} + b_{21} & a_{22} + b_{22} & a_{23} + b_{23} \\ a_{31} + b_{31} & a_{32} + b_{32} & a_{33} + b_{33} \end{bmatrix}. \end{aligned} \quad (2.59)$$

Matricų elementus, kurie yra bendrai kompleksiniai skaičiai, čia vadiname a_{ij} ir b_{ij} . Pirmasis simbolis i padeda įvardyti eilutės numerį (skaičiuojant nuo viršaus), antrasis simbolis j – stulpelio

numerj (skaičiuojant iš kairės). Taip rašyti nebūtina, tačiau pravartu siekiant greičiau įvardyti matricos elementus. Norint sudėti dvi matricas A ir B jų stupelių ir eilučių skaičius turi būti vienodos. Sudėties tvarka yra nesvarbi $A + B = B + A$, vadovaujantis minėta abstraktesne operatorių aritmetika. Skaičiaus ir matricos sandauga $zA = Az$ yra apibréžta:

$$z \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} za_{11} & za_{12} & za_{13} \\ za_{21} & za_{22} & za_{23} \\ za_{31} & za_{32} & za_{33} \end{bmatrix}. \quad (2.60)$$

Dviejų matricų sandauga tarpusavyje galima, jeigu pirmosios matricos stupelių skaičius yra lygus antrosios matricos eilučių skaičiui. Kadangi čia susiduriame tik su kvadratinės formos matricomis, tai reiškia, kad matricos sandaugoje turi būti tokio paties dydžio. Sandaugoje gautas naujas darinys yra tokio paties dydžio nauja matrica. Matricų $AB = C$ sandauga atliekama taip:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}. \quad (2.61)$$

Kiekvienas elementas c_{ij} yra atitinkamos A eilutės ir B stupelio elementų sandaugų suma:

$$C = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} & a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} & a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33} \\ a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31} & a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32} & a_{31}b_{13} + a_{32}b_{23} + a_{33}b_{33} \end{bmatrix}. \quad (2.62)$$

Matricų daugyba yra asociatyvi, pavyzdžiui, trijų operatorių sandaugoje $ABC = A(BC) = (AB)C$. Galime pasirinkdami sudauginti pirmiausiai BC arba AB tarpusavyje. Šiame skyriuje minėtos vektorių operacijos: vektoriaus transpozicija, kompleksinė bei ermitinė jungtys yra taip pat naudojamos matricoms. Matricos transpozicija A^T sukeičia eilučių ir stupelio elementus vietomis taip:

$$A^T = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix}. \quad (2.63)$$

Matricų sandaugoje transpozicija sukeičia operatorių eiliškumą, $(AB)^T = B^T A^T$. Matricos kompleksinė jungtis A^* kiekvienam jo elementui atlieka kompleksinę jungtį a_{ij}^* . Ermitinė matricos jungtis taip pat yra kartu atliekama transpozicija ir elementų kompleksinė jungtis $A^\dagger = (A^T)^*$:

$$A^\dagger = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}^\dagger = \begin{bmatrix} a_{11}^* & a_{21}^* & a_{31}^* \\ a_{12}^* & a_{22}^* & a_{32}^* \\ a_{13}^* & a_{23}^* & a_{33}^* \end{bmatrix}. \quad (2.64)$$

Matricų sandaugoje tai sukeičia eiliškumą $(AB)^\dagger = B^\dagger A^\dagger$. Taip pat galima parodyti, kad $(A^\dagger)^\dagger = A$, $(A + B)^\dagger = A^\dagger + B^\dagger$ ir $(zA)^\dagger = z^* A^\dagger$.

Pateikiame keturis kvantinėje kompiuterijoje dažnai aptinkamus operatorius ir jų matricų išraiškas skaičiuojamujų vektorių bazėje $\{|0\rangle, |1\rangle\}$:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (2.65)$$

Šios matricos vadinamos fiziko **Pauli vardu** (angl. *Wolfgang Pauli*) ir sudaro vadinamąjį Pauli bazinių matricų rinkinį $\{I, X, Y, Z\}$. Trys Pauli matricos X, Y, Z yra tarpusavyje nekomutatyvios, bet kurių dviejų Pauli matricų sandauga gražina trečiąjį:

$$XY = iZ, \quad XZ = -iY, \quad YZ = iX; \quad (2.66)$$

$$YX = -iZ, \quad ZX = iY, \quad ZZ = -iX. \quad (2.67)$$

Tai galime lengvai patikrinti, pavyzdžiu XY ir YX :

$$XY = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = \begin{bmatrix} 0 \cdot 0 + 1 \cdot i & 0 \cdot 0 + 1 \cdot 0 \\ 1 \cdot 0 + 0 \cdot i & 1 \cdot (-i) + 0 \cdot 0 \end{bmatrix} = i \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = iZ; \quad (2.68)$$

$$YX = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 \cdot 0 + 1 \cdot (-i) & 0 \cdot 1 + 0 \cdot (-i) \\ i \cdot 0 + 0 \cdot 1 & i \cdot 1 + 0 \cdot 0 \end{bmatrix} = i \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = -iZ. \quad (2.69)$$

Akivaizdu, kad X, Y, Z Pauli operatoriai yra tarpusavyje antikomutatyvūs.

Grįžkime prie vektorių veikimo operatoriumi, norėdami parodyti, kokios operacijos yra matematiškai apibrėžtos. Tą galime pavaizduoti matricų reprezentacija, nes vektorius gali būti taip pat formaliai vadinamas matrica. Pavyzdžiu, vieno kubito *ket* vektorius $|v\rangle$ yra išreiškiamas (2×1) dydžio matrica (vektorius stulpelis), tiesinėje algebroje turinčia dvi eilutes ir vieną stulpelį. O štai *bra* vektorius $\langle v|$ yra išreiškiamas (1×2) matrica (vektorius eilutė), turinčia vieną eilutę ir du stulpelius.

Prisimenant matricų daugybą, leidžiama dauginti dvi $(n \times n)$, $(n \times 1)$ dydžio matricas; tai atlikę gauname naują $(n \times 1)$ dydžio matricą, o tiksliau – vektorių stulpelį (*ket*). Tačiau šią dviejų matricų atvirkštinė daugyba $(n \times 1)(n \times n)$ nėra galima, nes stulpelių skaičius pirmojoje matricoje ir eilucių skaičius antrojoje neatitinka. Tai parodo, kad negalimas *ket* vektoriaus veikimas operatoriumi; jeigu *ket* stovi operatoriaus kairėje, $|v\rangle A$ yra neapibrėžta operacija. Teisingas eiliškumas yra $A|v\rangle$. *Bra* vektoriaus veikimas operatoriumi atliekamas būtent *bra* esant operatoriaus kairėje $\langle v|A$. Taip gaunamas kitas *bra* vektorius, nes $(1 \times n)(n \times n) = (1 \times n)$. Šioje situacijoje neteisingas eiliškumas būtų $A\langle v|$. Iliustracijai imkime $X|0\rangle$ ir $\langle 0|X$:

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \cdot 1 + 1 \cdot 0 \\ 1 \cdot 1 + 0 \cdot 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle; \quad (2.70)$$

$$\langle 0|X = [1 \ 0] \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = [1 \cdot 0 + 0 \cdot 1 + 1 \cdot 1 + 0 \cdot 0] = [0 \ 1] = \langle 1|. \quad (2.71)$$

Pamename, kad $|v\rangle^\dagger = \langle v|$ bei $(AB)^\dagger = B^\dagger A^\dagger$. Iš to gauname, kad $(A|v\rangle)^\dagger = \langle v|A^\dagger$. Todėl dualioje erdvėje $A|v\rangle$ operacijos atitinkmuo yra ermitinė A^\dagger operatoriaus jungtis, veikianti *bra* vektorių iš dešinės $\langle v|A^\dagger$. Viršuje X operatorius turi tokią savybę – jo ermitinė jungtis yra lygi jam pačiam, tad $X = X^\dagger$.

Taip pat atkreipime dėmesį į tokio tipo operaciją:

$$\langle u|A|v\rangle = a_{uv}. \quad (2.72)$$

Dėl daugybos asociatyvumo šią operaciją galime atliliki dviem būdais: $\langle u|A|v\rangle = \langle u|(A|v)\rangle = (\langle u|A)|v\rangle$. Naryje $\langle u|(A|v)\rangle$ operatorius A veikia vektorių $|v\rangle$ iš kairės, grąžindamas bendrai kitą vektorių, sakykime, $A|v\rangle = |g\rangle$. Iš to gaunamas vidinės sandaugos $\langle u|g\rangle$ rezultatas – skaičius a_{uv} . Antruoju būdu naryje $(\langle u|A)|v\rangle$ pirmiausiai atliekama $\langle u|A$ ir toliau – vidinė sandauga su $|v\rangle$.

2.6 Unitariniai ir ermitiniai operatoriai

Unitariniai operatoriai yra plačiausiai naudojama ir aptinkama operatorių klasė kvantinėje kompiuterijoje. Šie operatoriai matematiškai apibūdina kvantinių loginių vartų efektą – jie transformuoja vektoriai nusakomas kubitų būsenas į kitas būsenas. Unitarinis operatorius U pasižymi savybe šia savybe:

$$U^\dagger U = UU^\dagger = I. \quad (2.73)$$

Tai parodo, kad unitarinio operatoriaus sandauga su jo ermitine jungtimi U^\dagger yra lygi vienetiniam operatoriui. Vadinas, U^\dagger yra atvirkštinis U operatoriui (rašoma $U^\dagger = U^{-1}$), ir todėl galima visada atstatyti pradinį vektorių: $U^\dagger U|v\rangle = I|v\rangle = |v\rangle$. Unitariniai operatoriai turi kitą svarbią savybę – veikdami vektorių $|v\rangle$ transformuoja jį į kitą vektorių, tačiau nekeičia vektoriaus ilgio. Dėl šios priežasties veikiantys du skirtinges vektorius unitariniai operatoriai išsaugo jų vidinės sandaugos reikšmę. Imkime du, nebūtinai ortogonalius vektorius $|v\rangle$ ir $|u\rangle$. Minėtas U savybes galime paprastai parodyti:

$$\langle u|U^\dagger U|v\rangle = \langle u|I|v\rangle = \langle u|v\rangle. \quad (2.74)$$

Jeigu $|v\rangle$ yra normuotasis vektorius, U išlaiko normuotumą $\langle v|U^\dagger U|v\rangle = 1$. Tad unitariniai operatorių efektas gali būti apibūdintas kaip vektorių posūkis vektorių erdvėje. Jeigu U veikia visus bazinius vektorius jų rinkinyje, tada naujai gauti vektoriai nusako kitą **transformuotą bazinių vektorių rinkinį** (angl. *basis transformation*).

Kita svarbi operatorių klasė kvantinėje kompiuterijoje yra **ermitiniai operatoriai** (angl. *hermitian operator*). Operatorių **tikrinių vektorių** (angl. *eigenvector*) ir **tikrinių verčių** (angl. *eigenvalue*) konceptai leidžia geriau suprasti ermitinių ir unitarinių operatorių savybes bei jų varojimą. Vektorius $|\lambda\rangle$ yra vadinamas tiesinio operatoriaus K tikriniu vektoriumi, o λ yra su šiuo vektoriumi susieta tikrinė vertė (skaicius), jeigu tenkinama lygybė:

$$K|\lambda\rangle = \lambda|\lambda\rangle. \quad (2.75)$$

Atkreipiame dėmesį, kad bendroje situacijoje operatorius, veikdamas vektorių, pakeičia jį į kitą vektorių. Tačiau jeigu $|\lambda\rangle$ yra K operatoriaus vienas iš tikrinių vektorių, tada operatoriaus veiksmas šiam vektoriui yra lygus vektoriaus $|\lambda\rangle$ ir skaičiaus λ sandaugai – pats vektorius nepakinta.

Operatoriaus tikrinės vertės λ tenkina charakteristinę determinantinę lygtį $\det(K - \lambda I) = 0$. Charakteristinė lygtis yra bendrai n laipsnio polinomas $p(\lambda) = 0$, todėl egzistuoja n skaičius tikrinių verčių $(\lambda_1, \lambda_2, \dots, \lambda_n)$, tenkinančių šią lygtį. Tikriniai vektoriai $|\lambda_i\rangle$ yra randami antrame žingsnyje iš lygties $(K - \lambda_i I)|\lambda_i\rangle = 0$, žinant su jais asocijuotas tikrines vertes λ_i .

Ermitinių operatorių tikrinės vertės λ_i yra visada realieji skaičiai. Tai galima lengvai parodyti pradedant nuo $K|\lambda\rangle = \lambda|\lambda\rangle$ ir ižvertinant šios lygties ermitinę jungtį $(K|\lambda\rangle)^\dagger = \langle \lambda|K^\dagger = \langle \lambda|K = \langle \lambda|\lambda^*$. Tad:

$$\langle \lambda|K|\lambda\rangle = \lambda^* \langle \lambda|\lambda\rangle = \lambda \langle \lambda|\lambda\rangle. \quad (2.76)$$

Matome, kad $\lambda^* = \lambda$, ir todėl λ gali būti tik realusis skaičius. Su skirtingomis tikrinėmis vertėmis λ_i asocijuoti tikriniai vektoriai $|\lambda_i\rangle$ yra ortogonalieji, $\langle \lambda_i|\lambda_j\rangle = 0$ (jeigu $i \neq j$). Imdami operatoriaus K du tikrinius vektorius $|\lambda_1\rangle$ ir $|\lambda_2\rangle$ bei su jais susietas vertes λ_1 ir λ_2 , galime paprastai parodyti ortogonalumą:

$$\langle \lambda_2|K|\lambda_1\rangle = \lambda_1 \langle \lambda_2|\lambda_1\rangle = \lambda_2 \langle \lambda_2|\lambda_1\rangle \rightarrow (\lambda_1 - \lambda_2) \langle \lambda_2|\lambda_1\rangle = 0. \quad (2.77)$$

Matome, kad jeigu $\lambda_2 \neq \lambda_1$, tada $\langle \lambda_2|\lambda_1\rangle = 0$. Ermitinių operatorių tikrinių verčių realumas ir tikrinių vektorių ortogonalumas yra naudojamas kvantinėje mechanikoje nusakyti stebimiems

fizikiniams dydžiams. Siekdami išsamiai aprašyti sistemą reikalaujame, kad ortogonaliuji tikriniai vektorių skaičius atitinktų visų fiziškai skirtinį kvantinių būsenų skaičių. Jeigu turime K ermitinę matricą ($d \times d$) dydžio, operuojančią d dimensijų kompleksinių vektorių erdvėje, tada galime būti tikri, kad rasime d tikriniai K matricos vektorių. Ortogonalieji ir normuotieji K operatoriaus tikriniai vektoriai gali būti naudojami kaip baziniai vektoriai toje vektorių erdvėje apibūdinti būsenoms, o fizikiniai dydžiai yra natūraliai nusakomi tikrinėmis vertėmis – realiaisiais skaičiais.

Vienalaikio diagonalizavimo teorema (angl. *simultaneous diagonalization theorem*) nusako, kad jeigu du operatoriai yra komutatyvūs $[A, B] = 0$, tada šie operatoriai dalijasi tais pačiais tikriniais vektoriais (su galimai skirtinomis tikrinėmis vertėmis). Mat pasitaiko atvejų, kai du ar daugiau operatoriaus tikriniai vektoriai yra susieti su ta pačia tikrine vertė λ . Tokie vektoriai yra lietuviškai vadinami **išsigimusiais** (angl. *degenerate*). Išsigimusios tiesiskai nepriklausomus vektorius visada galima užrašyti forma, kurioje jie yra vienas kitam ortogonalūs. Tad n skaičius išsigimusų tikriniai vektoriai su ta pačia tikrine vertė λ apibūdina n dimensijų išsigimusų poerdvį. Kvatinės mechanikos praktikoje dažnai galima rasti papildomą operatorių (ar operatorius), kuris taip pat dalijasi tais pačiais tikriniais vektoriais (yra komutatyvus), tačiau turi skirtinas tikrines vertes šiemis vektoriams. Kartu jie leidžia panaikinti išsigimimą ir taip unikaliai atskirti fiziškai skirtinges būsenas.

Kitaip nei ermitinių operatorių, unitarinių operatorių tikrinės vertės λ gali būti kompleksiniai skaičiai. Jų tikriniai verčių modulis yra visada lygus vienetui, $|\lambda| = 1$. Todėl unitarinių operatorių tikrinės vertės turi bendrą formą $\lambda = e^{i\theta}$; θ yra realusis skaičius.

Kvantineje kompiuterijoje dažnai aptinkami X , Y ir Z yra ermitiniai operatoriai ($X = X^\dagger$, $Y = Y^\dagger$, $Z = Z^\dagger$), bet tuo pačiu ir unitariniai (nes, pavyzdžiui, $ZZ^\dagger = I$). Dėl to jie gali atlikti loginių vartų vaidmenį ir kartu apibūdinti fizines sistemos stebimas savybes. Vieno kubito baziniai vektoriai $|0\rangle$ ir $|1\rangle$ buvo pasirinkti neatsitiktinai, jie yra Pauli- Z operatoriaus tikriniai vektoriai. Pauli- Z operatoriaus tikrines vertes λ_0 ir λ_1 , susietas su $|0\rangle$ ir $|1\rangle$, galime patikrinti taip:

$$\lambda_0 = \langle 0 | Z | 0 \rangle = 1 \langle 0 | 0 \rangle = 1, \quad \lambda_1 = \langle 1 | Z | 1 \rangle = -1 \langle 1 | 1 \rangle = -1. \quad (2.78)$$

Akivaizdu, kad Pauli- Z tikriniai vektoriai $|0\rangle$ ir $|1\rangle$ nėra X ir Y operatorių tikriniai vektoriai. Ši dėsningumą pirmiausia matome iš to, kad X ir Y operatoriai veikdami $|0\rangle$ ir $|1\rangle$ pakeičia šiuos vektorius, pavyzdžiui, $X|0\rangle = |1\rangle$. Formaliai, Pauli X , Y , Z operatoriai yra tarpusavyje nekomutatyvūs ir todėl neturi bendrų tikriniai vektorių. Pauli- X ir Y operatorių tikriniai vektoriai yra jau minėti:

$$|0_x\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad |1_x\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}; \quad (2.79)$$

$$|0_y\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}, \quad |1_y\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}. \quad (2.80)$$

Pauli- X ir Y operatorių tikrinės vertės yra taip pat 1 ir -1, pavyzdžiui, $\lambda_{0x} = \langle 0_x | X | 0_x \rangle = 1$, $\lambda_{1x} = \langle 1_x | X | 1_x \rangle = -1$. Operatoriai, turintys identiškas tikrines vertes, yra **unitariškai ekvivalentiški** (angl. *unitary equivalent*). Tai reiškia, kad jie gali būti transformuojami vienas į kitą panaudojus tam tikrus unitarinus operatorius U , $A = UBU^\dagger$. Ši operacija matricos reprezentacijoje atlieka operatoriaus B perteikimą kitais baziniais vektoriais.

Panašiai kaip baziniai vektoriai leidžia išreikšti bet kuriuos kitus vektorius toje vektorių erdvėje, visos (2×2) dydžio unitarinės matricos U gali būti išreikštinos Pauli matricų $\{I, X, Y, Z\}$ suma

su atitinkamais koeficientais:

$$U = c_I I + c_x X + c_y Y + c_z Z. \quad (2.81)$$

Čia c_I yra realusis skaičius, c_x, c_y, c_z – kompleksiniai skaičiai, kurie kartu tenkina lygybę:

$$|c_I|^2 + |c_x|^2 + |c_y|^2 + |c_z|^2 = 1. \quad (2.82)$$

Unitariniai ir ermitiniai operatoriai priklauso platesnei operatorių grupei, vadinamai **normaliaisiais operatoriais** (angl. *normal operators*). Normalieji operatoriai pasižymi $AA^\dagger = A^\dagger A$, o jų tikriniai vektoriai, susieti su skirtingomis tikrinėmis vertėmis, yra ortogonalieji. Toliau matysime, kad normaliuosius operatorius galima paprastai išreikšti taikant vadinamąjį spektrinę dekompoziciją.

2.7 Diadinė operatorių dekompozicija

Šiame poskyryje pateikiame būdą išreikšti operatoriams taikant *bra* ir *ket* vektorių matematičné konstrukciją, vadinama **išorine sandauga** (angl. *outer product*), dar žinoma kaip **diadinė** (angl. *dyad*) **dekompozicija**. Ši matematiné konstrukcija itin supaprastina abstraktų operatorių ir vektorių veikimo skaičiavimą kvantinéje kompiuterijoje nereikalaujant naudoti matricų reprezentaciją.

Pirmiausiai atkreipiame dėmesį, kad bendrą matricą galima visada išreikšti kitų matricų suma (nebūtinai unikalia). Imkime šį (2×2) matricos pavyzdį:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = a \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + b \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + c \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + d \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \quad (2.83)$$

Kiekviena iš keturių (2×2) matricų yra vektoriaus stulpelio ir vektoriaus eilutės sandaugos rezultatas. Pavyzdžiui, antra matrica viršuje gaunama:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 & 1 \cdot 1 \\ 0 \cdot 0 & 0 \cdot 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} = |1\rangle\langle 0|. \quad (2.84)$$

Tai galima palyginti su vidine vektorių sandauga, kuri yra vektoriaus eilutės ir vektoriaus stulpelio sandauga, o jos rezultatas yra skaičius. Prisiminę, kad vektorius stulpelis nusako *ket*, o eilutė *bra*, matome, kad viršuje pavyzdžyje pateikta išorinė sandauga ir jos matricos išraiška gali būti užrašoma glaustai $|1\rangle\langle 0|$. Čia $|1\rangle$ ir $\langle 0|$ yra kubito Pauli-Z *ket* (*bra*) tikriniai vektoriai. Tad bet kokia (2×2) matrica yra išreiškiama diadine dekompozicija taip:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = a|0\rangle\langle 0| + b|1\rangle\langle 0| + c|0\rangle\langle 1| + d|1\rangle\langle 1|. \quad (2.85)$$

Apžvelkime diadų bendruosius principus ir jų taikymą. Imkime du vektorius $|\nu\rangle$ ir $|\omega\rangle$, priklausantys V vektorių erdvėi. Jų išorinė sandauga užrašoma $|\nu\rangle\langle\omega|$ ir yra operatorius, veikiantis V erdvėje. Galime patikrinti, kad diada yra tiesinis operatorius, veikdami ja vektorius $a|\psi\rangle$ ir $b|\phi\rangle$:

$$\begin{aligned} |\nu\rangle\langle\omega|(a|\psi\rangle + b|\phi\rangle) &= a|\nu\rangle\langle\omega|\psi\rangle + b|\nu\rangle\langle\omega|\phi\rangle = az|\nu\rangle + bc|\nu\rangle \\ &= (az + bc)|\nu\rangle. \end{aligned} \quad (2.86)$$

Diada veikianti vektorių jį transformuoja į kitą vektorių. Matome, kad diada $|\nu\rangle\langle\omega|$ pasuka vektorius $|\psi\rangle$ ir $|\phi\rangle$ į $|\nu\rangle$ vektoriaus kryptį, jeigu jie persikloja su $|\omega\rangle$. Persiklojimą atspindi

skaičius, gautas vidinėje sandaugoje, čia $z = \langle \omega | \psi \rangle$ ir $c = \langle \omega | \phi \rangle$. Dėl daugybos asociatyvumo, $|\nu\rangle(\langle \omega | \psi \rangle) = (|\nu\rangle\langle \omega|)|\psi\rangle$, į $|\nu\rangle\langle \omega | \psi \rangle$ galime žvelgti dviem būdais. Pirmuoju būdu nusakoma vektoriaus ir skaičiaus sandauga, antruoju – vektoriaus veikimas diada. Diada $|\nu\rangle\langle \omega |$ gali operuoti *bra* vektorius iš dešinės, pavyzdžiu, naryje $\langle \psi | \nu \rangle \langle \omega |$. Šiuo atveju vidinė sandauga atliekama su $|\nu\rangle$, o *bra* $\langle \psi |$ būtų pasuktas į $\langle \omega |$.

Diados $D = |\nu\rangle\langle \omega |$ ermitinė jungtis yra:

$$D^\dagger = |\omega\rangle\langle \nu |. \quad (2.87)$$

Vienetinis V vektorių erdvės operatorius I yra išreiškiamas diadomis naudojant šios erdvės bazinių vektorių $\{|m\rangle\}$ rinkinį ir visus juos susumuojant:

$$I = \sum_m |m\rangle\langle m|. \quad (2.88)$$

Atkreipiame dėmesį, kad vienodi *bra* ir *ket* simboliai diadoje nurodo matricos pagrindinės įstrižainės elementus a_{mn} . Visi tiesiniai operatoriai turi diadinę dekompoziciją. Šią dekompoziciją galime formaliai išreikšti įterpę norimą A operatorių tarp vienetinių operatorių ir panaudodami vienetinio operatoriaus diadines dekompozicijas:

$$\begin{aligned} A = & IAI = \sum_{m,n} |m\rangle\langle m|A|n\rangle\langle n| = \sum_{m,n} \langle m|A|n\rangle|m\rangle\langle n| \\ & = \sum_{m,n} a_{mn} |m\rangle\langle n|. \end{aligned} \quad (2.89)$$

Čia $a_{mn} = \langle m|A|n\rangle$ yra jau matytas objektas, jis įvardija a_{mn} matricos elementą m eilutėje ir n stulpelyje. Pauli operatorius ir Hadamardo transformaciją galima paprastai išreikšti diadomis vektorių $\{|0\rangle, |1\rangle\}$ bazėje:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = |0\rangle\langle 1| + |1\rangle\langle 0|; \quad (2.90)$$

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = -i|0\rangle\langle 1| + i|1\rangle\langle 0|; \quad (2.91)$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1|; \quad (2.92)$$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = |0\rangle\langle 0| + |1\rangle\langle 1|; \quad (2.93)$$

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}} (|0\rangle\langle 0| + |1\rangle\langle 0| + |0\rangle\langle 1| - |1\rangle\langle 1|). \quad (2.94)$$

Pavyzdžiui, Hadamardo matricos elementai $h_{01} = 1$, $h_{11} = -1$. Iliustruodami diadų naudojimą, pateikiame $X|1\rangle$ bei $YX|0\rangle$:

$$X|1\rangle = |1\rangle\langle 0|1\rangle + |0\rangle\langle 1|1\rangle = |0\rangle; \quad (2.95)$$

$$\begin{aligned} YX|0\rangle &= (-i|0\rangle\langle 1| + i|1\rangle\langle 0|)(|0\rangle\langle 1| + |1\rangle\langle 0|)|0\rangle \\ &= -i|0\rangle\langle 0|0\rangle + i|1\rangle\langle 1|0\rangle = -i|0\rangle. \end{aligned} \quad (2.96)$$

Taikant bazinių vektorių ortogonalumą, vidinės sandaugos $\langle 0|0\rangle = 1$, $\langle 1|0\rangle = 0$. Atkreipiame dėmesį, kad čia svarbus operatorių veikimo eiliškumas, kadangi Pauli operatoriai yra tarpusavyje nekomutatyvūs.

Normaliuosius operatorius ($AA^\dagger = A^\dagger A$), kuriems priklauso unitariniai ir ermitiniai operatoriai, galima visada išreikšti vadinamąja **spektrine dekompozicija** (angl. *spectral operator decomposition*):

$$A = \sum_k \lambda_k |k\rangle\langle k| = \sum_k \lambda_k P_k. \quad (2.97)$$

Skaičius λ_k yra operatoriaus A tikrinė vertė, asocijuota su tikriniu vektoriumi $|k\rangle$. Tikrinių verčių rinkinys $\{\lambda_k\}$ yra operatoriaus A spektras. Čia $P_k = |k\rangle\langle k|$ nusako svarbią, vadinamąją **projekcinių operatorių** (angl. *projection operator*) klasę. Jie atlieka projekciją į poerdvį, susietą su tikrine verte λ_k , kurj dengia su ja susieti tikriniai vektoriai $|k\rangle$. Tai matome iš išraiškos:

$$P_k|v\rangle = |k\rangle\langle k|v\rangle = \langle k|v\rangle|k\rangle = a|k\rangle. \quad (2.98)$$

Čia $a = \langle k|v\rangle$ nusako persiklojimą $|v\rangle$ vektoriaus išilgai $|k\rangle$, kitaip tariant – jo projekciją. Jeigu sistemoje nėra $|k\rangle$ tikrinių vektorių išsigimimo, tada šis poerdvis yra 1 dimensijos, dengiamas $|k\rangle$ vektoriaus. Kitos projekcinių operatorių savybės yra:

$$\sum_k P_k = I; \quad (2.99)$$

$$P_i P_j = P_i \delta_{ij}. \quad (2.100)$$

Pirmai savybė, vadinamoji **pilnumo lygtis** (angl. *completeness equation*) teigia, kad susumavę visus vektorių erdvės projekcinius operatorius gausime vienetinį operatorių I . Antroji savybė rodo, kad dvių skirtingų projekcinių operatorių sandauga yra nulinė ($i \neq j$), arba lygia pačiam operatoriui ($i = j$). Taip yra todėl, kad taip apibūdinti skirtingi P_k atlieka vektorių projekcijas į ortogonaliuosius poerdvius.

Verta paminėti, kad unitariniai operatoriai U turi spektrinę dekompoziciją, kurios forma yra:

$$U = \sum_k e^{i\varphi_k} |k\rangle\langle k|. \quad (2.101)$$

Čia φ_k yra realusis skaičius ir, kaip minėta, tikrinės vertės yra fazės faktoriai $\lambda_k = e^{i\varphi_k}$.

Matome, kad projekcinių operatorių dekompozicijoje yra tik matricos pagrindinės įstrižainės elementai (diagonalioji matrica). Tokią formą turi viršuje diandomis išreikštias Pauli- Z operatorius, kadangi dekompozicijoje buvo naudojami jo tikriniai vektoriai $\{|0\rangle, |1\rangle\}$. Identiškai atrodytų ir Pauli- X bei Y operatoriai, jeigu operatorių diadinėje dekompozicijoje išreikštume juos su jų tikriniais vektoriais:

$$X = |0_x\rangle\langle 0_x| - |1_x\rangle\langle 1_x|; \quad (2.102)$$

$$Y = |0_y\rangle\langle 0_y| - |1_y\rangle\langle 1_y|. \quad (2.103)$$

2.8 Matricos pēdsakas

Matricos **pēdsakas** (angl. *trace*) yra operacija, kuri sudeda visus matricos pagrindinės įstrižainės elementus $a_{mm} = \langle m|A|m\rangle$:

$$\text{Tr}(A) = \sum_m \langle m|A|m\rangle = \sum_m a_{mm}. \quad (2.104)$$

Operatoriaus pēdsakas yra apibrežtas kaip šio operatoriaus matricos reprezentacijos pēdsakas. Matricos pēdsakas nepriklauso nuo to, su kokiais baziniais vektoriais išreiškiamas operatorius.

Įterpdami vienetinį operatorių $I = \sum_i |i\rangle\langle i|$ išreikštą $\{|i\rangle\}$ vektorių bazėje į išraišką viršuje gauname:

$$\begin{aligned}\text{Tr}(A) &= \sum_m \langle m|A|m\rangle = \sum_{m,i} \langle m|i\rangle\langle i|A|m\rangle \\ &= \sum_{m,i} \langle i|A|m\rangle\langle m|i\rangle = \sum_i \langle i|A|i\rangle.\end{aligned}\quad (2.105)$$

Visi trys Pauli operatoriai X, Y, Z turi nulinį pėdsaką. Pavyzdžiui:

$$\text{Tr}(Y) = \text{Tr} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = 0 + 0 = 0; \quad (2.106)$$

$$\text{Tr}(Z) = \text{Tr} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = 1 + (-1) = 0. \quad (2.107)$$

Matricos pėdsakas turi šias savybes:

$$\text{Tr}(A + B) = \text{Tr}(A) + \text{Tr}(B); \quad (2.108)$$

$$\text{Tr}(zA) = z\text{Tr}(A); \quad (2.109)$$

$$\text{Tr}(AB) = \text{Tr}(BA). \quad (2.110)$$

Pirmoje eilutėje užrašyta savybė yra vadinamas tiesiškumas; antroje eilutėje z – skaičius. Trečia savybė nusako matricos pėdsako cikliškumą. Pavyzdžiui, $\text{Tr}(ABC) = \text{Tr}(BCA) = \text{Tr}(CAB)$. Taip pat galima lengvai patikrinti, kad diados $|m\rangle\langle n|$ pėdsakas yra $\text{Tr}(|m\rangle\langle n|) = \langle n|m\rangle = \delta_{nm}$.

2.9 Tenzorinė vektorių sandauga

Norint apibūdinti sudėtinę kvantinę sistemą, sudarytą iš daugiau nei vieno kubito, pasitelkdami tenzorinę sandaugą formaliai konstruojame didesnę jų būsenas talpinančią vektorių erdvę. Imkime kaip pavyzdži du kubitus. Kiekvienas iš jų individualiai yra nusakytas identiškose 2 dimensijų kompleksinėse vektorių erdvėse, vadinsime jas V ir U . Sistema, sudaryta iš dviejų kubitų, nusakoma erdvėje, kuri yra tenzorinė šių vektorių erdviių sandauga $V \otimes U$, žymima ženklu \otimes . Šios vektorių erdvės dimensija d yra V ir U dimensijų sandauga, $d = 2 \times 2 = 4$. Bendrai n kubitų erdvė nusakoma n skaičiumi 2 dimensijų sandaugomis, tai yra 2^n dimensijų erdvė.

$V \otimes U$ erdvės elementai yra individualių erdviių vektorių tenzorinės sandaugos, $|v\rangle \otimes |u\rangle$, ir jų tiesinės kombinacijos. Galime 2 kubitų vektorius išreikšti tiesinėje algebroje, imkime:

$$|v\rangle = \begin{bmatrix} a \\ b \end{bmatrix}, \quad |u\rangle = \begin{bmatrix} c \\ d \end{bmatrix}. \quad (2.111)$$

Tada jų tenzorinė sandauga $|v\rangle \otimes |u\rangle$ yra:

$$|v\rangle \otimes |u\rangle = \begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} a & \begin{bmatrix} c \\ d \end{bmatrix} \\ b & \begin{bmatrix} c \\ d \end{bmatrix} \end{bmatrix} = \begin{bmatrix} ac \\ ad \\ bc \\ bd \end{bmatrix}. \quad (2.112)$$

Gautas vektorius stulpelis yra sudarytas iš keturių elementų. Naudodami V ir U vektorių erdviių bazinius vektorius galime iš jų suformuoti $V \otimes U$ erdvę dengiančius bazinius vektorius. Tokia erdvė yra 4 dimensijų, tad turėtume rasti keturis bazinius vektorius. Skaičiuojamas ortogonalus

2 kubitų rinkinys yra sudarytas iš V ir U erdviių $\{|1\rangle, |0\rangle\}$ bazinių vektorių tensorinių sandaugų $\{|0\rangle \otimes |0\rangle, |1\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |1\rangle\}$:

$$\begin{aligned} |0\rangle \otimes |0\rangle &= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, & |1\rangle \otimes |0\rangle &= \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \\ |0\rangle \otimes |1\rangle &= \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, & |1\rangle \otimes |1\rangle &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \end{aligned} \quad (2.113)$$

Skaičiuojamas n kubitų bazinių vektorių rinkinys analogiškai bus sudarytas iš visų skirtingu 2^n tensorinių 1 kubito bazinių vektorių sandaugų kombinacijų. Kiekvienas toks vektorius stulpelis turės 2^n elementų, kuriame vienas iš elementų bus 1, o visi likusieji $n - 1$ elementai 0.

Kai nėra rizikos suklaidinti skaičytoją, stengsimės supaprastinti vektorių simboliką praleisdami \otimes ir sujungdami tensorinę vektoriaus išraišką į vieną skaičių eilutę. Pavyzdžiui, 2 kubitų bazinių vektorių rinkinys identiškai rašomas $\{|00\rangle, |10\rangle, |01\rangle, |11\rangle\}$. Literatūroje taip pat galima sutikti naudojamą tensorinės vektorių sandaugos žymėjimą su praleistu tensoriaus ženklu, pavyzdžiui, $|\psi\rangle|\varphi\rangle = |\psi\rangle \otimes |\varphi\rangle$. Šioje knygoje vartojame tik du būdus – įterpdami \otimes tarp vektorių arba sujungdami juos vienu vektoriaus simboliu.

Pateikiame pagrindines aritmetines operacijas su vektoriais išreikštais tensorinėmis sandaugomis. Sandauga su skaičiumi:

$$z(|\psi\rangle \otimes |\phi\rangle) = (z|\psi\rangle) \otimes |\phi\rangle = |\psi\rangle \otimes (z|\phi\rangle). \quad (2.114)$$

Tensorinė dviejų vektorių sandauga, kai vienas vektorius yra išreikštas kitų dviejų sudėtimi (superpozicijoje):

$$(|\psi\rangle + |u\rangle) \otimes |\phi\rangle = |\psi\rangle \otimes |\phi\rangle + |u\rangle \otimes |\phi\rangle. \quad (2.115)$$

Dualusis tensorinis vektorius (*bra*) yra formuojamas taip:

$$(|\psi\rangle \otimes |\phi\rangle)^\dagger = \langle \psi | \otimes \langle \phi |. \quad (2.116)$$

Naudojame susitarimą, kuriamo išlaikomas simboliai ($\phi, \psi, u \dots$) eiliukumas ir tik pakeiciama skliaustelių forma iš *ket* į *bra*. Atliekant įvairias operacijas yra svarbu sekti, kuriai vektorių erdvėi priskirtas posistemės vektorius tensorinėje sandaugoje. Mat vidinė vektorių sandauga yra apibrėžta tik tarp tos pačios vektorių erdvės elementų. Imkime dvi tensorines dviejų kubitų būsenas $|\psi\rangle \otimes |\phi\rangle$ ir $|u\rangle \otimes |u\rangle$, kai $|\psi\rangle, |v\rangle \in V$ bei $|\phi\rangle, |u\rangle \in U$. Vidinė sandauga atliekama tarp vektorių, priklausančių tai pačiai posistemei:

$$(\langle u | \otimes \langle \omega |)(|\psi\rangle \otimes |\phi\rangle) = \langle u | \psi \rangle \langle \omega | \phi \rangle. \quad (2.117)$$

Rezultatas bus vėlgi bendrai kompleksinis skaičius, kuris yra dviejų skaičių sandauga $\langle u | \psi \rangle \langle \omega | \phi \rangle = cz$, gauta iš šių dviejų vidinių sandaugų: $\langle u | \psi \rangle = c$, $\langle \omega | \phi \rangle = z$.

2.10 Tensorinė operatorių sandauga

Operatoriai, gauti iš 1 kubito operatorių tensorinių sandaugų, formaliai leidžia apibūdinti didesnės kubitų sistemos transformacijas. Dviejų 1 kubito operatorių tensorinę sandaugą $A \otimes B$,

veikiančią $V \otimes U$ vektorių erdvėje, galime išreikšti matricos forma taip:

$$A \otimes B = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \otimes \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} aB & bB \\ cB & dB \end{bmatrix} = \begin{bmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{bmatrix}. \quad (2.118)$$

Atkreipiame dėmesį į blokinę struktūrą bei (4×4) matricos dydžių. Tokio matricos dydžio ir tikimės iš operatoriaus, veikiančio 4 dimensijų erdvėje, nusakančioje dviejų kubitų būsenas. Iš unitariųjų operatorių A ir B sukonstruotas operatorius $A \otimes B$ yra taip pat unitarusis. Tai galima parodyti formaliai prisimenant, kad unitariojo operatoriaus ir jo ermitinės jungties sandauga yra vienetinis operatorius:

$$(A \otimes B)(A \otimes B)^\dagger = (A \otimes B)(A^\dagger \otimes B^\dagger) = AA^\dagger \otimes BB^\dagger = I \otimes I = I. \quad (2.119)$$

Kitos operatorių, išreikštų tensorinėmis sandaugomis, savybės:

$$A \otimes (B + C) = A \otimes B + A \otimes C; \quad (2.120)$$

$$(A \otimes B)(C \otimes D) = (AC \otimes BD); \quad (2.121)$$

$$\text{Tr}(A \otimes B) = \text{Tr}(A)\text{Tr}(B). \quad (2.122)$$

Iliustruodami tensorinės operatorių sandaugos naudojimą, imkime dviejų kubitų sistemą, kuriai atliekama pirmam kubitui Pauli- X , o antram Pauli- Z transformacijos:

$$X \otimes Z(|v\rangle \otimes |u\rangle) = X|v\rangle \otimes Z|u\rangle. \quad (2.123)$$

Tai yra tiesinis operatorius, o X ir Z veikia atitinkamo kubito vektorių erdvėje. Šią operaciją galime perteikti ir naudodami tiesinę algebrą:

$$\begin{aligned} X \otimes Z(|0\rangle \otimes |1\rangle) &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ -1 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}. \end{aligned} \quad (2.124)$$

Arba, pirmiau išskleidę tensorines sandaugas, randame tą patį:

$$\begin{aligned} X \otimes Z(|0\rangle \otimes |1\rangle) &= \begin{bmatrix} 0 \cdot Z & 1 \cdot Z \\ 1 \cdot Z & 0 \cdot Z \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}. \end{aligned} \quad (2.125)$$

Galime parodyti, kad operatorių $A \otimes B$, veikiančių $V \otimes U$ erdvėje, tikrinės vertės yra individualių operatorių A ir B tikrinės verčių sandaugos. Imkime $A|k\rangle = \lambda_k|k\rangle$ ir $B|l\rangle = \lambda_l|l\rangle$, kai λ_k ir λ_l yra atitinkamų operatorių tikrinės vertės, asociuotos su jų tikriniais vektoriais $|k\rangle$ bei $|l\rangle$. Tada:

$$(A \otimes B)(|k\rangle \otimes |l\rangle) = A|k\rangle \otimes B|l\rangle = \lambda_k|k\rangle \otimes \lambda_l|l\rangle = \lambda_k \lambda_l (|k\rangle \otimes |l\rangle). \quad (2.126)$$

Matome, kad $A \otimes B$ operatoriaus tikriniai vektoriai yra $\{|k\rangle \otimes |l\rangle\}$ vektorių rinkinys, o tikrinės vertės $\{\lambda_k \lambda_l\}$. Jeigu A yra $(n \times n)$ matrica, o B $(p \times p)$, kai n ir p gali būti vienodi arba skirtinti, tada egzistuoja np -skaičius tikriniai vektoriai ir $\{|k\rangle \otimes |l\rangle\}$ rinkinys pateikia juos visus. Šie vektoriai yra tarpusavyje ortogonalūs:

$$(\langle k' | \otimes \langle l' |)(|k\rangle \otimes |l\rangle) = \langle k' | k \rangle \langle l' | l \rangle = \delta_{k'k} \delta_{l'l}, \quad (2.127)$$

ir todėl gali būti naudojami kaip bazinių vektorių rinkinys. Toliau imkime normaluojti operatorių A , išreikštą spektrine dekompozicija:

$$A = \sum_k \lambda_k |k\rangle \langle k| = \sum_k \lambda_k P_k. \quad (2.128)$$

Tenzorinė dviejų normaliujujų operatorių sandauga $A \otimes B$ taip pat gali būti išreiškiama naudojant šių dviejų operatorių projekcinius operatorius:

$$\begin{aligned} A \otimes B &= \sum_k \lambda_k |k\rangle \langle k| \otimes \sum_l \lambda_l |l\rangle \langle l| = \sum_{k,l} \lambda_k \lambda_l |k\rangle \langle k| \otimes |l\rangle \langle l| \\ &\equiv \sum_m \lambda_m |m\rangle \langle m|. \end{aligned} \quad (2.129)$$

Tai nusako diagonaliąjį matricą, o $\{|kl\rangle \equiv |k\rangle \otimes |l\rangle \equiv |m\rangle\}$ yra bazinių vektorių rinkinys, dengiantis $V \otimes U$ erdvę. Čia $A \otimes B$ operatoriaus tikrinės vertės, asocijuotos su tikriniais vektoriais $\{|m\rangle\}$, yra A ir B operatorių tikriniai verčių sandaugos, $\lambda_m = \lambda_k \lambda_l$. Pavyzdžiui, Pauli- Z operatorių tenzorinė sandauga taikant spektrinę dekompoziciją išreiškiama:

$$\begin{aligned} Z \otimes Z &= |0\rangle \langle 0| \otimes |0\rangle \langle 0| - |1\rangle \langle 1| \otimes |0\rangle \langle 0| - |0\rangle \langle 0| \otimes |1\rangle \langle 1| + |1\rangle \langle 1| \otimes |1\rangle \langle 1| \\ &= |00\rangle \langle 00| - |10\rangle \langle 10| - |01\rangle \langle 01| + |11\rangle \langle 11| \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (2.130)$$

Matome, kad yra tik dvi skirtinės operatoriaus $Z \otimes Z$ tikrinės vertės, $\lambda_k \lambda_l \in (1, -1)$, ir keturi tikriniai vektoriai. Tad su šiomis tikrinėmis vertėmis susieti poerdviai yra dvigubai išsigimė.

Kvantinėje kompiuterijoje dažnai aptinkami operatoriai, kurie vienu metu keičia tik vieno kubito būseną. Pavyzdžiui, operatorius trečiam kubitui 4 kubitų registre, atliekantis Pauli- Y vartus, nekeičiant kitų, yra išreiškiamas $I \otimes I \otimes Y \otimes I$. Operatoriai, turintys formą $A \otimes B \otimes C \otimes D \dots$, yra vadinami lokalaisiais, kadangi jie atlieka operacijas su atskirais kubitais nepriklausomai nuo kitų kubitų būsenos. Bendresnio pobūdžio, vadinamosios nelokaliosios transformacijos, yra nusakomos lokalijų operatorių sumomis, pavyzdžiui, veikiančios du kubitus $A \otimes B + C \otimes D$.

2.11 Operatorių funkcijos

Kvantinėje kompiuterijoje dažnai sutinkamos transformacijos, nusakomos matricų funkcijomis. Kitaip nei įprastinės funkcijos, kurių reikšmės bei vertės yra skaičiai, matricų funkcijų reikšmės bei vertės yra matricos. Laimei, kvantinėje kompiuterijoje visos matricos, naudojamos atlikti būsenų transformacijoms, priklauso normaliujujų operatorių klasei. Šiuos operatorius galime perteikti spektrine dekompozicija:

$$A = \sum_k \lambda_k |k\rangle \langle k| = \sum_k \lambda_k P_k. \quad (2.131)$$

Analitinės normaliųjų operatorių funkcijos $f(A)$ tada randamos paprasta žinoma formule:

$$f(A) = \sum_k f(\lambda_k) P_k. \quad (2.132)$$

Operatoriaus funkcija f yra įvertinama imant operatoriaus A tikrines vertes λ_k , kaip jos reikšmes, kurios daugina atitinkamus projekcinius operatorius P_k . Pavyzdžiui, dažnai algoritmuose aptinkama ermitinio operatoriaus A eksponentė, $f(\alpha A) = e^{i\alpha A}$, kai α – realusis skaičius:

$$e^{i\alpha A} = \sum_k e^{i\alpha \lambda_k} P_k. \quad (2.133)$$

Šiuo principu galime rasti, pavyzdžiui, Pauli- Z operatoriaus ($\lambda_0 = 1$, $\lambda_1 = -1$) eksponentę $f(\alpha A) = e^{i\alpha Z}$:

$$e^{i\alpha Z} = e^{i\alpha} |0\rangle\langle 0| + e^{-i\alpha} |1\rangle\langle 1| = \begin{bmatrix} e^{i\alpha} & 0 \\ 0 & e^{-i\alpha} \end{bmatrix}. \quad (2.134)$$

Kitas būdas rasti matricos eksponentę yra taikant **Teiloro eilutę** (angl. *Taylor series*):

$$e^{i\alpha Z} = \sum_{j=0}^{\infty} \frac{(i\alpha Z)^j}{j!}. \quad (2.135)$$

Atkreipiame dėmesį, kad Pauli operatoriai pasižymi savybe $Z^j = Z$, kai j – nelyginis skaičius, ir $Z^j = I$, kai j – lyginis skaičius. Sugrupavę lyginius bei nelyginius narius ir panaudodami kosinusų bei sinusų Teiloro eilutes randame:

$$\begin{aligned} e^{i\alpha Z} &= \left(I - \frac{\alpha^2 I}{2!} + \frac{\alpha^4 I}{4!} - \dots \right) + \left(i\alpha Z - \frac{i\alpha^3 Z}{3!} + \frac{i\alpha^5 Z}{5!} - \dots \right) \\ &= \cos(\alpha) I + i \sin(\alpha) Z. \end{aligned} \quad (2.136)$$

Perteikdami I ir Z matricų formą bei taikydami Oilerio formulę prieiname prie tos pačios išraiškos:

$$\begin{aligned} \cos(\alpha) I + i \sin(\alpha) Z &= \begin{bmatrix} \cos(\alpha) + i \sin(\alpha) & 0 \\ 0 & \cos(\alpha) - i \sin(\alpha) \end{bmatrix} \\ &= \begin{bmatrix} e^{i\alpha} & 0 \\ 0 & e^{-i\alpha} \end{bmatrix}. \end{aligned} \quad (2.137)$$

Taikydami spektrinę dekompoziciją galime taip pat rasti ir normaliųjų operatorių tensorinės sandaugos funkcijas. Pavyzdžiui, dviejų operatorių $A \otimes B$ funkcija:

$$f(A \otimes B) = \sum_{k,l} f(\lambda_k \lambda_l) P_k \otimes P_l. \quad (2.138)$$

Funkcijos f argumentai yra P_k ir P_l operatorius atitinkančių tikrinių verčių sandaugos $\lambda_k \lambda_l$. Imkime Pauli- Z operatorių eksponentę $f(A \otimes B) = e^{i\alpha Z \otimes Z}$:

$$\begin{aligned} e^{i\alpha Z \otimes Z} &= e^{i\alpha} |0\rangle\langle 0| \otimes |0\rangle\langle 0| + e^{-i\alpha} |1\rangle\langle 1| \otimes |0\rangle\langle 0| \\ &\quad + e^{-i\alpha} |0\rangle\langle 0| \otimes |1\rangle\langle 1| + e^{i\alpha} |1\rangle\langle 1| \otimes |1\rangle\langle 1| \\ &= \begin{bmatrix} e^{i\alpha} & 0 & 0 & 0 \\ 0 & e^{-i\alpha} & 0 & 0 \\ 0 & 0 & e^{-i\alpha} & 0 \\ 0 & 0 & 0 & e^{i\alpha} \end{bmatrix}. \end{aligned} \quad (2.139)$$

III skyrius

Kvantinės mechanikos pagrindai

Kvantinė mechanika yra šiuo metu bendriausia fizikinė teorija ir iš princiopo pritaikoma nuo subatominio lygio iki kasdieninių daiktų skalės. Siekiant paaiškinti kvantinius reiškinius ši teorija buvo suformuluota naudojant tiesinės algebrros matematinę struktūrą, kurios elementus pristatėme ankstesniame skyriuje. Šiame skyriuje pateikiame kvantinės mechanikos postulatus. Postulatai yra taikomi siekiant sujungti abstraktų šios teorijos matematinį formalizmą su stebimu fiziniu pasauly ir suteikia pagrindines taisykles, kuriomis remiantis perteikiami skaičiavimai kvantiniame kompiuteryje.

3.1 Kvantinės mechanikos postulatai

I postulatas – kvantinės būsenos. Uždaros kvantinės sistemos būsena yra visiškai nusakoma normuotuoju vektoriumi $|\psi\rangle$ Hilberto vektorių erdvėje.

Kvantinė būsena, aprašyta $|\psi\rangle$ vektoriumi, atspindi visą informaciją, kurią galima sužinoti apie kvantinę sistemą. Hilberto erdvė (\mathcal{H}) – tai kompleksinių vektorių erdvė su matematiškai apibrėžta vidine sandauga $\langle\psi|\psi\rangle$. Reikalavimas, kad vektorius $|\psi\rangle$ būtų normuotas, išreiškiamas $\langle\psi|\psi\rangle = 1$. Kiekvienai fizinei sistemai priskiriame atskirą \mathcal{H} erdvę, kurioje galime aprašyti jos visas būsenas. Kvantinė mechanika nenurodo, kokia yra specifinės kvantinės sistemos Hilberto erdvė, tai bendrai gali būti nelengva užduotis sudėtingose sistemose. Tačiau kubito \mathcal{H} erdvė yra viena iš paprasčiausių: tai 2 dimensijų kompleksinė vektorių erdvė. Jeigu $|\psi\rangle$, $|\phi\rangle$ yra du vektoriai, priklausantys tai pačiai \mathcal{H} erdvėi, tada $|u\rangle = a|\psi\rangle + b|\phi\rangle$ yra kita galima tos pačios sistemos būsena, vadinamoji superpozicija. Būsenų normuotumas naudojant amplitudes išreiškiamas $\langle u|u\rangle = |a|^2 + |b|^2 + 2\text{Re}(a^*b\langle\psi|\phi\rangle) = 1$. Jeigu $|\psi\rangle$ ir $|\phi\rangle$ yra ortogonalieji vektoriai, tada trečiasis narys, nusakantis interferenciją tarp būsenų iškrenta iš lygybės, $2\text{Re}(a^*b\langle\psi|\phi\rangle) = 0$, ir randame $|a|^2 + |b|^2 = 1$.

Pirmąjį postulatą galima matematiškai tiksliau perfrazuoti sakant, kad kvantinės būsenos yra nusakomos spinduliu. Spindulys čia reiškia klasę vektorių, kurie vienas nuo kito skiriasi tik globalia faze $e^{i\theta}$. Kadangi globali fazė nėra stebima matavimuose, visos kvantinės būsenos, besiskiriančios tik fazės nariu, yra tarpusavyje ekvivalenčios. Pavyzdžiu, vektoriai $|\psi\rangle$ ir $e^{i\theta}|\psi\rangle$ nusako identišką būseną. Matome, kad dėl kompleksinės jungties naudojimo vidinėje sandaugoje globalios fazės narys pradingsta:

$$\langle\psi|e^{-i\theta}e^{i\theta}|\psi\rangle = e^0\langle\psi|\psi\rangle = 1. \quad (3.1)$$

Atitinkamai ir matavimų metu jis nėra stebimas. Tačiau svarbu pabrėžti skirtumą tarp globalios ir santykinės fazės. Santykinė fazė yra fiziškai svarbi ir jos įtaka stebima matavimų rezultatuose. Matematiškai, globali fazė superpozicijos būsenoje daugina abu narius kartu, pavyzdžiu, $|\psi\rangle = e^{i\theta}(a|0\rangle + b|1\rangle)$. O štai santykinė fazė atrodytų bendrai taip: $|\psi\rangle = a|0\rangle + e^{i\theta}b|1\rangle$. Kaip pavyzdži imkime Pauli-X bazinių vektorių būsenas $|0_x\rangle$ ir $|1_x\rangle$, kurios nusako $|0\rangle$ ir $|1\rangle$ superpozicijas bei skiriasi santykine faze:

$$|0_x\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad |1_x\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \quad (3.2)$$

Akivaizdu, kad šios dvi būsenos yra skirtinos, o tiksliau – ortogonalios, nes $\langle 0_x|1_x\rangle = 0$.

II(a) postulatas – būsenų evoliucija laike. Kvantinės sistemos, izoliuotos nuo išorinių sąveikų, evoliucija laike yra nusakoma unitarine transformacija. Sistemos būsena $|\psi(t_0)\rangle$ laiku t_0 yra susiejama su būsena $|\psi(t_1)\rangle$ vėlesniu laiku t_1 unitarine transformacija $U(t_0, t_1)$:

$$|\psi(t_1)\rangle = U(t_0, t_1)|\psi(t_0)\rangle. \quad (3.3)$$

Antrasis postulatas nusako, kad žinant pradinę uždaros sistemos būseną laiku t_0 , galima apskaičiuoti, kokia bus sistemos būsena laiku t_1 , $|\psi(t_1)\rangle$. Matematiškai tai yra sandauga tarp unitariosios matricos $U(t_0, t_1)$, nulemiančios laiko evoliuciją, ir būseną laiku t_0 nusakančio vektoriaus $|\psi(t_0)\rangle$. Kitaip tariant, uždarą kvantinių sistemų, kaip ir klasikinių, evoliucija laike yra deterministinė. Unitarinėse evoliucijose nėra tikimybėmis nusakomų būsenų kitimų. Toliau pateikiame antrojo postulato versiją, naudojančią Šriodingerio lygtį (angl. *Schrödinger equation*), kuri yra labiau tinkama apibūdinti fizikinėms sąveikoms.

II(b) postulatas – Uždaros kvantinės sistemos būsena laike yra nusakoma Šriodingerio lygtimi:

$$i\hbar \frac{d|\psi(t)\rangle}{dt} = H|\psi(t)\rangle. \quad (3.4)$$

Šriodingerio diferencialinėje lygtje \hbar yra Planko konstanta, i – menamasis vienetas, H – uždaros sistemos hamiltonianas (deja, čia vartojame tą pačią raidę, kaip ir Hadamardo transformacijai). Hamiltonianas – ermitinis operatorius, kurio tikrinės vertės yra sistemos energijos lygmenys. Pavyzdžiu, jeigu $|k\rangle$ yra H operatoriaus tikrinis vektorius, tada yra tenkinama lygybė $H|k\rangle = \lambda_k|k\rangle$, kurioje tikrinė vertė λ_k nusako būsenos $|k\rangle$ energiją.

Sąryšis tarp antrojo postulato dviejų versijų slypi Šriodingerio lygties integracijoje laiko intervale nuo t_0 iki t_1 . Neprarasdami bendrumo darome prielaidą, kad šiuo laiko intervalu hamiltonianas nekinta. Tada, atlikę integraciją laike, gauname:

$$|\psi(t_1)\rangle = e^{-\frac{iH(t_1-t_0)}{\hbar}}|\psi(t_0)\rangle. \quad (3.5)$$

Palyginę su unitarinės evoliucijos laike forma, matome jos bendrą išraišką:

$$U(t_0, t_1) = e^{-\frac{iH(t_1-t_0)}{\hbar}}. \quad (3.6)$$

Kadangi hamiltonianas yra ermitinis operatorius, ši operatoriaus eksponentė nusako unitarųjį operatorių. Kvantiniame kompiuteryje be išorinių sąveikų kubito bendra būsena $|\psi\rangle = a|0\rangle + b|1\rangle$ laike nekinta, tad $U(t) = I$ efektyviai nusako vienetinį operatorių. Kvantiniai loginiai vartai fiziniame lygmenyje yra laike atliekama kubito būsenos transformacija. Hamiltonianas tuo laiko intervalu yra „išjungiamas” pridedant sąveikas nusakančius narius, vadinančias sistemos perturbacijas. Algoritmas nusako seką sistemos perturbacijų, trunkančių tam tikrus laiko intervalus

Δt . Pavyzdžiui, laiko intervale $\Delta t = t_1 - t_0$ gali būti atliekama kubito sąveika su elektromagnetinio lauko pulsu, toliau gali sekti kitas pulsas, taip atliekant seką skirtingų unitarinį transformacijų $\dots U(t_1, t_2)U(t_0, t_1)|\psi(t_0)\rangle$. Analizuojant algoritmus loginiu lygmeniu laiko kintamujų (svarbių fiziniame lygmenyje) neberašome, kadangi $U(t)$ yra perteikiamas jo suminį efektą baziniams vektoriams išreiškiančiu efektyviu unitariniu operatoriumi $U(t) \rightarrow U$.

Be loginius vartus nusakančių sąveikų kvantiniame kompiuteryje galima išskirti dar dvię tipų sąveikas, kuriose sistema traktuojama kaip nebeuždara. Pirmasis tipas – tai nepageidaujamos ir nekontroliuojamos sąveikos su išorinėmis kvantinėmis sistemomis. Jos įveda nežinomas unitarines transformacijas ir nenuspėjamai paveikia kubitų būsenas. Vienintelė tikra uždara kvantinė sistema yra (veikiausiai) pati visata. Kai sakoma, kad kubitali apibūdina uždaras kvantines sistemas, turima omenyje, kad išorinės sąveikos yra itin retos ir todėl galima gerai aproksimuoti jas esant uždaras. Antrojo tipo sąveika yra su makroskopiniais įrenginiais, naudojamais pamatuoti kvantinei būsenai. Pastaruoju atveju sistemas evoliucijos laike negalime nusakyti taikydami antrajį postulatą (net ir išplėtus tai, ką vadiname uždara sistema), nes jos dinamika tampa nedeterministinė. Trečiasis postulatas nusako, kaip kvantinė sistema keičiasi atlikus jos būsenos matavimą.

III postulatas – fizikinių dydžių matavimai. *Kiekvienam stebimam kvantinės sistemos fizikiniam dydžiui egzistuoja ermitinis operatorius P , veikiantis tos sistemos Hilberto erdvėje. Atlikus fizikinio dydžio matavimą galimi rezultatai yra viena iš P operatoriaus tikrinių verčių λ_k . Tikimybė $p(\lambda_k)$ gauti λ_k pamatavas sistemą esančią būsenoje $|\psi\rangle$ randama $p(\lambda_k) = \langle\psi|P_k|\psi\rangle = |\langle k|\psi\rangle|^2$; čia P_k yra projekcinis operatorius $P_k = |k\rangle\langle k|$. Kvantinės sistemos būsena iš karto po matavimo yra P operatoriaus tikrinis vektorius $|k\rangle$, susietas su rasta tikrine verte λ_k .*

Šiame postulate nurodomas būdas apskaičiuoti tikimybes yra dar žinomas kvantinėje mechanikoje kaip **Borno taisykla** (angl. *Born rule*). Kvantinė mechanika nepasako, kokie ermitiniai operatoriai susieti su fizikiniais dydžiais. Analizuojant fizinės sistemos savybes ir matavimo konfigūraciją vis dėlto galima tokius operatorius aprašyti. Selektivus matavimas, naudojantis projekcinius operatorius, yra dažniausiai aptinkamas būsenų matavimo būdas kvantinėje kompiuterijoje. Kaip minėjome antrame skyriuje, ermitinis operatorius gali būti išreikštasis spektrine dekompozicija:

$$P = \sum_k \lambda_k P_k. \quad (3.7)$$

Projekcinis operatorius P_k atlieka vektoriaus projekciją į ortogonalų poerdvį, asocijuotą su tikrine verte λ_k . Kubito būsenos $|\psi\rangle$ selektivusis matavimas pasitelkiant P_k nusako tikimybę gauti būtent λ_k rezultatą. Būseną po P_k projekcinio matavimo, vadinsime ją bendrai $|\varphi\rangle$, galima formaliai rasti:

$$|\varphi\rangle = \frac{P_k|\psi\rangle}{\sqrt{\langle\psi|P|\psi\rangle}} = \frac{|k\rangle\langle k|\psi\rangle}{\sqrt{p(\lambda_k)}} = \frac{a_k|k\rangle}{\sqrt{p(\lambda_k)}}. \quad (3.8)$$

Skaitiklyje, $P_k|\psi\rangle$ atlieka $|\psi\rangle$ būsenos projekciją į $|k\rangle$ vektoriaus poerdvį, o vidinė sandauga $\langle k|\psi\rangle = a_k$ nusako šių vektorių persiklojimą. Vardiklyje esantis narys, $\sqrt{p(\lambda_k)} = \sqrt{|a_k|^2} = |a_k|$, atlieka užduotį sunormuoti vektorių $a_k|k\rangle$.

Dažniausiai susidursime su projekciniais matavimais į ortogonalias kubitų $|0\rangle$ arba $|1\rangle$ būsenų poerdvius (1 dimensijos poerdviai), susietus su tikrinėmis vertėmis $\lambda_0 = 1$ ir $\lambda_1 = -1$, atitinkamai. Tai atlieka Pauli-Z operatorius, kurį galima išreikšti:

$$Z = \sum_k \lambda_k P_k = \lambda_0|0\rangle\langle 0| + \lambda_1|1\rangle\langle 1| = |0\rangle\langle 0| - |1\rangle\langle 1|. \quad (3.9)$$

Projekcinj matavimą, naudojant Pauli- Z , čia vadinsime pakaitomis „standartiniu”, „Pauli- Z operatoriaus matavimu”, arba „matavimu vektorių $\{|0\rangle, |1\rangle\}$ bazėje”. Be Pauli- Z taip pat kvantinių ryšių protokoluose galima aptikti Pauli- X operatoriaus matavimą. Projekciniai operatoriai jį išreiškiame:

$$X = \sum_k \lambda_k P_k = \lambda_0 |0_x\rangle\langle 0_x| + \lambda_1 |1_x\rangle\langle 1_x| = |0_x\rangle\langle 0_x| - |0_x\rangle\langle 0_x|. \quad (3.10)$$

Čia $\{|0_x\rangle, |1_x\rangle\}$ yra Pauli- X tikriniai vektoriai, iš kurių dengiamą poerdvį atliekama $|\psi\rangle$ būsenos projekcija.

Grįžtant prie standartinių Pauli- Z matavimų, jeigu kubitas yra $|0\rangle$ arba $|1\rangle$ būsenoje, bet ne jų superpozicijoje, tada projekcinis matavimas užtikrina, kad šią būseną ir rasime, o pats matavimas būsenos nepakeičia. Pavyzdžiui, jeigu $|\psi\rangle = |1\rangle$, tada tikimybė rasti $|1\rangle$ yra $p(-1)$:

$$p(-1) = \langle 1|1\rangle\langle 1|1\rangle = |\langle 1|1\rangle|^2 = 1, \quad (3.11)$$

ir būsena iškart po matavimo $|\varphi\rangle$:

$$|\varphi\rangle = \frac{|1\rangle\langle 1|1\rangle}{\sqrt{|\langle 1|1\rangle|^2}} = |1\rangle, \quad (3.12)$$

Pakartojė tą patį kubito būsenos matavimą esame užtikrinti, kad rasime vėl tą pačią būseną kaip ir po pirmojo matavimo. Mat formaliai projekcinis operatorius šiuo atveju nepakeičia būsenos $P_k|k\rangle = |k\rangle$, $k \in \{0, 1\}$.

Panagrinėkime, kaip matavimas veikia bendrą normuotą kubito būseną, esančią superpozicijoje $|\psi\rangle = a|0\rangle + b|1\rangle$. Jeigu norime sužinoti, kokia tikimybė rasti kubitą būsenoje $|1\rangle$, tam vėl renkamės $P_1 = |1\rangle\langle 1|$ projekcinį operatorių. Trečiasis postulatas formaliai parodo, kaip vektorių amplitudės yra susietos su matavimo rezultatų tikimybėmis. Randame:

$$\begin{aligned} p(-1) &= \langle \psi | 1 \rangle \langle 1 | \psi \rangle \\ &= b^* b \langle 1 | 1 \rangle \langle 1 | 1 \rangle + a^* a \langle 0 | 1 \rangle \langle 1 | 0 \rangle + a^* b \langle 0 | 1 \rangle \langle 1 | 1 \rangle + b^* a \langle 1 | 1 \rangle \langle 1 | 0 \rangle \\ &= b^* b = |b|^2. \end{aligned} \quad (3.13)$$

Viršuje pritaikėme kubito būsenų ortogonalumą, tad nariai $\langle 1 | 0 \rangle = \langle 0 | 1 \rangle = 0$. Kaip ir tikėjomės, tikimybė rasti kubitą $|0\rangle$ arba $|1\rangle$ būsenoje yra lygi tos būsenos amplitudės (kompleksiniams) kvadratui. Po šio matavimo, kubito būsena yra:

$$|\varphi\rangle = \frac{|1\rangle\langle 1|1\rangle}{\sqrt{|b|^2}} = \frac{b|1\rangle}{|b|} = e^{i\theta}|1\rangle = |1\rangle. \quad (3.14)$$

Gavome $|1\rangle$ su koeficientu $b/|b|$ ir pasinaudojome sakyga, kad kompleksinj skaičių b galima išreikšti $b = |b|e^{i\theta}$. Kadangi kvantinių būsenų vektoriai yra nusakomi iki globalios fazės, narys $e^{i\theta}$ gali būti praleidžiamas galutinėje išraiškoje. Atitinkamai naudotume $P_0 = |0\rangle\langle 0|$ norint apskaičiuoti tikimybę rasti kubitą $|0\rangle$ būsenoje, o $p(1) = |a|^2$.

Viršuje atliktas matavimas bendrai kubito būsenai, esančiai $|0\rangle$ ir $|1\rangle$ superpozicijoje parodo, kad po matavimo ši superpozicija yra sugriaunama. Tai literatūroje dar vadinama kvantinės būsenos suirimu. Po superpozicijos suirimo būsena tampa $|0\rangle$ su tikimybe $|a|^2$, arba $|1\rangle$ su tikimybe $|b|^2$. Tai reiškia, kad jeigu daug kartų pakartosime ši eksperimentą atlikdami kubito paruošimą iš identiškų pradinę būseną bei matavimus, statistinis rezultatas bus nusakomas $p(\lambda_k)$.

Kvantinėje mechanikoje tik būsenos, kurios yra viena kitai ortogonalios, gali būti patikimai atskirtos. Jeigu dvi būsenos, nusakytos $|\psi\rangle$ ir $|\phi\rangle$ vektoriais, yra skirtinges, bet ne ortogonalios $\langle\psi|\phi\rangle \neq 0$, tai reiškia, kad jos iš dalies persikloja. Kitaip tariant, $|\phi\rangle$ būsena turi savyje statmeną ir lygiagretę komponentą $|\psi\rangle$ vektoriaus atžvilgiu. Dėl esamo lygiagretaus komponento atsiranda tikimybė, kad $|\phi\rangle$ būsenos matavimo metu bus rasta $|\psi\rangle$ būsena. Tai yra viena iš priežasčių, kodėl kvantinėje kompiuterijoje naudojami ortogonalieji baziniai vektoriai bei projekciniai matavimai į ortogonalius poerdvius.

