LATVIJAS UNIVERSITĀTE
FIZIKAS UN MATEMĀTIKAS FAKULTĀTE
DATORIKAS NODAĻA

KVANTU VAICĀJOŠIE ALGORITMI BŪLA FUNKCIJU RĒĶINĀŠANAI

MAĢISTRA DARBS

Autors: **Taisija Miščenko-Slatenkova**
Stud. apl. DatZ010026
Darba vadītājs: profesors Rūsiņš-Mārtiņš Freivalds

RĪGA 2007

ANOTĀCIJA

Šajā darbā ir aplūkoti kvantu vaicājošie algoritmi Būla funkciju rēķināšanai. Apskatītas problēmas ir cieši saistītas ar uzdevumu uzkonstruēt efektīvu kvantu algoritmu patvaļīgai Būla funkcijai vai funkciju kopai. Pētījuma mērķis ir atrast kvantu algoritmus ar eksaktu sarežģītību mazāku par determinētu. Darbā ir aprakstīta jaunu eksaktu algoritmu kopa ar kvantu sarežģītību divreiz mazāku par determinētu. Darbā ir aprakstītas divas plašas kvantu algoritmu ar ierobežotu kļūdu kopas: algoritmi ar sarežģītību $D(f) = 2n$ pret $Q_{3/4}(f) = \left\lceil \frac{n}{2} \right\rceil$, kur $2n$ ir Būla funkcijas mainīgo skaits, un algoritmi ar sarežģītību $D(f) = 3n$ pret $Q_{9/10}(f) = n$, kur $3n$ ir Būla funkcijas mainīgo skaits. Īpaša uzmanība ir pievērsta kvantu jaukto stāvokļu jēdzienam ar cerību uzlabot iespējamu kvantu algoritma sarežģītību.

Atslēgvārdi: kvantu vaicājošais algoritms, kvantu sarežģītība, determinēta sarežģītība, Būla funkcija.

ANNOTATION

The problem discussed in this work is related to search for Booleans function with deterministic complexity higher than quantum. It is not a trivial task to design an effective quantum query algorithm for an arbitrary Boolean function. Main purpose of the research is to find quantum query algorithms with exact quantum complexity lower than deterministic one. The work presents a set of new exact quantum algorithms with quantum complexity 2 times lower than deterministic. Moreover, there are two sets of bounded-error quantum algorithms described: algorithms with complexity $D(f) = 2n$ versus $Q_{3/4}(f) = \left\lceil \frac{n}{2} \right\rceil$ for $2n$ - variable Boolean functions, and with complexity $D(f) = 3n$ versus $Q_{9/10}(f) = n$ for $3n$ - variable Boolean functions. Huge attention has been paid to quantum mixed states with a hope to enlarge quantum and deterministic complexity gap.

Key words: Quantum query algorithms, quantum complexity, deterministic complexity, Boolean function

AUTOREFERĀTS

Šis darbs sastāv no ievada, teorētiskās daļas, pētījumu rezultātu apraksta, nobeiguma un literatūras saraksta.

Darbā ir mēģinājums precizēt jau zināmus Būla funkciju sarežģītību novērtējumus un nosacījumus, pie kuriem Būla funkcijas kvantu sarežģītība būtu ievērojami mazāka par determinētu. Tika aplūkoti dažādi Būla funkcijas skaitļošanas modeļi- determinēts, varbūtisks un kvantu vaicājošie algoritmi. Ir izpētīts jautājums par kvantu sistēmas jaukto stāvokļu ietekmi uz kvantu vaicājoša algoritma sarežģītību salīdzinājumā ar Būla funkcijas determinētu sarežģītību. Pētījumu mērķis bija izstrādāt un prezentēt efektīvus jaunus kvantu algoritmus Būla funkciju kopai. Šim nolūkam tika izpētītas vairākas grāmatas un raksti par šo tēmu.

Darbā ir precīzi aprakstīta autora izdomāta kvantu eksakto algoritmu kopa ar kvantu sarežģītību $Q_E(f) = D(f)/2$, kas atkārto PARITY funkcijas determinētas un kvantu sarežģītību starpību.

Darbā ir aprakstīti kvantu algoritmi ar ierobežotu kļūdu:

- algoritmu kopa ar sarežģītību D(f)=2n pret $Q_{3/4}(f)=\left\lceil \frac{n}{2} \right\rceil$, kur 2n ir Būla funkcijas mainīgo skaits, bet $Q_{3/4}$ nozīmē, ka pareiza atbilde parādās ar varbūtību ne mazāku par ¾
- divi dažādi kvantu algoritmi Būla funkcijai EQUALITY$_3$ no 3 mainīgajiem: funkcijas vērtība ir 1 tad un tikai tad, ja visu argumentu vērtības sakrīt. Viens paņēmiens dod algoritmu ar sarežģītību $Q_{8/9}(EQUALITY_3)=1$ un cits paņēmiens ļauj uzkonstruēt algoritmu ar sarežģītību $Q_{9/10}(EQUALITY_3)=1$ pret $D(EQUALITY_3) = 3$.
- algoritmu kopa ar sarežģītību $D(EOX_{3n}) = 3n$ pret $Q_{9/10}(EOX_{3n}) = n$ vai $Q_{8/9}(EOX_{3n}) = n$ kur 3n ir Būla funkcijas mainīgo skaits. Algoritma pamats ir funkcijas $EQUALITY_3$ algoritms.
- algoritmu kopa ar sarežģītību $D(TAND_{4n}) = 4n$ pret $Q_{3/4}(TAND_{4n}) = n$, kur 4n ir Būla funkcijas mainīgo skaits. Algoritma pamats ir funkcijas $EQUALITY_3$ algoritms.

Īpaši uzmanīgi ir izpētīta iespēja vēl uzlabot kvantu algoritma sarežģītību ar jaukto stāvokļu palīdzību. Šīm nolūkam tika kārtīgi izstudēts kvantu jauktā stāvokļa jēdziens, nodefinēts jaukto stāvokļu kvantu vaicājošā algoritma jēdziens. Kā pētījumu rezultāts ir noformulēta un pierādīta teorēma par neiespējamību uzkonstruēt eksaktu kvantu algoritmu ar $Q_E(f) < D(f)/2$ 2-kubitu kvantu sistēmas gadījumā.

Darba rezultāti ir publicēti rakstā, ar kuru autori piedalīsies pasākumā „Workshop on Probabilistic and Quantum Automata, July 7,2007, Turku, Finland” (Informācija ir pieejama mājas lapā http://www.math.utu.fi/projects/dlt2007/pqa/)

SATURS

IEVADS

Algoritmu sarežģītība ir datorzinātnes nodaļa, kuras mērķis ir novērtēt dotā skaitļošanas uzdevuma sarežģītību, noskaidrot skaitļošanas ilguma apakšējās un augšējās robežas. Diemžēl, daudziem praktiski pielietojamiem uzdevumiem ciešas robežas nav zināmas. Risināt vienkāršāku problēmu un rezultātu pielietot grūtākas problēmas risinājumam- ir paņēmiens sarežģītu uzdevumu risināšanai. Ņemot vērā tādu piegājienu, vispirms tika izpētīti vienkāršākie skaitļošanas modeļi. Par tādu var saukt lēmumu koku, kura uzdevums ir atrast Būla funkcijas $f:\{0,1\}^n \rightarrow \{0,1\}$ vērtību, vaicājot mainīgo vērtības kādā noteiktā secībā. Šī algoritma *k*-tais vaicājums var būt atkarīgs no k-1 iepriekšējiem.
Determinētas funkcijas *f* sarežģītība ir minimāls vaicājumu skaits. Pētot funkciju, tās determinēto sarežģītību salīdzina ar kādu citu, piemēram, ar varbūtisku vai kvantu. Determinēta algoritma sarežģītība ir vienāda vai lielāka par kvantu algoritma sarežģītību visām Būla funkcijām. Tas nozīmē, ka funkcijas rēķināšanā kvantu algoritmam var būt svarīgas priekšrocības salīdzinājumā ar determinēto.

Darba mērķis ir atrast tādas Būla funkcijas, kurām determinēta algoritma sarežģītība $D(f)$ ir ievērojami lielāka par kvantu $Q_E(f)$. Vēlme uzlabot kvantu algoritma sarežģītību pašreiz nozīmē atrast Būla funkcijas ar $Q_E(f)<D(f)/2$. Šajā jomā diezgan vāji ir izpētīti kvantu algoritmi kvantu jaukto stāvokļu sistēmā, to priekšrocības un trūkumi.
Iterētas Būla funkcijas, tas ir Būla funkcijas tipa f(f(), f()), nedod labus kvantu algoritmus, tāpēc jācenšas meklēt oriģinālas funkcijas. Viss aprakstītais ir mēģinājums virzīties tajā jomā. Pašreiz pasaules rekords ir Būla funkcija PARITY no *2n* mainīgajiem: determinēti vajag *2n* jautājumus, bet kvantiski pietiek ar *n* jautājumiem un ir pierādīts, ka ar (n-1) jautājumiem nepietiek.
No otras puses, ja determinēti vajag *n* jautājumus, tad kvantiski nevar ar mazāk, nekā kubsakne no *n* jautājumiem, jo ir pierādīta sakarība $D(f) \leq 16\, Q_E(f)^3$[3].
Maģistra darba mērķis ir dziļāk izpētīt funkcijas sarežģītības novērtējumus un sakarus starp tiem, un, pēc iespējas, atrast vispārinātu paņēmienu eksaktu kvantu algoritmu konstruēšanai patvaļīgai Būla funkcijai.

Maģistra darbam ir sekojoša struktūra:
Nodaļās 1. – 4. ir teorijas izklāsts, kura ir nepieciešama pētījumu rezultātu saprašanai un novērtēšanai.
Nodaļās 5. – 8. ir aprakstīti autora rezultāti.

5. nodaļā ir prezentēta efektīvu eksaktu algoritmu kopa daudzargumentu Būla funkciju rēķināšanai ar sarežģītību $Q_E(f) = D(f)/2$, kā arī daži citu algoritmu piemēri.
6. un 7. nodaļas ir veltītas kvantu vaicājošiem algoritmiem ar ierobežotu kļūdu, piedāvātās algoritmu kopas ļauj ar mazu kļūdas varbūtību samazināt kvantu vaicājumu skaitu salīdzinājumā ar eksakta algoritma nepieciešamu vaicājumu skaitu. 6. nodaļā ir prezentēta 3-argumentu Būla funkcija $\mathrm{EQUALITY}_3$ ar determinētu sarežģītību $D(EQUALITY_3) = 3$ un divi atšķirīgi kvantu algoritmi funkcijas rēķināšanai ar sarežģītību $Q_{8/9}(EQUALITY_3) = 1$ un $Q_{9/10}(EQUALITY_3) = 1$. Kā arī ir prezentēta algoritmu kopa ar sarežģītību $Q_{3/4}(f) = \left\lceil \frac{n}{2} \right\rceil$ pret $D(f) = 2n$.
7. nodaļā ir piedāvāti paņēmieni(un definētas algoritmu kopas ilustrācijai), kā uzkonstruēt sarežģītāku algoritmu (lielākai mainīgo kopai), ņemot vienkāršāko par pamatu.
8. nodaļā ir mēģinājums pierādīt, ka jaukto stāvokļu sistēmā nevar uzbūvēt eksaktu algoritmu ar kvantu sarežģītību mazāku par D(f)/2, tas ir uzlabot funkcijas PARITY rezultātu. Ir aplūkots privātgadījums 2-kubitu kvantu sistēmai, kuram šī hipotēze ir pierādīta.

Daļēji darba rezultāti ir publicēti rakstā, kurš pieņemts pasākumam „Workshop on Probabilistic and Quantum Automata, July 7,2007, Turku, Finland”
(Informācija ir pieejamā mājas lapā http://www.math.utu.fi/projects/dlt2007/pqa/)

1. PAMATJĒDZIENI

1.1. Būla funkcijas

Definīcija 1. ***Būla funkcija*** ir visur definēta funkcija $f:\{0,1\}^n \to \{0,1\}$. Funkcijas f arguments $x \in \{0,1\}^n$, $x=x_1...x_n$, kur ar x_i apzīmē x i-to bitu. x *Heminga svars* |x| ir x_i=1 skaits. Ja ar S apzīmē mainīgo indeksu kopu, tad x^S ir ievads, kuru iegūst, mainot katra x_i, $i \in S$, vērtību uz pretējo (0 uz 1, un otrādi). Ja S={i}, t.i. kopas S elementu skaits ir 1, tad raksta vienkārši x^i. Piemēram, ja
x = 0011, tad $x^{\{2,3\}} = 0101$ un $x^4 = 0010$. Funkciju f sauc par *simetrisku*, ja $f(x)$ vērtība ir atkarīga tikai no $|x|$.
Simetrisku funkciju piemēri:

- $OR_n(x) = 1$ tad un tikai tad, ja $|x| \geq 1$
- $AND_n(x) = 1$ tad un tikai tad, ja $|x| = n$
- $PARITY_n(x) = 1$ tad un tikai tad, ja $|x|$ ir nepārskaitlis
- $MAJ_n(x) = 1$ tad un tikai tad, ja $|x| > n/2$.

1.2. Funkcijas jutība pret viena vai dažādu bitu izmaiņām

Jutība pret viena vai dažādu bitu izmaiņām nosaka, cik jutīga ir funkcijas f vērtība pret izmaņām ievadā.

Definīcija 2. Funkcijas f jutība pret ievada x viena mainīga izmaiņu $s_x(f)$ ir mainīgo x_i skaits, kuriem $f(x) \neq f(x^i)$. Funkcijas f ***jutība*** $s(f)=\max_x s_x(f)$.

Definīcija 3. Funkcijas f jutība pret dažādu ievada x mainīgu izmaņu $bs_x(f)$ ir maksimālais skaitlis b, tāds, ka eksistē netukšas kopas $B_1,...,B_b$, $\forall i,j(i \neq j \Rightarrow B_i \cap B_j = \varnothing)$, un $f(x) \neq f(x^{Bi})$. Netukšas kopas $B_1,...,B_b$ sauc par ievada ***x atdalītiem jutības blokiem***. Funkcijas f ***bloku jutība*** ir $bs(f)=\max_x bs_x(f)$. (ja f ir konstante, definē s(f)=bs(f)=0.)

Jutība ir *bloku jutība* ar bloku B_i apjomu vienādu ar 1.

Teorēma 1.[4] Ja x nesatur fiktīvus mainīgus, tad

$$s(f) \leq \frac{1}{2}\log_2(n) - \frac{1}{2}\log_2\log_2(n) + \frac{1}{2} \ .$$

Teorēma 2.[2] $s(f) \le bs(f) \le n$.

Pierādījums. Ievada x atdalītie jutības bloki ir $B_1,...,B_{bs_x(f)}$. Ja visi B_i ir kopas ar apjomu 1, tad $s(f) \le n$ un $bs(f) \le n$, tad $s_x(f)$ ir jutību bloku skaits :

$$s_x(f) = \left|\left\{B_i \middle| = 1 \mid B_i \in \{B_1,...,B_{bs_x(f)}\}\right\}\right| \le bs_x(f) \ .$$

Lemma 1.[2] Ja B ir minimāls jutības bloks ievadam x, tad $|B| \le s(f)$.

Pierādījums. Ja maina vienu no B-mainīgajiem ievadā x^B, tad funkcijas vērtībai jāmainās no $f(x^B)$ uz f(x)(citādi B nav minimāls), tāpēc f ir jutīga pret katru B-mainīga izmaiņu ievadā x^B. Tātad, $|B| \le s_{xB}(f) \le s(f)$.

$bs(f)$ apakšēja robeža nav zināma, ja f nav konstante. Pastāv hipotēze, ka $bs(f) \in O(s(f)^2)$.

Teorēma 3.[2] Ja f(0)=0 un katram x, tādam, ka |x|=1, f(x)=1, tad s(f)=n.

Pierādījums. Pieņem, ka f(0)=0, f(x)=1 katram x ar īpašību |x|=1. Mainot jebkuru bitu nulles ievadā, iegūst x tādu, ka |x|=1. Tātad, funkcijas vērtība mainās uz katra nulles ievada bita izmaiņu, un $f(0^i) \ne f(0)$.

2. VAICĀJOŠIE ALGORITMI

Šajā nodaļā tiek definēts lēmumu koka jēdziens determinētiem un varbūtiskiem algoritmiem.

2.1. Lēmumu koks. Determinēts algoritms

Definīcija 4. ***Lēmumu koks*** ir algoritms, kurš rēķina kādu funkciju $f(x_1, \ldots, x_N)$, uzstādot jautājumus par x_i vērtībām.

