LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

KVANTU ALGORITMU KONSTRUĒŠANA, IZMANTOJOT ČAULU PROGRAMMAS

BAKALAURA DARBS

Autors: **Nikita Larka**
Studenta apliecības Nr.: nl13005
Darba vadītājs: profesors Dr. dat. Andris Ambainis

RĪGA 2017

ANOTĀCIJA

Čaulu programma ir lineāri algebrisks skaitļošanas modelis, ar kura palīdzību var konstruēt programmas Būla funkciju rēķināšanai. Ir zināms, kā čaulu programmu var pārtaisīt par kvantu vaicājošo algoritmu. Pie tam, čaulu programmām var definēt sarežģītību tā, ka pārtaisītajam kvantu algoritmam sarežģītība sakristu ar čaulu programmas sarežģītību. Līdz ar to čaulu programmas ir spēcīgs rīks kvantu algoritmu konstruēšanai.
Ir zināms veids, kā uztaisīt čaulu programmu, kura rēķinātu Būla formulu $F(x_1, ..., x_n)$, kas sastāv no loģiskajiem elementiem (NOT, OR, AND). Šī darba mērķis ir izveidot metodi, ar kuras palīdzību varētu konstruēt pēc iespējas optimālas čaulu programmas, kuras rēķinātu Būla funkcijas, balstoties uz lēmumu kokiem.

Atslēgvārdi: čaulu programma, lēmumu koks, kvantu vaicājošie algoritmi, Būla funkcijas.

ABSTRACT

Constructing quantum algorithms via span programs

Span program is a linear-algebraic model of computation. It can be used for constructing programs that compute Boolean functions. It is well known, that any span program can be transformed into quantum query algorithm. Moreover, complexity of span program can be defined in a way, that transformed quantum query algorithm would have the same complexity. This means that span program is a powerful tool for constructing quantum query algorithms.
Given a Boolean formula $F(x_1, ..., x_n)$ consisting of NOT, OR, AND gates, one can transform it to span program. The main goal of this work is to create a method for constructing span programs using a given decision tree.

Keywords: span program, decision tree, quantum query algorithms, Boolean functions.