Viršuje pateikėme selektivų matavimo būdą, kuris figūruoja kvantinėje kompiuterijoje. Be šio būdo, galima sutikti eksperimentinėje fizikoje labiau įprastą **neselektivų būsenų matavimo metodą** (angl. *non-selective measurement*). Jis suteikia matuojamą fizikinio dydžio, nusakomo tikrinėmis vertėmis λ_k , **vidurkį** (angl. *expectation value*). Pavyzdžiu, 1 kubito atveju, atlikę jo bendros superpozicijos būsenos $|\psi\rangle$ Pauli-Z matavimus n kartų, rastume λ_1 tikrinę vertę ir m kartų λ_{-1} . Tad šių tikrinų verčių vidurkis $\langle\psi|Z|\psi\rangle$ yra:

$$\langle\psi|Z|\psi\rangle = \frac{n\lambda_1 + m\lambda_{-1}}{n + m} = p(1)\lambda_1 + p(-1)\lambda_{-1}. \quad (3.15)$$

Viršuje pateikta formulė, kurią taikytume atlikdami eksperimentą. Apibūdinsime, kaip formaliai apskaičiuojame $\langle\psi|Z|\psi\rangle$. Matuojamą fizikinį dydį nusako ermitinis operatorius P , turintis λ_k tikrines vertes, susietas su vektoriais $|k\rangle$. Viršuje naudojome Pauli-Z operatorių, $P = Z$. Imkime normuotą, nebūtinai kubito, superpozicijos būseną $|\psi\rangle$:

$$|\psi\rangle = \sum_k a_k |k\rangle. \quad (3.16)$$

Čia $|\psi\rangle$ yra išreikšta P operatoriaus tikriniai vektoriai $|k\rangle$ su amplitudėmis a_k . Daug kartų atlikę operatoriaus P neselektivųjų matavimų, trumpai užrašant $\langle P \rangle$, identiškai paruoštai būsenai $|\psi\rangle$, randame:

$$\begin{aligned} \langle P \rangle &= \langle\psi|P|\psi\rangle = \left(\sum_{k'} a_{k'}^* \langle k' | \right) \left(\sum_k \lambda_k P_k \right) \left(\sum_{k''} a_{k''} |k''\rangle \right) \\ &= \sum_k |a_k|^2 \lambda_k. \end{aligned} \quad (3.17)$$

Viršuje formaliai panaudojome $\langle k'|P_k|k''\rangle = \delta_{k'k} \delta_{kk''}$ panaikindami du suminius indeksus. Tad vidurkis yra nulemtas tikimybė $p(\lambda_k) = |a_k|^2$, nusakančių rasti $|k\rangle$ būsenas, sietinas su atitinkamomis tikrinėmis vertėmis λ_k . Tai identiška išraiška tai, kurią naudojome aprašyti eksperimente apskaičiuojamą 1 kubito $\langle\psi|Z|\psi\rangle$.

IV postulatas – sudėtinės sistemos. Sudėtinės kvantinės sistemos būsenų erdvė yra nusakoma jų sudarančių individualių posistemės erdvinių tensorių sandauga, $\mathcal{H}^{\otimes n} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \dots \otimes \mathcal{H}_n$, o sistemos būsena yra tensorinė vektorių sandauga, $|\psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \otimes \dots \otimes |\phi_n\rangle$. Vektorių, išreikštų tenzorinėmis sandaugomis superpoziciją, pavyzdžiu, $|\psi\rangle = a|\phi_1\rangle \otimes \dots \otimes |\phi_n\rangle + b|\phi_1\rangle \otimes \dots \otimes |\phi_n\rangle$, yra kita galima būsena $\mathcal{H}^{\otimes n}$ erdvėje.

Ketvirtasis postulatas praplečia pirmąjį postulatą ir formaliai nusako, kaip matematiškai aprašyti sudėtinės sistemos būsenas. Sistemos, sudarytos iš n kubitų būsenos, vektorius yra nusakomas 2^n dimensijų Hilberto vektorių erdvėje. Ši erdvė yra formuojama iš individualių posistemės erdvinių taikant tensorių sandaugą, kuri žymima simboliu \otimes . Matematinį formulavimą bei sudėtinės sistemų transformacijas jau trumpai apibendrinome antrame skyriuje.

Norint supaprastinti simboliką, vietoj $|\phi_1\rangle \otimes |\phi_2\rangle \otimes \dots$ galime rašyti $|\phi_1\phi_2\dots\rangle$. Kaip pavyzdži imkime 2 kubitų sistemą. Naudojant skaičiuojamąjį bazinių vektorių rinkinį, $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, galima išreikšti bet kokią kubito $|\psi\rangle$ būseną 4 dimensijų Hilberto erdvėje pritaikius jų tiesinę sumą:

$$|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle. \quad (3.18)$$

Čia naudojame kubito numeraciją $|k_1 k_2\rangle$. Kadangi kvantinės būsenos yra nusakomos normuotais vektoriais, amplitudžių modulių kvadratų suma susideda į vienetą:

$$|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1. \quad (3.19)$$

Atlikus 2 kubitų matavimus $|\psi\rangle$ būsenoje, tikimybę rasti bitų sekas, nurodytas baziniuose vektoriuose, galima nusakyti tiesiogiai iš prie jų prišlietų amplitudžių. Rasime $|00\rangle$ su tikimybe $|a|^2$, $|01\rangle$ su $|b|^2$ ir kitas analogiškai. Norėdami apskaičiuoti tikimybę rasti, pavyzdžiui, vien tik pirmą kubito būsenoje $|1\rangle$, formaliai pasitelkiame trečiąjį postulatą apskaičiuoti $\langle\psi|P_1^1|\psi\rangle$. Dirbdami 2 kubitų erdvėje naudojame $P_1^1 = |1\rangle\langle 1| \otimes I$ projekcinį operatorių, kuris per \otimes ženklą nusako, kad projekcinis matavimas į poerdvį, dengiamą $|1\rangle$ būsenos, bus atliekamas su pirmuoju kubitu:

$$\langle\psi|P_1^1|\psi\rangle = (\langle 10|c^* + \langle 11|d^*)(c|10\rangle + d|11\rangle) = |c|^2 + |d|^2. \quad (3.20)$$

Normuotą būseną $|\varphi\rangle$ po šio matavimo randame:

$$|\varphi\rangle = \frac{P_1^1|\psi\rangle}{\sqrt{\langle\psi|P_1^1|\psi\rangle}} = \frac{c|10\rangle + d|11\rangle}{\sqrt{|c|^2 + |d|^2}}. \quad (3.21)$$

Norėdami nusakyti, kokia tikimybė rasti $|\psi\rangle$, pavyzdžiui, $|01\rangle$ būsenoje, naudotume $P_{0,1}^{1,2} = |0\rangle\langle 0| \otimes |1\rangle\langle 1| = |01\rangle\langle 01|$ projekcinį operatorių.

Kvantinės mechanikos postulatas, nusakantis sistemos evoliuciją laike, yra analogiškai taikomas ir sudėtinėse sistemose. Pavyzdžiui, jeigu unitarusis operatorius U_1 nusako pirmo kubito evoliuciją laike, o U_2 antro, tai $U = U_1 \otimes U_2$ formaliai aprašo sudėtinės 2 kubitų sistemos $|\chi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle$ unitarinę evoliuciją:

$$U|\chi\rangle = (U_1 \otimes U_2)|\phi_1\rangle \otimes |\phi_2\rangle = U_1|\phi_1\rangle \otimes U_2|\phi_2\rangle. \quad (3.22)$$

Viršuje pateikti $U_1 \otimes U_2$ operatoriai yra vadinami lokalaisiais, kadangi jie transformuoja individualius posistemės kubitus nepriklausomai nuo kitų būsenos. Bendresnio pobūdžio išplėstiniai operatoriai yra išreiškiami naudojant lokaliuju operatorių sumas. Pavyzdžiui, 2 kubitų $U = U_1 \otimes U_2 + I \otimes U_1$:

$$U|\chi\rangle = U_1|\phi_1\rangle \otimes U_2|\phi_2\rangle + |\phi_1\rangle \otimes U_1|\phi_2\rangle. \quad (3.23)$$

Išplėstiniai operatoriai gali perteikti fizines sąveikas tarp atskirų kubitų ir yra svarbūs realizuojant universalius kvantinių loginių vartų rinkinius. Dėl sąveikų tarp kubitų gali atsirasti kvantinis supynimas, kuris aptariamas toliau.

3.2 Kvantinis supynimas

Klasikinėje fizikoje sistemą galima išsamiai aprašyti nusakant, kokioje būsenoje yra jos atskiro posistemės. Kvantinėje fizikoje galimos ir tokios būsenos, kuriose atskiro posistemės praranda individualią reikšmę ir sistema elgiasi efektyviai kaip vienas darinys. Tokios kvantinės sistemos yra supintosios, jų neįmanoma išsamiai apibūdinti sužinant, kokioje būsenoje yra atskiro posistemės.

Sudėtinės kubitų sistemos būseną galima klasifikuoti įvertinant, ar jos posistemės yra supintosios, ar vis tik galima aprašyti kiekvieną kubitą individualiai. Jeigu sudėtinės sistemos vektorių galima faktorizuoti į atskirus kubitus, pavyzdžiu, $|\kappa\rangle = |\phi_1\rangle \otimes |\phi_2\rangle$, tokia sistema nėra supintoji. Kad tai parodytume formaliau, imkime du kubitus būsenose: $|\phi_1\rangle = e|0\rangle + f|1\rangle$ ir $|\phi_2\rangle = g|0\rangle + h|1\rangle$. Šių dviejų kubitų sudėtinė būsena yra nusakyta jų tenzorių sandauga:

$$\begin{aligned} |\phi_1\phi_2\rangle &= (e|0\rangle + f|1\rangle) \otimes (g|0\rangle + h|1\rangle) \\ &= eg|00\rangle + eh|01\rangle + fg|10\rangle + fh|11\rangle. \end{aligned} \quad (3.24)$$

Tokia būsena leidžia interpretaciją, nusakančią, kad pirmas kubitas yra $|\phi_1\rangle$ būsenoje, o antras $|\phi_2\rangle$. Tai yra dviejų klasikinių bitų būsenos kvantinis atitikmuo. Tolesnė būseną taip pat galima faktorizuoti:

$$|\tau\rangle = a|00\rangle + b|10\rangle, \quad (3.25)$$

nes $|\phi_1\phi_2\rangle$ būsenoje galima rasti koeficientus $h = 0$, $a = eg$, $b = fg$. Vienas supintosios būsenos pavyzdys būtų:

$$|\chi\rangle = a|00\rangle + b|11\rangle. \quad (3.26)$$

Atkreipiame dėmesį, kad neįmanoma rasti $\{e, f, g, h\}$ koeficientų $|\phi_1\phi_2\rangle$ būsenoje, kuri leistų išreikšti $|\chi\rangle$. Šios būsenos neįmanoma faktorizuoti, ir todėl individualių kubitų būsenos yra neapibréžtos. Supintuosiuose kubituose dalis informacijos yra laikoma kvantinėse koreliacijose tarp kubitų. Kadangi šios informacijos fundamentaliai negalime priskirti nei vienam, nei kitam kubitui, sakoma, kad ji yra delokalizuota.

Kad iliustruotume koreliacijų vaidmenį, panagrinėkime sudėtinės 2 kubitų sistemos būsenų matavimus. Imkime viršuje minėtas dvi būsenas, faktorizuojamą $|\tau\rangle$ ir supintą $|\chi\rangle$. Atlikime pirmo kubito projekcinius matavimus naudodami $P_0^1 = |0\rangle\langle 0| \otimes I$ ir $P_1^1 = |1\rangle\langle 1| \otimes I$. Pradėdami nuo faktorizuojamos būsenos $|\tau\rangle$, norime nusakyti, kokia yra tikimybė rasti pirmą kubitą $|0\rangle$ būsenoje:

$$\langle \tau | P_0^1 | \tau \rangle = |a|^2. \quad (3.27)$$

Bendra 2 kubitų būsena, vadinsime ją $|\tau'\rangle$, po šio matavimo yra:

$$|\tau'\rangle = \frac{P_0^1 |\tau\rangle}{\sqrt{\langle \tau | P_0^1 | \tau \rangle}} = |0\rangle \otimes |0\rangle. \quad (3.28)$$

Tą patį pakartokime nusakyti, kokia tikimybė rasti pirmą kubitą $|1\rangle$ būsenoje, bei galutinę 2 kubitų būseną:

$$\langle \tau | P_1^1 | \tau \rangle = |b|^2 \quad (3.29)$$

$$|\tau'\rangle = \frac{P_1^1 |\tau\rangle}{\sqrt{\langle \tau | P_1^1 | \tau \rangle}} = |1\rangle \otimes |0\rangle. \quad (3.30)$$

Akivaizdu, kad nepriklausomai nuo to, ar pirmas kubitas randamas $|0\rangle$, ar $|1\rangle$ būsenoje, tai nepaveikia antro kubito būsenos, kuri lieka visada $|0\rangle$. Tačiau atlikę tuos pačius matavimus su supintąja kubitų būseną $|\chi\rangle$ matome, kad radus pirmą kubitą $|0\rangle$ būsenoje ($|a|^2$ tikimybė), galutinė būsena yra $|0\rangle \otimes |0\rangle$; tai parodo, kad antras kubitas yra užtikrintai $|0\rangle$. O štai radus pirmą kubitą $|1\rangle$ būsenoje ($|b|^2$ tikimybė), galutinė būsena yra $|1\rangle \otimes |1\rangle$, tad antras kubitas yra $|1\rangle$ būsenoje. Nesvarbu, kurį kubitą matuosime pirmą, supintojoje būsenoje po atlikto P_0^1 ar P_1^1 matavimo antrojo kubito būsena tampa tiksliai žinoma.

Be viršuje pateikto 2 kubitų bazinių vektorių skaičiuojamojo rinkinio, **Belo 2 kubitų bazinių vektorių rinkinys** (angl. *Bell basis*) yra taip pat dažnai aptinkamas. Visos būsenos Jame yra supintosios ir tarpusavyje ortogonalios:

$$|\chi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \quad |\chi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle); \quad (3.31)$$

$$|\eta^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \quad |\eta^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \quad (3.32)$$

Įsivaizduokime situaciją, kurioje Evelina paruošia vieną iš keturių Belo būsenų ir nusiunčia vieną kubitą Agnei, o kitą Benui, neatskleisdama, kokia tai Belo būsena. Ar Agnė ir Benas, turėdami galimybę atlikti matavimus su savo individualiai kubitais, gali pasakyti, kokia tai Belo būsena? Sakykime, kad Agnė ir Benas abu atlieka matavimus su kubitais ir perduoda vienas kitam savo rezultatus, rasdami $|00\rangle$ būseną. Tad jie gali spėti, kad tai buvo $|\chi^+\rangle$ arba $|\chi^-\rangle$, tačiau kuri iš šių dviejų – žinoti tiksliai neįmanoma. Šis neapribrėžtumas atspindi faktą, kad supintojoje kubitų būsenoje dalis informacijos yra laikoma kvantinėse koreliacijose, kurios vien lokaliais atskirų sistemų matavimais pasiekti neįmanoma.

Belo baziniai vektoriai yra visi supintieji – tai akivaizdu iš jų būsenos išraiškų. Vis dėlto įvertinti, ar pateikta išraiška nusako supintus kubitus, ne visada paprasta net ir 2 kubitų būsenose. Kaip pavyzdži imkime šią būseną:

$$|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle). \quad (3.33)$$

Ją galimą pergrupuoti taip:

$$|\psi\rangle = \frac{1}{2}\left[(|0\rangle + |1\rangle) \otimes |0\rangle + (|0\rangle - |1\rangle) \otimes |1\rangle \right]. \quad (3.34)$$

Matome, kad atlikę antro kubito matavimą, rasime $|0\rangle$ arba $|1\rangle$ su lygiomis $p = 0.5$ tikimybėmis. Tačiau pirmo kubito būsena lieka superpozicijoje $1/\sqrt{2}(|0\rangle + |1\rangle)$, jeigu antro būsena randama $|0\rangle$, arba $1/\sqrt{2}(|0\rangle - |1\rangle)$, jeigu antro kubito būsena $|1\rangle$. Kitaip nei Belo būsenose, šiuo atveju pirmas 2 kubitų būsenos $|\psi\rangle$ matavimas neleidžia tiksliai sužinoti, kokia bus rasta kito kubito būsena (vektorių $\{|0\rangle, |1\rangle\}$ bazėje). Būsena $|\psi\rangle$ vis dėlto yra supintoji, ją gavome atlikę $|\chi^+\rangle$ lokalią unitariąją pirmo kubito transformaciją (Hadamardo transformaciją), kuri pakeičia $\{|0\rangle, |1\rangle\}$ Pauli-Z bazinius vektorius į Pauli-X bazinius $\{|0_x\rangle, |1_x\rangle\}$. Šie yra išreikšti $|0\rangle$ ir $|1\rangle$ superpozicijomis. Sakome, kad $|\psi\rangle$ ir $|\chi^+\rangle$ yra **lokaliai unitariškai ekvivalenčios** (angl. *local unitary equivalence*). Svarbu atsiminti, kad lokalios unitariosios transformacijos negali nei panaikinti, nei įvesti kvantinio supynimo tarp kubitų. Kvantis supynimas išlieka sistemoje, nebent ji yra veikiama nelokalios unitariosios transformacijos arba atliekamas lokalus projekcinis matavimas.

Nusakyti, ar pateikta kvantinė 2 kubitų būsena yra supintoji, galima taikant minėtą tensorinę faktorizaciją bei vadinamąją Šmito dekompoziciją (angl. *Schmidt decomposition*). Tačiau, kitaip nei 2 kubituose, n kubitų registre ($n > 2$) galimos ir tokios būsenos, kuriose tik dalis kubitų yra tarpusavyje supinti. Vis dar nėra standartinio būdo, kaip tiksliai klasifikuoti supynimą didesnėse kvantinėse sistemose.

3.3 Tankio operatorius

Praktikoje kvantinių sistemų pakartotinis būsenų paruošimas gali būti ne itin tikslus dėl įrangos netikslumų ar klasikinių atsitsiklinių procesų. Pavyzdžiu, **termioninėje elektronų emisijoje**

(angl. *thermionic electron emission*) volframo metalas yra pakaitinamas iki itin didelės temperatūros, kuri suteikia elektronams pakankamai kinetinės energijos, kad šie galėtų pasprukti iš medžiagos. Tačiau neįmanoma iš anksto tiksliai nuspėti kiekvieno pasprukusio elektrono galutinę kinetinę energiją, galime nusakyti tik jų statistinį pasiskirstymą. Šiuo atveju, mūsų informacija apie elektronų energijos būseną yra neįšamai; galime tik pasakyti, kad su klasikine tikimybe p_i (realusis skaičius, $0 \leq p_i \leq 1$) elektronas i bus rastas būsenoje $|\psi_i\rangle$.

Jeigu užtikrintai žinoma, kad kvantinė sistema yra tam tikroje būsenoje $|\psi_k\rangle$, tokia būsena vadinama **gryna** (angl. *pure state*). Remiantis klasikinėmis tikimybėmis, tai reiškia $p_{i=k} = 1$ ir $p_{i \neq k} = 0$. Visas galimų sistemos būsenų (nebūtinai ortogonalijų) rinkinys su atitinkamomis tikimybėmis rasti tam tikrą būseną, $\{p_i, |\psi_i\rangle\}$, yra vadinamas **grynujų būsenų ansambliu** (angl. *ensemble of pure states*). Reikalaujame, kad susumavus visas tikimybes p_i jos susidėtų į vienetą, $\sum_i p_i = 1$. Žinant grynasias būsenas $|\psi_i\rangle$ ir klasikines tikimybes p_i , kurias nustato atitinkamas eksperimentinis paruošimo metodas, vėl galima apskaičiuoti pavienių kvantinių sistemų būsenų matavimo tikimybes.

Pavyzdžiui, jeigu norėtume atlikti selektyvų kinetinės energijos matavimą termioninių elektronų ansambliui $\{p_i, |\psi_i\rangle\}$, tikimybė $p(E_k)$, kad rasime elektroną su energija E_k , yra:

$$p(E_k) = \sum_i p_i \langle \psi_i | P_k | \psi_i \rangle = \sum_i p_i p_i(E_k). \quad (3.35)$$

Čia P_k atlieka projekciją į būsenų su energija E_k poerdvį, o $\langle \psi_i | P_k | \psi_i \rangle = p_i(E_k)$ nusako tikimybę, kad atlikus grynosios būsenos $|\psi_i\rangle$ energijos matavimą bus rasta energija E_k . Išraiškoje matome dvi tikimybes – klasikinę tikimybę p_i bei gryna kvantinio pobūdžio tikimybę $p_i(E_k)$, kuri atsiranda dėl būsenos matavimo proceso. Kvantinei tikimybei negalime priskirti informacijos trūkumo interpretacijos, šis procesas yra fundamentaliai nenuuspėjamas.

Termioninių elektronų emisija yra vienas iš galimų kvantinių sistemų būsenų paruošimo būdų. Kadangi, kaip ir grynujų būsenų atveju, galime tiksliai apibūdinti pavienių ansamblio kvantinių sistemų matavimo rezultatus, taip paruoštos kvantinės sistemos būseną formaliai vadiname **statistiniu grynujų būsenų mišiniu** (angl. *mixed state*). Būsenų mišinio neįmanoma apibūdinti vektoriumi ar vidutiniu vektoriumi Hilberto erdvėje. Norint pilnai aprašyti tokią kvantinę sistemą – apimti klasikines tikimybes bei kvantines amplitudes būsenų mišinyje, formaliai yra pasitelkiamas **tankio operatorius** (angl. *density operator*), arba jo realizacija tiesinėje algebroje – **tankio matrica** (angl. *density matrix*). Visi kvantinės mechanikos postulatai gali būti paprastai formuluojami taikant tankio operatorius vietoj grynasias būsenas nusakančių vektorių. Tai yra bendriausio pobūdžio kvantinės mechanikos formalizmas. Kadangi apibūdindami kvantinės kompiuterijos ir ryšių principus šioje knygoje daugiausia naudosime grynasias būsenas, čia pateikiame tik pagrindines tankio operatorių savybes.

Tankio operatorius, ρ , nusakantis bendriausią kvantinės sistemos būseną, turi šią matematinę išraišką:

$$\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i|. \quad (3.36)$$

Čia $|\psi_i\rangle \langle \psi_i|$ yra projekcinis operatorius į normuotojo $|\psi_i\rangle$ vektoriaus dengiamą poerdvį, šalia prišlietos klasikinės tikimybės p_i susideda $\sum_i p_i = 1$. Grynosios būsenos irgi aprašomos tankio operatoriumi, šiuo atveju $p_i = 1$ tam tikram $|\psi_i\rangle$ ir todėl:

$$\rho = |\psi_i\rangle \langle \psi_i|. \quad (3.37)$$

Pavyzdžiui, kubito, esančio grynojoje superpozicijos būsenoje $|\psi\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, tankio operatorius ir jo realizacija matrica, naudojant $\{|0\rangle, |1\rangle\}$ bazinius vektorius, yra:

$$\rho = |\psi\rangle\langle\psi| = \frac{1}{2}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| + |1\rangle\langle 1|) = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}. \quad (3.38)$$

O štai kubito būsenų mišinio, kuriame yra lygios klasikinės tikimybės rasti $|0\rangle$ arba $|1\rangle$, tankio operatorius yra:

$$\rho = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1| = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \frac{1}{2}I. \quad (3.39)$$

Tankio operatorius $\rho = \frac{1}{2}I$ nusako **maksimaliai sumaišytą būseną** (angl. *maximally mixed state*). Kitaip tariant, turima informacija apie tokią sistemos būseną yra mažiausia, kokia tik gali būti.

Tankio operatorius ρ yra ermitinis operatorius ($\rho = \rho^\dagger$), nes p_i yra realieji skaičiai. Taip pat, ρ yra neneigiamasis operatorius $\rho \geq 0$, nes panaudojus bet kokį vektorių $|\varphi\rangle$ Hilberto erdvėje, $\langle\varphi|\rho|\varphi\rangle \geq 0$:

$$\langle\varphi|\rho|\varphi\rangle = \langle\varphi| \left(\sum_i p_i |\psi_i\rangle\langle\psi_i| \right) |\varphi\rangle = \sum_i p_i |\langle\varphi|\psi_i\rangle|^2 \geq 0. \quad (3.40)$$

Kadangi $|\psi\rangle$ visada galime išreikšti pasirinktais baziniais vektoriais, $|\psi\rangle = \sum_k a_k |k\rangle$, nėra unikalaus būdo perteikti tankio matricai. Viršuje nurodytą grynaąjį būseną galėtume perteikti pasitelkdami patį $|\psi\rangle$ kaip vieną iš bazinių vektorių rinkinyje: $\{|\psi\rangle = |0_x\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, $|1_x\rangle = (|0\rangle - |1\rangle)/\sqrt{2}\}$. Šiuo atveju, tankio matrica turėtų išraišką:

$$\rho = |\psi\rangle\langle\psi| = |0_x\rangle\langle 0_x| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \quad (3.41)$$

Iš tiesų, kadangi ρ yra ermitinis operatorius, galima visada ρ išreikšti diagonaliąja matrica. Vis dėlto yra keletas būdų grynašias būsenas matematiškai atskirti nuo mišriųjų. Vienas būdas – pasitelkiant matricos pėdsaką. Tankio matricos pėdsakas yra:

$$\text{Tr}(\rho) = \sum_i p_i \text{Tr}(|\psi_i\rangle\langle\psi_i|) = \sum_i p_i = 1. \quad (3.42)$$

O štai tankio matricos kvadrato pėdsakas:

$$\text{Tr}(\rho^2) = \sum_{i,j} p_i p_j \text{Tr}(|\psi_i\rangle\langle\psi_i|\psi_j\rangle\langle\psi_j|) = \sum_i p_i^2 \leq 1. \quad (3.43)$$

Viršuje panaudojome $\langle\psi_i|\psi_j\rangle = \delta_{ij}$ bei $0 \leq p_i \leq 1$. Lygybė tenkinama, jeigu $p_i = 1$ su tam tikru i ir lygi nuliui su visais kitais. Todėl grynosios būsenos tenkina sąlygą: $\text{Tr}(\rho^2) = \text{Tr}(\rho) = 1$, na o mišriosios – sąlygą: $\text{Tr}(\rho) = 1$ ir $\text{Tr}(\rho^2) < 1$. Toks būdas atskirti mišriųjų ir grynašias būsenas yra paprastas, nes matricos pėdsakas nepriklauso nuo bazinių vektorių, kuriais išreiškiama matrica.

Diagonaliojoje reprezentacijoje grynosios būsenos visada turi tik vieną matricos elementą, o štai mišrioji – daugiau negu vieną. Dėl to dažnai sakoma, kad diagonalieji tankio matricos elementai įvardija statistinę būsenų populiaciją. Nediagonalieji elementai (angl. *off-diagonal*) suteikia informacijos apie koherencijos būtimą, nes jie yra kvantinių būsenų amplitudžių sandaugos, $a_k a_l^*$. Tai matėme anksčiau perteikdami superpozicijos būsenos $|\psi\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ tankio matricą.

Dekoherencijos procesas yra sietinas su nediagonaliųjų tankio matricos elementų sunykimu ir artėjimu prie maksimaliai sumaišyto būsenos.

Toliau panagrinėsime, kaip formaliai apskaičiuoti kvantinės sistemos, apibūdintos tankio operatoriumi ρ , selektyvaus matavimo tikimybes. Grįžkime prie anksčiau pateikto selektyvaus matavimo termioninių elektronų ansambliai $\{p_i, |\psi_i\rangle\}$, nusakytam mišria būsena $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$. Iterpdam i vienetinį operatorių $I = \sum_j |j\rangle\langle j|$ tarp $\langle\psi_i|$ ir P_k randame:

$$\begin{aligned} p(E_k) &= \sum_i p_i \langle\psi_i|P_k|\psi_i\rangle = \sum_{i,j} p_i \langle j|P_k|\psi_i\rangle\langle\psi_i|j\rangle = \sum_j \langle j|P_k\rho|j\rangle \\ &= \text{Tr}(P_k\rho). \end{aligned} \quad (3.44)$$

Tad tikimybė $p(E_k)$ randama atlikus projekcinės ir tankio matricų sandaugos pėdsaką. Iterp I tarp P_k ir $|\psi_i\rangle$ matytume, kad $\text{Tr}(P_k\rho) = \text{Tr}(\rho P_k)$; tai dar kartą parodo matricų pėdsako cikliškumo savybę. Selektyvus matavimas, pavyzdžiu, norint nusakyti tikimybę $p(1)$ rasti kubitą $|0\rangle$ būsenoje, gali būti atliktas pasitelkiant $P_1 = |0\rangle\langle 0|$ projekcinį operatorių:

$$\begin{aligned} p(1) &= \text{Tr}(\rho P_1) = \text{Tr} \left(\sum_i p_i |\psi_i\rangle\langle\psi_i|P_1 \right) = \sum_i p_i |\langle 0|\psi_i\rangle|^2 \\ &= \sum_i p_i |a_{i,0}|^2. \end{aligned} \quad (3.45)$$

Selektyvus būsenų mišinio matavimas, jeigu nėra būsenų išsigimimo, sukuria galutinę grynažą būseną. Tai išplaukia iš trečiojo kvantinės mechanikos postulato ir yra vienas būdas, kaip paruošti grynažias būsenas. O štai neselektyvus P operatoriaus matavimas būsenų mišiniui apskaičiuojamas iš $\langle P \rangle = \text{Tr}(\rho P)$ ir bendrai sukuria kitą būsenų mišinį.

Kvantinių sistemų, aprašytų tankio operatoriumi, evoliucija laike nuo t_0 iki t_1 nusakoma unitariiniu operatoriumi $U(t_0, t_1)$ taip:

$$\rho(t_1) = U(t_0, t_1)\rho(t_0)U^\dagger(t_0, t_1) = \sum_i p_i U(t_0, t_1)|\psi_i(t_0)\rangle\langle\psi_i(t_0)|U^\dagger(t_0, t_1). \quad (3.46)$$

Norint apibūdinti loginių vartų U efekta pradinei būsenai, apibūdintai tankio matrica ρ , glaustai rašome:

$$\rho' = U\rho U^\dagger. \quad (3.47)$$

II(b) postulatas, nusakantis sistemos evoliuciją Šriodingerio lygtimi, yra paprastai formuliuojamas taikant fon Noimano–Liuvilio (angl. *von Neumann–Liouville*) lygtį:

$$i\hbar \frac{d\rho(t)}{dt} = [H, \rho(t)]. \quad (3.48)$$

Lygtje H nusako sistemos hamiltonianą, skliausteliai indikuoją H ir ρ komutatorių.

Sudėtinę kvantinę sistemą, kurios būsena yra faktorizuojama, galima išreikšti tenzorine tankio operatorių sandauga, $\rho = \rho_1 \otimes \rho_2 \otimes \rho_3 \otimes \dots$; čia ρ_i nusako posistemės i būseną. Jeigu tankio matricos išreikšti tenzorine posistemų sandauga neįmanoma, $\rho \neq \rho_1 \otimes \rho_2$, tokia sistema yra supintoji. Imkime kaip pavyzdži 2 kubitų supintą grynažą būseną $|\chi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Jos

tankio operatorius ir matrica, vektorių $\{|0\rangle, |1\rangle\}$ bazėje yra:

$$\begin{aligned}\rho &= |\chi^+\rangle\langle\chi^+| = \frac{1}{2}(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|) \\ &= \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.\end{aligned}\quad (3.49)$$

Tankio operatorių formalizmas suteikia būdą aprašyti posistemės būseną nepriklausomai nuo to, ar sudėtinė sistema yra supinta, ar faktorizuojama. Tam pasitelkiamas **dalinis matricų pėdsakas** (angl. *partial trace*), kuriame atliekamas pėdsakas tik vienai iš pasirinktų posistemų. Imkime, kad ρ_{AB} nusako dviejų sistemų A ir B tankio matricą, tada:

$$\text{Tr}_A(\rho_{AB}) = \rho_B, \quad \text{Tr}_B(\rho_{AB}) = \rho_A. \quad (3.50)$$

Gautas darinys ρ_A arba ρ_B yra vadinamas **redukuota tankio matrica** (angl. *reduced density matrix*) ir nusako atitinkamos posistemės būseną. Formaliai tariant, dalinis pėdsakas yra operatorius, veikiantis tik vieną posistemę. Pavyzdžiu, jeigu $\rho_{AB} = |a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|$, tada:

$$\text{Tr}_A(\rho_{AB}) = \sum_j (\langle j| \otimes I) \rho_{AB} (|j\rangle \otimes I) = \text{Tr}(|a_1\rangle\langle a_2|) |b_1\rangle\langle b_2|. \quad (3.51)$$

Kaip ir tikėtasi, faktorizuojama būsenos ρ_{AB} redukuota tankio matrica yra $|b_1\rangle\langle b_2|$ (iki globalios konstantos). Grįžtant prie supintosios 2 kubitų grynosios būsenos $\rho = |\chi^+\rangle\langle\chi^+|$, išdomu ivertinti jos redukuotą tankio matricą ρ_1 arba ρ_2 . Atlikdami dalinį pėdsaką pirmajam kubitui randame:

$$\rho_2 = \text{Tr}_1(\rho) = \sum_{j=0,1} (\langle j| \otimes I) \rho (|j\rangle \otimes I) = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \frac{1}{2} I. \quad (3.52)$$

Matome, kad redukuota tankio matrica apibūdina maksimaliai sumaišytą būseną, nepaisant to, kad sudėtinės sistemos ρ būsena yra grynoji (žinoma užtikrintai). Tai – dar vienas supintųjų kvantinių būsenų skiriamasis ženklas.

3.4 EPR paradoksas

Į kvantinių koreliacijų įtaką supintųjų būsenų matavimui atkreipė dėmesį ir A. Einšteinas, su kolegomis 1935 metais publikavęs žinomą darbą, dabar vadinamą **EPR paradoksu** (angl. *Einstein-Podolsky-Rosen paradox*). Čia pateikiamo jų įsivaizduojamo eksperimento versiją, pasiūlytą Davido Bohmo ir naudojančią dviejų lygių kvantinę sistemą (kubitus). Įsivaizduokime, kad yra paruošiama supintoji 2 kubitų būsena, pavyzdžiu, ši:

$$|\chi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \quad (3.53)$$

Pirmas fizinis kubitas paliekamas Žemėje, o antras nusiunčiamas į tolimą planetą pavadinimu W . Nors šių supintųjų kubitų būsena yra grynoji ir tiksliai žinoma, tačiau pavienių kubitų būsenos neapibrėžtos. Atlikus matavimą yra lygios tikimybės rasti pirmą kubitą $|0\rangle$ arba $|1\rangle$ būsenoje. Tačiau pamatavus pirmą kubitą Žemėje ir radus, pavyzdžiu, $|1\rangle$ būseną, antro kubito būsena planeteje W bus taip pat užtikrintai rasta $|1\rangle$, ($|\chi^+\rangle \rightarrow |11\rangle$). Einšteinas su kolegomis čia ižvelgė potencialų prieštaravimą reliatyvumo teorijai, kadangi vienos sistemos būsenos matavimas

akimirksniu paveikia kitos sistemos būseną, nepaisant atstumo tarp jų. Kvintinė mechanika, EPR kompanijos požiūriu, negali būti išsami teorija, apibūdinanti gamtą. Nors jų argumentai neįtikino fizikų bendruomenės, tačiau tai buvo svarbus darbas, atkreipiantis dėmesį į kvantinės mechanikos ivariuose kontekstuose pasikartojančias keistąsias savybes.

Norėdami iliustruoti EPR paradoksui artimą gyvenimiską pavyzdį imkime, kad Benas turi dvi identiškas dėžutes ir vienoje iš jų paslepią raudoną kamuolį, o kitoje – mėlyną. Agnė ir Benas žino, kad dėžutėse yra dviejų skirtingų spalvų kamuoliai, tačiau tik Benas žino, kuriose jie paslépti. Agnės požiūriu, tikimybė atidarius pasirinktą dėžutę rasti vieną ar kitą kamuolį yra $p = 0.5$. Tačiau atidariusi vieną dėžutę ji automatiškai žinos, kokios spalvos kamuolys yra kitoje. Šiuo atveju Agnės nežinojimas apie išankstinį Beno kamuolių išdėstymą jai suteikė koreliacijos pagrindą, panašų į atsirandantį dėl kubitų supynimo.

Sekant šį klasikinį pavyzdį vienas argumentas paaikiinti EPR paradoksą būtų agitujant, kad kubitai kažkaip susimokė nuo pat pradžių ir jų būseną vis dėlto buvo $|00\rangle$ arba $|11\rangle$, tačiau mes šios informacijos neturėjome. Norėdami patikrinti tokį argumentą, galime žvelgti į EPR paradoksą kitu būdu. Kaip pamename, kvintinę būseną visada galima išreikšti bet kuriais kitais pageidaujamais baziniais vektoriais. Vietoj Pauli- Z , imkime Pauli- X bazinius vektorius $\{|0_x\rangle, |1_x\rangle\}$. Perteikę jais $|\chi^+\rangle$ randame:

$$|\chi^+\rangle = \frac{1}{\sqrt{2}}(|0_x\rangle \otimes |0_x\rangle + |1_x\rangle \otimes |1_x\rangle). \quad (3.54)$$

Šiuo atveju, $|\chi^+\rangle$ turi identišką formą, kaip ir naudojant Pauli- Z bazinius vektorius. Taip pat galime pasirinkti skirtingą matavimo būdą. Pirmam kubitui būsenoje $|\chi^+\rangle$ galime atlikti ne standartinius Pauli- Z operatoriaus matavimus: $P_0^1 = |0\rangle\langle 0| \otimes I$ ir $P_1^1 = |1\rangle\langle 1| \otimes I$, bet pasirinkti Pauli- X : $P_{0x}^1 = |0_x\rangle\langle 0_x| \otimes I$ ir $P_{1x}^1 = |1_x\rangle\langle 1_x| \otimes I$. Fiziskai tai galėtų reikšti, kad fotono poliarizacijos būseną matuojame ne išilgai su horizontaliaja–vertikaliacija poliarizacijos ašimis, tačiau pasukę šias abi matavimo kryptis 45° kampu (istrižoji poliarizacija). Vėlgi matome, kad atlikus P_{0x}^1 ar P_{1x}^1 būsenai $|\chi^+\rangle$ yra lygios tikimybės rasti pirmą kubitą $|0_x\rangle$ arba $|1_x\rangle$ būsenoje. Galutinės dviejų kubitų būsenos po tokio matavimo tampa:

$$|\chi^+\rangle \rightarrow \frac{P_{0x}^1 |\chi^+\rangle}{\sqrt{\langle \chi^+ | P_{0x}^1 | \chi^+ \rangle}} = |0_x\rangle \otimes |0_x\rangle, \quad (3.55)$$

arba:

$$|\chi^+\rangle \rightarrow \frac{P_{1x}^1 |\chi^+\rangle}{\sqrt{\langle \chi^+ | P_{1x}^1 | \chi^+ \rangle}} = |1_x\rangle \otimes |1_x\rangle. \quad (3.56)$$

Tai, kokį matavimo būdą pasirinksime pirmam kubitui Žemėje, nulems, kokia bus antro kubito būsena planeteje W . Jeigu renkamės Pauli- Z operatoriaus matavimus, tada antro kubito būsenos galimi variantai bus $|0\rangle$ arba $|1\rangle$, o jeigu renkamės Pauli- X , jie bus $|0_x\rangle$ arba $|1_x\rangle$. Tačiau kubitas negali būti apibrėžtoje Pauli- X ir Pauli- Z bazinių vektorių būsenoje vienu metu, nes šie operatoriai yra nekomutatyvūs ir todėl neturi bendrų bazinių vektorių. Tad, jeigu kubitai buvo $|00\rangle$ arba $|11\rangle$ būsenoje nuo pat pradžių (bet mes to nežinojome), pirmo kubito matavimo būdo pasirinkimas negali turėti įtakos antram kubitui. Antraip matavimo krypties pasirinkimas turėtų būti komunikuojamas akimirksniu, greičiau už šviesos greitį.

EPR paradoksas jau buvo patvirtintas ne kartą eksperimentuose naudojant supintąsias fotonų būsenas. Atstumai tarp dviejų lokacijų, kuriose susinchronizuotai tuo pačiu metu buvo matuojamos supintujų fotonų būsenos, buvo tokie, kad šviesai neužtektų laiko komunuoti matavimo

rezultatus, kažkaip juos paveikiant. Pagal kvantinės mechanikos standartinę (ortodoksinę) interpretaciją, neįmanoma žinoti, kuris kvantinis „kamuolys” yra kurioje dėžutėje. Būsena $|\chi^+\rangle$ nusako, kad jie yra abiejuose dėžutėse vienu metu. Tad pagal standartinę interpretaciją, EPR paradoksas nepriestarauja reliatyvumo teorijai, nes neįmanoma nuspėti, kokie bus pirmo kubito individualių matavimų rezultatai, žinomas tik tikimybės rasti $|0\rangle$ arba $|1\rangle$. Nusiųsti informaciją, užkoduotą kubitu būsenose, greičiau už šviesos greitį į planetą W neįmanoma, nes tiesiog ne galime suformuluoti žinutės – jos turinys bus atsitiktinis. Tik patikrinę Žemėje ir planetoje W atliktų rezultatų statistiką matysime koreliaciją tarp matavimo rezultatų. Na, o norint palyginti rezultatus, tektų komunikuoti šviesos greičiu arba lėčiau, jeigu keliautume į planetą W .

IV skyrius

Kvantiniai loginiai vartai ir grandinės

Šiame skyriuje pateikiame įprastinių kvantinių loginių elementų arsenalą. Juos pasitelkus formuojami žemiausio loginio lygio kvantiniai algoritmai. Matysime, kaip loginės registro būsenų operacijos yra pavaizduojamos grafiškai ir kaip iš jų sekų formuojamos kvantinės grandinės.

4.1 Vieno kubito loginiai vartai

Pradėkime nuo paprasčiausių loginių elementų, atliekančių 1 kubito būsenų transformacijas. Kaip pamename, kvantiniai loginiai vartai yra nusakomi unitariniais operatoriais. Tokio operatoriaus veiksmas būsenai $|\psi\rangle$ yra pavaizduojamas grafiškai (žr. 4.1 pav.).

$$|\psi\rangle \xrightarrow{\text{U}} U|\psi\rangle$$

4.1 pav.: Loginių vartų U veiksmas kubito būsenai $|\psi\rangle$.

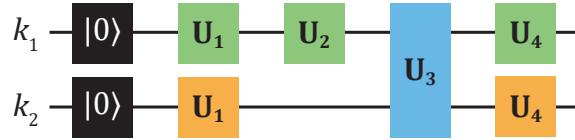
Kiekvienam kubitui yra priskiriama atskira grandis – horizontali linija. Kvatinė grandis pati nedaro įtakos kubito būsenai ir nusako jo laisvą evoliuciją laike. Grandys yra skaitomos iš kairės į dešinę ir atspindi laike atliekamas loginių vartų sekas. Kairėje pusėje nusakoma pradinė kubito būsena, čia $|\psi\rangle$, jai atliekami loginiai vartai U ir dešinėje išvedama pakitusi būsena $U|\psi\rangle$. Visi 1 kubito loginiai vartai turi vieną įvedimą ir vieną išvedimą, o bendras kubitų skaičius algoritmo metu nesikeičia. Žinoma, ne visi kubitai registre turi būti panaudojami.

Čia pateikiame loginius vartus ir jų efektą skaičiuojamiesiems baziniams vektoriams $|0\rangle$ ir $|1\rangle$. Norint nusakyti, kaip loginiai vartai keičia kubito būseną, esančią superpozicijoje, pakanka žinoti, kaip loginiai vartai keičia $|0\rangle$ ir $|1\rangle$ atskirai, nes operatoriai veikia tiesiniu būdu kiekvieną vektorių superpozicijoje:

$$U(|0\rangle + |1\rangle) = U|0\rangle + U|1\rangle. \quad (4.1)$$

Skaičiavimuose galima atlikti bazinių vektorių transformaciją ir naudoti kitų bazinių vektorių rinkinį, pavyzdžiui Pauli- X tikrinius vektorius $|0_x\rangle$ ir $|1_x\rangle$. Svarbu prisiminti, kad loginių vartų efektas, pateiktas $|0\rangle$ ir $|1\rangle$ baziniams vektoriams, bus bendrai kitoks, jeigu tie patys vartai pritaikomi, pavyzdžiui, $|0_x\rangle$ ir $|1_x\rangle$ ir jais išreikštoms būsenomis.

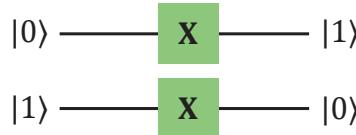
4.2 pav. matome dviejų kubito registro kvantinės grandinės pavyzdį.



4.2 pav.: 2 kubitų registro kvantinė grandinė. Matomi individualius kubitus veikiantys loginiai vartai ir abu kubitus vienu metu veikiantys U_3

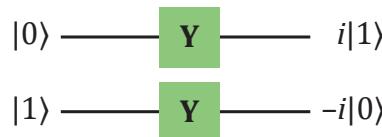
Standartiškai algoritmo pradžioje kubitai yra inicijuojami į $|0\rangle$ būsenas (nurodome *ket* juodame fone), o atskiri kubitai sunumeruojami simboliais k_1, k_2, \dots . Viršuje kubitui k_1 yra atliekama 1 kubito loginių vartų seka U_1, U_2 , abu kubitus veikiantis U_3 ir galiausiai U_4 . Atkreipime dėmesį, kad tiesinėje algebroje pirmo kubito loginių vartų sekos efektas yra apskaičiuojamas taip: $U_4 U_3 U_2 U_1 |0\rangle$. Tai yra, pirmiausiai operatorius U_1 iš kairės daugina dešinėje vektorių $|0\rangle$, toliau jis dauginamas su U_2, U_3 , ir galiausiai U_4 . Ši atvirkštinė tvarka tarp matematinio skaiciavimo ir atvaizdavimo kvantinėse grandinėse gali būti klaidinanti. Kadangi tai yra plačiai paplitusi ir standartine tapusi vartosena, ši skirtumą belieka tik įsiminti. Norėdami formaliai išsamiai aprašyti parodytas logines operacijas, veikiančias du kubitus $|k_1\rangle \otimes |k_2\rangle$, rašytume jas taip $(U_4 \otimes U_4)(U_3)(U_2 \otimes I)(U_1 \otimes U_1)(|0\rangle \otimes |0\rangle)$. Operatorius U_3 veikia abu kubitus, o U_2 atlikimo metu kubitui k_2 nepritaikoma jokia loginė operacija (laukimo stadija) – tai nusakoma vienetiniu operatoriumi I .

Pradėsime nuo Pauli- X , Y ir Z loginių vartų, kurių matematinės išraiškos ir savybės buvo pri-
statyti 2 skyriuje. Pauli- X loginiai vartai yra klasikinių NE loginių vartų atitinkmuo, jie sukeičia 0 ir 1 vertes (žr. 4.3 pav.)



4.3 pav.: Pauli- X loginių vartų efektas skaičiuojamiesiems baziniams vektoriams

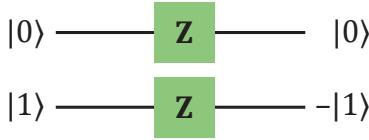
Pauli- Y vartai taip pat sukeičia 0 ir 1 vertes, tačiau dar prideda $\pm\pi/2$ fazę ($e^{\pm i\pi/2} = \pm i$), nusakomą menamuoju skaičiumi i (žr. 4.4 pav.)



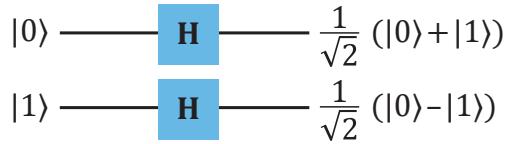
4.4 pav.: Pauli- Y loginių vartų efektas

Pauli- Z vartai (žr. 4.5 pav.) nekeičia įvedamos $|0\rangle$ būsenos, tačiau prideda π fazę $|1\rangle$ būsenai ($e^{i\pi} = -1$).

Hadamardo vartai (žr. 4.6 pav.) suformuoja lygią $|0\rangle$ ir $|1\rangle$ bazinių vektorių superpoziciją.



4.5 pav.: Pauli-Z loginiu vartu efektas



4.6 pav.: Hadamardo loginiu vartu efektas skaičiuojamiesiems baziniams vektoriams

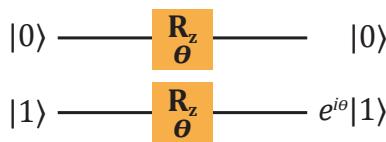
Kaip pamename, Blocho sferoje 1 kubito būsena yra parametrizuojama dviem kampais θ ir φ . Čia θ nusako kampą su z ašimi, o φ kampas yra skaičiuojamas nuo x ašies imant Blocho vektoriaus projekciją į $x-y$ plokštumą. Tačiau praktikoje atliekant Blocho vektoriaus posūkius kvantiniame kompiuteryje yra dažnai naudojami loginiai vartai, kurie nusako posūkį apie vieną iš trijų Blocho sferos (x , y , z) ašių. Šie loginiai vartai yra vadinami $R_x(\theta)$, $R_y(\theta)$ ir $R_z(\theta)$, parametras θ nusako, kad posūkis atliekamas apie nurodytą ašį kampu θ :

$$R_x(\theta) \equiv e^{-\frac{i\theta X}{2}} = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) X = \begin{bmatrix} \cos(\theta/2) & -i \sin(\theta/2) \\ -i \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}; \quad (4.2)$$

$$R_y(\theta) \equiv e^{-\frac{i\theta Y}{2}} = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) Y = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}; \quad (4.3)$$

$$R_z(\theta) \equiv e^{-\frac{i\theta Z}{2}} = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) Z = \begin{bmatrix} e^{-\frac{i\theta}{2}} & 0 \\ 0 & e^{\frac{i\theta}{2}} \end{bmatrix}. \quad (4.4)$$

Panagrinėkime šiek tiek plačiau $R_z(\theta)$ loginius vartus (žr. 4.7 pav.), kuriuos kvantinėje grandinėje ir supaprastinta matricos bei diadų forma išreiškiame:

4.7 pav.: Posūkio apie Blocho sferos z aši $R_z(\theta)$ loginiu vartu efektas skaičiuojamiesiems baziniams vektoriams

$$R_z(\theta) \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}; \quad (4.5)$$

$$R_z(\theta) = |0\rangle\langle 0| + e^{i\theta}|1\rangle\langle 1|. \quad (4.6)$$

Supaprastinome $R_z(\theta)$ išraišką iškeldami bendrą $e^{-i\theta/2}$ nari, kurį toliau ištrynėme. Ši nari 1 kubito loginiuose vartuose galima ignoruoti, kadangi jis daugina bendrą kubito būseną ir nusako įtakos neturinčią globalią fazę. Dėl šio supaprastinimo kampas, kuriuo vartais $R_z(\theta)$ pasukamas Blocho vektorius apie z ašę, yra, kaip nurodyta, θ , o ne $\theta/2$.

Kaip matome, $R_z(\theta)$ neturi įtakos $|0\rangle$ būsenai, bet prideda fazę θ būsenai $|1\rangle$. Pavyzdžiu, jeigu ši būsena būtų $|\psi\rangle = a|0\rangle + e^{i\alpha}b|1\rangle$, tada $R_z(\theta)|\psi\rangle = a|0\rangle + e^{i(\alpha+\theta)}b|1\rangle$. Dėl šios savybės $R_z(\theta)$ yra dažnai vadinami 1 kubito **fazės vartais** (angl. *phase gate*). Pauli-Z loginiai vartai yra ne kas kita, kaip $R_z(\theta)$ su $\theta = \pi$ (iki globalios fazės). Algoritmuose taip pat dažnai naudojami $R_z(\pi/4)$ ir $R_z(\pi/2)$, kurie standartiškai yra nurodomi raidėmis T ir S , atitinkamai:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}; \quad (4.7)$$

$$S = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}. \quad (4.8)$$

Loginiai vartai S yra dar vadinami \sqrt{Z} , kadangi $S^2 = Z$.

Naudojant trijų operatorių su trimis skirtingais parametrais seką, $R_x(\theta)R_y(\gamma)R_z(\varphi)$, galima formaliu išreikšti bet kokią 1 kubito būsenos unitariają transformaciją $U_3(\theta, \gamma, \varphi)$. Tačiau praktikoje dažnai aptinkamos vadinamosios **Oilerio dekompozicijos N-M-N** (angl. *Euler decomposition*), kuriose $N, M \in \{R_x, R_y, R_z\}$. Pavyzdžiu, $Z-Y-Z$ dekompozicija, naudojanti du skirtingus rotacijos operatorius:

$$U_3(\theta, \gamma, \varphi) = R_z(\theta)R_y(\gamma)R_z(\varphi) = \begin{bmatrix} e^{-i(\theta+\varphi)/2} \cos(\gamma/2) & -e^{i(-\theta+\varphi)/2} \sin(\gamma/2) \\ e^{i(\theta-\varphi)/2} \sin(\gamma/2) & e^{i(\theta+\varphi)/2} \cos(\gamma/2) \end{bmatrix}. \quad (4.9)$$

4.2 Kvantinių grandinių lygbių ir atvirkštiniai loginiai vartai

Yra begalė būdų pasukti vektorių Blocho sferoje į norimą orientaciją naudojant skirtingus loginius vartus. Pavyzdžiu, Hadamardo transformacija gali būti išreikšta rotacijų seka apie x, y, z ašis. Keletas būdų tai atliskti:

$$H = R_x(\pi)R_y\left(-\frac{\pi}{2}\right); \quad (4.10)$$

$$H = R_x\left(-\frac{\pi}{2}\right)R_z\left(-\frac{\pi}{2}\right). \quad (4.11)$$

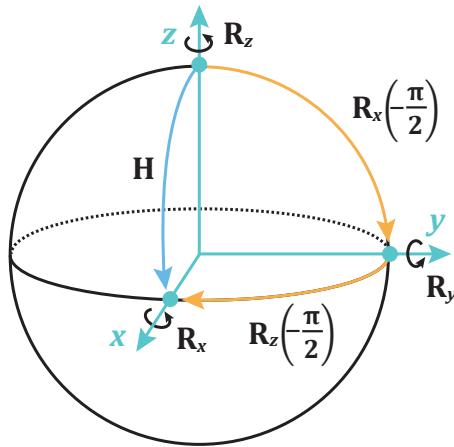
Antrają išraišką iliustruojame 4.8 pav., kaip trajektoriją Blocho sferos paviršiuje, pradedant nuo $|0\rangle$ būsenos iki galutinės $|0_x\rangle$. Atkreipiame dėmesį, kad žvelgiant į teigiamą x, y, z ar z ašę nuo sferos centro į jos išorę, posūkis teigiamu kampu $+\theta$ apibūdinamas prieš laikrodžio rodyklę, o neigiamas $-\theta$ pagal laikrodžio rodyklę.

Skirtingose kvantinių procesorių architektūrose gali būti apribotas tam tikrų 1 kubito loginių vartų naudojimas. Todėl gali atsirasti poreikis sukompiliuoti kvantinę grandinę kitomis loginių vartų sekomis ar ją supaprastinti sumažinant bendrą loginių vartų skaičių. Pateikiame keletą svarbesnių 1 kubito loginių vartų lygbių pavyzdžių:

$$HXH = Z, \quad HYH = -Y, \quad HZH = X; \quad (4.12)$$

$$ZXZ = -X, \quad ZYZ = -Y, \quad ZZZ = Z; \quad (4.13)$$

$$SXS^\dagger = Y, \quad SYS^\dagger = -X, \quad SZS^\dagger = Z. \quad (4.14)$$



4.8 pav.: Ekvivalentinis H loginės operacijos realizavimas naudojant rotacijos loginius vartus R_x ir R_z pavaizduojant kubito $|0\rangle$ vektoriaus sukimo trajektoriją Blocho sferoje. Šalia x, y, z ašių pažymėti atitinkamai posūkio operatoriai ir nurodytos teigiamos $+\theta$ sukimo kryptys

Kaip galime pastebėti, viršuje pateikiamos tokios loginių vartų sekos: $KPK^\dagger = P'$. Čia P ir P' yra Pauli operatoriai, o K yra parenkamas iš 1 kubito **Klifordo grupės** (angl. *Clifford group*) operatorių. Šiai grupei priklausančius 1 kubito loginius vartus, tarp kurių yra ir Pauli- $\{X, Y, Z\}$, galima sugeneruoti vien tik H ir S loginių vartų sekomis. Viršuje matėme kaip realizuoti Pauli- Y , na o X ir Z randami:

$$HSH = X, \quad SS = Z. \quad (4.15)$$

Norédami patikrinti, ar lygybės egzistuoja tarp skirtingų vartų sekų, galime jas nusakančias matricas tiesiog sudauginti ir palyginti, ar jos vienodos. Iliustracijai:

$$HXH = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = Z. \quad (4.16)$$

Klifordo vartais taip pat galima tarpusavyje transformuoti tolydžiai parametrizuotus posūkio loginius vartus, pavyzdžiui $R_z(\theta)$ ir $R_x(\theta)$:

$$HR_z(\theta)H = R_x(\theta). \quad (4.17)$$

Naudodami matricų funkcijos $R_z(\theta)$ išraišką, lygybę viršuje patikriname:

$$\begin{aligned} H[\cos(\theta/2)I - i\sin(\theta/2)Z]H &= \cos(\theta/2)HIH - i\sin(\theta/2)HZH \\ &= \cos(\theta/2)I - i\sin(\theta/2)X = R_x(\theta). \end{aligned} \quad (4.18)$$

Trys Pauli bei Hadamardo loginiai vartai yra ermitiniai operatoriai ($U = U^\dagger$) ir todėl patys sau atvirkštiniai. Atliekant du ermitinius operatorius vieną po kito anuliuojamas jų efektas, pavyzdžiui, $ZZ = I$, $HH = I$. Kitaip nei Pauli ir Hadamardo vartai, dauguma kitų kvantinių loginių vartų nėra sau atvirkštiniai. Pavyzdžiui, jau minėti S ir T nėra ermitiniai operatoriai, nes $S \neq S^\dagger$ ir $T \neq T^\dagger$. Tačiau, kaip ir visi unitariniai operatoriai, jie turi sau atvirkštinius

operatorius. Tai yra jų ermitinės jungties operatoriai, žymimi su durklu, pavyzdžiu T^\dagger ir S^\dagger :

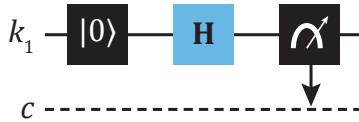
$$T^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\pi/4} \end{bmatrix}; \quad (4.19)$$

$$S^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\pi/2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}. \quad (4.20)$$

Matome, kad ermitinėje jungtyje atsiranda minuso ženklas prie nario, nusakančio kampą. Atvirkštinių vartų efektą galima interpretuoti kaip Blocho vektoriaus sukimą apie tą pačią ašį priešinga kryptimi tokiu pačiu kampu. Dėl šios priežasties atlikus operaciją $UU^\dagger|\psi\rangle = I|\psi\rangle = |\psi\rangle$ būsena nepakinta. Tai savo ruožtu reiškia, kad kvantinėse transformacijose, nusakytose unitariaisais operatoriais, visada įmanoma grąžinti kvantinę būseną į pradinę (kitaip tariant, atsukti laiką atgal). Tam tereikia atlikti atvirkštinius loginius vartus atbuline seką.

4.3 Kubitų būsenų matavimas

Norint sužinoti kubito būseną, algoritmo vykdymo metu ar jo pabaigoje atliekamas matavimas. Šioje knygoje matavimo operacija yra žymima pusapskritimių su rodyklyte, nukreipta į klasikinį registrą, žymimą brūkšniuota linija ir raide c . Praktikoje dažnai taikomas principas, kad algoritmo gale visi kubitai yra pamatuojami, net jeigu tai nėra parodoma simboliais kvantinės grandinės pabaigoje. Matavimai standartiškai atliekami santykinai su Pauli- Z operatoriaus tikriniaus vektoriais, tai yra skaičiuojamaisiais baziniais vektoriais $|0\rangle$ ir $|1\rangle$. Kubitui esant šių bazinių vektorių superpozicijoje, galimi matavimo rezultatai yra Pauli- Z tikrinės vertės $+1$ arba -1 . Radus $+1$ indikuojama, kad būsena yra $|0\rangle$, bei $|1\rangle$, jeigu tikrinė vertė yra -1 . Tai nusako vieną klasikinį bitą informacijos, o atlikus matavimą gautas rezultatas yra įrašomas klasikiniame bitų registre.



4.9 pav.: Loginė grandinė sukuria kubito k_1 lygią superpoziciją ir atlieka jo būsenos matavimą. Rezultatas įrašomas klasikiniame registre c

Grandinėje 4.9 pav. yra pateiktas vienas iš paprasčiausių praktinės svarbos kvantinių algoritmu, kuris atlieka atsitiktinių skaičių generavimą (angl. *random number generator*). Atlikę Hadamardo transformaciją gauname lygią superpoziciją:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \quad (4.21)$$

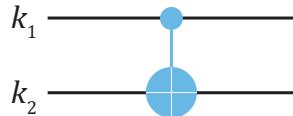
Pamatavus šią būseną tikimybės rasti $|0\rangle$ arba $|1\rangle$ yra lygios:

$$|\langle 0|\psi\rangle|^2 = |\langle 1|\psi\rangle|^2 = \frac{1}{2}. \quad (4.22)$$

Tad po pakartotinio šio algoritmo įvykdymo yra sugeneruojama atsitiktinė dvejetainė 0 ir 1 skaičių sekā. Dėl kvantinio atsitiktinumo matavimo procese, šios sekos fundamentaliai neįmanoma nuspėti.

4.4 Dviejų kubitų loginiai vartai *CNOT*

Dviejų kubitų vartai, veikdami $|k_1\rangle \otimes |k_2\rangle$ būseną, atlieka sąlyginės logines operacijas: „Jeigu kubitas k_1 yra būsenoje $|x\rangle$, tada kubitiui k_2 atliekama operacija U “. Čia U gali būti bet kokia 1 kubito būsenos unitarioji transformacija. Vienas iš plačiausiai naudojamų 2 kubitų vartų yra *CNOT* (angl. *controlled not*, trumpinys *CNOT*), kuriuos grandinėse vadinsime cX . Pasirinktas pirmas kubitas yra naudojamas kaip **kontrolinis** (angl. *control*), nuo kurio būsenos priklauso ar antram, **adresatiniam** (angl. *target*) kubitiui, bus taikomi Pauli- X loginiai vartai. Vartai cX yra iliustruojami 4.10 pav.



4.10 pav.: 2 kubitų sąlyginiai loginiai vartai cX (arba *CNOT*)

CNOT vartus galime išreikšti diadomis ir matricos forma:

$$cX = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X; \quad (4.23)$$

$$cX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (4.24)$$

Kaip pavyzdži imkime cX , kuriame pirmas kubitas yra kontrolinis, ir pritaikykime jį $|1\rangle \otimes |0\rangle$ būsenai:

$$cX|1\rangle \otimes |0\rangle = |0\rangle\langle 0|1\rangle \otimes I|0\rangle + |1\rangle\langle 1|1\rangle \otimes X|0\rangle = |1\rangle \otimes X|0\rangle = |1\rangle \otimes |1\rangle; \quad (4.25)$$

$$cX|1\rangle \otimes |0\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (4.26)$$

Apibendrinus, cX vartų efektas 2 kubitų skaičiuojamiesiems baziniams vektoriams:

$$cX|0\rangle \otimes |0\rangle = |0\rangle \otimes |0\rangle; \quad (4.27)$$

$$cX|0\rangle \otimes |1\rangle = |0\rangle \otimes |1\rangle; \quad (4.28)$$

$$cX|1\rangle \otimes |0\rangle = |1\rangle \otimes |1\rangle; \quad (4.29)$$

$$cX|1\rangle \otimes |1\rangle = |1\rangle \otimes |0\rangle. \quad (4.30)$$

Matome, kad kai kontrolinis kubitas yra $|0\rangle$, antro kubito būsena nesikeičia, o jeigu kontrolinis kubitas yra $|1\rangle$, antro kubito vertė apverčiama. Tai lengva pamatyti žvelgiant į diadų formą, kurioje Pauli- X vartai veikia kartu tik su pirmo kubito $|1\rangle$ būseną. cX vartų efektą baziniams vektoriams galima interpretuoti ir kaip modulio(2) bitų sudėtį ($0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$, $1 + 1 = 0$), naudojančią simbolį \oplus . Pritaikius cX tarp kubitų k_1 ir k_2 , mod(2) sudėtis yra užrašoma antro kubito būsenoje:

$$cX|k_1\rangle \otimes |k_2\rangle = |k_1\rangle \otimes |k_1 \oplus k_2\rangle. \quad (4.31)$$

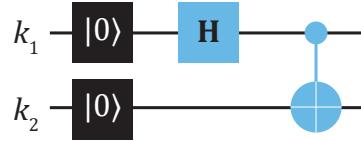
Galima atsakymą užrašyti ir pirmo kubito būsenoje naudojant cX vartus, kuriuose antras kubitas yra kontrolinis, o pirmas adresatinis.

Norint aiškiau pateikti matematines lygtis gali būti pravartu indikuoti operatoriuose kontrolinį ir adresatinį kubitus. Šioje knygoje, kai bus tą pravartu daryti, $CNOT$ vartuose nurodysime kaip pirmą skaičių kontrolinį kubitą, antrą adresatinį, pavyzdžiu cX_{12} . Formulėse (4.23)–(4.24) pateikėme diadų ir matricų formas cX_{12} , sukeitus juos vietomis, cX_{21} , atitinkamai pasikeistų ir matematinės išraiškos:

$$cX_{21} = I \otimes |0\rangle\langle 0| + X \otimes |1\rangle\langle 1|; \quad (4.32)$$

$$cX_{21} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \quad (4.33)$$

Matėme, kaip 2 kubitų būsena keičiasi pritaikius cX , kai kubitai yra viename iš bazinių vektorių. Gauti rezultatai yra identiški klasikiniams cX vartams. Tačiau kvantinėje kompiuterijoje kontrolinio kubito būsena gali būti $|0\rangle$ ir $|1\rangle$ superpozicijoje. Koks yra cX efektas, šiuo atveju panagrinėsime žvelgdami į 4.11 pav. grandinę.



4.11 pav.: Kvantinė 2 kubitų supynimą atliekanti grandinė

Po Hadamardo transformacijos kontrolinis k_1 kubitas yra lygioje superpozicijoje. Toliau yra cX vartai, kurie veikia tiesiškai kiekvieną narį superpozicijos būsenoje:

$$\begin{aligned} cX \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle &= \frac{1}{\sqrt{2}}(cX|0\rangle \otimes |0\rangle + cX|1\rangle \otimes |0\rangle) \\ &= \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle). \end{aligned} \quad (4.34)$$

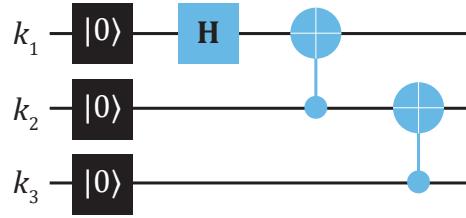
Matome, kad po šių dviejų loginių vartų sekos gavome supintą 2 kubitų būseną $|\chi^+\rangle$ Belo bazinių vektorių. Naudojant Hadamardo ir cX vartus šia tvarka galima unikalai konvertuoti skaičiuojamuosius 2 kubitų bazinius vektorius į Belo bazinius vektorius:

$$|00\rangle \rightarrow |\chi^+\rangle, \quad |10\rangle \rightarrow |\chi^-\rangle, \quad |01\rangle \rightarrow |\eta^+\rangle, \quad |11\rangle \rightarrow |\eta^-\rangle. \quad (4.35)$$

Pavyzdžiui, norėdami paruošti Belo $|\eta^-\rangle$ būseną, pirmiausia pakeičiame k_1 ir k_2 pradines būsenas į $|11\rangle$ naudodami Pauli- X vartus ir tada vėl atliekame H ir cX .

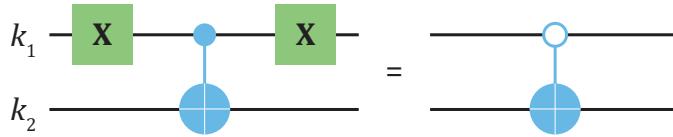
Kadangi cX operatorius yra ermitinis, atlikus vieną po kito du cX su tais pačiais kontroliniu ir adresatiniu kubitais, jų būsenos nepakinta. Kvantinio supynimo panaikinimas tarp dviejų kubitų taip pat yra atliekamas naudojant cX vartus. Pavyzdžiui, atliekant cX vartus supintajai $|\chi^+\rangle$ Belo būsenai supynimas yra panaikinamas ir grįztame į faktorizuojamąjį būseną:

$$cX|\chi^+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle. \quad (4.36)$$



4.12 pav.: Kvantinė 2 kubitų supynimą atliekanti grandinė

Kita gerai žinoma visiškai supinta 3 kubitų GHZ būseną (angl. *Greenberger-Horne-Zeilinger*) galima gauti atlikę 4.12 pav. parodytą grandinę.

4.13 pav.: 2 kubitų salyginį loginių vartų cX variacija, kurioje antro kubito būsena keičiama, jeigu pirmojo būsena $|0\rangle$

$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle). \quad (4.37)$$

Užbaigdami šį poskyrį paminėsime naudingą cX loginių vartų variaciją, kurioje adresatiniam kubitiui pritaikomi Pauli- X vartai, jeigu kontrolinio kubito būsena yra $|0\rangle$, o ne $|1\rangle$. Šio operatoriaus, vadinsime jį cX_0 , išraiška yra:

$$cX_0 = |0\rangle\langle 0| \otimes X + |1\rangle\langle 1| \otimes I. \quad (4.38)$$

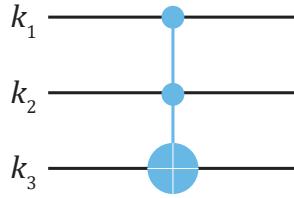
Kvantinėje grandinėje cX_0 vartai yra žymimi su tuščiu apskritimu kontroliniame kubite, bei gali būti paprastai konvertuojami iš standartinio cX pasitelkiant dvejus Pauli- X vartus (žr. 4.13 pav.). Pirmieji Pauli- X sukeičia kontrolinio kubito būsenas $|0\rangle \leftrightarrow |1\rangle$, o antrieji atstato jas atgal po atliktu standartinių cX .