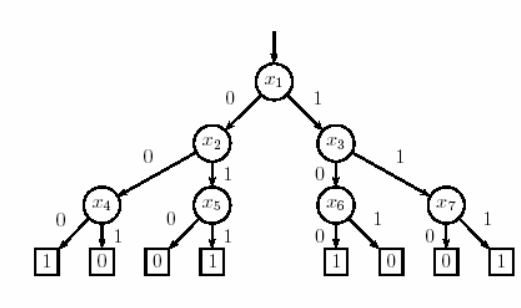
Lēmumu koka sarežģītība ir maksimālais uzdoto jautājumu skaits. Funkcijas $f(x_1, \ldots, x_N)$ lēmumu koka sarežģītība ir optimāla lēmumu koka, kurš rēķina funkciju f, sarežģītība.

Skaitļošanas teorija apskata dažādus skaitļošanas modeļus: determinēts, nedeterminēts, varbūtisks. Pastāv lēmumu koki katram modelim.

Definīcija 5. ***Determinēts algoritms*** ir algoritms, kura rīcība un gala rezultāts ir pilnīgi atkarīgi no ievada. Tas nozīmē to, ka katram ievadam algoritms izdod noteikto rezultātu katru reizi, kad satiek to ievadu.

Definīcija 6. ***Determinēts lēmumu*** *koks* ir sakārtots binārs koks T, kuram katra iekšēja virsotne satur patvaļīgo mainīgo x_i , un katra lapa satur 0 vai 1.

Piemēram, attēlā 1 ir determinēts lēmumu koks 7-mainīgo funkcijai:

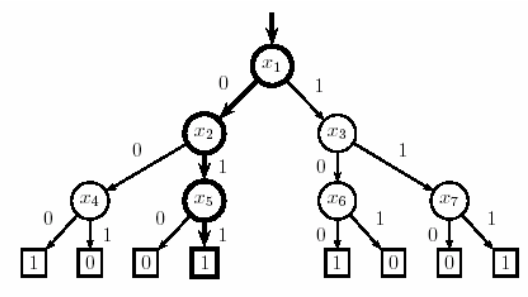


1. att. Determinēts lēmumu koks funkcijai no 7 mainīgajiem.

Katram ievadam $x \in \{0,1\}^n$ lēmumu koka vērtība tiek aprēķināta sekojoši:

1. sākt ar sakni.
2. apstāties, ja sastop lapu. Koka rezultāts ir lapas vērtība.
3. ja virsotne nav lapa, tad prasīt mainīga x_i vērtību, kuru tā satur.
4. ja $x_i=0$, tad rekursīvi aprēķināt kreiso apakškoku.
5. ja $x_i=1$, tad rekursīvi aprēķināt labo apakškoku.

Koka apstaigāšanas piemērs ir attēlā 2:



2. att. Determinēta lēmumu koka apstaigāšanas piemērs.

Ievads $x = x_1x_2...x_n$ determinēti nosaka gala lapu, tātad arī aprēķina rezultātu.

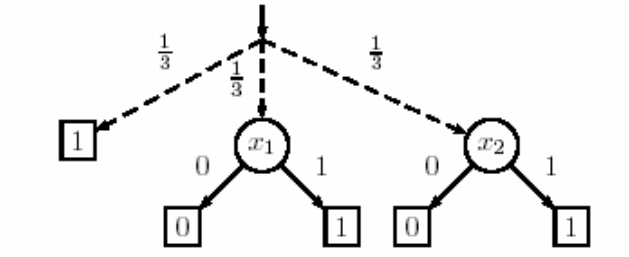
Definīcija 7. Saka, ka ***lēmumu koks aprēķina*** f, ja lēmumu koka aprēķināta vērtība ir vienāda ar $f(x)$ visiem $x \in \{0,1\}^n$. Ir acīmredzami, ka patvaļīgas funkcijas f ***aprēķināšanai*** eksistē dažādi lēmumu koki.

Koka sarežģītība ir tā dziļums, tas ir vaicājumu skaits sliktākajā gadījumā. Mūsu piemērā (attēls 2) ir 7-argumentu funkcijas determinēta lēmumu koka sarežģītība ir 3.

Definīcija 8. f ***determinēta lēmumu koka sarežģītība D(f)*** ir optimāla determinēta lēmumu koka dziļums, kurš aprēķina f. Šeit optimāls koks nozīmē minimāla dziļuma koku.

2.2. Lēmumu koks. Varbūtisks algoritms

Varbūtiska algoritma galvenā īpašība ir sekojoša: šķautnēm, izejošām no vienas virsotnes, ir iespējams piešķirt varbūtību, bet ir jāsaglabā nosacījums, ka no vienas virsotnes izejošo šķautņu varbūtību summa nepārsniedz 1. Kaut gan šī summa var būt vienāda ar 0, kas ir līdzvērtīgs situācijai, kad virsotnei nav izejošo šķautņu, piemēram, 3. attēlā ir funkcijas OR varbūtisks lēmumu koks:



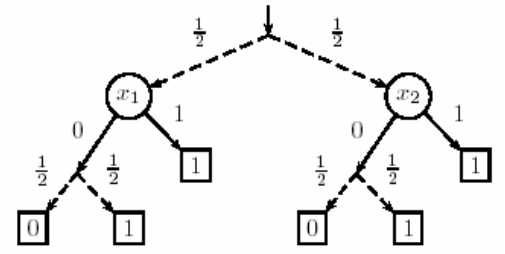
3. att. Varbūtisks lēmumu koks priekš funkcijas OR.

Varbūtiska lēmumu koka vērtība tiek aprēķināta ar noteikto varbūtību.

Pastāv divi varbūtisku lēmumu koku konstruēšanas veidi, ekvivalenti savā starpā:

- Pirmkārt, ir iespējams pievienot kokam virsotnes, kurās ,lai pieņemtu lēmumu, pa kuru zaru iet tālāk, mēt monētu. Kad izkrīt ērglis, rekursīvi aprēķina labo apakškoku, un kreiso apakškoku, kad izkrīt raksts. Ievads x vairs nenosaka precīzi, kura lapa būs sasniegta, bet inducē varbūtības sadalījumu pie visas koka lapu kopas. Tātad, koks dod rezultātu 0 vai 1

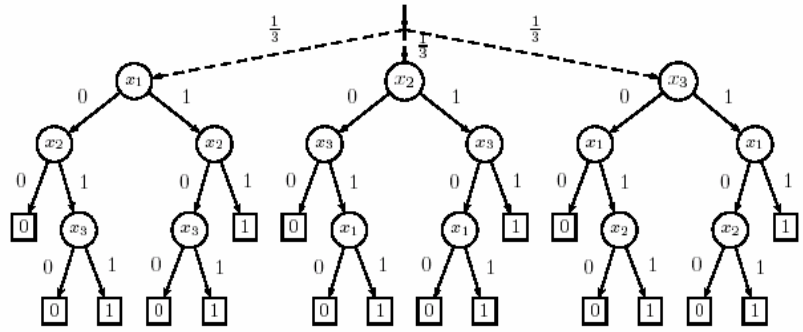
ar noteikto varbūtību. Algoritms attēlā 4 aprēķina funkcijas OR vērtību, tas vienmēr dod rezultātu, bet tas var būt aplams. Ja $x_1 = x_2 = 0$, tad īstais rezultāts 0 producēts ar varbūtību ½. Visos citos gadījumos, kad funkcijas vērtība ir 1, pareizais rezultāts 1 tiek sasniegts ar varbūtību p =1/4 + ½ = 2/3.



4. att. Varbūtisks lēmumu koks priekš funkcijas OR.

Koka sarežģītība ir vaicājumu skaits sliktākajam ievadam un tā sliktāka varbūtība. Ar attēla 4 koku funkcija OR tiek aprēķināta ar 1 vaicājumu ar varbūtību vismaz ½ .

- Otrkārt, ir iespējams noteikt varbūtību sadalījumu μ starp determinētiem lēmumu kokiem. (Tādi ir piemērs no attēliem 3 un 5) Apzīmēsim ar $T=\{T_1,...,T_k\}$ dota varbūtiska lēmumu koka determinētu apakškoku kopu. Atbilstoši varbūtību sadalījumam μ, izvēlās zaru, pa kuru iet. Aiziet pie T_i, novērtē to kā determinētu lēmumu koku. Tipiski šo gadījumu raksturo piemērs no attēla 5. Tas ir varbūtisks lēmumu koks, kurš rēķina funkciju MAJORITY(x_1,x_2,x_3). Funkcija MAJORITY dod 1, ja vieninieku ir vairāk starp x_i, nekā nuļļu, un 0 pretējā gadījumā.



5. att. Varbūtisks lēmumu koks priekš funkcijas MAJORITY.

Šī veida varbūtiska lēmumu koka sarežģītība ir T_i dziļums, kur T_i ir kopas T dziļākais koks, kuram $\mu(T_i)>0$. Piemērā 5 koka sarežģītība ir 3, šis algoritms ir ***eksakts***, jo jebkuram ievadam tas dod pareizo rezultātu ar varbūtību 1.

Definīcija 9. Saka, ka varbūtisks lēmumu koks aprēķina Būla funkciju *f* ar ***ierobežotu kļūdu***, ja visiem $x\in\{0,1\}^n$ koka novērtējums ir vienāds ar funkcijas *f(x)* vērtību ar varbūtību vismaz

½+δ. Ar $R_2(f)$ apzīmē optimāla varbūtiska lēmumu koka sarežģītību, kurš aprēķina *f* ar ierobežotu kļūdu.

2.3. Lēmumu koka sarežģītības novērtējums

Sarežģītību mēri, kuri tika aplūkoti iepriekšējā nodaļā, ir cieši saistīti ar funkcijas *f* dažādu modeļu lēmumu koka sarežģītību. Sarežģītības novērtējumam kvantu un varbūtiskam modelim ir trīs gadījumi atkarībā no tā, cik liela kļūda ir atļauta: nulles, vienpusīga un divpusīga kļūda. Algoritma sarežģītību attiecīgi apzīmē ar R_0, R_1, R_2 varbūtiskā un Q_0, Q_1, Q_2 kvantu gadījumā. Kvantu sarežģītības novērtējumi ir aprakstīti tālāk - nodaļā 4.3.

2.3.1. Determinēts lēmumu koks

Aplūko 2 dažādus D(f) novērtējumus no apakšas:

Teorēma 4. [5] bs(f) ≤ D(f).

Pierādījums. Ja ir dots ievads *x* ar atdalītu jutību bloku kopu $B=\{B_1, ..., B_{bs(f)}\}$, tad determinētam lēmumu kokam jāprasa vismaz viens mainīgais no katra bloka B_i, citādi būtu iespējams mainīt B_i bloka mainīgo vērtības uz pretējām, protams, mainot arī funkcijas vērtību uz pretēju, bet koks to nepamanīs. Tāpēc lēmumu kokam jāveic vismaz *bs(f)* vaicājumi priekš dota *x*.

Sekas: no bs(f) ≤ D(f)(Teorēma 4) un $s(f) \le bs(f) \le n$ (Teorēma 2) seko, ka $s(f) \le bs(f) \le D(f)$.

Teorēma 5. [5] deg(f) ≤ D(f).

Pierādījums. Ņem funkcijas *f* determinēto lēmumu koku ar dziļumu D(f). Ar L apzīmē 1-lapu (lapu ar vērtību 1) un ar x_1, ..., x_r mainīgos ar vērtībām b_1, ..., b_r, kurus satiek virzoties pa šķautnēm pie L. Definē polinomu $p_L(x) = \prod_{i:b_i=1} x_i \prod_{i:b_i=0} (1-x_i)$. Polinomam p_L ir pakāpe

r ≤ D(f), pie kam $p_L(x)=1$, ja lapa L tiek sasniegta uz ievada *x*, un citādi $p_L(x)=0$. Apzīmē ar $p = \sum_L p_L$ -visu tādu p_L summa visām 1-lapām. Tāpēc *p* pakāpe nav lielāka par *D(f),* un $p(x)=1$ tad un tikai tad, ja sasniedz 1-lapu ievadot *x*. Tas arī nozīmē, ka *p* reprezentē *f*.

D(f) novērtējumi no augšas seko no

Teorēma 6. [5] $D(f) \le s(f)bs^2(f) \le bs^3(f)$ un

Teorēma 7. [5] $D(f) \le deg^2(f)bs(f) \le 2deg^4(f)$.

2.3.2. Varbūtisks lēmumu koks

Triviāli novērtējumi ir $R_2(f) \le R_1(f) \le R_0(f) \le D(f)$ .Pirmais netriviāls rezultāts $D(f) = O(R_0(f)^2)$ tika neatkarīgi aprakstīts dažādu autoru darbos, Nisan vispārināja to līdz vienpusīgas kļūdas gadījumam $D(f) = O(R_1(f)^2)$ un līdz divpusīgas kļūdas gadījumam $D(f) = O(R_2(f)^3)$.

Teorēma 8. [5] $\tilde{deg}(f) \le R_2(f)$.

Pierādījums. Pieņem, ka funkcijas *f* varbūtisks lēmumu koks attēlo varbūtības sadalījumu μ virs dažādiem determinētiem lēmumu kokiem T, katra koka dziļums nav lielāks par $R_2(f)$. No (Teorēma 5) deg(f) ≤ D(f) seko, ka katru no $t \in T$ kokiem ir iespējams pierakstīt ka vairāku mainīgo polinomu p_t, kura pakāpe nav lielāka par $R_2(f)$. Polinoms funkcijas f varbūtiskam lēmumu kokam ir vienāds ar $p(x) = \sum_{t \in T} \mu(t)\, p(t)$, kura pakāpe nav lielāka par $R_2(f)$. Polinoms *p* ir ievada x akceptēšanas varbūtība, tātad *p* aproksimē funkciju *f*.

Teorēma 9. [5],[6] $bs(f) \le 3\, R_2(f)$.

Pierādījums. Pieņem, ka algoritmam nepieciešami $R_2(f)$ vaicājumi funkcijas *f* vērtības atrašanai, un ievads *x* sasniedz jutību bloku. S ir mainīgo kopa ar īpašību: $f(x) \ne f(x^S)$. Varbūtībai, ka algoritms uzprasīs S mainīga vērtību, ir jābūt lielākai par 1/3, citādi algoritms neuztvers atšķirību starp ievadiem *x* un x^S ar nepieciešamu varbūtību. Algoritmam ir jāprasa vismaz 1/3 mainīgo vērtības no katra no *bs(f)* bloka, kopīgais vaicājumu skaits ievadam *x* ir vismaz *bs(f)*/3 .

3. KVANTU SKAITĻOŠANA

3.1. Pamatmodelis

Aprakstīsim kvantu skaitļošanas pamatmodeli. Sīkāk tas ir aprakstīts J.Gruskas[1], Nielsen un Chuang[2] grāmatās.

Bits ir klasiskās atmiņas vienība, kurai ir divi stāvokļi - 0 un 1. Kvantu skaitļošanā par vienību uzskata ***kvantu bitu***(quantum bit), jeb ***qubit***, kurš ir vienāds ar divu klasisku vērtību lineāru kombināciju(***superpozīciju***):

$$\alpha_0|0>+\alpha_1|1> \qquad (3.1.1)$$

Kur kompleksi skaitļi α_0 un α_1 ir stāvokļu |0> un |1> amplitūdas. Stāvokļus $|0\rangle,|1\rangle,\ldots,|m-1\rangle$ sauc par bāzes stāvokļiem, katrs m-qubit stāvoklis $|\phi>$ ir 2^m stāvokļu (klasisko m-bitu rindu) superpozīcija:

$$|\phi>= \sum_{i\in\{0,1\}^m} \alpha_i |i>,$$

kur α_i ir kompleksais skaitlis, kuru sauc par bāzes stāvokļa |i> amplitūdu, un izpildās nosacījums $\sum_i |\alpha_i|^2 = 1$. Matemātiski tas nozīmē, ka m-qubit stāvoklis ir 2^m dimensionālas Hilberta telpas vektors ar garumu(normu) 1, un { |0>,...,$|2^m$-1>} ir Hilberta telpas ortonormēta bāze.

Seko dažas definīcijas no lineāras algebras.

Definīcija 10. Matrica A = {a_{ij}} ir ***vienības matrica I***, ja $a_{ij} = \delta_{ij} = \begin{cases} 0, & ja\ i \neq j \\ 1, & ja\ i = j \end{cases}$.

Definīcija 11. Matrica A^{-1} ir matricas A ***inversā***, ja $AA^{-1} = A^{-1}A = I$.