SATURS

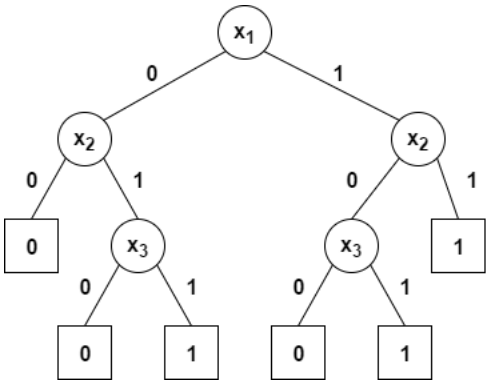
Ievads

Kvantu skaitļošana ir svarīga datorzinātnes nozare, kas ir kvantu mehānikas un datorzinātnes apvienojums. Kvantu skaitļošanai ir liels potenciāls, jo tā var būt pielietojamā: kriptogrāfijā, ķīmisko reakciju modelēšanā, meklēšanā un pārlases pāatrināšanā.
Būla funkciju rēķināšanai kvantu pasaule lieto kvantu vaicājošos algoritmus. Ar $Q_2(f)$ apzīmēsim labākā kvantu vaicājošā algoritma sarežģītību. Interesanti ka $Q_2(f)$ var ierobežot no apakšas $Q_2(f) = \Omega(Adv^{\pm}(f))$, kur $Adv^{\pm}(f)$ ir aprakstāms ar vienkāršu algebrisku izteiksmi, jo klasiski nekas tāds nav zināms.
Kvantu vaicājošo algoritmu sarežģītība ir saistīta ar čaulu programmām. Čaulu programma ir lineāri algebrisks skaitļošanas modelis [8]. Salīdzinoši nesen čaulu programmām tika ieviests sarežģītības mērs, liecinieka izmērs [3]. Vēlāk bija pierādīts $Q_2(f) = O(WSIZE(f))$ [4]. Bet, tā kā $WSIZE(f)$ ir duāls ar $Adv^{\pm}(f)$, tad $Q_2(f) = \Theta(WSIZE(f)) = \Theta(Adv^{\pm}(f))$. Savukārt tas nozīmē, ka no efektīvas čaulu programmas var iegūt efektīvu kvantu vaicājošo algoritmu.
Ir zināms veids, kā iegūt čaulu programmu, kura rēķinātu Būla funkciju $F(x_1, ..., x_n)$, kas sastāv no loģiskajiem elementiem (NOT, OR, AND). Savā darbā autors kā pamatelementu izvēlās funkciju IF THEN ELSE, šai funkcijai ir liels potenciāls gadījumā, kad mērķis ir pārveidot lēmumu koku par čaulu programmu.
Savā kursa darbā autors ir atradis veidu, kā apvienot divas funkcijas f un g funkcijā IF x THEN f ELSE g [1]. Līdz ar to bakalaura darbā autors mēģina novērtēt sarežģītību funkcijai, kas ir uzdota ar lēmumu koku ar dziļumu, kas nepārsniedz d un virsotņu skaitu, kas nepārsniedz v. Autoram izdevās iegūt sekojošos augšējos novērtējumus $O(\sqrt{d \log_2 v})$ un $O(\sqrt{d} \log_d v)$. Šie novērtējumi palīdz uzrādīt sakarību starp čaulu programmām un bumbas meklēšanas modeli [7], salīdzinoši nesen atklāto modeli, kurš palīdz radīt jaunus kvantu algoritmus. Visbeidzot, šajā darbā autors turpina meklēt optimālāko čaulu programmu funkcijai IF x_1 THEN x_2 ELSE $x_3 \vee x_4$. Autors izstrādāja programmatūru, ar kuras palīdzību ir ērtāk meklēt optimālākas čaulu programmas. Bet tomēr atrast optimālāko čaulu programmu funkcijai IF x_1 THEN x_2 ELSE $x_3 \vee x_4$ autoram neizdevās.
Darbs ir sadalīts vairākās nodaļās. Pirmajā nodaļa autors definē problēmas nostādni un biežāk lietotos apzīmējumus. Otrajā daļa autors pierāda augšējos un apakšējos novērtējumus čaulu programmām, kuras rēķina lēmumu kokus, kā arī pierāda bumbas meklēšanas konstrukciju ar čaulu programmām. Trešajā nodaļa autors uzrāda savus pašreizējos rezultātus optimālas čaulu programmas meklēšanai funkcijai IF x_1 THEN x_2 ELSE $x_3 \vee x_4$. Pēdējā nodaļā ir apkopoti darba rezultāti.

1. Problēmas apraksts

1.1. Vaicājumu sarežģītība

Pieņemsim ka f ir dota Būla funkcija kas ievadā pieņem n bitus un atgriež 1 bitu: $f : \{0,1\}^n \to \{0,1\}$. Vaicājumu sarežģītība pēta cik ieejas bitus ir nepieciešams pavaicāt lai uzzinātu funkcijas vērtību. Piemēram, ja f ir AND funkcija un ir pavaicāts bits kas ir 0, tad jau ir zināms ka funkcijas vērtība ir 0 un nav nepieciešams vaicāt pārējus bitus. Tomēr AND funkcijai sliktākajā gadījumā, kad visi ieejas biti ir 1, ir nepieciešams vaicāt visus n bitus. Svarīgi ka šajā modelī mēs neinteresējamies par algoritma atmiņu vai izpildes laiku, bet tikai par vaicājumu skaitu. Līdz ar to vaicājumu sarežģītība ierobežo algoritma sarežģītību no apakšas.
Jebkādu determinēto algoritmu kurš rēķina Būla funkciju f var aplūkot kā lēmumu koku, kur pavaicāta bita vērtība atbilst koka šķautnei, bet informācija par jau pavaicātiem bitiem atbilst koka virsotnei. (skat. 1. att.) Skaidrs ka garākais koka dziļums ir vienāds ar



1. att. Lēmumu koks funkcijai $Maj(x_1, x_2, x_3)$

vaicājumu skaitu sliktākajā gadījumā. Ar $D(f)$ apzīmē labāka algoritma, kurš rēķina f, garāko lēmumu koka dziļumu.
Kvantu pasaulē vaicājumu sarežģītība atšķiras no vaicājumu sarežģītības klasiskā pasaulē. Kvantu pasaulē vaicājums ir unitāras transformācijas pielietošana kvantu stāvoklim: $U|i\rangle = (-1)^{x_i}|i\rangle$. Ar $Q_2(f)$ apzīmē minimālo kvantu vaicājumu skaitu kas ir nepieciešams lai izrēķināt funkcijas f vērtību ar ierobežotu kļūdu (algoritms atgriež pareizo atbildi ar varbūtību $\geq \frac{2}{3}$). Piemēram, lai aprēķināt funkciju $x_1 \oplus x_2$ kvantu vaicājošam algoritmam pietiek ar $Q_2(f) = 1$ vaicājumu, lai aprēķināt funkciju $\bigvee_{i=1}^{n} x_i$ kvantu vaicājošam algoritmam pietiek ar $Q_2(f) = O(\sqrt{n})$ vaicājumiem.
Tālāk paskaidrosim ka kvantu vaicājošie algoritmi ir saistītie ar čaulu programmām.