4.5 Tofoli loginiai vartai

Trijų kubitų salyginiai loginiai vartai, kuriuose du kubitai naudojami kaip kontroliniai, yra vadinami **$CCNOT$** (angl. *controlled controlled NOT*, trumpinys $CCNOT$) ir yra geriau žinomi, kaip **Tofoli vartai** (angl. *Toffoli*). Grandinėse juos žymėsime ccX . Kubito vertė yra apverčiama, jeigu abu kontroliniai kubitai yra $|1\rangle$ būsenose. 4.14 pav. iliustruojame šiuos vartus, veikiančius būseną $|k_1\rangle \otimes |k_2\rangle \otimes |k_3\rangle$ su k_1 ir k_2 kontroliniais bei k_3 adresatiniu kubitu. Išreiškus ccX vartus diodomis:

$$ccX = (|00\rangle\langle 00| + |01\rangle\langle 10| + |10\rangle\langle 01|) \otimes I + |11\rangle\langle 11| \otimes X. \quad (4.39)$$

Kvantiniuose kompiuteriuose ccX vartai paprastai nėra elementarūs, tačiau sukompiliuojami iš 1 kubito ir 2 kubitų loginių vartų sekų.



4.14 pav.: Toffoli loginiai vartai

4.6 SWAP ir Fredkin loginiai vartai

Loginiai vartai SWAP, grandinėse žymimi sutrumpintai W , veikdami tarp dviejų kubitų k_1 ir k_2 sukeičia jų būsenas vietomis:

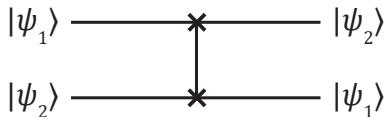
$$W|\psi_1\rangle \otimes |\psi_2\rangle = |\psi_2\rangle \otimes |\psi_1\rangle. \quad (4.40)$$

Šie loginiai vartai efektyviai sukeičia kubitus vietomis ir todėl gali būti naudojami pergrupuoti kubitus registre. Tai yra itin naudinga situacijose, kuriuose kvantinis procesorius neturi fizinės galimybės atlikti, pavyzdžiu, cX loginių vartų tarp tam tikrų kubitų porų. Siekiant apeiti šį apribojimą galima naudoti SWAP operacijas kaskadų principu sukeičiant kubitus pozicijas į vietas, kuriuose cX loginiai vartai yra leidžiami, bei vėl grąžinti kubitus į pradines pozicijas. Šiuos vartus galima išreikšti diadomis arba matricos pavidalu taip:

$$W = |0\rangle \otimes |0\rangle \langle 0| \otimes |0\rangle \langle 0| + |0\rangle \otimes |1\rangle \langle 1| \otimes |0\rangle \langle 0| + |1\rangle \otimes |0\rangle \langle 0| \otimes |1\rangle \langle 1| + |1\rangle \otimes |1\rangle \langle 1| \otimes |0\rangle \langle 0| \\ = |00\rangle \langle 00| + |01\rangle \langle 10| + |10\rangle \langle 01| + |11\rangle \langle 11|; \quad (4.41)$$

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.42)$$

SWAP vartai iliustruojami 4.15 pav.

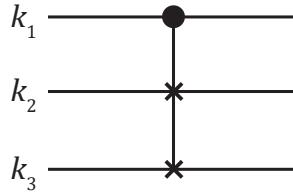


4.15 pav.: SWAP loginiai vartai

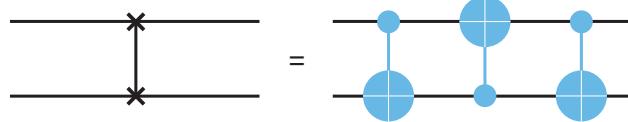
SWAP taip pat galima iškomponuoti į sąlyginius loginius vartus, kurie yra vadinami Fredkin vartais. Lygtyste juos žymėsime cW . 4.16 pav. pavaizduoti Fredkin vartai sukeičia kubitus k_2 ir k_3 būsenas, jeigu kontrolinis kubitas k_1 yra būsenoje $|1\rangle$.

$$cW = |0\rangle \langle 0| \otimes I \otimes I + |1\rangle \langle 1| \otimes W. \quad (4.43)$$

Kaip ir Tofoli loginiai vartai, SWAP bei Fredkin paprastai nėra elementarios kubitus transformacijos fiziniame lygmenyje, tačiau yra konstruojami iš 1 kubito ir 2 kubitus loginių vartų sekų. Vienas būdas atlikti SWAP 2 kubitus būsenai naudojant cX vartus parodytas 4.17 pav.:



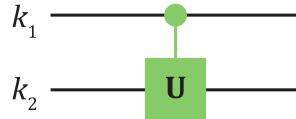
4.16 pav.: Fredkin, arba kontroliuojami SWAP, loginiai vartai

4.17 pav.: SWAP loginiu vartų realizavimas pasitelkiant tris $*cX*$ loginius vartus

4.7 Bendro tipo sąlyginiai loginiai vartai cU

Bendro tipo 2 kubitų sąlyginiuose vartuose cU adresatiniam kubitui pritaikoma bet kokia 1 kubito unitarinė transformacija U , jeigu pirmas kubitas yra $|1\rangle$ būsenoje. Šiuos vartus diadų forma ir grandinėje išreiškiame:

$$cU = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U. \quad (4.44)$$

4.18 pav.: Bendro tipo 2 kubitų loginiai vartai cU , kuriuose antram kubiti pritaikomi bendro tipo 1 kubito loginiai vartai U , kontroliuojant pirmu kubitu

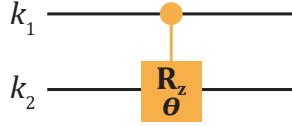
Prieš pereidami prie kitų temų, atkreipsime dėmesį į 1 kubito globalios fazės svarbą 2 kubitų (ir bendrai n kubitų) loginiuose vartuose. Bendro tipo 2 kubitų transformacijose cU globali pavienių kubitų fazė tampa svarbi santykinė fazė tarp atskirų kubitų. Kaip matėme šio skyriaus pirmoje dalyje, bendriausio tipo 1-kubito loginius vartus $U_3(\theta, \gamma, \varphi)$ galime išreikšti tokia matricos forma:

$$U_3 = \begin{bmatrix} a & -b^* \\ b & a^* \end{bmatrix}. \quad (4.45)$$

Čia a ir b yra kompleksiniai skaičiai, o matricos U_3 determinantas $\det U_3 = |a|^2 + |b|^2 = 1$. Visos unitarinės (2×2) dydžio matricos U , kurių determinantas lygus vienetiui, sudaro specialiąjį unitarinijų matricų grupę $SU(2)$. Ši grupė išsamiai apibūdina visas 1 kubito transformacijas, tačiau yra platesnės unitarinijų matricų $U(2)$ grupės pogrupis, $SU(2) \subset U(2)$. Unitarinę matricą U , priklausančią $U(2)$, galime išreikšti sudauginant $SU(2)$ matricą V su fazės nariu, $U = e^{i\eta}V$. Globalios fazės narys $e^{i\eta}$, veikiantis 1 kubito būseną nedaro fizinės įtakos, nes $e^{i\eta}V|\psi\rangle = V|\psi\rangle$. Todėl jų 1-kubito transformacijose ignoruojame. Tačiau sąlyginiuose 2 kubitų loginiuose vartuose unitarinė (1 kubito) transformacija U , pritaikoma adresatiniam kubitiui, gali priklausyti $U \in$

$U(2)$, o ne $U \in \text{SU}(2)$ grupei. Pabréždami šį aspektą toliau pateikiame sąlyginius posūkio apie z ašį vartus $cR_z(\theta)$ bei fazės vartus $cP(\theta)$ (angl. *controlled phase gate*).

Vartai $cR_z(\theta)$ pritaiko antram kubitui posūkio operatorių $R_z(\theta)$, jeigu pirmojo būsena yra $|1\rangle$. Grandinėje, diadū ir matricos formoje $cR_z(\theta)$ atrodo taip (žr. 4.19 pav.):



4.19 pav.: Salyginiai 2 kubitų loginiai vartai $cR_z(\theta)$

$$cR_z(\theta) = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes R_z(\theta); \quad (4.46)$$

$$cR_z(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{-i\theta/2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\theta/2} \end{bmatrix}. \quad (4.47)$$

Pritaikę $cR_z(\theta)$ 2 kubitų bazinių vektorių superpozicijos būsenai $|\psi\rangle = (|00\rangle + |10\rangle + |01\rangle + |11\rangle)/\sqrt{2}$ randame:

$$\begin{aligned} cR_z(\theta)|\psi\rangle &= \frac{1}{2} \left[(|00\rangle + |01\rangle) + |1\rangle \otimes R_z(\theta)(|0\rangle + |1\rangle) \right] \\ &= \frac{1}{2} \left[|00\rangle + e^{-i\theta/2}|10\rangle + |01\rangle + e^{i\theta/2}|11\rangle \right]. \end{aligned} \quad (4.48)$$

Šio skyriaus pirmoje dalyje analizuodami 1 kubito vartus $R_z(\theta)$ iškélėme fazės nari $e^{-i\theta/2}$. Tačiau šioje situacijoje $e^{-i\theta/2}$ nusako svarbią santykinę fazę tarp 2 kubitų bazinių vektorių, o ne globalią visos 2 kubitų būsenos $|\psi\rangle$ fazę. Globali 2 kubitų fazė būtų vėlgi būtų nusakoma $e^{i\eta}$, dauginančiu visą $|\psi\rangle$ būseną kartu.

Salyginiai fazės vartai $cP(\theta)$ matricos forma atrodo taip:

$$cP(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\theta} \end{bmatrix}. \quad (4.49)$$

Pritaikę juos tai pačiai visų 2 kubitų bazinių vektorių superpozicijos būsenai $|\psi\rangle$ randame, kad fazė suteikiama tik $|11\rangle$ baziniam vektoriui:

$$cP(\theta)|\psi\rangle = \frac{1}{2} [|00\rangle + |10\rangle + |01\rangle + e^{i\theta}|11\rangle]. \quad (4.50)$$

Vartų $cP(\theta)$ efektas skiriasi nuo $cR_z(\theta)$ santykinė 2 kubitų fazė, kuri yra „globali“ antrojo kubito fazė. Norėdami formaliau perteikti 1 kubito „globalią“ fazę $e^{i\eta}$, tokius 1 kubito fazės vartus žymime $\Phi(\eta)$:

$$\Phi(\eta) = \begin{bmatrix} e^{i\eta} & 0 \\ 0 & e^{i\eta} \end{bmatrix} = e^{i\eta} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = e^{i\eta} I. \quad (4.51)$$

Sąlyginiai $c\Phi(\eta)$ vartai, kurie perteikia „globalią” $e^{i\eta}$ fazę adresatinio kubito $|0\rangle$ ir $|1\rangle$ būsenoms, jeigu kontrolinis kubitas $|1\rangle$ būsenoje, yra:

$$c\Phi(\eta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\eta} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\eta} \end{bmatrix}; \quad (4.52)$$

$$c\Phi(\eta) = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes \Phi(\eta) = (|0\rangle\langle 0| + e^{i\eta}|1\rangle\langle 1|) \otimes I. \quad (4.53)$$

Iš diadinės $c\Phi(\eta)$ dekompozicijos galima atkreipti dėmesį, kad iš tiesų nereikia vykdyti 2 kubitu sąlyginių vartų. Šie loginiai vartai susiprastina į 1 kubito loginius vartus, veikiančius vien kontrolinį kubitą, su identitetu $\otimes I$ adresatiniam kubitui:

$$\begin{aligned} \Phi(\eta/2)R_z(\eta) &= (e^{i\eta/2}|0\rangle\langle 0| + e^{i\eta/2}|1\rangle\langle 1|)(e^{-i\eta/2}|0\rangle\langle 0| + e^{i\eta/2}|1\rangle\langle 1|) \\ &= |0\rangle\langle 0| + e^{i\eta}|1\rangle\langle 1|. \end{aligned} \quad (4.54)$$

Tad sąlyginius 2 kubitu fazės vartus $c\Phi(\eta)$ galime perteikti 1 kubito vartais, $\Phi(\eta/2)R_z(\eta) \equiv D$:

$$c\Phi(\eta) = D \otimes I. \quad (4.55)$$

Dabar akivaizdu, kad sąlyginius 2 kubitu santykinės fazės vartus $cP(\theta)$ galime realizuoti pasitelkdami $cR_z(\theta)$ ir $c\Phi(\eta)$. (4.48) lygyje gautai būsenai $cR_z(\theta)|\psi\rangle$ pritaikome $c\Phi(\eta)$ su faze $\eta = \theta/2$:

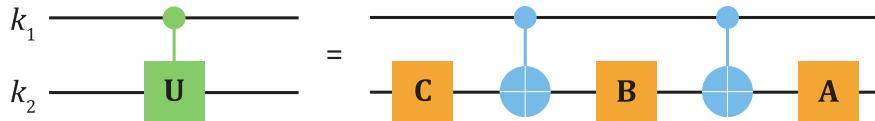
$$c\Phi(\theta/2)cR_z(\theta)|\psi\rangle = cP(\theta)|\psi\rangle. \quad (4.56)$$

Realiame kvantiniame kompiuteryje 2 kubitu sąlyginių loginių vartų assortimentas gali būti itin ribotas, o bendro tipo sąlyginiai vartai cU yra veikiau aukštesnio lygio loginių operacijų abstrakcija. Šios operacijos bus sudarytos iš kvantiniame kompiuteryje prieinamų elementariųjų loginių vartų. Dažnai daroma prielaida, kad iš 2 kubitu vartų yra prieinami tik cX . Tad norint įvykdyti visas įmanomas n kubitų registro transformacijas universaliai kompiuteryje reikalingas būdas, kaip perteikti cU naudojant tik 1 kubito bendrus loginius vartus bei cX .

Pirma dekompozicija, realizuojanti cU , yra:

$$cU = AcXBcXC. \quad (4.57)$$

Čia A , B ir C yra 1 kubito loginiai vartai, kurie veikdami būseną $|\psi\rangle$ tenkina lygybę $ABC|\psi\rangle = I|\psi\rangle$. Ši dekompozicija yra pagrįsta teiginiu, kad bet kokį 1 kubito operatorių $U \in \text{SU}(2)$ galima išreikšti $U = AXBXC$. Operatoriai A , B , C turi būti rasti kiekvienam norimam U ir yra bendrai sudaryti iš posūkio operatorių $R_x(\theta)$, $R_y(\theta)$, $R_z(\theta)$.

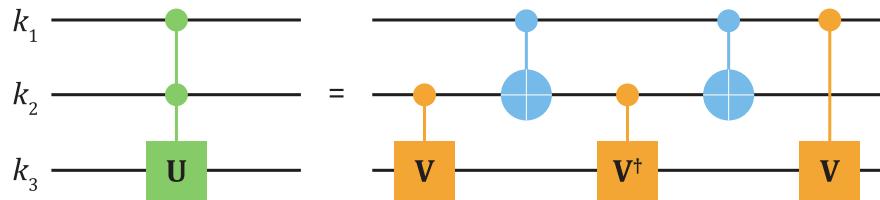


4.20 pav.: Sąlyginių loginių vartų cU realizavimas naudojant ABC dekompoziciją

4.20 pav. grandinėje matome, kad jeigu kontrolinis kubitas yra $|0\rangle$ būsenoje, tada adresatiniam kubitui pritaikomi paeiliui trys operatoriai, $ABC|\psi\rangle = I|\psi\rangle$ ir todėl nepakeičia jo būsenos. O štai jeigu kontrolinis kubitas yra $|1\rangle$ būsenoje, $AXBXC|\psi\rangle = U|\psi\rangle$, kaip ir norima. Taip pat, jeigu sąlyginiuose vartuose cU unitarūs operatorius $U \in \text{U}(2)$, tada $(D \otimes I)AXBXC|\psi\rangle = U|\psi\rangle$.

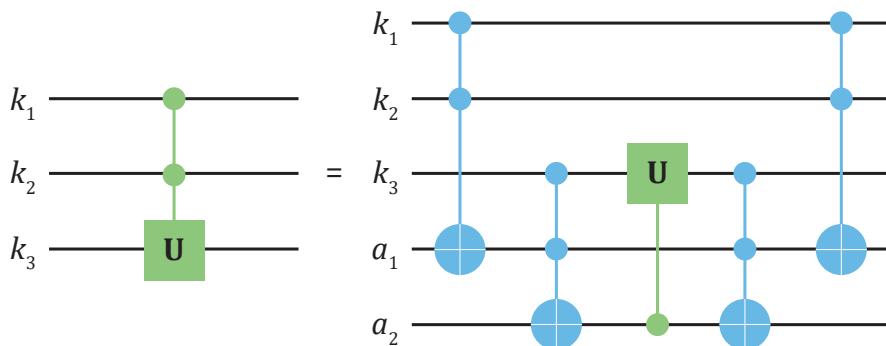
Papildomi fazės vartai D , veikiantys kontrolinj kubitą šios grandinės pabaigoje, leidžia teisingai perteikti norimą santykinę 2 kubitų būsenos fazę.

Loginiuose vartuose gali būti daugiau nei vienas kontrolinis kubitas. Ši aukštesnio lygio operacijų abstrakcija yra itin pravartai konstruojant logines grandines, tačiau praktikoje taip pat reikia dekompozicijos į elementarius loginius vartus jai realizuoti. 4.21 pav. pateikti 3 kubitų bendro tipo ccU ir jų dekompozicija naudojant elementarius cX , 1 kubito loginius vartus V ir atvirkštinus V^\dagger , $V^\dagger V = I$. Jie yra parenkami, kad tenkintų lygybę $VV = U$. Pavyzdžiui, Tofoli loginiai vartai gali būti išreikšti šia dekompozicija pasirinkus $V = (1 - i)(I + iX)/2$, nes $VV = X$.



4.21 pav.: 3 kubitų sąlyginių loginių vartų ccU realizavimas

Tofoli loginiai vartai ccX yra plačiai naudojami kvantinėse grandinėse ir juos pasitelkus galima lengvai išreikšti k skaičiumi kubitais kontroliuojamus loginius vartus $c^k U$. Iliustracijai, 4.22 pav. parodyta ccU dekompozicija, naudojanti Tofoli sekas. Ši metodą panagrinėsime detaliau.



4.22 pav.: 3-kubitų sąlyginių loginių vartų $*ccU*$ realizavimas pasitelkiant Tofoli loginius vartus

Tofoli vartais pagrįstas metodas naudoja papildomus kubitus, vadinamus **ancilomis** (angl. *ancilla qubits*), kurios atlieka juodraščio funkciją kvantinėse grandinėse. Ancilos yra inicijuojamos į $|a_1 a_2\rangle = |00\rangle$ būsenas, o atlikus norimą operaciją vėl grąžinamos į pradines $|00\rangle$. Imkime konkretų pavyzdį, kuriame $ccU = ccP(\theta)$ nusako minėtus, tačiau dvigubai kontroliuojamus, sąlyginius 3 kubitų fazės vartus. Pritaikius $ccP(\theta)$ 3-kubitų registrui būsena $|k_1 k_2 k_3\rangle = |111\rangle$ įgauna fazę, $e^{i\theta}|111\rangle$, o štai visi kiti baziniai vektoriai nėra paveikiami. Sakysime, kad 3 kubitų registras yra pradinėje lygioje visų bazinių vektorių superpozicijoje, tad kartu su dvimi ancilomis (kubitais) jų būsena formoje $|k_1 k_2 k_3\rangle \otimes |a_1 a_2\rangle$ yra:

$$|\psi\rangle = \frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle) \otimes |00\rangle. \quad (4.58)$$

Paskutiniai du kubitai yra minėtos ancilos, atskirtos tensorių daugybos ženklu dėl aiškumo. Tolesniuose žingsniuose pritaikysime dvejus Tofoli loginius ccX vartus. Pirmasis, vadinsime jį

ccX_1 , naudoja įvesties registro kubitus k_1 ir k_2 kaip kontrolinius, o adresatinis yra pirmasis ancila kubitas a_1 . Iš to randame:

$$\begin{aligned} ccX_1|\psi\rangle &= \frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle) \otimes |00\rangle \\ &+ \frac{1}{\sqrt{8}}(|110\rangle + |111\rangle) \otimes |10\rangle. \end{aligned} \quad (4.59)$$

Šie vartai parenka būsenas $|11k_3\rangle$ ir jas supina su pirmu ancila kubitu, kuriam būsena pakeičiama į $|1\rangle$. Tolesnis ccX_2 naudoja įvesties registro kubitą k_3 ir ancilą a_1 kaip kontrolinį kubitą, o adresatinis yra antrasis ancila kubitas a_2 :

$$\begin{aligned} (ccX_2)(ccX_1)|\psi\rangle &= \frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle) \otimes |00\rangle \\ &+ \frac{1}{\sqrt{8}}|110\rangle \otimes |10\rangle + \frac{1}{\sqrt{8}}|111\rangle \otimes |11\rangle. \end{aligned} \quad (4.60)$$

Šie vartai supina $|111\rangle$ būseną su ancilų $|11\rangle$ būseną. Trečiame žingsnyje pritaikome sąlyginius fazės vartus $cP(\theta)$, kuriuose kontrolinis kubitas yra antra ancila a_2 , o adresatinis šiuo atveju gali būti bet kuris iš trijų įvesties kubitų, sakysime, k_3 :

$$\begin{aligned} cP(\theta)(ccX_2)(ccX_1)|\psi\rangle &= \frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle) \otimes |00\rangle \\ &+ \frac{1}{\sqrt{8}}|110\rangle \otimes |10\rangle + \frac{e^{i\theta}}{\sqrt{8}}|111\rangle \otimes |11\rangle. \end{aligned} \quad (4.61)$$

Šie trys loginiai žingsniai efektyviai pritaikė fazę $e^{i\theta}$ būsenai $|111\rangle$. Tolesniuose dviejuose žingsniuose pritaikome Tofoli vartus atbuline tvarka siekdamai atstatyti abiejų ancila kubitų būsenas atgal į $|00\rangle$, taip paruošiant jas potencialiai tolimesniems skaičiavimams:

$$\begin{aligned} (ccX_1)(ccX_2)(cP(\theta))(ccX_2)(ccX_1)|\psi\rangle &= \frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle \\ &+ |011\rangle + |100\rangle + |101\rangle + |110\rangle + e^{i\theta}|111\rangle) \otimes |00\rangle. \end{aligned} \quad (4.62)$$

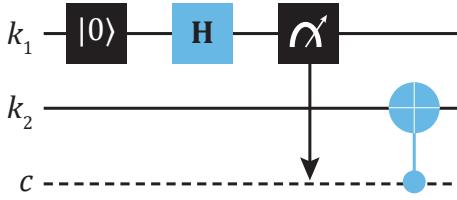
Bet kokia k kontroliuojamų sąlyginių loginių vartų c^kU dekompozicija, pagrįsta Tofoli vartais ccX , paremta iliustruotu kaskadų principu. Šiam metodui reikalingi $2(k - 1)$ skaičius Tofoli vartų ir papildomų $k - 1$ ancila kubitų.

Kvantinių ryšių protokoluose, klaidų taisymo ir kituose algoritmuose galima aptikti mišrių kvantinių-klasikinių loginių operacijų. 4.23 pav. parodytas grandinės pavyzdys, kuriame cX yra kontroliuojami klasikinio registro c būsenos.

Šioje grandinėje po Hadamardo vartų kubitas k_1 yra lygioje būsenų superpozicijoje, o bendra būsena:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle. \quad (4.63)$$

Atlikus būsenos matavimą su k_1 kubitu, yra lygi tikimybė rasti jį $|0\rangle$ arba $|1\rangle$ būsenoje. Šis atsakymas yra įregistruojamas klasikiniame bitų registre c kaip atitinkamai 0 arba 1 bito vertė. Jeigu bito vertė yra 1, kubitui k_2 atliekami X vartai ir pakeičia būseną į $|1\rangle$. Tad dviejų kubitų būsena tampa $|1\rangle \otimes |1\rangle$. O štai, jeigu pirmo kubito matavimo rezultatas nusako $|0\rangle$ būseną, tada



4.23 pav.: Mišri kvantinė-klasikinė loginė grandinė, kurioje k_2 kubitui pritaikomi Pauli- X , kontroliuojami klasikiniu registru

galutinė abiejų kubitų būseną lieka $|0\rangle \otimes |0\rangle$. Grandinė generuoja mišrią kvantinę būseną, kurią galima užrašyti tankio operatoriumi:

$$\rho = \frac{1}{2}|00\rangle\langle 00| + \frac{1}{2}|11\rangle\langle 11|. \quad (4.64)$$

Atkreipiame dėmesį, kad ρ yra nekoherentinė būsena, nes $1/2$ nusako klasikines tikimybes, o ne kvantines amplitudes. Tačiau, jeigu realizuotume šią grandinę daug kartų ir atliktume Pauli- Z matavimus, rezultatuose matytume idealią koreliaciją tarp kubitų būsenų. Nežinant visos loginės procedūros tokiais matavimais būtų neįmanoma pasakyti, ar stebima supintoji 2 kubitų grynoji būsena nusakoma $\rho = |\chi^+\rangle\langle\chi^+|$, ar faktorizuojamoji mišrioji ρ . Kvantinių koreliacijų nebuvinamas pasimatyti, pavyzdžiui, jeigu vietoj Pauli- Z matavimo atliktume nelokalųjį Belo tipo matavimą. Prie to sugrižime kitame poskyryje.

4.8 Bendro tipo būsenų matavimai

Projekciniai kubitų būsenų matavimai gali būti atliekami bet kokių vektorių bazėje. Niekaip ypatingai neišsiskiria standartiskai naudojama Pauli- Z bazė $\{|0\rangle, |1\rangle\}$. Kvantinės kriptografijos protokoluose dažnai aptinkami projekciniai matavimai Pauli- Z ir Pauli- X tikrinių vektorių bazėje. Norint atlikti Pauli- X matavimus praktiškai, nebūtina naudoti skirtinę fizinį matavimo įrenginį. Tam tereikia standartinius Pauli- Z bazinius vektorius transformuoti į tuos, kurių atžvilgiu norima matuoti būsenas, ir atlikti įprastinę Pauli- Z matavimą. Ekvivalentiškumas čia atsiranda dėl to, kad vidinės sandaugos modulio kvadrato reikšmė, randama skaičiuojant projekcinio matavimo tikimybes, nepriklauso nuo eiliškumo: $|\langle\psi|\chi\rangle|^2 = |\langle\chi|\psi\rangle|^2$.

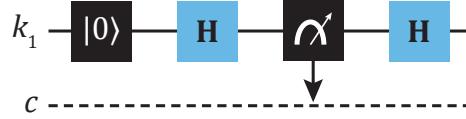
Kaip konkretų pavyzdį atlikime būsenos, išreikštostas Pauli- Z tikriniai vektoriai $|\psi\rangle = a|0\rangle + b|1\rangle$, projekcinius matavimus Pauli- X tikrinių vektorių bazėje:

$$|0_x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |1_x\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \quad (4.65)$$

Hadamardo loginiai vartai atlieka norimą transformaciją tarp šių bazinių vektorių: $H|0\rangle = |0_x\rangle$, $H|1\rangle = |1_x\rangle$. Atlikus kubito esančio $|\psi\rangle$ būsenoje projekcinį matavimą $\{|0_x\rangle, |1_x\rangle\}$ Pauli- X bazėje, tikimybės randamos $p = |\langle 0_x|\psi\rangle|^2 = |a + b|^2/2$ ir $p = |\langle 1_x|\psi\rangle|^2 = |a - b|^2/2$. Tačiau gauname pirmiausiai atlikę $|\psi\rangle$ būsenos transformaciją, $H|\psi\rangle \equiv |\psi_x\rangle$, ir toliau matuodami įprastiniu būdu Pauli- Z bazėje: $|\langle 0|\psi_x\rangle|^2 = |\langle 0_x|\psi\rangle|^2$, $|\langle 1|\psi_x\rangle|^2 = |\langle 1_x|\psi\rangle|^2$.

Kadangi pabaigoje vis tiek atliekame Pauli- Z matavimą, galutinės būsenos bus tikriniai vektoriai $|0\rangle$ arba $|1\rangle$. Todėl, jeigu norima pilnai imituoti Pauli- X matavimą, reikia atlikti dar vieną

Hadamardo transformaciją gautai būsenai po matavimo. Tokiu atveju, gavus bitą 0 arba 1, galutinė būsena bus deterministiškai pakeičiama į $|0_x\rangle$ arba $|1_x\rangle$, atitinkamai. Grandinė, atliekanti Pauli- X matavimą $|0\rangle$ būsenai, iliustruota 4.24 pav.



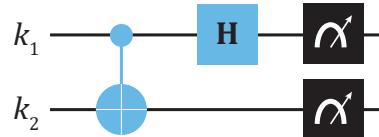
4.24 pav.: Grandinė, atliekanti Pauli- X matavimą $|0\rangle$ būsenai

Atkreipiame dėmesį į praktikoje pasitaikančius iš eilės atliekamus skirtingo tipo matavimus. Pavyzdžiui, jeigu pirmiausiai $|0_x\rangle$ kubito būsenai atliksim Pauli- X matavimą, užtikrintai rasime $|0_x\rangle$ būseną. Tačiau Pauli- Z matavime rezultatas bus būsena $|0\rangle$ arba $|1\rangle$ su $p = 0.5$ tikimybėmis. Po Pauli- Z matavimo sekantys Pauli- X matavimai taip pat suteiktų $|0_x\rangle$ arba $|1_x\rangle$ su $p = 0.5$ tikimybę, kadangi Pauli- Z matavimas pakeičia $|0_x\rangle$ būseną į $|0\rangle$ arba $|1\rangle$. Tai formaliai išplaukia iš Haizenbergo neapibréžtumo princiupo (angl. *Heisenberg uncertainty principle*), kuris teigia, kad dviejų (ar daugiau) nekomutatyvių ermitinių operatorių A ir B , $[A, B] \neq 0$, matavimuose, pirmojo operatoriaus matavimo rezultatas turi įtakos antrojo operatoriaus matavimo rezultatams. Kaip pamenename, Pauli $\{X, Y, Z\}$ operatoriai yra visi tarpusavyje nekomutatyvi.

Belo projekcinis matavimo būdas (angl. *Bell measurement*) yra svarbus siekiant unikaliai atskirti supintąsias 2 kubitų būsenas. Vien tik 1 kubito transformacijomis ir standartiniais lokaliais matavimais neįmanoma atskleisti Belo būseną. Šiame skyriuje minėjome, kad skaičiuojamuosius 2 kubitų bazinius vektorius galima unikaliai asocijuoti su Belo būsenomis. Belo matavimas yra pagristas atvirkštine kvantinio supynimo grandine, kurioje kiekviena būsena pakeičiama atgal į su ja asocijuotą 2 kubitu skaičiuojamąjį bazinį vektorių:

$$|\chi^+\rangle \rightarrow |00\rangle, \quad |\chi^-\rangle \rightarrow |10\rangle, \quad |\eta^+\rangle \rightarrow |01\rangle, \quad |\eta^-\rangle \rightarrow |11\rangle. \quad (4.66)$$

Po šios transformacijos, atlikus Pauli- Z projekcinius matavimus su abiem kubitais galima užtikrintai sužinoti, kokia tai Belo būsena. Tai yra nedestruktyvus matavimo būdas, nes po matavimo galime vėl deterministiškai atstatyti pradinę Belo būseną. Kvantinė grandinė, atliekanti Belo matavimą, yra parodyta 4.25 pav.



4.25 pav.: 2 kubitų Belo matavimas

Užbaigdami šį poskyrį sugrįžtame palyginti, kokie yra Belo matavimo rezultatai šio skyriaus ankstesniame poskyryje minėtai mišriai būsenai ρ , kuri pasižymi klasikinėmis, o ne kvantinėmis koreliacijomis kaip grynoji supintoji būsena. Atlikę mišriajai būsenai ρ Belo matavime aptinkamą cX ir H transformacijų seką randame:

$$\begin{aligned} (H \otimes I)(cX)\rho(cX)^\dagger(H^\dagger \otimes I) &= \frac{1}{2}|00\rangle\langle 00| + \frac{1}{2}|10\rangle\langle 10| \\ &= \frac{1}{2}(|0\rangle + |1\rangle)(\langle 0| + \langle 1|) \otimes |0\rangle\langle 0|. \end{aligned} \quad (4.67)$$

Akivaizdu, kad šis tankio operatorius yra faktorizuojamas $\rho = \rho_1 \otimes \rho_2$. Atlikus Pauli- Z matavimą, antras kubitas visada bus rastas $|0\rangle$ būsenoje, nes $\rho_2 = |0\rangle\langle 0|$. O štai pirmas kubitas gali būti rastas $|0\rangle$ arba $|1\rangle$ būsenoje su $p = 0.5$ tikimybe, tad galutinė dviejų kubitų būsena bus $|00\rangle$ arba $|10\rangle$. Čia ir matome skirtumą, nes supintajai grynajai būsenai $\rho = |\chi^+\rangle\langle \chi^+|$ atlikus tą pačią loginių vartų seką, kubitų pora bus visada randama $|00\rangle$ būsenoje. Galimybė aptikti $|10\rangle$ būseną pradingsta dėl destruktyvios interferencijos loginėse Belo matavimo operacijose.

4.9 Universalių loginių vartų rinkinys

Skirtingų loginių vartų rinkinys, leidžiantis įvykdyti bet kokią n kubitų registro būsenų transformaciją norimu tikslumu, yra vadinamas universaliu. Barenko teorema (angl. *Barenco theorem*) teigia, kad kvantinių loginių vartų rinkinys $\{U_3(\theta, \gamma, \varphi), cX\}$ yra universalus. Todėl vien tik 1 kubito $U_3(\theta, \gamma, \varphi)$ bei 2 kubitų cX loginių vartų kombinacijomis galima įvykdyti visus algoritmus kvantiniame kompiuteryje. Matematiškai tai reiškia, kad bet kokią $(2^n \times 2^n)$ dydžio unitarinę matricą U , nusakančią vektoriaus transformaciją $d = 2^n$ dimensijų Hilberto erdvėje, galima išskaidyti į paprastesnes $(2^n \times 2^n)$ matricas, kurių kiekviena netrivialiai veikia tik 1 kubito (2 dimensijų) ir 2 kubito (4 dimensijų) poerdvius. Jos yra vadinamos **dviejų lygių unitariosiomis matricomis** (angl. *two-level unitary matrix*). Tiksliau, galima išskaidyti $U = U_1 U_2 \cdots U_N$, čia U_i yra 2 lygių unitariosios matricos, atlikti dekompozicijai reikalingas jų skaičius N tenkina $N \leq d(d-1)/2$.

Norint įvykdyti visas įmanomas 1 kubito unitarines transformacijas $U_3(\theta, \gamma, \varphi)$ reikia, kad parametrizuoti loginiai vartai galėtų tolydžiai pasukti Blocho vektorių. Tačiau nėra žinoma, kaip atlikti tolydžias transformacijas su begaliniu tikslumu turint ribotus išteklius. Tam tikrose situacijose gali būti naudingiau pakeisti riboto tikslumo tolydžiai parametrizuotus loginius vartus diskrečiaisiais, bet tiksliau veikiančiais loginiais vartais. Solovéjaus-Kitaev teorema (angl. *Solovay-Kitaev theorem*) teigia, kad tolydžias 1 kubito transformacijas galima apytikriai pakeisti naudojant vien tik H , S , ir T loginių vartų kombinacijas. Pageidaujamai mažo dydžio bet kokių 1 kubito loginių vartų paklaida ε (angl. *arbitrarily small error*) gali būti pasiekta su $O(\log^2(1/\varepsilon))$ skaičiumi loginių vartų iš šio rinkinio. Tai bendrai yra priimtinės loginių vartų skaičiaus padidėjimas. Keturių loginių vartų rinkinys $\{H, S, T, cX\}$, vadinamas **Klifordo-T grupė** (angl. *Clifford-T group*), yra universalus ir dažnai aptinkamas praktikoje. Egzistuoja ir kitų universaliųjų rinkinių, $\{H, ccX\}$ ir $\{H, S, cX, ccX\}$.

Šiame skyriuje trumpai paminėjome Klifordo loginių vartų grupę. Klifordo grupės 1 ir 2 kubitų loginių vartų rinkinys yra sudarytas iš $\{H, S, cX\}$. Ši grupė yra plačiai naudojama kvantinių klaidų taisymo algoritmuose. Nors jais pagrįsti kvantiniai algoritmai gali sukurti daug įvairių supintųjų registro būsenų, tačiau Klifordo grupė nesudaro universalaus loginių vartų rinkinio. Būtent T loginiai vartai, įtraukti Klifordo-T grupėje, negali būti pageidaujamai tiksliai išreikšti vien su $\{H, S, cX\}$. Gotsmano-Nilo teorema (angl. *Gottesman-Knill theorem*) teigia, kad Klifordo grupe pagrįsti kvantiniai algoritmai gali būti efektyviai modeliuojami klasikiniu kompiuteriu su polinominiu laiko kompleksiškumu.

V skyrius

Kvantinė informacija ir ryšiai

5.1 Kvantinės informacijos kopijavimas

Klasikinės informacijos kopijavimas (skaitmeniniu ar kitokiu pavidalu) yra kasdieninis dalykas. Tai atliekame kopijuodami failus kompiuterio atminties laikmenose, informacijos kopijomis yra apsikeičiama tarp kompiuterių naudojant internetą. Nors skamba paradoksaliai, tačiau kvantinės informacijos kopijavimas yra fundamentaliai neįmanomas. **Uždraustojo kopijavimo teorema** (angl. *no-cloning theorem*) nusako, kad neįmanoma sukurti nežinomos bendros kvantinės būsenos identiškos kopijos (klono). Dėl šios teoremos svarbos bei gan paprastų argumentų pateiksime jos įrodyti.

Įrodymas: darome prielaidą, kad vis dėlto egzistuoja tokia unitarinė transformacija U , kuri gali sukurti identišką nežinomos kvantinės būsenos $|\psi\rangle$ kopiją. Kopijavimui atlikti naudojame kvantinį registrą, esantį sutartinėje būsenoje $|\phi\rangle$, o U veikia tarp šio registro ir kopijuojamos kvantinės sistemos. Pagal kopijavimo apibūdinimą, U formaliai atlieka:

$$U|\psi\rangle \otimes |\phi\rangle = |\psi\rangle \otimes |\psi\rangle. \quad (5.1)$$

Kitaip tariant, antrojo registro būsena yra pakeičiama į identišką pirmojo registro $|\psi\rangle$ būsenos kopiją, $|\phi\rangle \rightarrow |\psi\rangle$. Toliau imkime bet kokias dvi normuotas būsenas $|\kappa\rangle$ ir $|\tau\rangle$, kurių kopijavimą norime atlikti. Kopijuojančios unitarinės transformacijos efektas šioms būsenoms individualiai yra:

$$U|\kappa\rangle \otimes |\phi\rangle = |\kappa\rangle \otimes |\kappa\rangle, \quad U|\tau\rangle \otimes |\phi\rangle = |\tau\rangle \otimes |\tau\rangle. \quad (5.2)$$

Tolesniame žingsnyje įvertinkime $|\kappa\rangle \otimes |\kappa\rangle$ ir $|\tau\rangle \otimes |\tau\rangle$ būsenų vidinę sandaugą:

$$\begin{aligned} (\langle \kappa | \otimes \langle \kappa |)(|\tau\rangle \otimes |\tau\rangle) &= \langle \kappa | \tau \rangle^2 = \langle \phi | \otimes \langle \kappa | U^\dagger U |\tau\rangle \otimes |\phi\rangle \\ &= \langle \kappa | \tau \rangle \langle \phi | \phi \rangle = \langle \kappa | \tau \rangle. \end{aligned} \quad (5.3)$$

Viršuje, neprarasdami bendrumo, panaudojome unitarumą $U^\dagger U = I$ bei registro būsenos normuotumą $\langle \phi | \phi \rangle = 1$. Tad randame šią lygtį:

$$\langle \kappa | \tau \rangle^2 = \langle \kappa | \tau \rangle. \quad (5.4)$$

Tai yra formaliai kvadratinė lygtis, $x^2 = x$, kuri turi du sprendinius: $x = 0$ ir $x = 1$. Pirmasis sprendinys ($x = 0$) nusako, kad $|\kappa\rangle$ ir $|\tau\rangle$ yra ortogonaliosios būsenos $\langle \kappa | \tau \rangle = 0$.

Antrasis sprendinys ($x = 1$) nusako, kad jos vienodos $\langle \kappa | \tau \rangle = 1$, $\rightarrow | \kappa \rangle = | \tau \rangle$ (iki nesvarbios globalios fazės $e^{i\alpha}$). Antrasis sprendinys mūsų nedomina, nes pasirinkome skirtingas būsenas, $|\kappa\rangle \neq |\tau\rangle$. Pirmasis sprendinys rodo, kad transformacija U gali atlikti tik ortogonalijų būsenų kopijavimą ir todėl prieštarauja prielaidai, kad egzistuoja universalis unitarioji transformacija U , galinti kopijuoti bet kokią būseną.

Kvantiniu kompiuteriu galima kopijuoti klasikinę informaciją. Jau žinome tokį U , kuris sugeba kopijuoti bazinių vektorių būsenas $|0\rangle$ ir $|1\rangle$, tai cX loginiai vartai. Tačiau cX negali nukopijuoti 1 kubito būsenos, kuri yra $|0\rangle$ ir $|1\rangle$ superpozicijoje. Imkime pirmajį kubitą, esantį lygioje $|0\rangle$ ir $|1\rangle$ superpozicijoje, taip pat antrojo registro kubitą $|0\rangle$ būsenoje. Randame:

$$cX \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \quad (5.5)$$

Šiuo atveju gauname supintąją Belo būseną. Tačiau, jeigu cX iš tiesų atliktų kopijavimą iš antrojo kubito vietą, mes turėtume gauti:

$$\frac{1}{2}(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle). \quad (5.6)$$

Be unitariųjų transformacijų taip pat yra matavimų tipo nedeterministinės transformacijos. Jos irgi nėra tinkamos atlikti kopijavimą, nes matavimų rezultatai yra atsitiktiniai.

5.2 Kvantinė teleportacija

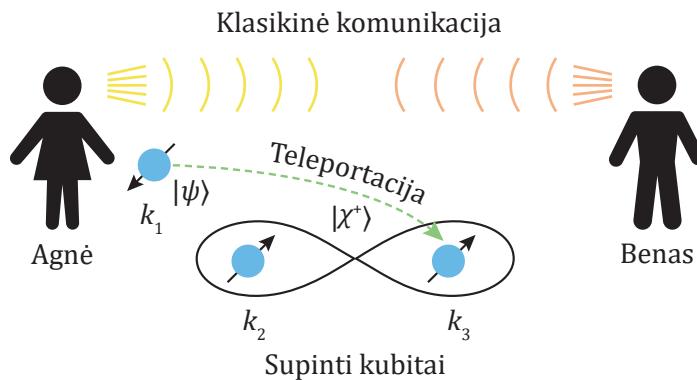
Klasikinė skaitmeninė informacija ryšiuose yra perduodama siučiant signalus tuščia erdve, elektros kabeliais bei šviesolaidžiais. Norint persiųsti kvantinę informaciją, kuri yra koduojama kubito būsenoje $|\psi\rangle = a|0\rangle + b|1\rangle$, reikia, kad perduotume amplitudžių vertes a ir b . Jeigu žinome $|\psi\rangle$ būsenos paruošimo žingsnius, tada taip pat galime klasikiniu būdu komunuoti būsenos loginių operacijų seką, kuriai atkartojojės gavėjas turės efektyviai identišką kvantinę informaciją savo kubite. Didesnis keblumas iškyla norint persiųsti informaciją, laikomą nežinomoje kvantinėje būsenoje. Šios informacijos iš esmės sužinoti negalime, tad ir ją komunuoti klasikiniu būdu neįmanoma.

Egzistuoja keletas būdų, kaip persiųsti nežinomo turinio kvantinę informaciją. Pirmasis būdas – kubitų fizinis apsikeitimas tarp lokacijų kvantinio ryšio priemonėmis. Kvantinis ryšys įprastai naudoja fotonus, dar vadinamus **skraidačiaisiais kubitais** (angl. *flying qubits*) ir, pavyzdžiui, jų poliarizacijos būsenas, kuriose koduojama informacija. Fotonus galima siušti dideliais atstumais naudojant įprastus telekomunikacijoms skirtus šviesolaidžius bei tuščia erdve. Tačiau delikačios fotonų būsenos yra lengvai pažeidžiamos siučiant juos dideliais atstumais. Atsirandanti depolarizacijos tikimybė gali pakeisti amplitudes ir įvesti klaidas iš siučiamos informacijos. Didėjant atstumui, taip pat didėja fotonų praradimo tikimybė dėl absorbcijos ir sklaidos. Kitaip nei klasikiniuose ryšiuose, siučiamo kvantinio signalo stiprinti neįmanoma dėl uždrausto kvantinių būsenų kopijavimo. Jeigu šie neigiami efektai nėra per daug žymūs, tada kvantinių klaidų taisymo algoritmai sėkmingo ryšio tikimybę galima itin padidinti.

Kvantinė teleportacija yra praktinės svarbos metodas siučiant informaciją, koduojamą nežinomoje kubitų būsenoje. Šis metodas nereikalauja tiesioginio dvipusio kvantinių ryšių kanalo tarp bendraujančių šalių, tačiau jos turi turėti kvantinį ryšį su joms bendru supintujų fotonų šaltiniu. Tai leidžia persiųsti kvantinę informaciją pasitelkiant klasikinių ryšių kanalą tarp bendraujančių šalių, o supintujų kubitu poros atlieka teleportacijos ištekliaus vaidmenį. Klasikiniu kanalu

tereikia nusiųsti po du bitus informacijos siekiant teleportuoti kiekviename kubite tolydžiai kintančiose amplitudėse koduojamą informaciją. Pirmiausiai įvardinkime, ką šiame kontekste reiškia žodis „teleportacija“.

Pagal kvantinę mechaniką dalelės, turinčios vienodas vidines fizikines savybes tokias kaip krūvis, masė ar sukinys, yra **identiškos** (angl. *identical particles*) ir negali būti atskirtos viena nuo kitos. Elektronai gali būti atskirti nuo pozitronų, nes pirmieji turi neigiamą, o antrieji – teigiamąjį krūvį; tačiau elektronai negali būti atskirti vienas nuo kito. Tad, jeigu turime du elektronus skirtinose lokacijose, tačiau identiškose sukinio būsenose, sukeitus juos vietomis fundamentaliai neįmanoma pasakyti, kad jie buvo sukeisti. Teleportacijos pavyzdys būtų, jeigu pradėdami nuo dviejų elektronų skirtinose lokacijose ir sukinio būsenose identiškai atkurtume pirmojo elektrono sukinio būseną antrajame elektrone. Teleportacija nėra ribojama vien tik vidiniams laisvės laipsniams, nors ir dažniausiai nagrinėjama jų kontekste. Šis procesas išsaugo visus fizikos principus – čia aktualūs kvantinės informacijos kopijavimo draudimas, energijos tvermė ir kad niekas negali keliauti greičiau už šviesos greitį. Fizinės sistemos akimirkniu nepradedingsta ir kitur neatsiranda – tik pakeičia savo būsenas.



5.1 pav.: Kvantinė k_1 kubito būsenos $|\psi\rangle$ teleportacija į k_3 kubitą

Standartiniame teleportacijos scenarijuje (žr. 5.1 pav.) Agnė turi kubitą k_1 būsenoje $|\psi\rangle = a|0\rangle + b|1\rangle$, $|a|^2 + |b|^2 = 1$, kurio amplitudžių a ir b ji nežino, tačiau nori $|\psi\rangle$ nusiųsti Benui. Agnė ir Benas turi galimybę tarpusavyje komunikuoti klasikiniu būdu ir tam naudoja bitus. Jie taip pat turi prieigą prie išorinio šaltinio, kuris proceso pradžioje sugeneruoja 2 kubitų supintą Belo būseną $|\chi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ ir išsiunčia pirmą kubitą (k_2) Agnėi, o antrą (k_3) Benui. Abu žino, kad yra gavę vieną iš $|\chi^+\rangle$ būsenos kubitų. Viso proceso pabaigoje Agnės $|\psi\rangle$ kubito būsena yra teleportuojama į Beno turimą k_3 kubitą.

Bendrą pradinę šių trijų kubitų būseną galime užrašyti taip:

$$\begin{aligned} |\Psi\rangle &= |\psi\rangle \otimes |\chi^+\rangle = \frac{1}{\sqrt{2}} \left[a|0\rangle \otimes (|00\rangle + |11\rangle) + b|1\rangle \otimes (|00\rangle + |11\rangle) \right] \\ &= \frac{1}{\sqrt{2}} \left[a(|000\rangle + |011\rangle) + b(|100\rangle + |111\rangle) \right]. \end{aligned} \quad (5.7)$$

Superpozicijos būsenose kubitai yra sunumeruoti taip: $|k_1 k_2 k_3\rangle$. Pradinėje būsenoje $|\Psi\rangle$ nėra nei klasikinių, nei kvantinių koreliacijų tarp Agnės k_1 kubito būsenoje $|\psi\rangle$ ir Belo poros kubitų

k_2 ir k_3 . Agnė atlieka dvejus kvantinius loginius vartus savo turimiems kubitams. Pirmiausia ji atlieka cX vartus, kuriame $|\psi\rangle$ kubitas yra „kontrolinis”. Randame naują būseną $|\Psi'\rangle$:

$$|\Psi'\rangle = (cX_{12} \otimes I)|\Psi\rangle = \frac{1}{\sqrt{2}} \left[a(|000\rangle + |011\rangle) + b(|110\rangle + |101\rangle) \right]. \quad (5.8)$$

Kitame žingsnyje ji atlieka norimam nusiušti $|\psi\rangle$ kubitui Hadamardo transformaciją:

$$\begin{aligned} |\Psi''\rangle &= (H \otimes I \otimes I)|\Psi'\rangle \\ &= \frac{1}{2} \left[a(|000\rangle + |011\rangle + |100\rangle + |111\rangle) \right. \\ &\quad \left. + b(|010\rangle + |001\rangle - |110\rangle - |101\rangle) \right]. \end{aligned} \quad (5.9)$$

Nedarant jokių kitų transformacijų, šią būseną galima pergrupuoti atskiriant Agnės ir Beno kubitus tenzorių ženklu:

$$\begin{aligned} |\Psi''\rangle &= \frac{1}{2} \left[|00\rangle \otimes (a|0\rangle + b|1\rangle) + |10\rangle \otimes (a|0\rangle - b|1\rangle) \right. \\ &\quad \left. + |01\rangle \otimes (b|0\rangle + a|1\rangle) + |11\rangle \otimes (-b|0\rangle + a|1\rangle) \right]. \end{aligned} \quad (5.10)$$

Tai leidžia lengviau pamatyti, kad Agnei atlikus dvi minėtas transformacijas Beno kubito k_3 būsena šioje trijų kubitų superpozicijoje jau primena $|\psi\rangle$. Toliau Agnė atlieka savo dvių kubitų būsenos matavimą. Ji gali rasti vieną iš keturių skirtingų dvių bitų kombinacijų su lygiomis 0.25 tikimybėmis. Pagal tai, kurią kombinaciją Agnė aptiks savo kubituose $|k_1 k_2\rangle$, tai automatiškai turės įtakos, kokia bus galutinė Beno kubito $|k_3\rangle$ būsena:

$$|k_1 k_2\rangle = |00\rangle \rightarrow |k_3\rangle = a|0\rangle + b|1\rangle; \quad (5.11)$$

$$|k_1 k_2\rangle = |10\rangle \rightarrow |k_3\rangle = a|0\rangle - b|1\rangle; \quad (5.12)$$

$$|k_1 k_2\rangle = |01\rangle \rightarrow |k_3\rangle = b|0\rangle + a|1\rangle; \quad (5.13)$$

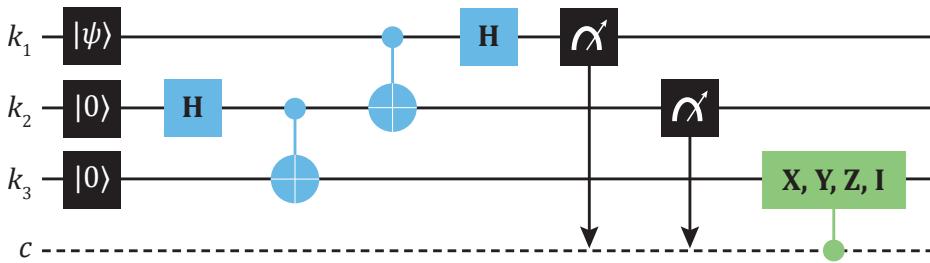
$$|k_1 k_2\rangle = |11\rangle \rightarrow |k_3\rangle = -b|0\rangle + a|1\rangle. \quad (5.14)$$

Norėdama užbaigti teleportaciją, Agnė klasikiniu kanalu nusunčia du bitus informacijos Benui pranešti, kokį rezultatą gavo. Jeigu ji rado $|00\rangle$, siunčiami bitai 00, jeigu $|01\rangle$ – bitai 01, ir atitinkamai su kitais dviem. Jeigu Benas gavo bitus 00, jam daryti nieko nereikia, $|\psi\rangle$ jau yra teleportuota ir jo „rankose”. Visais kitais atvejais Beno kubitas k_3 yra susietas su norima teleportuoti būsena $|\psi\rangle$ paprasta transformacija – Blocho vektoriaus posūkiu 180° kampu aplink x , y , arba z ašis. Pavyzdžiu, jeigu Benas gavo bitus 01, savo kubitui k_3 jis atlieka kvantinius loginius vartus X , kurie sukeičia amplitudes vietomis ir taip gaunama $|\psi\rangle$. Jeigu gauti bitai yra 10, jis atlieka Z loginius vartus, o jeigu 11 – Y vartus.

Panagrinėkime, kas šiame procese įvyko. Pirmiausiai matome, kad Agnės kubito k_1 būsena $|\psi\rangle$, esanti bendrai $|0\rangle$ ir $|1\rangle$ superpozicijoje, matavimo metu yra panaikinama. Pas ją lieka kubitai k_1 ir k_2 , esantys viename iš skaičiuojamųjų bazinių vektorių. Todėl $|\psi\rangle$ būsenos kopijavimas neįvyks ta ir Agnė nesužino $|\psi\rangle$ būsenos a ir b amplitudžių, taip patvirtinama uždraustojo kopijavimo teorema. Teleportacijoje pagrindinį vaidmenį atlieka supintoji kubitų pora, kuria Agnė ir Benas pasidalijo proceso pradžioje. Agnės atliekama cX transformacija tarp jos supinto kubito k_2 bei teleportuojamo k_1 supina ir šiuos kubitus. Taip sukuriama trijų kubitų supintoji būsena. Dėl naujai įvestų trijų kubitų kvantinių koreliacijų, Agnės atliekami Hadamardo vartai jos kubitui nelokaliai paveikia bendrą trijų kubitų būseną. Tai ir yra matoma lygtynė Beno kubito amplitudėse. Prieš Agnei atliekant savo kubitų matavimą, Beno kubitas yra superpozicijoje, sudarytoje

iš keturių skirtingų būsenų. Dėl įvestų koreliacijų, Agnės matavimas nulemia Beno kubito būsenos pasikeitimą į vieną iš šių keturių galimų. Tai galime interpretuoti kaip projekcinį Belo būsenų matavimą – antri cX ir H vartai tai formaliai realizuoja. Galutinė Beno transformacija k_3 kubitui atlieka minimalius pataisymus atstatyti $|\psi\rangle$.

Teleportacija yra praktinis būdas siųsti kvantinę informaciją ryšių tikslais ar skaičiavimams kvantinių kompiuterių tinkluose. Galime įsivaizduoti scenarijų, kuriame Agnės turimas kvantinis procesorius yra pranašesnis už Beno. Nors Beno kompiuteris turi ribotas skaičiavimų galimybes, tačiau gali patikimai atliliki Pauli- X , Y ir Z transformacijas kubitams. Tad Benas gali atliliki jam rūpimus kvantinius skaičiavimus pas Agnė kvantiniame debesyje. Parsisiųsti $|\psi\rangle$ būseną tolimesniams apdorojimui Benui tereikia bendros prieigos su Agne prie Belo būsenų generavimo šaltinio ir klasikinių ryšių kanalo. Teleportacijos metodas gali būti naudojamas persiųsti ne vien pavienių kubitų būsenoms, bet ir sudėtinėms kubitų supintosioms kvantinėms būsenoms.



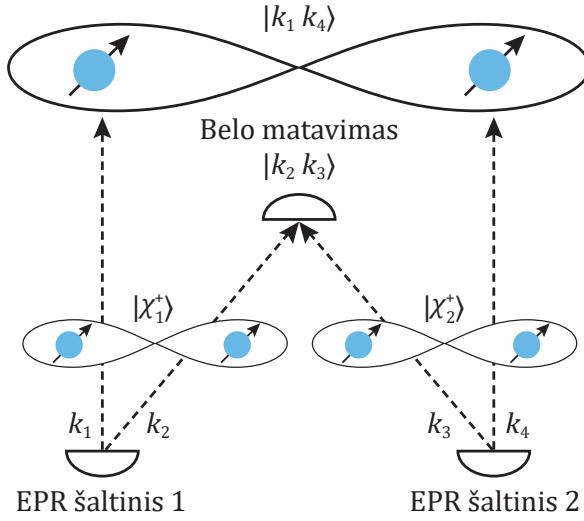
5.2 pav.: Loginė grandinė, realizuojanti kvantinę teleportaciją. Paskutiniame žingsnyje kubitui k_3 pritaikomi sąlyginiai Pauli loginiai vartai, kontroliuojami klasikinio registro, kurio būseną nulemia pirmų dviejų kubitų matavimo rezultatai

Teleportaciją galime atliliki ir kvantiniame procesoriuje tarp kubitų. Vienas būdas tai užrašyti loginiaiš vartais yra parodytas 5.2 pav. Matome Agnės kubitą k_1 pradinėje $|\psi\rangle$ būsenoje, $|\chi^+\rangle$ Belo būsenos generavimą tarp k_2 ir k_3 , bei Belo matavimą. Pagal 2 bitų kombinaciją, gautą atlukus Agnės kubitų k_1 ir k_2 matavimus, kubitui k_3 pritaikomi klasiskai kontroliuojami atitinkami sąlyginiai vartai Pauli- X , Y , Z , arba I .

5.3 Kvantinio supynimo sukeitimas

Norint atliliki supyninę tarp dviejų kubitų, paprastai yra naudojama 2 kubitų unitarinė transformacija, pavyzdžiui, cX loginiai vartai. **Kvantinio supynimo sukeitimo metodas** (angl. *entanglement swapping*) leidžia supinti vieną nuo kito nutolusius kubitus nereikalaujant jų tiesioginės tarpusavio sąveikos. Čia taip pat pasitelkiamos supintosios Belo būsenos, atliekančios svarbią rolę kvantiniuose ryšiuose, jų generatorius įprasta vadinti EPR šaltiniai (angl. *EPR source*, trumpinys nuo *Einstein-Podolsky-Rosen*).

Kvantinio supynimo sukeitimas praktiškai pritaikomas kvantiniuose tinkluose, nes gali atliliki **signalo kartotuvo funkciją** (angl. *quantum repeater*). Norint išvengti signalo sumenkimo nuostolių ir padidinti atstumą tarp kvantinės komunikacijos galutinių taškų, viena išeitis yra pastatyti tarpinius signalo kartotuvus. Gavę supintus kubitus iš nutolusią EPR šaltinių kartotuvai atlieka supynimo sukeitimą (žr. 5.3 pav.). Tai leidžia efektyviai padidinti kvantinio ryšio atstumą ir realizuoti teleportaciją ar kitus protokolus, naudojančius supintąsių būsenas.



5.3 pav.: Kvantinio supynimo sukeitimo protokolo iliustracija. Supintosios Belo būsenos kubitų porose (k_1, k_2)

Supynimo sukeitimo scenarijuje dalyvauja Agnė, Benas ir Cita. Panašiai kaip ir kvantinėje teleportacijoje, Agnė ir Benas turi pasidaliję po vieną kubitą iš supintos $|\chi_1^+\rangle$ Belo būsenos. Benas ir Cita taip pat turi po vieną kubitą iš antros sugeneruotos Belo būsenos $|\chi_2^+\rangle$. Protokolo pradžioje tarp šių dviejų Belo porų nėra jokių koreliacijų. Kvantinio supynimo sukeitimo tikslas yra supinti Agnės ir Citos kubitus. Bendrą pradinę 4 kubitų būseną galime užrašyti taip:

$$\begin{aligned} |\Psi\rangle &= |\chi_1^+\rangle \otimes |\chi_2^+\rangle = \frac{1}{\sqrt{2}} \left[(|00\rangle + |11\rangle) \otimes (|00\rangle + |11\rangle) \right] \\ &= \frac{1}{2} [|0000\rangle + |0011\rangle + |1100\rangle + |1111\rangle]. \end{aligned} \quad (5.15)$$

Superpozicijos būsenose kubitai sunumeruoti $|k_1 k_2 k_3 k_4\rangle$. Agnės kubitas k_1 yra supintas su Beno kubitu k_2 , Beno kubitas k_3 yra supintas su Citos kubitu k_4 . Stebint kvantinės teleportacijos žingsnius, Benas atlieka cX loginius vartus tarp savo turimos kubitų poros k_2 ir k_3 , kuriuose kubitas k_2 atlieka „kontrolinę“ rolę. Randame naują būseną $|\Psi'\rangle$:

$$|\Psi'\rangle = (I \otimes cX_{23} \otimes I) |\Psi\rangle = \frac{1}{2} [|0000\rangle + |0011\rangle + |1100\rangle + |1101\rangle]. \quad (5.16)$$

Kitame žingsnyje Benas atlieka Hadamardo transformaciją savo kubitiui k_2 :

$$|\Psi''\rangle = (I \otimes I \otimes H \otimes I) |\Psi'\rangle. \quad (5.17)$$

Norėdami lengviau pamatyti rezultatą, sugrupuojame narius $|\Psi''\rangle$ skliausteliuose:

$$\begin{aligned} |\Psi''\rangle &= \frac{1}{\sqrt{8}} \left[(|0\rangle \otimes |00\rangle \otimes |0\rangle + |1\rangle \otimes |00\rangle \otimes |1\rangle) \right. \\ &\quad + (|0\rangle \otimes |01\rangle \otimes |1\rangle + |1\rangle \otimes |01\rangle \otimes |0\rangle) \\ &\quad + (|0\rangle \otimes |10\rangle \otimes |0\rangle - |1\rangle \otimes |10\rangle \otimes |1\rangle) \\ &\quad \left. + (|0\rangle \otimes |11\rangle \otimes |1\rangle - |1\rangle \otimes |11\rangle \otimes |0\rangle) \right]. \end{aligned} \quad (5.18)$$

Kiekvienoje šios superpozicijos būsenoje 4 kubitai yra užrašyti šia forma $|k_1\rangle \otimes |k_2 k_3\rangle \otimes |k_4\rangle$. Benas galiausiai atlieka savo abiejų kubitų matavimą, taigi gali rasti vieną iš keturių kombinacijų $|k_2 k_3\rangle$ su lygiomis 0.25 tikimybėmis. Matome, kad pagal jo rastą kubitų būseną galutinė Agnės ir Citos kubitų būsena $|k_1 k_4\rangle$ pasikeičia:

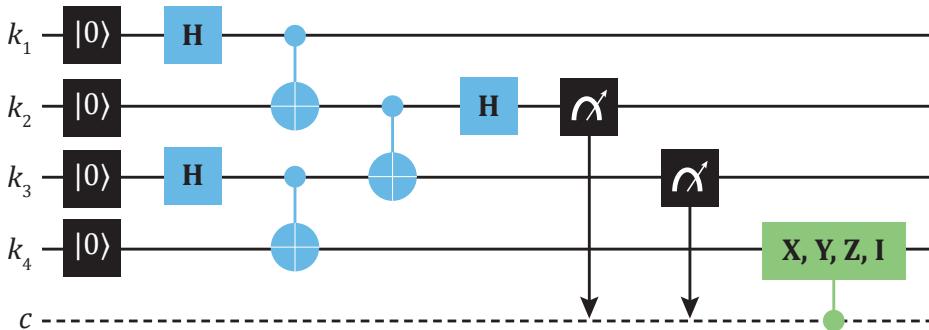
$$|k_2 k_3\rangle = |00\rangle \rightarrow |k_1 k_4\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\chi^+\rangle; \quad (5.19)$$

$$|k_2 k_3\rangle = |01\rangle \rightarrow |k_1 k_4\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) = |\eta^+\rangle; \quad (5.20)$$

$$|k_2 k_3\rangle = |10\rangle \rightarrow |k_1 k_4\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = |\chi^-\rangle; \quad (5.21)$$

$$|k_2 k_3\rangle = |11\rangle \rightarrow |k_1 k_4\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) = |\eta^-\rangle. \quad (5.22)$$

Agnės ir Citos kubitai $|k_1 k_4\rangle$ visais atvejais lieka vienoje iš supintųjų Belo būsenų. Benas matavimo rezultatus perduoda Agnei ir Citai klasikiniu būdu; tai leidžia sužinoti, kokią supintąją būseną jos turi. Kad Agnė ir Cita pakeistų savo gautą būseną į bet kurią kitą Belo būseną, jos gali atlikti atitinkamas lokalias Pauli transformacijas savo turimiems kubitams. Atkreipame dėmesį, kad šio proceso metu kvantiniai supynimai tarp Agnės ir Beno kubitų poros k_1 ir k_2 , taip pat Beno ir Citos kubitų k_3 ir k_4 yra panaikinami.



5.4 pav.: Kvantinio supynimo sukeitimo realizacija loginėje grandinėje

Kvantinė grandinė, atliekanti supynimo sukeitimo algoritmą, yra pavaizduota 5.4 pav. Grandinės pabaigoje įdėti klasikai kontroliuojami loginiai vartai, atliekantys k_4 kubito transformaciją. Ši transformacija leidžia pakeisti Agnės ir Citos supintąją būseną į bet kurią kitą Belo būseną. Kaip matysime vėliau šiame skyriuje, pakanka vienos lokaliosios transformacijos, atliekamos bet kuriam kubitui supintoje poroje, norint pakeisti jų bendrą Belo būseną į bet kurią kitą Belo būseną.

5.4 Kvantinė kriptografija

Įsivaizduokime scenarijų, kuriamė Agnė ir Benas ketina apsikeisti svarbia informacija. Norint užtikrinti, kad Evelina, kuri yra slapukavusi tarp jų anksčiau, nepamatytų ryšių turinio, Agnė ir Benas nusprendžia jį užšifruoti. Tokiu atveju, nors Evelina ir perimtų siunciāmą turinį, ji negalėtų suprasti, kas Jame sakoma. Siekdamai užšifruoti turinį Agnė ir Benas naudoja raktą.

Kaip paprastą to pavyzdį imkime, kad turinys (t) yra išreikštas dvejetainė forma $t = 001011010$, o šifravimas atliekamas sudedant kiekvieną turinio bitą modulo(2) su atitinkamu raktu $r = 011101100$ bitu. Taip turinys pasikeičia į šią šifruotą seką: $t \oplus r = 0101110110$. Norint turinį iššifruoti, tereikia vėl mod(2) sudėti šifruotą turinį su tuo pačiu raktu, nes $(t \oplus r) \oplus r = t$.

Agnė ir Benas gali pasirinkti, ar jie naudos privataus, ar atvirojo raktu kriptografinę sistemą. Privataus raktu kriptografijoje Agnė ir Benas susitikę pasirenka raktą, arba paprašo, kad raktą jiems perduotų trečiasis asmuo, Cita. Nors privataus raktu metodas yra saugus (naudojant pakankamai ilgą raktą), jam reikalingas apsikeitimasis nėra praktiškas ir turi savo saugumo spragų. Pavyzdžiu, gali būti neįmanoma susitikti apsikeisti raktu ar tai atlikti kiekvieną kartą prieš inicijuojant ryšį. Antraip jie dar turėtų užtikrinti ilgai laikomo ir naudojamo raktu saugumą nuo įsibrovimų. Trečiojo asmens naudojimas apsikeitimui irgi neužtikrina saugumo, nes Cita gali raktą pasidalinti ar perduoti Evelinai.

Atvirojo raktu kriptografijos metodas buvo sukurtas 1970-aisiais ir yra plačiai taikomas interneto tinkluose. Šiuo metodu naudojamas asimetrinis raktu pasidalijimas. Norėdama suteikti Benui galimybę saugiai nusiųsti informaciją jai, Agnė atvirai paskelbia raktą, kuriuo reikia užšifruoti siunčiamą turinį. Tačiau Agnė pas save turi kitą raktą, žinomą tik jai, kuriuo galima iššifruoti turinį užšifruotu jos viešai paskelbtu raktu. Šie raktai yra sugeneruojami automatiškai kiekvienos sesijos metu, ir tam naudojamas atsitiktinių skaičių generatorius. Atvirojo raktu kriptografijos saugumas yra pagrįstas matematiškai sunkiai apskaičiuojamomis funkcijomis. Vis dėlto yra parodyta, kad plačiai paplitusi **RSA kriptografijos sistema** (angl. *Rivest-Shamir-Adleman*), pagrįsta pirminių skaičių faktorizacija, yra efektyviai įveikiama Šoro algoritmu kvantiniuose kompiuteriuose. Agnė ir Benas tiki, kad Evelina dar neturi pakankamai galingo kvantinio kompiutero įveikti RSA, ir todėl yra linkę naudoti ši kriptografijos protokolą. Tačiau, jeigu Evelina tokį įrenginį turės artimoje ateityje, perimtą Agnės ir Beno ryšį ji galės nuskaityti ir vėliau.