Definīcija 12. Matrica A^T = {b_{ij}} ir matricas A = {a_{ij}} ***transponētā*** matrica, ja tā ir iegūta no matricas A, mainot vietām kolonas un rindas, tas ir $b_{ij} = a_{ji}$.

Definīcija 13. Matrica $\overline{A} = \{b_{ij}\}$ ir matricas $A = \{a_{ij}\}$ kompleksi saistīta matrica, ja $b_{ij} = \overline{a_{ij}}$.

Definīcija 14. Matrica U ir ***unitāra*** tad un tikai tad, ja tās inversā matrica U^{-1} ir U transponēta kompleksi saistīta matrica U*. Unitārā transformācija ir aprakstīta ar unitāro matricu, tas ir U*U = UU* = I.

m-qubit sistēmas stāvoklis mainās, pielietojot unitāras transformācijas un kvantu mērījumu:

1. *Mērījums* . Viena no kvantu mehānikas aksiomām saka, ka veicot m-qubit stāvokļa |ϕ> mērījumu, rezultātā iegūst stāvokli |i> ar varbūtību $|\alpha_i|^2$. $\sum_i |\alpha_i|^2 = 1$, tāpēc varbūtību sadalījums pie klasiskām m-bitu rindām ir derīgs. Pēc mērījuma m-qubit stāvoklis |ϕ> pāriet uz apskatīto bāzes stāvokli |i>, bet visa cita informācija par |ϕ> tiek zaudēta.

2. *Unitāra transformācija.* Aplūkojot 2^m stāvokļa $|\phi>$ amplitūdas kā C^{2^m} vektoru, iegūst jaunu stāvokli $|\psi>=\sum_{i\in\{0,1\}^m}\beta_i|i>$ kā $|\phi>$ un unitāras matricas U reizinājumu :$|\psi>$ $=U|\phi>$, kas arī ir C^{2^m} vektors. Tā kā unitārums ir ekvivalents Eiklīda normas saglabāšanai, jaunais stāvoklis $|\psi>$ arī būs ar īpašību $\sum_i|\beta_i|^2=1$. Citos vārdos, unitāra transformācija pārveido telpu par to pašu telpu citā bāzē, jo matricas U kolonas(rindas) veido ortonormētu bāzi.

3.2 Jauktais stāvoklis

Līdz šīm brīdim viss attiecās tikai uz kvantu tīrajiem stāvokļiem

$$|\phi>=\sum_{i\in\{0,1\}^m}\alpha_i|i> \tag{3.2.1}$$

Tas nav stāvokļa vispārīga forma. Pieņemsim, ka mums ir tīru stāvokļu varbūtību sadalījums, piemēram, |0> ar varbūtību ½ un |1> ar varbūtību ½ . Cits piemērs ir stāvoklis

$$\begin{cases}|+>=\frac{1}{\sqrt{2}}(|0>+|1>), & p=\frac{1}{2}\\ |->=\frac{1}{\sqrt{2}}(|0>-|1>), & p=\frac{1}{2}\end{cases} \tag{3.2.2}$$

Vispārīgi runājot, mēs varam iedomāties jaukto stāvokli kā tīro stāvokļu $|\psi_i>$ kolekciju, kur katram ir piekārtota varbūtība p_i ar nosacījumu $0\le p_i\le 1,\ \sum_i p_i=1$.

Kvantu sistēmas ir grūti izolēt, un tādejādi, tās bieži ir saistītas ar apkārtni. Viens iemesls, kāpēc mēs ievērojam jauktos stāvokļus, ir tas, ka kvantu sistēmas uzvedību un stāvokli dotajā brīdī labāk apraksta jauktais stāvoklis, kurš ir atvasināts no sistēmas un vides stāvokļu kombinācijas.

3.3. Kvantu stāvokļa blīvuma matrica. Blīvuma matricas īpašības

Aplūkosim jauktā kvantu stāvokļa mērījuma rezultātu. Pieņemsim, ka mums ir kvantu stāvokļu lineāra kombinācija: $|\psi_i>$ ar koeficientu (varbūtību) p_i. Katru $|\psi_i>$ reprezentē C^{2^n} telpas vektors, tāpēc operators $|\psi_i><\psi_i|=\psi_i\psi_i^*$ ir 2^n x 2^n matrica

$$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix}\begin{pmatrix} \overline{a_1} & \overline{a_2} & \dots & \overline{a_N} \end{pmatrix}=\begin{pmatrix} a_1\overline{a_1} & a_1\overline{a_2} & \dots & a_1\overline{a_N} \\ a_2\overline{a_1} & a_2\overline{a_2} & \dots & a_2\overline{a_N} \\ \vdots & & & \vdots \\ a_N\overline{a_1} & a_N\overline{a_2} & \dots & a_N\overline{a_N} \end{pmatrix}$$

(3.3.1)

Jauktā stāvokļa $\{p_i, |\psi_i>\}$ blīvuma matrica ir blīvuma matricu summa $\rho=\sum_i p_i |\psi_i><\psi_i|$.

Parādīsim dažus piemērus. Aplūkosim jaukto stāvokli |0> ar varbūtību ½ un |1> ar varbūtību ½.

Tad $|0><0|=\begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}1 & 0\end{pmatrix}=\begin{pmatrix}1 & 0\\0 & 0\end{pmatrix}$, $|1><1|=\begin{pmatrix}0\\1\end{pmatrix}\begin{pmatrix}0 & 1\end{pmatrix}=\begin{pmatrix}0 & 0\\0 & 1\end{pmatrix}$

(3.3.2)

un jauktā stāvokļa blīvuma matrica ir $\rho=\frac{1}{2}|0><0|+\frac{1}{2}|1><1|=\begin{pmatrix}1/2 & 0\\0 & 1/2\end{pmatrix}$

(3.3.3)

Tagad aplūkosim citu jaukto stāvokli $\begin{cases}|+>=\frac{1}{\sqrt{2}}(|0>+|1>), & p=\frac{1}{2}\\ |->=\frac{1}{\sqrt{2}}(|0>-|1>), & p=\frac{1}{2}\end{cases}$

(3.3.4)

$$|+><+|=\frac{1}{2}\begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}1 & 1\end{pmatrix}=\frac{1}{2}\begin{pmatrix}1 & 1\\1 & 1\end{pmatrix},\quad |-><-|=\frac{1}{2}\begin{pmatrix}1\\-1\end{pmatrix}\begin{pmatrix}1 & -1\end{pmatrix}=\frac{1}{2}\begin{pmatrix}1 & -1\\-1 & 1\end{pmatrix}$$

(3.3.5)

Šajā gadījumā blīvuma matrica atkal ir

$$\rho=\frac{1}{2}|+><+|+\frac{1}{2}|-><-|=\begin{pmatrix}1/2 & 0\\0 & 1/2\end{pmatrix}$$

(3.3.6)

Svarīgi ir atzīmēt, ka blīvuma matricas ir identiskas, kaut gan sākumstāvokļi bija dažādi, tas nozīmē, ka dažādiem jauktajiem stāvokļiem atbilst viena blīvuma matrica. Tas nozīmē, ka stāvokļi ir atšķirīgi, ja blīvuma matricas nav vienādas.

Teorēma 10. [7] Pieņemsim, ka notiek jauktā stāvokļa mērījums ortonormētā bāzē $\{|\beta_k>\}$, tad rezultāts ir $|\beta_k>$ ar varbūtību $<\beta_k|\rho|\beta_k>$.

Pierādījums. Apzīmēsim varbūtību nomērīt $|\beta_k>$ ar Pr[k]. Tad

$$\Pr[k] = \sum_j p_j |< \psi_j | \beta_k >|^2 =$$

$$\sum_j p_j < \beta_k | \psi_j >< \psi_j | \beta_k >=$$

$$\left\langle \beta_k \left| \sum_j p_j | \psi_j >< \psi_j \right| \beta_k \right\rangle =$$

$$=< \beta_k | \rho | \beta_k >$$

Sekas. Ja notiek jauktā stāvokļa $\{p_i, |\psi_i>\}$ mērījums standartā bāzē, tad $Pr[k]=\rho_{k,k}$, tas ir blīvuma matricas diagonāles elementi.

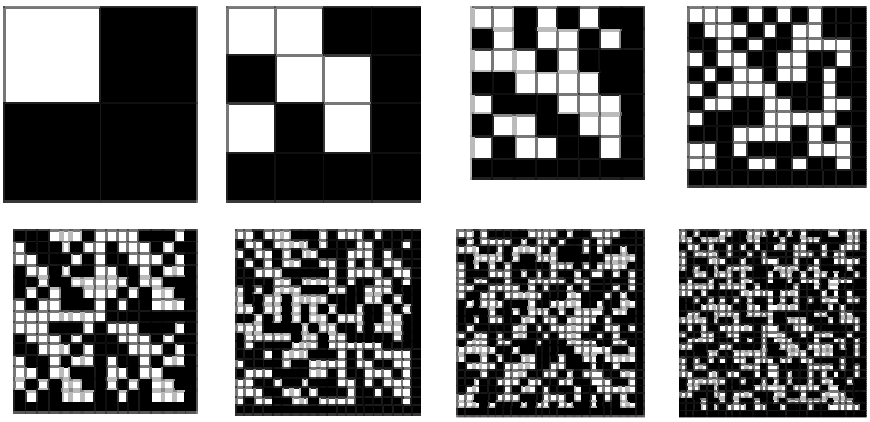
Blīvuma matricas īpašības:

1. tr[ρ]=1. Seko no varbūtību Pr[k], kas ir blīvuma matricas diagonāles elementi, summas $\sum_k \Pr[k] = 1 = tr[\rho]$.
2. ρ ir Ermita matrica, tas ir ρ $\rho^* = \rho^*$ ρ. Šis fakts seko no tā, ka ρ ir Ermita operatoru($|\psi_i><\psi_i|$) summa $((\psi_i \psi_i^*)^* = \psi_i \psi_i^*)$.
3. ρ īpašvērtības ir nenegatīvas. Vispirms, Ermita matricas īpašvērtības ir reālas. Pieņemsim, ka λ un |e> ir īpašvērtības – īpašvektora pāris. Ja mēs mēram īpašvektoru bāzē, $\Pr[e] =< e | \rho | e >= \lambda < e | e >= \lambda$. Varbūtības ir nenegatīvas, tāpēc arī $\lambda \geq 0$.

3.4. Adamāra matricas

Definīcija 15. [9]***Adamāra matrica*** ir kvadrātiskā matrica, kura satur tikai 1 un -1, pie kam katrās divās blakus rindās (kolonās) puse no blakus elementiem (i,k) un (j,k)(vai attiecīgi (i,k) un (i,m)) ir vienādi, bet puse ir pretējie, neskaitot matricas robežu L-formā, kura sastāv no tīriem vieniniekiem.

Ir acīmredzams, ka mainot rindu(kolonu) secību, Adamāra matricas īpašība nepazūd. N x N Adamāra matricai ir n(n-1)/2 „-1" un n(n+1)/2 „1". Ja trotuāra melniem rūtiņiem ir vērtība „1" un baltiem ir „-1", tad Adamāra matricas izskatās šādi:



6. att. Adamāra matricas.

Adamāra matrica ar pakāpi (4n+4) atbilst Adamāra dizainam (4n+3, 2n+1, n).

Ja ir dotas H_n un H_m, tad H_{nm} var iegūt, aizvietojot visus H_m matricas vieniniekus ar H_n un visus „-1” ar $-H_n$, piemēram

$$H_2 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ -H_2 & H_2 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \\ -\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

H_8 var līdzīgi uzkonstruēt no H_4, kaut gan priekš $n \le 100$, Adamāra matricas ar n= 12, 20, 28, 36, 44, 52, 60, 68, 76, 84, 92 un 100 nevar uzbūvēt no zemākas pakāpes matricām.

Adamāra matricas ir ļoti svarīgs un bieži lietojāms instruments kvantu algoritmu konstruēšanas procesā.

3.5. Lineāru vienādojumu sistēma: Kramera kārtula

Kramera kārtula ļauj izteikt LVS(lineāru vienādojumu sistēmas) atrisinājumu vispārīgā formā.

Dotai LVS

$$\begin{cases} a_1 x + b_1 y + c_1 z = d_1 \\ a_2 x + b_2 y + c_2 z = d_2 \\ a_3 x + b_3 y + c_3 z = d_3, \end{cases}$$

(3.5.1)

Definē determinantu

$$D \equiv \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}.$$

(3.5.2)

No determinanta īpašībām, ja x ir konstante, tad reizinot kolonu ar x mainām D uz xD:

$$x\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = \begin{vmatrix} a_1 x & b_1 & c_1 \\ a_2 x & b_2 & c_2 \\ a_3 x & b_3 & c_3 \end{vmatrix}.$$

(3.5.3)

Cita determinantu īpašība ļauj pieskaitīt kolonai citu kolonu, reizinātu ar konstanti:

$$x\,D = \begin{vmatrix} a_1 x + b_1 y + c_1 z & b_1 & c_1 \\ a_2 x + b_2 y + c_2 z & b_2 & c_2 \\ a_3 x + b_3 y + c_3 z & b_3 & c_3 \end{vmatrix} = \begin{vmatrix} d_1 & b_1 & c_1 \\ d_2 & b_2 & c_2 \\ d_3 & b_3 & c_3 \end{vmatrix}.$$

(3.5.4)

Ja $d = 0$, tad izteiksme reducējās uz $xD = 0$, tad sistēmai ir nedeģenerēts atrisinājums tad un tikai tad, ja $D = 0$. šajā gadījumā LVS ir atrisinājumu kopa.

Ja $d \neq 0$ un $D = 0$, tad sistēmai nav atrisinājumu. Ja $d \neq 0$ un $D \neq 0$, tad atrisinājumu var iegūt no

$$x = \frac{\begin{vmatrix} d_1 & b_1 & c_1 \\ d_2 & b_2 & c_2 \\ d_3 & b_3 & c_3 \end{vmatrix}}{D}, \; y = \frac{\begin{vmatrix} a_1 & d_1 & c_1 \\ a_2 & d_2 & c_2 \\ a_3 & d_3 & c_3 \end{vmatrix}}{D} \quad z = \frac{\begin{vmatrix} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ a_3 & b_3 & d_3 \end{vmatrix}}{D}.$$

(3.5.5)

Procedūru var vispārināt n-mainīgo un n-vienādojumu sistēmai:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix}, \; D \equiv \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}.$$

(3.5.6)

Ja $d = 0$, tad sistēmai ir nedeģenerēts atrisinājums tad un tikai tad, ja $D = 0$.

Ja $d \neq 0$ un $D = 0$, tad sistēmai nav atrisinājumu. Ja $d \neq 0$ un $D \neq 0$,

$$D_k \equiv \begin{vmatrix} a_{11} & \cdots & a_{1\,(k-1)} & d_1 & a_{1\,(k+1)} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{n\,(k-1)} & d_n & a_{n\,(k+1)} & \cdots & a_{nn} \end{vmatrix}.$$

(3.5.7)

$X_k = D_k/D$, kur k mainās no 1 līdz n.

4. KVANTU VAICĀJOŠAIS ALGORITMS

4.1. Melnas kastes modelis

Vaicājošais algoritms iespējami ir pats vienkāršākais Būla funkciju rēķināšanas modelis. Šajā modelī ievads $x_1, x_2, ..., x_n$ atrodas melnajā kastē un var būt pieejams caur jautājumiem par x_i vērtībām. Vaicājošam algoritmam jāspēj noteikt funkcijas f(X) vērtība pareizi jebkuram patvaļīgam ievadam X, kurš atrodas melnajā kastē. Algoritma sarežģītības mērs ir jautājumu skaits, kurus uzdod melnai kastei par mainīgo vērtībām. Klasiskā šī algoritma versija ir pazīstama kā lēmumu koks. Sīkāku aprakstu var atrast Buhrman un de Wolf rakstā [5]. Šeit mēs uzskatam, ka Būla funkcijas vērtība tiek rēķināta ar kvantu vaicājošo algoritmu. Sīkāku aprakstu vajag meklēt Ambaiņa rakstā [10] un Gruskas un de Wolf grāmatās[1], [11].

4.2. Skaitļošanas process

Kvantu skaitļošanas process ar T vaicājumiem ir unitāru transformāciju virkne:

$$U_0 \to Q_0 \to U_1 \to Q_1 \to ... \to U_T \to Q_{T-1} \to U_T \tag{4.2.1}$$

U_i ir patvaļīga unitāra transformācija, kura nav atkarīga no ievada bitiem $x_1, x_2, ..., x_n$.