1.2. Čaulu programmas

Čaulu programma ir lineāri algebrisks veids, kā uzdot Būla funkciju f. Formāli čaulu programma $P_f = (n, d, t, V_{free}, \{V_{i,b}\})$, kur t ir $d-$dimensionāls vektors iekš $\mathbb{C}^d$ un $V_{free}, V_{1,0}, V_{1,1}, V_{2,0}, V_{2,1}, ..., V_{n,1}$ ir multikopas, kas sastāv no $d-$dimensionālajiem vektoriem iekš $\mathbb{C}^d$. Čaulu programma P_f atbilst funkcijai f:

$$f(x) = \begin{cases} 1, \text{ ja } t \in span(V_{free} \cup \bigcup_{i=1}^{n} V_{i,x_i}) \\ 0, \text{ citādi} \end{cases}$$

Ieejai x ar A_x apzīmēsim matricu, kuras kolonnas ir vektori no multikopas $V_{free} \cup \bigcup_{i=1}^{n} V_{i,x_i}$, un ar U_x apzīmēsim matricu, kuras kolonnas ir vektori no multikopas $\bigcup_{i=1}^{n} V_{i,\neg x_i}$.
Ja $f(x) = 1$, tad eksistē tāds w, ka $A_x w = t$. Vektoru w sauc par pozitīvo liecinieku.

Definīcija 1. *Par čaulu programmas P_f pozitīva liecinieka izmēru sauc:*

$$WSIZE^+(P_f) = \max_{x:f(x)=1} \min_{A_x w = t} \|w\|^2$$

Ja $f(x) = 0$, tad vektora t projekcija uz $span(V_x)$ nevar būt vienāda ar t. Un līdz ar to eksistē tāds vektors w, kurš ir perpendikulārs katram vektoram no $span(V_x)$, bet nav perpendikulārs t. Šādu vektoru w sauc par negatīvo liecinieku. Piezīme, vektoram w ir jābūt normētam tā, ka $|(w,t)| = 1$.

Definīcija 2. *Par čaulu programmas P_f negatīvā liecinieka izmēru sauc:*

$$WSIZE^-(P_f) = \max_{x:f(x)=0} \min_{A_x^\dagger w = 0 \ \wedge \ |(w,t)|=1} \|U_x^\dagger w\|^2$$

Definīcija 3. *Par čaulu programmas sarežģītību (liecinieka izmēru) sauc:*

$$WSIZE(P_f) = \sqrt{WSIZE^+(P_f) WSIZE^-(P_f)}$$

1.3. Risinātā problēma

Definīcija 4. *Ar ITE apzīmēsim Būla funkciju, kas ir definēta trim Būla funkcijām f, g, h un ir vienāda ar $ITE(f,\ g,\ h) =$ **if** f **then** g **else** h*

Definīcija 5. *Definēsim $f_{ITE} : \mathbb{R}^2 \to \mathbb{R}$ šādi: $f_{ITE}(x,y) = \frac{x+y+\sqrt{(x-y)^2+4}}{2}$*