Trečiasis būdas, apie kurį sužinojo Agnė ir Benas, yra naudoti privatų **kvantinį raktu pasidalijimo protokolą** (angl. *quantum key distribution*, trumpinys QKD). Šiame protokole užšifruoti ir iššifruoti turiniui taip pat naudojamas klasikinis raktas (dvejetainis kodas), tačiau raktu pasidalijimui yra naudojami kvantiniai ryšiai. Privatų raktu pasidalijimą jie gali atlikti per atvirą kanalą nebijdami, kad Evelina raktą sužinos, kadangi kvantiniuose ryšiuose raktu atskleidimas pakeičia raktą, na, o kopijuoti kvantinio raktu fundamentaliai neįmanoma. Atlikę kvantinį raktu pasidalijimą jie gali toliau naudoti klasikinį ryšių kanalą simetriškai užšifruodami ir iššifruodami siunčiamą turinį. Toliau pateikiame BB84 ir EPR kvantinius raktu pasidalijimo protokolus.

5.4.1 BB84 kvantinis raktu pasidalijimo protokolas

Dvejetainio raktu persiuntimui **BB84** (angl. *C. Bennet ir G. Brassard, 1984*) protokole yra naudojamas vienpusis kvantinis kanalas nuo Agnės iki Beno ir klasikinis dvipusis ryšių kanalas. Klasikinis kanalas yra viešas ir gali būti pasiklausomas, kvantiniu kanalu siunčiama informacija taip pat gali būti perimta. Kaip matysime, tai netrukdo inicijuoti saugų ryšį.

Jeigu norimo raktu ilgis yra n bitų, tada Agnė, pirmiausiai naudodama atsitiktinių skaičių generatorių, sugeneruoja dvi $4n$ bitų ilgio sekas, kurias vadinsime a_A ir b_A . Pirmoji bitų seka a_A nusako patį raktą, o antroji b_A – kokią šifravimo sistemą naudoti kiekvienam a_A raktu bitui. Šias dvi vienodo ilgio sekas galima sugrupuoti poromis $\{a_A, b_A\} = \{(a_1, b_1), (a_2, b_2), \dots, (a_{4n}, b_{4n})\}$, pagal kurias Agnė paruoš $4n$ kubitų siuštį Benui. Jeigu b_A bitas yra 0, tada išreikšti a_A bitui (kurio sugeneruota vertė yra 0 arba 1) Agnė taiko Pauli- Z bazinių vektorių šifravimą. Tai yra, jeigu raktu bito vertė yra $a_A = 0$, jis perteikiamas kubito $|0\rangle$ būsenai, o $a_A = 1$ bitas perteikiamas būsenai $|1\rangle$. Jeigu šifravimo bitas $b_A = 1$, Agnė išreiškia atitinkamą a_A raktu bitą Pauli- X

baziniai vektoriai $\{|0_x\rangle, |1_x\rangle\}$. Taip Agnė nusunčia Benui $4n$ kubitų $|\psi_{ab}\rangle$, kurių kiekvienas yra vienoje iš šių būsenų:

$$|\psi_{00}\rangle = |0\rangle; \quad (5.23)$$

$$|\psi_{10}\rangle = |1\rangle; \quad (5.24)$$

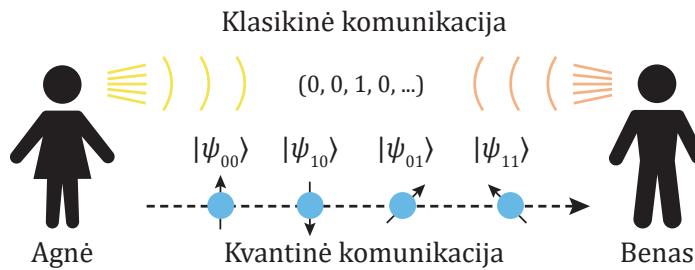
$$|\psi_{01}\rangle = |0_x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle); \quad (5.25)$$

$$|\psi_{11}\rangle = |1_x\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \quad (5.26)$$

Gavęs kubitus Benas neturi informacijos, kokiose būsenose jie yra. Benas imasi atlikti kubitų būsenų matavimus ir atsitiktiniu būdu pasirenka atlikti Pauli- Z arba Pauli- X projekcinius matavimus. Kaip minėjome 4 skyriuje, norint atlikti Pauli- X projekcinį matavimą tereikia atlikti gautiems kubitams Hadamardo transformaciją ir toliau jiems vykdyti standartinį (Pauli- Z) matavimą. Atkreipiame dėmesį, kad šios keturių kubitų būsenos nėra visos viena kitai ortogonalios, todėl negali būti patikimai atskirtos atliekant tik vieno tipo matavimą. Matome, kad (atsitiktinai) atlikęs Pauli- Z matavimą kubitams, esantiems $|\psi_{00}\rangle$ arba $|\psi_{10}\rangle$ būsenose, Benas (pats nežinodamas) teisingai išmatuoja šias būsenas ir gauna sutapimą su jais, koduojamais Agnės raktu a bitais. Tačiau, jeigu būsenos yra $|\psi_{01}\rangle$ arba $|\psi_{11}\rangle$, tikimybė, kad atlikdamas Pauli- Z matavimą Benas teisingai iššifruos raktu bitus, téra 0.5, kadangi $|0_x\rangle$ ir $|1_x\rangle$ yra lygios $|0\rangle$ ir $|1\rangle$ superpozicijos. Analogiška situacija atsiranda, kai Benas atlieka Pauli- X matavimą. Jis gauna sutapimus su atitinkamais Agnės raktu bitais, jeigu matuoja $|\psi_{01}\rangle$ ir $|\psi_{11}\rangle$ būsenas, tačiau atsiranda 0.5 raktu iššifravimo paklaidos tikimybė matuojant $|\psi_{00}\rangle$ ir $|\psi_{10}\rangle$ būsenas. Taip atlikęs matavimus Benas sugeneruoja $\{a_B, b_B\}$ bitų porų seką, kurioje a_B bitas (0 arba 1) nusako gautą būsenos matavimo rezultatą, o b_B bitas užrašo, ar šis matavimas naudojo Pauli- Z (bito vertė 0) ar Pauli- X (bito vertė 1).

Kitame žingsnyje Agnė ir Benas, komunikuodami per atvirą klasikinį kanalą, palygina šifravimo $4n$ bitų b_A ir b_B sekas. Taip jie turėtų rasti, kad tarp jų bitų vidutiniškai $2n$ buvo atsitiktiniu būdu pasirinkti vienodai. Tai reiškia, kad Beno naudojamas matavimo būdas bei Agnės bitų šifravimo būdas šiaisiai $2n$ atvejais sutapo, ir jie abu žino, kad buvo teisingai iššifruoti šie atitinkamai a raktu bitai. Jie atsikrato a_A ir a_B bitų, kurių porose esantys b_A ir b_B nesutampa ir pasiliauka likusių $2n$ raktų seką ($a_A = a_B$) neatskleisdami jos.

Norėdami patikrinti, kad Evelina neslapukavo perimdama Agnės siunčiamus kubitus, jiedu atsitiktinai pasirenka iš turimų $2n$ raktų n bitų ir per klasikinį kanalą palygina, ar jie sutampa. Jeigu priimtinas bitų skaičius sutampa, jiedu užbaigia raktu apsikeitimo protokolą ir gali saugiai naudoti likusius n bitų šifruoti ryšių turiniui.



5.5 pav.: BB84 protokolo iliustracija

Panagrinėkime BB84 protokolo (žr. 5.5 pav.) saugumą. Pirmiausia, uždraustojo kvantinių būsenų kopijavimo teorema garantuoja, kad Evelina negali patikimai kopijuoti Agnės siunčiamų kubitų. Jeigu tai būtų įmanoma, turėdama siunčiamų kubitų būsenų kopijas ir perėmus b_A ir b_B sekų komunikavimą tarp Agnės ir Beno ji galėtų atkurti raktą bei turinį. Sakykime, kad Evelina visgi pamėgina atlikti Agnės siunčiamų kubitų kopijavimą naudodama cX vartus. Jos pradinės kubitų būsenos yra $|0\rangle$, o po cX vartų ji toliau persiunčia Agnės kubitus Benui. Po šios transformacijos 2-kubitų būsenos yra:

$$cX|\psi_{00}\rangle = |00\rangle ; \quad (5.27)$$

$$cX|\psi_{10}\rangle = |11\rangle ; \quad (5.28)$$

$$cX|\psi_{01}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) ; \quad (5.29)$$

$$cX|\psi_{11}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) . \quad (5.30)$$

Evelina laukia, kol Benas atliks kubitų matavimą, galiausiai tai paveikia ir jos kubitų būseną. Toliau ji perima Agnės ir Beno klasikinę komunikaciją, kurioje jiedu atskleidžia, kokie baziniai vektoriai buvo naudojami šifruoti raktui ir atlikti matavimams. Evelina kartu su Agne ir Benu atmeta tuos kubitus, kuriuose b_A ir b_B nesutampa. Jos strategija yra toliau atlikti tokius pačius matavimus su savo turimais kubitais, kuriuos atliko Benas. Matome, kad Evelina sėkmingai atkuria $|\psi_{00}\rangle$ ir $|\psi_{10}\rangle$ būsenas, kurioms Benas atliko Pauli- Z matavimus. Deja, kvantinis supynimas paveikia Benui $|\psi_{01}\rangle$ ir $|\psi_{11}\rangle$ būsenų matavimo rezultatus, kurioms teisingai atkurti jis (atsitiktinai) pasirinktu Pauli- X matavimuis. Pažvelkime į galimus Beno matavimo rezultatus, prieš tai atlikę H transformaciją:

$$\begin{aligned} (H \otimes I)|\psi_{01}\rangle &= \frac{1}{\sqrt{2}} \left[(|0\rangle + |1\rangle) \otimes |0\rangle + (|0\rangle - |1\rangle) \otimes |1\rangle \right] \\ &= \frac{1}{\sqrt{2}} \left[|0\rangle \otimes (|0\rangle + |1\rangle) + |1\rangle \otimes (|0\rangle - |1\rangle) \right] . \end{aligned} \quad (5.31)$$

Antroje eilutėje pergrupavome narius, norėdami parodyti, kad yra 0.5 tikimybė, jog Benas atlikdamas matavimus ras $|0_x\rangle$ arba $|1_x\rangle$, nes $|0\rangle$ ir $|1\rangle$ yra lygios $|0_x\rangle$ ir $|1_x\rangle$ superpozicijos. Tačiau atlikdamas teisingą matavimą Agnės siūstai $|\psi_{01}\rangle$ būsenai jis turėtų rasti $|0_x\rangle$ kiekvieną kartą. Šis nesutapimas tarp jūdviejų turimų a_A ir a_B bitų bus aptiktas Agnei ir Benui atliekant atsitikinių a raktų bitų palyginimą protokolo pabaigoje. Tad nors Evelina šioje BB84 protokolo atakoje sugeba teisingai atkurti kiekvieną iš keturių skirtinų būsenų, kurias Benas randa pas save, tačiau kvantinis supynimas pakeičia Beno rezultatus ir įveda neatitikimus.

Kadangi unitariosios transformacijos neatitinka norimo tikslų, antra galima Evelinos taktika – naudoti matavimo tipo transformacijas. BB84 protokolas uždaro ir šią spragą, kadangi naujoja neortogonalias kvantines būsenas. Nežinodama, kokį matavimo būdą naudoti, Evelina gali pasirinkti, pavyzdžiu, visus perimtus Agnės kubitus pamatuoti Pauli- Z projekcija. Atlikus matavimą Evelina šiuos kubitus toliau persiunčia Benui, norėdama neišsiduoti, kad pasiklauso ryšio. Tačiau Pauli- Z matavimas, atliekamas $|0_x\rangle$ ir $|1_x\rangle$ būsenoms, neleidžia atskleisti, kokia yra būsena, kadangi atsitiktinai randamas $|0\rangle$ arba $|1\rangle$ su vienodomis tikimybėmis. Dar blogiau, kad Evelinos Pauli- Z matavimas negrįztamai pakeičia $|0_x\rangle$ ir $|1_x\rangle$ kubitų būsenas į $|0\rangle$ arba $|1\rangle$. Agnė ir Benas, atlikdami atsitiktinai pasirinktų raktą bitų palyginimą protokolo pabaigoje, gali įvertinti, ar jų buvo pasiklausoma ir nutraukti arba kartoti protokolą iš naujo esant dideliam bitų nesutapimui.

5.4.2 EPR kvantinis raktų pasidalijimo protokolas

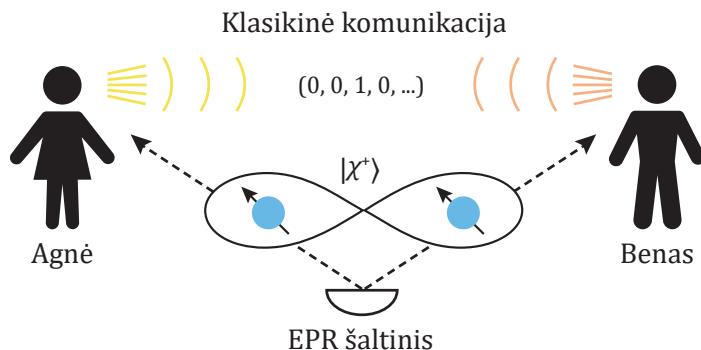
EPR protokolas (žr. 5.6 pav.) yra dauguma aspektų panašus į BB84, tačiau tarp Agnės ir Beno nėra tiesioginio kvantinio ryšio kanalo. Vietoj jo naudojamas EPR šaltinis, paskirstantis tarp jų supintuosius kubitus. EPR protokole šis šaltinis paruošia $4n$ supintąsių $|\chi^+\rangle$ kubitų poras ir iš kiekvienos poros nusiunčia po vieną kubitą Agnėi ir Benui. Pradžioje jie abu turi po $4n$ supintujų kubitus, kurių kiekvienas yra būsenoje:

$$|\chi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \quad (5.32)$$

Agnė ir Benas atsitiktiniu būdu pasirenka išmatuoti savo visus turimus kubitus santykinių su Pauli-Z arba Pauli-X bazinių vektoriais. Taip jie sugeneruoja bitų sekas $\{a_A, b_A\}$ ir $\{a_B, b_B\}$; čia b bitas nusako, ar buvo pasirinktas Pauli-Z (0) ar Pauli-X (1) matavimas, o a bitas nusako gautą rezultatą (0 arba 1). Iš to matome, kad jeigu Agnės ir Beno atsitiktiniai matavimo būdo pasirinkimai sutapo, šių kubitus porų būsenas jie visada ras vienodas, ir todėl jų atitinkami raktai sutaps. Tai akivaizdu žvelgiant į $|\chi^+\rangle$ ir naudojant Pauli-Z matavimus. Norėdami atlikti Pauli-X matavimus abiem kubitams $|\chi^+\rangle$ Belo būsenoje pirmiausia atlikime Hadamardo vartus:

$$\begin{aligned} H \otimes H |\chi^+\rangle &= \frac{1}{2\sqrt{2}} \left[(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) + (|0\rangle - |1\rangle) \otimes (|0\rangle - |1\rangle) \right] \\ &= \frac{1}{\sqrt{2}} [|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle]. \end{aligned} \quad (5.33)$$

Matome, kad Pauli-X matavimuose taip pat yra ideali koreliacija tarp dviejų kubitus būsenų, tad atlikus norimus matavimus jų vertės visada sutaps. Tolimesni žingsniai paremti BB84 protokolu: Agnė ir Benas per klasinį kanalą palygina b_A ir b_B šifravimo bitus ir pasilieka tuos $2n$ raktų bitus, kuriems šifravimo bitai sutapo, $a_A = a_B$. Galiausiai jiedu atsitiktinai pasirenka iš turimų $2n$ raktų n bitų ir palygina, ar jie sutampa, siekdami įvertinti, ar ryšys patikimas.



5.6 pav.: EPR protokolo iliustracija

Šiuo atveju Evelina gali pabandyti įsiterpti perimdama abu kubitus poroje, skirtus Agnėi ir Benui. Evelina atlieka kubitus būseną matavimą ir toliau jiems persiunčia jau nebesupintus, o klasiskai koreliuotus kubitus, taip tiksliai žinodama, kokie bus Agnės ir Beno rezultatai. Tačiau Evelinos įsiterpimas pakeičia jų rezultatus. Pavyzdžiu, jeigu Evelina atlikusi Pauli-Z matavimus randa $|00\rangle$ kubitus būseną, tada Agnė ir Benas, atlikę Pauli-Z neaptinka nesutapimų. Na, o jeigu jie atsitiktinai abu pasirenka Pauli-X matavimą $|00\rangle$ būsenai, tada 0.5 jų matavimų rezultatai

nesutaps, nes yra lygi tikimybė rasti $|0_x\rangle$ arba $|1_x\rangle$ būsenas. Atlikdami šifravimo ir rakto bitų patikrą Agnė ir Benas tai gali pastebėti ir nutraukti protokolą.

Agnės ir Beno matavimo rezultatų nesutapimai gali atsirasti ir dėl kitų išorinių veiksnių, kurie įveda ryšio signalui triukšmą. Praktikoje jiems reikia būdo patikrinti supintosios kvantinės būsenos tikslumą. Kituose poskyriuose smulkiau panagrinėsime supintujų kubitų savybes ir aptarsime, kaip įmanoma patikrinti būsenų tikslumą, taip pat ar jos pasižymi ryšiuose pageidaujamomis kvantinėmis koreliacijomis.

5.5 Lokalios operacijos ir klasikiniai ryšiai

EPR šaltiniai kvantiniuose tinkluose yra nepamainomi, ir dėl to svarbu įvertinti jų tikslumą. Jeigu yra galimybė tai atlikti kubitams vos palikus EPR šaltinį, tada galime apskaičiuoti **būsenų tikslumą** (angl. *state fidelity*) palygindami sugeneruotą 2-kubitų būseną su grynaja Belo būsenai $|\psi\rangle$, kuri, tikimasi, turėtų būti ir EPR šaltinio sugeneruota. Vienas būdas tai apskaičiuoti:

$$F[\rho, |\psi\rangle] = \sqrt{\langle\psi|\rho|\psi\rangle}. \quad (5.34)$$

Tikslumas F nusako būsenų persiklojimą ir yra apibrėžtas intervale $0 \leq F \leq 1$. $F = 0$ reiškia, kad būsenos yra ortogonalios (maksimaliai skirtinges), o $F = 1$ – kad jos fiziškai vienodos. Čia išreiškiame EPR sugeneruotą būseną tankio operatoriumi $\rho = |\phi\rangle\langle\phi|$, nes dėl paruošimo netikslumų ar kitų mums nežinomų veiksnių ji gali būti mišri. Matome, kad tikslumas F nusako persiklojimo tarp būsenų ρ ir $|\psi\rangle$ šaknį.

Supintujų kubitų patikrinimą taip pat galima atlikti teleportuojant Beno kubitą pas Agnė, šitaip ji atliktų Belo matavimą. Tačiau tam reikalinga dar viena Belo būsena. Jeigu Agnė ir Benas yra galutiniai Belo būsenų vartotojai, neturintys tokios galimybės, tada patikrinti šaltinio tikslumui ar kvantinio ryšio kanalo švarumui jiems reikalingas kitas būdas. Imkime standartinį scenarijų, kuriame Agnė ir Benas neturi tarpusavyje kvantinio kanalo ir gali savo individualiems kubitams, gautiems iš EPR šaltinio, atlikti tik lokalias unitarišias transformacijas bei matavimus ir tarpusavyje komunikuoti klasikiniu būdu. Pavyzdžiui, atlikęs norimą matavimą Benas gali rezultatą pranešti Agnei. Pagal gautą rezultatą, ji savo ruožtu pasirenka norimą transformaciją ar matavimo būdą, siekdama sužinoti kuo daugiau informacijos apie jų turimą kvantinę būseną. Tai yra vadinamasis **lokalių operacijų ir klasikinių ryšių metodas kvantinėje informatikoje** (angl. *local operations classical communication*, trumpinys LOCC). Toliau aptarkime, kokias LOCC operacijas Agnė ir Benas gali atlikti norėdami patikrinti keturias Belo būsenas:

$$|\chi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \quad |\chi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle); \quad (5.35)$$

$$|\eta^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \quad |\eta^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \quad (5.36)$$

Kaip ir skaičiuojamieji 2 kubitų baziniai vektoriai $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, Belo baziniai vektoriai $\{|\chi^+\rangle, |\chi^-\rangle, |\eta^+\rangle, |\eta^-\rangle\}$ kiekvienas individualiai savyje koduoja maksimaliai du bitus klasikinės informacijos. Standartiniame rinkinyje kiekvieną iš dviejų kubitų unikalai nusako jų reikšmės 0 ir 1. Norint nusakyti supintąsias Belo būsenas akivaizdu, kad tai netinka dėl esamų superpozicijų. Belo rinkinyje pirmasis bitas nusako **būsenų lyginumą** (angl. *parity bit*), kuris parodo, ar kubitų būsenos nariuose vienodos, ar skirtinges. Lyginį lyginumą (angl. *even parity*) turi $|\chi^+\rangle$ ir $|\chi^-\rangle$ būsenos, nes abu kubitai jų nariuose $|00\rangle$ ir $|11\rangle$ yra lygūs. Nelyginius lyginumus (angl. *odd parity*) turi $|\eta^+\rangle$ ir $|\eta^-\rangle$, sudaryti iš $|01\rangle$ ir $|10\rangle$ narių superpozicijos nelygiose būsenose. Antrasis

bitas informacijos Belo būsenose yra vadinamas **fazės bitu** (angl. *phase bit*). Būsenos $|\chi^+\rangle$ ir $|\eta^+\rangle$ turi tokį patį fazės bitą (+), nes nariai yra sudedami (santykinė fazė lygi nuliui). Būsenose $|\chi^-\rangle$ ir $|\eta^-\rangle$ nariai yra atimami (santykinė fazė π) ir todėl jų fazės bitas yra priešingas $|\chi^+\rangle$ ir $|\eta^+\rangle$ būsenoms (-). Tad norėdami nusakyti Belo būsenas dviem bitais informacijos galime susieti lyginį ir nelyginį lyginumą su pirmo bito 0 ir 1 vertėmis, taip pat + ir - fazės bitus su antro bito 0 ir 1 vertėmis. Tai leidžia unikaliai įvardyti visas keturias Belo būsenas.

Kitas svarbus aspektas yra tai, kad supintosiose Belo būsenose lyginumo ir fazės bitus galima keisti atliekant vien lokaliąsias transformacijas vienam iš pasirinktų kubitų, nepaisant atstumo tarp dviejų kubitų. Pavyzdžiui, Agnė, turinti $|\chi^+\rangle$ būseną, gali pakeisti lyginumo bitą iš 0 į 1 atlikdama savo kubitui Pauli-X transformaciją. Arba tą patį gali padaryti Benas. Norėdami pakeisti fazės bitą iš 0 į 1, bet kuris iš jų atliktų Pauli-Z transformaciją. Pauli-Y transformacija pakeistų lyginumo ir fazės bitus kartu. Apačioje pateikiame lyginumo ir fazės bitų keitimo transformacijas pirmam kubitui $|\chi^+\rangle$ būsenoje:

$$(X \otimes I)|\chi^+\rangle = |\eta^+\rangle; \quad (5.37)$$

$$(Y \otimes I)|\chi^+\rangle = -i|\eta^-\rangle; \quad (5.38)$$

$$(Z \otimes I)|\chi^+\rangle = |\chi^-\rangle. \quad (5.39)$$

Primename, kad globali būsenos fazė nėra svarbi, tad $-i|\eta^-\rangle$ ir $|\eta^-\rangle$ nusako identiškas kvantines būsenas. Tad vien lokaliomis transformacijomis viena iš dalyvaujančių šalių gali Belo būseną pakeisti į bet kurią kitą Belo būseną. Atkreipiame dėmesį, kad lokaliųs unitarinės transformacijos negali panaikinti supynimo, tad pradedant iš Belo būsenų visos taip pasiekiamos būsenos yra supintosios. Ši rezultatą galima palyginti su faktorizuojamomis būsenomis standartiniuose baziniuose vektoriuose, pavyzdžiui, $|00\rangle$. Agnė gali keisti savo turimo kubito vertę tarp 0 ir 1, todėl jos lokaliųs transformacijos leidžia pasiekti $|00\rangle$ ir $|10\rangle$ būsenas. Tačiau be Beno pagalbos ji negali globaliai pakeisti $|00\rangle$ būsenos į $|01\rangle$ arba $|11\rangle$. Individualiomis lokaliomis transformacijomis nesupintojoje būsenoje šiuo atveju įmanoma keisti tik vieną bitą informacijos, o ne du, kaip supintosiose.

Grįžtant prie būsenų patikros, imkime situaciją, kurioje Agnė ir Benas, turėdami po vieną kubitą iš nežinomos Belo būsenos ir taikydamai LOCC metodą nori sužinoti, kokia tai būsena. Pradėdama paprasčiausiu būdu, Agnė pasirenka atsitiktinai pamatuoti savo kubitą naudodama $P = |0\rangle\langle 0| \otimes I$. Dėl šios priežasties nežinoma Belo būseną pasikeis į $|00\rangle$ arba $|01\rangle$. Akivazdu, kad nežiūrint, kokį matavimą pasirinks Benas, jie galės pasakyti tik turimos Belo būsenos lyginumo bitą. Fazės bito šiaisiai matavimais jie sužinoti negali, tad ir įvardyti Belo būsenos nepavyks. Agnė ir Benas vis dėlto gali sužinoti supintosios būsenos lyginumo ir fazės bitus LOCC metodu paaukodami dvi supintąsias būsenas, o ne vieną. Tam darome prielaidą, kad EPR šaltinis siunčia identiškas būsenas. Gavę po vieną kubitą iš pirmosios Belo būsenos poros, jie abu atlieka Pauli-Z matavimą savo kubitams, o tikrines vertes sudaugina. Primename, kad atliekant Pauli-Z matavimus individualiems kubitams galimos tikrinės vertės yra +1 ($|0\rangle$ būsena) arba -1 ($|1\rangle$ būsena). Matome, kad jeigu Belo būsenos lyginumas yra lyginis, tada Pauli-Z tikrinių verčių sandauga yra 1, o jeigu nelyginis, gaunama -1. Naudojant Pauli-X matavimą antrajai (identiškai) Belo būsenai jie gali rasti ir fazės bitą. Šiuo atveju, jeigu būsenos fazė yra 0, tada tikrinių verčių sandauga yra 1, o jeigu π – gaunama -1. Norėdami pademonstruoti fazės bitų matavimą imkime $|\chi^+\rangle$ ir $|\chi^-\rangle$ būsenas. Pauli-X matavimas atliekamas pirmiausia pritaikius

abiems kubitams Hadamardo vartus:

$$(H \otimes H)|\chi^+\rangle = \frac{1}{\sqrt{2}}[|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle]; \quad (5.40)$$

$$(H \otimes H)|\chi^-\rangle = \frac{1}{\sqrt{2}}[|0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle]. \quad (5.41)$$

Tada atlikę įprastą Pauli- Z matavimą matome, kad $|\chi^+\rangle$ ir $|\chi^-\rangle$ būsenose tikriniu verčiu sandauga bus +1 ir -1, atitinkamai. Agnė ir Benas, paaukojė dalį supintujų kubitų, gali dviem matavimo būdais ir tarpusavio ryšiu tiksliai pasakyti, kokią būseną jiems siunčia EPR šaltinis. Neatitikimai tarp Agnės ir Beno matavimų yra indikatorius, kad šaltinis nėra tikslus ar būsenos yra pažeidžiamos duomenims keliaujant kvantiniu kanalu.

Pabandykime suprasti, kodėl LOCC metodu reikia paaukoti du kubitus. Kaip žinome, 1 kubito skaičiuojamieji baziniai vektoriai yra Pauli- Z operatoriaus tikriniai vektoriai, $Z|0\rangle = |0\rangle$, $Z|1\rangle = -|1\rangle$. Todėl Pauli- Z matavimas, atliekamas vienam iš jo bazinių vektorių, nepakeičia šio vektoriaus ir užtikrintai nusako, kokia tai būsena. Tačiau lokalūs Pauli matavimai vienam iš kubitų Belo būsenose pakeičia bendrą 2 kubitų būseną ir sunaikina supynimą, kadangi jos nėra lokaliųjų Pauli operatorių tikriniai vektoriai. Tad reikalingos dvi supintosios būsenos kopijos ir du lokalūs matavimai, nusakantys lyginimo ir fazės bitus.

Belo būsenos yra operatorių $\bar{X} = X \otimes X$, $\bar{Y} = Y \otimes Y$ ir $\bar{Z} = Z \otimes Z$ tikriniai vektoriai:

$$\bar{X}|\chi^\pm\rangle = \pm|\chi^\pm\rangle, \quad \bar{Y}|\chi^\pm\rangle = \mp|\chi^\pm\rangle, \quad \bar{Z}|\chi^\pm\rangle = |\chi^\pm\rangle; \quad (5.42)$$

$$\bar{X}|\eta^\pm\rangle = \pm|\eta^\pm\rangle, \quad \bar{Y}|\eta^\pm\rangle = \pm|\eta^\pm\rangle, \quad \bar{Z}|\eta^\pm\rangle = -|\eta^\pm\rangle. \quad (5.43)$$

Trys operatoriai \bar{X} , \bar{Y} ir \bar{Z} yra tarpusavyje komutatyvūs, todėl atlikus \bar{X} ir \bar{Z} matavimus sekoje Belo būsenai dvi gautos tikrinės vertės ją unikaliai atskirtų nuo kitų nesugriaunant pačios būsenos. Tačiau \bar{X} , \bar{Y} ar \bar{Z} projekciniai matavimai yra nelokalūs – tai yra ne tas pats, kas atlikti du lokalius matavimus ir tikrines vertes sudauginti. Kvantiame kompiuteryje projekciniai Belo būsenų matavimai atliekami kombinacija, susidedančia iš nelokalios unitarinės transformacijos cX , lokalių H bei įprastinio Pauli- Z matavimo abiem kubitams.

5.6 Belo nelygybė

A. Einšteinas nepasidavė idėjai, kad gamtoje fundamentaliai gali egzistuoti nedeterminizmas ir **nuotolinė įtaka** (angl. *spooky action at a distance*), apsireiškiantys supintų kvantinių būsenų matavimuose. Dėl to jis laikėsi pozicijos, kad kvantinė mechanika nors ir teisinga, bet vis dėlto negali būti išsami teorija. Siekdamas patikrinti, ar alternatyvios deterministinės teorijos, įkomponuojančios paslėptus lokalius kintamuosius, sugebėtų atkurti kvantinių koreliacijų efektus, Johnas Bellas 1970-aisiais išvedė vadinamąją **Belo nelygybę** (angl. *Bell inequality*).

Be fundamentalių tyrimų srities, Belo nelygybių klasės testai yra taikomi praktikoje siekiant patikrinti, ar sugeneruotos 2 kubitų būsenos yra supintosios. Eksperimentuose standartiskai naudojama Belo nelygybės versija, vadinama CHSH (angl. *J. Clauser, M. Horne, A. Shimony, R. Holt*), kuri labiau tinkta atlikti testams su šviesos nesėjais fotonais. Toliau pateikiame CHSH nelygybės supaprastintą įrodymą ir jos testo protokolą, naudojantį supintuosius fotoninius kubitus.

Įsivaizduokime scenarijų, kuriame Cita paruošia dvi sistemas ir neatskleisdama paruošimo būdo pateikia vieną Agnei, o kitą Benui. Šioje stadijoje nedarome jokių prielaidų, kokios tai sistemos

ir kokiais gamtos dėsniais jos paremtos. Agnė ir Benas, abu gavę po vieną sistemą, atsitiktiniu būdu renkasi pamatuoti šių sistemų vieną iš savybių. Šitos savybės yra objektyvios ir gali būti atskleistos atlikus matavimus. Klasikinės savybės būtų, pavyzdžiui, geometrinė forma ar svoris. Agnės galimus pasirinkimus vadinsime A_1 ir A_2 , Beno – B_1 ir B_2 . Kiekvienos iš šių keturių savybių galimi matavimų rezultatai yra įvardijami skaitmeniškai, +1 arba -1. Beno sistema bei jos matuojamos savybės B_1 ir B_2 gali skirtis nuo Agnės A_1 ir A_2 , tai nėra svarbu. Svarbu tik tai, kad Cita gali paruošti šias dvi sistemas identiškai n kartų. Kiekvieno paruošimo metu ji savo nuožiūra pasirenka, kokios bus matuojamų sistemų savybės (A_1 ar A_2 , B_1 ar B_2). Agnė ir Benas atlieka atsitiktiniu būdu pasirinktos savybės matavimus tuo pačiu metu ir būdami labai toli vienas nuo kito.

Agnė ir Benas pakartoja šiuos matavimus n kartų ir susitikę apskaičiuoja aritmetinius vidurkius keturių skirtingų narių: A_1B_1 , A_2B_1 , A_1B_2 , A_2B_2 . Pavyzdžiui, jeigu vienam iš bandymų Agnė pasirinko A_1 savybės matavimą (rezultatas +1 arba -1), o Benas B_2 matavimą (rezultatas +1 arba -1), tada šios A_1B_2 matavimų poros rezultatus jie sudaugina ir gauna +1 arba -1. Aritmetinį A_1B_2 matavimo porą vidurkį, vadinsime jį $\langle A_1B_2 \rangle$, jie randa sudėjė visus šios atsirandančios sandaugos rezultatus ir padaliję iš skaičiaus, kuris nusako, kiek kartų A_1B_2 matavimo pora buvo atlikta. Nariai $\langle A_1B_2 \rangle$ nusako koreliacijas tarp šių matuojamų savybių. Jeigu, pavyzdžiui, A_1 ir B_2 matavimais rastos savybės idealiai koreliuoja (antikoreliuoja), tada jų sandauga A_1B_2 ir vidurkis $\langle A_1B_2 \rangle$ visada bus +1 (-1). Tačiau, jeigu A_1 ir B_2 matavimų rezultatai ir todėl jų sandaugų vertės vis atsitiktinai keičiasi tarp +1 ir -1, tada koreliacija tarp šių pamatuotų savybių bus mažesnė arba koreliacijos išvis nebus.

Nežiūrėdami į Agnės ir Beno rezultatus pabandykime įvertinti Citos paruoštos vienos atskiro serijos galimus atsakymus. Tam sudėsime tris pirmus narius bei atimsime paskutinį:

$$C = A_1B_1 + A_2B_1 + A_1B_2 - A_2B_2 = (A_1 + A_2)B_1 + (A_1 - A_2)B_2. \quad (5.44)$$

Matome, kad jeigu A_1 ir A_2 vertės yra skirtingos, gausime $A_1 + A_2 = 0$ ir $A_1 - A_2 = \pm 2$. Tačiau, jeigu jos yra vienodos, $A_1 + A_2 = \pm 2$ ir $A_1 - A_2 = 0$. Todėl, priklausomai nuo rastų A_1 ir A_2 verčių, C gali būti tik +2 arba -2. Akivaizdu, kad atskirų C serijų vidurkio absolūčioji vertė $|\langle C \rangle|$ gali būti ir mažiau nei 2, tad bendrai $|\langle C \rangle| \leq 2$. Panaudodami vidurkių apibrėžimą, $\langle C \rangle = \langle A_1B_1 \rangle + \langle A_2B_1 \rangle + \langle A_1B_2 \rangle - \langle A_2B_2 \rangle$, gauname CHSH nelygybę:

$$|\langle A_1B_1 \rangle + \langle A_2B_1 \rangle + \langle A_1B_2 \rangle - \langle A_2B_2 \rangle| \leq 2. \quad (5.45)$$

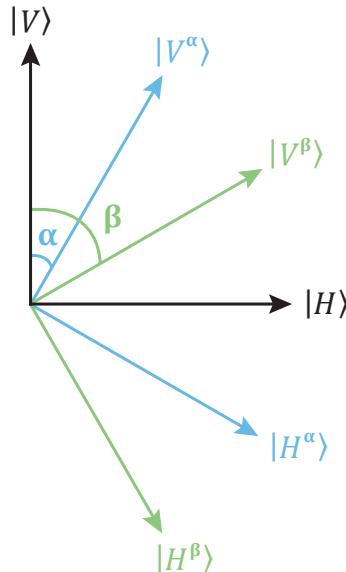
Agnė ir Benas, po daugelio matavimų radę individualius vidurkius $\langle A_iB_j \rangle$, gali įvertinti, ar koreliacijos tarp jų matuojamų sistemų savybių tenkina CHSH nelygybę.

CHSH nelygybė buvo išvesta atsižvelgiant į sistemos savybių paruošimo metodą ir jų matavimo konfigūraciją. Savybės paruošimas veda prie pirmosios prielaidos, vadinamos **realizmu** (angl. *realism*). Realizmas įvardija, kad sistemos savybės yra tiksliai apibrėžtos ir egzistuoja neprilausomai nuo to, ar jos yra stebimos, ar ne. Tai matome iš to, kad visos keturios savybės $\{A_1, A_2, B_1, B_2\}$ jau prieš Agnė ir Benui jas atskleidžiant turi deterministines vertes +1 arba -1, nustatytas iš anksto Citos. Antra, vadinamoji **lokalumo** (angl. *locality*), prielaida įvardija, kad tik artimoji sistemos aplinka gali daryti įtaką jos būsenai. Dėl įvardinto didelio atstumo tarp Agnės ir Beno lokacijų, kuriose jiedu atlieka sistemos savybių matavimus, jų rasti rezultatai neturėtų akimirksniu paveikti vienas kito. Antraip, reliatyvumo teorijos pamatinė aksioma, tvirtinanti, kad niekas negali sklisti greičiau už šviesos greitį, būtų pažeista. Priimant realizmą bei lokalumą, visos gamtoje egzistuojančios sistemos turėtų tenkinti CHSH nelygybę. Klasikinės sistemos tenkina šias, mums visiems intuityvias, prielaidas.

Panagrinėkime, ar šią nelygybę tenkina kvantinės sistemos. Imkime dviejų fotonų Belo būseną $|\chi^+\rangle$, kurioje kubitų būsenos $|0\rangle$ ir $|1\rangle$ nusako fotonų vertikalią ir horizontalią poliarizacijas, atitinkamai. Pervadiname būsenas taip $|0\rangle \rightarrow |V\rangle$, $|1\rangle \rightarrow |H\rangle$:

$$|\chi^+\rangle = \frac{1}{\sqrt{2}}(|V\rangle + |H\rangle). \quad (5.46)$$

Siekdami patikrinti CHSH nelygybę, Agnė ir Benas taiko keturias detektorių poliarizacijos matavimo konfigūracijas, kurios yra A_1, A_2, B_1, B_2 sistemos savybių atitinkmuo. Šiame eksperimente atstumai tarp Agnės ir Beno detektorių vėlgi yra dideli, užtikrinant, kad gauti poliarizacijos rezultatai negali paveikti vienas kito. Kiekvienam bandymui Agnė savo detektoriuje atsitiktiniu būdu pasirenka kampą α_1 arba α_2 , o Benas β_1 arba β_2 . 5.7 pav. iliustruojame dvi tokias detektorių konfigūracijas, pasuktas kampu α ir β :



5.7 pav.: Koordinačių sistemos pasukimas kampais α ir β pradinės koordinačių sistemos atžvilgiu. Pradinė koordinačių sistema apibrėžia horizontaliosios ir vertikaliosios poliarizacijos kryptis

Matome, kad poliarizacijos matavimai Agnės ir Beno detektoriuose yra pasukami pagal laikrodžio rodyklę nurodytais kampais vertikalios ašies lygiagrečios $|V\rangle$ atžvilgiu, išlaikant statujį kampą tarp naujų $|V\rangle$ ir $|H\rangle$ ašių. Naujos vertikali ir horizontali poliarizacijos būsenos, pasuktos kampu α pradiniai būsenai $|V\rangle$ ir $|H\rangle$ atžvilgiu, yra lengvai randamos:

$$|V^\alpha\rangle = \cos(\alpha)|V\rangle + \sin(\alpha)|H\rangle; \quad (5.47)$$

$$|H^\alpha\rangle = -\sin(\alpha)|V\rangle + \cos(\alpha)|H\rangle. \quad (5.48)$$

Tada ermitinis operatorius $P(\alpha)$, nusakantis poliarizacijos dydžius išilgai naujų, pasuktų kampu α ašių, išreikštas spektrinėje dekompozicijoje, yra:

$$P(\alpha) = |V^\alpha\rangle\langle V^\alpha| - |H^\alpha\rangle\langle H^\alpha|. \quad (5.49)$$

Lygtijoje (5.49) $|V^\alpha\rangle\langle V^\alpha|$ ir $|H^\alpha\rangle\langle H^\alpha|$ yra diados, nusakančios projekcijas į vertikaliai ir horizontaliai poliarizuotų būsenų poerdves. Šie operatoriai nusako $|V\rangle$ ir $|H\rangle$ būsenų matavimą

skirtinguose baziniuose vektoriuose – analogišku principu naudojami ir Pauli- Z arba Pauli- X matavimai kriptografijos protokoluose. Kaip ir Pauli- Z operatoriuje, jeigu randama vertikali $|V^\alpha\rangle$ arba horizontali $|H^\alpha\rangle$ poliarizacija, tada fiksuojama atitinkamai +1 arba -1 vertės. Identiškai randami ir kiti trys matavimo operatoriai, nusakantys skirtingus Agnės ir Beno detektoriaus pasukimo kampus α ir β . Siekdami įvertinti $\langle AB\rangle$ koreliacijų koeficientus skirtingose fotonų matavimo konfigūracijose apskaičiuojame:

$$\langle\alpha\beta\rangle = \langle\chi^+|P(\alpha)\otimes P(\beta)|\chi^+\rangle. \quad (5.50)$$

Panaudojė viršuje išreikštus $P(\alpha)$ ir atlikę šiek tiek algebras veiksmų randame, kad dviejų matavimų tikriniai verčių sandaugos vidurkis priklauso tik nuo skirtumo tarp detektorių pasukimo kampų:

$$\langle\alpha\beta\rangle = \cos[2(\alpha - \beta)]. \quad (5.51)$$

Fotoninėse Belo būsenose CHSH nelygybė yra pažeidžiama didžiausia vertė, kai skirtumai tarp kampų keturiose konfigūracijose skiriasi 22.5° laipsnio. Čia renkamės $\alpha_1 = 22.5^\circ$, $\alpha_2 = 45^\circ$, $\beta_1 = 67.5^\circ$, $\beta_2 = 90^\circ$. Tada koreliacijos koeficientai yra $\langle\alpha_1\beta_1\rangle = \langle\alpha_2\beta_1\rangle = \langle\alpha_1\beta_2\rangle = 1/\sqrt{2}$, $\langle\alpha_2\beta_2\rangle = -1/\sqrt{2}$, ir randame $|\langle C\rangle|$:

$$|\langle\alpha_1\beta_1\rangle + \langle\alpha_2\beta_1\rangle + \langle\alpha_1\beta_2\rangle - \langle\alpha_2\beta_2\rangle| = 2\sqrt{2}. \quad (5.52)$$

Akivaizdu, kad CHSH nelygybė, $|\langle C\rangle| \leq 2$, nėra tenkinama supintosiose kvantinėse sistemose. Tai parodo, kad CHSH nelygybės įrodyme daromos realizmo ir lokalumo prielaidos apie sistemą negali būti teisingos. Realizmo prielaida atsiremia į kvantinės mechanikos trečiąjį postulatą, kuris teigia, kad matavimas priverčia sistemą, esančią būseną superpozicijoje, pasirinkti (nedeterministiskai) vieną iš galimų būsenų. Kvantinė mechanika taip pat meta iššūkį mums suprantam lokalumui.

Belo nelygybės eksperimentiniai testai ne kartą parodė, kad kvantinei mechanikai alternatyvios deterministinės lokalios paslėptų kintamųjų teorijos nesugeba atkurti stebimų koreliacijų. Tai palaiko argumentą, kad nėra paslėptų ar kažkaip mums pro pirštus praslydusių veiksnių, kuriuos iutraukus būtų galima visada tiksliai atspėti būseną matavimo rezultatus. Kadangi koreliacijų supintuosiuose kubituose negali imituoti jokia klasikinė sistema ar klasiskai koreliuota kvantinė sistema, CHSH nelygybės testas suteikia būdą įvertinti kvantinio supynimo egzistavimui, reikalingą vykdyti kvantinių ryšių protokolus.

VI skyrius

Skaičiavimai kvantiniu kompiuteriu

Šiame skyriuje supažindiname su funkciniais skaičiavimais kvantiniu kompiuteriu ir parodome keletą žymų kvantinių algoritmų. Šie algoritmai leidžia geriau suprasti kvantinį skaičiavimo modelį ir naudojamus triukus bei ugdyti „kvantinę intuiciją“ sudėtingesnėms užduotims spręsti.

6.1 Bazinių vektorių numeracija

Pirmiausiai pristatysime nomenklatūras, naudojamas numeruoti n kubitų registro baziniams vektoriams, ir jų konvertavimą tarp dvejetainės ir dešimtainės skaičių sistemų. Kvantinio registro, sudaryto iš n kubitų, bendra būsena $|\psi\rangle$ yra 2^n ortogonalųjų bazinių vektorių superpozicija. Praeitų skyrių pavyzdžiuose dažnai naudojome 2 kubitų skaičiuojamąjį rinkinį $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. Esant dideliam kubitų skaičiui taip rašyti vektorius tampa nepraktiška, nes kiekvienas bazinis vektorius $| \dots \rangle$ bus sudarytas iš ilgos n dvejetainių skaičių sekos. Be to, neretai atsiranda poreikis perteikti ar atliliki operacijas su dešimtainiais skaičiais.

Kvantinėje kompiuterijoje dažnai naudojami du skirtini būdai konvertuoti tarp šių skaičių sistemų. Kubitų numeracija juose skiriasi, ir todėl loginių vartų išraiškos skirsis. Fizikoje yra įprasta atliliki n kubitų numeraciją taip: $|k_1 k_2 \dots k_{n-1} k_n\rangle$; čia $k_i \in \{0, 1\}$. Pirmojo kubito, nusakyto pačios viršutinės kvantinės grandinės, būsena k_1 yra rašoma *ket* kairėje ir paeiliui užbaigiamama paskutiniuoju kubitu k_n dešinėje. Ši būdą mes naudojome knygoje iki šiol. Norint perteikti tokią būseną dešimtaine formą $|x\rangle$, $x \in \mathbb{N}$:

$$|x\rangle = |k_1 k_2 \dots k_{n-1} k_n\rangle \quad (6.1)$$

taikome formulę, konvertuojančią dešimtainį skaičių į dvejetainį:

$$x = k_1 2^{n-1} + k_2 2^{n-2} + \dots + k_{n-1} 2^1 + k_n 2^0 = \sum_{i=1}^n k_i 2^{n-i} \quad (6.2)$$

Pavyzdžiui, dvejetainis skaičius 011 paverčiamas į dešimtainį $x = 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 3$. Standartiniai 2 kubitų baziniai vektoriai dešimtainėje sistemoje tampa $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\} \rightarrow \{|0\rangle, |1\rangle, |2\rangle, |3\rangle\}$.

Tačiau kompiuterių moksle ši nomenklatūra skiriasi dvim aspektais. Kubitai yra numeruojami pradedant nuo 0, o ne nuo 1, ir rašomi vektoriuose atvirkštine eiliškumo tvarka, $|k_{n-1} k_{n-2} \dots k_1 k_0\rangle$.

Tai yra, pirmas kubitas žymimas k_0 ir rašomas dešinėje bei užbaigiamas paskutiniuoju k_{n-1} kairėje. Dešimtaine forma išreiškė vektorių $|x\rangle$:

$$|x\rangle = |k_{n-1}k_{n-2}\cdots k_1k_0\rangle \quad (6.3)$$

taikytume šią formulę:

$$x = k_02^0 + k_12^1 + \cdots + k_{n-2}2^{n-2} + k_{n-1}2^{n-1} = \sum_{i=0}^{n-1} k_i 2^i. \quad (6.4)$$

Nepriklausomai nuo to, kuris būdas taikomas dvejetainę formą paverčiant dešimtaine, rezultatas yra vienodas. Pavyzdžiui, skaičius 011 paverčiamas į dešimtainę $x = 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 = 3$. Tačiau, n -kubitų loginių vartų išraiška skiriasi, pavyzdžiui, $CNOT$ su pirmu kubitu, atliekančiu kontrolinio vaidmenį $|k_1k_2\cdots k_{n-1}k_n\rangle$ numeracijoje yra $cX = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$. Antruoju būdu, $|k_{n-1}k_{n-2}\cdots k_1k_0\rangle$, jie yra $cX = I \otimes |0\rangle\langle 0| + X \otimes |1\rangle\langle 1|$. Tai nusako skirtinges matricas. Šioje knygoje vartojame fizikoje įprastą numeraciją $|k_1k_2\cdots k_{n-1}k_n\rangle$, nebent iš anksto įspėjama, kad naudojamas kompiuterių moksle įprastas būdas.

Iš n kubitų sudarytas 2^n skaičiuojamujų bazinių vektorių rinkinys dešimtaine forma yra $\{|0\rangle, |1\rangle, |2\rangle, \dots, |2^n - 1\rangle\}$. Bendra būsena $|\psi\rangle$, normavimo sąlyga ir ortogonalumas yra išreiškiami:

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle + c_2|2\rangle + \cdots + c_{2^n-1}|2^n - 1\rangle = \sum_{x=0}^{2^n-1} c_x|x\rangle; \quad (6.5)$$

$$\langle x'|x\rangle = \delta_{x'x}; \quad (6.6)$$

$$|c_0|^2 + |c_1|^2 + |c_2|^2 + \cdots + |c_{2^n-1}|^2 = \sum_{x=0}^{2^n-1} |c_x|^2 = 1. \quad (6.7)$$

6.2 Funkcinis skaičiavimas

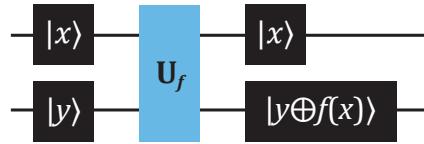
Skaitmeniniuose skaičiavimuose kintamieji turi dvi reikšmes $\{0, 1\}$. Tad galima traktuoti, kad visuose tokiuose skaičiavimuose yra įvertinamos Būlio funkcijos (angl. *Boolean function*) $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$. Ši išraiška nusako, kad Būlo funkcija f transformuoja n bitų ilgio seką į kitą m bitų ilgio seką, kurioje m ir n gali būti vienodi arba skirtis. Kvartinėje kompiuterijoje normima funkcija f turi būti perteikiama unitariuoju operatoriumi U_f . Veikiantis n kubitų registro būseną U_f transformuoja ją į kitą n kubitų būseną. Tai yra fundamentaliai invertuojamas procesas, atlikus šiam registrui atvirkštinį operatorių U_f^\dagger bus grąžinta pradinė registro būsena. Visgi didelė dalis skaičiavimuose mus dominančių funkcijų néra invertuojamos. Funkcija $f(x)$ yra invertuojama, jeigu su kiekvienu jos argumentu x galima unikalai susieti vieną reikšmę, tai yra 1:1 funkcijos. Pavyzdžiui, funkcija $f : \{0, 1\}^n \rightarrow \{0, 1\}$ néra invertuojama, nes su n skirtingu argumentu yra asocijuojamos tik dvi skirtinges reikšmės. Taip pat egzistuoja ir $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ neinvertuojamų funkcijų. Pavyzdžiui, parabolė $f(x) = x^2$, apibrėžta visiems argumentams x , néra invertuojama, nes $(+x)^2 = (-x)^2$ ir todėl kiekvienai f reikšmei yra du skirtini argumentai ($+x$ ir $-x$). Dėl šių priežasčių kvartinėje kompiuterijoje funkciniams skaičiavimams atlikti yra dažnai pasitelkiami du kubitų registrų, vadinami **įvesties** (angl. *input register*) ir **išvesties** (angl. *output register*) registras. Kvartinis skaičiavimas, naudojant du registrus, turi tokią formą:

$$U_f|x\rangle \otimes |0\rangle = |x\rangle \otimes |f(x)\rangle. \quad (6.8)$$

Unitarinė transformacija U_f čia veikia abu registrus. Kiekvienas bazinis vektorius $|x\rangle$ įvesties registre atlieka funkcijos argumento rolę x , o funkcijos reikšmė $f(x)$ yra užrašoma išvesties registro būsenoje $|f(x)\rangle$. Jeigu išvesties registras yra ne $|0\rangle$, o kitoje pradinėje būsenoje $|y\rangle$, tada:

$$U_f|x\rangle \otimes |y\rangle = |x\rangle \otimes |y \oplus f(x)\rangle. \quad (6.9)$$

Šiuo atveju naudojame dvejetainę formą, tad išvesties registre yra atliekama y ir $f(x) \bmod(2)$ bitų sudėtis, žymima ženklu \oplus . Paprasčiausią funkcijos skaičiavimo pavyzdį matėme 4 skyriuje naudojant $U_f = cX$, kuris atlieka $\bmod(2)$ bitų sudėtį. 6.1 pav. iliustruojame ši bendrą skaičiavimo principą kvantine grandine.



6.1 pav.: Kvantine grandinė, realizuojanti funkcijų skaičiavimą $U_f|x\rangle \otimes |y\rangle = |x\rangle \otimes |y \oplus f(x)\rangle$

6.3 Kvantinis paralelizmas

Norint klasikiniu kompiuteriu apskaičiuoti n skirtingų argumentų x_1, x_2, \dots, x_n funkcijos $f(x)$ reikšmes, išprastgai reikia šią funkciją pateikti ir ivertinti n kartų. Kvantine kompiuteris leidžia $f(x)$ ivertinti lygiagrečiai visuose n -argumentuose vienu funkcijos iškvietimu. Norėdami tai pamatyti, pradékime vėl nuo 2 kubitų sistemos, turinčios po vieną kubitą įvesties ir išvesties registre siekiant apskaičiuoti funkciją argumentuose $f(0)$ ir $f(1)$. Nelokalus unitarusis operatorius U_f atlieka funkcijos f ivertinimą, randame:

$$U_f \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |f(0)\rangle + |1\rangle \otimes |f(1)\rangle) \quad (6.10)$$

Matome, kad jeigu įvesties registras yra paruoštas į $|0\rangle$ ir $|1\rangle$ superpoziciją, tada dėl operatorių tiesiškumo yra lygiagrečiai ivertinamos $f(0)$ ir $f(1)$ vertės vienu funkcijos f pritaikymu.

Ši principą galima praplėsti ir atlikti funkcijos ivertinimą 2^n argumentuose paruošiant pradinę išvesties registro būseną į lygią visų n -kubitų 2^n skaičiuojamujų bazinių vektorių superpoziciją. Superpozicija paruošiama atliekant kiekvienam kubitiui Hadamardo loginius vartus. Pavyzdžiu, registre sudarytame iš 2-kubitų $H \otimes H|00\rangle$:

$$\begin{aligned} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) &= \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2} \\ &= \frac{|0\rangle + |1\rangle + |2\rangle + |3\rangle}{2}. \end{aligned} \quad (6.11)$$

Jeigu rašysime $H^{\otimes n} = H \otimes H \otimes \dots \otimes H$ nusakyti Hadamardo transformaciją kiekvienam iš n -kubitų, kurių bendra pradinė būsena dešimtainėje sistemoje yra $|0\rangle$, tada gausime lygią visų 2^n būsenų superpoziciją:

$$H^{\otimes n}|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle. \quad (6.12)$$

Atkreipiame dėmesį, kad eksponentiškai didelės 2^n bazinių vektorių superpozicijos sukūrimas reikalauja tik tiesinio n skaičiaus Hadamardo transformacijų. Skaičiavimo ištaklių atžvilgiu tai yra itin efektyvus metodas.

Norint įvertinti $f(x)$ funkciją jos 2^n skirtinį argumentų x , superpozicijai pritaikome U_f :

$$U_f \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |f(x)\rangle. \quad (6.13)$$

Šis įspūdingas rezultatas yra vadinamas kvantinis paralelizmas. Pavyzdžiui, jeigu turime 100 kubitų kiekviename registre, sukūrus jų lygią superpoziciją ir atlikus U_f transformaciją paraleliai yra įvertinamas astronominis skaičius $\sim 10^{30}$ funkcijos f verčių.

Tačiau šioje stadijoje dar nėra užtikrinta, kad kvantiniu kompiuteriu bus paspartintas skaičiavimas. Norėdami sužinoti skaičiavimo rezultatą bendrai turime išmatuoti visus kubitus įvesties ir išvesties registruose. Sakykime, kad $f(x)$ yra invertuojama funkcija. Kadangi $|x\rangle$ būsena yra supinta su $|f(x)\rangle$, nesvarbu, kurį registrą pasirinksime matuoti pirmą, tad pradėkime nuo įvesties. Įvesties registras yra lygioje visų 2^n skirtinį $|x\rangle$ būsenų superpozicijoje, todėl yra lygi $\frac{1}{2^n}$ tikimybė rasti bet kurią vieną iš šių būsenų. Atlikus įvesties registro kubitų matavimą ir radus $|x_0\rangle$, bendra įvesties ir išvesties registrų būsena tampa $|x_0\rangle \otimes |f(x_0)\rangle$. Kitame žingsnyje išmatuojame išvesties registrą, taip sužinodami $f(x_0)$ reikšmę. Jeigu $f(x)$ yra neinvertuojamoji funkcija, tada keletas skirtinį $|x\rangle$ būsenų gali būti supintos su ta pačia $|f(x_0)\rangle$. Šiuo atveju taip pat nėra svarbu, kurį registrą pasirinksime matuoti pirmą. Pavyzdžiui, pirmiausia išmatavus išvesties registro kubitus ir radus $|f(x_0)\rangle$, bendra būsena tampa $(|x_m\rangle + \dots + |x_1\rangle) \otimes |f(x_0)\rangle$. Matome įvesties registrą esantį būsenų superpozicijoje, kurios nusako argumentus su vienodomis funkcijos reikšmėmis $f(x_0)$. Tikimybė rasti bet kurią vieną būseną yra lygi kitoms. Šiame procese atsitiktiniu būdu randame vieną iš $f(x)$ reikšmių, negalėdami pasirinkti, kuriame argumente x norime įvertinti f .

Kvantiniai algoritmai pasitelkia papildomus triukus ir turi būti formuluojami kitaip ne klasikiniai. Viena tokia algoritminė strategija, vadinama **amplitudės amplifikacija** (angl. *amplitude amplification*), leidžia padidinti registro būsenų, koduojančių ieškomus atsakymus, amplitudes. To pavyzdži matysime šiame skyriuje Groverio paieškos algoritme. Kitas būdas yra tiesiogiai nustatyti tam tikras funkcijos savybes – jos globalius parametrus, minimalias ar maksimalias vertes, periodiškumą ir panašiai. Ši strategija išnaudoja interferenciją ar koreliacijas tarp kvantinių būsenų. Ją taiko nemaža dalis algoritmų, išskaitant Doičo bei Šoro algoritmus, Hadamardo ir SWAP testai, fazės nustatymo algoritmas, kai kurie mašininio mokymosi metodai ir kvantinių klaidų taisymo kodai.

6.4 Duomenų kodavimo būdai

Skaičiavimo procesą kvantiniame kompiuteryje galima apibūdinti trimis žingsniais:

Duomenų kodavimas → *Duomenų apdorojimas* → *Būsenų matavimas*

Norint pateikti duomenis pirmiausia reikia juos koduoti kubitais. Tai nusako specifinį kubitų registro **būsenos paruošimo procesą** (angl. *state preparation*). Analizuojant kvantinių algoritmų sudėtingumą tenka atsižvelgti į tai, kad būsenos paruošimo žingsniai gali savaime pareikalauti daug resursų. Blogiausiu atveju, perteikti bendrai n kubitų būsenai gali prireikti eksponentiškai didelio $O(2^n)$ loginių operacijų skaičiaus nustatant visiems 2^n skaičiuojamiesiems baziniams vektoriams skirtinas amplitudes.

Duomenų tipas bei kodavimo metodas nulemia kodavimo žingsnio sudėtingumą. Egzistuoja ne vienas kodavimo metodas – plačiausiai taikomi yra **bazinių vektorių kodavimas** (angl. *basis encoding*) ir **amplitudžių kodavimas** (angl. *amplitude encoding*). Bazinių vektorių metodas tinka koduoti informaciją dvejetainėje formoje. Visa klasikinė duomenų bazė D , kurioje kiekvienas atskiras įrašas l yra N -bitų seka $b^{(l)} = \{b_1, b_2, \dots, b_N\}$, $b_i \in \{0, 1\}$, tiesiogiai perteikiama normuota skaičiuojamųjų bazinių vektorių $b^{(l)} \rightarrow |b^{(l)}\rangle$ lygia superpozicija:

$$|D\rangle = \frac{1}{\sqrt{L}} \sum_{l=1}^L |b^{(l)}\rangle. \quad (6.14)$$

Čia L nusako įrašų skaičių ir todėl naudojamų bazinių vektorių skaičių duomenų bazėje $|D\rangle$. Kubitų skaičius n turi būti ne mažesnis nei ilgiausios duomenų bazės įrašo $b^{(l)}$ bitų skaičius N , tad $L \leq 2^N$. Visų kitų galimai nekoduojančių $2^N - L$ bazinių vektorių amplitudės lygios nuliui. Pavyzdžiui, duomenų bazę, turinčią du įrašus $\{00110, 10100\}$, perteiktume 5 kubitų būseną $|D\rangle$:

$$|D\rangle = \frac{|00110\rangle + |10100\rangle}{\sqrt{2}}. \quad (6.15)$$

Čia vėl naudojame išprastą fizikoje kubitų numeraciją perteikti dvejetainiams skaičiams. Šis kodavimo būdas nėra itin efektyvus kubitų skaičiaus atžvilgiu, tačiau natūraliai tinka atliki skaitmenines aritmetines operacijas ir funkcinius skaičiavimus. Nemažai algoritmų naudoja bazinių vektorių kodavimo būdą: Groverio, kvantinė Furjė transformacija, Šoro pirminių skaičių faktorizavimas.

Kaip pavadinimas indikuoja, amplitudžių kodavimo metode duomenys yra koduojami bazinių vektorių amplitudėse. Imkime vieną klasikinį duomenų bazės įrašą $x^{(l)} = \{x_1, x_2, \dots, x_N\}$ turintį N elementų. Bendrai toks įrašas nusako N dimensijų vektorių ar duomenis su N skaičiumi savybių. Kiekvienas elementas x_i gali būti bet kokios formos skaičiai – dešimtainėje formoje realieji ar kompleksiniai. Reikalaujama, kad visi elementai x_i būtų normalizuojami: $x_i \rightarrow x_i/\sqrt{N}$. Kiekvienas įrašas $x^{(l)}$ yra perteikiamas skaičiuojamųjų bazinių vektorių $|i\rangle$ superpozicijos amplitudėse x_i :

$$|x^{(l)}\rangle = \sum_{i=1}^N x_i |i\rangle. \quad (6.16)$$

Šis kodavimo būdas yra kubitų skaičiaus atžvilgiu efektyvus, nes N elementų įrašas reikalauja tik $\log_2(N)$ kubitų. Duomenų bazę D , turinčią M skaičių su N elementų ilgio įrašais $x^{(l)}$, $D = \{x^{(1)}, x^{(2)}, \dots, x^{(M)}\}$, galima perteikti $(M \times N)$ dimensijų vektoriumi kubitų registre:

$$|D\rangle = \sum_{i=1}^{MN} x_i |i\rangle. \quad (6.17)$$

Tai reikalauja $n \geq \log_2(MN)$ kubitų skaičiaus. Kitaip tariant, norint koduoti visus MN įrašus bazinių vektorių skaičius 2^n turi būti $2^n \geq MN$. Potencialų nekoduojančių elementų perteklių būsenose irgi galima užpildyti nuliais. Amplitudžių kodavimas taikomas daugelyje kvantinio mašininio mokymosi algoritmų ir kvantinių sistemų modeliavime. Šio metodo trūkumas tas, kad skaičiavimo pabaigoje amplitudžių x_i tiesiogiai negalima nuskaityti. Reikalingas kitas būdas naudoti juose koduojamą informaciją, pavyzdžiui, apskaičiuojant tam tikrą amplitudžių funkciją $f(x_i)$, išreikštą kvantiniu operatoriumi U taip realizuojant $f(x_i) = \langle x_i | U | x_i \rangle$. Hadamardo testas (žr. 6.7 poskyri) leidžia efektyviai apskaičiuoti šiuos narius.

6.5 Doičo algoritmas

Doičo algoritmas (angl. *Deutsch algorithm*) yra vienas iš pirmųjų ir paprasčiausių pavyzdžių, iliustruojantis kvantinio algoritmo pranašumą prieš klasikinį. Doičo algoritmas nėra savaime ypač naudingas, tačiau parodo esminį būsenų superpozicijos ir interferencijos panaudojimą kvantiniuose skaičiavimuose.

Įsivaizduokime scenarijų, kuriame Agnė turi juodąją dėžę (angl. *black box*), atliekančią vieno bito manipuliacijas. Ši dėžė apskaičiuoja funkciją f , kuriai pateikus bitą su verte 0 arba 1, ji išveda kitą bitą, taip pat 0 arba 1. Egzistuoja iš viso keturios skirtinges 1 bito funkcijos $f : \{0, 1\} \rightarrow \{0, 1\}$:

$$f_1(0) = f_1(1) = 0, \quad f_2(0) = f_2(1) = 1; \quad (6.18)$$

$$f_3(0) = 0, \quad f_3(1) = 1, \quad f_4(0) = 1, \quad f_4(1) = 0. \quad (6.19)$$

Matome, kad pirmose dviejose funkcijose f_1 ir f_2 reikšmės nepriklauso nuo pateiktų argumentų, $f(0) = f(1)$. Šias dvi funkcijos vadiname pastoviosiomis. Funkcijose f_3 ir f_4 reikšmės priklauso nuo argumentų, $f(0) \neq f(1)$. Jas vadiname subalansuotosiomis.

Agnė turi tik vieną bandymą, skirtą sužinoti, ar juodojoje dėžėje slepiasi pastovioji, ar subalansuotoji funkcija. Akivaizdu, kad dviem bandymais ji galėtų tai lengvai padaryti. Turint tik klasikinius išteklius neįmanoma vienu juodosios dėžės panaudojimu atlikti norimą funkcijos klasifikaciją, tačiau kvantiniu kompiuteriu pakanka vieno.

Kvantiniame kompiuteryje juodosios dėžės funkciją $f : \{0, 1\} \rightarrow \{0, 1\}$ atlieka unitarioji transformacija minėtu principu: $U_f|x\rangle \otimes |y\rangle = |x\rangle \otimes |y \oplus f(x)\rangle$. Naudojama po 1-kubitų įvesties (k_1) ir išvesties (k_2) registruose. Doičo algoritmas pasitelkia vadinamąjį **fazės atatrankos** (angl. *phase kickback*) metodą. Prieš panaudojant juodosios dėžės funkciją, šiame metode išvesties registro būsena yra paruošiama į superpoziciją $H|1\rangle$. Tada abiems registrams pritaikius U_f gaunama:

$$U_f|x\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) = (-1)^{f(x)}|x\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right). \quad (6.20)$$

Matome, kad fazės atatrankoje išvesties registro būsena nepakinta, tačiau bendrai būsenai $|x\rangle \otimes H|1\rangle$ yra perteikiama santykinė fazė $(-1)^{f(x)}$ priklausomai nuo įvesties būsenos $|x\rangle$, $x \in \{0, 1\}$. Tad $(-1)^{f(x)=0} = 1$ ir $(-1)^{f(x)=1} = -1$. Kitaip nei įprastiniame funkciniame skaičiavime, galime traktuoti, kad U transformacija efektyviai perkelia būsenos pokytį, šiuo atveju fazę, į išvesties registrą. Tai yra vadinama **fazės atatranka** (angl. *phase kickback*), ji aptinkama ne viename algoritme.

Pradedant Doičo algoritmą, įvesties ir išvesties registro kubitams, esantiems atitinkamai $|0\rangle$ ir $|1\rangle$ būsenose, pirmiausiai pritaikome Hadamardo transformacijas:

$$|\psi\rangle = (H \otimes H)|0\rangle \otimes |1\rangle = \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle). \quad (6.21)$$

Toliau panaudojame juodosios dėžės funkciją f , nusakančią unitariajų transformaciją U_f :

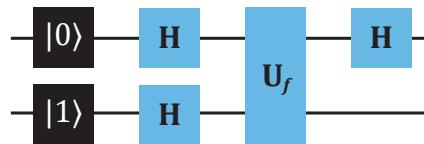
$$U_f|\psi\rangle = \frac{1}{2}[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle] \otimes (|0\rangle - |1\rangle). \quad (6.22)$$

Skliausteliuose matome fazės atatrankos efekta įvesties registro kubitams, esantiems superpozicijoje. Kadangi išvesties registro būsena nebeturi įtakos likusio algoritmo žingsniams, ją pašaliname

iš tolimesnės analizės. Įvesties registro kubitui pritaikome dar vieną Hadamardo transformaciją $H \otimes I$:

$$(H \otimes I)U_f|\psi\rangle = \frac{1}{2}\{[(-1)^{f(0)} + (-1)^{f(1)}]|0\rangle + [(-1)^{f(0)} - (-1)^{f(1)}]|1\rangle\}. \quad (6.23)$$

Jeigu juodosios dėžės funkcija yra pastovioji $f(0) = f(1)$, tada $(-1)^{f(0)} - (-1)^{f(1)} = 0$, ir viršuje išlieka tik $|0\rangle$ būsena. Jeigu funkcija subalansuota, $f(0) \neq f(1)$, tada išlieka tik $|1\rangle$ būsena (iki nesvarbios globalios fazės). Įvesties registro būseną, prieš atliekant matavimą, galima glaustai užrašyti $|f(0) \oplus f(1)\rangle$. Matavimo rezultatas užtikrintai klasifikuja šią funkciją, o procese juodoji dėžė panaudojama tik vieną kartą. Doičio algoritmą nusakanti kvantinė grandinė yra pateikta 6.2 pav.