Q_i ir vaicājums. Skaitļošana sākas stāvoklī $|\vec{0}\rangle$, pēc tam pielieto transformāciju virkni

U_0, Q_0,…, Q_{T-1}, U_T un uztaisa mērījumu beigu stāvoklim.

Pastāv dažādas ekvivalentas kvantu vaicājošā algoritma definīcijas. Pats galvenais ir izvelēties piemērotu vaicājumu melnas kastes definīciju, precīzāk, jautājumu uzdošanas veidu un no orākula saņemtu atbilžu formu.

Aprakstīsim precīzāk, kā definē kvantu vaicājošo algoritmu, kā arī paskaidrosim notāciju, kuru izmantosim arī tālāk.

Katram kvantu vaicājošam algoritmam ir sekojošie parametri:

1. *Unitāras transformācijas.* Definē visas unitāras transformāciju un orākulu matricas.

Vispārīgā formā algoritms ar *T* vaicājumiem ir $|\vec{0}\rangle \to U_0 \to Q_1 \to ... \to Q_T \to U_N \to [QM]$,

kur $|\vec{0}\rangle$ ir sākumstāvoklis, *[QM]* – kvantu mērījums.

T-vaicājumu kvantu lēmumu koks atbilst vienai lielai transformācijai $U_N Q_T ... Q_1 U_0$, kur U_i ir fiksētas neatkarīgas no *X* unitāras transformācijas. No algebras ir zināms, ka unitāru matricu reizinājums arī ir unitāra matrica. Beigu stāvoklis $A|\vec{0}>$ ir atkarīgs no ievada *X* tikai caur T O-vaicājumu transformācijas pielietojumiem.

Mūsu ērtībai stāvokļu vektoru un algoritma plūsmai aprakstam mēs šeit izmantosim *bra* notāciju. Kvantu mehānika izmanto tādu notāciju stāvokļu vektoru apzīmējumam[2]:

$$\textit{Ket}\text{ notācija: } |\psi\rangle = \begin{pmatrix} \alpha_1 \\ ... \\ \alpha_n \end{pmatrix} \qquad \textit{Bra}\text{ notācija: } <\psi| = |\psi>^* = (\overline{\alpha_1},...,\overline{\alpha_n}) \tag{4.2.2}$$

Algoritma apraksts *bra* notācijā var būt pārveidots uz *ket* notāciju, aizvietojot katru unitāras transformācijas matricu ar transponētu kompleksi saistītu:

Kvantu vaicājošā algoritma plūsma *bra* notācijā: $\langle\psi| = \langle\overline{0}|U_0 Q_0 ... Q_{N-1} U_N$ (4.2.3)

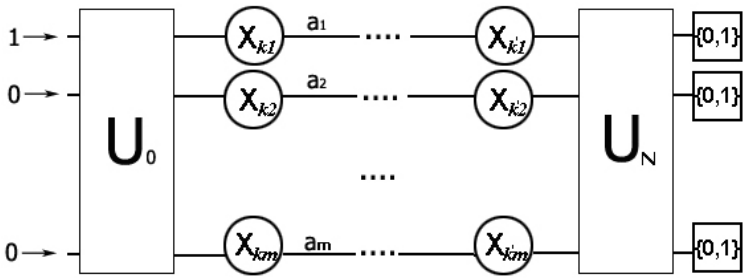
Kvantu vaicājošā algoritma plūsma *ket* notācijā: $|\psi> = U_N^* Q_{N-1}^* ... Q_0^* U_0^* |0>$ (4.2.4)

2. *Vaicājumi*. Pastāv dažādas vaicājuma transformācijas definīcijas, kuras ir ekvivalentas sava starpā. Ja uz ieejas mums ir stāvoklis $|\psi\rangle = \sum_i a_i |i\rangle$, tad izejā ir $|\phi\rangle = \sum_i (-1)^{x_k} a_i |i\rangle$, kur katrai amplitūdai mēs varam piekārtot patvaļīgu mainīgo x_k.

Ja ir dots kvantu stāvoklis ar *m* amplitūdām $\langle\psi| = (\alpha_1, \alpha_2, ..., \alpha_m)$, tad n-argumentu funkcijai definē vaicājumu (quantum query) $QQ_i = (\alpha_1 \equiv k_1, ..., \alpha_m \equiv k_m)$, kur *i* ir vaicājuma numurs, $k_j \in \{1..n\}$ ir mainīga numurs, kura vērtība ir prasīta. Ja $x_{k_j} = 1$, vaicājums maina *j*-tās amplitūdas zīmi uz pretēju, citādi, zīme paliek nemainīga. Kvantu vaicājuma $QQ_i = (\alpha_1 \equiv k_1, ..., \alpha_m \equiv k_m)$ unitāra matrica ir:

$$QQ = \begin{pmatrix} (-1)^{k1} & 0 & ... & 0 \\ 0 & (-1)^{k2} & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & (-1)^{km} \end{pmatrix} \tag{4.2.5}$$

3. *Mērījums*. Katra beigu kvantu stāvokļa amplitūda atbilst algoritma izejai. Piekārtosim funkcijas vērtību katrai izejai un apzīmēsim to ar $QM = (\alpha_1 \equiv k_1, ..., \alpha_m \equiv k_m)$, kur $k_i \in \{0,1\}$ (QM nozīmē quantum measurement). Algoritmu nostrādā uz ievada *X*, rezultāts ir j ar varbūtību, vienādu ar visu amplitūdu vērtību kvadrātu summu, kuras atbilst funkcijas vērtībai *j*. 7. attēls parāda algoritmu vispārīgā formā.



7. att. Kvantu vaicājoša algoritma grafiskais attēlojums.

4.3. Kvantu algoritma sarežģītība

Kvantu algoritma sarežģītība ir atkarīga no vaicājumu skaita, kurš ir nepieciešams, lai aprēķinātu funkcijas vērtību sliktākā gadījuma ievadam *X*.

Atšķirībā no klasiska, determinēta vai varbūtiska lēmumu kokiem, kvantu algoritms vairs nav koks. Termins tiek lietots, jo kvantu algoritmi vispārina klasiskus kokus – pilnīgi simulē tos.

Ir pierādītas sakarības [5]: $Q_2(f) \le R_2(f) \le D(f) \le n$ un $Q_2(f) \le Q_E(f) \le D(f) \le n$ visām f, no tām seko, ka funkcijas kvantu sarežģītība var būt vienāda vai mazāka par determinēto un varbūtisku sarežģītību, kas neapšaubāmi ir izdevīgi priekš rēķināšanas.

Precīzāka informācija par šo tēmu var tikt atrasta H.Buhrman un R.de Wolf rakstā [5].

Saka, ka kvantu lēmumu koks ***izrēķina funkciju f precīzi***, ja $\forall x \in \{0,1\}^n$ izeja ir vienāda ar f(x) ar varbūtību 1. Ar $Q_E(f)$ apzīmē optimāla kvantu lēmumu koka vaicājumu skaitu, kurš aprēķina *f* precīzi

Saka, ka koks ***aprēķina f ar ierobežotu kļūdu***, ja $\forall x \in \{0,1\}^n$ izeja ir vienāda ar f(x) ar varbūtību ne mazāku par 1/2. Ar $Q_p(f)$ apzīmē funkcijas *f* kvantu vaicājošā algoritma vaicājumu skaitu, kurš izdod pareizu atbildi ar varbūtību *p*.

Tāpat, kā varbūtiskā gadījumā, pastāv triviālas sakarības[2], [12], [13]

$Q_2(f) \le Q_1(f) \le Q_0(f) \le Q_E(f)$ un

$Q_2(f) \le R_2(f), Q_1(f) \le R_1(f), Q_0(f) \le R_0(f), Q_E(f) \le D(f)$.

Beals et al. [12] parādīja, ka $D(f) = O(Q_2(f)^6)$, Buhrman et al.[13] parādīja, ka $D(f) = O(Q_1(f)^4)$,[3] $D(f) \in O(Q_E(f)^4)$

4.4. Kvantu ķēdes jaukto stāvokļu sistēmā

Skaitļošanas procesā ir atļauts izmantot tikai unitāras transformācijas un stāvoklis ir tīrais. Skaitļošanas beigās tiek pielietota transformācija, kas nav unitāra, precīzāk, mērījums, stāvoklis kļūst par varbūtību sadalījumu starp tīriem stāvokļiem – par jaukto stāvokli. Tas parāda, ka kvantu fizikā visas operācijas nav unitāras un stāvokļiem nav obligāti jābūt tīriem.

Rakstā [2] ir definētas kvantu ķēdes, kurām ir atļauts atrasties vispārīgā stāvoklī, tas ir jauktajā, un mainīt sistēmas stāvokli ar jebkuru kvantu operāciju.

Teorēma 11. [8] Modelim, kurā izmanto kvantu ķēdes un jauktos sistēmas stāvokļus, skaitļošanas jauda ir polinomiāli ekvivalenta standartam unitāram modelim. Galvenās problēmas, kas ir sarežģītas, neiespējamas parastajā modelī, pazūd modelī ar kvantu jauktajiem stāvokļiem.

- *Mērījums skaitļošanas vidū*: ir atzīmēts, ka kvantu skaitļošanā nav aizliegts pielietot mērījumu transformācijas skaitļošanas procesa vidū, kaut gan sistēmas stāvoklis kļūst par jaukto, kas nebija atļauts standartā modelī.
- *Troksnis un traucējumi*: tie ir galvenie šķēršļi, kas parādās īstenojot kvantu datoru ierīces. Pamatproblēma saistot kvantu fizikas un skaitļošanas modeļus ir tas fakts, ka kvantu troksnis nav unitāras operācijas, tātad, veido jaukto stāvokli no tīra stāvokļa.

Dabiskā funkcija, kuru atgriež kvantu dators, ir varbūtiskā funkcija: ieejai X izeja ir sadalīta atbilstoši D_X, atkarīgs no ieejas sadalījums. Pielietojot šādu varbūtisku funkciju kā subrutīnu, stāvoklis ir ietekmēts neunitārā veidā. Konceptuāli sarežģīti ir pielietot varbūtiskas subrutīnas kvantu superpozīcijai, jo nav skaidrs, kādai ir jābūt dabiskai definīcijai. Precīzāk jautājums par kvantu ķēdēm ir aplūkots rakstā [8].

4.5. Kvantu vaicājošais algoritms jaukto stāvokļu sistēmā

Kvantu vaicājoša algoritma definīcija paliek tāda paša, ka tīru stāvokļu gadījumā: algoritms ir unitāru transformāciju un orākulu virkne, kuru pielieto kvantu jauktajam stāvoklim.
Mērījums notiek pa katru tīru stāvokli, saskaitot rezultātus, ņemot vērā tīru stāvokļu varbūtības, ar kurām tie piedalās šajā jauktajā stāvoklī.
Ja stāvoklis no jauktā kļūst tīrais, pielietojot mērījuma transformāciju, atgriezties uz sākotnējo jaukto jau vairs nav iespējams, piemēram, viena kubita kvantu sistēmā divu stāvokļu blīvuma matricas:

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \xrightarrow{QM} \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}.$$ Apgrieztā transformācija nav iespējama.

5. EKSAKTIE KVANTU VAICĀJOŠIE ALGORITMI

5.1. Eksaktie kvantu vaicājošie algoritmi 2n-mainīgo Būla funkcijām ar sarežģītību $Q_E(f) = n$ pret $D(f) = 2n$

Šajā nodaļā mēs apskatīsim eksakto kvantu vaicājošo algoritmu kopu ar attiecīgu Būla funkciju determinētu sarežģītību divreiz lielāku par eksakta kvantu algoritma sarežģītību. Aprakstot algoritmu, ir nepieciešams norādīt

1. transformāciju matricas U_i un vaicājumu matricas Q_i, ar kuru palīdzību notiek pāreja no viena kvantu stāvokļa uz nākamo.
2. kvantu stāvokļu blīvuma matricas $\rho_0, \rho_1, \ldots, \rho_n$

Būla funkcija: 4-mainīgo Būla funkcija T_4 ir uzdota ar patiesuma tabulu

x1	x2	y1	y2	T_4(XY)
0	0	0	0	1
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1
citādi				0

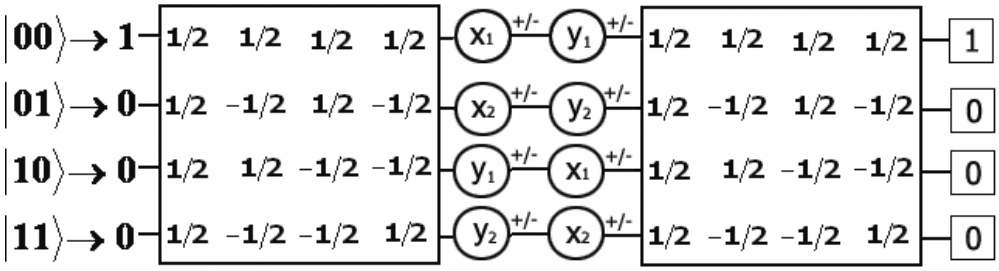
1. tab. Būla funkcija T_4

Citādi funkciju var uzdot sekojoši:

$$T_4(x1, x2, y1, y2) = 1 \Leftrightarrow ((x1 = y1 \; un \; x2 = y2) vai (x1 \neq y1 \; un \; x2 \neq y2)) un \; | \, XY \, | \bmod 2 = 0$$

Determinēta sarežģītība: $D(T_4) = 4$, tas seko no funkcijas jutības uz jebkuras ieejas rindas XY, tādas, ka T_4(XY) = 1 (piemēram, XY = 0000, XY = 0011).

Algoritms. Eksakts kvantu vaicājošais algoritms priekš Būla funkcijas *T_4,* kas rēķina funkciju uzdodot 2 jautājumus, ir redzams Attēlā 8.



8. att. Būla funkcijas T_4 eksakts kvantu vaicājošais algoritms.

Apzīmējumu ērtībai sadalīsim ieejas virkni divās daļās X un Y attiecīgi. Lai ar |XY| apzīmē ieejas virknes XY = <x1,x2,y1,y2> Heminga svaru, tas ir '1' skaits virknē.

Rēķināšanas process ir sekojoša unitāru transformāciju un vaicājumu virkne:

$$\rho_0 \xrightarrow{U_0} \rho_1 \xrightarrow{Q_1 Q_2} \rho_2 \xrightarrow{U_1} \rho_3 \xrightarrow{[M]} \rho_{final} \tag{5.1.1}$$

[M] ir mērījums, kurš paziņo stāvokļa |00> varbūtību $|\alpha_i|^2$,

$$H4 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}; \quad \rho_0 = |00\rangle\langle 00| = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$U_0 = U_1 = H4 \tag{5.1.2}$$

$$Q_1 = \begin{pmatrix} (-1)^{x1} & 0 & 0 & 0 \\ 0 & (-1)^{x2} & 0 & 0 \\ 0 & 0 & (-1)^{y1} & 0 \\ 0 & 0 & 0 & (-1)^{y2} \end{pmatrix}; \quad Q_2 = \begin{pmatrix} (-1)^{y1} & 0 & 0 & 0 \\ 0 & (-1)^{y2} & 0 & 0 \\ 0 & 0 & (-1)^{x1} & 0 \\ 0 & 0 & 0 & (-1)^{x2} \end{pmatrix}. \tag{5.1.3}$$

Ir acīmredzams, ka $Q_E(T_4) = 2$.

Skaitļošanas procesā pēc mērījuma mēs iegūsim 0, ja $T_4(XY) = 0$, un 1, ja $T_4(XY) = 1$.

Izradās, ka iepriekš aprakstītu ideju var pielietot arī funkcijai ar lielāku mainīgo skaitu.

Piemēram, definēsim funkciju T_6 no 6 mainīgajiem

Būla funkcija: uzdota ar patiesuma tabulu

XY	T_6	XY	T_6
000000	1	100001	1
000101	1	100100	1
001001	1	101000	1
001100	1	101101	1
010010	1	110011	1
010111	1	110110	1
011011	1	111010	1
011110	1	111111	1
citādi	0		

2. tab. Būla funkcija T_6

Determinēta sarežģītība: $D(T_6) = 6$, tas seko no funkcijas jutības uz jebkuras ieejas rindas XY, tādas, ka $T_6(XY) = 1$ (piemēram, XY = 000000).