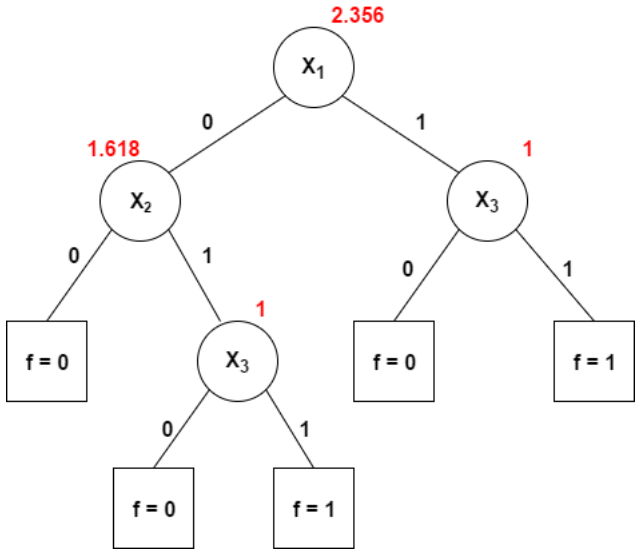
Šajā darbā tiek turpināts autora kursa darbs [1]. Savā kursa darbā autors ir pētījis ITE funkcijas ar čaulu programmām. Precīzāk, ir dota Būla funkcija $ITE(x_0,\ f(x_1,\ x_2, ..., x_n),\ g(x_1,\ x_2, ..., x_n))$ un pie tā jau ir zināmas čaulu programmas ar liecinieku izmēriem W_f un W_g funkcijām f un g respektīvi, kursa darba mērķis bija uztaisīt pēc iespējas optimālāku čaulu programmu ITE funkcijai.
Kursa darbs bija motivēts ar sekojošo novērojumu: darbā [2] ir parādīts kā uztaisīt čaulu programmas funkcijām $f \vee g$ un $f \wedge g$ ar liecinieka izmēru $\sqrt{W_f^2 + W_g^2}$, ja ir zināmas čaulu programmas funkcijām f un g, kur W_f ir funkciju W_g liecinieka izmēri. Tas nozīmē kā funkcijai $ITE(x_0, f, g) = (x_0 \wedge f) \vee (\neg x_0 \wedge g)$ eksistē čaulu programma ar liecinieka izmēru $\sqrt{W_f^2 + W_g^2 + 2}$. Ja $W_f = W_g$, šis novērtējums ir aptuveni $\sqrt{2} W_f$. Tajā pašā laikā, pavaicājot bitu x_0 klasiski un, atkarībā no rezultāta, vaicājot funkciju f vai g, mēs varam panākt sarežģītību $1 + W_f$. Šīs piemērs motivē meklēt labākas čaulu programmas funkcijai ITE.
Kursa darbā bija pierādīta sekojoša teorēma:

Teorēma 1. *Funkcijai $ITE(x_0,\ f(x_1,\ x_2, ..., x_n),\ g(x_1,\ x_2, ..., x_n))$ var konstruēt čaulu programmu ar liecinieka izmēru*

$$f_{ITE}(W_f, W_g) = \frac{W_f + W_g + \sqrt{(W_f - W_g)^2 + 4}}{2}$$

kur W_f, W_g ir funkciju f un g liecinieku izmēri.

Šī teorēma dod iespēju pārtaisīt lēmumu koku par čaulu programmu. Ja T_f ir funkcijas f lēmumu koks, x_1 ir pirmais mainīgais, kuru jautā lēmumu koks, T_{f_0} ir lēmumu koks funkcijai kur $x_1 = 0$ un T_{f_1} ir lēmumu koks funkcijai kur $x_1 = 1$, tad f =ITE(x_1, f_0, f_1) skat 2. att.

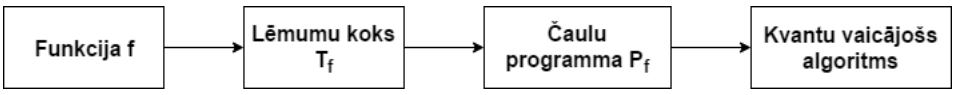


2. att. $D(T) = 3$, $V(T) = 4$ un $TW(T) \approx 2.356$

Definīcija 6. *Ja T ir lēmumu koks, tad:*

- *Ar $D(T)$ apzīmēsim lēmumu koka dziļumu.*
- *Ar $V(T)$ apzīmēsim lēmumu koka virsotņu skaitu.*
- *Ar $TW(T)$ apzīmēsim čaulu programmas, kas iegūtas, lietojot kursa darba konstrukciju, [1], liecinieka izmēru funkcijai, kas ir uzdota ar lēmumu koku T.*

Rakstā [3] ir uzrādīts, kā čaulu programmu var pārveidot par kvantu vaicājošu algoritmu. Šīs rezultāts kopā ar teorēmu 1 dod jaunu veidu kvantu algoritmu konstruēšanai:



3. att. Veids, kā var konstruēt kvantu vaicājošo algoritmu

Šajā transformācijā nav skaidrs, kā zinot lēmumu koka struktūru, novērtēt rezultējoša kvantu algoritma sarežģītību $Q_2(f)$. Rakstā [4] ir pierādīts, ka $Q_2(f) = O(WSIZE(f))$. Tātad, lai novērtētu $Q_2(f)$ caur lēmumu koku, pietiek novērtēt $WSIZE(f)$ caur lēmumu koku.
Ja ņemt vērā tikai funkcijas determinēto sarežģītību $D(f)$(kas atbilst lēmumu koka dziļumam), tad gadījumā, ja lēmumu koks ir pilns $Q_2(f) = O(D(f))$. Kas nedod priekšrocību kvantu algoritmiem salīdzinājot ar parastiem algoritmiem. Šis piemērs parāda , ka labākam $Q_2(f)$ novērtējumam vajadzētu ņemt vēra ne tikai funkcijas f determinēto sarežģītību $D(f)$. Darba 2. nodaļā autors mēģina novērtēt $Q_2(f)$, zinot lēmumu koka dziļumu un virsotnes skaitu.