6.2 pav.: Doičio algoritmo loginė grandinė

Doičio algoritmas leidžia palyginti, ar funkcijos vertė skirtinguose argumentuose yra vienoda, ar skirtinė, neatskleidžiant, kokios yra tos vertės. Tai skaičiavimo pavyzdys, panaudojantis paralelumą bei būsenų interferenciją, siekiant nusakyti globalią funkcijos savybę. Paralelumas matomas $f(0)$ ir $f(1)$ funkcijos įvertinime vienu metu. O štai konstruktyvios ir destruktyvios būsenų interferencijos efektai matomi, kai pasitelkus Hadamardo transformaciją atliekama narių sudėtis su $(-1)^{f(x)}$ amplitudėse, dėl kurios panaikinama $|0\rangle$ arba $|1\rangle$ būsena.

Doičo-Jodžos algoritmas (angl. *Deutsch-Josza algorithm*), kurio čia plačiau neanalizuojame, paprastai praplečia Doičio algoritme naudojamos funkcijos f dydį nuo 2 iki 2^n argumentų ir leidžia vienu juodosios dėžės panaudojimu klasifikuoti ją kaip esančią pastoviąją arba subalansuotąją. Pastovioji f funkcija 2^n argumentų yra tokia, kurios vertės skirtinguose argumentuose yra vienodos. Subalansuotojoje funkcijoje pusė jos visų verčią ($2^n/2$) yra 0 ir pusė 1. Tad klasikiniame kompiuteryje gali reikėti iki $2^{n-1} + 1$ funkcijos f įvertinimų norint atlikti šią klasifikaciją.

6.6 Kvantinė paieška ir Groverio algoritmas

Kvantinis kompiuteris gali reikšmingai paspartinti įrašų paieškos procesą nestruktūrizuotose duomenų bazėse. Vienas tokios paieškos pavyzdys būtų „vardenio-pavardenio“ paieška telefonų knygoje žinant telefono numerį, nes telefonų knyga yra struktūrizuota pagal vardus. Jeigu imsimė, kad duomenų bazė turi N įrašų, klasikiniu algoritmu nėra kito būdo, kaip tik tikrinti visus įrašus, tad blogiausiu atveju gali prireikti $N - 1$ patikrų, o tikimybė rasti įrašą po k -skaičiaus bandymų yra k/N . Bet štai kvantiniu kompiuteriu pakanka \sqrt{N} įrašų patikrinimų norint rasti norimą su praktiškai 100% tikimybe. Šis kvadratinis paspartinimas gali suteikti esminį pranašumą didėjant įrašų skaičiui duomenų bazėje. Irodyta, kad Groverio algoritmas yra optimaliausias kvantinės paieškos nestruktūrizuotoje duomenų bazėje algoritmas. Bet koks kitas gali nebent pakeisti $O(\sqrt{N})$ algoritmo laiko sudėtingumą bendraja konstanta.

Paieškos problemą galima formaliai apibūdinti Būlio funkcija $f : \{0, 1\}^n \rightarrow \{0, 1\}$, kuriai pateikus n -bitų seką (funkcijos argumentą) funkcijos reikšmė yra 1, jeigu x argumentas atitinka paieškomą,

vadinsime jį d . Visais kitais atvejais, kai $x \neq d$, funkcijos reikšmė yra 0:

$$f(x) = \begin{cases} 1, & x = d, \\ 0, & x \neq d. \end{cases} \quad (6.24)$$

Tokio tipo funkcijos yra vadinamos **orakulu** (angl. *oracle*). Orakulas savaime negali pasakyti ieškomo įrašo, tačiau atpažista, kai šis yra jam pateiktas. Paieškos algoritmo užduotis – rasti norimą įrašą d su kuo mažiau kreipimusi į orakulą (funkcijos f panaudojimų).

Groverio algoritme orakulas yra visa duomenų bazė su $N = 2^n$ įrašų. Tarp jų yra vienas elementas d , kurį norima rasti. Ieškomų įrašų Groverio algoritmo taikymuose gali būti ir daugiau negu vienas, tačiau čia iliustruojame pavyzdį, kai yra tik vienas. Pirmame šio algoritmo žingsnyje visi 2^n įrašai yra perteikiami įvesties registro būsenomis $|x\rangle$ dešimtainėje sistemoje sukuriant lygią superpoziciją:

$$|\psi\rangle = H^{\otimes n}|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle. \quad (6.25)$$

Įvesties registras yra sudarytas iš vieno kubito ir inicijuojamas į $H|1\rangle$ būseną siekiant panaudoti fazės atatrankos metodą. Orakulo funkcijos, vadinsime ją U_f , efektas vienai tokiai būsenai yra:

$$U_f|x\rangle \otimes H|1\rangle = (-1)^{f(x)}|x\rangle \otimes H|1\rangle. \quad (6.26)$$

Tad jeigu orakului pateikiama neteisinga būsena $|x\rangle \neq |d\rangle$, tada $f(x) = 0$ ir abiejų registrų būsena nepakinta. Pateikus ieškomą būseną $|x\rangle = |d\rangle$, bendra būsena tampa $-|d\rangle \otimes H|1\rangle$. Kitaip tariant, orakulas paženklina teisingą būseną įvesdamas jai fazę $|d\rangle \rightarrow -|d\rangle$.

Matome, kad išvesties ir įvesties registrai néra supinti nei prieš, nei po U_f panaudojimo. Kadangi Groverio paieškos algoritme U_f yra vienintelė transformacija, veikianti abu registrus, tačiau ji nekeičia išvesties registro būsenos, toliau analizuojant šį algoritmą galima koncentruotis vien į įvesties registrą. Orakulo transformaciją U_f pakeisime kita efektyvia transformacija, vadinkime ją V , kuri veikia vien tik įvesties registrą (vienetinis operatorius $\otimes I$ išvesties registrui), ir atlieka jam tokią pačią funkciją kaip ir U_f . Tai užrašome:

$$V|x\rangle = (-1)^{f(x)}|x\rangle = \begin{cases} -|x\rangle, & x = d, \\ |x\rangle, & x \neq d. \end{cases} \quad (6.27)$$

Nors matricos forma V operatoriaus čia tiesiogiai nenaudosime, tačiau verta atkreipti dėmesį, kad tai yra diagonalioji transformacija:

$$V = \begin{bmatrix} (-1)^{f(0)} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & (-1)^{f(2^n-1)} \end{bmatrix}. \quad (6.28)$$

Nenuliniai skaičiai yra tik išilgai pagrindinės įstrižainės. Jeigu ieškoma tik viena būsena, tada vienas atitinkamas skaičius įstrižainėje yra -1, o visi kiti 1. Pritaikę operatorių V visam įvesties registrui pradinėje būsenoje randame:

$$V|\psi\rangle = -\frac{1}{\sqrt{2^n}}|d\rangle + \frac{1}{\sqrt{2^n}} \sum_{x \neq d}^{2^n-1} |x\rangle. \quad (6.29)$$

Antrame naryje matome visų būsenų išskyrus $|d\rangle$ sumą, tad sumuojama $2^{n-1} - 1$ narių. Šiek tiek ilgėliau pažiūrėjus į išraišką (6.29) tampa aišku, kad V transformacija turi šią formą:

$$V = -2|d\rangle\langle d| + I. \quad (6.30)$$

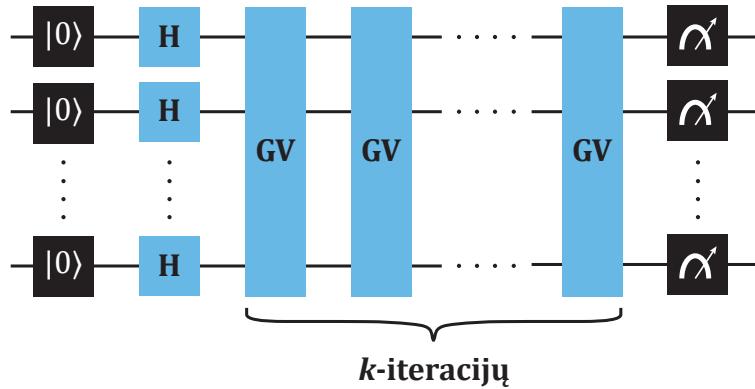
Čia I – vieneticis operatorius, o $|d\rangle\langle d|$ yra projekcinis operatorius į $|d\rangle$ vektoriaus poerdvj. Skaičius -2 atspindij tai, kad viršuje vektorius $|d\rangle$ yra atimamas du kartus – iš būsenų sumos nario, o pavienis $|d\rangle$ narys yra su minuso ženklu.

Siekiant atlikti paiešką, Groverio algoritmas naudoja unitariajā transformaciją, vadinkime ją G , kuri nereikalauja orakulo panaudojimo. Jos išraiška nepriklauso ir nuo paieškomo $|d\rangle$ vektoriaus bei turi šią formą:

$$G = 2|\psi\rangle\langle\psi| - I. \quad (6.31)$$

Čia $|\psi\rangle\langle\psi|$ yra projekcinis operatorius į pradinės būsenos $|\psi\rangle$ poerdvj. Groverio algoritmas (žr. 6.3 pav.) pakartotinai pritaiko, arba iteruoja, GV operatorius, o tai lemia $|d\rangle$ būsenos amplitudės amplifikaciją likusių būsenų sąskaita:

$$|d\rangle \approx GVGV \cdots GV|\psi\rangle. \quad (6.32)$$



6.3 pav.: Groverio algoritmo bendrą principą nusakanti loginė grandinė

Toliau parodysime, kaip ši iteracija atlieka paieškos užduotj amplifikujant ieškomosios būsenos amplitudę ir kiek kartų reikia iteruoti norint užtikrinti, kad $|d\rangle$ bus rasta su didele tikimybe. Galima pažvelgti į Groverio algoritmą dviem būdais: algebriskai analizuojant amplitudžių pokyčius kiekviename iteracijos žingsnyje, arba geometriškai atsižvelgiant į būsenos vektoriaus posūki 2 dimensijų erdvėje. Šie būdai savaip apšviečia algoritmo esmę, tad panagrinėkime juos abu.

6.6.1 Algebrinė interpretacija

Viršuje pateikėme transformacijos V efektą įvertindami $V|\psi\rangle$. Norėdami pamatyti vienos GV iteracijos efektą bendrai būsenai $|\phi\rangle$, toliau įvertinsime $G|\phi\rangle$. Būseną $|\phi\rangle$ galima traktuoti esant tarpinę tarp pradinės $|\psi\rangle$ ir visų kitų galimų registro būsenų algoritmo metu. Ji išreiškiama tais pačiais 2^n skaičiuojamaisias baziniais vektoriais, bet kitomis amplitudėmis c_x :

$$|\phi\rangle = \sum_{x=0}^{2^{n-1}} c_x |x\rangle. \quad (6.33)$$

Kadangi pradinėje $|\psi\rangle$ būsenoje visos amplitudės yra realieji skaičiai, o V ir G operacijos jas tokias išlaiko, viso algoritmo metu amplitudės c_x išlieka realiaisiais skaičiais. Apskaičiuodami $G|\phi\rangle = 2|\psi\rangle\langle\psi|\phi\rangle - |\phi\rangle$, pirmiausiai įvertinsime vidinę sandaugą $\langle\psi|\phi\rangle$:

$$\langle\psi|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x',x=0}^{2^n-1} c_x \langle x'|x\rangle = \frac{\sqrt{2^n}}{2^n} \sum_{x=0}^{2^n-1} c_x = \sqrt{2^n} \bar{c}_x. \quad (6.34)$$

Pirmaje eilutėje vidinių sandaugų suma yra atliekama su $|\psi\rangle$ ir $|\phi\rangle$ baziniais vektoriais $\{|x\rangle\}$, indeksuotais x' ir x , atitinkamai. Kadangi $\{|x\rangle\}$ yra ortogonalų normuotųjų vektorių rinkinys, $\langle x'|x\rangle = \delta_{xx'}$, panaikinamas vienas suminis indeksas. Antroje dalyje identifikavome \bar{c}_x , nusakantį visų 2^n amplitudžių c_x vidurkį:

$$\bar{c}_x = \frac{1}{2^n} \sum_{x=0}^{2^n-1} c_x. \quad (6.35)$$

Dabar galima rasti $G|\phi\rangle$:

$$G|\phi\rangle = 2 \left(\sum_{x=0}^{2^n-1} |x\rangle \right) \bar{c}_x - \sum_{x=0}^{2^n-1} c_x |x\rangle = \sum_{x=0}^{2^n-1} (2\bar{c}_x - c_x) |x\rangle. \quad (6.36)$$

Palyginę su $|\phi\rangle$ būseną matome, kad jai pritaikius G operatorių visos amplitudės pakeičiamos $c_x \rightarrow 2\bar{c}_x - c_x$.

Pradinėje $|\psi\rangle$ būsenoje visos $c_x = 1/\sqrt{2^n}$, tad ir jų vidurkis $\bar{c}_x = 1/\sqrt{2^n}$. Atlikus $V|\psi\rangle$, ieškomos būsenos $|d\rangle$ amplitudė tampa $c_{x=d} = -1/\sqrt{2^n}$, todėl vidurkis \bar{c}_x sumažėja. Tolesniame žingsnyje $GV|\psi\rangle$ ieškomos būsenos amplitudė tampa vėl teigiamą (dėl atimties ženkle $2\bar{c}_x - c_x$ išraiskoje) ir didesnė nei prieš tai, nes pridedamas vidurkis, padaugintas iš dviejų, $2\bar{c}_x + 1/\sqrt{2^n}$. Tačiau visų likusių būsenų amplitudės sumažėja, nes jų amplitudės yra atimamos iš sumažėjusio vidurkio (padauginto iš dviejų). Pirmos iteracijos žingsniai iliustruoti 6.4 pav.

Antrosios iteracijos pirmame žingsnyje $GVG|\psi\rangle$, būsenos $|d\rangle$ amplitudė vėl padaroma neigiamąja, taip sumažinant vidurkį, o $GVGV|\psi\rangle$ vėl amplifikuoją $|d\rangle$ padarant ją teigiamąja ir pridedant vidurkį. O štai visų kitų būsenų amplitudės vėl sumažinamos. Pakartojus GV iteraciją apytiksliai $(\pi/4)\sqrt{N}$ kartų, tikimybė pamatavus registro būseną rasti $|d\rangle$, kai N yra didelis skaičius, gali būti pageidaujamai artima 100 %. Geometrinė Groverio algoritmo interpretacija leidžia intuityviau pademonstruoti, kodėl yra reikalingas būtent toks skaičius iteracijų.

6.6.2 Geometrinė interpretacija

Nors bendra registro būseną $|\phi\rangle$ yra apibūdinama 2^n dimensijų erdvėje, viso Groverio algoritmo metu $|\phi\rangle$ pokyčiai vyksta tik 2 dimensijų vektorių poerdvyje, ir tai leidžia atlikti paprastą geometrinę analizę. Norint tai pamatyti, transformaciją V pradinei būsenai $|\psi\rangle$ bei ieškomajai $|d\rangle$ perteikiamame taip:

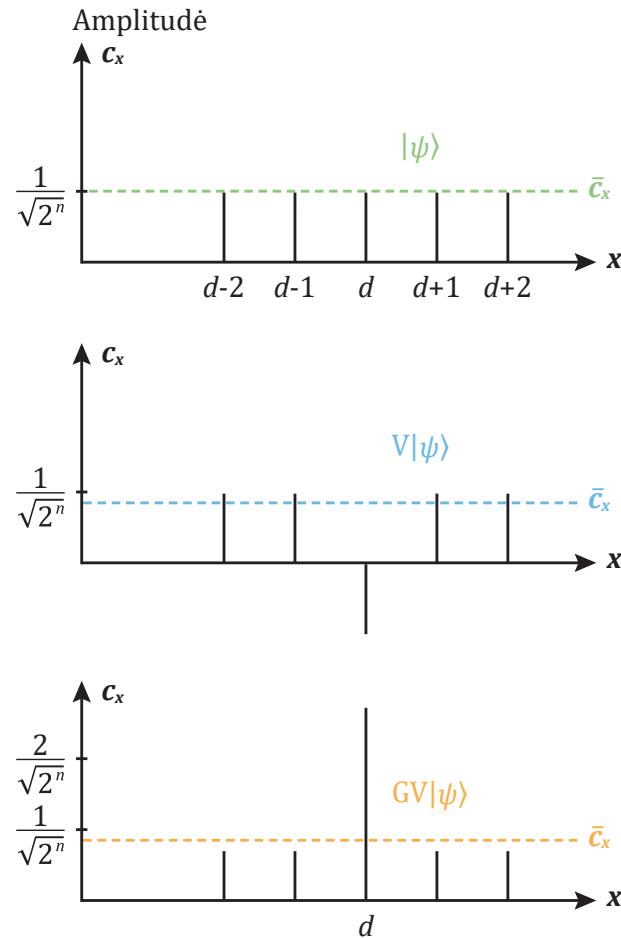
$$V|\psi\rangle = |\psi\rangle - \frac{2}{\sqrt{2^n}} |d\rangle; \quad (6.37)$$

$$V|d\rangle = -|d\rangle. \quad (6.38)$$

Operatoriaus G efektas:

$$G|\psi\rangle = |\psi\rangle; \quad (6.39)$$

$$G|d\rangle = \frac{2}{\sqrt{2^n}} |\psi\rangle - |d\rangle. \quad (6.40)$$



6.4 pav.: Būsenų $|x\rangle$ amplitudžių c_x pokytis pirmos Groverio iteracijos GV metu. Ieškomosios būsenos $|d\rangle$ amplitudė žymima d , visų būsenų amplitudžių vidurkis \bar{c}_x nurodytas brūkšniuota linija

Matome, kad individualūs V ir G (todėl ir bendra GV), veikdami $|\psi\rangle$ ir $|d\rangle$, sukuria kitas šių būsenų tiesines kombinacijas išlaikant amplitudes realiaisiais skaičiais. Visos tiesinės dviejų vektorių kombinacijos realioje vektorių erdvėje apibrėžia 2 dimensijų plokštumą. Yra pravartu iliustruoti šią plokštumą identifikuojant ortogonalius vektorius $\{|d\rangle, |d_{\perp}\rangle\}$; čia $|d_{\perp}\rangle$ yra statmenas ieškomajam $|d\rangle$ vektoriui. Tad pradinę būseną $|\psi\rangle$ perteikiamo jų sudėtimi: $|\psi\rangle = \langle\psi|d\rangle|d\rangle + \langle\psi|d_{\perp}\rangle|d_{\perp}\rangle$. Jeigu imsime, kad θ yra kampus tarp $|\psi\rangle$ ir $|d_{\perp}\rangle$, tada $\langle\psi|d_{\perp}\rangle = \cos(\theta)$ ir $\langle\psi|d\rangle = \cos(\frac{\pi}{2} - \theta) = \sin(\theta)$. Taigi bendrą būseną $|\phi\rangle$ šioje 2 dimensijų plokštumoje galima išreikšti ir taip:

$$|\phi\rangle = \sin(\theta)|d\rangle + \cos(\theta)|d_{\perp}\rangle. \quad (6.41)$$

Groverio algoritme GV iteracijos atlieka registro būseną nusakančio vektoriaus $|\phi\rangle$ posūkį link $|d\rangle$ vektoriaus. Tikimybė rasti $|d\rangle$ būseną yra $p = |\langle\phi|d\rangle|^2 = \sin^2(\theta)$, todėl tikslas yra pasiekti kampą θ , kuo artimesnį $\frac{\pi}{2}$, padarant $p \approx 1$. Kadangi $\langle\psi|d\rangle = \frac{1}{\sqrt{2^n}}$ yra artimas nuliu kai $2^n \gg 1$, $|\psi\rangle$ ir $|d\rangle$ yra beveik lygiagretieji vektoriai. Kampas θ tokiu atveju yra labai mažas ir todėl $\langle\psi|d\rangle = \sin(\theta) \approx \theta$. Taip randame pradinį kampą θ tarp $|\psi\rangle$ ir $|d_{\perp}\rangle$. Kampas tarp $|d\rangle$ ir būsenos $|\psi\rangle$ yra $\frac{\pi}{2} - \theta$, tad reikalingas iteracijų skaičius k randamas iš lygybės $2\theta k = \frac{\pi}{2} - \theta$, arba $k = \frac{\pi}{4\theta} - \frac{1}{2} \approx \frac{\pi}{4}\sqrt{2^n}$, suapvalinus k iki artimiausio sveikojo skaičiaus.

Sugržkime dar kartą prie V ir pritaikykime šią transformaciją vektoriais $\{|d\rangle, |d_{\perp}\rangle\}$ perteiktais pradinei būsenai, $V|\psi\rangle = -\langle\psi|d\rangle|d\rangle + \langle\psi|d_{\perp}\rangle|d_{\perp}\rangle$. Ši išraiška parodo, kad vektoriaus komponentas lygiagretus $|d_{\perp}\rangle$ lieka nepakeistas, o komponentas lygiagretus $|d\rangle$ iجاuna minuso ženklą, $\langle\psi|d\rangle \rightarrow -\langle\psi|d\rangle$. Kadangi $\langle\psi|d\rangle \approx \theta$, matome, kad $V|\psi\rangle$ dabar sudaro kampą $-\theta$ su ašimi, nusakyta $|d_{\perp}\rangle$, tad ši transformacija pasuko $|\psi\rangle$ kampu 2θ pagal laikrodžio rodyklę. Jeigu pradinis kampus būtų $-\theta$, V pasuktu $|\psi\rangle$ kampu 2θ priėš laikrodžio rodyklę. Toks simetriškas vektoriaus pasukimas apie ašį yra vadinamas **atspindžiu** (angl. *reflection*). Pritaikius V bendrai būsenai $|\phi\rangle$, kurią galima išreikšti vektoriais $|d\rangle$ ir $|d_{\perp}\rangle$, $V|\phi\rangle$ geometriškai nusako $|\phi\rangle$ atspindį ašies nusakytos $|d_{\perp}\rangle$ vektoriumi, atžvilgiu.

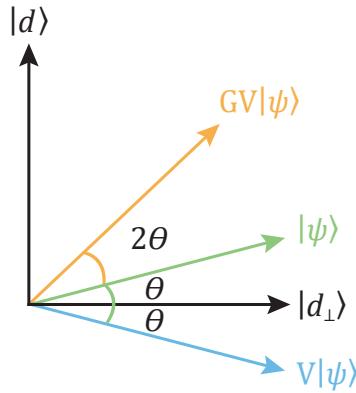
Transformacijos G efektas yra taip pat atspindėti $|\phi\rangle$ šioje plokštumoje, tačiau atžvilgiu ašies lygiagrečios pradinės būsenos $|\psi\rangle$ vektoriui. Norėdami tuo įsitikinti, išreikškime $|\phi\rangle$ komponentais paraleliai ir statmenai pradinei būsenai $|\psi\rangle$, $|\phi\rangle = |\psi_{\parallel}\rangle + |\psi_{\perp}\rangle$. Lygiagretus komponentas randamas $|\psi_{\parallel}\rangle = \langle\psi|\phi\rangle|\psi\rangle$, tad statmenas $|\psi_{\perp}\rangle = |\phi\rangle - \langle\psi|\phi\rangle|\psi\rangle$. Norėdami rasti $|\phi\rangle$ vektoriaus atspindį $|\psi\rangle$ atžvilgiu, atimame iš jo du statmenus komponentus:

$$\begin{aligned} |\phi\rangle - 2|\psi_{\perp}\rangle &= |\phi\rangle - 2(|\phi\rangle - \langle\psi|\phi\rangle|\psi\rangle) = 2|\psi\rangle\langle\psi|\phi\rangle - |\phi\rangle \\ &= (2|\psi\rangle\langle\psi| - I)|\phi\rangle. \end{aligned} \quad (6.42)$$

Paskutinėje eilutėje atpažįstame skliausteliuose G operatorių, tai patvirtina atspindžio efektą. Tad GV iteracija pritaiko du atspindžius arba, kitaip tariant, du vektorių pasukimus, vieną apie $|d_{\perp}\rangle$ ašį ir kitą apie $|\psi\rangle$.

Pradedant algoritmą ir atlikus $V|\psi\rangle$, $|\psi\rangle$ yra pasukamas 2θ kampu pagal laikrodžio rodyklę (žr. 6.5 pav.). Toliau pritaikius G šiai būsenai, ji yra dar kartą atspindima, ši kartą apie $|\psi\rangle$, todėl vektorius pasukamas prieš laikrodžio rodyklę. Kadangi kampus tarp $V|\psi\rangle$ ir $|\psi\rangle$ yra 2θ , kampus tarp $GV|\psi\rangle$ ir $V|\psi\rangle$ yra 4θ . Tad vienos iteracijos GV dėka pradinė $|\psi\rangle$ pasisuka 2θ kampu link $|d\rangle$.

Tolesnėje iteracijoje V vėl atspindi $GV|\psi\rangle$ būseną apie $|d_{\perp}\rangle$, ir ši pasisuka 6θ pagal laikrodžio rodyklę, nes sudaro 3θ kampą su $|d_{\perp}\rangle$. Tada pritaikius G būsenai $VGV|\psi\rangle$, ši atspindima apie $|\psi\rangle$ ir dėl šios priežasties pasisuka dar 2θ kampu link $|d\rangle$. Taip kiekviena GV iteracija pasuka registro būseną 2θ kampu arčiau ieškomosios $|d\rangle$.



6.5 pav.: Geometrinė Groverio algoritmo iliustracija. Pradinės registro būsenos (vektorius) $|\psi\rangle$ pasukimas vienos iteracijos GV metu link ieškomosios būsenos $|d\rangle$. Ieškomoji būsena, kartu su jai statmena $|d_{\perp}\rangle$, apibūdina 2 dimensijų realų poerdvį (plokštumą) visoje 2^n dimensijų kubitų registro būsenų erdvėje

6.6.3 Groverio paieška su $N = 8$

Čia iliustruojame Groverio paieškos algoritmą duomenų bazėje, sudarytoje iš $N = 8$ elementų pasitelkiant 3 kubitų registrą. Sakykime, kad $|d\rangle = |101\rangle = |5\rangle$ yra ieškomoji būsena. Orakulo funkcija, pažyminti $|101\rangle$ būseną matricos forma:

$$V = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6.43)$$

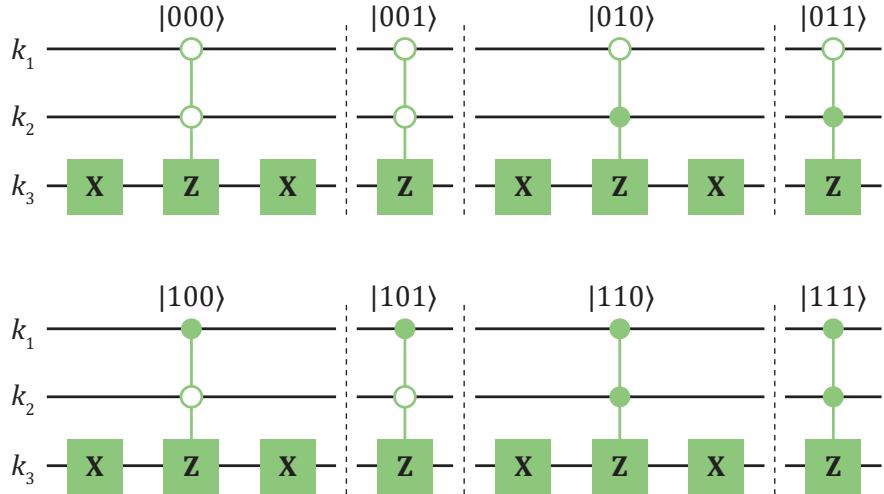
Naudosime dvejetainę sistemą siekdamis aiškiau iliustruoti orakulo funkcijos efektą. Tiesiogiai pritaikę orakulo funkciją $V = -2|d\rangle\langle d| + I$, pakeičiančią ieškomosios būsenos $|d\rangle = |101\rangle$ fazę, randame:

$$V|\psi\rangle = \frac{|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle - |101\rangle + |110\rangle + |111\rangle}{\sqrt{8}}. \quad (6.44)$$

Visos įmanomos 3 kubitų orakulo funkcijos, atliekančios diagonaliajų transformaciją ir pažymintios vieną skaičiuojamąjį bazinį vektorių iš 8 skirtinių, yra iliustruotos 6.6 pav. loginiai vartais. Juose 3 kubitų loginiai vartai gali būti perteikti, pavyzdžiui, Tofoli vartų dekompozicijos metodu, parodytu IV skyriuje.

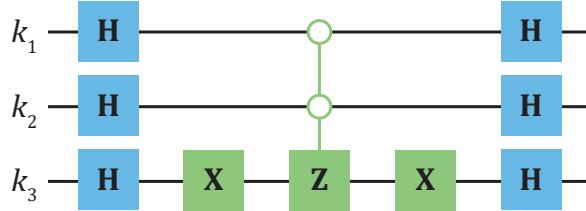
Operatorius $G = 2|\psi\rangle\langle\psi| - I$ išraišką galima rasti pirmiausiai atkreipiant dėmesį, kad $|\psi\rangle = H^{\otimes 3}|000\rangle$, tad G , veikiantis 3 kubitų registrą, užrašomas taip:

$$G = 2H^{\otimes 3}|000\rangle\langle 000|H^{\otimes 3} - I = H^{\otimes 3}(2|000\rangle\langle 000| - I)H^{\otimes 3}. \quad (6.45)$$



6.6 pav.: Visos galimos 3 kubitų registro orakulo funkcijas (V operatorių) realizuojančios loginės grandinės, atskirtos viena nuo kitos brūkšniuota linija. Viršuje pažymėtos būsenos, kurioms atitinkama grandinė pritaiko santykinę fazę

6.45 eilutėje panaudojome $(H^{\otimes 3})^\dagger = H^{\otimes 3}$ ir $H^{\otimes 3}IH^{\otimes 3} = I$. Skliausteliuose matome transformaciją $(2|000\rangle\langle 000| - I)$, kuri veikdama bazinius vektorius įveda santykinę π fazę tarp $|000\rangle$ ir visų likusių būsenų. Norėdami realizuoti ši operatorių kvantinėje grandinėje, atkreipiame dėmesį, kad galima identiškai perrašyti ši operatorių $(2|000\rangle\langle 000| - I) \rightarrow -(I - 2|000\rangle\langle 000|)$, faktorizuojant globalią (neturinčią įtakos) fazę. Šioje formoje minuso ženklas priskiriamas $|000\rangle$ būsenai, o ne visoms likusioms būsenoms. Kvantinėje grandinėje G operatorius 3 kubitų registrui yra perteikiamas 6.7 pav.:



6.7 pav.: Groverio 3 kubitų paieškos algoritme G operatorių realizuojanti loginė grandinė

Turime visus ingredientus atlikti Groverio paiešką. Norint rasti reikiama GV iteracijų skaičių, pirmiausiai nustatome pradinį kampą θ :

$$\langle \psi | d \rangle = \cos\left(\frac{\pi}{2} - \theta\right) = \sin(\theta) = 1/\sqrt{8}. \quad (6.46)$$

Randame $\theta = \arcsin\left(\frac{1}{\sqrt{8}}\right) \approx 20.7^\circ$ ir todėl 3-kubitų dydžio registro atveju pasiekti 90° kampą tiksliai neišeis. Po vienos GV iteracijos kampas tampa $\theta \approx 62.1^\circ$, o dviejų $\theta \approx 103.5^\circ$. Tikiybė rasti $|d\rangle = |101\rangle$ būseną yra $p = \sin^2(\theta)$, todėl po vienos iteracijos ji yra $p = 0.78$, o po dviejų $p = 0.95$ ir toliau pradeda mažėti, nes vektorius yra prasukamas per toli.

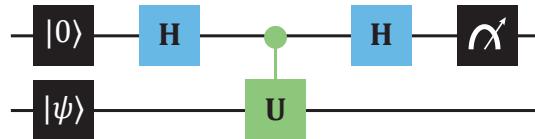
Praktiniuose taikymuose mus domina paieška didelėje duomenų bazėje. Šioje situacijoje būtent ir galime rasti atsakymą vienu matavimu su praktiškai $p = 1$ tikimybe. Mat didėjant įvesties būsenų skaičiui N , pradinis kampus θ mažėja. Na, o kadangi kiekviena GV iteracija pasuka vektorių 2θ kampu, mažesnis pradinis kampus θ leidžia tiksliau priartinti $|\phi\rangle$ prie ieškomosios $|d\rangle$ būsenos ir ją amplifikuoti.

6.7 Hadamardo ir SWAP testai

Šioje dalyje pristatome plačiai žinomus ir algoritmuose aptinkamus Hadamardo ir SWAP testus (angl. *Hadamard, SWAP tests*). Jie yra naudojami kaip moduliai ir procedūros kvantiniuose algoritmuose, leidžiantys apytikriai apskaičiuoti tokų narių reikšmes: unitariojo operatoriaus U tikrinį verčių vidurkį $\langle U \rangle = \langle \psi | U | \psi \rangle$, dviejų būsenų vidinę sandaugą $\langle \phi | \psi \rangle$, ir vidinės sandaugos kvadratą $|\langle \phi | \psi \rangle|^2$.

6.7.1 Hadamardo testas

Kvantinės grandinės, realizuojančios Hadamardo, modifikuotą Hadamardo ir SWAP testus panaudoja ancila 1 kubito registrą bei papildomą registrą (ar registrus) koduojančius būsenas, kurių atžvilgiu apskaičiuojami minėti nariai. Šios grandinės pasižymi identiška loginių vartų seką: Hadamardo vartai yra pritaikomi ancila kubitui prieš ir po sąlyginių loginių vartų cU , kontroluojamų ancila kubito būsenų, taip pat išmatuojama ancila kubito būsena. Hadamardo testą realizuojanti grandinė yra iliustruota 6.8 pav.



6.8 pav.: Hadamardo testą atliekanti loginė grandinė

Matome ancila kubitą pradinėje $|0\rangle$ būsenoje ir antrą kubitas bendroje $|\psi\rangle$ būsenoje, tad turime faktorizuojamą $|0\rangle \otimes |\psi\rangle$. Darome prielaidą, kad galime pakartotinai paruošti $|\psi\rangle$. Atlikus Hadamardo loginius vartus ancilai randame:

$$H \otimes I(|0\rangle \otimes |\psi\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |\psi\rangle. \quad (6.47)$$

Toliau atliekami kontroluojamai U antrajam kubitui, vadinkime juos cU :

$$cU \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi\rangle + |1\rangle \otimes U|\psi\rangle). \quad (6.48)$$

Ir dar vieni H ancila kubitui:

$$\begin{aligned} H \otimes I \frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi\rangle + |1\rangle \otimes U|\psi\rangle) = \\ \frac{1}{2} \left((|0\rangle + |1\rangle) \otimes |\psi\rangle + (|0\rangle - |1\rangle) \otimes U|\psi\rangle \right). \end{aligned} \quad (6.49)$$

Šią būseną, vadinkime ja $|\chi\rangle$, galime pergrupuoti taip:

$$|\chi\rangle = |0\rangle \otimes \left(\frac{I+U}{2} \right) |\psi\rangle + |1\rangle \otimes \left(\frac{I-U}{2} \right) |\psi\rangle. \quad (6.50)$$

Sekant parodytą kvantinę grandinę, išmatuojame ancila kubito būseną. Toliau apskaičiuosime neselektyvaus būsenų matavimo rezultataą, suteikiantį ancila kubito tikrinių verčių $\lambda_k \in \{1, -1\}$ vidurkį. Tam formaliai naudojame operatorių $Z \otimes I$, tad turėsime apskaičiuoti $\langle \chi | Z \otimes I | \chi \rangle$. Taikant projekcinę dekompoziciją:

$$Z \otimes I = \sum_k \lambda_k P_k \otimes I = \lambda_0 |0\rangle\langle 0| \otimes I + \lambda_1 |1\rangle\langle 1| \otimes I = |0\rangle\langle 0| \otimes I - |1\rangle\langle 1| \otimes I. \quad (6.51)$$

Randame $\langle \chi | Z \otimes I | \chi \rangle$:

$$\begin{aligned} \langle \chi | Z \otimes I | \chi \rangle &= \left[\langle 0 | \otimes \langle \psi | \left(\frac{I+U^\dagger}{2} \right) + \langle 1 | \otimes \langle \psi | \left(\frac{I-U^\dagger}{2} \right) \right] (Z \otimes I) \\ &\times \left[|0\rangle \otimes \left(\frac{I+U}{2} \right) |\psi\rangle + |1\rangle \otimes \left(\frac{I-U}{2} \right) |\psi\rangle \right]. \end{aligned} \quad (6.52)$$

Atlikus vidines sandaugas:

$$\langle \chi | Z \otimes I | \chi \rangle = \frac{1}{2} \langle \psi | \psi \rangle + \frac{1}{2} \langle \psi | U + U^\dagger | \psi \rangle = \frac{1}{2} \langle \psi | \psi \rangle + \text{Re}[\langle \psi | U | \psi \rangle]. \quad (6.53)$$

Primename, kad unitariojo operatoriaus U tikrinės vertės λ_k yra kompleksiniai skaičiai su vieniniu moduliu ir forma $\lambda_k = e^{i\theta_k}$. Paskutinėje eilutėje panaudojome $\langle \psi | U^\dagger | \psi \rangle = (\langle \psi | U | \psi \rangle)^\dagger$, tad bendrai, sudėjė kompleksinį skaičių $z \equiv \langle \psi | U | \psi \rangle = a + ib$ ir jo kompleksinę jungtį $z^* \equiv \langle \psi | U^\dagger | \psi \rangle = a - ib$ gauname realiąją z skaičiaus dalį $(2a)$. Kaip matome, šis rezultatas tiesiogiai leidžia apskaičiuoti U operatoriaus tikrinių verčių vidurkio realiąją dalį $\text{Re}[\langle \psi | U | \psi \rangle]$.

Žinoma, mus gali taip pat dominti ir menamoji $\langle \psi | U | \psi \rangle$ nario dalis, $\text{Im}[\langle \psi | U | \psi \rangle]$. Ją irgi prastai randame pakoregavę Hadamardo testo grandinę. Po pirmųjų Hadamardo loginių vartų, pritaikytų ancila kubitui, pritaikome jam papildomus vartus S^\dagger , suteikiančius $|1\rangle$ būsenai fazę, $|1\rangle \rightarrow -i|1\rangle$:

$$S^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}. \quad (6.54)$$

Tada vėl atliekami vartai cU ir $H \otimes I$:

$$\begin{aligned} (H \otimes I)cU \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \otimes |\psi\rangle &= (H \otimes I) \frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi\rangle - i|1\rangle \otimes U|\psi\rangle) \\ &= |0\rangle \otimes \left(\frac{I - iU}{2} \right) |\psi\rangle + |1\rangle \otimes \left(\frac{I + iU}{2} \right) |\psi\rangle. \end{aligned} \quad (6.55)$$

Vadindami šią būseną $|\chi\rangle$, apskaičiuojame $\langle \chi | Z \otimes I | \chi \rangle$:

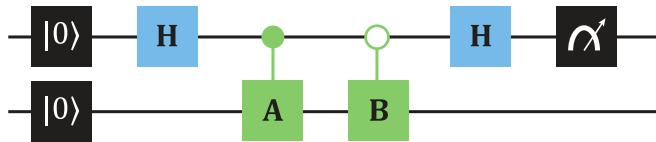
$$\langle \chi | Z \otimes I | \chi \rangle = \frac{1}{2} \langle \psi | \psi \rangle + \frac{i}{2} \langle \psi | U^\dagger - U | \psi \rangle = \frac{1}{2} \langle \psi | \psi \rangle + \text{Im}[\langle \psi | U | \psi \rangle]. \quad (6.56)$$

Viršuje irgi pritaikėme kompleksinių skaičių aritmetiką, $i(z^* - z) = i(-2ib) = 2b$, taip rasdami menamąją dalį $\text{Im}[\langle \psi | U | \psi \rangle]$. Skaičiavimo išteklių atžvilgiu Hadamardo testas bus įvykdytas efektyviai, jeigu galime efektyviai paruošti $|\psi\rangle$ ir atlikti cU . Atkreipiame dėmesį, kad antrojo registro būsena po ancilos matavimo yra žinoma kaip $(\frac{I+iU}{2})|\psi\rangle$, su ženklu + arba -, kuris priklauso nuo rastos ancila kubito būsenos. Hadamardo testo algoritmo laiko sudėtingumas auga kaip $O(1/\epsilon)$ su norimu pasiekti tikslumu ϵ .

6.7.2 Modifikuotas Hadamardo testas

Šis testas suteikia būdą apytikriai apskaičiuoti dviejų kvantinių būsenų, $\{|\psi\rangle, |\phi\rangle\} \in V$, kurių kiekviena yra sudaryta iš n kubitų, vidinę sandaugą $\langle\psi|\phi\rangle$. Dviejų normuotųjų būsenų vidinės sandaugos reikšmė yra bendrai kompleksinis skaičius, o modulis $|\langle\psi|\phi\rangle| \leq 1$.

Turime du registrus, kurių pradinė būsena yra $|0\rangle \otimes |0\rangle$. Pirmasis registras yra 1 kubito ancila, o štai antrasis registras turi n kubitų ir naudojamas paruošti $|\psi\rangle$ bei $|\phi\rangle$ būsenų superpozicijai. Darome prielaidą, kad galime įvykdyti unitarišias transformacijas A ir B pradinei registro būsenai $|0\rangle$, kurios leidžia paruošti norimas būsenas: $A|0\rangle = |\psi\rangle$ ir $B|0\rangle = |\phi\rangle$. Šis algoritmas skiriasi nuo Hadamardo testo tuo, kad cU tarp Hadamardo vartų yra pakeičiamas dvejais sąlyginiais loginiais vartais cA ir cB (žr. 6.9 pav.).



6.9 pav.: Modifikuotą Hadamardo testą atliekanti loginė grandinė

Pirmausiai atliekame Hadamardo vartus ancila kubitui:

$$(H \otimes I)|0\rangle \otimes |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle. \quad (6.57)$$

Toliau atliekami dveji sąlyginiai vartai cA ir cB antrajam registrui, tačiau kontroliuojame skirtingose ancilos būsenose. Pritaikome A , jeigu ancilos kubito būsena yra $|0\rangle$, ir B , jeigu ancilos būsena yra $|1\rangle$. Kadangi ancila yra superpozicijoje, randame:

$$cBcA \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi\rangle + |1\rangle \otimes |\phi\rangle). \quad (6.58)$$

Atliekame H ancilai:

$$\begin{aligned} (H \otimes I) \frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi\rangle + |1\rangle \otimes |\phi\rangle) &= \frac{1}{2}|0\rangle \otimes (|\psi\rangle + |\phi\rangle) + \frac{1}{2}|1\rangle \otimes (|\psi\rangle - |\phi\rangle) \\ &\equiv |\chi\rangle. \end{aligned} \quad (6.59)$$

Galiausiai išmatuojame ancilos būseną, apskaičiuodami tikrinių verčių vidurkį $\langle\chi|Z \otimes I|\chi\rangle$:

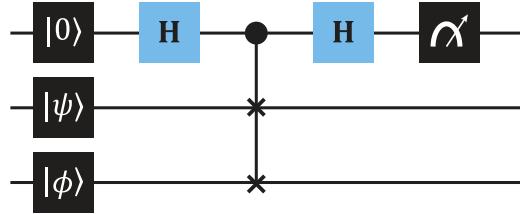
$$\langle\chi|Z \otimes I|\chi\rangle = \text{Re}[\langle\psi|\phi\rangle]. \quad (6.60)$$

Šis modifikuotas Hadamardo testas suteikia realiąją vidinės sandaugos $\langle\psi|\phi\rangle$ dalį. Menamąją dalį $\text{Im}[\langle\psi|\phi\rangle]$ galime rasti analogiškai kaip ir Hadamardo teste, pritaikę ancila kubitui papildomus vartus S^\dagger .

6.7.3 SWAP testas

Šis testas (žr. 6.10 pav.) leidžia apskaičiuoti dviejų n kubitų kvantinių būsenų, $\{|\psi\rangle, |\phi\rangle\} \in V$, vidinės sandaugos kompleksinį kvadratą $|\langle\phi|\psi\rangle|^2$. Šis neneigiamasis skaičius, $|\langle\phi|\psi\rangle|^2 \leq 1$, parodo, kiek būsenos persikloja ir yra panašios. Kvantinė grandinė realizuojant šią procedūrą yra sudaryta iš trijų registrų: 1 kubito ancilos ir dviejų n kubitų registrų, koduojančių būsenas

$|\psi\rangle$ ir $|\phi\rangle$. Galime daryti prielaidą, kad šios dvi būsenos yra mums pateikiamos po prieš tai atlikto skaičiavimo, arba iš trečiosios šalies, pavyzdžiui, atsiųstos kvantiniu ryšiu. Tad pradinė faktorizuojama 3 kubitų būsena yra $|0\rangle \otimes |\psi\rangle \otimes |\phi\rangle$. Be šių skirtumų tarp Hadamardo testo, cU čia yra salyginiai *SWAP* (*Fredkin*) loginiai vartai, cW , kontroliuojami ancila kubito.



6.10 pav.: SWAP testo loginė grandinė

Pirmausiai, atlikę Hadamardo vartus ancila kubitui, turime:

$$\frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi\rangle \otimes |\phi\rangle + |1\rangle \otimes |\phi\rangle \otimes |\psi\rangle). \quad (6.61)$$

Toliau atliekami Fredkin loginiai vartai, sukeičiantys antrojo ir trečiojo registro būsenas vietomis:

$$\begin{aligned} cW \frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi\rangle \otimes |\phi\rangle + |1\rangle \otimes |\phi\rangle \otimes |\psi\rangle) = \\ \frac{1}{\sqrt{2}}(|0\rangle \otimes |\phi\rangle \otimes |\psi\rangle + |1\rangle \otimes |\psi\rangle \otimes |\phi\rangle). \end{aligned} \quad (6.62)$$

Pritaikius antrus Hadamardo loginius vartus ancila kubitui $H \otimes I \otimes I$, galutinė būsena $|\chi\rangle$ yra:

$$|\chi\rangle = \frac{1}{2}|0\rangle \otimes (|\psi\rangle \otimes |\phi\rangle + |\phi\rangle \otimes |\psi\rangle) + \frac{1}{2}|1\rangle \otimes (|\psi\rangle \otimes |\phi\rangle - |\phi\rangle \otimes |\psi\rangle). \quad (6.63)$$

Išmatuojame ancilos būseną, apskaičiuodami tikriniu verčių vidurkį $\langle \chi | Z \otimes I | \chi \rangle$:

$$\langle \chi | Z \otimes I | \chi \rangle = |\langle \phi | \psi \rangle|^2. \quad (6.64)$$

Tai tiesiogiai suteikia norimą reikšmę. Jos tikslumas gali būti pasiektas pageidaujamo dydžio didinant SWAP testo skaičių pateikiamoms identiškoms būsenoms $\{|\psi\rangle, |\phi\rangle\}$. SWAP testo algoritmo laiko sudėtingumas auga kaip $O(1/\epsilon^2)$ su siekiamu tikslumu ϵ . Norint atlikti SWAP testą dviem n kubitų būsenoms, klasikinių algoritmų laiko sudėtingumas auga eksponentiškai su kubitų skaičiumi n , o štai kvantinės SWAP testo grandinės gylis auga tiesiškai. Tai suteikia eksponentinį paspartinimą.

Pabaigoje trumpai pakomentuosime stebimą Hadamardo ir SWAP testų kvantinės grandinės bendrąjį efektą. Ancila kubitas, pasitelkus Hadamardo vartus, yra pastatomas į būseną superpoziciją, ir šios dvi būsenos supinamos su skirtingomis antrojo registro būsenomis, koduojančiomis $\{|\psi\rangle, |\phi\rangle\}$. Kvantinės optikos požiūriu, tai padalija antrą registrą į du skirtinges optinius „kelius“, kuriuose kvantinės būsenos patiria skirtinges loginius vartus U . Antrieji Hadamardo vartai, prietaikyti ancila kubitui, šiuos du kelius vėl sugrąžina į vieną. Tai priverčia antrojo registro kvantinę būseną patirti interferencinius efektus, kurie matavimuose sekā iš vidinės sandaugos tipo atsirančių narių, $\langle \chi | U | \chi \rangle$ ir $\langle \phi | \psi \rangle$.

VII skyrius

Furjė transformacija ir jos taikymai

Šiame skyriuje toliau tesiame kvantinių skaičiavimų apžvalgą pristatydami **Furjė transformaciją** (angl. *Fourier transform*) ir ja pagrįstus algoritmus. Klasikinė Furjė transformacija yra plačiai naudojama atliekant duomenų ir signalų analizę bei apdorojimą ir yra esminės reikšmės įrankis matematinėje funkcijų analizeje. Kaip praktinę Furjė naudojimo pavyzdį, nebūtinai sunku skaičiavimo išteklių atžvilgiu, imkime funkciją $f(t)$, kuri nusako garso šaltinio kitimą laike. Šios funkcijos Furjė transformacija, žymima $\text{FT}[f(t)] = f(v)$, išreiškia ją dažnių (v) spektrą pavidale $f(v)$. Norėdami atlirkti pasirinktų dažnių filtravimą, pavyzdžiu, siekiant nuslopinti aukštojo dažnio garsus, galime užmaskuoti šias amplitudes dažnio srities funkcijoje $f(v)$. Atlirkę atvirkštinę $f(v)$ funkcijos Furjė transformaciją, žymimą $\text{FT}^{-1}[f(v)] = f(t)$, grąžiname ją atgal į laiko sritį, taip atstatydamis modifikuotą garso įrašą.

7.1 Kvartinė Furjė transformacija

Kvartinė Furjė transformacija (angl. *quantum Fourier transform*) yra klasikinės Furjė transformacijos kvantinis realizavimas. Palyginus su vadinamuoju **greituoju klasikiniu Furjė transformacijos algoritmu** (angl. *fast Fourier transform*), naudojamu diskretiesiems (skaitmeniuziems) signalams, kvantinė jos versija pasiekia eksponentinį pagreitinimą loginių operacijų skaičiaus atžvilgiu. Todėl kvartinė Furjė transformacija atveria galimybes paspartinti aibę skaičiavimo užduočių. Tarp jų yra kvantinių sistemų modeliavimas, tiesinių lygčių sprendinių paieška, kvantinio mašinorio mokymosi algoritmai, Šoro pirminių skaičių faktorizavimas.

Kvartinė Furjė transformacija (vartosime trumpinį FT) priima kaip įvestį n elementų vektorių ir pateikia išvestyje kitą n elementų vektorių. FT yra realizuojama unitariaja transformacija, vadinkime ją U_{FT} , kurios efektas n kubitų skaičiuojamiesiems baziniams vektoriams $|x\rangle$ nusakomas:

$$U_{\text{FT}}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{i2\pi xy/2^n} |y\rangle. \quad (7.1)$$

Čia x ir y yra skaičiai, išreikšti dešimtaine sistema. Iš to gaunama U_{FT} operatoriaus matematinė išraiška diodomis:

$$U_{\text{FT}} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^n-1} e^{i2\pi xy/2^n} |y\rangle\langle x|. \quad (7.2)$$

Matome, kad U_{FT} , veikiantis bet kurį bazinių vektorių $|x\rangle$, transformuoja jį į lygią visų standartinių bazinių vektorių superpoziciją. Šie vektoriai yra perindeksuojami raide y ir kartu nešasi skirtingus fazės narius $e^{i2\pi xy/2^n}$. Šios eksponentės argumentuose xy yra daugyba, pavyzdžiu, jeigu $|x\rangle = |5\rangle$ ir $|y\rangle = |3\rangle$, tada $xy = 15$. Operatorių U_{FT} , žinoma, galima išreikšti ir $(2^n \times 2^n)$ dydžio matrica. Jos y stulpelio ir x eilutės įrašai yra fazės nariai $e^{i2\pi xy/2^n} / \sqrt{2^n}$.

Būdama unitarioji transformacija, FT išlaiko vektorių normuotumą ir ortogonalumą $\langle x'|U_{FT}^\dagger U_{FT}|x\rangle = \langle x'|x\rangle = 0$. Tad U_{FT} leidžia sukurti naują n kubitų bazinių vektorių rinkinį, vadinamą **Furjė bazinių vektorių rinkiniu** (angl. *Fourier basis*). Viršuje U_{FT}^\dagger yra U_{FT} operatoriaus ermitinė jungtis ir nusako atvirkštinę Furjė transformaciją, dar žymimą FT^\dagger arba FT^{-1} . Pritaikius $U_{FT}^\dagger|x\rangle$ skirtumas nuo viršuje parodytos išraiškos bus tik kompleksinėje fazės narių jungtyje $e^{i2\pi xy/2^n} \rightarrow e^{-i2\pi xy/2^n}$. Ar minuso ženklas vartojamas U_{FT}^\dagger , ar U_{FT} , neturi esminės įtakos, svarbu tik sistemiškai juos vartoti.

Pritaikę FT bendrai būsenai $|\psi\rangle$, esančiai bazinių vektorių $|x\rangle$ superpozicijoje, randame:

$$U_{FT}|\psi\rangle = U_{FT} \sum_{x=0}^{2^n-1} c_x |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^n-1} c_x e^{i2\pi xy/2^n} |y\rangle. \quad (7.3)$$

Atkreipiame dėmesį, kad $U_{FT}|\psi\rangle$ nusako faktorizuojamą kvantinę n kubitų būseną. Šią išraišką galima supaprastinti:

$$U_{FT}|\psi\rangle = \sum_{y=0}^{2^n-1} b_y |y\rangle. \quad (7.4)$$

Čia b_y yra gaunamas laikant y fiksuočią ir atliekant sumą x atžvilgiu, tai formaliai nusako funkciją, priklausančią tik nuo argumento y :

$$b_y = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} c_x e^{i2\pi xy/2^n}. \quad (7.5)$$

Toliau pateikiame keletą kvantinės Furjė transformacijos pavyzdžių. Vieno kubito FT, $n = 1$, yra randama:

$$U_{FT}|x\rangle = \frac{1}{\sqrt{2}} \sum_{y=0}^1 e^{i2\pi xy/2} |y\rangle = \frac{1}{\sqrt{2}} (|0\rangle + e^{i\pi x} |1\rangle). \quad (7.6)$$

Fazės narys $e^{i\pi x} = 1$, jeigu U_{FT} veikia $|x\rangle = |0\rangle$, ir $e^{i\pi x} = -1$, jeigu U_{FT} veikia $|x\rangle = |1\rangle$. Tad $n = 1$ Furjė transformaciją galima išreikšti tiesiog Hadamardo loginiai vartais, $U_{FT} = H$. Taip pat matome ir kitą specifinę situaciją, kai Furjė transformacija pritaikoma n kubitų registrui, esančiam $|0\rangle$ būsenoje (dešimtainėje sistemoje):

$$U_{FT}|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} |y\rangle. \quad (7.7)$$

Jos efektą galima taip pat nusakyti kiekvienam iš n kubitų pritaikant Hadamardo vartus, $H^{\otimes n}$, taip sukuriant lygią visų bazinių vektorių superpoziciją.

Paskutiniame pavyzdyme imkime vieno iš bazinių vektorių $|x\rangle$, nusakančio $n = 3$ kubitų registro

būseną, Furjė transformaciją:

$$\begin{aligned}
 U_{\text{FT}}|x\rangle &= \frac{1}{\sqrt{8}} \sum_{y=0}^7 e^{i2\pi xy/8} |y\rangle \\
 &= \frac{1}{\sqrt{8}} (|0\rangle + e^{i\pi x/4}|1\rangle + e^{i\pi x/2}|2\rangle + e^{i\pi x/4}|3\rangle \\
 &\quad + e^{i\pi x}|4\rangle + e^{i\pi x/4}|5\rangle + e^{i\pi x/2}|6\rangle + e^{i\pi x/4}|7\rangle).
 \end{aligned} \tag{7.8}$$

Toliau pateikiame itin naudingą n kubitų bazinio vektoriaus $|x\rangle$ kvantinės Furjė transformacijos formą naudojant pavienių kubitų n tenzorinę sandaugą:

$$\begin{aligned}
 U_{\text{FT}}|x\rangle &= \frac{1}{\sqrt{2^n}} \prod_{k=1}^n (|0\rangle + e^{\frac{i2\pi x}{2^k}}|1\rangle) \\
 &= (|0\rangle + e^{i2\pi x/2}|1\rangle) \otimes (|0\rangle + e^{i2\pi x/4}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{i2\pi x/2^n}|1\rangle).
 \end{aligned} \tag{7.9}$$

Tai nusako n kubitų, kurių kiekvienas yra superpozicijoje, tenzorinę sandaugą ir todėl – faktorizuojamąją būseną. Šią išraišką galima taip pat perteikti dvejetainė forma pasitelkiant kubitų numeraciją $|x\rangle = |k_1 k_2 \cdots k_n\rangle$ su $k_i \in \{0, 1\}$ ir dvejetainės trupmenos apribėžimą:

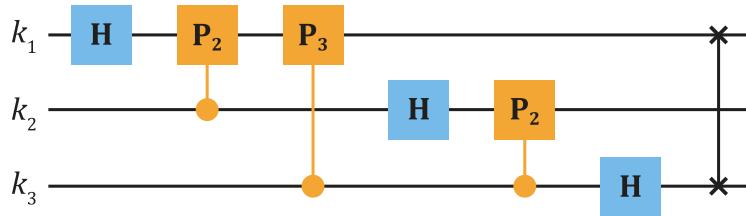
$$0.k_1 k_2 \cdots k_n = \frac{k_1}{2} + \frac{k_2}{4} + \cdots + \frac{k_n}{2^n} = \sum_{i=1}^n k_i 2^{-i}. \tag{7.10}$$

Keletas dvejetainių trupmenų pavyzdžių: $0.1=1/2$, $0.001=1/8$, $0.011=3/8$. Viską sudėjus kartu, $U_{\text{FT}}|x\rangle$ dvejetainė forma yra:

$$\begin{aligned}
 U_{\text{FT}}|k_1 k_2 \cdots k_n\rangle &= (|0\rangle + e^{i2\pi 0.k_n}|1\rangle) \otimes (|0\rangle + e^{i2\pi 0.k_{n-1}k_n}|1\rangle) \otimes \cdots \\
 &\quad \otimes (|0\rangle + e^{i2\pi 0.k_1 k_2 \cdots k_n}|1\rangle).
 \end{aligned} \tag{7.11}$$

7.2 Furjė transformacijos realizavimas kvantinėje grandinėje

Kvantinė Furjė transformacija yra efektyviai realizuojama kvantinėje grandinėje naudojant 1 kubito ir 2 kubitų loginius vartus. Pirmiausiai pateikiame kaip pavyzdį 3 kubitų registrui U_{FT} atliekančią kvantinę grandinę (žr. 7.1 pav.):



7.1 pav.: 3 kubitų registrui kvantinę Furjė transformaciją atliekanti loginė grandinė

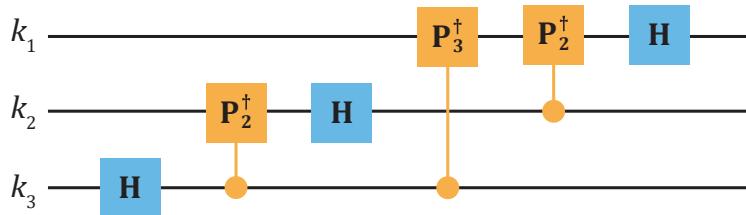
Čia matome Hadamardo, 2-kubitų cP_k bei SWAP loginų vartų kombinaciją. Joje cP_k yra IV skyriuje minėti sąlyginiai fazės vartai $cP_k(\theta)$ su $\theta = e^{i2\pi/2^k}$, k – sveikasis skaičius. Matricos

forma atrodo taip:

$$cP_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i2\pi/2^k} \end{bmatrix}. \quad (7.12)$$

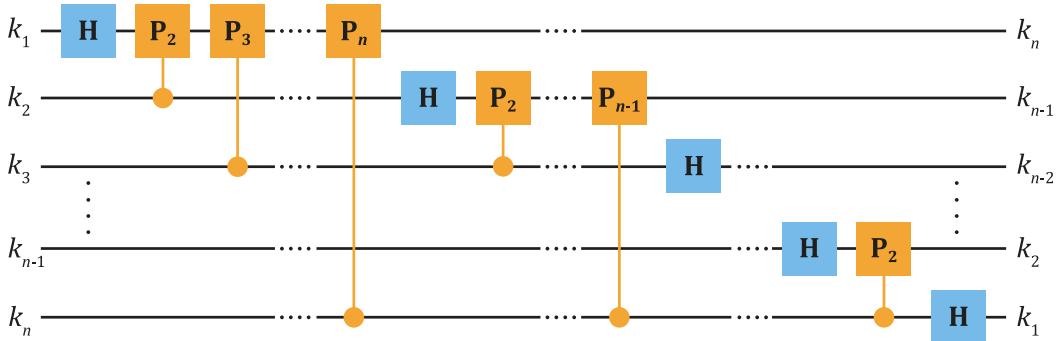
Vartai W panaudojami grandinės pabaigoje, kadangi FT algoritmas sukeičia kubitų indeksavimą aplink centrinius kubitus. Šiuo atveju k_1 sukeičiamas su k_3 , tad W , panaudoti grandinės pabaigoje, atstato juos atgal į standartinę $\{k_1, k_2, k_3\}$ seką. Aišku, nebūtina to daryti, jeigu toliau atliekami loginiai vartai atsižvelgia į indeksavimo pasikeitimą.

Atvirkštinė Furjė transformacija FT^\dagger yra realizuojama atvirkštine loginių vartų seką panaudojant atvirkštinius loginius vartus. Hadamardo bei SWAP loginiai vartai yra patys sau atvirkštiniai, o štai atvirkštiniai cP_k vartai yra jų ermitinė jungtis cP_k^\dagger . Atvirkštinė 3 kubitų transformaciją FT^\dagger atliekanti grandinė yra pateikta 7.2 pav.



7.2 pav.: Atvirkštinė 3 kubitų kvantinė Furjė transformacija

Furjė transformacijoje, atliekamoje n -kubitų registru, galime ižvelgti loginių vartų seką. Pradedant nuo viršutinio kubito k_1 , jam atliekami H bei sąlyginiai cP_k vartai poromis su visais $n - 1$ likusiais kubitais. Tai kartojama su k_2, k_3, \dots ir likusiais kubitais, paskutiniajam atliekant tik H .



7.3 pav.: Loginė grandinė, atliekanti kvantinę Furjė transformaciją n kubitų registru

Kaip minėjome, galima FT užbaigtį W loginiai vartais siekiant atstatyti visų kubitų indeksavimo eiliškumą. Apibendrinus, norint atlikti FT n kubitų registru yra panaudojami n skaičius H vartų, $n(n - 1)/2$ sąlyginiai cP_k , ir ne daugiau nei $n/2$ W vartų. Šie vartai gali būti realizuojami trimis cX , o cP_k galima realizuoti pasitelkus ne daugiau negu 6 elementariuosius loginius vartus. Tad kvantinės n kubitų Furjė transformacijos sudėtingumas yra nulemtas $O(n^2)$ elementarių logi-

nių operacijų, o štai klasikinei diskrečiajai Furjė transformacijai prireiks eksponentiškai daugiau operacijų, $O(n2^n)$.

Galima pastebėti, kad sąlyginiuose 2 kubitų vartuose cP_k adresatiniam kubitui pritaikoma fazė $\theta = e^{i2\pi/2^k}$ vis mažėja su didėjančiu k . Todėl FT atliekama registrui, sudarytam iš didelio skaičiaus kubitų, nuo tam tikro k skaičiaus galima atitinkamems kubitams nebetaikyti cP_k , nes θ bus nereikšmingai maža. Tai leidžia dar labiau sumažinti loginių vartų skaičių, atkreipiant dėmesį dar ir į tai, kad didėjantis k nusako atliekamus cP_k tarp vis toliau vienas nuo kito esančių kubitų.

7.3 Funkcijos periodiškumo paieška

Pirmame Furjė transformacijos taikymo pavyzdje parodysime funkcijos periodiškumo nustatymo algoritmą. Funkcija $f(x)$ yra periodinė su periodu P , jeigu $f(x) = f(x + P)$ visiems x funkcijos f apibrėžimo intervale. Vienas periodinės funkcijos pavyzdys būtų trigonometrinė funkcija $f(x) = \cos(2\pi x/P)$, nusakanti osciliacijas su periodu P . Klasikiniai algoritmai gali nustatyti P su eksponentiniu laiko sudėtingumu, augančiu didėjant įvesties dydžiui N . O štai paprastas kvantinis algoritmas, naudojantis FT, leidžia pasiekti eksponentinį paspartinimą.

Imkime šio algoritmo pavyzdį su įvesties bei išvesties registrais, turinčiais po 3 kubitus, ir funkciją $f(x)$, kurios periodas yra $P = 2$. Dėl šio periodiškumo funkcijos reikšmės lyginiuose ir nelyginiuose argumentuose yra nusakytos y' ir y'' vertėmis: $y' = f(0) = f(2) = f(4) = f(6)$ ir $y'' = f(1) = f(3) = f(5) = f(7)$. Pirmiausia pradinės būsenos $|\psi_0\rangle = |0\rangle \otimes |0\rangle$ įvesties registrui pritaikome Hadamardo vartus $H^{\otimes 3}$, kurie sukuria lygią visų bazinių vektorių superpoziciją:

$$(H^{\otimes 3} \otimes I)|\psi_0\rangle = \frac{1}{\sqrt{8}} \sum_{x=0}^7 |x\rangle \otimes |0\rangle. \quad (7.13)$$

Kitame žingsnyje pritaikome abu registrus veikiančią transformaciją U_f , kuri nusako periodinę funkciją $f(x)$:

$$|\psi_1\rangle = U_f \frac{1}{\sqrt{8}} \sum_{x=0}^7 |x\rangle \otimes |0\rangle = \frac{1}{\sqrt{8}} \sum_{x=0}^7 |x\rangle \otimes |f(x)\rangle. \quad (7.14)$$

Atliekame Furjė transformaciją įvesties registrui:

$$(U_{FT} \otimes I)|\psi_1\rangle = \frac{1}{8} \sum_{x=0}^7 \sum_{y=0}^7 e^{i2\pi xy/8} |y\rangle \otimes |f(x)\rangle. \quad (7.15)$$

Šią dvigubą sumą apskaičiuosime pirmiausia sudėdami x indeksuotuosius narius ir laikant y fiksuotus, nes norime išnaudoti supaprastinimą, atsirandantį dėl $f(x)$ periodiškumo. Vadindami šioje stadioje bendrą būseną $|\psi_2\rangle$ randame:

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{8} \sum_{y=0}^7 |y\rangle \otimes (|f(0)\rangle + e^{\frac{i\pi y}{4}} |f(1)\rangle + e^{\frac{i\pi y}{2}} |f(2)\rangle + e^{\frac{i\pi y 3}{4}} |f(3)\rangle \\ &\quad + e^{i\pi y} |f(4)\rangle + e^{\frac{i\pi y 5}{4}} |f(5)\rangle + e^{\frac{i\pi y 3}{2}} |f(6)\rangle + e^{\frac{i\pi y 7}{4}} |f(7)\rangle). \end{aligned} \quad (7.16)$$

Toliau panaudojame $f(x)$ periodiškumą identifikuodami anksčiau minėtas vertes y' ir y'' ir su-grupuojame šiuos narius:

$$|\psi_2\rangle = \frac{1}{8} \sum_{y=0}^7 |y\rangle \otimes \left((1 + e^{\frac{i\pi y}{2}} + e^{i\pi y} + e^{\frac{i\pi y 3}{2}}) |y'\rangle + (e^{\frac{i\pi y}{4}} + e^{\frac{i\pi y 3}{4}} + e^{\frac{i\pi y 5}{4}} + e^{\frac{i\pi y 7}{4}}) |y''\rangle \right). \quad (7.17)$$

Būsenos $|\psi_2\rangle$ išraiška taip pat susiprastina dėl atsirandančių destruktyviųjų interferencijų sudendant skliausteliuose fazės narius. Pavyzdžiuui, jeigu $y = 1$, tada:

$$\left(1 + e^{\frac{i\pi}{2}} + e^{i\pi} + e^{\frac{i\pi}{2}}\right) |y'\rangle = (1 + i - 1 - i) |y'\rangle = 0. \quad (7.18)$$

$$\begin{aligned} \left(e^{\frac{i\pi}{4}} + e^{\frac{i\pi}{4}} + e^{\frac{i\pi}{4}} + e^{\frac{i\pi}{4}}\right) |y''\rangle &= \left(\frac{1}{\sqrt{2}} + \frac{i}{\sqrt{2}} - \frac{1}{\sqrt{2}} + \frac{i}{\sqrt{2}}\right. \\ &\quad \left.- \frac{1}{\sqrt{2}} - \frac{i}{\sqrt{2}} + \frac{1}{\sqrt{2}} - \frac{i}{\sqrt{2}}\right) |y''\rangle = 0. \end{aligned} \quad (7.19)$$

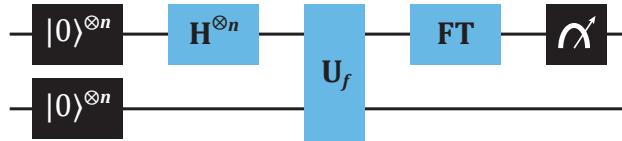
Taip patikrinę visas y vertes sumoje matome, kad konstruktyvioji interferencija atsiranda tik įvesties registro būsenose $|y\rangle = |0\rangle$ ir $|y\rangle = |4\rangle$. Visų kitų būsenų amplitudės susideda į 0. Galutinė abiejų registru būsena $|\psi_3\rangle$ yra:

$$|\psi_3\rangle = \frac{1}{2} |0\rangle \otimes (|y'\rangle + |y''\rangle) + \frac{1}{2} |4\rangle \otimes (|y'\rangle - |y''\rangle). \quad (7.20)$$

Kadangi $\langle y'|y''\rangle = \delta_{y'y''}$, pamatavę įvesties registrą rasime būsenas $|0\rangle$ arba $|4\rangle$ su $p(0) = p(4) = 0.5$ tikimybe. Būsena $|0\rangle$ nepriklauso nuo periodo, tad radus ją néra suteikiama informacijos ir teks kartoti žingsnius kol, šiuo atveju, bus rasta $|4\rangle$. Tai priseda prie algoritmo laiko sudėtingumo. Būsena $|4\rangle$ tiesiogiai atspindi $P = 2$ periodiškumą, nes pritaikius šį algoritmą n kubitų sistemai įvesties registre (žr. 7.4 pav.) bendrai išlieka tik šiu būsenų superpozicija:

$$|0\rangle, |1 \cdot 2^n/P\rangle, |2 \cdot 2^n/P\rangle, |3 \cdot 2^n/P\rangle, \dots, |(P-1) \cdot 2^n/P\rangle. \quad (7.21)$$

Darome prielaidą, kad $2^n/P$ yra sveikasis skaičius, tad jeigu randama būsena $|y\rangle$, periodas yra $P = k(2^n/y)$. Čia $k = 0, 1, 2, \dots, (P-1)$ yra sveikasis neneigiamasis skaičius, o 2^n nusako sistemos dimensiją ir yra žinomas skaičius. Taikydami šią formulę matome, kad minėta būsena $|4\rangle$ indikuoja $P = \frac{2^3}{4} = 2$ periodą. Bendrai, jeigu perteiksite sėryši taip $P/k = 2^n/y$, tada atlikus $2^n/y$ naryje abiejų skaičių padalijimą iš jų didžiausio bendrojo daliklio, gautas vardiklis bus periodas P . Taip pat galima parodyti, kad jeigu $2^n/P$ ir néra sveikasis skaičius, amplitudės vis tiek išlieka tik tų būsenų $|y\rangle$, kurios yra artimos $k2^n/P$ sveikajam skaičiui.