Algoritms. Eksakts kvantu vaicājošs algoritms priekš Būla funkcijas T_6 ir sekojoša unitāru transformāciju un vaicājumu virkne.

$$\rho_0 \xrightarrow{U_0} \rho_1 \xrightarrow{Q_1Q_2Q_3} \rho_2 \xrightarrow{U_1} \rho_3 \xrightarrow{[M]} \rho_{final}, \text{ kur}$$

(5.1.5)

$$Q_1 = \begin{pmatrix} (-1)^{x1} & 0 & 0 & 0 \\ 0 & (-1)^{x2} & 0 & 0 \\ 0 & 0 & (-1)^{y1} & 0 \\ 0 & 0 & 0 & (-1)^{y2} \end{pmatrix}; \quad Q_2 = \begin{pmatrix} (-1)^{x2} & 0 & 0 & 0 \\ 0 & (-1)^{x3} & 0 & 0 \\ 0 & 0 & (-1)^{y2} & 0 \\ 0 & 0 & 0 & (-1)^{y3} \end{pmatrix};$$

$$Q_3 = \begin{pmatrix} (-1)^{y1} & 0 & 0 & 0 \\ 0 & (-1)^{y3} & 0 & 0 \\ 0 & 0 & (-1)^{x1} & 0 \\ 0 & 0 & 0 & (-1)^{x3} \end{pmatrix}$$

(5.1.6)

Ir acīmredzams, ka $Q_E(T_6) = 3$. Mēģināsim vispārināt šo ideju.

Teorēma 12. 2n-mainīgo Būla funkcijai T_{2n} eksistē kvantu vaicājošs algoritms, kurš rēķina funkciju ar sarežģītību $Q_E(T_{2n}) = n$ pret $D(T_{2n}) = 2n$.

Pierādījums.

Algoritma vispārinājuma T_{2n} eksistenci pierādīsim pēc indukcijas ar bāzi - funkciju T_4 un T_6 algoritmiem. Mēs pielietojām sākumstāvoklim šādu transformāciju virkni

$\rho_0 \xrightarrow{U_0QU_1[M]} \rho_{final}$, kur Q ir vaicājumu virkne. Piemēram, T_6 funkcijas gadījumā Q ir

$$\begin{pmatrix} (-1)^{X1+X2+Y1} & 0 & 0 & 0 \\ 0 & (-1)^{X2+X3+Y3} & 0 & 0 \\ 0 & 0 & (-1)^{X1+Y1+Y2} & 0 \\ 0 & 0 & 0 & (-1)^{X3+Y2+Y3} \end{pmatrix}.$$

(5.1.7)

Šeit ir ērtāk attēlot vaicājumu virkni Q matricas veidā, kur i-tais vaicājums ir atzīmēts kā matricas Q i-tā kolonā.

Piemēram, Q T_6 funkcijas algoritmam ir $Q = \begin{pmatrix} x1\ x2\ y1 \\ x2\ x3\ y3 \\ y1\ y2\ x1 \\ y2\ y3\ x3 \end{pmatrix}$.

(5.1.8)

Vispārinājums. Vaicājumu matrica Q funkcijas T_{2n} algoritmam ir viena no sekojošām matricām.

Ja n ir pāra skaitlis, tad $Q_{even} = \begin{pmatrix} x_1 & \dots & x_{\frac{n}{2}} & y_1 & \dots & y_{\frac{n}{2}} \\ x_{\frac{n}{2}+1} & \dots & x_n & y_{\frac{n}{2}+1} & \dots & y_n \\ y_1 & \dots & y_{\frac{n}{2}} & x_1 & \dots & x_{\frac{n}{2}} \\ y_{\frac{n}{2}+1} & \dots & y_n & x_{\frac{n}{2}+1} & \dots & x_n \end{pmatrix}$,

(5.1.9)

Ja n ir nepāra, tad $Q_{odd} = \begin{pmatrix} x_1 & \dots & x_{\left\lfloor \frac{n}{2} \right\rfloor+1} & y_1 & \dots & y_{\left\lfloor \frac{n}{2} \right\rfloor} \\ x_{\left\lfloor \frac{n}{2} \right\rfloor+1} & \dots & x_n & y_{\left\lfloor \frac{n}{2} \right\rfloor+2} & \dots & y_n \\ y_1 & \dots & y_{\left\lfloor \frac{n}{2} \right\rfloor+1} & x_1 & \dots & x_{\left\lfloor \frac{n}{2} \right\rfloor} \\ y_{\left\lfloor \frac{n}{2} \right\rfloor+1} & \dots & y_n & x_{\left\lfloor \frac{n}{2} \right\rfloor+2} & \dots & x_n \end{pmatrix}$

(5.1.10)

Algoritms, kurš rēķina Būla funkciju T_{2n}, dod amplitūdu virkni <a_1, a_2, a_3, a_4 >. Lai algoritms būtu eksakts, virknei <a_1, a_2, a_3, a_4 > jāpilda nosacījums: viena no a_i (i = 1..4) ir 1 vai -1, citas amplitūdas ir vienādas ar nulli. Ņemot vērā Lemmas 2 apgalvojumu, pēc vaicājumu virknes Q mēs iegūsim tādu amplitūdu sadalījumu, ka pēc unitāras transformācijas U_1 pielietojuma mums ir augstāk prasīts amplitūdu sadalījums ar vienīgu $|a_i| = 1$ un pārējas amplitūdas vienādas ar nulli.

Lemma 2. Pēc vaicājumu virknes Q amplitūdu sadalījumā <a_1, a_2, a_3, a_4> katra $|a_i| = \frac{1}{2}$ un negatīvu a_i vērtību ir pāra skaitlis, piemēram, $(\frac{1}{2},\frac{1}{2},-\frac{1}{2},-\frac{1}{2}),(-\frac{1}{2},\frac{1}{2},\frac{1}{2},-\frac{1}{2})$.

Pierādījums.

Funkcijām T_4 un T_6 lemmas nosacījums izpildās, ņemsim tos par indukcijas bāzi. Pieņemsim, ka funkcija T_{2n} dod nepieciešamo amplitūdu sadalījumu pēc vaicājuma Q pielietojuma: negatīvu a_i ir pāra skaits. Vajag pierādīt, ka palielinot mainīgo skaitu –

pievienojot 2 jaunus mainīgos x_{n+1} un y_{n+1} – rodas jauna funkcija $T_{2(n+1)}$ ar atbilstošo eksakto kvantu algoritmu.

Ja n ir pāra skaitlis, tad divu mainīgo pievienošana ir ekvivalenta vaicājumu matricas Q_{even} aizstāšanai ar Q'_{even} vaicājumu matricu.

$$Q'_{even} = \begin{pmatrix} x_1 & \dots & x_{\frac{n}{2}+1} & y_1 & \dots & y_{\frac{n}{2}} \\ x_{\frac{n}{2}+1} & \dots & x_{n+1} & y_{\frac{n}{2}+2} & \dots & y_{n+1} \\ y_1 & \dots & y_{\frac{n}{2}+1} & x_1 & \dots & x_{\frac{n}{2}} \\ y_{\frac{n}{2}+1} & \dots & y_{n+1} & x_{\frac{n}{2}+2} & \dots & x_{n+1} \end{pmatrix}, \quad Qdif = \begin{pmatrix} x_{\frac{n}{2}+1} \\ x_{n+1} y_{\frac{n}{2}+1} y_{n+1} \\ y_{\frac{n}{2}+1} \\ y_{n+1} x_{\frac{n}{2}+1} x_{n+1} \end{pmatrix} \tag{5.1.11}$$

Q_{dif} ir vaicājumu starpība, tas ir izpildīt Q_{even} un Q_{dif} pēc kārtas ir ekvivalenti Q'_{even} izpildīšanai.

Ir viegli pārbaudīt, ka Q_{dif} vaicājums maina pāra skaita amplitūdu vērtības uz pretējām, tādējādi, saglabājot faktu, ka negatīvu amplitūdu ir pāra skaits, arī vaicājumam Q'_{even} .

Gadījumā, ja n ir nepāra, pievienot divus mainīgos ir ekvivalenti vaicājumu matricas Q_{odd} aizstāšanai ar Q'_{odd} vaicājumu matricu.

.

$$Q'_{odd} = \begin{pmatrix} x_1 & \dots & x_{\left\lfloor \frac{n}{2} \right\rfloor+1} & y_1 & \dots & y_{\left\lfloor \frac{n}{2} \right\rfloor+1} \\ x_{\left\lfloor \frac{n}{2} \right\rfloor+2} & \dots & x_{n+1} & y_{\left\lfloor \frac{n}{2} \right\rfloor+2} & \dots & y_{n+1} \\ y_1 & \dots & y_{\left\lfloor \frac{n}{2} \right\rfloor+1} & x_1 & \dots & x_{\left\lfloor \frac{n}{2} \right\rfloor+1} \\ y_{\left\lfloor \frac{n}{2} \right\rfloor+2} & \dots & y_{n+1} & x_{\left\lfloor \frac{n}{2} \right\rfloor+2} & \dots & x_{n+1} \end{pmatrix}, \quad Qdif = \begin{pmatrix} y_{\left\lfloor \frac{n}{2} \right\rfloor+1} \\ x_{\left\lfloor \frac{n}{2} \right\rfloor+1} x_{n+1} y_{n+1} \\ x_{\left\lfloor \frac{n}{2} \right\rfloor+1} \\ y_{\left\lfloor \frac{n}{2} \right\rfloor+1} y_{n+1} x_{n+1} \end{pmatrix} \tag{5.1.12}$$

Tāpat ka iepriekšējā gadījumā, Q_{dif} maina amplitūdas pēc vaicājuma Q_{odd} pielietošanas tā, lai tas būtu ekvivalenti Q'_{odd} izpildīšanai. Ir viegli pārbaudīt, ka Q_{odd} vaicājums maina pāra skaita amplitūdu vērtības uz pretējām, tādējādi, saglabājot faktu, ka negatīvu amplitūdu ir pāra skaits, arī vaicājumam $Q'_{odd.}$

Mēs pierādījām, ka var izmainīt eksaktu algoritmu, kurš rēķina funkciju T_{2n} tā, lai tas rēķinātu funkciju $T_{2(n+1)}$, aizvietojot vaicājumu virknes Q_{even} par Q'_{even} vai Q_{odd} par $Q'_{odd.}$

Secinājums. Eksistē bezgalīga eksakto kvantu vaicājošu algoritmu kopa, kas rēķina attiecīgas Būla funkcijas T_{2n} ar *n* vaicājumiem, kamēr determinēti nepieciešami *2n* vaicājumi, kas ir noteikts ar funkcijas T_{2n} jutību uz viena bita izmaiņu ieejas virknē 00...0.

5.2. SYMMETRY$_3$ funkcija

Būla funkcija. *SYMMETRY$_3$(X)* = 1 tad un tikai tad, ja X ∈ {000, 010, 101, 111}.

Determinētā sarežģītība. Ja ieejas virknei 000 maina otra bita vērtību, rezultāts paliek nemainīgs, mainot pārējus bitus mainās arī funkcijas vērtība, tāpēc akceptējošās ieejas jutība ir 2, kas arī nosaka determinētas sarežģītības novērtējumu, D(f) = 2, kvantu eksakta algoritma sarežģītība ir 1.

Kvantu algoritms. Eksakts kvantu vaicājošs algoritms Būla funkcijai SYMMETRY$_3$ ir sekojoša unitāru transformāciju un vaicājumu virkne:

$$\rho_0 \xrightarrow{U} \rho_1 \xrightarrow{Q} \rho_2 \xrightarrow{U} \rho_3 \xrightarrow{[M]} \rho_{final} \tag{5.2.1}$$

Mērījums [M] paziņo stāvokļa |00> varbūtību.

$$U = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}; \quad Q = \begin{pmatrix} (-1)^{x1} & 0 & 0 & 0 \\ 0 & (-1)^{x2} & 0 & 0 \\ 0 & 0 & (-1)^{x3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \tag{5.2.2}$$

$$\rho_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \quad M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{5.2.3}$$

Precīzāk,

$$\rho_0 \xrightarrow{U} \rho_1 \xrightarrow{Q} \rho_2 \xrightarrow{U} \rho_{final}$$
$$\rho_1 = U\rho U^*; \quad \rho_2 = Q\rho_1 Q; \quad \rho_{final} = U\rho_2 U^*$$
$$T = UQU$$
$$T^* = U^* Q^* U^* \Rightarrow \rho_{final} = T\rho T^*$$

Tātad, algoritms dod 1, ja ieejā bija trijnieks no kopas {000, 010, 101, 111}, tātad SYMMETRY$_3$ funkcijai D(SYMMETRY$_3$) = 2, Q_E(SYMMETRY$_3$)=1.

5.3. 6-mainīgo funkcija ar D(f) = 4 pret $Q_E(f) = 2$

Būla funkcija. Būla funkcija uzdota ar patiesuma tabulu:

X	f(X)	X	f(X)
000000	1	100000	1
000011	1	100011	1
000100	1	100100	1
000111	1	100111	1
001001	1	101001	1
001010	1	101010	1
001101	1	101101	1
001110	1	101110	1
010001	1	110001	1
010010	1	110010	1
010101	1	110101	1
010110	1	110110	1
011000	1	111000	1
011011	1	111011	1
011100	1	111100	1
011111	1	111111	1
Citādi	**0**		

3. tab. Būla funkcija f(X)

Kvantu algoritms.

$$H4 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}; \quad H22 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}; \quad flip12 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_1 = H4H22\,flip12$$

$$U_0 = -U_1^{\,T} = (-1)H4H22\,flip12$$

$$Q_0 = \begin{pmatrix} (-1)^{x1} & 0 & 0 & 0 \\ 0 & (-1)^{x2} & 0 & 0 \\ 0 & 0 & (-1)^{x3} & 0 \\ 0 & 0 & 0 & (-1)^{x4} \end{pmatrix}; \quad Q_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & (-1)^{x5} & 0 & 0 \\ 0 & 0 & (-1)^{x6} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \quad \rho = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

(5.3.1)

$$T = U_1 Q_1 Q_0 U_0$$

$$T^* = U_0^* Q_0^* Q_1^* U_1^*$$

$$\rho_{final} = T\rho T^*$$

(5.3.2)

Rezultātā mums ir kvantu vaicājošais algoritms ar D(f) = 4 un $Q_E(f)=2$. Šī algoritma beigu mērījuma rezultāts ir stāvokļa |00> varbūtība.

6. EFEKTĪVI KVANTU VAICĀJOŠIE ALGORITMI AR MAZU KĻŪDAS VARBŪTĪBU

6.1. Algoritmi ar sarežģītību D(f)=2n pret $Q_{3/4}(f) = \left\lceil \frac{n}{2} \right\rceil$

Šajā nodaļā ir aprakstīti kvantu vaicājošie algoritmi ar ierobežotu kļūdu. Algoritmu kopa ir bezgalīga – n ir jebkurš naturāls skaitlis.

Algoritma vispārinājums.

Definēsim algoritmu līdzīgi nodaļas 5 algoritmiem:

$$\rho_0 \xrightarrow{U_0 Q U_1 [M]} \rho_{final},$$

(6.1.1)

Unitāras transformācijas U_0, U_1 un [M] ir definētas tāpat ka eksaktam algoritmam, Q vispārīga forma T_{2n} funkcijai:

Priekš pāra n $Q_{even} = \begin{pmatrix} x_1 & \dots & x_{\frac{n}{2}} \\ x_{\frac{n}{2}+1} & \dots & x_n \\ y_1 & \dots & y_{\frac{n}{2}} \\ y_{\frac{n}{2}+1} & \dots & y_n \end{pmatrix}$, priekš nepāra n $Q_{odd} = \begin{pmatrix} x_1 & \dots & x_{\left\lceil \frac{n}{2} \right\rceil} \\ x_{\left\lceil \frac{n}{2} \right\rceil} & \dots & x_n \\ y_1 & \dots & y_{\left\lceil \frac{n}{2} \right\rceil} \\ y_{\left\lceil \frac{n}{2} \right\rceil} & \dots & y_n \end{pmatrix}$

(6.1.2)

Dotajam *n* algoritms rēķina funkciju sekojoši:

Atgriež ‘1’ ar varbūtību 1, atgriež ‘0’ ar varbūtību ¾. Funkcijas paplašinājums, pievienojot klāt divus mainīgus, nemaina rezultāta varbūtību.

Teorēma 13. Jebkurai Būla funkcijai T_{2n} no $2n$ mainīgajiem, kur $n \in N$, eksistē kvantu vaicājošs algoritms ar ierobežotu kļūdu, kurš rēķina funkciju ar $\left\lceil \frac{n}{2} \right\rceil$ vaicājumiem, algoritms atgriež ‘1’ ar varbūtību 1, un ‘0’ ar varbūtību ¾.