Definīcija 7. *Ar $W(v,d)$ apzīmēsim maksimālo liecinieka izmēru kokam ar v virsotnēm un dziļumu, kas nepārsniedz d.*

$$W(v,d) = \max_{D(T)\leq d \ \wedge \ V(T)=v} TW(T)$$

Darba 3. nodaļā autors aplūko funkciju $ITE(x_1,\ x_2,\ x_3 \vee x_4)$. Kursa darba autors atradis šai funkcijai čaulu programmu ar liecinieku $\frac{1+\sqrt{2}+\sqrt{7-2\sqrt{2}}}{2} \approx 2.228$, kas sakrīt ar labāko jau zināmo liecinieku izmēru [5]. Tapāt ir zināms, ka optimālākai čaulu programmai funkcijai $ITE(x_1,\ x_2,\ x_3 \vee x_4)$ liecinieka izmērs ir ≈ 2.208.
Autoram neizdevās uzlabot kursa darba rezultātu, bet 3. nodaļā autors apraksta savus uzlabošanas mēģinājumus un pašreizējos rezultātus.

2. Lēmumu koka analīze

2.1. Novērtējumi

Šajā nodaļā tiks sniegti asimptotiski novērtējumi funkcijai $W(v,d)$. Autors pierāda vienu apakšējo novērtējumu un divus augšējos:

- $W(v,d) = \Omega\left(\sqrt{\frac{d\log_2 v}{\log_2 \frac{d}{\log_d v}}}\right)$
- $W(v,d) = O(\sqrt{d}\log_d v)$
- $W(v,d) = O(\sqrt{d\log_2 v})$

Visbeidzot, autors pierādis bumbu meklēšanas algoritma konstrukciju ([6] teorēma 8), balstoties tikai uz čaulu programmām.

Lemma 1. *Funkcijas f_{ITE} īpašības:*

(a) $f_{ITE}(x,y)$ ir augoša pa katru mainīgo.

(b) $f_{ITE}(x+t, y+t) = f_{ITE}(x,y) + t$

Pierādījums:

(a) $\frac{df_{ITE}}{dx} = \frac{1}{2}\left(1 + \frac{x-y}{\sqrt{(x-y)^2+4}}\right) > 0$. Gadījums $\frac{df_{ITE}}{dy} > 0$ ir līdzīgs.

(b) Uzreiz seko no funkcijas definīcijas: $f_{ITE}(x,y) = \frac{x+y+\sqrt{(x-y)^2+4}}{2}$

Lemma 2. $W(v,d) = \max\limits_{x+y=v-1} f_{ITE}(W(x,d-1), W(y,d-1))$

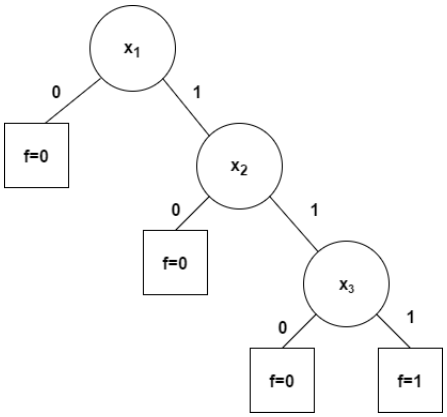
Pierādījums:
Jebkādu lēmumu koku T var sadalīt: saknes virsotnē, kreisajā apakškokā T_L un labajā apakškokā T_R. Tātad $V(T) = 1 + V(T_L) + V(T_R)$ un

$$\begin{aligned} W(v,d) &= \max_{D(T)\le d \wedge V(T)=v} TW(T) = \\ &= \max_{D(T_L),D(T_R)\le d-1 \wedge 1+V(T_L)+V(T_R)=v} f_{ITE}(TW(T_L), TW(T_R)) = \\ &= \max_{x+y=v-1} f_{ITE}(W(x,d-1), W(y,d-1)) \end{aligned}$$

Kur pēdējā vienādība seko no lemmas 1. □

Lemma 3. *Ar T_{AND_k} apzīmēsim Būla funkcijas $AND(x_1, x_2 ..., x_k)$ lēmumu koku (4. att.). Ja $s_k = TW(T_{AND_k})$, tad $s_k \ge \sqrt{k}$*