7.4 pav.: Funkcijos periodiškumą nustatanti loginė grandinė. Furjé transformacija n kubitų registrui glaučtai užrašyta kaip modulis FT

7.4 Kvartinis fazės nustatymo algoritmas

Šis algoritmas (angl. *quantum phase estimation*) pasitelkia kvartinę Furjé transformaciją ir aptinkamas kaip modulis kituose algoritmuose. Tarp jų yra kvartinis tiesinių lygčių sprendimo algoritmas, kurį pristatome kitame poskyryje. Fazės nustatymo algoritmo sudėtingumas yra nulemtas FT modulio, kuriam prireiks $O(n^2)$ elementariųjų operacijų. Fazės nustatymo algoritmo

užduotis yra rasti unitariojo operatoriaus U tikrines vertes. Unitariųjų operatorių tikrinės vertės λ_u turi bendrą formą $\lambda_u = e^{i2\pi\theta_u}$ ir tenkina lygtį:

$$U|u\rangle = e^{i2\pi\theta_u}|u\rangle. \quad (7.22)$$

Čia U yra unitarusis operatorius, kurio tikrinė vertė $\lambda_u = e^{i2\pi\theta_u}$ asocijuota su tikriniu vektoriumi $|u\rangle$ (darome priežiūrą, kad tikriniai vektoriai yra neišsigimę). Parametras, įvardijantis U skirtinges tikrines vertes λ_u , yra fazė θ_u (realusis skaičius) ir $0 \leq \theta_u < 1/2\pi$ dėl periodiškumo. Literatūroje galima rasti pavadintą θ_u iškomponuojant 2π , tada $\lambda_u = e^{i\theta_u}$ ir $\theta_u \rightarrow \theta_u/2\pi$.

Fazės nustatymo algoritmas leidžia, skaičiavimo išteklių atžvilgiu, efektyviai apskaičiuoti θ_u pagėdaujamu tikslumu. Šiuo metodu naudojamas fazės atatrankos triukas, kurį jau matėme Doičo ir Groverio algoritmuose. Imkime paprastą 2 kubitų būsenos pavyzdį:

$$|k_1\rangle \otimes |k_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |u\rangle. \quad (7.23)$$

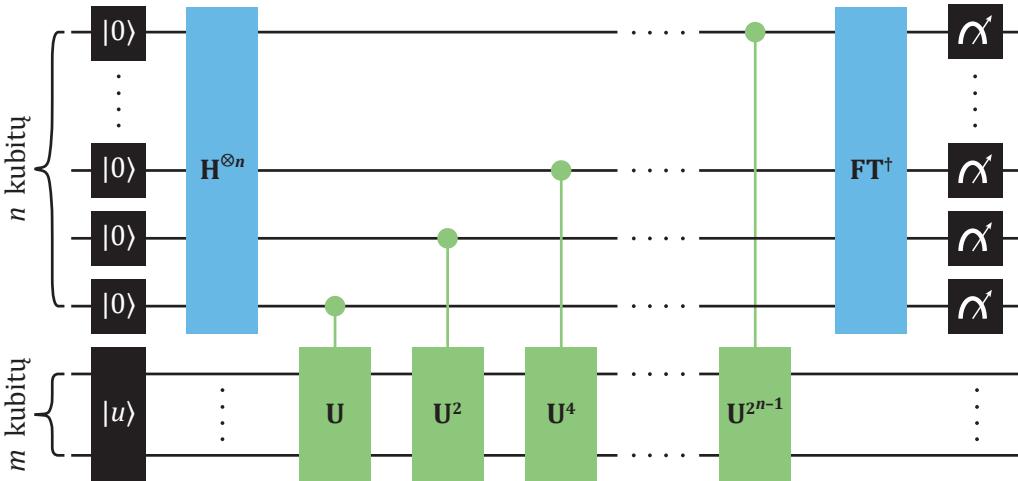
Čia kubitas k_2 yra U operatoriaus tikrinio vektoriaus būsenoje $|u\rangle$. Pritaikome šiemis dviem kubitams sąlyginį 2 kubitų operatorių cU , kuriame k_2 kubitas yra adresatinis:

$$cU|k_1\rangle \otimes |k_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |u\rangle + |1\rangle \otimes U|u\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + e^{i2\pi\theta_u}|1\rangle) \otimes |u\rangle. \quad (7.24)$$

Matome, kad amplitudė, nusakanti $|u\rangle$ būsenos tikrinę vertę $\lambda_u = e^{i2\pi\theta_u}$, yra perkeliama kubitiui k_1 . Fazės nustatymo algoritmas pasikliauja gebėjimu paruošti U operatoriaus tikrinį vektorių $|u\rangle$ ir atlikti sąlyginius cU^{2^n} vartus, kai U^{2^n} yra U pritaikytas 2^n kartų (pakeltas 2^n laipsniu). Pavyzdžiui, $U^{2^n}|u\rangle = UU|u\rangle = e^{i2\pi\theta_u}U|u\rangle = e^{i2\pi 2\theta_u}|u\rangle$. Taip teisdami matome:

$$U^{2^n}|u\rangle = e^{i2\pi 2^n \theta_u}|u\rangle. \quad (7.25)$$

Kvantinė grandinė, realizuojanti fazės nustatymą, yra parodyta 7.5 pav.



7.5 pav.: Bendra loginė grandinė, realizuojanti fazės nustatymo algoritmą

Pirmasis registras yra sudarytas iš n kubitių, čia skaičius n yra parenkamas pagal tai, kokį norima pasiekti fazės θ tikslumą bitais. Atkreipime dėmesį, kad naudojamas bazinių vektorių kodavimo

metodas (žr. 6.4 poskyri). Antrasis registras sudarytas iš m kubitų, reikalingų U operatoriaus tikriniam vektoriui $|u\rangle$ perteikti. Pritaikę $H^{\otimes n}$ bei cU^{2^n} vartų sekas, randame šią bendrą būseną:

$$\begin{aligned} |\psi\rangle &= \frac{1}{\sqrt{2^n}} (|0\rangle + e^{i2\pi 2^{n-1}\theta_u} |1\rangle) \otimes (|0\rangle + e^{i2\pi 2^{n-2}\theta_u} |1\rangle) \otimes \dots \\ &\quad \otimes (|0\rangle + e^{i2\pi 2^0\theta_u} |1\rangle) \otimes |u\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{i2\pi y\theta_u} |y\rangle \otimes |u\rangle. \end{aligned} \quad (7.26)$$

Antroje eilutėje atlikome supaprastinimą, išskleisdami visas n kubitų tensorines sandaugas. Tada, atpažindami n kubitų superpoziciją, pervađinome $|y\rangle$ bazinius vektorius dešimtainėje sistemoje. Antrasis m kubitų registras $|u\rangle$ būsenoje nebenturi įtakos likusiems algoritmo žingsniams. Atliekame atvirkštinę FT^\dagger pirmam n kubitų registrui:

$$(U_{FT}^\dagger \otimes I) \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{i2\pi y\theta_u} |y\rangle \otimes |u\rangle = \frac{1}{2^n} \sum_{l=0}^{2^n-1} \sum_{x=0}^{2^n-1} e^{\frac{-i2\pi l(x-2^n\theta_u)}{2^n}} |x\rangle \otimes |u\rangle. \quad (7.27)$$

Atlikus dvigubą indeksų l ir x sudėtį, pirmojo registro amplitudė turi didelę vertę tik ties $x \approx 2^n\theta_u$, tai yra $|x\rangle = |2^n\theta_u\rangle$ būsenai. Galutinę $|\psi\rangle$ galima užrašyti:

$$|\psi\rangle \cong |2^n\theta_u\rangle \otimes |u\rangle. \quad (7.28)$$

Todėl pirmojo registro būsenos matavimas su didele tikimybe grąžins $|2^n\theta_u\rangle$ būseną. Fazė yra randama $\theta_u = x/2^n$, čia x nusako pirmojo registro matavimo rezultatą.

Atvirkštinės FT^\dagger naudojimo reikšmę šiame algoritme galima pamatyti, jeigu fazę perteiksimė dvejetainės trupmenos forma, $\theta_u = 0.k_1k_2\dots k_n$ su n bitų tikslumu. Tada, pritaikius $H^{\otimes n}$ ir cU^{2^n} sekas, būsena $|\psi\rangle$ įgauna formą dvejetainėje sistemoje, kuri yra identiškai nusakoma FT atlikimu $U_{FT}|2^n k_1 k_2 \dots k_n\rangle$. Iškvietę atvirkštinę FT^\dagger , randame $U_{FT}^\dagger U_{FT}|2^n k_1 k_2 \dots k_n\rangle = |2^n k_1 k_2 \dots k_n\rangle$, ir tai vėl tiesiogiai įvardija θ_u dvejetainėje formoje.

Fazės nustatymo algoritme reikalinga paruošti pradinę registrų būseną $|0\rangle^{\otimes n} \otimes |u\rangle$. Tačiau itin tiksliai paruošti U operatoriaus tikrinį vektorių $|u\rangle$ gali būti keblu. Kaip tada veiks algoritmas? Atkreipiamė dėmesį, kad bendrą būseną $|\phi\rangle$ visada galima perteikti U operatoriaus tikriniais vektoriais $|u\rangle$:

$$|\phi\rangle = \sum_u c_u |u\rangle. \quad (7.29)$$

Čia amplitudės $c_u = \langle u|\phi\rangle$ nusako šių būsenų persiklojimą. Pritaikius fazės nustatymo algoritmą (FN), realizuojamą U_{FN} būsenai $|\phi\rangle$, rezultatas bus superpozicija:

$$U_{FN}|\phi\rangle \cong \sum_u c_u |2^n\theta_u\rangle \otimes |u\rangle. \quad (7.30)$$

Tikimybė rasti būseną $|u\rangle$ bei su ja supintą pirmojo registro būseną $|2^n\theta_u\rangle$, kuri nusako tikrinę vertę, yra $|c_u|^2$. Kitaip tariant, netikslus $|u\rangle$ paruošimas gali įvesti atsitiktinumo į rezultatą, dėl šios priežasties bus rasta kita U operatoriaus tikrinė vertė su tikimybe $|c_u|^2(1-\epsilon)$. Faktorius $(1-\epsilon)$ atsiranda dėl tikslumo, kuriuo pasirenkama nustatyti fazę. Todėl nebūtina idealiai tiksliai paruošti $|u\rangle$, pakanka pradinę būseną $|\phi\rangle$ padaryti kuo panašesnę į norimą $|u\rangle$ maksimizuojant jų persiklojimą $|\langle u|\phi\rangle|$.

Jeigu $2^n\theta_u$ ir nėra sveikasis skaičius, šis algoritmas vis tiek grąžina ieškomą fazę su didesne nei $p = 0.4$ tikimybe. Tikimybė rasti ieškomą rezultatą ir θ skaičiaus tikslumas gali būti padidinti pasitelkus daugiau n kubitų pirmajame registre. Galima formaliai parodyti, kad kubitų skaičius n grandinėje auga kaip $O(\log(1/\epsilon))$ siekant ϵ paklaidos bei reikalauja $O(1/\epsilon)$ sąlyginių loginių vartų cU .

7.5 Tiesinių lygčių sprendimas HHL algoritmu

Kvantinis HHL algoritmas (jo kūrėjų *Harrow-Hassidim-Lloyd* pavardžių trumpinys) leidžia spręsti tiesinių lygčių sistemas ir suteikia žymų paspartinimą prieš klasikinius algoritmus: **Gauso pašalinimą** (angl. *Gauss elimination*) ir **konjuguotojo gradiento metodą** (angl. *conjugate gradient method*). HHL algoritmo laiko vykdymo trukmė auga kaip $O(\kappa^2 \log 2^m)$ ir suteikia eksponentinį paspartinimą sistemas dydžio 2^m atžvilgiu prieš klasikinius algoritmus, kurie pasižymi $O(\kappa 2^m)$. Čia κ yra **matricos sąlygos skaičius** (angl. *condition number*), kuris nusako didžiausios ir mažiausios matricos A tikrinių verčių santykį, $\kappa = \lambda_{\max}/\lambda_{\min}$. Algoritmo stabilumas mažėja, kai $\lambda_{\min} \rightarrow 0$. Apžvelgsime HHL algoritmo bendruosius veikimo principus ir pateiksime jo realizaciją kvantinėje grandinėje.

Tiesinių lygčių sistema yra išreiškiama lygtimi:

$$A|x\rangle = |b\rangle. \quad (7.31)$$

Sprendžiant lygčių sistemą kvantiniu kompiuteriu, operatorius A tiesinėje algebroje yra ermitinė ($2^m \times 2^m$) dydžio matrica, $|x\rangle$ ir $|b\rangle$ yra 2^m dimensijų normuotieji vektoriai. Matrica A bei vektorius $|b\rangle$ yra nurodyti, algoritmo užduotis – rasti vektorių $|x\rangle$, tenkinantį šią lygčių sistemą. Bendrai ši sistema nusako 2^m skaičių lygčių ir 2^m skaičių nežinomujų, kurie yra $|x\rangle$ vektoriaus amplitudės. Sprendimas $|x\rangle$ randamas invertuojant A matricą, tai yra apskaičiuojant A^{-1} ($A^{-1}A = I$), nes:

$$|x\rangle = A^{-1}|b\rangle. \quad (7.32)$$

Reikalavimas, kad A būtų ermitinė matrica, gali būti sušvelminamas; mat jeigu A nėra ermitinė, galime apibūdinti naują matricą B , kuri yra ermitinė:

$$B = \begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}, \quad (7.33)$$

ir išspresti susijusią lygčių sistemą:

$$B \begin{pmatrix} 0 \\ |x\rangle \end{pmatrix} = \begin{pmatrix} |b\rangle \\ 0 \end{pmatrix}. \quad (7.34)$$

Toliau darome prielaidą, kad A yra ermitinis operatorius, perteikę spektrine dekompozicija:

$$A = \sum_{k=0}^{2^m-1} \lambda_k |a_k\rangle \langle a_k|. \quad (7.35)$$

Diados yra sudarytos iš operatoriaus A tikrinių vektorių $|a_k\rangle$, asocijuotų su tikrinėmis vertėmis λ_k . Tiesinėje algebroje A yra diagonalioji matrica, kurios ištirižainės yra tikrinės vertės λ_k . Tad jos atvirkštinė matrica A^{-1} diadų formoje yra tiesiog:

$$A^{-1} = \sum_{k=0}^{2^m-1} \frac{1}{\lambda_k} |a_k\rangle \langle a_k|. \quad (7.36)$$

Tai irgi diagonalioji matrica su įstrižainės vertėmis λ_k^{-1} . Vektorius $|b\rangle$ taip pat gali būti išreikštas A operatoriaus tikriniai vektoriais $|a_i\rangle$:

$$|b\rangle = \sum_{i=0}^{2^m-1} b_i |a_i\rangle. \quad (7.37)$$

Sudėjus šias išraiškas ir pritaikius bazinių vektorių ortogonalumą $\langle a_k | a_i \rangle = \delta_{k,i}$, sprendinys yra ieškomas tokia forma:

$$|x\rangle = \sum_{k=0}^{2^m-1} \lambda_k^{-1} b_k |a_k\rangle. \quad (7.38)$$

Tad ieškomasis vektorius $|x\rangle$ bus koduojamas kaip tikriniai vektorių $|a_k\rangle$ amplitudės (žr. 6.4 poskyri). Atkreipiame dėmesį, kad būsenų matavimas galiausiai yra atliekamas Pauli- Z tikriniai vektorių bazėje $\{|0\rangle, |1\rangle\}$, o ne $\{|a_k\rangle\}$. Tačiau galima matematiškai patikrinti, kad šiuo atveju vis tiek bus gautos teisingos amplitudės, jeigu kubitai nėra supintieji.

HHL algoritmo pagrindinis tikslas yra perteikti būseną $|b\rangle$ operatoriaus A tikriniai vektoriais $|a_k\rangle$ ir nustatyti jo tikrines vertes λ_k . Kadangi A yra ermitinis operatorius, jo eksponentė nusako unitarinį operatorių, ir todėl galime panaudoti kvantinį fazės nustatymo algoritmą rasti norimoms vertėms. **Hamiltoniano kodavimo metodas** (angl. *hamiltonian encoding*) leidžia simuliuoti matricą A unitariuoju operatoriumi ir išreikšti loginiai vartais:

$$U = e^{iAt}. \quad (7.39)$$

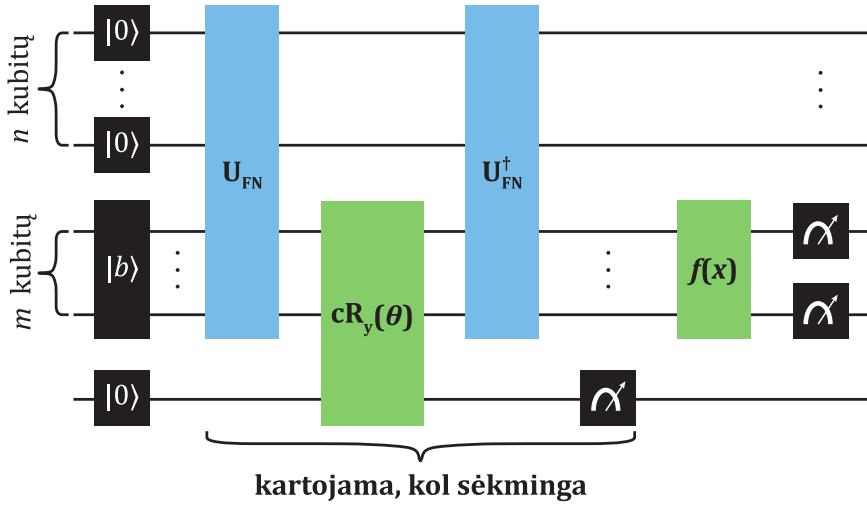
Kodavimo pavadinime žodis „hamiltonianas“ atsiranda dėl to, kad A yra ermitinis operatorius ir todėl iš princiopo nusako tam tikros kvantinės sistemos energijos lygius. Tad iš A suformuotas unitarusis operatorius U nusako šios sistemos laiko evoliuciją (žr. III skyrių), (7.39) lygybėje matome laiko kintamąjį t . Pavyzdžiui, mašininio mokymosi algoritmuose operatorių realizuojanti matrica A gali nusakyti mokymosi duomenis, o hamiltoniano kodavimas suteikia vieną būdą duomenis pateikti kvantiniam kompiuteriui. Ermitinio operatoriaus A matematinė išraiška nusakys, kaip galima realizuoti eksponentę e^{iAt} . Tai gali būti itin paprasta diagonalioji matrica, kurią galima išreikšti be aproksimacijų, arba gali prireikti kitų metodų. Vieną bendrojo tipo aproksimacijos metodą, vadinamą „troterizacija“, pristatome VIII skyriuje.

Tęsiant algoritmo apibūdinimą, išreiškus U spektrine dekompozicija:

$$e^{iAt} = \sum_{k=0}^{2^m-1} e^{i\lambda_k t} |a_k\rangle \langle a_k|. \quad (7.40)$$

Operatoriaus tikrinės vertės $e^{i\lambda_k t}$ yra asocijuotos su U tikriniai vektoriais $|a_k\rangle$, kurie čia taip pat yra ir A tikriniai vektoriai. Pažvelgus į fazės nustatymo algoritme naudojamus tikriniai verčių apibrėžimus ir lyginant su šiuo, $e^{i2\pi\theta_k} \rightarrow e^{i\lambda_k t}$, ermitinio operatoriaus A tikrinė vertė susieta su faze $\theta_k = \lambda_k t / 2\pi$. Tai yra visi reikalingi matematiniai įrankiai, naudojami HHL algoritme. Toliau apibūdinsime jo žingsnius, parodytus kvantinėje grandinėje 7.6 pav.

HHL naudoja tris kubitų registrus. Pirmųjų dvių registrų funkcijos čia yra iš esmės tokios pačios, kaip ir fazės nustatymo algoritme. Pirmasis registras turi n skaičių kubitų, tame užrašomos tikrinės vertės dvejetainė forma su atitinkamu bitų tikslumu. Antrasis registras turi m skaičių kubitų, tame išrašoma ir transformuojama būsena $|b\rangle$, kurią čia perteikėme tikriniai vektoriais



7.6 pav.: HHL algoritmo realizavimas loginėje grandinėje. Daroma prielaida, kad būsena $|b\rangle$ jau yra paruošta antrajame registre. Sąlyginiai $m + 1$ kubitu loginiai vartai $cR_y(\theta)$ yra kontroliuojami m kubitalis ir keičia adresato ancila kubito būseną. Kartu su parodytomis Furjė transformacijomis ir ancila kubito matavimu ši algoritmo dalis yra kartojama, kol ancila kubitas randamas $|1\rangle$ būsenoje

$|a_k\rangle$. Antrajame registre ir bus galiausiai užrašomas sprendinys $|x\rangle$. Trečasis registras yra 1 kubito ancila, kurios paskirtį iliustruojame toliau. Pradinė normuotoji trijų registrų būsena $|\phi\rangle$ po būsenos $|b\rangle$ paruošimo yra:

$$|\phi\rangle = \sum_k b_k |0\rangle \otimes |a_k\rangle \otimes |0\rangle. \quad (7.41)$$

Tolesniame žingsnyje pritaikome fazės nustatymo algoritmą tarp pirmų dviejų registrų, kurio visus žingsnius kompaktiškai žymime U_{FN} . Randame:

$$U_{FN}|\phi\rangle = \sum_k b_k |\tilde{\lambda}_k\rangle \otimes |a_k\rangle \otimes |0\rangle. \quad (7.42)$$

Čia fazė yra koduojama: $\tilde{\lambda}_k = 2^n \lambda_k t / 2\pi$. Tai leidžia tiesiogiai susieti $|\tilde{\lambda}_k\rangle$ būseną su ieškomosiomis tikrinėmis vertėmis λ_k . Parametras t dydis yra pasirenkamas pagal U realizavimo metodą, siekiant sumažinti U atlikimo kladas ir nepadaryti algoritmo ilgo loginių vartų atžvilgiu. Šioje stadijoje norima atlikti pirmojo registro transformaciją: $|\tilde{\lambda}_k\rangle \rightarrow C\lambda_k^{-1}|\tilde{\lambda}_k\rangle$; čia C yra konstanta, reikalinga užtikrinti būsenų normavimui, ir ji turėtų būti mažesnė nei mažiausia tikrinė vertė, $|C| < \lambda_{\min}$. Tam pasitelkiamas ancila kubitas, kuriam atliekami sąlyginiai loginiai vartai $cR_y(\theta)$, kontroliuojami pirmojo registro $|\tilde{\lambda}_k\rangle$ būsenomis, su atitinkamai pasirinktu θ :

$$cR_y(\theta)U_{FN}|\phi\rangle = \sum_k b_k |\tilde{\lambda}_k\rangle \otimes |a_k\rangle \otimes \left[\sqrt{1 - (C\lambda_k^{-1})^2} |0\rangle + C\lambda_k^{-1} |1\rangle \right]. \quad (7.43)$$

Matome, kad jeigu išmatuotume ancila kubitą, radus jo būseną esant $|1\rangle$ bendrai būsenai bus perteikta amplitudė $C\lambda_k^{-1}$. Atkreipiame dėmesį, kad pirmi du registrai yra supintieji, tad norint perteikti teisingas amplitudes antrajam registrui reikia panaikinti supynimą. Paskutiniame

žingsnyje pritaikome atvirkštinę fazės nustatymo rutiną, U_{FN}^\dagger , grąžindami pirmąjį registrą į pradinę būseną ir panaikindami supynimą tarp pirmų dviejų registrų. Ignoruodami potencialius netikslumus, kylančius iš U_{FN} ir U_{FN}^\dagger , randame:

$$U_{\text{FN}}^\dagger cR_y(\theta)U_{\text{FN}}|\phi\rangle = \sum_k b_k|0\rangle \otimes |a_k\rangle \otimes \left[\sqrt{1 - (C\lambda_k^{-1})^2}|0\rangle + C\lambda_k^{-1}|1\rangle \right]. \quad (7.44)$$

Šioje stadijoje atliekame ancila kubito matavimą. Radus jo būseną esant $|1\rangle$, sprendinys $|x\rangle$ matomas antrajame registre:

$$|0\rangle \otimes |x\rangle \otimes |1\rangle = \frac{1}{\sqrt{N}} \sum_k |0\rangle \otimes b_k \lambda_k^{-1} |a_k\rangle \otimes |1\rangle. \quad (7.45)$$

Su būsenos normavimo koeficientu N :

$$N = \sum_k |b_k \lambda_k^{-1}|^2. \quad (7.46)$$

Radus ancila kubitą $|0\rangle$ būsenoje tektų kartoti žingsnius iš naujo, tai prisideda prie algoritmo laiko sudėtingumo. Kadangi taikomas amplitudžių kodavimas, atsakymo $|x\rangle$ tiesiogiai nuskaityti nepavyks. Vis dėlto dažnai yra svarbiau sužinoti ne patį atsakymą, bet jo tam tikras savybes, vidutines vertes, momentus, santykinę būsenos orientaciją sprendinių erdvėje. Pasitelkdami Hadamardo testą (6.7.1 poskyri) galime apskaičiuoti sprendinio funkciją $f(x) = \langle x|M|x\rangle$, realiuojant ją atitinkamai suformuluotu kvantiniu operatoriumi M .

VIII skyrius

Kvantinių sistemų modeliavimas ir mašininis mokymasis

8.1 Dinaminių sistemų modeliavimas

Klasikinių bei kvantinių sistemų modeliavimas fizikoje yra pagristas diferencialinių lygčių sprendinių paieška. Jie apibūdina dinamiką laike bei erdvėje ir atspindi dėsnius, kuriais šios sistemos pagrįstos. Klasikinėje fizikoje dažnai aptinkamos Niutono, Maksvelo ir Einšteino diferencialinės lygtys. Jos apibūdina sistemos pozicijų, elektromagnetinių ir gravitacinių laukų, atitinkamai, konfigūracijas bei kitimą. Tik itin specifinėse situacijose sprendinius galima rasti ir užrašyti žinomomis funkcijomis, tad yra pasitelkiami kompiuteriniai skaičiavimai. Iprastai, kompiuteriniai skaičiavimai diskretizuojant diferencialinės lygties kintamuosius, tokius kaip laiko ir erdvės. Tuomet iteracinė procedūra pradinę sistemos būseną nuveda prie ieškomosios galutinės. Diskretizavimo žingsnių dydis pasirenkamas atsižvelgiant į toleruotinus paklaidos dydžius, kuriuos dažnai galima tiksliai įvertinti. Modeliuojamos sistemos dydis ir jos dinamikos tikslumas bus nulemti prieinamų skaičiavimo išteklių.

Pavienių ir sudėtinių kvantinių sistemų dinamiką apibūdina 3 skyriuje minėta Šriodingerio lygtis:

$$i\hbar \frac{d|\psi(t)\rangle}{dt} = H|\psi(t)\rangle. \quad (8.1)$$

Ši lygtis tinka kvantinėms sistemoms, kuriose relatyvistiniai efektai yra nereikšmingi arba gali būti aproksimuojami efektyviu hamiltonianu H . Tokio tipo sistemas čia ir aptarsime.

8.1.1 Aizingo modelis

Pirmiausia pradėkime nuo pastebėjimo, kad sudėtinių kvantinių sistemų modeliavimas, kurių posistemės būsenos aprašomas identiškai kubitams, natūraliai tinka kvantinės kompiuterijos taisykams. Tai iš prinčio atstoja gerai kontroliuojamą eksperimentą su šiomis sistemomis. Tarp tokių yra elektronas, protonas ir neutronas, turintys $1/2$ sūki (angl. $1/2$ spin). Šios dalelės magnetiniame lauke elgiasi panašiai kaip magnetiniai dipoliai. Blocho vektorius, orientuotas į ortogonaliasias $|0\rangle$ arba $|1\rangle$ būsenas, yra atitinkamai magnetinė „šiaurė“ arba „pietūs“. Vienas pavyzdys – periodiškai išsidėlioję atomai kristale, kuriame kiekvienoje atomo pozicijoje yra po vieną nesuporuotą elektroną (kubitą).

Aizingo modelio (angl. *Ising model*) hamiltonianas leidžia apibūdinti sistemos, sudarytos iš $1/2$ sukių, dinamiką. **Skersinio lauko Aizingo modelio** versija (angl. *transverse-field Ising model*) turi šią formą:

$$H = -j \sum_{i=1}^n Z_i \otimes Z_{i+1} - g \sum_{i=1}^n X_i. \quad (8.2)$$

Hamiltonianas apibūdina n kubitų, išrikiuotų eilėje. Nariai pirmoje suminėje dedamojoje nusako magnetines sąveikas tarp vienas šalia kito ($i, i + 1$) esančių kubitų išilgai z ašies, apibūdintas Pauli- Z operatoriai ir sąveikos stiprumu j . Pavyzdžiui, vienas toks narys, veikiantis tarp antrojo ir trečiojo kubito 4 kubitų sistemoje būtų $-jI \otimes Z \otimes Z \otimes I$. Antra dedamoji hamiltoniane apibūdina išorinio magnetinio lauko išilgai x ašies efektą kiekvienam iš n kubitų; veikdamas trečią kubitą 4 kubitų sistemoje jis būtų $-gI \otimes I \otimes X \otimes I$. Nepaisant paprastos išraiškos, kubitų elgesys Aizingo modelyje ir jo variacijose pasižymi fenomenu gausa. Pavyzdžiui, taip vadinamas laiko kristalas (angl. *time crystal*) gali būti realizuojamas kvantiniame procesoriuje naudojant Aizingo hamiltonianą su periodinėmis išorinio lauko perturbacijomis ir lokaliomis netvarkos dedamosiomis. Aizingo modelis 2 dimensijoje leidžia stebėti ferolektrinį tvarkos-netvarkos fazinį virsmą (angl. *ferroelectric order-disorder phase transition*).

Norėdami apskaičiuoti kubitų evoliuciją laike turime įvertinti unitariojo operatoriaus $U = e^{\frac{-iHt}{\hbar}}$ veiksmą (žr. III skyrių):

$$|\psi(t)\rangle = e^{\frac{-iHt}{\hbar}} |\psi(0)\rangle = U |\psi(0)\rangle. \quad (8.3)$$

Čia pradinė būsena $|\psi(0)\rangle$ laiku $t = 0$ yra bendrai n kubitų superpozicija. Toliau matysime vieną universalų metodą, kaip apskaičiuoti kubitų būsenos laiko evoliuciją kvantiniame kompiuteryje.

8.1.2 Trotterizacija

Daugeliu atvejų hamiltonianą H galime išskaidyti sumą:

$$H = \sum_{i=1}^k H_i. \quad (8.4)$$

Minėtame Aizingo modelyje hamiltonianas yra sudarytas iš dviejų narių: $H_1 = -j \sum_{i=1}^n Z_i \otimes Z_{i+1}$ ir $H_2 = -g \sum_{i=1}^n X_i$. Atkreipiame dėmesį, kad nors visi nariai H_1 ir H_2 sumose yra komutatyvūs, tačiau H_1 ir H_2 yra tarpusavyje nekomutatyvūs, $[H_1, H_2] \neq 0$. Todėl negalime šio unitariojo operatoriaus tiesiogiai pritaikyti nustatydami sistemos evoliuciją laike, nes:

$$e^{\frac{-iH_1 t}{\hbar}} e^{\frac{-iH_2 t}{\hbar}} \neq e^{\frac{-i(H_1 + H_2)t}{\hbar}}. \quad (8.5)$$

Vadinamoji **troterizacija** (angl. *Suzuki-Trotter approximation*) leidžia apeiti iškilusią kliūtį ir apytikriai realizuoti norimą operatorių. Ši aproksimacija diskretizuojant laiko intervalą t į s žingsnių, kuriame kiekvienas laiko intervalas trunka $\Delta t = t/s$. Unitarusis operatorius tampa:

$$U = e^{\frac{-i(H_1 + H_2)t}{\hbar}} \cong \left(e^{\frac{-iH_1 \Delta t}{\hbar}} e^{\frac{-iH_2 \Delta t}{\hbar}} \right)^s. \quad (8.6)$$

Naudodami pasirinktinai mažą žingsnį Δt ir atlikdami iteraciją s kartų, apskaičiuosime norimą sistemos laiko evoliuciją visame laiko intervale t . Viršuje parodyta pirmosios eilės U aproksimacija, joje įterpiamos klaidos yra ne didesnės nei žingsnio dydžio kvadratas $O(\Delta t^2)$. Trotterizacija

taip pat išsaugo laiko evoliucijos operatoriaus unitarumą ir todėl užtikrina, kad visame procese kvantinė būsena išlieka normuota. Galutinė sistemos būsena randama $|\psi(t)\rangle \cong |\psi(s\Delta t)\rangle$:

$$|\psi(s\Delta t)\rangle = e^{-\frac{iH_1\Delta t}{\hbar}} e^{-\frac{iH_2\Delta t}{\hbar}} \cdots e^{-\frac{iH_1\Delta t}{\hbar}} e^{-\frac{iH_2\Delta t}{\hbar}} |\psi(0)\rangle. \quad (8.7)$$

Troterizacija natūraliai tinka realizuoti ir hamiltonianus, kurie kinta laike. Tai gali nusakyti, pavyzdžiui, periodines sistemos perturbacijas ar kintančio stiprumo ir krypties išorines sąveikas. Tam vėlgi pasirenkamas diskretizuotas laiko žingsnis Δt , pagal kurį laiko evoliucija yra strobuojama, taip pateikiant efektyvią seką:

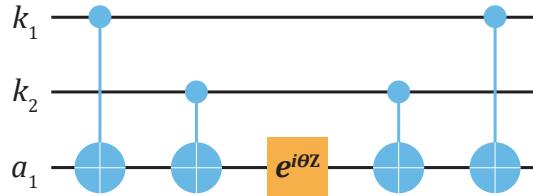
$$|\psi(s\Delta t)\rangle = U_s(\Delta t) \cdots U_3(\Delta t) U_2(\Delta t) U_1(\Delta t) |\psi(0)\rangle. \quad (8.8)$$

8.1.3 Aizingo modelio realizavimas kvantinėje grandinėje

Aizingo modelyje matome 1 kubito operatorių, turintį formą: $e^{-\frac{iH_2\Delta t}{\hbar}} = e^{-\frac{i\theta X}{2}} \equiv R_x(\theta)$. Tai yra pažistamas posūkio operatorius aplink x ašį, čia jis atlieka Blocho vektoriaus posūkį kampu $\theta = -\frac{2g\Delta t}{\hbar}$ kiekviename iš Δt dydžio iteracijos laiko žingsnių s . Taip pat turime realizuoti 2 kubitų loginius vartus, nusakytus operatoriumi $e^{-\frac{iH_1\Delta t}{\hbar}} = e^{i\theta Z_k \otimes Z_l}$. Čia $\theta = \frac{j\Delta t}{\hbar}$, o $Z_k \otimes Z_l$ veikia k ir l kubitus kiekviename žingsnyje. Dėl paprastumo praleidžiame identitetus, $\otimes I$, kurie veikia likusius kubitus. Tam galime panaudoti II skyriuje pateiktą tenzorinės operatorių sanudaugos funkcijos išraišką:

$$e^{i\theta Z_k \otimes Z_l} = \sum_{k,l} e^{i\theta \lambda_k \lambda_l} P_k \otimes P_l = \begin{bmatrix} e^{i\theta} & 0 & 0 & 0 \\ 0 & e^{-i\theta} & 0 & 0 \\ 0 & 0 & e^{-i\theta} & 0 \\ 0 & 0 & 0 & e^{i\theta} \end{bmatrix}. \quad (8.9)$$

(8.9) lygybės tenzorinėje sandaugoje $P_{k,l} \in (|0\rangle\langle 0|, |1\rangle\langle 1|)$ yra Pauli-Z projekciniai operatoriai, o tikrinių verčių sandauga yra $\lambda_k \lambda_l \in (1, -1)$. Atkreipiame dėmesį, kad lyginį lyginumą turinčioms 2 kubitų būsenoms ($|00\rangle$ ir $|11\rangle$) yra pritaikoma fazė $e^{i\theta}$, o štai nelyginio lyginumo būsenos ($|01\rangle$ ir $|10\rangle$) išgauna $e^{-i\theta}$. 8.1 pav. parodyta grandinė, realizuojanti $e^{i\theta Z_k \otimes Z_l}$ operatorių kubitams k_1 ir k_2 pasitelkiant ancilą a_1 .



8.1 pav.: Loginė grandinė, realizuojanti $e^{i\theta Z_1 \otimes Z_2}$ operatorių. Ancila kubitui pritaikomi R_z posūkio loginiai vartai

Pirma kvantinės grandinės užduotis yra atskirti būsenų lyginumą. Tam iškviečiami dveji cX vartai, supinančiai ancilą kubitą a_1 su k_1 ir k_2 kubitalis. Norėdami tai aiškiau pamatyti, imkime, kad k_1 ir k_2 kubitalai yra lygioje 2 kubitų skaičiuojamųjų bazinių vektorių superpozicijoje. Tad bendra pradinė būsena yra $|\psi\rangle = |k_1 k_2\rangle \otimes |a_1\rangle$:

$$|\psi\rangle = \frac{1}{2}(|00\rangle + |10\rangle + |01\rangle + |11\rangle) \otimes |0\rangle. \quad (8.10)$$

Pritaikius pirmus dvejus cX vartus su ancila adresatiniu kubitu, skaičiumi nurodant kontroluojantį k kubitą, randame:

$$cX_2cX_1|\psi\rangle = \frac{1}{2}(|00\rangle + |11\rangle) \otimes |0\rangle + \frac{1}{2}(|10\rangle + |01\rangle) \otimes |1\rangle. \quad (8.11)$$

Matome, kad tai supina ancilos kubito būseną $|0\rangle$ su lyginio lyginumo k_1 ir k_2 kubitų būsenomis, o $|1\rangle$ yra supinama su nelyginėmis. Atskyre lyginumo būsenas, kitame žingsnyje pritaikome grandinėje parodytus $R_z(\theta) = e^{i\theta Z}$ posūkio aplink z ašį loginius vartus ancila kubitui. Pagal 4 skyriuje apibūdintus posūkio vartus $R_z(\varphi) = e^{-i\varphi Z/2}$, kampus čia yra $\theta = -\varphi/2$. Dėl kvantinio supynimo $R_z(\varphi)$ efektyviai suteikia skirtinges fazes skirtingo lyginumo būsenoms:

$$\begin{aligned} & (I \otimes I \otimes e^{i\theta Z})cX_2cX_1|\psi\rangle \\ &= \frac{1}{2}(|00\rangle + |11\rangle) \otimes e^{i\theta}|0\rangle + \frac{1}{2}(|10\rangle + |01\rangle) \otimes e^{-i\theta}|1\rangle \\ &= \frac{1}{2}(e^{i\theta}|00\rangle + e^{i\theta}|11\rangle) \otimes |0\rangle + \frac{1}{2}(e^{-i\theta}|10\rangle + e^{-i\theta}|01\rangle) \otimes |1\rangle. \end{aligned} \quad (8.12)$$

Ancilos kubito būseną atstatome atgal su dviem cX vartais kartu panaikindami supynimą su k_1 ir k_2 kubitais. Gauname galutinę 2 kubitų būseną, veikiamą laiko evoliucijos operatoriumi $e^{\frac{-iH_1t}{\hbar}} = e^{i\theta Z_k \otimes Z_l}$:

$$cX_1cX_2(I \otimes I \otimes e^{i\theta Z})cX_2cX_1|\psi\rangle = \frac{1}{2}(e^{i\theta}|00\rangle + e^{-i\theta}|10\rangle + e^{-i\theta}|01\rangle + e^{i\theta}|11\rangle) \otimes |0\rangle. \quad (8.13)$$

Tad turime visus įrankius realizuoti Aizingo modeliui kvantiniame kompiuteryje.

8.2 Erdvinės Šriodingerio lygties sprendimo algoritmas

Banginė funkcija (angl. *wave function*), kaip ir būseną apibūdinantis vektorius $|\psi\rangle$, nusako viską, ką galima žinoti apie kvantinę sistemą. Erdvinė banginė funkcija $\psi(x)$ praktikoje leidžia analizuoti ir aprašyti erdvines kvantinės sistemos savybes, pavyzdžiui, nusakant jos poziciją, judėjimo kryptį ir greitį, kiek tai leidžia Haizenbergo neapibrėžtumo principas. Čia verta įsivaizduoti specifinę sistemą, pavyzdžiui, elektroną. Neprarasdami bendrumo imsieme, kad elektronas juda vienoje erdvinėje dimensijoje – tai artimai atspindi keletą realių situacijų, kuriose yra apriboti erdviniai laisvės laipsniai. Norėdami apskaičiuoti, kaip ši sistema kinta laike, turime išspręsti Šriodingerio lygtį.

8.2.1 Banginė funkcija

Pirmiausiai perteiksime elektrono kvantinę būseną, išreikštą vektoriumi $|\psi\rangle$, į erdvinį jos atvaizdavimą. Tam įvesime erdvinės pozicijos operatorių \hat{x} ir jo tikrinus vektorius $|x\rangle$, kurie tenkina lygtį:

$$\hat{x}|x'\rangle = x'|x'\rangle. \quad (8.14)$$

(8.14) lygybėje \hat{x} operatorius, veikiantis vieną iš tikrinių vektorių $|x'\rangle$, grąžina jo sandaugą su tikrine verte x' . Tikriniai vektoriai $|x'\rangle$ priskiriami kiekvienai erdvės pozicijai x' ir kadangi yra tolydūs, jų iš principo yra begalybė. Formaliai šios būsenos yra apibūdinamos pasitelkiant vadinamąją **suklastotą Hilberto erdvę** (angl. *rigged Hilbert space*). Skirtingų pozicijos tikrinių vektorių ortonormalumas išreiškiamas **Dirako delta funkcija** (angl. *Dirac delta function*),

$\langle x|x' \rangle = \delta(x - x')$. Kvintinės sistemos būsena $|\psi\rangle$, išreikšta erdviniaiš baziniaiš vektoriaiš $\{|x\rangle\}$ ir pasitelkiant pilnumo savybę (II skyrius), atrodo taip:

$$|\psi\rangle = \int_{-\infty}^{\infty} |x\rangle \langle x|\psi\rangle dx = \int_{-\infty}^{\infty} \psi(x) |x\rangle dx. \quad (8.15)$$

Pilnumo savybėje naudojame integralą, o ne sumą, kadangi x kinta tolydžiai. Integrale matome $\psi(x)|x\rangle$ narius, kuriuose pozicijos vektoriaus $|x\rangle$ kompleksinė amplitudė yra nusakoma $\psi(x) = \langle x|\psi\rangle$. Kvintinių būsenų pozicijos atvaizdavime amplitudžių $\psi(x)$ visuma yra vadinama banginė funkcija.

Laike besikeičiančios kvintinės sistemos banginė funkcija yra pasklidusi erdvėje, ir tai formaliai nusako jos pozicijos būsenų superpoziciją. Banginę funkciją galima apriboti išoriniaiš veiksniaiš, pavyzdžiui, elektroną – elektrinio lauko barjeru. Todėl tik tam tikrame erdvės intervale ji turės nenulinės vertes, $\psi(x) \neq 0$. Elektroną taip pat galima lokalizuoti atlikus jo pozicijos matavimą. Matuojant elektronas yra lokalizuojamas erdvės intervale, nusakytame matavimo įrenginio savybėmis, pavyzdžiui, jo erdvine skiriamaja geba. Tikimybė rasti dalelę nykstamai mažo dx dydžio erdvės intervale nusako banginės funkcijos $\psi(x)$ modulio kvadrato šioje pozicijoje ir intervalo dx sandauga:

$$|\langle x|\psi\rangle|^2 dx = \psi^*(x)\psi(x)dx = |\psi(x)|^2 dx. \quad (8.16)$$

Susumavę (integravę) tokius narius visoje erdvėje reikalaujame, kad tikimybė p susidėtu į 1, mat elektronas turi būti vis tiek rastas kažkur erdvėje:

$$\int_{-\infty}^{\infty} |\psi(x)|^2 dx = 1. \quad (8.17)$$

Praktiniuose taikymuose, banginė kvintinės sistemos funkcija $\psi(x)$ yra išplitusi tik tam tikroje erdvės dalyje ir nesitęsia iki begalybės. Todėl integraciją pakanka atlikti tik toje erdvės dalyje, kur $\psi(x)$ turi apčiuopiamą dydžio vertes.

8.2.2 Diskretizavimas

Norėdami kvantiniu kompiuteriu spręsti Šriodingerio lygtį erdvinėje išraiškoje, pirmiausiai atlikime erdvinių laisvės laipsnių diskretizavimą. Sakykime, kad mus domina L dydžio erdvės intervalas $-L/2 \leq x \leq L/2$. Šį intervalą diskretizuosime 2^n skaičiumi taškų su lygiais $\Delta x = L/2^n$ dydžio intervalais. Tai reiškia, kad šiuose 2^n erdvės taškuose bus įvertintos banginės funkcijos $\psi(x)$ reikšmės. Tokią diskretizacijos taškų visumą vadinsime **gardele** (angl. *lattice*).

Kiekvienas gardelės taškas yra indeksuojamas vienu iš n kubitų skaičiuojamujų bazinių vektorių $|x\rangle$. Norint išvengti x simbolų dubliavimo su erdvinių simboliais, dešimtainėje sistemoje naudojamus x simbolius pakeisime į v . Jeigu registras yra sudarytas iš n kubitų, tada turime 2^n diskretizacijos taškus, nusakytus 2^n bazinių vektoriaiš $\{|v\rangle\}$. Pavyzdžiui, 3 kubitų registras leidžia sukurti 8 taškų gardelę $\{|0\rangle, |1\rangle, |2\rangle, |3\rangle, |4\rangle, |5\rangle, |6\rangle, |7\rangle\}$.

Čia natūraliai pasitelkiame amplitudžių kodavimo metodą, kadangi banginė funkcija, kurios kitimą apskaičiuosime, yra bazinių vektorių $\{|v\rangle\}$ amplitudės. Atkreipiame dėmesį, kad dėl eksponentinio būsenų augimo d skaičiui taškų tereikia $\log_2(d)$ kubitu – tai yra itin efektyvus diskretizacijos būdas. Kvintinėms sistemoms yra papildomas privalumas, kadangi kiekviename gardelės taške kubito būsenos amplitudė yra natūraliai koduojama kompleksiniu skaičiumi.

Kiekvieną (diskretizuotą) poziciją x unikaliai susiejame su baziniais vektoriais $|v\rangle$ ir pervažindami ją x_v turime:

$$x_v = -\frac{L}{2} + v\Delta x. \quad (8.18)$$

Diskretizuotą kvantinę būseną $|\psi\rangle$ šioje erdvėje išreiškiame taip:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{v=0}^{2^n-1} \psi(x_v) |v\rangle. \quad (8.19)$$

Būsena $|\psi\rangle$ yra normuojama L -dydžio intervale faktoriumi \sqrt{N} :

$$N = \sum_{v=0}^{2^n-1} |\psi(x_v)|^2 \Delta x. \quad (8.20)$$

Siekdami supaprastinti simboliką, laikinai praleisime pozicijos x indeksaciją v simboliais. Tolesniame žingsnyje perteikiame Šriodingerio lygtimi apibūdinamą sistemos dinamiką į matematinę formą, kuri yra tinkama diskretizacijai. Naudojant $\psi(x, t) = \langle x | \psi(t) \rangle$:

$$i\hbar \frac{d\psi(x, t)}{dt} = H\psi(x, t). \quad (8.21)$$

Ši lygtis nusako $\psi(x, t)$ banginės funkcijos (amplitudžių) kitimą erdvėje ir laike. Hamiltonianas bendrai susideda iš **kinetinės energijos** (angl. *kinetic energy*) ir **potencinės energijos** (angl. *potential energy*) operatorių:

$$H = -\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + V(x) \equiv K + V. \quad (8.22)$$

Čia d^2/dx^2 atlieka banginės funkcijos antros eilės išvestinę, m yra kvantinės sistemos masė. Potencinės energijos funkcija V nusako sąveikas su išorinėmis sistemomis. Imsime, kad $V(x)$ priklauso tik nuo erdinės pozicijos x ir nekinta laike. Pradinės banginės funkcijos $\psi(x, 0)$ evoliucija po laiko intervalo t randama:

$$\psi(x, t) = e^{\frac{-i(K+V)t}{\hbar}} \psi(x, 0). \quad (8.23)$$

Atkreipiame dėmesį, kad $e^{\frac{-i(K+V)t}{\hbar}} \neq e^{\frac{-iKt}{\hbar}} e^{\frac{-iVt}{\hbar}}$. Šie išskaidyti unitarieji operatoriai bendroje situacijoje yra nekomutatyvūs. Tam vėl pasitelksime troterizaciją diskretizuodami laiko intervalą t į s skaičių žingsnių:

$$e^{\frac{-i(K+V)t}{\hbar}} \cong \left(e^{\frac{-iK\Delta t}{\hbar}} e^{\frac{-iV\Delta t}{\hbar}} \right)^s; \quad (8.24)$$

$$\psi(x, s\Delta t) = e^{\frac{-iK\Delta t}{\hbar}} e^{\frac{-iV\Delta t}{\hbar}} \dots e^{\frac{-iK\Delta t}{\hbar}} e^{\frac{-iV\Delta t}{\hbar}} \psi(x, 0). \quad (8.25)$$

Šioje stadioje diskretizavome erdvės ir laiko kintamuosius, taip pat perteikėme banginę funkciją gardelėje. Toliau parodysime, kaip apskaičiuoti troterizuoto laiko evoliucijos operatoriaus efektą banginei funkcijai.

Potencinės energijos V operatoriaus eksponentė $e^{\frac{-iV(x)\Delta t}{\hbar}}$, laiko žingsnyje Δt daugindama banginę funkciją $\psi(x, t)$, suteikia jai kiekvienoje diskretizuotos erdvės pozicijoje x_v fazę $\theta_v = \frac{V(x_v)\Delta t}{\hbar}$. Kadangi naudojame 2^n erdvės diskretizacijos taškų, nusakytų kubitų baziniais vektoriais, ši

operatorių galime išreikšti $(2^n \times 2^n)$ dydžio diagonaliaja matrica:

$$e^{\frac{-iV(x)\Delta t}{\hbar}} = \begin{bmatrix} e^{-i\theta_0} & 0 & \dots & 0 & 0 \\ 0 & e^{-i\theta_1} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & e^{-i\theta_{2^n-2}} & 0 \\ 0 & 0 & \dots & 0 & e^{-i\theta_{2^n-1}} \end{bmatrix}. \quad (8.26)$$

Kinetinės energijos operatoriaus eksponentės $e^{\frac{-iK\Delta t}{\hbar}}$ efektas banginei funkcijai lengviausiai apskaičiuojamas banginės funkcijos **judesio kieko atvaizdavime** (angl. *momentum space representation*). Pritaikius pozicijos x ir judesio kieko p **konjuguojamumą** (angl. *conjugation*), banginės funkcijos Furjė transformacija konvertuoja ją tarp šių atvaizdavimų:

$$U_{\text{FT}}\psi(x) = \psi(p); \quad (8.27)$$

$$\psi(p) = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \psi(x) e^{i2\pi x p / 2^n}. \quad (8.28)$$

Jeigu $\psi(x)$ yra normuotoji, tada ir $\psi(p)$ bus normuotoji dėl U_{FT} unitarumo. Judesio kieko operatorius pozicijos atvaizdavime yra $p = -\frac{\hbar}{i} \frac{d}{dx}$, tad judesio kieko atvaizdavime kinetinė energija yra tiesiog $K = \frac{p^2}{2m}$. Iš to išplaukia unitarinis operatorius $e^{\frac{-iK\Delta t}{\hbar}} = e^{\frac{-ip^2\Delta t}{2m\hbar}}$, kuris veikia $\psi(p)$ banginę funkciją:

$$e^{\frac{-iK\Delta t}{\hbar}} \psi(x) = U_{\text{FT}}^\dagger e^{\frac{-ip^2\Delta t}{2m\hbar}} U_{\text{FT}} \psi(x). \quad (8.29)$$

Atvirkštinė Furjė transformacija U_{FT}^\dagger grąžina banginę funkciją $\psi(p)$ atgal į pozicijos atvaizdavimą $\psi(x)$. Kinetinės energijos operatorius yra $(2^n \times 2^n)$ dydžio diagonalioji matrica šiame atvaizdavime. Jo efektas banginei funkcijai apskaičiuojamas analogiškai, kaip ir potencinės energijos, tačiau suteikiant fazę $\theta_j = \frac{p_j^2 \Delta t}{2m\hbar}$ kiekvienam judesio kieko diskretizacijos taške j ir turi kvadratinę p_j^2 priklausomybę.

Realizuojant šią dalį, judesio kieko banginę funkciją $\psi(p) = \langle p | \psi \rangle$ išreiskiame gardelėje. Jis sudarytas iš tų pačių 2^n skaičiuojamujų bazinių vektorių, gardelės taškus indeksuojame su $|j\rangle$:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{2^n-1} \psi(p_j) |j\rangle. \quad (8.30)$$

Normavimo faktorius vardiklyje N išlieka nepakiteš:

$$N = \sum_{v=0}^{2^n-1} |\psi(x_v)|^2 \Delta x = \sum_{j=0}^{2^n-1} |\psi(p_j)|^2 \Delta p. \quad (8.31)$$

Nustatę erdvinį intervalo dydį ir erdvinį atstumą tarp 2^n taškų, galime tiesiogiai susieti juos su judesio kieko gardele (arba atvirkščiai). Judesio kiekis bus apibrežtas $-\pi/\Delta x \leq p \leq \pi/\Delta x$ ir turės $\Delta p = 2\pi/L$ dydžio intervalus. Judesio kieko koordinatę p_j gardelėje yra:

$$p_j = -\pi/\Delta x + j\Delta p. \quad (8.32)$$

Judesio kieko gardelė yra paprastai centruojama apie $p_j = 0$.

Sudėjė viską kartu, banginės funkcijos $\psi(x, t)$ evoliuciją laike iki $t = s\Delta t$ randame iteraciniu algoritmu:

$$\psi(x, t) = U_{\text{FT}}^\dagger e^{\frac{-ip^2\Delta t}{2m\hbar}} U_{\text{FT}} e^{\frac{-iV(x)\Delta t}{\hbar}} \dots U_{\text{FT}}^\dagger e^{\frac{-ip^2\Delta t}{2m\hbar}} U_{\text{FT}} e^{\frac{-iV(x)\Delta t}{\hbar}} \psi(x, 0). \quad (8.33)$$

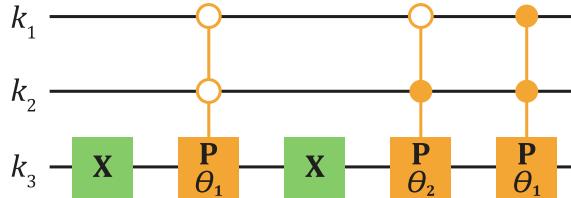
Erdvinės funkcijos diskretizacijos gardelė pasirenkama didesnė negu banginės funkcijos išsiplėtimas skaičiavimo metu, nes Furjė transformacija automatiškai padaro gardelę periodinę. Mat, jeigu banginė funkcija plėsdamas pasieks gardelės kraštą, ji vėl atsiras priešingame krašte ir plėsis į gardelės vidų, o ten gali įvesti klaidingas amplitudes ir padaryti banginę funkciją nebenormuotą. Erdvinio ir jūsės kieko gardelės žingsnių dydžiai parenkami norint pasiekti reikalaujamą erdinę skiriamąjā gebą ir atkurti norimus didžiausius sistemoje atsirandančius jūsės kiekius, atitinkamai.

8.2.3 Perteikimas kvantinėje grandinėje

Toliau pažiūrėkime, kaip kvantinėje grandinėje realizuoti $e^{\frac{-iV(x)\Delta t}{\hbar}}$ ir $e^{\frac{-ip^2\Delta t}{2m\hbar}}$ narius. Specifinė fizinė situacija diktuos, kokia yra potencinės energijos funkcijos $V(x)$ erdinė priklausomybė. Kaip paprastą pavyzdį imkime 3 kubitų gardelės diskretizaciją ir potencinės energijos $V(x)$ funkciją, kuri nusako a dydžio barjerus (energijos vienetai) kairiajame ir dešiniajame krašte ($x = 0$ ir $x = 7$), taip pat b dydžio barjerą $x = 3$ pozicijoje:

$$V(x) = \begin{cases} a, & x = 0 \text{ ir } 7 \\ b, & x = 3 \\ 0, & \text{likusiems } x \end{cases} \quad (8.34)$$

Kvantinė grandinė, perteikianti $\theta = \frac{V(x)\Delta t}{\hbar}$ fазes $\{|0\rangle, |3\rangle, |7\rangle\}$ būsenoms, parodyta 8.2 pav.



8.2 pav.: Loginė grandinė, atliekanti potencinės energijos funkcijos $V(x)$ su trimis barjerais veiksmą banginei funkcijai $\psi(x)$ viename laiko intervale

Matome dvigubai kontroliuojamus 3 kubitų fazės $ccP(\theta)$ loginius vartus, kuriuose „kontrolinės“ yra 1 kubito būsenos $|0\rangle$ (tušti apskritimai) arba $|1\rangle$ (užpildyti apskritimai). Šiuos aukštesnio lygio abstrakcijos loginius vartus, selektyviai suteikiančius fazę pasirinktais būsenais, galima realizuoti IV skyriuje parodytu Tofoli vartais pagrįstu metodu (žr. 4.7 poskyrį). Šitoks metodas yra bendro pobūdžio, tačiau geriausiai tinkta paprastoms potencinės energijos funkcijoms. Mat visoms skirtingoms 2^n būsenoms selektyviai parinkti reikėtų daug išteklių: papildomu $n - 1$ ancila kubitu; pritaikyti fazę vienam gardelės taškui reikalaujama $2(n - 1)$ Tofoli loginių vartų, o taškų skaičius yra eksponentinis 2^n .

Potencinės energijos funkcijos, pasižyminčios simetrijomis, gali būti efektyviai išreikštose loginiais vartais. Vienas pavyzdys yra harmoninio osciliatoriaus funkcija $V(x) = \gamma x^2$ (γ – realusis skaičius), turinti veidrodinę simetriją apie $x = 0$. Unitarinį operatorių $e^{\frac{-i\gamma x^2\Delta t}{\hbar}}$, veikiantį n kubitų registrą, galima realizuoti naudojant tik $O(n^2)$ kompleksiškumą nulemiančių 2 kubitų loginių vartų. Kadangi kinetinės energijos unitarasis operatorius $e^{\frac{-ip^2\Delta t}{2m\hbar}}$ taip pat turi identišką kvadratinę išraišką, pažiūrėsime jo perteikimą loginiais vartais:

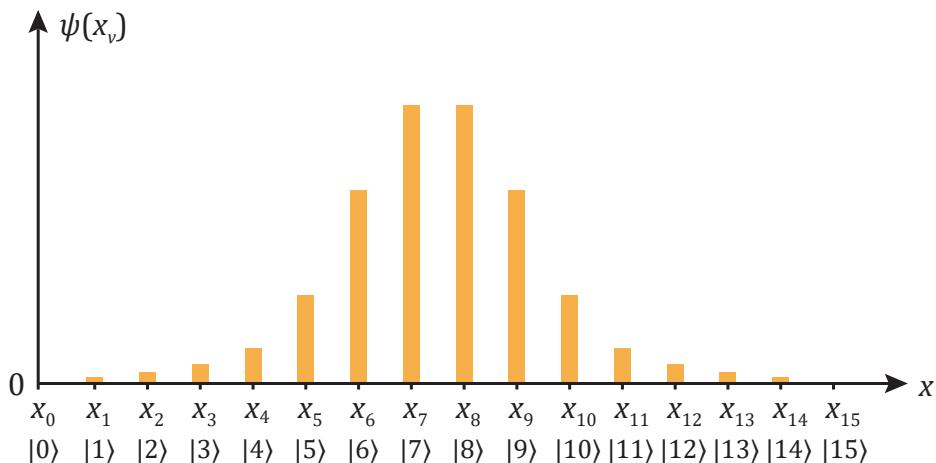
$$e^{\frac{-ip^2\Delta t}{2m\hbar}} = e^{\frac{i\theta}{2^{2n-3}} (1 + \sum_{k=1}^n 2^{n-k} Z_k)^2} \quad (8.35)$$

Čia θ nusako efektinę fazę, suteikiamą kiekvienam laiko žingsnyje Δt , Z_k yra Pauli- Z loginiai vartai, veikiantys k kubitą. Suma atliekama iki algoritme naudojamų n kubitų skaičiaus. Norėdami tai iliustruoti, pateikiame 3 kubitų registro pavyzdį:

$$e^{\frac{-ip^2\Delta t}{2m\hbar}} = e^{i\theta(Z_1 + \frac{1}{2}Z_2 + \frac{1}{4}Z_3 + 2Z_1 \otimes Z_2 + Z_1 \otimes Z_3 + \frac{1}{2}Z_2 \otimes Z_3)}. \quad (8.36)$$

Jau žinome, kaip kvantinėje grandinėje loginiai vartais perteikti visus čia matomus narius.

Šriodingerio lygties sprendime turime paruošti pradinę banginę funkciją $\psi(x, 0)$. Pageidautina, kad šis žingsnis nereikalautų itin didelio loginių operacijų skaičiaus. Vėlgi, funkcijos, pasižyminčios simetrijomis, išprastai gali būti efektyviai koduojamos. Pavyzdžiui, **Gauso funkcija** (angl. *Gaussian function*) yra išreiškiama polinominiu skaičiumi loginių vartų $O(\text{pol}(n + 1/\Delta))$; čia Δ nusako erdvinę skiriamąją gebą. Diskretizuota Gauso funkcija $\psi(x_v)$, koduojama 4 kubitų registro 16-oje jų amplitudžių, yra pateikta 8.3 pav. Kitas galimas būdas – pasitelkti banginės funkcijos transformaciją iš efektyviai koduojamos funkcijos. Pavyzdžiui, gan komplikuotą **Beselio** J_0 (angl. *0th-order Bessel function of the 1st kind*) funkciją galima sukurti atliekant Furjė transformaciją funkcijai, kurios forma turi lygias amplitudes tam tikru spinduliu ir yra nulinė visur kitur. Ši Beselio funkcija dažnai aptinkama ir praktikoje, nes ji nusako apvalios apertūros sukurtą tolimojo lauko difrakcijos funkciją.



8.3 pav.: Gauso formos diskretizuota banginė funkcija, perteikta 16-os taškų gardelėje

Erdvinės Šriodingerio lygties algoritmo pabaigoje turime banginę funkciją $\psi(x, t)$ arba, ekvivalentiškai, jos jūdesio kiekiu atvaizdavimą $\psi(p, t)$. Kadangi taikome amplitudžių kodavimo metodą perteikti banginei funkcijai, tiesiogiai visos banginės funkcijos vienu matavimu sužinoti neįmanoma. Pavyzdžiui, galime pasirinkti atlikti kiekvieno kubito standartinį Pauli- Z matavimą, formaliai toks n kubitų matavimas nusakomas apskaičiuojant $\langle \psi | Z^{\otimes n} | \psi \rangle$. Dešimtainėje sistemoje rasime vieną iš galimų pozicijos būseną $|x\rangle$ su tikimybe $p(x) = \psi^*(x)\psi(x)dx = |\psi(x)|^2dx$. Pakartojė algoritmą daug kartų ir atlikę tokį matavimą, apytikriai rasime visą **tikimybių tankio funkciją** $|\psi(x)|^2$ (angl. *probability density function*), kuri nusako tikimybes rasti kvantinę sistemą kiekvienam erdvės taške x .

Vietoj šio matavimo, pasitelkiant Hadamardo testą (žr. 6.7.1 poskyri), galima apskaičiuoti globalius banginės funkcijos parametrus, $\langle \psi | M | \psi \rangle$; čia M yra kvantinis operatorius. Pavyzdžiui, M

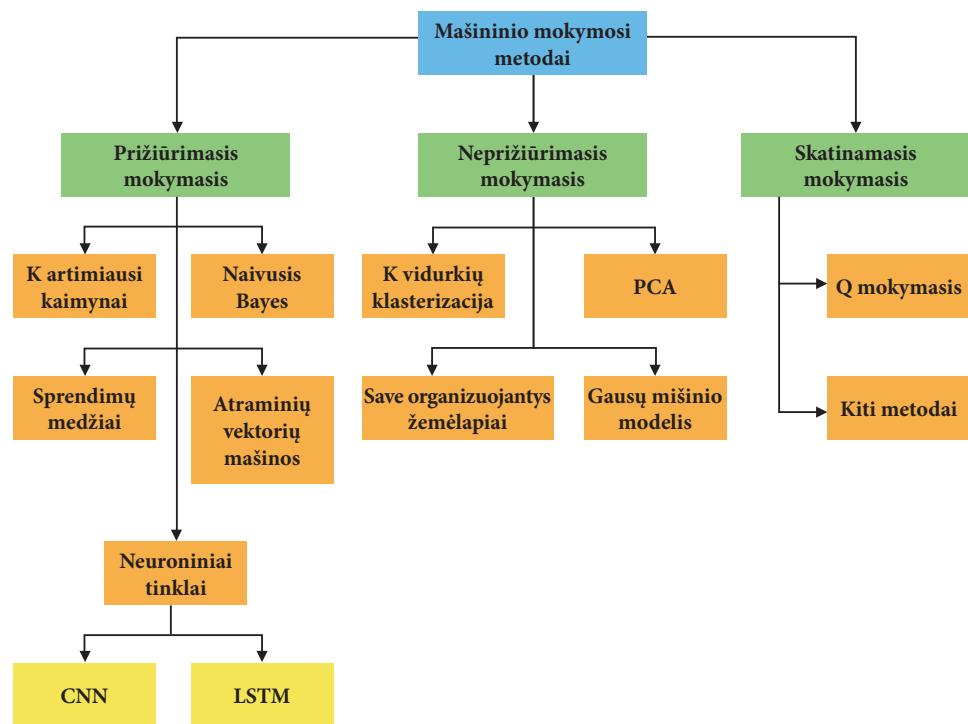
gali būti hamiltonianas arba judesio kieko momento operatorius. Tai suteiks sistemos, esančios būsenoje $\psi(x)$, vidutinę energiją bei judesio kieko momentą, atitinkamai. Globalių parametru nustatymui gali pakakti kartoti algoritmą vos keletą kartų.

8.3 Mašininis mokymasis

Mašininis mokymasis (angl. *machine learning*, ML) yra tyrimų sritis, kurianti metodus, išgaliinančius vis geriau atlikti nurodytas užduotis panaudojant duomenis. Mašininio mokymosi algoritmai, remiantis duomenų pavyzdžiais, sukuria prognozę atliekantį ar sprendimus priimantį modelį, tų užduočių neprogramuojant. Mašininis mokymasis yra plačiai taikomas įvairiose srityse, įskaitant kalbos atpažinimą, kompiuterinę regą, mašininių vertimą.

8.3.1 Klasikinis mašininis mokymasis

Klasikinio mašininio mokymosi metodai gali būti sugrupuoti į tris sritis: **prižiūrimasis mokymasis** (angl. *supervised learning*), **neprižiūrimasis mokymasis** (angl. *unsupervised learning*) ir **skatinamasis mokymasis** (angl. *reinforcement learning*). Iš visų mašininio mokymosi metodų prižiūrimojo mokymosi algoritmai yra labiausiai ištobulinti ir dažniausiai taikomi praktikoje. Tačiau prižiūrimasis mokymasis turi tą trūkumą, kad jam reikia sužymėtų duomenų, kuriuos gali būti brangu ar sunku gauti. Kai kurie dažniausiai taikomi mašininio mokymosi metodai pavaizduoti 8.4 pav.