Pierādījums ir līdzīgs 5.1. nodaļas teorēmas pierādījumam.

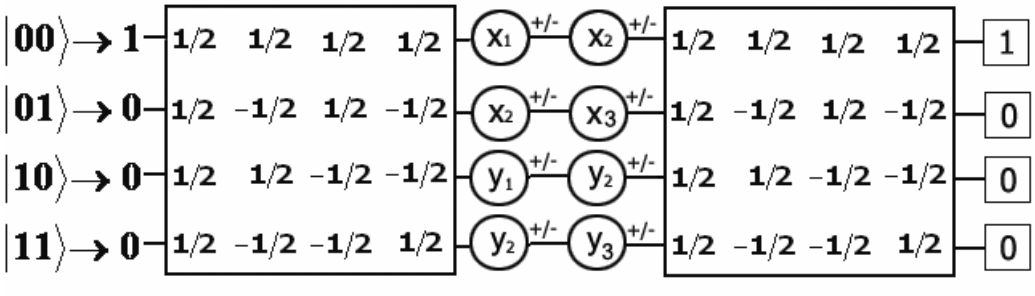
Piemērs. Būla funkcija T_6 (x1,…,x3, y1,…,y3) uzdota ar patiesuma tabulu:

XY	T_6	XY	T_6
000000	1	101010	1
000111	1	101101	1
010010	1	111000	1
010101	1	111111	1
citādi	0		

4. tab. Būla funkcija T_6

Determinēta sarežģītība: $D(T_6) = 6$. Seko no funkcijas jutības uz ieejas virknes 000000 viena bita maiņu.

Algoritms. Attēlā 9 ir kvantu algoritms, kurš rēķina funkciju T_6(XY), uzdodot 2 jautājumus, un varbūtība, ka rezultāts ir pareizs, nav mazāka par 3/4.



9. att. Būla funkcijas T_6 kvantu vaicājošs algoritms.

Vaicājums šīm algoritmam ir $Q = \begin{pmatrix} x1\ x2 \\ x2\ x3 \\ y1\ y2 \\ y2\ y3 \end{pmatrix}$.

(6.1.3)

6.2. Funkcija EQUALITY3 , $Q_{8/9}(EQUALITY_3) = 1, \quad D(EQUALITY_3) = 3$

Būla funkcija. $EQUALITY_3(x_1, x_2, x_3) = 1 \Leftrightarrow [x_1 = x_2 = x_3]$. Tas ir, $EQUALITY_3(X) = 1$, ja ieejas virkne ir 000 vai 111.

Determinēta sarežģītība: $D(EQUALITY_3) = 3$, tas seko no funkcijas jutības uz ieejas rindas 000 vai 111 viena bita izmaiņu.

Teorēma 14. Eksistē kvantu vaicājošais algoritms, kas rēķina Būla funkciju *EQUALITY₃(X)* ar sarežģītību $Q_{8/9}(EQUALITY_3) = 1, \quad D(EQUALITY_3) = 3$.

Pierādījums.

Algoritma apraksts. Kvantu vaicājošs algoritms ar ierobežotu kļūdu Būla funkcijai $EQUALITY_3$ ir sekojoša unitāru transformāciju un vaicājumu virkne:

$$\rho_0 \xrightarrow{U} \rho_1 \xrightarrow{Q} \rho_2 \xrightarrow{U} \rho_3 \xrightarrow{[M]} \rho_{final} \tag{6.2.1}$$

$$Q = \begin{pmatrix} (-1)^{x1} & 0 & 0 & 0 \\ 0 & (-1)^{x2} & 0 & 0 \\ 0 & 0 & (-1)^{x3} & 0 \\ 0 & 0 & 0 & (-1)^{x3} \end{pmatrix}; \quad U = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{pmatrix} \tag{6.2.2}$$

Beigu stāvoklis tiek pakļauts mērījumam – nomēra stāvokli |00>. Ja beigu stāvokļa blīvuma matrica $\rho_{final} = (r_{i,j})$, mērījums [M] dod r_{11} vērtību.

Pārbauda rezultātu katrai ieejas virknei. Ja ieejas virkne ir 000 vai 111, beigu stāvokļa blīvuma matricas elements r_{11} vienāds ar 1, citādi $\frac{1}{9}$, kas nozīmē, ka tādām ieejas virknēm $EQUALITY_3\ (X) = 0$ ar varbūtību $\frac{8}{9}$. Determinēti vajag uzdot jautājumu par katra mainīga vērtību, $D(EQUALITY_3)=3$.

6.3. Funkcija EQUALITY3 , $Q_{9/10}(EQUALITY_3) = 1, \quad D(EQUALITY_3) = 3$

Ir dota Būla funkcija $EQUALITY_3(x_1, x_2, x_3) = 1 \Leftrightarrow [x_1 = x_2 = x_3]$,

$D(EQUALITY_3) = 3$ (definēta nodaļā 6.2.)

Teorēma 15. Eksistē kvantu vaicājošais algoritms, kas rēķina Būla funkciju $EQUALITY_3(X)$ ar sarežģītību $Q_{9/10}(EQUALITY_3) = 1, \quad D(EQUALITY_3) = 3$.

Pierādījums.

Algoritma apraksts. Kvantu vaicājošs algoritms ar ierobežotu kļūdu Būla funkcijai $EQUALITY_3$ ir sekojoša unitāru transformāciju un vaicājumu virkne:

$$\rho_0 \xrightarrow{U_0} \rho_1 \xrightarrow{Q} \rho_2 \xrightarrow{U_1} \rho_3 \xrightarrow{U_2} \rho_4 \xrightarrow{[M]} \rho_{final} \tag{6.3.1}$$

$$U_0 = H4 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}; \quad \rho_0 = |00\rangle\langle 00| = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

(6.3.2)

$$Q = \begin{pmatrix} (-1)^{x1} & 0 & 0 & 0 \\ 0 & (-1)^{x1} & 0 & 0 \\ 0 & 0 & (-1)^{x2} & 0 \\ 0 & 0 & 0 & (-1)^{x3} \end{pmatrix}; \quad U_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

(6.3.3)

$$U_2 = \begin{pmatrix} \frac{1}{\sqrt{5}} & 0 & \frac{2}{\sqrt{5}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ \frac{2}{\sqrt{5}} & 0 & -\frac{1}{\sqrt{5}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

(6.3.4)

Mērījums [M] paziņo stāvokļa |00> varbūtību.

Pārbauda rezultātu katrai ieejas virknei. Ja ieejas virkne ir 000 vai 111, algoritms pasaka, ka stāvokļa |00> varbūtība ir $\frac{9}{10}$, citādi $\frac{1}{10}$, kas nozīmē, ka tādām ieejas virknēm *EQUALITY₃ (X)* = 0 ar varbūtību $\frac{9}{10}$. Determinēti vajag uzdot jautājumu par katra mainīga vērtību, *D(EQUALITY₃)=3*.

6.4. Adamāra matricas un citu unitāru matricu loma kvantu algoritmu konstruēšanā

Kvantu vaicājošo algoritmu konstruēšanā ļoti bieži parādās Adamāra matrica. Tā, piemēram, ļāuj pāriet no stāvokļa $\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ uz $\frac{1}{2^{k/2}}\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$, vienmērīgi sadala amplitūdas. Kādas priekšrocības var būt unitārām matricām, kuras sadala amplitūdas nevienmērīgi? Tādas matricas apraksta transformācijas, kuras palielina dažas amplitūdas un samazina visas citas. Ja tas notiek pirms

mērījuma, tad algoritms var būt ar ierobežotu kļūdu, bet, iespējams, varēs samazināt vaicājumu skaitu, paliekot pie iespējami mazas kļūdas varbūtības.

Seko matricas vispārīgā izskatā.

Unitārā matrica 1.

$$M0 = \begin{pmatrix} \sqrt{n} & \sqrt{1-n} \\ \sqrt{1-n} & -\sqrt{n} \end{pmatrix}, \quad n \in [0,1]$$

$$M1 = M0 \circ M0 = \begin{pmatrix} \sqrt{nk} & \sqrt{(1-n)k} & \sqrt{n(1-k)} & \sqrt{(1-n)(1-k)} \\ \sqrt{(1-n)k} & -\sqrt{nk} & \sqrt{(1-n)(1-k)} & -\sqrt{n(1-k)} \\ \sqrt{n(1-k)} & \sqrt{(1-n)(1-k)} & -\sqrt{nk} & -\sqrt{(1-n)k} \\ \sqrt{(1-n)(1-k)} & -\sqrt{n(1-k)} & -\sqrt{(1-n)k} & \sqrt{nk} \end{pmatrix},$$

$n, k \in [0,1]$

(6.4.1)

Unitāra matrica M1 ir vispārīga forma nodaļas 6.2. algoritma matricai (n = 2/3, k = ½, funkcija EQUALITY_3). Ņemot citus *n* un *k*, var iegūt citu kļūdas varbūtību, tomēr dotajā algoritmā vērtības n = 2/3, k = 1/2 dod labāku rezultātu.

Unitārā matrica 2.

$$M2 = \begin{pmatrix} \sqrt{n} & 0 & \sqrt{1-n} & 0 \\ 0 & \sqrt{k} & 0 & \sqrt{1-k} \\ \sqrt{1-n} & 0 & -\sqrt{n} & 0 \\ 0 & \sqrt{1-k} & 0 & -\sqrt{k} \end{pmatrix}, \quad n, k \in [0,1]$$

(6.4.2)

Unitāra matrica M2 ir vispārīga forma nodaļas 6.3. algoritma matricai (n = 1/5, k = 1/2, funkcija EQUALITY_3). Ņemot citus *n* un *k*, var iegūt citu kļūdas varbūtību, tomēr dotajā algoritmā vērtības n = 1/5, k = 1/2 dod labāku rezultātu.

7. EFEKTĪVU ALGORITMU KONSTRUĒŠANA DAUDZARGUMENTU BŪLA FUNKCIJĀM

Ir dabiski izmantot definētas funkcijas kā blociņus sarežģītākās funkcijas definīcijai. Šīs nodaļas mērķis ir parādīt, kā var izmantot 2.nodaļas Būla funkciju algoritmus sarežģītāku algoritmu konstruēšanai.

7.1. Būla funkcija $EOX_{3n}(X)$

Būla funkcija EOX_{3n} = Equality Of XOR, funkcija no *3n* mainīgajiem,

$$EOX_{3n}(x_1, x_2, x_3, \ldots, x_{3i+1}, x_{3i+2}, x_{3i+3}, \ldots, x_{3n-2}, x_{3n-1}, x_{3n}) \equiv$$
$$\equiv EQUALITY_3(\underset{i=0}{\overset{n-1}{XOR}}(x_{3i+1}), \underset{i=0}{\overset{n-1}{XOR}}(x_{3i+2}), \underset{i=0}{\overset{n-1}{XOR}}(x_{3i+3}))$$

(7.1.1)

Determinēta sarežģītība: $D(EOX_{3n}) = 3n$, tas seko no funkcijas jutības uz ieejas rindu $(000)^n$ vai $(111)^n$ viena bita izmaiņu, $s(EOX_{3n}) = 3n$.

$EOX_{3n}(X)$ ir $EQUALITY_3$ funkcijas vispārinājums, kurš ir līdzīgs 5.1. nodaļas algoritma vispārinājuma paņēmienam.

7.1.1. $EOX_{3n}(X)$ kvantu algoritms: $Q_{8/9}(EOX_{3n}) = n$

Piemērs. Būla funkcija $EOX_6(x_1,\ldots,x_6)$ ir uzdota ar patiesuma tabulu

X	EOX_6	X	EOX_6
000000	1	011011	1
000111	1	011100	1
111000	1	100011	1
111111	1	100100	1
001001	1	101010	1
001110	1	101101	1
010010	1	110001	1
010101	1	110110	1
citādi	0		

5. tab. Būla funkcija EOX_6

Determinēta sarežģītība: $D(EOX_6) = 6$, tas seko no funkcijas jutības uz jebkuras ieejas rindas X = 000000 viena bita maiņu.

Kvantu vaicājošs algoritms.

Pievienosim vēl vienu vaicājumu funkcijas EQUALITY$_3$ algoritmam(6.2. nodaļa, $Q_{8/9}(EQUALITY_3)=1, \quad D(EQUALITY_3)=3$):

$$Q_1=\begin{pmatrix}(-1)^{x1} & 0 & 0 & 0\\ 0 & (-1)^{x2} & 0 & 0\\ 0 & 0 & (-1)^{x3} & 0\\ 0 & 0 & 0 & (-1)^{x3}\end{pmatrix}; \quad Q_2=\begin{pmatrix}(-1)^{x4} & 0 & 0 & 0\\ 0 & (-1)^{x5} & 0 & 0\\ 0 & 0 & (-1)^{x6} & 0\\ 0 & 0 & 0 & (-1)^{x6}\end{pmatrix} \tag{7.1.1.1}$$

Jauns kvantu vaicājošais algoritms ir sekojoša transformāciju virkne

$$\rho_0 \xrightarrow{U} \rho_1 \xrightarrow{Q} \rho_2 \xrightarrow{U} \rho_3 \xrightarrow{[M]} \rho_{final} \tag{7.1.1.2}$$

Vaicājums ir divu vaicājumu virkne $Q=Q_1Q_2=\begin{pmatrix}x1\,x4\\ x2\,x5\\ x3\,x6\\ x3\,x6\end{pmatrix}$, unitāra transformācija U paliek

nemainīga.

Algoritma kvantu sarežģītība ir $Q_{8/9}(EOX_6)=2$.

Teorēma 16. Funkcijai *EOX$_{3n}$(X)* eksistē kvantu vaicājošais algoritms ar sarežģītību $Q_{8/9}(EOX_{3n})=n$.

Pierādījums. Algoritma kvantu sarežģītība ir vienāda ar 3-mainīgo „blociņu" skaitu, tas ir *n*.

Vaicājums ir *n* vaicājumu virkne $Q=\begin{pmatrix}x_1 & \ldots & x_{3i+1} & \ldots & x_{3(n-1)+1}\\ x_2 & \ldots & x_{3i+2} & \ldots & x_{3(n-1)+2}\\ x_3 & \ldots & x_{3i+3} & \ldots & x_{3(n-1)+3}\\ x_3 & \ldots & x_{3i+3} & \ldots & x_{3(n-1)+3}\end{pmatrix}, \quad 0\le i\le n-1$

(7.1.1.3)

Sareizināsim visas vaicājumu matricas

$$Q=\begin{pmatrix}(-1)^{z1} & 0 & 0 & 0\\ 0 & (-1)^{z2} & 0 & 0\\ 0 & 0 & (-1)^{z3} & 0\\ 0 & 0 & 0 & (-1)^{z3}\end{pmatrix};\quad \begin{cases} z1=\sum_{i=0}^{n-1} x_{3i+1} \\ z2=\sum_{i=0}^{n-1} x_{3i+2} \\ z3=\sum_{i=0}^{n-1} x_{3i+3}\end{cases}$$

$$EOX_{3n}(X)=EQUALITY_3(z1,z2,z3)=1\Leftrightarrow[z1=z2=z3]$$

(7.1.1.4)

Ja z1, z2, z3 ir (-1) pakāpes, tad būtiski ir saskaitīt pēc moduļa 2: $\sum_{i=0}^{n-1} x_{3i+1} \equiv \underset{i=0}{\overset{n-1}{XOR}}(x_{3i+1})$.

Analoģiski $\sum_{i=0}^{n-1} x_{3i+2} \equiv \underset{i=0}{\overset{n-1}{XOR}}(x_{3i+2})$, $\sum_{i=0}^{n-1} x_{3i+3} \equiv \underset{i=0}{\overset{n-1}{XOR}}(x_{3i+3})$.

Ja X=$(000)^n$, tad viena bita maiņa mainīs kādu no z1,z2,z3, attiecīgi, mainīs *EQUALITY$_3$(z1,z2,z3)* vērtību. Tātad, jauns algoritms saglabā EQUALITY$_3$ algoritma precizitāti, atgriež „1" ar varbūtību 1, un „0" ar varbūtību 8/9, uzdod *n* kvantu jautājumus.

7.1.2. EOX$_{3n}$(X) kvantu algoritms: $Q_{9/10}(EOX_{3n})=n$

Šajā nodaļā tiek piedāvāts cits kvantu vaicājošs algoritms Būla funkcijai *EOX$_{3n}$*.

Kvantu vaicājošs algoritms.