Pierādījums:
Pierādījums ar matemātisko indukciju uz k.
Bāze $k = 1$: $s_1 = 1 \ge 1$
Induktīvā pārēja: $s_{k+1} = f_{ITE}(s_k, 0) \ge f_{ITE}(\sqrt{k}, 0) = \frac{\sqrt{k}+\sqrt{k+4}}{2} \ge \sqrt{k+1}$.
Pēdējo nevienādību var pamatot ar matemātiskiem pārveidojumiem. □



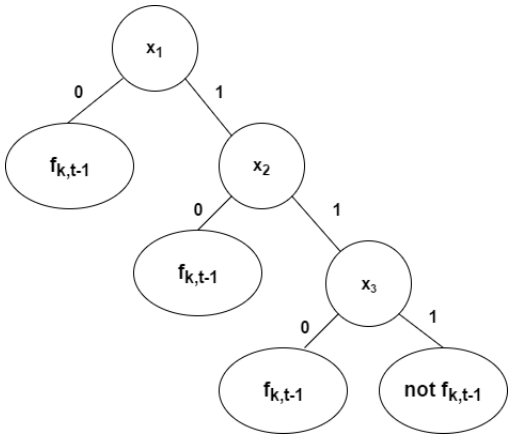
4. att. AND(x_1, x_2, x_3) lēmumu koks

Teorēma 2. *Ja* $d \geq kt$ *un* $v \geq (k+1)^t - 1$ *tad* $W(v,d) \geq t\sqrt{k}$

Pierādījums:
Apskatīsim Būla funkciju

$$f_{k,t}(x_1, x_2, ..., x_{kt}) = (x_1 \wedge x_2 \wedge ... \wedge x_k) \oplus (x_{k+1} \wedge ... \wedge x_{2k}) \oplus ... \oplus (x_{k(t-1)+1} \wedge ... \wedge x_{kt}) =$$
$$= \bigoplus_{i=1}^{t} (\bigwedge_{j=1}^{k} x_{k(i-1)+j})$$

Šai funkcijai apskatām lēmumu koku $T_{k,t}$, kur katram $\wedge$ blokam pēc kārtas tiek jautāti biti, kamēr netiek sastapts bits 0 vai visi bloka mainīgie ir 1. (skat. 5. att.)



5. att. Lēmumu koks funkcijai $f_{k,t}$, pie $k = 3$

Šim kokam izpildās:

- $D_{k,t} = kt \leq d$
- $V_{k,t} = k + (k+1)V_{k,t-1} = k + (k+1)k + (k+1)^2 V(k, t-2) = ... =$
  $= k\big(1 + (k+1) + (k+1)^2 + ...(k+1)^{t-1}\big) = (k+1)^t - 1 \leq v$
- $TW(T_{k,t}) \geq \sqrt{k} + TW(T_{k,t-1}) \geq t\sqrt{k}$
  Nevienādības seko no lemmas 1. un 3. $\square$

Teorēma 3. *Ja* $v = \Omega(d)$ *tad* $W(v,d) = \Omega\left(\sqrt{\frac{d\log_2 v}{\log_2 \frac{d}{\log_d v}}}\right)$

Pierādījums:
Ja $v < 4d$ tad pēc teorēmas 3, $W(v,d) = \Omega(\sqrt{d})$ un $\frac{log_2(v)}{\log_2 \frac{4d}{log_d(v)}} = \Theta(1)$, kas pamato vajadzīgo novērtējumu.
Ja $v \geq 4d$, tad definēsim $t = \left[log_{\frac{4d}{\log_d(v)}}(v)\right] \geq 1$ un $k = \left[\frac{d}{log_{\frac{4d}{\log_d(v)}}(v)}\right] \geq 1$.
Un izpildās sekojošās nevienādības.
$kt \leq d$

$$(k+1)^t - 1 < \left(\frac{2d}{log_{\frac{4d}{\log_d(v)}}(v)}\right)^t \overset{*}{<} \left(\frac{4d}{\log_d(v)}\right)^t \leq v$$

Nevienādība $(*)$ ir ekvivalenta ar $log_d(v) < 2log_{\frac{4d}{\log_d(v)}}(v) \Leftrightarrow 4 < dlog_d(v)$
Lai pabeigtu pierādījumu, pietiek pielietot teorēmu 2. skaitļiem k un t. □

Teorēma 4. $W(v,d) = O(\sqrt{d\log_2 v})$

Pierādījums:
$W(v,d) \leq \sqrt{8d\log_2 v} + 1$
Pieradījums ar matemātisko indukciju uz d. Viegli redzēt, ka šīs apgalvojums izpildās pie $d = 1$. Ja $d \geq 1$, tad lietot lemmu 1 un 2.