8.4 pav.: Klasikinio mašininio mokymosi metodai

Prižiūrimojo mokymosi atveju modelio kūrimui panaudojami duomenys, kuriuose kiekvienas pa-

vyzdys yra pateikiamas kartu su norima išvestimi. Optimizuojant tikslą funkciją, kuri nusako, kiek modelio išvestis yra arti norimos, prižiūrimojo mokymosi algoritmas išmoksta prognozuo- ti išvestį, susijusią su nauja išvestimi. Yra sukurta daug prižiūrimojo mokymosi algoritmų, kai kuriuos jų paminėsime toliau.

Sprendimų medis (angl. *decision tree*) ir su juo susijęs **atsitiktinis miškas** (angl. *random forest*) yra modelis, pagrįstas srauto diagramomis, kuriose kiekvienas mazgas atitinka duomenų atributo testą, o šakos vaizduoja testo rezultatus. Sprendimų medžių parametrai yra norimas medžio gylis ir mazgų skaičius. Šis modelis nereikalauja išankstinių žinių apie duomenis ir yra atsparus labai nukrypusiems duomenų išrašams ar triukšmui žymėse. **Atraminiai vektorių mašinos** (angl. *support vector machines*, SVM) naudoja treniravimo duomenis surasti hiperplokštu- mai, kuri atskiria dvi duomenų klasses taip, kad atstumas iki kraštinių skirtingų klasų duomenis vaizduojančių taškų būtų kuo didesnis. Šie taškai yra vadinami atraminiais vektoriais ir nusako galutinį modelį. Kai gera atskyrimo plokštuma negali būti rasta, dažniausiai yra pritaikomi branduolio metodai, projektuojantys duomenis į aukštesnės dimensijos erdvę, kur skirtingos duomenų klasės tampa tiesiškai atskiriamos. Tinkamas branduolio parametru parinkimas yra svarbus geram modelio veikimui; šių parametrų palaikoma apsunkina metodo pritaikymą. Atraminiai vektorių mašinos pasižymi didele sparta, kai duomenų nedaug, tačiau skaičiavimo ir atminties išteklių poreikis sparčiai auga didėjant duomenų apimčiai.

Dabartiniu metu plačiai taikomas dirbtinis neuroninis tinklas yra modelis, sudarytas iš tarpusvyje sujungtų mazgų, vadinamų neuronais. Kiekvienas neuronas susumuoją informaciją iš kitų neuronų ir duoda išvestį, priklausomą nuo neurono netiesinės aktyvacijos funkcijos. Ryšių tarp skirtingų neuronų stiprumą nusako adaptyvūs svoriai. Neuroninio tinklo mokymo metu tinklo svoriai yra keičiami tol, kol tinklo išvestis pasidaro beveik lygi norimai. Pagal neuronų sujungimo pobūdį, dar vadinamą tinklo architektūra, neuroniniai tinklai skirstomi į tipus. **Konvoliucinis neuroninis tinklas** (angl. *convolutional neural network*, CNN) naudoja konvoliucijos operacijas su filtru rinkiniu, užuot pilnai sujungus neuronų sluoksnius. Tokie neuroniniai tinklai naudoja- mi erdviniams duomenims apdoroti, nes konvoliucijos operacijos išlaiko invariantiškumą duome- nių erdvinio poslinkio atžvilgiu. Kitas neuroninių tinklų tipas yra **rekurentiniai neuroniniai tinklai** (angl. *recurrent neural network*), skirti nuosekliems duomenims apdoroti. Rekurentiniai neuroniniai tinklai naudoja grįztamąsias jungtis tarp neuronų sluoksnio ir prieš jį einančių sluoksnių. Mokant paprastos architektūros rekurentinius neuroninius tinklus iškyla gestančių ar sprogstančių gradientų problemos, kurios apsunkina mokymą. Tam išvengti yra pasiūlytas specialus rekurentinių tinklų tipas **LSTM** (angl. *long short-term memory*), kuris į modelį įveda sklendžių rinkinį. Taigi sujungimas tarp skirtingų neuronų sluoksnių, jungčių svorių atnaujinimo procesas bei taikomas aktyvacijos funkcijos yra svarbiausi neuroninio tinklo parametrai. Neu- roniniai tinklai pasižymi daugeliu lokalų minimumų, todėl gali pateikti klaudingus rezultatus išvendant kitokius duomenis negu treniravimo metu.

Neprižiūrimojo mokymosi atveju yra pateikiami nesužymėti duomenys. Kuriamo modelio tikslas – aptikti struktūrą duomenyse, pavyzdžiui, juos sugrupuoti. Dažniausiai neprižiūrimojo mokymosi pritaikymai yra duomenų klasterizavimas, matmenų sumažinimas bei anomalijų aptikimas. Klasterizavimo tikslas – sugrupuoti duomenis. Tarp populiariausią klasterizavimo metodų yra **k vidurkių klasterizacija** (angl. *k-means clustering*) ir **save organizuojantys žemėlapiai** (angl. *self-organizing maps*, SOM). k vidurkių klasterizavimo metu duomenys dalijami į klas- terius taip, kad kiekvienas duomenų taškas priklauso klasteriui su artimiausiu taškų vidurkiu, klasterio viduje taškų dispersija minimizuojama. Klasteriams surasti dažniausiai naudojamas iteratyvusis algoritmas: sukuriama k klasterių, kiekvienas taškas susiejamas su artimiausiu vi- durkiu, tada kiekvieno iš naujų k klasterių centrali tampa naujais vidurkiais. Saveorganizuojančių

žemėlapių metodu duomenys yra pateikiami neuroniniams tinklui, o jis sukuria duomenų erdvės mažos dimensijos vaizdą. Matmenų sumažinimo metodai iš didelės dimensijos duomenų sukuria mažos dimensijos modelius. Pavyzdžiui, **pagrindinių komponentų analizė** (angl. *principal component analysis*, PCA) sukuria naujas duomenų požymių kombinacijas. Kombinacijos, turinčios didžiausią dispersiją, yra paliekamos, o visos kitos pašalinamos, taip sumažinant dimensiją.

Skatinamojo mokymosi tikslas – programiniams agentui, esančiam nurodytoje aplinkoje, išmokti pasirinkti veiksmus, maksimizuojančius gautą atlygi. Skatinamojo mokymosi modelis yra sudarytas iš aplinkos būsenų, galimų agento veiksmų, perėjimų tarp aplinkos būsenų taisyklių, atlygių už perėjimus tarp būsenų ir stebėjimo taisyklių. Agentas sąveikauja su aplinka pasirinkdamas veiksmus. Aplinka keičiasi atsakydama į tuos veiksmus, ir agentas gauna skaitinį atlygi. Skatinamajame mokymame agentas siekia maksimizuoti atlygi bėgant laikui. Mokymasis gali būti sukonzentruotas viename agente, arba paskirstytas keliuose. Skatinamasis mokymasis yra naudingas kontrolės uždaviniuose, kai negalima pateikti išreikštų taisyklių, o žinoma tik atlygio funkcija. Yra daug skatinamojo mokymosi algoritmų. Vienas iš populiarų metodų yra Q mokymasis. Jame algoritmas skaičiuoja tikėtiną atlygi (Q) veiksmo, atlikto esant nurodytai aplinkos būsenai, nepriklausomai nuo taikomos veiksmų strategijos. Kitas veiksmas yra pasirenkamas remiantis Q verte. Gautos atlygis yra naudojamas Q atnaujinimui imant senos vertės bei naujos informacijos pasvertą vidurkį.

8.3.2 Kvantinis mašininis mokymasis

Kvantinis mašininis mokymasis panaudoja kvantinį irenginį mašininio mokymosi uždaviniamams išspręsti su didesniu greičiu ar didesniu tikslumu, negu leidžia klasikiniai mašininio mokymosi metodai. Yra pasiūlyta įvairių klasikinio mašininio mokymosi kvantinių analogų, kurie paspartina klasterizavimą ar atraminių vektorių mašinas. Galimas dar platesnis apibrėžimas, kai kvantinio mašininio mokymosi algoritmai naudoja kvantinį irenginį klasifikuoti kvantinėms būsenoms, o ne klasikiniams duomenims. Pavyzdžiui, kvantinė pagrindinių komponentų analizė suranda tikrinus vektorius, atitinkančius didžiausias tikrines vertes. Nemažai mašininio mokymosi algoritmų taiko tiesinių lygčių sistemų sprendimą. Kadangi kvantinis HHL algoritmas potencialiai gali greičiau išspręsti tiesinių lygčių sistemą negu klasikiniai algoritmai, kvantinis kompiuteris gali būti panaudojamas mašininiam mokymuisi spartinti. HHL algoritmą naudoja kvantinis k vidurkių metodas bei kvantinės atraminių vektorių mašinos.

Klasikiniai neuroniniai tinklai turi netiesines aktyvacijos funkcijas, o štai kvantinių sistemų evo liucija, kaip žinome, yra aprašoma tiesinėmis lygtimis. Kyla klausimas, kaip padaryti kvantinį klasikinio neuroninio tinklo analogą? Vienas iš sprendimo būdų – naudoti hibridinius (iš dalies kvantinius, iš dalies klasikinius) algoritmus. Juose netiesiškumą į evoliuciją įveda kvantinės sistemos matavimas ir su juo susijęs superpozicijos būsenos suirimas.

Nemažai kvantinio mašininio mokymosi metodų naudoja hibridinius algoritmus: parametrizuotos kvantinės grandinės yra treniruojamos taikant klasikinius optimizavimo metodus. Tačiau čia iškyla problema: inicializuojant parametrus visiškai atsitiktinai, dėl eksponentiškai didelės būsenų erdvės gradientas parametru atžvilgiu dažniausiai yra eksponentiškai mažas didėjant kubitų skaičiui. Šis reiškinys, pavadintas **nederlingų plynaukščių** (angl. *barren plateaus*) buvimu, apsunkina kvantinių grandinių treniravimą. Toks nykstamai mažų gradientų buvimas taip pat gali atsirasti dėl triukšmo ar dėl per didelio supynimo tarp kubitų kvantinėje grandinėje. Kvantinis supynimas informaciją patalpina nelokaliai, koreliacijose tarp kubitų. Matuojant tik išvesties kubitus, dalis informacijos prarandama.

Visais atvejais, kai kvantinio mašininio mokymosi algoritmas apdoroja klasikinius duomenis, iš

pradžių reikia duomenis užkoduoti į kvantinę būseną. Dažniausiai taikomi kodavimo būdai yra bitų kodavimas ir amplitudžių kodavimas (žr. 6.4 poskyrį). Bitų kodavime l -tasis įrašas yra N bitų seka $b^{(l)} = \{b_1, b_2, \dots, b_N\}$, $b_i \in \{0, 1\}$. Operatorius O , dar vadinamas kvantiniu orakulu arba kvantine atmintimi, yra naudojamas iškvieсти l -tajį įrašą iš duomenų bazės kvantinėje būsenoje:

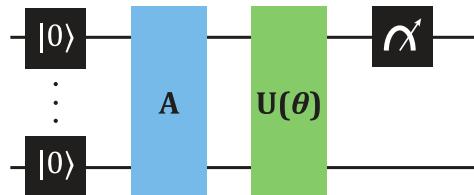
$$O|l\rangle \otimes |0\rangle = |l\rangle \otimes |b^{(l)}\rangle. \quad (8.37)$$

Toks vaizdavimas taikomas kvantinėse atraminių vektorių mašinose ir artimiausiu kaimynu klasifikatoriuje. Bitų kodavimo privalumas yra tas, kad jis pateikia duomenis tuo pačiu pavidalu, kaip ir atitinkamam klasikiniam mašininiu mokymosi algoritmu. Tačiau trūkumas – didelio kubitų skaičiaus poreikis, jeigu bitų skaičius N didelis. Kitas kodavimo būdas – tai amplitudės kodavimas, kai duomenys pavaizduojami kvantinės bazinių vektorių $|i\rangle$ superpozicijos amplitudėse x_i .

8.3.3 Kvantinėmis grandinėmis paremtas klasifikatorius

Kaip pavyzdį panagrinėkime vieną iš kvantinio mašininio mokymosi algoritmų – kvantinėmis grandinėmis paremtą klasifikatoriu. Tai prižiūrimojo mokymosi algoritmas, kuriame yra pateikiami treniravimo duomenys kartu su teisingomis žymomis. Naudojant treniravimo duomenis klasifikatorius apmokomas priskirti žymę dar nematytiems duomenims. Šis hibridinis klasikinis-kvantinis algoritmas, pristatytas Shuld, Wiebe ir bendraautorų iš *Microsoft* nereikalauja didelio kubitų skaičiaus bei gilių grandinių ir todėl yra tinkamas ankstyvosios raidos kvantiniam procesoriams. Algoritme yra panaudojamas duomenų amplitudžių kodavimas (žr. 6.4 poskyrį), tad $N = 2^n$ dydžio duomenų bazė perteikiama tik su n kubitų. Darant prielaidą, kad duomenis galima efektyviai užrašyti į kvantinę registro būseną, kvantinis paraleлизmas leidžia sparčią šios būsenos transformaciją ir rezultatų apskaičiavimą. Be to, dėl unitariųjų būsenų transformacijų, tokis klasifikatorius nestiprino triukšmo, esančio duomenyse ir jų žymose.

Kaip ir dauguma prižiūrimųjų mašininio mokymosi algoritmų, šis hibridinis algoritmas išmoko modelį, formaliai perteikiama funkcija $f(x, \theta) = y$, pateikiant duomenis x ir pažymint juos y . Binariosios klasifikacijos atveju duomenys priklauso grupei a arba $b, y \in a, b$. Tam yra optimizuojamas modelio parametru rinkinys θ . Gerai apmokytas modelis $f(x, \theta)$ turėtų gebėti teisingai priskirti nematytius duomenis x grupei $y \in a, b$. Kvantinėmis grandinėmis paremtas klasifikatorius yra pavaizduotas 8.5 pav.

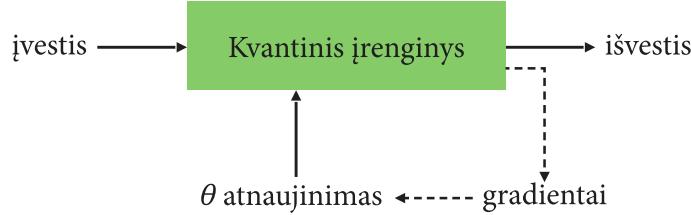


8.5 pav.: Kvantinėmis grandinėmis paremtas klasifikatorius. Loginių operacijų grandinė A atlieka pradinės būsenos paruošimą; ji yra toliau apdorojama modelio $U(\theta)$ grandinėje optimizuojant klasifikatoriu. Galiausiai, atliekama binarinė klasifikacija išmatuojant pirmojo kubito būseną

Kvantinė grandinė yra parametrizuota parametru rinkiniu θ . Pradinė n kubito būsena $|0\rangle^{\otimes n}$ naujodama koduoti įvesčiai x pasitelkiant būsenos paruošimo unitarųjį operatorių A . Operatorius A gali būti ir minėta orakulo funkcija, iškviečianti kvantinėje atmintyje laikomą duomenų bazę įvesti $|\varphi(x)\rangle$. Šios būsenos transformacija yra aprašoma unitariuoju operatoriumi $U(\theta)$. Kvantine grandinės parametrai θ optimizuojami taip, kad klasifikavimo rezultatas atitinktų treniravimo

duomenyse pateiktą teisingą žymę. Duomenų klasifikavimo rezultatas $f(x, \theta)$ yra nuskaitomas iš būsenos $U(\theta)|\varphi(x)\rangle$ atliekant kubitų matavimą. Jeigu klasifikacija yra binarioji, pakanka išmatuoti tik vieną kubitą, pavyzdžiui, grandinėje parodytą pirmąjį, kurio būsena $|1\rangle$ indikuoja teisingą pateiktą duomenų klasifikavimą. Ši būsena bus randama su tam tikra tikimybe, kurią norima modeliu maksimizuoti. Visgi, norint šią statistiką pamatyti, visą grandinę reikia atlikti keletą kartų.

Grandinės parametrams θ optimizuoti yra naudojamas klasikinis gradientinio nusileidimo algoritmas. Šiam algoritmui reikalingos unitariojo operatoriaus išvestinės $\partial_\theta U(\theta)$ parametru θ atžvilgiu. Kvatinė algoritmo dalis yra naudojama gradientų ir $f(x, \theta)$ rezultatams apskaičiuoti, klasikinė – parametrams θ atnaujininti. Nuodugnus mokymo procesas yra pavaizduotas 8.6 pav. Klasikinė modelio mokymo dalis pavaizduota brūkšnine linija.



8.6 pav.: Hibridinis kvantinis-klasikinis mokymo algoritmas. Kvatinis įrenginys yra naudojamas išvesties ir gradientų suskaičiavimui; kvatinės grandinės parametrai θ atnaujinami naudojantis klasikiniu algoritmu

Panagrinėkime algoritmą išsamiau. Įvesties duomenys x yra pavaizduojami į n kubitų pradinę būseną taiknt amplitudžių kodavimą. Duomenų įrašas $x = \{x_1, x_2, \dots, x_N\}$, $x \in \mathbb{R}$, turintis $N = 2^n$ elementų, yra pavaizduojamas bazinių vektorių $|i\rangle$ superpozicijos amplitudėse x_i :

$$|\varphi(x)\rangle = \frac{1}{\chi} \sum_{i=1}^N x_i |i\rangle, \quad (8.38)$$

kur

$$\chi = \sqrt{\sum_{i=1}^N x_i^2}. \quad (8.39)$$

Prognozuojamos žymos $\ell(x) = \lambda_1, \lambda_2, \dots$, atitinkančios duomenis x , yra nuskaitomos taikant išvesties būsenos matavimą. Ši matavimą atitinka ermitinis operatorius C , kurio tikrinės vertės yra $\lambda_1, \lambda_2, \dots$:

$$C = \sum_j \lambda_j P_j. \quad (8.40)$$

Čia P_j yra projekcinis operatorius, projektuojantis į atitinkamą poerdvį. Klasifikatoriaus mokymo tikslas yra teisingos žymos y prognozavimo maksimizavimas. Tam naudojama tikslo funkcija $\mathcal{L}(\theta)$:

$$\mathcal{L}(\theta) = \frac{1}{M} \sum_{\lambda_j} \sum_{x: \ell(x)=\lambda_j} \langle \varphi(x) | U^\dagger(\theta) P_{\lambda_j} U(\theta) | \varphi(x) \rangle, \quad (8.41)$$

čia M – duomenų rinkinio dydis. Binariosios klasifikacijos atveju, kai yra dvi žymos λ_1 ir λ_2 ,

tikslo funkciją galime užrašyti taip:

$$\begin{aligned}\mathcal{L}(\theta) = & \frac{1}{M} \sum_{x: \ell(x)=\lambda_j} \langle \varphi(x) | U^\dagger(\theta) P_1 U(\theta) | \varphi(x) \rangle \\ & - \frac{1}{M} \sum_{x: \ell(x)=\lambda_2} \langle \varphi(x) | U^\dagger(\theta) P_2 U(\theta) | \varphi(x) \rangle.\end{aligned}\quad (8.42)$$

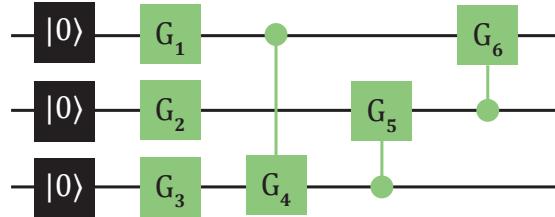
Vidines sandaugas galima efektyviai apskaičiuoti pasitelkiant papildomą ancila kubitą, kaip aprašyta Hadamardo teste. Tikslo funkcija $\mathcal{L}(\theta)$ yra maksimizuojama taikant klasikinį gradientinio nusileidimo algoritmą. Kvantine grandinė, realizuojanti $U(\theta)$, konstruojama taip, kad sparčiai sukurtų kvantine supynimą ir nepareikalautų gilių grandinių. Kaip pamename, didžioji dalis 2^n būsenų yra supintosios, tad panaudojant supintąsių būsenas atsiranda didesnė tikimybė teisingai apmokytį klasifikatoriu. Operatorių $U(\theta)$ realizuojanti grandinė yra sudaroma iš blokų:

$$U(\theta) = G_{\text{out}}(\theta_{\text{out}}) B_L(\theta_L) \cdots B_2(\theta_2) B_1(\theta_1). \quad (8.43)$$

Kiekvienas blokas $B_j(\theta_j)$ yra sudarytas iš n skaičiaus 1 kubito loginių vartų, n skaičiaus salyginių 2 kubito loginių vartų. Užbaigama 1 kubito loginiai vartais G_{out} matavimui, skirtam pirmajam grandinės kubitiui. Vieno kubito loginiai vartai yra bendriausio tipo $U_3(\alpha, \beta, \gamma) \equiv G$ (žr. 4.1 poskyri). O štai 2 kubito salyginiai cG galime užrašyti:

$$cG = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes G. \quad (8.44)$$

Salyginiai vartai čia sudaro ciklinį kodą, kuris yra charakterizuojamas artumo parametru r ($0 < r < n$). Kiekvienam kubitiui, kurio numeris j , pritaikomi salyginiai loginiai vartai, kuriuose j -asis kubitas yra adresatas, o $(j+r)$ mod (n) -tas kubitas kontroliuojantis. Trijų kubitų atveju, kai $r = 1$, toks blokas pavaizduotas 8.7 pav.



8.7 pav.: 3 kubitų kvantine supynimą sudarančios grandinės blokas. G_j yra 1 kubito loginiai vartai

Tikslo funkcijos $\mathcal{L}(\theta)$ maksimizavimui taikant gradientinio nusileidimo metodą reikia operatorių $U(\theta)$ išvestinių θ parametru atžvilgiu. Kadangi realizuojame $U(\theta) \rightarrow U_3(\alpha, \beta, \gamma)$, tai savo ruožtu reiškia išvestines (α, β, γ) atžvilgiu. Vieno kubito loginių vartų išvestinės, pvz., $I \otimes \partial_\theta G \otimes I \otimes \cdots \otimes I$, yra taip pat vieno kubito loginiai vartai, tačiau salyginių loginių vartų išvestinės $\partial_\theta(cG)$ nėra unitarūs operatorius. Vis dėlto $\partial_\theta(cG)$ gali būti realizuojamas kaip dviejų unitariųjų operatorių suma:

$$\partial_\theta(cG) = |1\rangle\langle 1| \otimes \partial_\theta G = \frac{1}{2}(I \otimes \partial_\theta G - Z \otimes \partial_\theta G). \quad (8.45)$$

Kiekviena iš šių grandinių, turinčių $I \otimes \partial_\theta G$ bei $Z \otimes \partial_\theta G$ narius, yra įvykdoma atskirai, o jų skirtumas suskaičiuojamas klasikinėje algoritmo optimizavimo dalyje. Galop, norint įvertinti tikimybes, kvantine grandinė turi būti pakartotinai įvykdoma kelis kartus. Pakartojimų skaičius auga su norimu tikslumu ϵ kaip $O(1/\epsilon^2)$.

8.3.4 Pagrindinių komponentų analizė

Kvantinė pagrindinių komponentų analizė (angl. *quantum principal component analysis*, trumpinys qPCA) leidžia nustatyti nežinomas kvantinės būsenos, apibūdinamos tankio matrica ρ , tikriniai vektorius, atitinkančius didžiausias tikrines vertes. Tankio matrica ρ yra išskaidoma tikriniais vektoriais:

$$\rho = \sum_{k=1}^N \lambda_k |a_k\rangle\langle a_k|, \quad (8.46)$$

čia N – erdvės dydis; $|a_k\rangle$ – tikriniai vektoriai; λ_k – atitinkamos tikrinės vertės. Kvantinės pagrindinių komponentų analizės tikslas – išrinkti $|a_k\rangle$ atitinkančius didžiausius λ_k . Kadangi tankio matrica ρ yra ermitinė, operatorius $U = e^{-i\rho t}$ yra unitarusis. Tikrinių vektorių nustatymui kvantiniu kompiuteriu galima pritaikyti hamiltoniano kodavimą, aprašytą (7.39) lygtimi. Operatoriaus U tikrines vertes $e^{i\lambda_k t}$ randamos panaudojant kvantinį fazes nustatymo algoritmą, aprašytą 7.4 poskyryje.

Algoritme reikia apskaičiuoti matricos ρ eksponentę. Ši skaičiavimą paspartina paprastas matematinis triukas pasitelkiant bet kokią pagalbinę tankio matricą σ . Laiką t padalijame į s dalis, $t = s\Delta t$, ir darome prielaidą, kad turime daug būsenos ρ kopijų. Galima pastebėti, kad galioja tokia lygybė:

$$\text{Tr}_P e^{-iW\Delta t} \rho \otimes \sigma e^{iW\Delta t} = \sigma - i\Delta t [\rho, \sigma] + O(\Delta t^2) = e^{-i\rho\Delta t} \sigma e^{i\rho\Delta t} + O(\Delta t^2), \quad (8.47)$$

kur Tr_P yra dalinis pėdsakas per pirmą kintamąjį, o W – *SWAP* loginiai vartai. *SWAP* operatoriaus eksponentė gali būti efektyviai realizuota kvantinėse grandinėse, todėl, naudojantis (8.47) lygtimi, galima efektyviai realizuoti ir tankio matricos eksponentės skaičiavimą.

IX skyrius

Kvantinių klaidų aptikimas ir taisymas

Klaidų taisymas iš pirmo žvilgsnio neatrodo itin estetiskas ar įdomus užsiemimas. Tačiau kvantinėje kompiuterijoje tai yra viena iš labiausiai apšviečiančių ir stebinančių sričių. Dekoherencija yra artimai susijusi su mus supančių klasikinių reiškinių atsiradimui iš pasaulio, kuris fundamentaliai vadovaujasi kvantinėmis taisyklemis. Dekoherencija – pagrindinė priežastis, kuri neleidžia realizuoti makroskopinio dydžio objektų, esančių superpozicijos būsenose. Atliekant klaidų analizę ir taisymą taip pat geriau atskleidžia gili informacijos sąvokos reikšmę ir kvantinio supynimo svarba. Informacijos atskleidimas gali sugriauti sistemos superpozicijos būseną net ir tada, jeigu su šia sistema nėra tiesioginės sąveikos.

9.1 Klasikinės ir kvantinės klaidos

Kvantinė kompiuterija pasikliauja delikačiomis kubitų superpozicijos būsenomis. Praktikoje kubitai nėra idealiai izoliuoti nuo aplinkos, jų neišvengiama sąveika su išorinėmis sistemomis mažina gebėjimą išlikti superpozicijos būsenose ilgą laiką. Pirmame skyriuje minėtos T_1 ir T_2 dekoherencijos trukmės atspindžiai, kaip intensyviai išorinės sąveikos vyksta su kubitais, taip pat įvardija laiko skalę, pagal kurią galima spręsti, kiek loginių operacijų įmanoma atlikti iki tol, kol neatsiras didelė klaidų tikimybė. Dekoherencija nėra vienintelis klaidų šaltinis – atliekamų loginių vartų netikslumai taip pat praktiskai neišvengiami. Unitariosios operacijos yra nusakomos tolydžiai kintančiais parametrais, todėl praktikoje atsiremiaama į ribotą loginių vartų tikslumą. Nepaisant šių klaidų šaltinių, praktinės svarbos pageidaujamo ilgumo kvantiniai skaičiavimai gali būti atlikti pasitelkiant klaidų taisymo algoritmus. Tam yra reikalaujama, kad dekoherencijos trukmės nebūtų per daug trumpos, o loginių vartų netikslumai – per daug dideli. Tada pasitelkus papildomus išteklius – kubitus ir logines operacijas – galima formaliai pasiekti kladoms atsparius skaičiavimus. Šiame procese klaidos yra taisomos dinaminėje situacijoje viso skaičiavimo proceso metu. Žinoma, kvantines būsenas norima apsaugoti ir statinėse situacijose – siunčiant kubitus kvantiniaiš rūšiai ar saugant kvantinio kompiuterio atmintyje. Kaip matysime, panašūs principai yra taikomi abiem situacijoms.

Šiuolaikiniai klasikiniai kompiuteriai yra itin atsparūs skaičiavimo kladoms ir šiuo atžvilgiu gali būti traktuojami kaip esantys be trūkumų. Klaidos turi didesnę tikimybę atsirasti siunčiant skaitmeninę informaciją komunikacijos kanalais ir ją užrašant į atmintį. Klaidų atsiradimas daugeliu atvejų yra nenuspėjamas (formaliau – stochastinis) ir šnekamojoje kalboje vadinamas **triukšmu** (angl. *noise*). Pagrindinis principas norint užtikrinti, kad esant triukšmui informacija nebus pra-

rasta, yra pasitelkti papildomą, vadinamąją perteklinę informaciją. Jeigu dalis informacijos ir yra prarandama, perteklinė informacija padeda užtikrinti, kad informacijos turinys bus sėkmingai atstatytas. Ši principą taiko ir žmonės tarpusavio komunikacijoje, kai paprašoma pakartoti gerai neišgirstą sakinį. Kaip to pavyzdjį kompiuterijoje imkime, kad Agnė ketina nusiųsti Benui informaciją dvejetainiu pavidalu naudodama triukšmingą komunikacijos kanalą, kuriame triukšmo efektas yra kiekvieną bitą apversti ($0 \leftrightarrow 1$) su tikimybe p ($0 \leq p \leq 1$), nepriklausomai nuo kitų bitų verčių. Tad tikimybė, kad bus gautas teisingas bitas, yra $1 - p$. Siekdama apsaugoti dvejetainę informaciją, Agnė kiekvieną turinio bitą prieš siuntimą pakeičia trimis identiškais bitais: $0 \rightarrow 000$, $1 \rightarrow 111$. Šios 0 ir 1 trijų bitų sekos yra formaliai vadinamos loginiais 0 ir 1 bitais, o pasirinktas specifinis būdas perteikti bitų vertėms vadinamas kodu. Benas, žinodamas Agnės kodavimo būdą ir taikydamas **daugumos balsavimo metodą** (angl. *majority voting*), gavęs bitų seką gali nuspresti, koks bitas jam buvo siūstas. Pavyzdžiui, jeigu gauta seka yra 001, daugumos balso principu jis nusprenčia, kad įvyko kaida trečiajame bite ir buvo siūsta 000. Toks trijų bitų kodavimo būdas bus sėkmingas, jeigu kode atsiranda ne daugiau nei viena kaida. Galima nesunkiai parodyti, kad bendra tikimybė, nusakanti, jog įvyks nepataisoma dviejų ar trijų bitų apvertimo kaida, yra $3p^2(1-p) + p^3$. Tad palyginus su p , kai siunčiamas pavienis bitas, trijų bitų kodas sumažina nepataisomą klaidų tikimybę, jeigu $p < 0.5$, ir suteikia sparčiai didėjantį pranašumą toliau mažėjant p .

Klasikiniuose klaidų taisymo koduose yra taikomas bitų kopijavimas pridedant perteklinę informaciją, o siekiant klaidas aptikti ir jas taisyti bitų sekos yra tiesiogiai nuskaitomos. Akivaizdu, kad abu šie procesai negali būti pritaikomi kvantiniam kompiuteriui ir kvantiniams ryšiams. Bendrosios kvantinės būsenos nejmanoma nukopijuoti, o tokios būsenos tiesioginis nuskaitymas sugriauna superpoziciją ir joje laikomą informaciją. Be to, bitų apvertimas yra vienintelė galima kaida klasikinėje terpéje; o štai kvantinių klaidų įvairovė yra didesnė, nes klaidos kinta tolydziai dėl analoginio kvantinių būsenų pobūdžio. Pavyzdžiui, 1 kubito būseną $|\psi\rangle$ nusakome Blocho vektoriumi, kuris apibūdinamas dvielem tolydziais parametrais, nurodančiais kampus. Jeigu loginiais vartais norima pasukti šį vektorių, sakykime, apie x ašį kampu θ , tačiau gaunamas $\theta + \varepsilon$, netikslumas ε yra viena galima kaida. Tai galima formaliai užrašyti dvielem unitariais operatoriais, veikiančiais kubitą paeiliui:

$$ER_x(\theta)|\psi\rangle = e^{-i\varepsilon X/2}e^{-i\theta X/2}|\psi\rangle. \quad (9.1)$$

Čia $R_x(\theta) = e^{-i\theta X/2}$ nusako tikslią operaciją, o po jos rašomas klaidos operatorius $E = e^{-i\varepsilon X/2}$. Nepageidaujamos išorinės sąveikos taip pat gali įvesti šias klaidas. Įsivaizduokime vėl, kad sąveikos efektas yra pasukti Blocho vektorių aplink x ašį kampu ε , kai pradinė kubito būsena yra $|0\rangle$. Siekiant supaprastinti simboliką, toliau minuso ženklą ir faktorių 2 įtrauksime į ε . Randame paveikštą būseną:

$$e^{i\varepsilon X}|0\rangle = [\cos(\varepsilon)I + i\sin(\varepsilon)X]|0\rangle = \cos(\varepsilon)|0\rangle + i\sin(\varepsilon)|1\rangle. \quad (9.2)$$

Tai formaliai nusako nepaveiktos būsenos $|0\rangle$ ir kladinės būsenos $|1\rangle$ superpoziciją. Tikimybė, kad atlikus matavimą kubitas bus rastas būsenose $|0\rangle$ arba $|1\rangle$, kai ε yra itin mažas, tampa:

$$p(|0\rangle) = \cos^2(\varepsilon) \approx 1 - \varepsilon^2; \quad (9.3)$$

$$p(|1\rangle) = \sin^2(\varepsilon) \approx \varepsilon^2. \quad (9.4)$$

Jeigu ši sąveika, ar loginių vartų paklaida, įvyktų sistemiškai n kartų, tada tikimybės būtų atitinkamai $1 - (n\varepsilon)^2$ ir $(n\varepsilon)^2$. Itin mažos paklaidos skaičiavimo metu gali būti toleruotinos, nes tikimybė rasti kladinę būseną bus itin maža. Tačiau dideliuose algoritmuose, tokiuose kaip atliekant Šoro pirminių skaičių faktorizavimą, loginių vartų skaičius gali siekti $\sim 10^{10}$ ir daugiau.

Loginių vartų paklaida ε atitinkamai turi būti mažesnė nei $\sim 10^{-10}$. Šios knygos rašymo metu loginių vartų tikslumas siekia $\sim 10^{-4}$, tad klaidų taisymo algoritmai yra pageidautini.

Kvantinis supynimas ir su juo įvedamos koreliacijos klaidų taisymo algoritmuose dar kartą iliustruoja ypatingą šio ištekliaus svarbą. Nepaisant klaidų analoginio pobūdžio ir begalinio tikslumo norint jas apibūdinti, šis išteklius užtikrina, kad reikia taisyti tik diskrečias trijų tipų klaidas. Kitaip tariant, kvantinėje kompiuterijoje klaidos yra efektyviai skaitmenizuojamos. Pirmojo tipo klaida yra, kaip ir klasikinėje skaitmeninėje kompiuterijoje, vadinama **bito apvertimo klaida** (angl. *bit-flip error*), kuri sukeičia kubito būsenas $|0\rangle \leftrightarrow |1\rangle$. Kubito apvertimo klaidos procesas yra nusakomas Pauli- X operatoriumi, kuris veikdamas bendrą superpozicijos būseną turi efekta:

$$X(a|0\rangle + b|1\rangle) = a|1\rangle + b|0\rangle. \quad (9.5)$$

Fazės apvertimo klaida (angl. *phase-flip error*) yra išskirtinai kvantinio pobūdžio, nes klasikinėje kompiuterijoje fazės atitinkmens nėra. Fazės apvertimo klaidos atsiradimas yra nusakomas Pauli- Z operatoriumi:

$$Z(a|0\rangle + b|1\rangle) = a|0\rangle - b|1\rangle. \quad (9.6)$$

Matyti, kad santykinė fazė tarp $|0\rangle$ ir $|1\rangle$ kubito būsenų yra pakeičiama. Galiausiai, bito ir fazės apvertimo klaidų kombinacija, XZ , yra trečia galima klaida. Primename Pauli operatorių sąryšį $XZ = -iY$. Tod šių dviejų klaidų kombinaciją iki globalios fazės galime išreikšti Pauli- Y loginiai vartais:

$$-iY(a|0\rangle + b|1\rangle) = a|1\rangle - b|0\rangle. \quad (9.7)$$

9.2 Kvantinis supynimas su aplinka ir klaidų atsiradimas

Loginių vartų netikslumai po kiekvieno jų pritaikymo gali įvesti tolydžiai kintančias klaidas. Sistemiškai atsirandancios vienodo tipo klaidos yra lengvai aptinkamos bei ištaisomos, na, o, kintančios atsitiktiniu būdu įveda triukšmo pobūdį. Tačiau, net ir palikus kubitus ramybėje, jų būsenos gali būti paveikiamos nekontroliuojamų sąveikų su išorinėmis sistemomis. Bendrą kubitų ir aplinkos kvantinę sistemą visada galime apibūdinti kaip naują išplėstinę sistemą, kuri kinta laike deterministiškai vadovaujantis Šriodingerio lygtimi. Tačiau dėl informacijos apie įvykusias sąveikas trūkumo mūsų požiūriu bus stebimi atsitiktiniai, triukšmo pobūdžio, kubitų būsenų pokyčiai. Dėl sąveikų tarp kvantinių sistemų bendroje situacijoje atsiranda supynimas. Siekdami iliustruoti supynimo įtaką imkime paprastą pavyzdį, kuriam kubitas yra paruoštas pradinėje superpozicijos būsenoje:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \quad (9.8)$$

Jeigu atliksime šiai būsenai (idealai veikiančius) Hadamardo loginius vartus, ji taps $H|\psi\rangle = |0\rangle$. Tad išmatavus kubito būseną su $p = 1$ tikimybe rasime $|0\rangle$. Sakykime, kad prieš atliekant H kubitas patyrė sąveiką U su aplinkos sistema $|e\rangle$, ir tai lėmė jų kvantinį supynimą ir bendrą būseną:

$$U|\psi\rangle \otimes |e\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |e_1\rangle + |1\rangle \otimes |e_2\rangle). \quad (9.9)$$

Čia $|e_0\rangle$ ir $|e_1\rangle$ yra aplinkos sistemos būsenos. Nežinodami apie įvykusią sąveiką, atliekame kubitui Hadamardo vartus:

$$H \otimes I(U|\psi\rangle \otimes |e\rangle) = \frac{1}{2}(|0\rangle + |1\rangle) \otimes |e_1\rangle + \frac{1}{2}(|0\rangle - |1\rangle) \otimes |e_2\rangle. \quad (9.10)$$

Matome, kad kvantinis supynimas neleidžia panaikinti kubito būsenos $|1\rangle$, kaip tai atsitinka dėl interferencijos $H|\psi\rangle$. Tolesniame žingsnyje vėl išmatuojame kubito būseną, tikimybės $p(0)$ ir $p(1)$ rasti $|0\rangle$ ir $|1\rangle$ yra:

$$\begin{aligned} p(0) &= \frac{1}{4}(\langle e_1|e_1\rangle + \langle e_2|e_2\rangle + \langle e_1|e_2\rangle + \langle e_2|e_1\rangle) \\ &= \frac{1}{4}(\langle e_1|e_1\rangle + \langle e_2|e_2\rangle + 2\text{Re}[\langle e_1|e_2\rangle]); \end{aligned} \quad (9.11)$$

$$p(1) = \frac{1}{4}(\langle e_1|e_1\rangle + \langle e_2|e_2\rangle - 2\text{Re}[\langle e_1|e_2\rangle]). \quad (9.12)$$

Norėdami įvertinti šias tikimybes, turime daugiau pasakyti apie išorinę sistemą. Darydami prie-laidą, kad ji yra normuotoji, o būsenos ortogonaliosios, $\langle e_1|e_2\rangle = 0$, randame $p(0) = p(1) = 1/2$. Tai nusako lygias tikimybes rasti $|0\rangle$ arba $|1\rangle$ pamatavus sistemos kubito būseną. Todėl nežinant apie įvykusią sąveiką ir kvantinį supynimą mūsų požiūriu atrodis, kad kubito būsena tampa viškai atsitiktinė, o ne $|0\rangle$, kaip tikėtasi. Informacija, koduojama bendrosios būsenos amplitudėse, tampa efektyviai nebepasiekama, nes yra delokalizuojama koreliacijose tarp aplinkos ir kubito. Sąveikos su aplinka priveda kvantines sistemas prie dekoherencijos, dėl kurios jos panašėja į klasikines triukšmingas sistemas.

Norėdami aiškiau pamatyti, kaip atsiranda trys minėtos Pauli operatoriai nusakomos kubitų būsenų kaidos, imkime bendriausio tipo unitariają transformaciją U , veikiančią kubito ir aplinkos kvantinę sistemą. Ji nebūtinai apibūdinama dviem skirtingomis būsenomis, kaip kubitai, bet gali turėti jų daug daugiau. Transformacijos įtaka kubito standartiniams baziniams vektoriams ir aplinkos pradinei būsenai $|e\rangle$ išreiškiama taip:

$$U|0\rangle \otimes |e\rangle = |0\rangle \otimes |e_1\rangle + |1\rangle \otimes |e_2\rangle; \quad (9.13)$$

$$U|1\rangle \otimes |e\rangle = |0\rangle \otimes |e_3\rangle + |1\rangle \otimes |e_4\rangle. \quad (9.14)$$

Bendroje situacijoje, aplinkos sistemos būsena, kuri dalyvavo sąveikoje, gali būti nenormuotoji ir skirtinį $|e_i\rangle$ tarpusavyje neortogonalūs, $\langle e_i|e_j\rangle \neq 0$. Imkime bendrą kubito būseną $|\psi\rangle = a|0\rangle + b|1\rangle$, tada randame:

$$\begin{aligned} U|\psi\rangle \otimes |e\rangle &= a(|0\rangle \otimes |e_1\rangle + |1\rangle \otimes |e_2\rangle) + b(|0\rangle \otimes |e_3\rangle + |1\rangle \otimes |e_4\rangle) \\ &= \frac{1}{2} \left[(a|0\rangle + b|1\rangle) \otimes (|e_0\rangle + |e_3\rangle) + (a|1\rangle + b|0\rangle) \otimes (|e_1\rangle + |e_2\rangle) \right. \\ &\quad \left. + (a|0\rangle - b|1\rangle) \otimes (|e_0\rangle - |e_3\rangle) + (a|1\rangle - b|0\rangle) \otimes (|e_1\rangle - |e_2\rangle) \right]. \end{aligned} \quad (9.15)$$

Antroje eilutėje pergrupavome būsenas siekdamis parodyti, kad kubito sąveika su aplinka gali būti išreikšta Pauli operatoriai, veikiančiais kubito superpozicijos būseną:

$$U|\psi\rangle \otimes |e\rangle = I|\psi\rangle \otimes |e_I\rangle + X|\psi\rangle \otimes |e_x\rangle + Z|\psi\rangle \otimes |e_z\rangle + XZ|\psi\rangle \otimes |e_{xz}\rangle. \quad (9.16)$$

Aplinkos būsenas pervadinome taip:

$$\begin{aligned} |e_I\rangle &= \frac{(|e_0\rangle + |e_3\rangle)}{2}, & |e_x\rangle &= \frac{(|e_1\rangle + |e_2\rangle)}{2}, \\ |e_z\rangle &= \frac{(|e_0\rangle - |e_3\rangle)}{2}, & |e_{xz}\rangle &= \frac{(|e_1\rangle - |e_2\rangle)}{2}. \end{aligned} \quad (9.17)$$

Matome, kad aplinkos ir kubito sistema tampa supintąja. Pirmoji superpozicijoje būsena $I|\psi\rangle \otimes |e_I\rangle$ įvardija nepakitusią kubito pradinę būseną. Kubito apvertimo kaida yra nusakoma nariu

$X|\psi\rangle \otimes |e_x\rangle$, o štai fazės kaida, taip pat fazės ir kubito apvertimo klaidų kombinacija nusako būsenas $Z|\psi\rangle \otimes |e_z\rangle$ ir $XZ|\psi\rangle \otimes |e_xz\rangle$, atitinkamai. Tai neturėtų būti stebinantis rezultatas, kadangi visos (2×2) dydžio unitariosios matricos, nusakančios visas įmanomas 1 kubito būsenų transformacijas, gali būti išreikštos Pauli matricų $\{I, X, Y, Z\}$ tiesinėmis kombinacijomis.

9.3 Bito apvertimo klaidos aptikimas ir taisymas

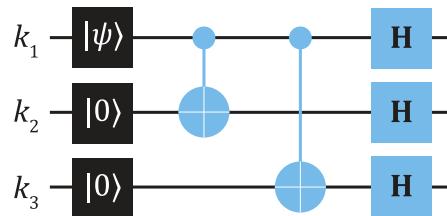
Darome prielaidą, kad klaidos atsiranda kiekviename registro kubite atskirai nuo kitų kubitų. Tai yra vadinamosios nekoreliuotos triukšmo pobūdžio klaidos. Formaliai sakysime, kad klaidos atsiranda siučiant kvantines būsenas per **triukšmingą kvantinį kanalą** (angl. *noisy quantum channel*). Tai gali įvardyti kubitų siuntimą kvantiniai ryšiais arba kubitų laiko evoliuciją tam tikru laiko intervalu kvantiniame kompiuteryje. Kubitas nukeliauja kanalą nepaveiktas su tikimybe $1 - p$ ir patiria kaidą su tikimybe p . Šioje stadijoje taip pat darome prielaidą, kad loginiai vartai veikia idealiai, be netikslumų. Dinaminę klaidų taisymo metodologiją, apimančią dekoherencijos ir loginių vartų efektus kartu, aptariame šio skyriaus pabaigoje.

Kvantinių klaidų taisymo algoritmai yra dažnai pristatomai pradedant nuo 3 kubitų kodų, skirtų taisyti bito apvertimo arba fazės apvertimo klaidas. Pavieniui jie nėra pilnieji kodai, galintys ištisinti visas klaidų kombinacijas, tačiau leidžia pamatyti esminius klaidų aptikimo ir taisymo principus. Šių dviejų kodų sujungimui **konkatenacijos būdu** (angl. *concatenation*) yra pagrįstas Šoro 9 kubitų kodas – vienas iš pirmųjų gebantis ištisinti bendrojo tipo klaidas.

Kaip ir klasikiniame bito apvertimo klaidos pavyzdyme, loginis kubitas yra sudaromas iš trijų fizinių kubitų. Vieno kubito būsena $|\psi\rangle$ yra perteikiama loginiu kubitu $|\psi\rangle_L$ taip:

$$|\psi\rangle = a|0\rangle + b|1\rangle \rightarrow |\psi\rangle_L = a|0\rangle_L + b|1\rangle_L = a|000\rangle + b|111\rangle. \quad (9.18)$$

Būsenos yra normuotosios, $|a|^2 + |b|^2 = 1$, o 1 kubito baziniai vektoriai koduojami $|0\rangle \rightarrow |000\rangle$, $|1\rangle \rightarrow |111\rangle$. Atkreipiame dėmesį, kad loginiame kubite $|\psi\rangle_L$ baziniai vektoriai yra „patrigubinami”, tačiau $|\psi\rangle$ būsena nėra nukopijuojama tris kartus, $|\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle$, ir todėl neprieštarauja uždraustojo kopijavimo teoremai. Loginis kubitas $|\psi\rangle_L$ nusako supintąjį trijų fizinių kubitų kvantinę būseną, kurią galima sukurti pradedant nuo kubito $|\psi\rangle$ būsenoje ir dviejų papildomų kubitu $|00\rangle$ būsenoje atliekant dvejus $CNOT$ loginius vartus.



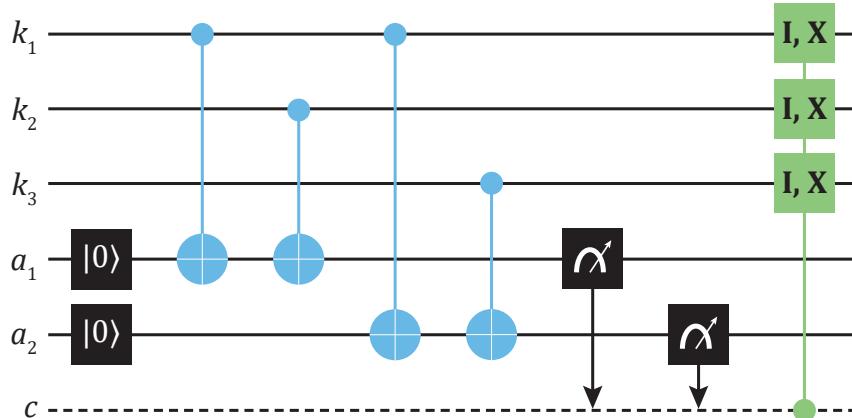
9.1 pav.: Loginio kubito būsenos $|\psi\rangle_L$ paruošimas naudojant 3 fizinius kubitus

Visi trys fiziniai kubitai, formuojantys loginį kubitą, gali būti paveikti triukšmo; tad šis kodas bus veiksmingas, jeigu bito apvertimo kaida atsiranda ne daugiau nei viename iš trijų kubitų. Toliau pažiūrėkime, kaip aptikti atsirandančią bito apvertimo kaidą loginiame kubite.

Tiesioginiai 3 kubitų būsenų matavimai nėra išeitis aptikti ir taisyti klaidas. Sakykime, kad atsiranda kaida pirmajame kubite, kuri pakeičia $|\psi\rangle_L$ būseną taip:

$$|\psi\rangle_L \rightarrow a|100\rangle + b|011\rangle. \quad (9.19)$$

Atlikę Pauli- Z matavimus su visais 3 kubitais rastume $|100\rangle$ arba $|011\rangle$ būsenas su $|a|^2$ ir $|b|^2$ tikimybėmis, atitinkamai. Tai užtikrintų kaidos aptikimą, tačiau superpozicija bus sugriauta, o negalėdami sužinoti a ir b amplitudžių šios būsenos nebeatstatysime. Aptikti kaidas 3 kubitų kode galima pritaikius vadinamajį nelokalų matavimą, pasitelkiant papildomus ancila kubitus. Atkreipiame dėmesį, kad abu kodai $|0\rangle_L$ ir $|1\rangle_L$ yra $Z \otimes Z \otimes I$, $Z \otimes I \otimes Z$ ir $I \otimes Z \otimes Z$ operatorių tikriniai vektoriai su vienodomis tikrinėmis vertėmis $\lambda = 1$. Pavyzdžiu, $Z \otimes Z \otimes I |0\rangle_L = |0\rangle_L$ ir $Z \otimes Z \otimes I |1\rangle_L = |1\rangle_L$. Toki dėsningumą matome iš to, kad Pauli- Z operatoriai tensorinėje operatorių sandaugoje, veikdami pavienių kubitų būsenas $|0\rangle$ ir $|1\rangle$ jas sudaugina su $\lambda = 1$ ir $\lambda = -1$ tikrinėmis vertėmis, atitinkamai. Tad $Z \otimes Z \otimes I$ veikdamas pirmąjį ir antrąjį kubitus $|000\rangle$ ir $|111\rangle$ būsenose, arba jų superpozicijoje, sudaugina bendrą būseną su $\lambda_1 \lambda_2 = 1$. Tačiau, jeigu vienas iš šių dviejų kubitų patyrė bito apvertimo kaidą, tada jų būsenos skirsis, o tikrinį verčių sandauga taps $\lambda_1 \lambda_2 = -1$. Siekiant nustatyti, kuriame iš trijų kubitų įvyko kaida, pakanka atlikti du matavimus $Z \otimes Z \otimes I$ ir $Z \otimes I \otimes Z$, arba bet kurią iš kitų dviejų porų kombinacijos. Pavyzdžiu, jeigu $Z \otimes Z \otimes I$ ir $Z \otimes I \otimes Z$ matavimų tikrinės vertės yra abi $\lambda_1 \lambda_2 = \lambda_1 \lambda_3 = -1$, galime unikaliai konstatuoti, kad įvyko kaida pirmajame kubite. Kitos dvi galimybės $\lambda_1 \lambda_2 = -1$ ir $\lambda_1 \lambda_3 = 1$ bei $\lambda_1 \lambda_2 = 1$ ir $\lambda_1 \lambda_3 = -1$ indikuoja kaidą antrajame ir trečiajame kubite, atitinkamai. Kvintinė grandinė 9.2 pav. iliustruoja bito apvertimo kaidos aptikimą ir taisymą.



9.2 pav.: Bito apvertimo kaidos aptikimą ir taisymą atliekanti grandinė

Siekdami atlikti $Z \otimes Z \otimes I$ ir $Z \otimes I \otimes Z$ operatorių matavimus panaudojame papildomus ancila kubitus, inicializuotus pradinėje $|00\rangle$ būsenoje. Keturi $CNOT$ vartai su ancila adresatiniais kubitais nusako aptikimo stadiją, kuri yra pagrįsta dviejų kubitų lyginimo nustatymu. Matome, kad $Z \otimes Z \otimes I$ matavimas realizuojamas keičiant pirmosios ancilos a_1 būseną. Esant skirtingoms kubitų k_1 ir k_2 būsenoms, ji tampa $|a_1\rangle = |k_1 \oplus k_2\rangle$; čia \oplus yra $\bmod(2)$ bitų sudėtis. Tad, jeigu $|k_1\rangle = |k_2\rangle$, tada $|a_1\rangle = |0\rangle$, ir tai atitinka tikrinę vertę $\lambda_1 \lambda_2 = 1$, indikuojančią lyginį lyginumą. Jeigu $|k_1\rangle \neq |k_2\rangle$, tada $|a_1\rangle = |1\rangle$, ir tai nusako tikrinę vertę $\lambda_1 \lambda_2 = -1$ bei nelyginį lyginumą. Operatoriaus $Z \otimes I \otimes Z$ matavimas yra analogiškai užrašomas keičiant antrosios ancilos būseną $|a_2\rangle = |k_1 \oplus k_3\rangle$. Užbaigiant aptikimo stadiją, bendra loginio kubito ir ancilių su kaida pirmajame fiziniame kubite būsena tampa:

$$|\psi\rangle_L \rightarrow (a|100\rangle + b|011\rangle) \otimes |11\rangle. \quad (9.20)$$

Atkreipiame dėmesį, kad ancilos ir loginio kubito būsena yra faktorizuojamoji, todėl ancilių kubitų matavimo procesas neturi įtakos loginio kubito būsenai. Ancilių kubitų būsenos yra išmatuoja-

mos, ir tai leidžia aptikti įvykusią klaidą. Čia svarbu atkreipti dėmesį, kad kode atlikti nelokalūs matavimai suteikia informaciją apie koreliacijas tarp dviejų būsenų, nusakančią, ar jos vienodos, ar skirtinges (lyginumas). Šios informacijos pakanka klaidų nustatymui neatskleidžiant loginio kubito būsenos amplitudžių a ir b . Jų atskleidimas sugriautų superpoziciją ir joje laikomą informaciją.

9.1 lentelė: Faktorizuojamosios loginio kubito ir 2-jų ancilų kubitų būsenos, kurios indikuoją klaidos sindromą loginiame kubite. Tikiemybės dešinėje nurodo rasti atitinkamas būsenas naudojant 3 kubitų bito apvertimo klaidos taisymo kodą. Pirma būsena lentelėje atitinka nepaveikštą, kitos trys nusako bito klaidą viename iš trijų fizinių kubitų. Dar kitos trys būsenos nusako būsenas su dviem bito klaidomis skirtinguose kubituose, paskutinioji – su bito klaidomis visuose trijuose.

Būsena po klaidos sindromo nustatymo	Tikimybė rasti šią būseną
$(a 000\rangle + b 111\rangle) \otimes 00\rangle$	$(1-p)^3$
$(a 100\rangle + b 011\rangle) \otimes 11\rangle$	$p(1-p)^2$
$(a 010\rangle + b 101\rangle) \otimes 10\rangle$	$p(1-p)^2$
$(a 001\rangle + b 110\rangle) \otimes 01\rangle$	$p(1-p)^2$
$(a 110\rangle + b 001\rangle) \otimes 01\rangle$	$p^2(1-p)$
$(a 101\rangle + b 010\rangle) \otimes 10\rangle$	$p^2(1-p)$
$(a 011\rangle + b 100\rangle) \otimes 11\rangle$	$p^2(1-p)$
$(a 111\rangle + b 000\rangle) \otimes 00\rangle$	p^3

Visos įmanomos ancilų kubitų būsenos, šiuo atveju keturios skirtinges, yra vadinamos **klaidos sindromais** (angl. *error syndrome*). 9.1 lentelė nusako visas apvertimo klaidas kartu su atitinkamomis būsenomis ir tikimybėmis šią būseną rasti.

Pagal sindromo būseną, yra pritaikomi Pauli- X loginiai vartai pažeistam kubitui ir taip ištaisoma bito apvertimo klaida: $|11\rangle \rightarrow X \otimes I \otimes I$, $|10\rangle \rightarrow I \otimes X \otimes I$, $|01\rangle \rightarrow I \otimes I \otimes X$, tačiau nieko nedaroma radus $|00\rangle \rightarrow I \otimes I \otimes I$. Tai atliekama naudojant parodytus klasiškai kontroluojamus loginius vartus, pritaikytus pažeistam kubitui. Šio algoritmo pabaigoje, pagal skaičiavimų paskirtį, galima atlikti dekodavimo žingsnį $|\psi\rangle_L \rightarrow |\psi\rangle \otimes |00\rangle$, kuris panaikina loginį kubitą ir palieka vieną fizinį kubitą $|\psi\rangle$ būsenoje. Tai yra atliekama naudojant kodavimo loginių vartų seką atvirkštine tvarka. Toliau panagrinėkime šio 3 kubitų kodo efektyvumą.

Tikiemybė, kad trys pavieniai kubitai, nusiųsti per triukšmingą kvantinį kanalą, neįgaus klaidos, yra $(1-p)^3$. O štai kiekviena iš būsenų, turinčių vieną klaidą, yra randama su tikimybe $p(1-p)^2$, būsenos su dviem klaidomis yra $p^2(1-p)$, ir p^3 su trimis. Sindromą nusakančios būsenos pradeda kartotis atsiradus dviem ir daugiau klaidų. Tačiau dviejų ir trijų klaidų tikimybė yra daug mažesnė, jeigu p yra itin mažas. Bendra tikimybė, kad pateiktas trijų kubitų kodas neveiks, yra visų kubitų dviejų ir trijų klaidų tikimybių suma $3p^2(1-p) + p^3$, ir tai galima palyginti su tikimybe p , kai nėra naudojamas klaidų taisymo kodas. Pavyzdžiui, kai $p = 0.1$, nepataisomos klaidos tikimybė trijų kubitų kode yra 10^2 kartų mažesnė, o kai $p = 0.01$, ji yra 10^4 mažesnė. Šios tikimybės susilygina kai $p = 0.5$, todėl, kaip ir klasikiniame pavyzdyje, trijų kubitų taisymo metodas suteiks pranašumo prieš pavienio kubito siuntimą triukšmingu kanalu, jeigu $p < 0.5$.

9.4 Fazės apvertimo klaidos aptikimas ir taisymas

Bito ir fazės klaidų aptikimas ir taisymas yra glaudžiai susijęs. Surinksime visą informaciją siekdami tai pademonstruoti. IV skyriuje matėme, kad Pauli- Z loginius vartus galima išreikšti dviejų Hadamardo ir Pauli- X vartų sandauga, $Z = HXH$ bei $X = HZH$. Hadamardo vartai, veikdami Pauli- Z bazinius vektorius transformuoja juos į Pauli- X bazinius vektorius: $H|0\rangle = |0_x\rangle$, $H|1\rangle = |1_x\rangle$. O pritaikę šiemis Pauli- Z vartus randame:

$$Z|0_x\rangle = |1_x\rangle, \quad Z|1_x\rangle = |0_x\rangle. \quad (9.21)$$

Kitaip tariant, fazės apvertimo klaida $\{|0\rangle, |1\rangle\}$ baziniuose vektoriuose yra ne kas kita, kaip bito apvertimo klaida $\{|0_x\rangle, |1_x\rangle\}$ baziniuose vektoriuose. Tai reiškia, kad atitinkamai transformavę 3 kubitų bito apvertimo kodą galime jį panaudoti norėdami aptikti ir taisyti fazės klaidas.

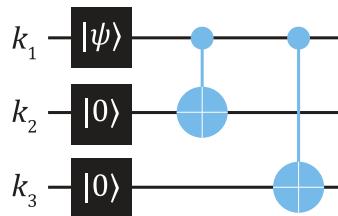
Siūsdami kubitus per triukšmingą kvantinį kanalą, kuriame atsiranda fazės klaidos, kubito būseną $|\psi\rangle$ koduojame dviem žingsniais. Pirmiausia, vėl „patrigubiname” bazinius vektorius:

$$|\psi\rangle = a|0\rangle + b|1\rangle \rightarrow a|000\rangle + b|111\rangle. \quad (9.22)$$

Tolesniame žingsnyje pritaikome Hadamardo transformacijas kiekvienam iš trijų kubitų:

$$|\psi\rangle_L = H^{\otimes 3}(a|000\rangle + b|111\rangle) = a|0_x0_x0_x\rangle + b|1_x1_x1_x\rangle. \quad (9.23)$$

9.3 pav. pateikiame grandinę, iliustruojančią šio loginio kubito paruošimą.



9.3 pav.: Loginė grandinė, paruošianti loginį kubitą, skirtą taisyti fazės klaidai

Atsiradusi fazės klaida, pavyzdžiui, pirmajame kodo kubite, pakeis loginę būseną taip:

$$|\psi\rangle_L \rightarrow a|1_x0_x0_x\rangle + b|0_x1_x1_x\rangle. \quad (9.24)$$

Norėdami pritaikyti bito apvertimo klaidos aptikimo ir taisymo algoritmą, turime pirmiausiai šią būseną transformuoti atgal į $\{|0\rangle, |1\rangle\}$ bazinius vektorius. Atlikę Hadamardo transformacijas pažeistai būsenai randame:

$$H^{\otimes 3}(a|1_x0_x0_x\rangle + b|0_x1_x1_x\rangle) = a|100\rangle + b|011\rangle. \quad (9.25)$$

Akivaizdu, kad šis fazės klaidų taisymo kodas turi identiškas charakteristikas bito apvertimo kodui, tad anksčiau pateikta analizė tinkta ir čia.

9.5 Tolydžiosios klaidos

Bendroje situacijoje 1 kubito klaidos gali kisti tolydžiai ir yra nusakomos Blocho vektoriaus posūkio operatoriai $R_x(\theta)$, $R_y(\theta)$ ir $R_z(\theta)$ aplink x , y ir z ašis kampu θ . Pritaikykime tolydžiąją

bito apvertimo klaidą $R_x(\theta) \otimes I \otimes I$ pirmajam kubitui $|\psi\rangle_L$ būsenoje 3 kubitų bito apvertimo taisymo kode:

$$R_x(\theta) \otimes I \otimes I |\psi\rangle_L = \cos(\theta/2)(a|000\rangle + b|111\rangle) - i \sin(\theta/2)(a|100\rangle + b|011\rangle). \quad (9.26)$$

Matome loginio kubito būseną, kuri yra klaidos nepaveiktos ir paveiktos būsenos superpozicijoje. Kad aptiktume klaidą, vėl galime taikyti $Z \otimes Z \otimes I$ ir $Z \otimes I \otimes Z$ operatorių matavimus naudodami identišką grandinę su dviem ancila kubitais. Tai atlikę randame:

$$\cos(\theta/2)(a|000\rangle + b|111\rangle) \otimes |00\rangle - i \sin(\theta/2)(a|100\rangle + b|011\rangle) \otimes |11\rangle. \quad (9.27)$$

Ancilų nusakyta sindromo būsena $|00\rangle$ yra supinta su loginio kubito būsena, kuriai neįvyko klaida. O štai $|11\rangle$ sindromo būsena yra supinta su kubitų būsena, kuriai įvyko bito apvertimo klaida. Kaip ir anksčiau, darome prielaidą, kad fizinis kubitas paveikiamas $R_x(\theta)$ klaidos nepriklausomai nuo kitų kubitų su tikimybe p . Atlikę ancilų kubitų būsenų matavimą galime rasti sindromą $|00\rangle$ su tikimybe $p \cos^2(\theta/2)$, dėl kurio loginio kubito būsena lieka nepažeista $|\psi\rangle_L = a|000\rangle + b|111\rangle$. Sindromo būsena $|11\rangle$ randama su tikimybe $p \sin^2(\theta/2)$, o po matavimo loginio kubito būsena tampa $X \otimes I \otimes I |\psi\rangle_L = a|100\rangle + b|011\rangle$; tai nusako bito apvertimo klaidą.

Klaidų tolydumą nusakantis faktorius θ atsiranda šalia amplitudžių ir paveikia tik tikimybes rasti pažeistą ir nepažeistą būsenas. Šiuo atveju, tikimybė rasti pažeistą būseną tampa $p \rightarrow p \sin^2(\theta/2)$. Atlikus ancilų kubitų matavimus θ faktorius iškrinta, o bendra loginių kubitų ir ancilų būsena lieka faktorizuojamoji. Radus $|a_1a_2\rangle = |00\rangle$ sindromą imtis veiksmų nereikia, o štai radus $|a_1a_2\rangle = |11\rangle$ pritaikomi $X \otimes I \otimes I$ loginiai vartai, ištaisantys klaidą pirmame kubite $(X \otimes I \otimes I)(X \otimes I \otimes I)|\psi\rangle_L \otimes |11\rangle = |\psi\rangle_L \otimes |11\rangle$.

Tai savo ruožtu demonstruoja itin svarbų teiginį, kad kvantinės klaidos gali būti efektyviai diskretizuojamos, nors pačios būsenos gali kisti ir tolydžiai. Tie patys metodai, taikomi taisyti diskrečiosioms kvantinėms klaidoms, kuriai yra tik trys rūšys $\{X, Y, Z\}$, kartu ištaiso ir tolydžias klaidas.

9.6 Bendrieji klaidų taisymo principai

Šiame poskyryje pateikiame bendruosius principus, kurie įvardija, kokias klaidas kodai gali ištisinti, ir bendrą taisymo proceso principą. Vadinkime kubitų klaidas nusakantį unitarųjį operatorių E . Klaidų operatorius, veikiantis n -kubitų registrą, yra sudarytas iš n tenzorinių Pauli operatorių sandaugų sekos $E \in \{I, X, Y, Z\}^{\otimes n}$. Kubitų bazinius vektorius koduojančias būsenas vadinkime $|i\rangle_L$. Pirma būtina salyga, norint užtikrinti klaidų taisymą, reikalauja, kad klaidų operatoriai, veikiantys skirtinges kodų būsenas, pakeistų jas į kitas, ortogonalias, klaidų būsenas. Tai galime užrašyti glauštai:

$$\langle i | E_a^\dagger E_b | j \rangle_L = 0, \text{ jeigu } i \neq j. \quad (9.28)$$

Jeigu skirtinges kodų būsenos $|i\rangle_L$ ir $|j\rangle_L$ klaidų operatoriais yra pakeičiamos į klaidų būsenas, kurios néra ortogonalios kodų būsenoms ir kitoms klaidų būsenoms, jos nebegali būti patikimai atskirtos, ir todėl taisymas tampa neįmanomas. Šiuo atžvilgiu, klaidų operatoriai E transformuoja kodo būsenas iš kodo erdvės į vieną iš kodo erdvei ortogonalų klaidų būsenų poerdvių.

Antroji salyga nusako, kad atliekant matavimą sindromui nustatyti gauta informacija negali atskleisti koduojamos kubitų būsenos. Informacijos atskleidimas bendrai paveikia kvantines būsenas ir atsitiktiniu būdu jas negrąžinamai pakeičia. Šią salygą galima glauštai užrašyti:

$$\langle i | E_a^\dagger E_b | i \rangle_L = c_{ab}. \quad (9.29)$$

Vertė c_{ab} negali priklausyti nuo būsenos $|i\rangle_L$, nes tai atskleistų apie ją informaciją. Matavimo rezultatas gali priklausyti tik nuo klaidų operatorių E_a ir E_b . Kubito apvertimo ir fazės klaidų aptikimo stadijoje matėme, kad $Z \otimes Z \otimes I$ ir $Z \otimes I \otimes Z$ operatorių matavimai leidžia identifikuoti kubitų būseną lyginumą, tačiau neatskleidžia informacijos, kokios tai būsenos, tai yra jų amplitudžių. Jeigu minėtos dvi sąlygos yra užtikrintos, tada E_a ir E_b priklauso ištaisomų klaidų operatorių rinkiniui, $\varepsilon \subseteq \{I, X, Y, Z\}^{\otimes n}$.

Siekdami iliustruoti bendrą klaidų taisymo principą, imkime išplėstinę sistemą, sudarytą iš n kubitų registro $|\psi\rangle$, aplinkos sistemos būseną $|e\rangle$ ir pradinio ancilos kubito būsenoje $|0\rangle$:

$$|\Psi\rangle = |\psi\rangle \otimes |e\rangle \otimes |0\rangle. \quad (9.30)$$

Šioje stadijoje kubitų registras yra veikiamas klaidų operatoriaus E_i ir supinamas su aplinkos būsenomis. Toliau atliekame unitariąją transformaciją U , supinančią ir ancilos kubitą:

$$U|\Psi\rangle = \sum_{E_i \in \varepsilon} E_i |\psi\rangle \otimes |e_i\rangle \otimes |a_i\rangle. \quad (9.31)$$

Siekiant identifikuoti ir atstatyti kodą, yra atliekama $U|\Psi\rangle$ būsenos projekcija į vieną iš ortogonalinių klaidų poerdvii. Tai matėme bito ir fazės apvertimo klaidų taisymę, kuriuose ancilos kubitai yra supinami su klaidų būsenomis ir atliekamas projekcinis matavimas. Dėl tokios priežasties superpozicija yra sugriaunama ir ši būsena, su tam tikra tikimybe, pasikeičia į vieną iš galimų:

$$E_i |\psi\rangle \otimes |e_i\rangle \otimes |a_i\rangle. \quad (9.32)$$

Atkreipiame dėmesį, kad šioje stadijoje kubitų registro būsena $E_i |\psi\rangle$ yra nebesupinta nei su aplinkos, nei su ancilų sistemų būsenomis. Tad norint atstatyti koduotą būseną yra pritaikomas atvirkštinis klaidos operatorius E_i^\dagger , nes $E_i^\dagger E_i |\psi\rangle = |\psi\rangle$.