Jauns kvantu vaicājošais algoritms ir sekojoša transformāciju virkne

$$\rho_0 \xrightarrow{U_0} \rho_1 \xrightarrow{Q} \rho_2 \xrightarrow{U_1} \rho_3 \xrightarrow{U_2} \rho_4 \xrightarrow{[M]} \rho_{final}$$

(7.1.2.1)

$$H4=\frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{pmatrix};\quad \rho_0=|00\rangle\langle 00|=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{pmatrix}$$

$$U_0=H4$$

(7.1.2.2)

$$U_2 = \begin{pmatrix} \frac{1}{\sqrt{5}} & 0 & \frac{2}{\sqrt{5}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ \frac{2}{\sqrt{5}} & 0 & -\frac{1}{\sqrt{5}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix}; \quad U_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

(7.1.2.3)

Q ir vaicājumu virkne $Q = \begin{pmatrix} x_1 & \dots & x_{3i+1} & \dots & x_{3(n-1)+1} \\ x_2 & \dots & x_{3i+2} & \dots & x_{3(n-1)+2} \\ x_3 & \dots & x_{3i+3} & \dots & x_{3(n-1)+3} \\ x_3 & \dots & x_{3i+3} & \dots & x_{3(n-1)+3} \end{pmatrix}, \quad 0 \le i \le n-1$

(7.1.2.4)

Teorēma 17. Funkcijai $EOX_{3n}(X)$ eksistē kvantu vaicājošais algoritms ar sarežģītību $Q_{9/10}(EOX_{3n}) = n$.

Pierādījums. Spriedumi no Teorēmas 16. pierādījuma noveda pie secinājuma, ka jauns algoritms saglabā $EQUALITY_3$ algoritma precizitāti. Šim algoritmam mēs izmantojām funkcijas $EQUALITY_3$ algoritmu ar sarežģītību $Q_{9/10}(EQUALITY_3) = 1$. Tātad, algoritms paziņo funkcijas $EOX_{3n}(X)$ vērtību ar varbūtību 9/10, kamēr uzdod *n* kvantu jautājumus.

7.2. Funkciju T_{2n} iterācija. Būla funkcija $TAND_{4n}$, $Q_{3/4}(TAND_{4n}) = n$ pret $D(TAND_{4n}) = 4n$

Atcerēsimies Būla funkciju $T_{2n}(XY)$, kuras eksakts kvantu vaicājošs algoritms ir prezentēts 5.1. nodaļā. Ņemsim to par eksakta kvantu algoritma piemēru un pārstāvi, un parādīsim, ka var definēt sarežģītāku algoritmu uz vienkāršākā eksakta algoritma pamata.

Algoritmus savieno paralēli, šīs piegājiens ļauj divreiz palielināt mainīgo skaitu, nepalielina vaicājumu skaitu, bet diemžēl padara jaunu salikto algoritmu par algoritmu ar ierobežotu kļūdu: algoritma rezultāts ir 1 precīzi, un 0 ar varbūtību ¾ sliktākajā gadījumā . Tieši tāpēc šo paņēmienu ir vērts pielietot tikai eksaktiem algoritmiem, citādi kļūdas varbūtība ir vēl augstākā, kaut gan tas ir atkarīgs no dota algoritma īpašībām.

Šī rīcība ir līdzīga funkcijas iterācijai, kad funkcijas arguments ir tās pašas funkcijas vērtība no cita argumenta, F(F(X),F(Y)).

Kvantu algoritms. Rēķināšanas process ir līdzīgs funkcijas T_{2n} eksaktam algoritmam:

unitāru transformāciju un vaicājumu virkne ir gandrīz tāda paša

$$\rho_0 \xrightarrow{U} \rho_1 \xrightarrow{Q} \rho_2 \xrightarrow{U} \rho_3 \xrightarrow{[M]} \rho_{final} \tag{7.2.1}$$

Mēs izmainījām unitāru transformāciju U un vaicājumu matricu Q sekojoši:

$$U = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \end{pmatrix}; \quad \rho_0 = |000\rangle\langle 000| = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{7.2.2}$$

Mērījums [M] paziņo stāvokļa |000> varbūtību.

$Q = \begin{pmatrix} Q_1 & 0 \\ 0 & Q_2 \end{pmatrix}$, Q_1 un Q_2 ir neatkarīgas vaicājumu virknes no $T_{2n}(XY)$ algoritma.

Vaicājumu var īsi aprakstīt ar matricu, kurai katra i-tā kolona ir Q_i vaicājums parastā nozīmē, katra rinda ir attiecīga stāvokļa |i> ($i \in [0;7]$ binārā pierakstā) amplitūda.

Ja n ir pāra skaitlis, tad $Q_{even} = \begin{pmatrix} x_1 & \dots & x_{\frac{n}{2}} & y_1 & \dots & y_{\frac{n}{2}} \\ x_{\frac{n}{2}+1} & \dots & x_n & y_{\frac{n}{2}+1} & \dots & y_n \\ y_1 & \dots & y_{\frac{n}{2}} & x_1 & \dots & x_{\frac{n}{2}} \\ y_{\frac{n}{2}+1} & \dots & y_n & x_{\frac{n}{2}+1} & \dots & x_n \\ z_1 & \dots & z_{\frac{n}{2}} & t_1 & \dots & t_{\frac{n}{2}} \\ z_{\frac{n}{2}+1} & \dots & z_n & t_{\frac{n}{2}+1} & \dots & t_n \\ t_1 & \dots & t_{\frac{n}{2}} & z_1 & \dots & z_{\frac{n}{2}} \\ t_{\frac{n}{2}+1} & \dots & t_n & z_{\frac{n}{2}+1} & \dots & z_n \end{pmatrix}$, (7.2.3)

Ja n ir nepāra, tad $Q_{odd} = \begin{pmatrix} x_1 & \dots & x_{\lfloor \frac{n}{2} \rfloor+1} & y_1 & \dots & y_{\lfloor \frac{n}{2} \rfloor} \\ x_{\lfloor \frac{n}{2} \rfloor+1} & \dots & x_n & y_{\lfloor \frac{n}{2} \rfloor+2} & \dots & y_n \\ y_1 & \dots & y_{\lfloor \frac{n}{2} \rfloor+1} & x_1 & \dots & x_{\lfloor \frac{n}{2} \rfloor} \\ y_{\lfloor \frac{n}{2} \rfloor+1} & \dots & y_n & x_{\lfloor \frac{n}{2} \rfloor+2} & \dots & x_n \\ z_1 & \dots & z_{\lfloor \frac{n}{2} \rfloor+1} & t_1 & \dots & t_{\lfloor \frac{n}{2} \rfloor} \\ z_{\lfloor \frac{n}{2} \rfloor+1} & \dots & z_n & t_{\lfloor \frac{n}{2} \rfloor+2} & \dots & t_n \\ t_1 & \dots & t_{\lfloor \frac{n}{2} \rfloor+1} & z_1 & \dots & z_{\lfloor \frac{n}{2} \rfloor} \\ t_{\lfloor \frac{n}{2} \rfloor+1} & \dots & t_n & z_{\lfloor \frac{n}{2} \rfloor+2} & \dots & z_n \end{pmatrix}$

(7.2.4)

Jauna Būla funkcija ir *4n*-mainīgo funkcija $TAND_{4n} = AND(T_{2n}, T_{2n})$, $TAND_{4n} = 1$ tad un tikai tad, ja abi argumenti $T_{2n} = 1$, citādi $TAND_{4n} = 0$ ar varbūtību ¾ sliktākajā gadījumā. $D(TAND_{4n}) = 4n$, jo $s(TAND_{4n}) = 4n$, pret kvantu sarežģītību $Q_{3/4}(TAND_{4n}) = n$.

8. DIVU KUBITU KVANTU SISTĒMAS EKSAKTA ALGORITMA SAREŽĢĪTĪBA

Šajā nodaļā ir mēģinājums saprast, vai pāreja uz jaukto stāvokli algoritma sākumā vai vidū var samazināt tā kvantu sarežģītību.

Apskatīsim divu kubitu kvantu sistēmu. Sākumstāvoklis ir jauktais stāvoklis, uzdots ar

blīvuma matricu $\rho = \begin{pmatrix} r1 & 0 & 0 & 0 \\ 0 & r2 & 0 & 0 \\ 0 & 0 & r3 & 0 \\ 0 & 0 & 0 & r4 \end{pmatrix}$, kur *r1, r2, r3, r4 > 0*

(8.1)

Transformāciju virkne T ir unitāru transformāciju un vaicājuma virkne $T = ZQU$,

$$U = \begin{pmatrix} u11 & u12 & u13 & u14 \\ u21 & u22 & u23 & u24 \\ u31 & u32 & u33 & u34 \\ u41 & u42 & u43 & u44 \end{pmatrix}; \quad Z = \begin{pmatrix} z11 & z12 & z13 & z14 \\ z21 & z22 & z23 & z24 \\ z31 & z32 & z33 & z34 \\ z41 & z42 & z43 & z44 \end{pmatrix};$$

$$Q = \begin{pmatrix} (-1)^{x1} & 0 & 0 & 0 \\ 0 & (-1)^{x2} & 0 & 0 \\ 0 & 0 & (-1)^{x3} & 0 \\ 0 & 0 & 0 & (-1)^{x4} \end{pmatrix}$$

$$\rho_{final} = T\rho T^{*} = ZQU\rho U^{*}QZ^{*}$$

(8.2)

Mērījums [M] nomērīs |00> stāvokli.

Ar zvaigznīti (*) apzīmē transponētu kompleksi saistītu matricu, ar svītru virs skaitļa ir domāts kompleksi saistīts skaitlis.

Vispārīgā formā mērījuma vērtība ir funkcija *P(X):*

$$\begin{aligned} &P(x1, x2, x3, x4) = \\ &r1((-1)^{x1}u_{11}z_{11} + (-1)^{x2}u_{12}z_{21} + (-1)^{x3}u_{13}z_{31} + (-1)^{x4}u_{14}b_{41}) \cdot \\ &\quad \cdot ((-1)^{x1}\overline{u_{11}z_{11}} + (-1)^{x2}\overline{u_{12}z_{21}} + (-1)^{x3}\overline{u_{13}z_{31}} + (-1)^{x4}\overline{u_{14}b_{41}}) + \\ &r2((-1)^{x1}u_{21}z_{11} + (-1)^{x2}u_{22}z_{21} + (-1)^{x3}u_{23}z_{31} + (-1)^{x4}u_{24}b_{41}) \cdot \\ &\quad \cdot ((-1)^{x1}\overline{u_{21}z_{11}} + (-1)^{x2}\overline{u_{22}z_{21}} + (-1)^{x3}\overline{u_{23}z_{31}} + (-1)^{x4}\overline{u_{24}b_{41}}) + \\ &r3((-1)^{x1}u_{31}z_{11} + (-1)^{x2}u_{32}z_{21} + (-1)^{x3}u_{33}z_{31} + (-1)^{x4}u_{34}b_{41}) \cdot \\ &\quad \cdot ((-1)^{x1}\overline{u_{31}z_{11}} + (-1)^{x2}\overline{u_{32}z_{21}} + (-1)^{x3}\overline{u_{33}z_{31}} + (-1)^{x4}\overline{u_{34}b_{41}}) + \\ &r4((-1)^{x1}u_{41}z_{11} + (-1)^{x2}u_{42}z_{21} + (-1)^{x3}u_{43}z_{31} + (-1)^{x4}u_{44}b_{41}) \cdot \\ &\quad \cdot ((-1)^{x1}\overline{u_{41}z_{11}} + (-1)^{x2}\overline{u_{42}z_{21}} + (-1)^{x3}\overline{u_{43}z_{31}} + (-1)^{x4}\overline{u_{44}b_{41}}) \end{aligned}$$

(8.3)

Pieņemsim, ka mēs būvējam kvantu eksaktu algoritmu ar sarežģītību $Q_E(f) = 1$ pret $D(f) = 4$.

No pieņēmuma seko vienādojumu sistēma S1:

$$S1 = \begin{cases} P(0,0,0,0) = 1 \\ P(1,0,0,0) = 0 \\ P(0,1,0,0) = 0 \\ P(0,0,1,0) = 0 \\ P(0,0,0,1) = 0 \end{cases} \tag{8.4}$$

Vienādojumu sistēma S2 ir S1 vienādojumu apakškopa:

$$S2 = \begin{cases} P(1,0,0,0) = 0 \\ P(0,1,0,0) = 0 \\ P(0,0,1,0) = 0 \\ P(0,0,0,1) = 0 \end{cases} \tag{8.5}$$

Īsuma dēļ apzīmēsim

$$\begin{aligned} &t_1 = u_{11} z_{11}; \quad t_2 = u_{12} z_{21}; \quad t_3 = u_{13} z_{31}; \quad t_4 = u_{14} z_{41}; \\ &t_5 = u_{21} z_{11}; \quad t_6 = u_{22} z_{21}; \quad t_7 = u_{23} z_{31}; \quad t_8 = u_{24} z_{41}; \\ &t_9 = u_{31} z_{11}; \quad t_{10} = u_{32} z_{21}; \quad t_{11} = u_{33} z_{31}; \quad t_{12} = u_{34} z_{41}; \\ &t_{13} = u_{41} z_{11}; \quad t_{14} = u_{42} z_{21}; \quad t_{15} = u_{43} z_{31}; \quad t_{16} = u_{44} z_{41}; \end{aligned} \tag{8.6}$$

Tagad pārrakstīsim *S2*:

$$S2 = \begin{cases} r1(-t_1 + t_2 + t_3 + t_4)(-\overline{t_1} + \overline{t_2} + \overline{t_3} + \overline{t_4}) + r2(-t_5 + t_6 + t_7 + t_8)(-\overline{t_5} + \overline{t_6} + \overline{t_7} + \overline{t_8}) + \\ \quad + r3(-t_9 + t_{10} + t_{11} + t_{12})(-\overline{t_9} + \overline{t_{10}} + \overline{t_{11}} + \overline{t_{12}}) + \\ \quad + r4(-t_{13} + t_{14} + t_{15} + t_{16})(-\overline{t_{13}} + \overline{t_{14}} + \overline{t_{15}} + \overline{t_{16}}) = 0 \\ r1(t_1 - t_2 + t_3 + t_4)(\overline{t_1} - \overline{t_2} + \overline{t_3} + \overline{t_4}) + r2(t_5 - t_6 + t_7 + t_8)(\overline{t_5} - \overline{t_6} + \overline{t_7} + \overline{t_8}) + \\ \quad + r3(t_9 - t_{10} + t_{11} + t_{12})(\overline{t_9} - \overline{t_{10}} + \overline{t_{11}} + \overline{t_{12}}) + \\ \quad + r4(t_{13} - t_{14} + t_{15} + t_{16})(\overline{t_{13}} - \overline{t_{14}} + \overline{t_{15}} + \overline{t_{16}}) = 0 \\ r1(t_1 + t_2 - t_3 + t_4)(\overline{t_1} + \overline{t_2} - \overline{t_3} + \overline{t_4}) + r2(t_5 + t_6 - t_7 + t_8)(\overline{t_5} + \overline{t_6} - \overline{t_7} + \overline{t_8}) + \\ \quad + r3(t_9 + t_{10} - t_{11} + t_{12})(\overline{t_9} + \overline{t_{10}} - \overline{t_{11}} + \overline{t_{12}}) + \\ \quad + r4(t_{13} + t_{14} - t_{15} + t_{16})(\overline{t_{13}} + \overline{t_{14}} - \overline{t_{15}} + \overline{t_{16}}) = 0 \\ r1(t_1 + t_2 + t_3 - t_4)(\overline{t_1} + \overline{t_2} + \overline{t_3} - \overline{t_4}) + r2(t_5 + t_6 + t_7 - t_8)(\overline{t_5} + \overline{t_6} + \overline{t_7} - \overline{t_8}) + \\ \quad + r3(t_9 + t_{10} + t_{11} - t_{12})(\overline{t_9} + \overline{t_{10}} + \overline{t_{11}} - \overline{t_{12}}) + \\ \quad + r4(t_{13} + t_{14} + t_{15} - t_{16})(\overline{t_{13}} + \overline{t_{14}} + \overline{t_{15}} - \overline{t_{16}}) = 0 \end{cases} \tag{8.7}$$

Turpināsim pārveidojumu

$$S2=\begin{cases} r1(-t_1+t_2+t_3+t_4)\overline{(-t_1+t_2+t_3+t_4)}+r2(-t_5+t_6+t_7+t_8)\overline{(-t_5+t_6+t_7+t_8)}+ \\ \quad +r3(-t_9+t_{10}+t_{11}+t_{12})\overline{(-t_9+t_{10}+t_{11}+t_{12})}+ \\ \quad +r4(-t_{13}+t_{14}+t_{15}+t_{16})\overline{(-t_{13}+t_{14}+t_{15}+t_{16})}=0 \\ r1(t_1-t_2+t_3+t_4)\overline{(t_1-t_2+t_3+t_4)}+r2(t_5-t_6+t_7+t_8)\overline{(t_5-t_6+t_7+t_8)}+ \\ \quad +r3(t_9-t_{10}+t_{11}+t_{12})\overline{(t_9-t_{10}+t_{11}+t_{12})}+ \\ \quad +r4(t_{13}-t_{14}+t_{15}+t_{16})\overline{(t_{13}-t_{14}+t_{15}+t_{16})}=0 \\ r1(t_1+t_2-t_3+t_4)\overline{(t_1+t_2-t_3+t_4)}+r2(t_5+t_6-t_7+t_8)\overline{(t_5+t_6-t_7+t_8)}+ \\ \quad +r3(t_9+t_{10}-t_{11}+t_{12})\overline{(t_9+t_{10}-t_{11}+t_{12})}+ \\ \quad +r4(t_{13}+t_{14}-t_{15}+t_{16})\overline{(t_{13}+t_{14}-t_{15}+t_{16})}=0 \\ r1(t_1+t_2+t_3-t_4)\overline{(t_1+t_2+t_3-t_4)}+r2(t_5+t_6+t_7-t_8)\overline{(t_5+t_6+t_7-t_8)}+ \\ \quad +r3(t_9+t_{10}+t_{11}-t_{12})\overline{(t_9+t_{10}+t_{11}-t_{12})}+ \\ \quad +r4(t_{13}+t_{14}+t_{15}-t_{16})\overline{(t_{13}+t_{14}+t_{15}-t_{16})}=0 \end{cases} \tag{8.8}$$

1. Kompleksā skaitļā reizinājums ar kompleksi saistīto ir pozitīvs reāls skaitlis,
   $(a+bi)\overline{(a+bi)}=(a+bi)(a-bi)=a^2+b^2, \quad a,b\in R$
2. Kvantu stāvokļa blīvuma matricas diagonāles elementi ir reāli skaitļi, kuru summa $r1 + r2 + r3 + r4 = 1$.