$$W(v,d+1) = \max_{x+y=v-1} f_{ITE}(W(x,d), W(y,d)) \leq \max_{x+y=v-1} f_{ITE}(\sqrt{8dlog_2(x)}, \sqrt{8dlog_2(y)}) + 1$$

Tātad pietiek pamatot:

$$\forall x, y \geq 0 : f_{ITE}(\sqrt{8dlog_2(x)}, \sqrt{8dlog_2(y)}) \leq \sqrt{8(d+1)log_2(x+y+1)}$$

Pēdējā nevienādība ir pierādāma ar algebriskiem pārveidojumiem, detaļas ir uzrakstītas 1. pielikumā. □

Teorēma 5. $W(v,d) = O(\sqrt{d}\log_d v)$

Pierādījums:
$W(v,d) \leq 20\sqrt{d}\log_d(v) + 56$
Pieradījums līdzīgs pierādījumam teorēmai 4. Bāze: $d \leq 56$
Pārējai jāpamato nevienādība:

$$\forall x, y \geq 0 : f_{ITE}(20\sqrt{d}log_d(x), 20\sqrt{d}log_d(y)) \leq 20\sqrt{d+1}log_{d+1}(x+y+1)$$

Tas tiek izdarīts ar algebriskiem pārveidojumiem, detaļas ir uzrakstītas 2. pielikumā. □

2.2. Pielietojumi

Autors uzrāda divus augšējos novērtējumus, tāpēc ka novērtējums $O(\sqrt{d}log_d(v))$ ir labs gadījumā, ja virsotņu skaits ir mazs, bet slikts gadījumā, ja virsotņu skaits ir liels. Tas ir, ja $v = poly(d)$, tad $O(\sqrt{d}log_d(v)) = O(\sqrt{d})$ un $\Omega\left(\sqrt{\frac{d\log_2 v}{\log_2 \frac{d}{\log_d v}}}\right) = \Omega(\sqrt{d})$, kas nozīmē $W(poly(d), d) = \Theta(\sqrt{d})$. Novērtējums $O(\sqrt{d\log_2 v})$ pretēji ir labs, ja virsotņu skaits ir liels,

un slikts, ja virsotņu skaits ir mazs. Šīs novērtējums arī parāda, ka augšējais novērtējums atšķiras no apakšējā ne vairāk kā $\sqrt{log_2(\frac{d}{log_d(v)})}$ reizes.

Lai redzētu aprakstītās metodes pielietojumu, apskatām Būla funkciju $TH_N^K(x_1, ...x_N)$ kā piemēru. Funkcija $TH_N^K(x_1, ...x_N)$ atgriež 1, ja vismaz K biti no $x_1, x_2...x_N$ ir 1. Funkcija $TH_N^K(x_1, ...x_N)$ jau ir izpētīta ar čaulu programmām [3]. Ir zināms, ka $WSIZE(TH_N^K) = \Theta(\sqrt{K(N-K+1)})$

Teorēma 6. *Eksistē čaulu programma, kura rēķina Būla funkciju* $TH_N^K(x_1, ...x_N)$ *ar liecinieka izmēru* $O(\sqrt{NKlog_2(\frac{N}{K})})$.

Pierādījums:
Būvējam lēmumu koku $T_{N,K}$ sekojošā veidā: vaicājam bitus $x_1, x_2...x_N$ pēc kārtas, kamēr neatrodam K vieniniekus. Nav grūti saprast:

$D(T_{N,K}) = N$

$V(T_{N,K}) = \#\{s \in \{0,1\}^* \mid |s| \leq N-1 \wedge s - \text{satur ne vairāk par } K-1 \text{ simbolu } 1\} =$

$$= \sum_{l=0}^{N-1}\sum_{v=0}^{K-1}\binom{l}{v} = \sum_{v=0}^{K-1}\sum_{l=0}^{N-1}\binom{l}{v} = \sum_{v=1}^{K}\binom{N}{v} \leq K\left(\frac{\mathrm{e}N}{K}\right)^K \leq \left(\frac{2\mathrm{e}N}{K}\right)^K$$

Piezīme: nevienādības ir spēkā tikai, ja $K \leq \frac{N}{2}$. Ja $K > \frac{N}{2}$, tad summu var novērtēt ar 2^N, un turpmākie spriedumi un rezultāts neizmainīsies.