9.7 Kvantinė Hamingo riba

Siekdami ištaisyti vieno tipo klaidą 1 kubito registre naudojome 3 kubitų kodą. Kyla natūralus klausimas, ar galima rasti kriterijų, pasakantį, kiek minimaliai reikia fizinių kubitų siekiant ištaisyti n kubitų dydžio registrą, kuriame atsiranda daugiausia t skaičius klaidų. Tai iš principo leistų ieškoti optimalaus kodo dydžio, neeikvojančio papildomų fizinių kubitų.

Kvantinė Hamingo riba (angl. *quantum Hamming bound*) suteikia būdą tai įvertinti klasei kodų, kurie yra lietuviškai vadinti **neišsigimusiais** (angl. *non-degenerate*). Neišsigimusiuose koduose su kiekvienu skirtingu sindromu galima susieti unikalų kubitą, kuriame įvyko klaida, ir nusakyti klaidos tipą. Klasikiniuose koduose visos klaidos yra neišsigimusios, išsigimusios atsiranda išskirtinai kvantinėje terpeje. Platesnės analizės, apimančios išsigimusius kodus, šios knygos rašymo metu dar néra, ir lieka išsiaiškinti, ar išsigimusieji kodai gali būti efektyvesni negu neišsigimusieji ir įveikti kvantinę Hamingo ribą.

Siekdami išvesti Hamingo ribą pirmiausia įvertinsime, kiek dominančio kodo dydyje egzistuoja skirtingu klaidų. Pirmiausiai, egzistuoja $\binom{n}{j}$ skirtingu konfigūracijų, nusakančių, kuriuose j skaičiuje kubitų iš esamų n kubitų įvyko klaida. Čia $\binom{n}{j} = n! / j!(n-j)!$ yra kombinatorinis skaičius. Kiekvienam kubitui yra galimos trys skirtinges klaidos, nusakomos $\{X, Y, Z\}$ operatoriais, tad skaičius $N(t)$ klaidų iš viso yra:

$$N(t) = \sum_{j=0}^t 3^j \binom{n}{j}. \quad (9.33)$$

Suma indeksuojama skaičiais j ir kinta nuo 0 (nėra klaidų) iki didžiausio skaičiaus klaidų t , kai $t < n$. Pavyzdžiuui, $\binom{3}{2}$ nusako, kad trijuose kubituose galimos dvi klaidos. Dviejų klaidų išsidėstymo skaičius tarp trijų kubitų, neskaičiuojant skirtingo tipo klaidų, yra iš viso trys ($k_1 - k_2$, $k_1 - k_3$, $k_2 - k_3$). Atsižvelgdami į tai, kad kiekviename iš kubitų gali būti viena iš trijų skirtingų klaidų, randame $N(2) = 3^2 \times 3 = 27$ galimų klaidų konfigūracijų.

Sakykime, kad turime k skaičių loginių kubitų, kurie yra koduojami naudojant n skaičių fizinių kubitų. Loginiai kubitai dengia 2^k dimensijų vektorių erdvę, ir visos jos būsenos gali būti išreikštos 2^k baziniais kodo vektoriais $|i\rangle_L$. Pavyzdžiuui, vieno loginio kubito Hilberto erdvė yra 2 dimensijų ir dengiama $|0\rangle_L$ bei $|1\rangle_L$. Ankstesniame poskyryje minėjome, kad taisytinų klaidų pirmoji salyga reikalauja, jog klaidų operatoriai, veikdami kodo būsenas, pakeistų jas į viena kitai ortogonalias klaidų būsenas. Taip pat kodo būsenos, paveiktos skirtingo klaidų operatoriaus, E_a , turi būti ortogonalios toms paveiktoms E_b . Tad kiekvienai skirtingai klaidai turi būti priskiriamas 2^k dimensijų poerdvvis. Kadangi yra $N(t)$ skaičius skirtingų klaidų konfigūracijų (ir todėl toks pat skaičius klaidų operatorių), erdvės dimensija, talpinanti visas ortogonaliasias klaidų būsenas (iskaitant klaidų nepažeistą būseną), turi būti bent $N(t)2^k$ dydžio. Tai nusako minimalų n fizinių kubitų, koduojančių k loginius kubitus, dimensijos dydį 2^n . Hamingo riba randama:

$$\sum_{j=0}^t 3^j \binom{n}{j} 2^k \leq 2^n. \quad (9.34)$$

Pavyzdžiuui, k loginių kubitų, kuriems leidžiama tik viena ($t = 1$) bendrojo tipo kvantinė klaida, ši nelygybė yra:

$$(1 + 3n)2^k \leq 2^n. \quad (9.35)$$

Galima patikrinti, kiek reikia minimaliai fizinių kubitų siekiant koduoti 1 loginį kubitą, kuriam leidžiama viena bendrojo tipo klaida. Šiuo atveju $k = t = 1$ ir Hamingo riba nusako, kad $n = 5$ suteikia lygybę. Tad neegzistuoja neišsigimusis kodas, kuris, koduodamas vieną loginį kubitą mažiau nei penkiuose fiziniuose kubituose, galėtų apsaugoti nuo visų galimų klaidų viename kubite.

Kadangi ne visi kodai yra neišsigimusieji, kvantinę Hamingo ribą galima taikyti veikiau kaip pirmąjį įvertinimą. Egzistuoja ir kitų kvantinių kodų ribų apibrėžimų, pavyzdžiuui, kvantinė Singlono riba (angl. *quantum Singleton bound*) tinkia abejoms kodų klasėms. Jos įrodymo čia nepateiksime, tačiau nelygybė yra:

$$n - k \geq 4t. \quad (9.36)$$

Simboliai n ir k nusako fizinių ir loginių kubitų skaičių, atitinkamai, taip pat didžiausią klaidų paveiktų kubitų skaičių t . Matome, kad mažiausias kubitų skaičius, kai $k = t = 1$, yra $n = 5$ ir atitinka Hamingo ribą neišsigimusiemis kodams. Šis optimalus kodas gali būti užrašomas $[n, k, t] = [5, 1, 1]$.

9.8 Šoro 9 kubitų kodas

Šoro 9 kubitų kodas yra vienas iš pirmųjų atrastų kodų, leidžiančių ištaisyti bet kuriame iš 9 kubitų vieną bendriausio tipo kvantinę klaidą. Taikant viršuje minėtą susitarimą, tai formaliai yra $[n, k, t] = [9, 1, 1]$ kodas. Šoro kodas naudoja dviejų lygių konkatenaciją. Pirmajame žingsnyje įvykdomas 3 kubitų kodavimas, naudojamas apsaugoti kubitus nuo fazės apvertimo klaidos, $|0\rangle \rightarrow |0_x 0_x 0_x\rangle$, $|1\rangle \rightarrow |1_x 1_x 1_x\rangle$. Antrajame žingsnyje kiekvienas iš šių 3 kubitų yra

toliau koduojamas dar 3 kubitais, apsaugant juos nuo bito apvertimo klaidos:

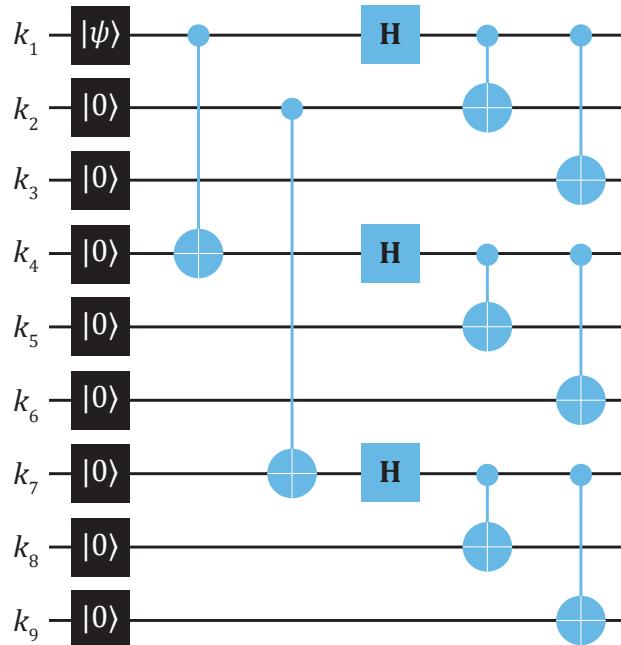
$$|0_x\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \rightarrow \frac{|000\rangle + |111\rangle}{\sqrt{2}}; \quad |1_x\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \rightarrow \frac{|000\rangle - |111\rangle}{\sqrt{2}}. \quad (9.37)$$

Bendroje loginio kubito būsenoje $|\psi\rangle_L = a|0\rangle_L + b|1\rangle_L$ baziniai vektoriai yra:

$$|0\rangle_L = \frac{1}{\sqrt{8}}(|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \\ \equiv |+\rangle \otimes |+\rangle \otimes |+\rangle; \quad (9.38)$$

$$|1\rangle_L = \frac{1}{\sqrt{8}}(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \\ \equiv |-\rangle \otimes |-\rangle \otimes |-\rangle. \quad (9.39)$$

Kubitus sugrupavome į tris blokus ir, siekdamai supaprastinti simboliką, blokus pavadinome atitinkamai $|+\rangle$ ir $|-\rangle$. Grandinė atliekanti kodo paruošimą pateikta 9.4 pav.



9.4 pav.: Šoro 9 kubitų kodo loginio kubito paruošimas

Kaip ir 3 kubitų kode, aptikti atsirandančioms bito apvertimo klaidoms pasitelkiami nelokaliūs matavimai. Sakykime, kad atsiranda bito apvertimo klaida pirmojo bloko pirmame kubite, $|+\rangle \rightarrow (|100\rangle + |011\rangle)$. Klaidos aptikimui pirmajame bloke taikome $Z_1 \otimes Z_2$ ir $Z_1 \otimes Z_3$ matavimus pasitelkdami du ancilų kubitus. Čia, supaprastindami simboliką, praleidome vienetinių operatorių I rašymą, tad $Z_1 \otimes Z_2 = Z \otimes Z \otimes I \otimes I \otimes I \otimes I \otimes I \otimes I$ ir $Z_1 \otimes Z_3 = Z \otimes I \otimes Z \otimes I \otimes I \otimes I \otimes I \otimes I \otimes I$. Antrajame ir trečiajame bloke klaidos aptikimui analogiškai naudojamos $Z_4 \otimes Z_5$ ir $Z_4 \otimes Z_6$, taip pat $Z_7 \otimes Z_8$ ir $Z_7 \otimes Z_9$ operatorių poros, kurių vertės įrašomos į dar dvi poras ancilų kubitų. Tai leidžia unikaliai nustatyti, kuriame iš 9 kubitų įvyko bito apvertimo klaida, ir ją ištaisyti pritaikius atitinkamam kubitui Pauli- X loginius vartus.

Jeigu atsiranda fazės klaida, pavyzdžiui, pirmojo bloko viename iš kubitų, šio bloko būsena pakinta taip:

$$|0\rangle_L : |000\rangle + |111\rangle \rightarrow |000\rangle - |111\rangle ; \quad (9.40)$$

$$|1\rangle_L : |000\rangle - |111\rangle \rightarrow |000\rangle + |111\rangle . \quad (9.41)$$

Atkreipiame dėmesį, kad nesvarbu, kuris iš trijų kubitų bloke patiria fazės klaidą, to bloko būsena pasikeičia lygiai taip pat. Tad identifikuoti, kuris kubitas patyrė fazės klaidą, neįmanoma, ir dėl to Šoro kodas yra formaliai išsigimės. Įvykus kladai pradinė būsena $|\psi\rangle_L$ pasikeičia taip:

$$|\psi\rangle_L \rightarrow a|-\rangle \otimes |+\rangle \otimes |+\rangle + b|+\rangle \otimes |-\rangle \otimes |-\rangle . \quad (9.42)$$

Vietoj pavienių kubitų bloke palyginimo, kaip daroma aptinkant bito apvertimo kladą, fazės apvertimo klados aptikimui tarpusavyje palyginami patys blokai. Tam atlikti pasitelkiame irgi du ancilų kubitus, į kuriuos užrašomi, šiuo atveju, $X_1 \otimes X_2 \otimes X_3 \otimes X_4 \otimes X_5 \otimes X_6$ ir $X_1 \otimes X_2 \otimes X_3 \otimes X_7 \otimes X_8 \otimes X_9$ operatorių matavimų rezultatai (praleidžiame vienetinių operatorių I rašymą likusiems kubitams). Kad tai būtų lengviau suprasti, atkreipiame dėmesį, jog Šoro kodo būsenos $|+\rangle \otimes |+\rangle \otimes |+\rangle$ ir $|-\rangle \otimes |-\rangle \otimes |-\rangle$ yra šių operatorių tikriniai vektoriai su tikrinėmis vertėmis $+1$. Galime išskaidyti šias tikrines vertes į dviejų pavienių blokų tikrinių verčių sandaugas. Naudodami pirmąjį bloką kaip pavyzdį matome, kad individualių blokų būsenos $|+\rangle$ ir $|-\rangle$ yra trijų Pauli- X tenzorių sandaugos operatorių tikriniai vektoriai su tikrinėmis vertėmis $\lambda = 1$ ir $\lambda = -1$, atitinkamai:

$$X_1 \otimes X_2 \otimes X_3 (|000\rangle + |111\rangle) = |000\rangle + |111\rangle = |+\rangle ; \quad (9.43)$$

$$X_1 \otimes X_2 \otimes X_3 (|000\rangle - |111\rangle) = -(|000\rangle - |111\rangle) = -|-\rangle . \quad (9.44)$$

Tad bet kurių dviejų blokų būsenų, nepaveiktų klados, tikrinių verčių sandauga yra visada $+1$.

Kladų aptikimo procese, vadindami $X_1 \otimes X_2 \otimes X_3$, $X_4 \otimes X_5 \otimes X_6$ ir $X_7 \otimes X_8 \otimes X_9$ operatorių, veikiančių kiekvieną iš trijų kubitų blokų, tikrines vertes atitinkamai λ_1 , λ_2 , λ_3 , užrašysime jų porų sandaugas $\lambda_1\lambda_2$ ir $\lambda_1\lambda_3$ į du ancilų kubitus. Pirmasis minėtas operatorius patikrina pirmą ir antrą blokus, o antrasis – pirmą ir trečią blokus. Jeigu viename iš blokų įvyksta fazės klaida, tikrinė vertė pasikeičia į -1 ir todėl tikrinių verčių poros sandauga tampa -1 .

Norėdami realizuoti kladų aptikimo procesą kvantinėje grandinėje, pirmiausiai Pauli- X operatorius perrašysime $X = HZH$. Kad įvykdytume X operatorių matavimus, kiekvienam kubitui atliekame Hadamardo transformacijas $H^{\otimes 9}$ ir Pauli- Z matavimus, pasitelkdami $CNOT$ vartus ir ancilos kubitą. Pirmų dviejų blokų palyginimui matavimas $Z_1 \otimes Z_2 \otimes Z_3 \otimes Z_4 \otimes Z_5 \otimes Z_6$ užrašomas ancilos būsenoje $|a_1\rangle = |k_1 \oplus k_2 \oplus k_3 \otimes k_4 \oplus k_5 \oplus k_6\rangle$ naudojant mod (2) bitų sudėtį. Taip randaime $|a_1\rangle = |0\rangle$, jeigu dviejų blokų būsenos yra vienodos, ir $|a_1\rangle = |1\rangle$, jeigu jos skiriasi. Tai savo ruožtu atspindi $X_1 \otimes X_2 \otimes X_3 \otimes X_4 \otimes X_5 \otimes X_6$ operatoriaus ± 1 tikrines vertes. Tas pats procesas atliekamas su pirmojo ir trečiojo bloko matavimais naudojant $Z_1 \otimes Z_2 \otimes Z_3 \otimes Z_7 \otimes Z_8 \otimes Z_9$. Abiejų matavimų rezultatas užrašomas į dviejų ancilų kubitų būsenas $|a_1a_2\rangle$, o standartinis Pauli- Z matavimas leidžia unikaliai nusakyti, kuriame bloke įvyko fazės klaida. Pavyzdžiui, fazės klaida loginio kubito pirmame bloke bus nusakyta šia bendra būsena:

$$(a|-\rangle \otimes |+\rangle \otimes |+\rangle + b|+\rangle \otimes |-\rangle \otimes |-\rangle) \otimes |11\rangle . \quad (9.45)$$

Fazės klaida pirmajame bloke ištaisoma pritaikius $Z_1 \otimes Z_2 \otimes Z_3$ loginius vartus. Galiausiai atliekami dar vieni Hadamardo vartai $H^{\otimes 9}$ visiems kubitams siekiant atstatyti būsenas į $|+\rangle$ ir $|-\rangle$ kodų formą, o ancilos grąžinamos į $|00\rangle$.

Šoro kodas gali ištisinti bendriausio tipo klaidą. Tai išplaukia iš šio skyriaus 9.2 poskyryje pateikto įrodymo, kad visas klaidas galima išreikšti Pauli operatorių $\{I, X, Y, Z\}$ ir jų tensorinių sandaugų tiesinėmis kombinacijomis. Todėl gebant taisinti X (bito apvertimo) ir Z (fazės apvertimo) klaidas, Šoro kode automatiškai galima taisinti ir šių dviejų klaidų kombinaciją, $XZ = -iY$. Šoro kodas bus efektyvus, jeigu atsiranda ne daugiau nei viena klaida 9 kubituose. Tikimybė, kad siunčiant loginį kubitą per triukšmingą kanalą nė vienas fizinis kubitas nebus pažeistas, yra $(1-p)^9$. Čia p nusako tikimybę, kad fizinis kubitas patirs klaidą. Tikimybė, kad Šoro kode įvyks viena klaida, yra $9p(1-p)^8$, ir kodas leidžia ją ištisinti. Tad dvi ar daugiau klaidų atsitiks su tikimybe $1 - 9p(1-p)^8 - (1-p)^9 \approx 36p^2$, jeigu p yra itin mažas. Palyginus su pavieniu fizinio kubito siuntimui, nepataisomų klaidų tikimybės susilygina, kai $p \approx 0.032$, ir Šoro kodas suteikia didžianti pranašumą toliau mažėjant p .

9.9 Kodų stabilizatoriai

Iki šiol analizavome klaidų taisymo kodus pradėdami nuo kodų būsenų. Idealiai, norėtume turėti sisteminę receptą, leidžiantį sugeneruoti kodą su mus dominančiomis savybėmis – fizinių ir loginių kubitų skaičiumi, ištaisomų klaidų skaičiumi ir juose naudojamais operatoriai aptinti ir taisinti klaidas. **Kodų stabilizatoriai** (angl. *code stabilizers*) formalizmas atlieka šią funkciją ir yra plačiai taikomas kvantinėje kompiuterijoje. Egzistuoja taisyklės, kaip stabilizatorių kodams konstruoti kvantines grandines, paruošiančias kodų būsenas, aptinkančias ir taisančias klaidas, taip pat leidžiančias lengviau formuliuoti klaidoms atsparius skaičiavimus. Kodų stabilizatorių algoritmai gali būti realizuojami vien tik Klifordo grupės loginiai vartais $\{H, S, cX\}$, todėl jų veikimą galima efektyviai modeliuoti ir testuoti klasikiniais kompiuteriais. Toliau glaustai pristatome šios plačios kodų grupės įvadinius principus.

Kodų stabilizatorių formalizmas yra pagrįstas ne kvantinių būsenų, o unitariųjų operatorių analize pasitelkiant **grupių teoriją** (angl. *group theory*). Sakoma, kad būsena $|\psi\rangle$ yra stabilizuojama operatoriaus K , jeigu ji yra šio operatoriaus tikrinis vektorius su $+1$ tikrine verte:

$$K|\psi\rangle = |\psi\rangle. \quad (9.46)$$

Pavyzdžiui, 1 kubito būsena $|0\rangle$ yra stabilizuojama Pauli-Z operatoriaus, nes $Z|0\rangle = |0\rangle$. Šoro 9 kubitų kodas yra taip pat stabilizatorių klasės kodas. Aštuoni operatoriai $Z_1 \otimes Z_2, Z_1 \otimes Z_3, Z_4 \otimes Z_5, Z_4 \otimes Z_6, Z_7 \otimes Z_8, Z_7 \otimes Z_9$ bei $X_1 \otimes X_2 \otimes X_3 \otimes X_4 \otimes X_5 \otimes X_6$ ir $X_1 \otimes X_2 \otimes X_3 \otimes X_7 \otimes X_8 \otimes X_9$ yra jo stabilizatoriai.

Apžvelgdami operatorių savybes stabilizuojančių n -kubitų būsenas, pirmiausiai apibūdiname 1 kubito Pauli grupę \mathcal{P} , kuri yra sudaryta iš Pauli operatorių:

$$\mathcal{P} = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}. \quad (9.47)$$

Pauli elementų rinkinys (Pauli operatoriai kartu su juos dauginančiais skaičiais $\{\pm 1, \pm i\}$) formuoja grupę operatorių sandaugos operacijų atžvilgiu. Pauli grupė pratesiama n kubitų sistemoms naudojant jos elementų n tensorių sandaugą, $\mathcal{P}^{\otimes n}$. Galima parodyti, kad visi Pauli grupės elementai yra tarpusavyje arba komutatyvūs, arba antikomutatyvūs. Primename, kad du komutatyvūs operatoriai A ir B tenkina $AB = BA$, ir tai standartiskai užrašoma $[A, B] = 0$. Tačiau A ir B yra antikomutatyvūs, jeigu sandaugoje sukeitus jų vietas atsiranda minuso ženklas, $AB = -BA$. Tai išreiškiama $\{A, B\} = AB + BA = 0$. N kubitų stabilizatorių būsena $|\psi\rangle$ yra nusakoma Pauli grupės $\mathcal{P}^{\otimes n}$ operatorių pogrupė $\mathcal{G}^{\otimes n}$, kurios visi elementai, vadinkime juos K_i , yra tarpusavyje komutatyvūs. Operatorių pogrupės $\mathcal{G}^{\otimes n}$ savybes glaustai užrašyti taip:

$$\mathcal{G}^{\otimes n} = \{K_i|\psi\rangle = |\psi\rangle, [K_i, K_j] = 0, \forall (i, j)\} \subset \mathcal{P}^{\otimes n}. \quad (9.48)$$

Stabilizatorių tarpusavio komutatyvumas užtikrina, kad ir jų sandauga $K_1 K_2 K_3 \dots$ taip pat stabilizuoją $|\psi\rangle$. Praktiškai taip pat reikalaudame, kad stabilizatorių rinkinyje visi operatoriai būtų tiesiskai nepriklausomi – negali būti išreikšti kitų rinkinio stabilizatorių sandauga.

Stabilizatoriai K_i efektyviai užfiksuoją dalį n kubitų 2^n dimensijų vektorių erdvės, kitaip tariant, jos poerdvį, kuriame atliekamas būsenų kodavimas. Imkime 2 kubitų pavyzdį, kai naudojamas Belo bazinių vektorių rinkinys $\{|\chi^+\rangle, |\chi^-\rangle, |\eta^+\rangle, |\eta^-\rangle\}$. Galima lengvai patikrinti, kad operatorius $X \otimes X$ unikaliai stabilizuoją $|\chi^+\rangle$ ir $|\eta^+\rangle$ būsenas. Tad, jeigu naudosime poerdvį, stabilizuotą $X \otimes X$, šias dvi 2 kubitų ortogonaliasias būsenas galime naudoti formuodami vieną loginį kubitą:

$$|0\rangle_L = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad \text{ir} \quad |1\rangle_L = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle). \quad (9.49)$$

Taip 4 dimensijų erdvę sumažinome iki 2 dimensijų, kuri musako vieno kubito būsenas. Šioje erdvėje operatorius $Z \otimes Z$ leidžia atskirti $|0\rangle_L$ ir $|1\rangle_L$ vieną nuo kitos, nes $Z \otimes Z|0\rangle_L = |0\rangle_L$ ir $Z \otimes Z|1\rangle_L = -|1\rangle_L$. Jis atlieka loginio Pauli-Z rolę, žymimą su brūkšniu, $\bar{Z} = Z \otimes Z$. Žinoma, galėtume pasirinkti ir $Z \otimes Z$ operatorių, stabilizuojantį $|\chi^+\rangle$ ir $|\chi^-\rangle$ būsenas, ir naudoti jas formuodami loginį kubitą. Tada loginis Pauli-Z būtų $\bar{Z} = X \otimes X$ ir leistų atskirti šio loginio kubito $|0\rangle_L$ ir $|1\rangle_L$ būsenas. Bendrai, jeigu n kubitų koduoja k loginių kubitu, tada yra k loginių Pauli-Z, $n - k$ stabilizatorių, o loginių kubito vektorių erdvės dimensija 2^{n-k} . Toks stabilizatorių kodas yra glaustai indikuojamas skliausteliuose $[n, k]$.

Kiekvienas stabilizatorius K_i yra ermitinis ir tuo pačiu unitarasis operatorius, todėl gali turėti dvi tikrines vertes, $\lambda = 1$ ir $\lambda = -1$. Klaidų operatoriai $\{E_i\}$ taip pat priklauso Pauli grupės operatoriams, $\mathcal{P}^{\otimes n}$. Todėl bet kuris E gali būti komutatyvus arba antikomutatyvus su tam tikru stabilizatoriumi K dominančioje n kubitų erdvėje. Jeigu E ir K yra komutatyvieji, tada:

$$KE|\psi\rangle = EK|\psi\rangle = E|\psi\rangle. \quad (9.50)$$

Tad klaidos operatorius išsaugo K stabilizatoriaus $+1$ tikrinę vertę. Tačiau, jeigu E ir K yra antikomutatyvieji:

$$KE|\psi\rangle = -EK|\psi\rangle = -E|\psi\rangle. \quad (9.51)$$

Tai galime interpretuoti kaip stabilizatorių K , veikiantį klaidos paveiktą $E|\psi\rangle = |\psi\rangle_E$ kodo būseną, $K|\psi\rangle_E = -|\psi\rangle_E$; tai galiausiai pakeičia stabilizatoriaus tikrinę vertę į -1 . Ši tikrinė vertė gali būti aptikta atlikus stabilizatoriaus matavimą siekiant taisyti klaidas. Kaip matėme Šoro 9 kubitų kode, jo stabilizatorių porų kombinacijų matavimas ir rastų tikrinų verčių kombinacijos $\{\pm 1, \pm 1\}$ leidžia identifikuoti kladą. Mat dalis kodo stabilizatorių yra antikomutatyvieji su specifiniais klaidų operatoriais, ir jų unikali kombinacija leidžia nustatyti, kokie klaidų operatoriai veikė kodo būsenas.

Formaliai h stabilizatorių turi 2^h skirtingas $\{\lambda_i, \lambda_j\}$ kombinacijas ir todėl gali identifikuoti tokį skaičių skirtingų kladų būsenų (išskaitant nepažeistą būseną). Pavyzdžiui, 1 loginio kubito, apsaugoto nuo vienos bendriausios klaidos 5 kubitų kode $[5, 1, 1]$, Hamingo riba yra $1 + 3 \cdot 5 \geq 2^4$. Keturi stabilizatoriai identifikuoja $2^4 = 16$ ortogonalius 2 dimensijų kladų būsenų poerdvius ir todėl suteikia lygybę Hamingo riboje. Šio optimalaus 5 kubitų kodo stabilizatoriai yra:

$$K_1 = X \otimes Z \otimes Z \otimes X \otimes I; \quad (9.52)$$

$$K_2 = I \otimes X \otimes Z \otimes Z \otimes X; \quad (9.53)$$

$$K_3 = X \otimes I \otimes X \otimes Z \otimes Z; \quad (9.54)$$

$$K_4 = Z \otimes X \otimes I \otimes X \otimes Z. \quad (9.55)$$

Atkreipiame dėmesį, kad cikliškai pakeistas operatorius $K_5 = Z \otimes Z \otimes X \otimes I \otimes X$ nėra tiesiskai nepriklausomas, nes gali būti išreikštas sandauga, $K_5 = K_1 K_2 K_3 K_4$. Penktasis operatorius, $\bar{Z} = Z \otimes Z \otimes Z \otimes Z$, yra komutatyvus su keturiais šio kodo stabilizatoriais ir atlieka loginio Pauli- Z vaidmenį. Taip pat yra apibūdinamas ir loginis Pauli- X operatorius, $\bar{X} = X \otimes X \otimes X \otimes X \otimes X$, kuris konvertuoja logines būsenas vieną tarp kitos $\bar{X}|0\rangle_L = |1\rangle_L$, $\bar{X}|1\rangle_L = |0\rangle_L$.

Toliau panagrinėkime, kaip sugeneruoti kodo būsenas iš pateiktų stabilizatorių rinkinio. Kodo stabilizatoriai bei loginiai Pauli- Z yra unitariniai ir kartu ermitiniai operatoriai, tad turi dvi tikrines vertes, +1 ir -1. Taikant spektrinę dekompoziciją, tokį n kubitų būsenas veikiantį operatorių K galima užrašyti:

$$K = \sum_{\lambda} \lambda P(\lambda) = P(1) - P(-1). \quad (9.56)$$

Pasitelkdami projekcinius operatorius $P(1)$ ir $P(-1)$ į vektorių poerdvius, asocijuotus su +1 ir -1 tikrinėmis vertėmis, atitinkamai, bei jų pilnumo savybę $P(1) + P(-1) = I$, randame:

$$P(1) = \frac{I + K}{2}, \quad P(-1) = \frac{I - K}{2}. \quad (9.57)$$

Bet kokią n kubitų būseną $|\psi\rangle$ galima išreikšti jos projekcijų į ± 1 stabilizatoriaus K poerdvių būsenas superpozicija:

$$|\psi\rangle = P(1)|\psi\rangle + P(-1)|\psi\rangle. \quad (9.58)$$

Norint paruošti stabilizatorių rinkinio kodo būseną $|0\rangle_L$, užduotis yra atlikti pradinės registro būsenos, standartiskai $|0\rangle^{\otimes n}$, projekciją į h skaičiaus stabilizatorių rinkinio bendrą +1 poerdvį bei +1 loginio Pauli- Z . Jeigu, sakykime, turime tris tarpusavyje komutatyvius ermitinius operatorius K_1, K_2, K_3 , tada projekcinis operatorius į jų bendrą +1 poerdvį, nusakyta $P(1)$, bus atitinkamai trijų projekcinių operatorių sandauga, $P(1) = P_1(1)P_2(1)P_3(1)$. Norint paruošti $|1\rangle_L$, projekcija atliekama į +1 stabilizatorių poerdvį bei loginio Pauli- Z -1 poerdvį.

Optimalaus [5, 1, 1] kodo atveju pradedame nuo $|00000\rangle$ ir $|11111\rangle$ registro būsenų, kurios yra \bar{Z} tikriniai vektoriai su ± 1 tikrinėmis vertėmis, atitinkamai. Kodo būsenas (normuotąsias) randame atlikdami projekciją į bendrą keturių stabilizatorių +1 poerdvį:

$$|0\rangle_L = \frac{1}{4}(I + K_1) \otimes (I + K_2) \otimes (I + K_3) \otimes (I + K_4)|00000\rangle; \quad (9.59)$$

$$|1\rangle_L = \frac{1}{4}(I + K_1) \otimes (I + K_2) \otimes (I + K_3) \otimes (I + K_4)|11111\rangle. \quad (9.60)$$

Hadamardo testo kvantinė grandinė (žr. 6.7.1 poskyri) gali atlikti norimą projekciją. Atkreipiame dėmesį, kad galutinė Hadamardo testo būsena $|\chi\rangle$ turi ieškomają formą:

$$\begin{aligned} |\chi\rangle &= |0\rangle \otimes \left(\frac{I + K}{2} \right) |\psi\rangle + |1\rangle \otimes \left(\frac{I - K}{2} \right) |\psi\rangle \\ &= |0\rangle \otimes P(1)|\psi\rangle + |1\rangle \otimes P(-1)|\psi\rangle. \end{aligned} \quad (9.61)$$

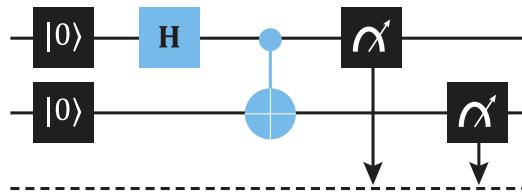
Atlikus ancilos kubito (pirmasis registras) Pauli- Z matavimą, pagal tai, ar bus rasta $|0\rangle$, ar $|1\rangle$ būsena, antrojo kubito būsenai $|\psi\rangle$ bus atlikta projekcija į K operatoriaus +1 arba -1 tikrinių verčių poerdvį, atitinkamai. Šis metodas yra elementariai praplečiamas n kubitų sistemai pasitelkiant daugiau ancilų kubitų, o K gali nusakyti kodų stabilizatorių rinkinio bendrąjį matavimą. Tad jeigu randama +1 ancilos tikrinė vertė, pradinė būsena yra konvertuojama į norimo stabilizatorių kodo būseną. Radus -1 tikrinę vertę, pasitelkus ancilų matavimus atliekama klaidų taisymo stadijai identiška grandinė, konvertuojanti šią būseną į +1 tikrinės vertės būseną. Taip paruošus, pavyzdžiui, loginį $|0\rangle_L$, kode apibrėžtu loginiu Pauli- X galima konvertuoti $|0\rangle_L$ į $|1\rangle_L$, $\bar{X}|0\rangle_L = |1\rangle_L$.

9.10 Klaidoms atsparus skaičiavimas

Iki šiol pristatytuose klaidų protokoluose darėme prielaidą, kad klaidos atsiranda tik tada, kai nėra atliekami loginiai vartai ar kubitų matavimai, ancilių kubitai patys nepatiria klaidų, ir kad loginiai vartai yra absoliučiai tikslūs. Šios prielaidos nėra realistinės. Klaidoms atsparus skaičiavimas yra kvantinių grandinių dizaino metodologija, kuri kartu su klaidų taisymo algoritmais leidžia sėkmingai įvykdyti skaičiavimus, kai visi skaičiavimo elementai ir atmintyje laikomi kubitai gali patirti klaidas.

Klaidoms atsparus skaičiavimas yra pagristas klaidų sklidimo užkirtimu. Galime išvardyti du pagrindinius šaltinius, kurie leidžia pasklisti klaidoms kvantinėse grandinėse. Pirmasis – tai loginiai vartai. Akivaizdu, kad loginiai vartai, veikiantys tik vieną kubitą, nesugeba leisti klaidoms daugintis ir propaguoti neteisingą informaciją. Tačiau, jeigu 2 kubitų cX loginiuose vartuose kontrolinis kubitas patyrė klaidą, tada ši klaida bus perteikta į adresatinį kubitą ir kaskados principu gali sklisti toliau. Taip pat galimas ir grynaus kvantinio efekto klaidų sklidime dėl 6 skyriuje minėtos fazės atatrankos. Jeigu, prieš atliekant cX loginius vartus, įvyksta fazės klaida adresatiniam kubite, tada ši klaida pernė fazės klaidą ir į kontrolinį kubitą. Antrasis klaidų sklidimo šaltinis yra kubitų būsenų matavimo procesas, kurio rezultatas naudojamas kaip sąlyga pritaikyti loginius vartus kitiems kubitams. Pavyzdžiu, jeigu klaidų nustatymo procese ancilos matavimas patiria klaidą, tada gali būti pritaikomi neteisingi loginiai vartai taisyti loginiam kubitui.

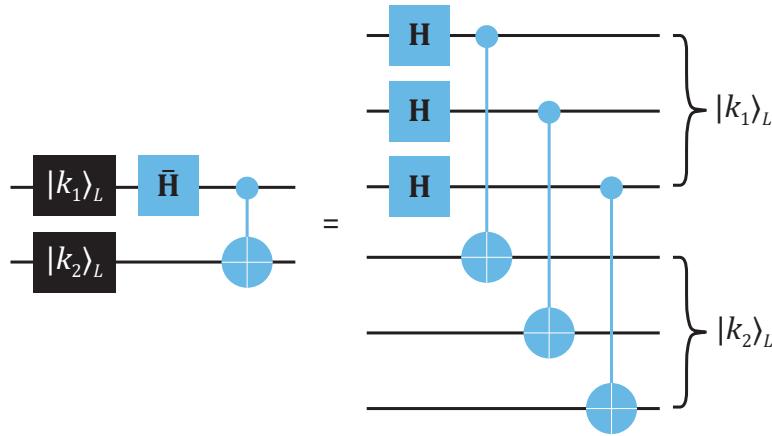
Siekiant užkirsti kelią klaidų dauginimuisi ir sklidimui, kvantinės grandinės loginiai elementai yra pakeičiami klaidoms atspariais loginiais elementais. Imkime kaip pavyzdži 9.5 pav. parodytą kvantinę grandinę, sudarytą iš klaidoms neatsparių loginių elementų, atliekančią 2 kubitų supynimą ir jų matavimus.



9.5 pav.: Klaidoms neatspari loginė grandinė, atliekanti kubitų supynimą ir jų matavimą

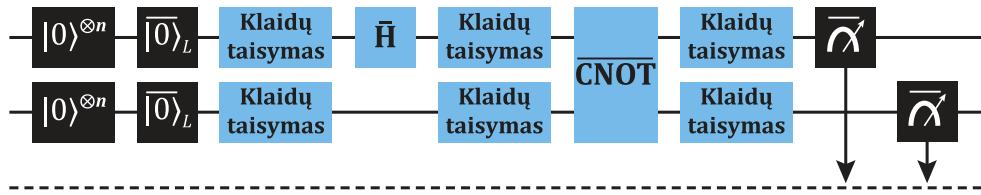
9.6 pav. pateikiame klaidoms atsparius \bar{H} ir \bar{cX} loginius elementus (vėlgi žymime su brūkšniuku) kodui, kuriame k_1 ir k_2 loginiai kubitai koduojami 3 fizinius kubitus, priklausančius k_1 arba k_2 , vadinkime kubitų blokais. Matome, kad jeigu vienam iš fizinių kubitų k_1 bloke pritaikomi netikslūs H loginiai vartai, ši klaida gali paveikti tik šį kubitą k_1 bloke. Loginiai vartai \bar{cX} , veikiantys tarp atskirų kubitų kiekviename bloke, užtikrina, kad H vartų klaida k_1 bloke paveiks daugiausiai vieną papildomą kubitą k_2 bloke. Tą patį galima pasakyti ir apie pačius \bar{cX} , kurių klaidingas atlikimas paveikia tik po vieną kubitą kiekviename bloke. Loginiai vartai, veikiantys tarp fizinių kubitų blokuose nepriklausomai nuo kitų kubitų, yra vadinami **skersiniai** (angl. *transversal gates*). Klaidoms atspari loginė operacija yra formaliai tokia, kurios metu vieno loginio komponento klaidingas veikimas paveikia ne daugiau negu vieną kubitą kiekviename bloke.

Klaidoms atspari kvantinė grandinė pakeičia loginius vartus H ir cX , taip pat matavimų operacijas klaidoms atspariais loginiais elementais. Be to, loginių kubitų paruošimas $|\bar{0}\rangle_L$ turi būti taip pat atliekamas naudojant klaidoms atsparius loginius vartus, o kubitų būsenos yra periodiškai



9.6 pav.: Kvantinė grandinė, dviem 3 kubitų kodo loginiams kubitams atliekanti jų supynimą klaidoms atspariais loginiais vartais

patikrinamos ir prireikus ištaisomos.



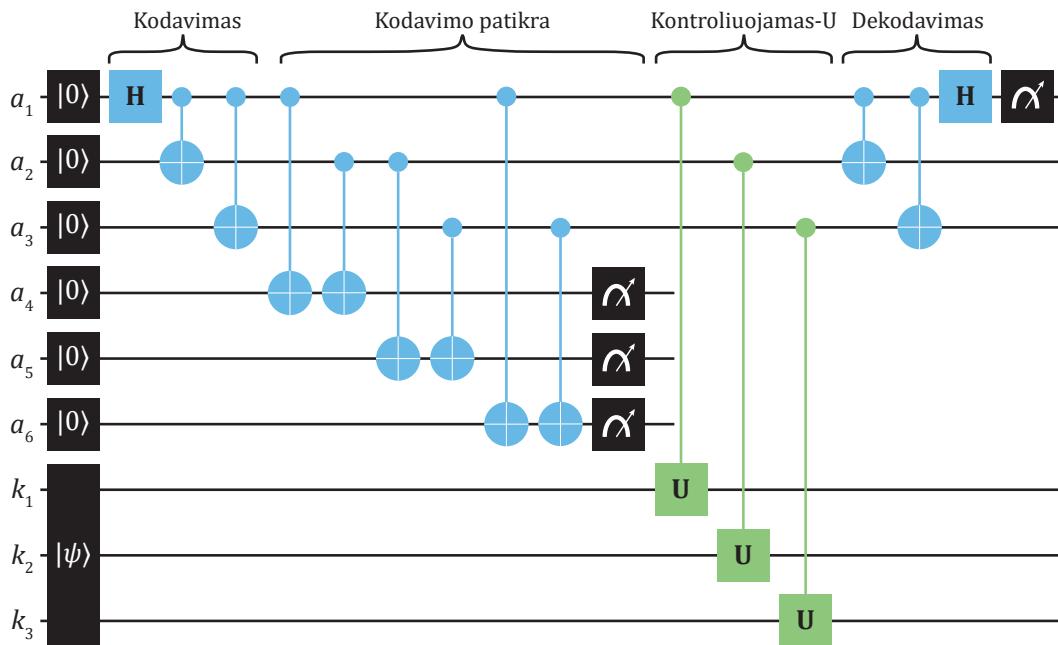
9.7 pav.: Iliustracijoje 9.5 pav. parodytos loginės grandinės klaidoms atspari versija. Du loginiai kubitai čia koduojami n kubitais, toliau patikrinama, ar koduojant neįvyko klaidų, ir atitinkamai atliekamas taisymas. Pirmam loginiam kubitui pritaikomi klaidoms atsparūs \bar{H} ; matomi dar du periodiškai atliekami klaidų nustatymai ir 2 loginių kubitų klaidoms atsparūs \bar{cX} ; galiausiai atliekami klaidoms atsparūs kubitų būsenos matavimai

Siekiant realizuoti klaidoms atsparų būsenų matavimą, vėlgi pasitelkiame Hadamardo testu pagrįstą metodą (žr. 6.7.1 poskyri). Šiuo metodu galima atlikti loginio Pauli- \bar{Z} ar bet kokio kito n kubitų ermitinio operatoriaus U , išreiškiamo skersiniai loginiai vartai, projekcinį matavimą į jo $+1$ ir -1 poerdvius. Iliustracijai, čia jį pritaikome 3 kubitų būsenos $|\psi\rangle$ matavimui. Tai atliekanti kvantinė grandinė pateikta 9.8 pav.

Kiekvienam $|\psi\rangle$ būsenos kubitiui yra pasitelkiamas papildomas ancilos kubitas. Šiuo atveju, trys fiziniai ancilų kubitai pradinėje būsenoje $|a_1a_2a_3\rangle = |000\rangle$ yra paruošiami į superpoziciją $|\varphi\rangle = (|000\rangle + |111\rangle)/\sqrt{2}$, panašią į 3 kubitų loginių kubitą. Kodavimo žingsnis néra klaidoms atsparus, nes naudojami klaidoms neatsparūs H ir cX loginiai vartai. Tačiau po kodo paruošimo kitais trim ancilų kubitais (a_4, a_5, a_6) kodas yra patikrinamas (atliekama kodavimo patikra). Patikra yra pagrįsta jau mums žinomu būsenų lyginimo matavimu. Jeigu kodas teisingas, tada šie ancilų kubitai naudojami tolimesnėms operacijoms. Kitu atveju kodavimas kartojamas iš naujo.

Tolesniame žingsnyje atliekamas skersinis sąlyginis cU (kontroliuojamas U). Ancilų kubitų būse-

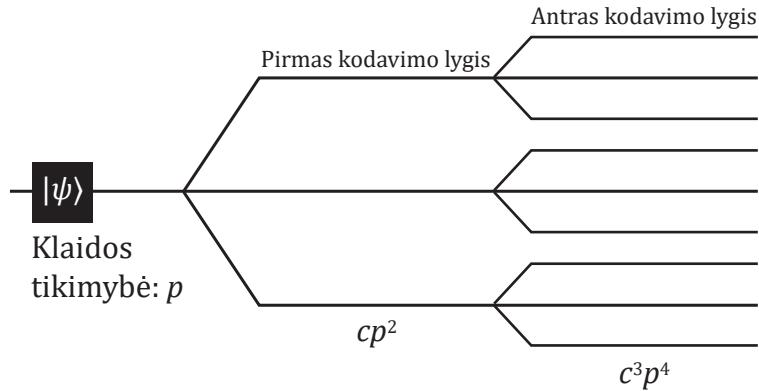
na $|\varphi\rangle$ užtikrina, kad klaidos neplinta šioje stadioje. Galiausiai ancilių kubitai yra dekoduojami, o galutinė bendra pirmo ancilos kubito ir $|\psi\rangle$ būseną lieka supintoji $|0\rangle \otimes P(1)|\psi\rangle + |1\rangle \otimes P(-1)|\psi\rangle$. Išmatavus ancilos kubito būseną bus rasta +1 arba -1 tikrinė vertė ir atlikta $|\psi\rangle$ būsenos projekcija į atitinkamą poerdvį. Siekiant sumažinti tikimybę, kad klaida dekodavimo stadijos loginiuose vartuose suteiks klaidingą matavimo rezultatą, visa ši matavimo procedūra kartojama tris kartus. Kai taikomas daugumos balso principas nustatant galutinio matavimo rezultatą, klaidingo atsakymo tikimybė sumažėja nuo p iki $O(p^2)$. Čia p yra tikimybė, kad atsiras klaida bet kuriamė grandinės elemente.



9.8 pav.: Klaidoms atsparūs 3 kubitų kodo loginio kubito būsenos bendrojo tipo matavimai

Galima parodyti, kad tikimybė, jog klaidoms atspariose grandinėse atsiras daugiau nei viena klaida kubitų bloke, yra cp^2 . Čia proporcijumo konstanta c priklauso nuo loginės operacijos bei kodavimo metodo ir bendrai nusako skaičių skirtinį vietų loginiame žingsnyje, kuriose gali įvykti klaida. Tad loginės operacijos klaidoms atsparioje kvantinėje grandinėje įvykdomos sėkmingai su tikimybe $1 - cp^2$ ir toliau įgaunamas pranašumas mažėjant p .

Kodų konkatenacijos metodas, taikomas kartu su klaidoms atspariomis loginėmis operacijomis, leidžia dar labiau sumažinti atsirandančių klaidų poveikį skaičiavimams. Kodų konkatenacijoje yra atliekamas aukštesnio lygio kodavimas taip sukuriant antro, trečio, ..., k lygio loginius kubitus (nebūtinai tuo pačiu kodu). Pavyzdžiui, jeigu naudotume tris fizinius kubitus sukurti 1 lygio loginį kubitą, tada kiekvieną iš šių kubitų koduodami dar trimis kubitais gautume 2 lygio loginį kubitą, iš viso panaudojė 3^2 fizinius kubitus. Atlikti logines operacijas šiems 2 lygio loginiams kubitams atitinkamai pasitelkiami jiems pritaikyti klaidoms atsparūs elementai, neleidžiantys klaidoms plisti. Pirmo lygio kodavime klaidos tikimybė yra cp^2 , antro lygio $c(cp^2)^2$, ir jeigu konkatenacija yra atliekama k kartų, loginiame k lygio kubite $-(cp)^{2^k}/c$. Dėl eksponentės, klaidingo atsakymo tikimybė gali būti padaroma pageidaujamai maža didinant konkatenacijų skaičių k , jeigu $cp < 1$.



9.9 pav.: Kodų konkatenacijos schema. Nurodytos nepataisomos klaidos tikimybė kiekviename kodavimo lygyje

Sakykime, kad norima atlikti skaičiavimą su ne didesne negu ε klaidos tikimybe, kuriam reikia pasitelkti iš viso $p(n)$ skaičių loginių vartų. Skaičius n nusako problemos dydį, o $p(n)$ – polinomiškai augantį loginių vartų skaičių. Tad klaidos tikimybė p per loginį žingsnį turėtų būti $p < \varepsilon/p(n)$. Todėl mažiausias konkatenacijų skaičius k , reikalingas pasiekti ši tikslą, randamas iš nelygybės:

$$\frac{(cp)^{2^k}}{c} \leq \frac{\varepsilon}{p(n)}. \quad (9.62)$$

Iš to išplaukia, kad konkatenacijų skaičius turi būti:

$$k \geq \left\{ 1 + \frac{\log \left[\frac{p(n)}{\varepsilon} \right]}{\log \left[\frac{1}{cp} \right]} \right\}. \quad (9.63)$$

Iš anksčiau pateiktų argumentų matome, kad kvantinės grandinės ilgis ir fizinių kubitų skaičius auga tik polilogaritmiškai su $p(n)/\varepsilon$. **Ribinė teorema** (angl. *threshold theorem*) formaliai įvardija, kad klaidos atsiradimo tikimybė kiekviename loginių operacijų žingsnyje turi būti $p_{th} < 1/c$, norint užtikrinti, kad k lygių konkatenacija leistų atlikti pageidaujamo tikslumo ir ilgio kvantinius skaičiavimus. Šios ribos apskaičiavimas yra svarbus kvantinių kompiuterių dizainui, įvairūs vertinimai rodo, kad $p_{th} \approx 10^{-4}\text{--}10^{-6}$.

9.11 Kvantinis tūris

Dekoherencijos trukmės bei loginių vartų tikslumas įvardija du esminius klaidų šaltinius (žr. 1.9 poskyrį). Tačiau didėjant kubitų skaičiui ir mažėjant šioms klaidoms atsiranda poreikis tiksliau įvertinti skaičiuojamąją galią. Tam idealiai norėtume suformuluoti rodiklį ar keletą rodiklių, kurie taikant standartinį protokolą leistų palyginti skirtingesius iženginius nepriklausomai nuo jų fizinio realizavimo. Galima įvardyti svarbiausius fizinius parametrus, kurie nulemia ankstyvosios NISQ raidos kvantinių kompiuterių skaičiuojamąją galią:

1. kubity skaičius;
2. 2 kubity (arba n kubity) loginių vartų realizavimo architektūra procesoriuje;
3. loginių grandinių gylis, kuri galima pasiekti, kol rezultatų neužmaskuoja klaidos;

4. pasiekiamas loginių vartų rinkinys;
5. operacijų skaičius, kurį galima vykdyti lygiagrečiai;

Siekdami apimti visus šiuos parametrus, panagrinėkime **kvantiniu tūriu** (angl. *quantum volume*) vadinamą rodiklį. Kvantinis tūris randamas nustačius didžiausią skaičių procesoriaus kubitų, kurie gali patikimai įvykdyti pateikto gylio specifinę grandinę. Kvanticinio tūrio rodiklis yra parametas **kvadratinėmis grandinėmis** (angl. *square circuits*), susidedančiomis iš d skaičiaus loginių vartų sluoksnių, nusakančių grandinės gylį, ir veikia tokiam pačiam d skaičiui kubitų, vadinamam **grandinės pločiu** (angl. *circuit width*). Imkime procesorių, sudarytą iš n kubitų. Pradėdami protokolą bandytume įvykdyti kvadratinę 2×2 grandinę, sudarytą iš $d = 2$ gylio su pasirinktais 2 kubitais. Čia turime laisvę pasirinkti geriausiai funkcionuojančius kubitus procesoriuje. Jeigu, pagal formaliai nustatytus rodiklius, grandinė atliekama sėkmingai, tada tēstume toliau: $3 \times 3, 4 \times 4, \dots$, kol rastume didžiausią $m \times m$ dydžio kvadratinę grandinę ($m \leq n$), kurios procesorius nebegali sėkmingai įvykdyti. Taip randamas procesoriaus kvantinis tūris, jis žymimas simboliu V_Q , o jo formalus apibrėžimas yra:

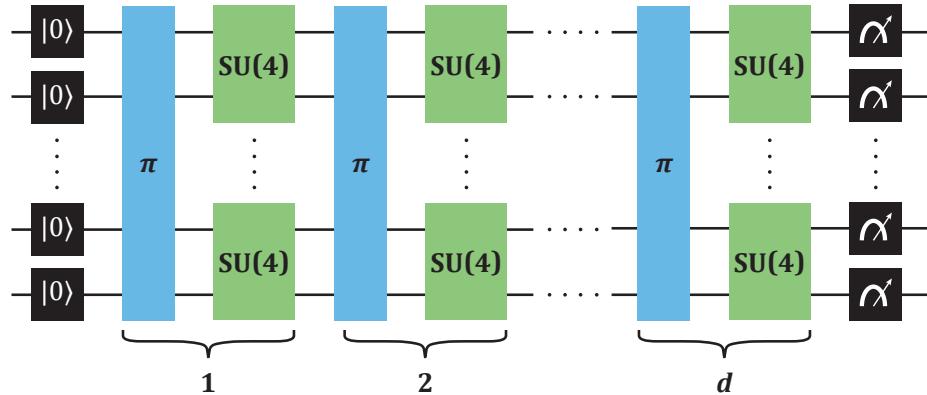
$$\log_2 V_Q = \max_{m \leq n} \{\min(m, d(m))\}. \quad (9.64)$$

Ši išraiška nusako, kad ieškomas didžiausias m skaičius kubitų tarp visų procesoriaus n kubitų, kurie sėkmingai įvykdo grandinę, $d(m)$ yra maksimalus gylis didžiausioje m kubitų skaičiaus kvadratinėje grandinėje. Čia tūris pateikiamas naudojant logaritmą su baze 2, tad $\log_2 V_Q$ galima interpretuoti kaip procesoriaus efektyvų kubitų skaičių, kuris gali būti lygus arba mažesnis nei fizinė kubito skaičius procesoriuje. Eksponentiškai kubito skaičiumi išreikštasis kvantinis tūris V_Q atspindi kvantinių būsenų erdvės dydį, kurį procesorius gali efektyviai pasiekti atliekant unitarišias transformacijas. Jeigu, sakykime $d(m) = 10$, tada kvantinis tūris, imant eksponentę su baze 2, yra randamas: $V_Q = 2^{10} = 1024$.

Toliau apžvelgiame kvantinio tūrio įvertinimo protokolą. Kiekvieną loginių vartų sluoksnį sudaro dvi dalys: atsitiktinis visų d kubito indeksavimo sukeitimai (angl. *permutation*) ir atsitikinės 2 kubito unitariosios transformacijos, atliekamos kiekvienai porai vienas šalia kito esančių (ar atsiradusiu po sukeitimo) kubito. Kubito indeksavimo sukeitimai kvantinėje grandinėje bendrai žymimas raide π ir pasitelkia *SWAP* loginius vartus. O štai kiekviena iš 2 kubito operacijų yra parenkama **Haar-atsitiktinai** (angl. *Haar random*) iš (4×4) matricų, nusakančių bendro tipo unitarišias 2 kubito transformacijas. Šių transformacijų assortimentas yra formaliai vadinamas **SU(4) grupe** (angl. *special unitary group*). Haar-atsitiktinis SU(4) matricos parinkimas yra analogiškas atsitiktiniams skaliarinio skaičiaus parinkimui iš lygai pasiskirsčiusių skaičių rinkinio. Kvanticinio tūrio protokolą atliekanti grandinė yra parodyta 9.10 pav.

Siekiant įvertinti, ar grandinė buvo įvykdyta sėkmingai, taikomas **sunkiųjų išvesties būsenų generavimas** (angl. *heavy output generation*). Išmatuotų galutinių būsenų pasiskirstymas yra nusakomas tikimybėmis $p_U(x) = |\langle x|U|0\rangle|^2$. Čia $|0\rangle$ yra pradinė n kubito registro būsena, U nusako visą n kubito protokolo unitariųjų transformacijų seką, o $|x\rangle$ yra galutinė registro būsena. Galutinės būsenos $|x\rangle$ ir tikimybės $p_U(x)$ randami atliekant modeliavimą klasikiniu kompiuteriu. Sunkiosios išvesties kubito būsenos yra tos, kurių tikimybės jas rasti yra didesnės nei visų galimų būsenų tikimybų mediana, $p_U(x) > p_{\text{med}}$. Taikant Haar-atsitiktinumą galima apskaičiuoti, kad tikimybė rasti būsenas $|x\rangle$ aukščiau medianos yra $p = 0.85$ ir asimptotiškai artėja prie $p = 0.5$ tikimybės, jeigu įrenginys veikia itin blogai. Kvadratinės grandinės testas yra laikomas įvykdytu, jeigu bent $2/3$ visų sugeneruotų būsenų atitinka būsenas su didesnėmis tikimybėmis nei tikimybų mediana.

Didesnis kvantinis tūris gali būti pasiektas procesoriuose, kurie turi daugiau kubito su ilgomis koherencijos trukmėmis ir mažomis loginių vartų klaidomis. Taip pat svarbu, kad protokolo

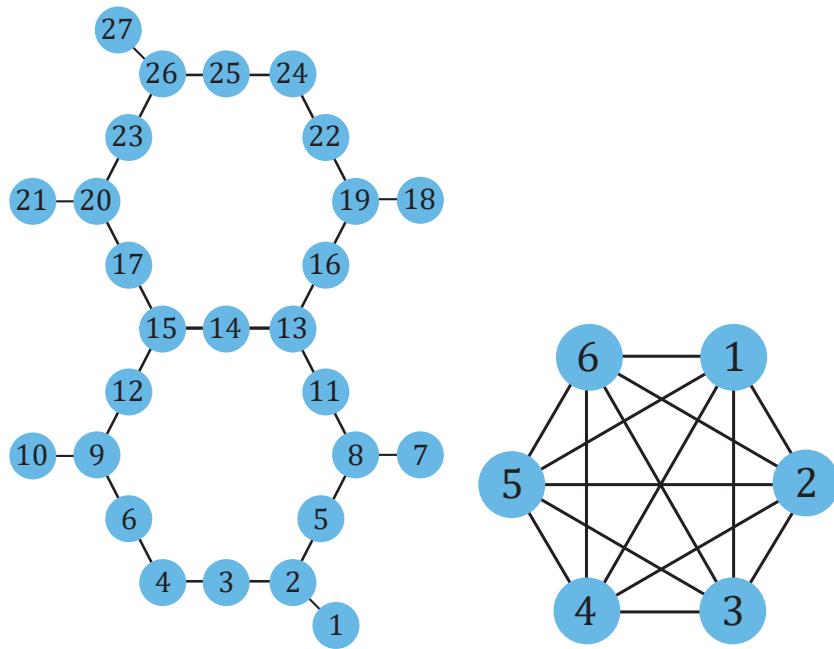
9.10 pav.: Kvadratinėmis ($d \times d$) grandinėmis pagrįstas kvantinio tūrio nustatymo protokolas

pateikti *SWAP* bei $SU(4)$ loginiai vartai galėtų būti efektyviai išreikšti kvantinio procesoriaus loginių vartų rinkiniu. Svarbu yra ir kubitų maksimalus tarpusavio jungimas, loginių operacijų parallelizavimo galimybės, taip pat optimaliai sukompliuotos loginės operacijos. Kitu atveju siekiant išvystyti pateiktą kvantinę grandinę reikės papildomų loginių operacijų skaičiaus, ir realus grandinės gylis įrenginyje bus didesnis, tad tikimybė ją sėkmingai išvystyti mažės.

Siekdami iliustruoti kubitų tarpusavio jungimo įtaką skaičiavimams, 9.11 pav. pateikiame dvi skirtinges kvantinio procesoriaus kubitų jungčių schemas. Pirmoje pavaizduota IBM superlaidininkais pagrįstų grupės *Falcon* procesorių architektūra, kurioje kubitų išdėstyti paremtas heksagonine simetrija. Toks kubitų išdėstymas yra specialiai gamintojų pritaikytas atlikti jų parinktiems klaidų taisymo algoritmams, atsižvelgiant į kitus procesoriaus parametrus. Didėjant kubitų skaičiui, jų išdėstymas *Falcon* procesoriuose toliau bus paremtas heksagonine simetrija. Čia kubitas, pagal savo poziciją, gali turėti nuo vienos iki trijų jungčių su **artimiausiai esančiais kubitais kaimynais** (angl. *nearest-neighbor connectivity*). Tai reiškia, kad tiktai tarp šių kubitų įmanomi 2 kubitų loginiai vartai, tokie kaip cX . Pavyzdžiui, atlikti cX tarp kubitų #1 ir #15 tiesiogiai neįmanoma. Tam būtina išvystyti seką *SWAP* loginių operacijų tarp tarpinių kubitų, šitaip realizuojant norimą operaciją šiemis dviem kubitams.

Antroji pateikta kvantinio procesoriaus architektūra leidžia 2 kubitų loginius vartus atlikti tiesiogiai tarp bet kurių kubitų. Tai vadinamas **visų su visais” jungimasis** (angl. *all-to-all connectivity*). Jonų gardelėmis pagrįsti kvantiniai procesoriai pasižymi galimybėmis realizuoti „visų su visais” architektūrą. Neatsižvelgiant į kitus procesoriaus veikimo faktorius, tokio tipo architektūra turi akivaizdū pranašumą prieš IBM *Falcon*, nes potencialiai sumažina reikalaujamą loginių vartų skaičių atlikti tam pačiam algoritmui.

Kvantinis tūris leidžia patikimai išvertinti pagrindinius skaičiuojamają galą nulemiančius faktorius, tačiau yra orientuotas į artimosios raidos NISQ kvantinius kompiuterius. Spartus klasikinių skaičiavimo išteklių augimas gali užkirsti kelią atlikti klasikiniam kvantinės grandinės modeliavimui, kuris reikalingas rezultatų patikrinimui procesoriuose su daugiau nei ~60 kubitų. Augant kubitų skaičiui bei mažėjant klaidoms bus reikalingas kitas būdas išvertinti kvantinių kompiuterių skaičiuojamajai galiai.



9.11 pav.: Dvi kvantinio procesoriaus architektūros, pasitelkiančios skirtinę kubitų tarpusavio jungimą. Tai nulemia, tarp kurių kubitų galima atlikti 2 ar daugiau kubitų sąlyginius loginius vartus

A skyrius

Debesyje pasiekiami kvantiniai kompiuteriai ir simuliatoriai

- <https://quantum-computing.ibm.com>
- <https://ionq.com>
- <https://www.xanadu.ai>
- <https://www.rigetti.com>
- <https://aws.amazon.com/braket/quantum-computers>
- <https://azure.microsoft.com/en-us/products/quantum>
- <https://quantumai.google>
- <https://qiskit.org>
- <https://pennylane.ai/>

176A SKYRIUS. DEBESYJE PASIEKIAMI KVANTINIAI KOMPIUTERIAI IR SIMULIATORIAI

Literatūra

- Einstein, A., B. Podolsky ir N. Rosen (1935 m.). „Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?“ In: *Physical Review* 47, p. 777.
- Bohm, D. (1951 m.). *Quantum Theory*. Englewood Cliffs: Prentice-Hall.
- Wootters, W. K. ir W. H. Zurek (1982 m.). „A single quantum cannot be cloned“. In: *Nature* 299, p. 802–803.
- Bell, J. S. (1989 m.). *Speakable and unspeakable in quantum mechanics*. Cambridge University Press.
- Deutsch, D. ir R. Jozsa (1992 m.). „Rapid Solution of Problems by Quantum Computation“. In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439, p. 1907.
- Sakurai, J. J. (1994 m.). *Modern Quantum Mechanics*. Addison-Wesley.
- Shor, P. W. (1997 m.). „Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer“. In: *SIAM Journal on Computing* 26.5, p. 1484–1509. DOI: 10.1137/s0097539795293172.
- James, D. F. V. (1998 m.). „Quantum dynamics of cold trapped ions with application to quantum computation“. In: *Applied Physics B: Lasers and Optics* 66, p. 181–190. DOI: 10.1007/s00340050373.
- Feynman, R. (2000 m.). *Feynman Lectures on Computation*. Perseus Books Group.
- Nielsen, M. A. ir I. L. Chuang (2000 m.). *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press.
- Shor, P. W. ir J. Preskill (2000 m.). „Simple Proof of Security of the BB84 Quantum Key Distribution Protocol“. In: *Physical Review Letters* 85, p. 441.
- Kitaev, A. Yu., A. H. Shen ir M. N. Vyalyi (2002 m.). *Classical and Quantum Computation*. T. 47. Graduate Studies in Mathematics. American Mathematical Society.
- Jaeger, R. C. ir T. N. Blalock (2003 m.). *Microelectronic circuit design*. Dubuque: McGraw-Hill.
- Benenti, G., G. Casati ir G. Strini (2005 m.). *Principles of Quantum Computation and Information. Volume I: Basic Concepts*. World Scientific.
- Soklakov, A. N. ir R. Schack (2006 m.). „Efficient state preparation for a register of quantum bits“. In: *Physical Review A* 73, p. 012307.
- Audretsch, J. (2007 m.). *Entangled Systems. New Directions in Quantum Physics*. Wiley.
- Benenti, G., G. Casati ir G. Strini (2007 m.). *Principles of Quantum Computation and Information. Volume II: Basic Tools and Special Topics*. World Scientific.
- Mermin, N. D. (2007 m.). *Quantum Computer Science. An Introduction*. Cambridge University Press.
- Benenti, G. ir G. Strini (2008 m.). „Quantum simulation of the single-particle Schrödinger equation“. In: *American Journal of Physics* 76.7, p. 657–662. DOI: 10.1119/1.2894532.

- Giovannetti, V., S. Lloyd ir L. Maccone (2008 m.). „Quantum Random Access Memory“. In: *Physical Review Letters* 100, p. 160501. DOI: 10.1103/physrevlett.100.160501.
- Nakahara, M. ir T. Ohmi (2008 m.). *Quantum Computing—From Linear Algebra to Physical Realizations*. CRC Press.
- Harrow, A. W., A. Hassidim ir S. Lloyd (2009 m.). „Quantum Algorithm for Linear Systems of Equations“. In: *Physical Review Letters* 103, p. 150502. DOI: 10.1103/physrevlett.103.150502.
- Eraerds, P. ir kt. (2010 m.). „Quantum key distribution and 1Gbps data encryption over a single fibre“. In: *New Journal of Physics* 12.6, p. 063027. DOI: 10.1088/1367-2630/12/6/063027.
- Devitt, S. J., W. J. Munro ir K. Nemoto (2013 m.). „Quantum error correction for beginners“. In: *Reports on Progress in Physics* 76.7, p. 076001. DOI: 10.1088/0034-4885/76/7/076001.
- Ma, X.-S. ir kt. (2013 m.). „Quantum erasure with causally disconnected choice“. In: *Proceedings of the National Academy of Sciences* 110.4, p. 1221–1226. DOI: 10.1073/pnas.1213201110.
- Monroe, C. ir J. Kim (2013 m.). „Scaling the Ion Trap Quantum Processor“. In: *Science* 339, p. 6124.
- Lloyd, S., M. Mohseni ir P. Rebentrost (2014 m.). „Quantum principal component analysis“. In: *Nature Physics* 10.9, p. 631–633. DOI: 10.1038/nphys3029.
- Wittek, P. (2014 m.). *Quantum Machine Learning. What Quantum Computing Means to Data Mining*. Elsevier.
- Somma, R. D. (2016 m.). „Quantum Simulations of One Dimensional Quantum Systems“. In: *Quantum Information and Computation* 16.13-14, p. 1125–1168. ISSN: 1533-7146.
- Biamonte, J. ir kt. (2017 m.). „Quantum machine learning“. In: *Nature* 549.7671, p. 195–202. DOI: 10.1038/nature23474.
- Figgatt, C. ir kt. (2017 m.). „Complete 3-Qubit Grover search on a programmable quantum computer“. In: *Nature Communications* 8, p. 1918. DOI: 10.1038/s41467-017-01904-7.
- Gambetta, J. M., J. M. Chow ir M. Steffen (2017 m.). „Building logical qubits in a superconducting quantum computing system“. In: *npj Quantum Information* 3, p. 2. DOI: 10.1038/s41534-016-0004-0.
- Yin, J. ir kt. (2017 m.). „Satellite-based entanglement distribution over 1200 kilometers“. In: *Science* 356.6343, p. 1140–1144. DOI: 10.1126/science.aan3211.
- Preskill, J. (2018 m.). „Quantum Computing in the NISQ era and beyond“. In: *Quantum* 2, p. 79. ISSN: 2521-327X. DOI: 10.22331/q-2018-08-06-79.
- Rebentrost, P. ir S. Lloyd (2018 m.). *Quantum computational finance: quantum algorithm for portfolio optimization*. arXiv:1811.03975 [quant-ph]. DOI: 10.48550/arxiv.1811.03975.
- Slussarenko, S. ir G. J. Pryde (2019 m.). „Photonic quantum information processing: A concise review“. In: *Applied Physics Reviews* 6.4, p. 041303. DOI: 10.1063/1.5115814.
- Schuld, M. ir kt. (2020 m.). „Circuit-centric quantum classifiers“. In: *Physical Review A* 101, p. 032308.
- Wiebe, N. (2020 m.). „Key questions for the quantum machine learner to ask themselves“. In: *New Journal of Physics* 22, p. 091001.
- Jurcevic, P. ir kt. (2021 m.). „Demonstration of quantum volume 64 on a superconducting quantum computing system“. In: *Quantum Science and Technology* 6.2, p. 025020.
- J, A. ir kt. (2022 m.). „Quantum Algorithm Implementations for Beginners“. In: *ACM Transactions on Quantum Computing* 3.4, p. 18. ISSN: 2643-6809. DOI: 10.1145/3517340.