Ņemot vērā minētos divus punktus, katrs vienādojums ir vienādība formā $a + b + c + d = 0$, kur a,b,c,d ir pozitīvi reāli skaitļi, ir likumīgs tāds katra vienādojuma pārveidojums:

$$a+b+c+d=0 \Leftrightarrow \begin{cases} a=0 \\ b=0 \\ c=0 \\ d=0 \end{cases}.$$

Pārrakstīsim *S2* vienādojumus un pārkārtosim to secību, sanāk:

$$S2=\begin{cases} r1(-t_1+t_2+t_3+t_4)\overline{(-t_1+t_2+t_3+t_4)}=0 \\ r1(t_1-t_2+t_3+t_4)\overline{(t_1-t_2+t_3+t_4)}=0 \\ r1(t_1+t_2-t_3+t_4)\overline{(t_1+t_2-t_3+t_4)}=0 \\ r1(t_1+t_2+t_3-t_4)\overline{(t_1+t_2+t_3-t_4)}=0 \\ r2(-t_5+t_6+t_7+t_8)\overline{(-t_5+t_6+t_7+t_8)}=0 \\ r2(t_5-t_6+t_7+t_8)\overline{(t_5-t_6+t_7+t_8)}=0 \\ r2(t_5+t_6-t_7+t_8)\overline{(t_5+t_6-t_7+t_8)}=0 \\ r2(t_5+t_6+t_7-t_8)\overline{(t_5+t_6+t_7-t_8)}=0 \\ r3(-t_9+t_{10}+t_{11}+t_{12})\overline{(-t_9+t_{10}+t_{11}+t_{12})}=0 \\ r3(t_9-t_{10}+t_{11}+t_{12})\overline{(t_9-t_{10}+t_{11}+t_{12})}=0 \\ r3(t_9+t_{10}-t_{11}+t_{12})\overline{(t_9+t_{10}-t_{11}+t_{12})}=0 \\ r3(t_9+t_{10}+t_{11}-t_{12})\overline{(t_9+t_{10}+t_{11}-t_{12})}=0 \\ r4(-t_{13}+t_{14}+t_{15}+t_{16})\overline{(-t_{13}+t_{14}+t_{15}+t_{16})}=0 \\ r4(t_{13}-t_{14}+t_{15}+t_{16})\overline{(t_{13}-t_{14}+t_{15}+t_{16})}=0 \\ r4(t_{13}+t_{14}-t_{15}+t_{16})\overline{(t_{13}+t_{14}-t_{15}+t_{16})}=0 \\ r4(t_{13}+t_{14}+t_{15}-t_{16})\overline{(t_{13}+t_{14}+t_{15}-t_{16})}=0 \end{cases} \Leftrightarrow \tag{8.9}$$

$$\Leftrightarrow \left[\begin{array}{l} \begin{cases} r1(-t_1+t_2+t_3+t_4)\overline{(-t_1+t_2+t_3+t_4)}=0 \\ r1(t_1-t_2+t_3+t_4)\overline{(t_1-t_2+t_3+t_4)}=0 \\ r1(t_1+t_2-t_3+t_4)\overline{(t_1+t_2-t_3+t_4)}=0 \\ r1(t_1+t_2+t_3-t_4)\overline{(t_1+t_2+t_3-t_4)}=0 \end{cases} \\ \begin{cases} r2(-t_5+t_6+t_7+t_8)\overline{(-t_5+t_6+t_7+t_8)}=0 \\ r2(t_5-t_6+t_7+t_8)\overline{(t_5-t_6+t_7+t_8)}=0 \\ r2(t_5+t_6-t_7+t_8)\overline{(t_5+t_6-t_7+t_8)}=0 \\ r2(t_5+t_6+t_7-t_8)\overline{(t_5+t_6+t_7-t_8)}=0 \end{cases} \\ \begin{cases} r3(-t_9+t_{10}+t_{11}+t_{12})\overline{(-t_9+t_{10}+t_{11}+t_{12})}=0 \\ r3(t_9-t_{10}+t_{11}+t_{12})\overline{(t_9-t_{10}+t_{11}+t_{12})}=0 \\ r3(t_9+t_{10}-t_{11}+t_{12})\overline{(t_9+t_{10}-t_{11}+t_{12})}=0 \\ r3(t_9+t_{10}+t_{11}-t_{12})\overline{(t_9+t_{10}+t_{11}-t_{12})}=0 \end{cases} \\ \begin{cases} r4(-t_{13}+t_{14}+t_{15}+t_{16})\overline{(-t_{13}+t_{14}+t_{15}+t_{16})}=0 \\ r4(t_{13}-t_{14}+t_{15}+t_{16})\overline{(t_{13}-t_{14}+t_{15}+t_{16})}=0 \\ r4(t_{13}+t_{14}-t_{15}+t_{16})\overline{(t_{13}+t_{14}-t_{15}+t_{16})}=0 \\ r4(t_{13}+t_{14}+t_{15}-t_{16})\overline{(t_{13}+t_{14}+t_{15}-t_{16})}=0 \end{cases} \end{array}\right. \tag{8.10}$$

r1, r2, r3, r4 ir nenulles skaitļi, kas seko no sākumstāvokļa definīcijas, tāpēc mēs varam vienkārši noīsināt tos katrā vienādojumā. Vēl viena komplekso skaitļu īpašība ir ļoti noderīga sistēmas pārveidojumam: $z\bar{z} = (a+bi)\overline{(a+bi)} = a^2 + b^2;$
$z\bar{z} = 0 \Leftrightarrow z = 0$

$$\begin{cases} (-t_1 + t_2 + t_3 + t_4) = 0 \\ (t_1 - t_2 + t_3 + t_4) = 0 \\ (t_1 + t_2 - t_3 + t_4) = 0 \\ (t_1 + t_2 + t_3 - t_4) = 0 \end{cases}; \quad \begin{cases} (-t_5 + t_6 + t_7 + t_8) = 0 \\ (t_5 - t_6 + t_7 + t_8) = 0 \\ (t_5 + t_6 - t_7 + t_8) = 0 \\ (t_5 + t_6 + t_7 - t_8) = 0 \end{cases};$$

$$\begin{cases} (-t_9 + t_{10} + t_{11} + t_{12}) = 0 \\ (t_9 - t_{10} + t_{11} + t_{12}) = 0 \\ (t_9 + t_{10} - t_{11} + t_{12}) = 0 \\ (t_9 + t_{10} + t_{11} - t_{12}) = 0 \end{cases}; \quad \begin{cases} (-t_{13} + t_{14} + t_{15} + t_{16}) = 0 \\ (t_{13} - t_{14} + t_{15} + t_{16}) = 0 \\ (t_{13} + t_{14} - t_{15} + t_{16}) = 0 \\ (t_{13} + t_{14} + t_{15} - t_{16}) = 0 \end{cases}$$

(8.11)

Četru iegūtu lineāru vienādojumu sistēmu determinanti ir vienādi, sistēmas ir ekvivalentas, nomainīti tikai mainīgo apzīmējumi. Apskatīsim pirmās LVS atrisinājumu:

$$S2_1 \equiv \begin{pmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}; \quad D \equiv \begin{vmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{vmatrix} = -16$$

$$t_1 D = 16t_1 = \begin{vmatrix} 0 & 1 & 1 & 1 \\ 0 & -1 & 1 & 1 \\ 0 & 1 & -1 & 1 \\ 0 & 1 & 1 & -1 \end{vmatrix} = 0 \Rightarrow t_1 = 0$$

(8.12)

Līdzīgi spriežot, iegūsim (t_1 ,t_2 , t_3, t_4) = (0,0,0,0). Tāpēc arī visi $t_i = 0$, i ir skaitlis no 1 līdz 16. Tad atcerēsimies, ka no vienādojumu sistēmas *S1* mēs izmetām pirmo vienādojumu, īsi atkārtosim pārveidojumu virkni tikai šīm vienam vienādojumam:

$$P(0,0,0,0) = 1 \Rightarrow$$
$$r1(t_1 + t_2 + t_3 + t_4)\overline{(t_1 + t_2 + t_3 + t_4)} + r2(t_5 + t_6 + t_7 + t_8)\overline{(t_5 + t_6 + t_7 + t_8)} +$$
$$+ r3(t_9 + t_{10} + t_{11} + t_{12})\overline{(t_9 + t_{10} + t_{11} + t_{12})} + r4(t_{13} + t_{14} + t_{15} + t_{16})\overline{(t_{13} + t_{14} + t_{15} + t_{16})} = 1 \Rightarrow$$
$$r1 \cdot 0 + r2 \cdot 0 + r3 \cdot 0 + r4 \cdot 0 = 1 \Rightarrow$$
$$0 = 1$$

(8.13)

Mēs nonācām pie pretrunas, tātad vienādojumu sistēmas *S1* atrisinājumu kopa ir tukša kopa. Gadījumā, ja mēs aizturēsim vienu mainīgo un mēģināsim uzbūvēt kvantu algoritmu ar vaicājumu tikai par 3 mainīgu vērtībām, spriedumi būs līdzīgi, un beigās arī tiksim pie pretrunas.

Tīrais stāvoklis parādās, ja atļauj kādām no r1, r2, r3, r4 vērtībām būt par nullēm, piemēram,

ja sākumstāvoklis ir $\rho = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$.

(8.14)

Teorēma 19. 2-kubitu kvantu sistēmā nevar uzbūvēt eksaktu kvantu algoritmu ar sarežģītību $Q_E(f) < D(f)/2$.

Pierādījums. Augstāk ir piedāvāta spriedumu virkne, kas pierāda teorēmu.

NOBEIGUMS

Darbā ir prezentēti oriģināli efektīvi kvantu vaicājošie algoritmi dažām vispārīgā formā definētām Būla funkciju kopām. Visu piedāvāto algoritmu kvantu sarežģītība ir mazāka par attiecīgās Būla funkcijas determinētu sarežģītību: eksaktiem algoritmiem ir $Q_E(f) = D(f)/2$, algoritmiem ar ierobežotu kļūdu kvantu sarežģītība ir mazāka pie mazas kļūdas varbūtības. Pētījuma laikā neizdevās izstrādāt paņēmienu, kurš ļautu konstruēt efektīvu algoritmu patvaļīgai Būla funkcijai.

Darbā ir mēģinājums noskaidrot, kādas priekšrocības un trūkumus dod kvantu jauktā stāvokļa jēdziens. Darbā ir mēģinājums pierādīt, ka jaukto stāvokļu sistēmā nevar uzbūvēt eksaktu algoritmu ar kvantu sarežģītību mazāku par D(f)/2, tas ir uzlabot funkcijas PARITY rezultātu. Tas ir tikai liela pētījuma sākumpunkts, darbā ir aplūkots privātgadījums 2-kubitu sistēmai. Nākamais darba virziens ir vispārināt un papildināt esošā pētījuma rezultātus jautājumā par kvantu jauktajiem stāvokļiem.

LITERATŪRAS SARAKSTS

[1] **Jozef Gruska**. *Quantum computing*. McGraw Hill, 1999.

[2] **M.Nielsen, I.Chuang**. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[3] **Gatis Midrijānis**. *Exact quantum query complexity for total Boolean functions*. 27.marts 2004.gads. Pieejams internetā http://arxiv.org/abs/quant-ph/0403168

[4] **H.U.Simon**. *A tight Ω(log log n)-bound on the time for parallel RAM's to compute non-degenerate Boolean functions*. ***In:*** Symposium on Foundations of Computation Theory, volume 158 of Lecture Notes in Computer Science, pp 439-444. Springer, 1983.

[5] **Harry Buhrman, Ronald de Wolf**. *Complexity Measures and Decision Tree Complexity: A Survey*. University of Amsterdam.
Pieejams internetā http://www.cwi.nl/rdewolf/publ/qc/dectree.ps.gz

[6] **N.Nisan**. *PRAM s and decision trees*, SIAM J.Comput. 20(6):999-1007,1991.

[7] **Umesh Vazirani** lekcijas internetā
http://www.math.uwaterloo.ca/~anayak/co781-s04/lectures/

[8] **D.Aharonov, A.Kitaev, N.Nisan**: *Quantum Circuits with Mixed States*.
Pieejams internetā http://arxiv.org/abs/quant-ph/9806029

[9] Adamāra matricas no MathWorld.
Pieejams internetā http://mathworld.wolfram.com/HadamardMatrix.html

[10] **A.Ambainis**. *Quantum query algorithms and lower bounds(survey article)*. Proceedings of FOTFS III.

[11] **R. De Wolf.** *Quantum Computing and Communication Complexity*. University of Amsterdam, 2001.

[12] **R.Beals, H.buhrman, R.Cleve, M.Mosca, R. de Wolf.** *Quantum lower bounds by polynomials*. In Proceeding of 39th FOCS.
Pieejams internetā http://arxiv.org/abs/quant-ph/9802049

[13] **H.Buhrman, R.Cleve, R. De Wolf, and Ch. Zalka.** *Bounds for Small-Error and Zero-Error Quantum Algorithms*. ***In:*** 40th IEEE Symposium on Foundations of Computer Science (FOCS'99), lpp. 358-368, 1999.
Pieejams internetā http://arxiv.org/abs/cs/9904019

[14] **A.Ambainis**. *Polynomial degree vs. quantum query complexity*. 6.maijs 2003.
Pieejams internetā http://arxiv.org/quant-ph/0305028

[15] **Rūsiņš Freivalds**. *Query Algorithms*.

Maģistra darbs „Kvantu vaicājošie algoritmi Būla funkciju rēķināšanai”

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Taisija Miščenko-Slatenkova _____
(Autora paraksts)

Ar savu parakstu apliecinu, ka esmu lasījis augšminēto maģistra darbu un atzīstu to par p i e m ē r o t u / n e p i e m ē r o t u (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes datorzinātņu maģistrantūrā. .

Darba vadītājs: profesors R. Freivalds _____
(Vadītāja paraksts)

Darbs iesniegts Datorikas nodaļā _____.
(Iesniegšanas datums)

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā sistēmā.

Metodiķe: _____ _____.
(Metodiķe) (Metodiķes paraksts)

Recenzents: Lelde Lāce _____
(Recenzenta paraksts)

Darbs aizstāvēts maģistra gala pārbaudījuma komisijas sēdē

_____ prot. Nr. _____, vērtējums _____

(Darba aizstāvēšanas datums)

Komisijas sekretārs: _____
(Sekretāra paraksts)