$TW(T_{N,K}) = O(\sqrt{D(T_{N,K})log_2(V(T_{N,K})}) = O(\sqrt{NKlog_2(\frac{N}{K})})$ □

Teorēma 7. *Ja* $K = O(N^c)$, *kur* c *ir konstante* $(0 < c < 1)$, *tad* $TW(T_{N,K}) = \Omega(\sqrt{NK})$

Pierādījums:

$$V(T_{N,K}) = \sum_{v=1}^{K}\binom{N}{v} \geq \binom{N}{K} \geq \left(\frac{N}{K}\right)^K$$

$$TW(T_{N,K}) = \Omega\left(\sqrt{\frac{d\log_2 v}{\log_2 \frac{d}{\log_d v}}}\right) = \Omega\left(\sqrt{NK\frac{log_2\frac{N}{K}}{log_2(\frac{N}{Klog_N(\frac{N}{K})})}}\right) = \Omega(\sqrt{NK})$$

□

Šis piemērs parāda, ka eksistē funkcijas, kurām apakšējais novērtējums nesakrīt ar augšējo. Tālāk pamatosim, ka problēma visticamāk nav apakšējā novērtējumā Ω un nav funkcijā f_{ITE}, bet tieši augšējā novērtējumā.

Teorēma 8. $TW(T_{N,K}) = O(\sqrt{NK})$

Pierādījums:
$TW(T_{N,K}) \leq \sqrt{5NK}$ Pieradījums ar matemātisko indukciju.
Bāze:
$K = 0$: $TW(T_{N,K}) = 0 \leq \sqrt{5 \cdot 0 \cdot N}$
$K = N$: $TW(T_{N,K}) = N \leq \sqrt{5 \cdot K \cdot N}$
Pārēja:

$$TW(T_{N,K}) = f_{ITE}(TW(T_{N-1,K}), TW(T_{N-1,K-1})) \leq$$
$$\leq f_{ITE}(\sqrt{5(N-1)K}, \sqrt{5(N-1)(K-1)})$$

Tātad pietiek pamatot:

$$f_{ITE}(\sqrt{5(N-1)K}, \sqrt{5(N-1)(K-1)}) \leq \sqrt{5NK}$$

Tas tiek izdarīts ar algebriskiem pārveidojumiem, detaļas ir uzrakstītas 3. pielikumā. □

2.3. Saistība ar bumbas meklēšanas modeli

Bumbas meklēšanas modelis ir nesen izgudrots modelis [6]. Šajā modelī kvantu algoritmi ir konstruējami, balstoties uz sekojošu teorēmu:

Teorēma 9. *([7] Teorēma 3.1) Pieņemsim, ka f ir funkcija: $f : D \to E$, kur $D \subseteq \{0,1\}^N$. Pieņemsim, ka A ir klasiskais varbūtisks vaicājumu algoritms, kas izrēķina $f(x)$ ar ierobežotu kļūdas varbūtību un izdara ne vairāk par T vaicājumiem. Pieņemsim, ka Π ir varbūtisks algoritms, kas katrā solī, "zinot" A iekšējo stāvokli (zinot iepriekšējo pavaicāto bitu vērtības un kārtējā vaicāta bita pozīciju x_{ind}), spēj uzminēt bita x_{ind} vērtību, katram x vidēji kļūdoties ne vairāk kā G reizes. Šādā gadījumā pastāv kvantu algoritms, kas izrēķina f ar ierobežotu kļūdas varbūtību un izdara $O(\sqrt{TG})$ vaicājumus.*

Autors paradīs, kā var pierādīt nedaudz vājāku teorēmu ar čaulu programmām. Un līdz ar to noreducēt bumbas meklēšanas modeli (vājāko variantu) uz čaulu programmām.

Teorēma 10. *Pieņemsim, ka f ir Būla funkcija: $f : \{0,1\}^N \to \{0,1\}$. Pieņemsim, ka A ir determinēts vaicājumu algoritms, kas izrēķina $f(x)$ un izdara ne vairāk par T vaicājumiem. Pieņemsim, ka Π ir varbūtisks algoritms, kas katrā solī, "zinot" A iekšējo stāvokli (zinot iepriekšējo pavaicāto bitu vērtības un kārtējā vaicātā bita pozīciju x_{ind}), spēj uzminēt bita x_{ind} vērtību, katram x vidēji kļūdoties ne vairāk kā G reizes. Šādā gadījumā pastāv čaulu programma, kas izrēķina f ar liecinieka izmēru $O(\sqrt{TG})$.*

Pierādījums:
Tā kā A ir determinēts vaicājumu algoritms, tad A var pārtaisīt par lēmumu koku T_A. Skaidrs ka šim kokam izpildās $D(T_A) \leq T$. Apzīmēsim ar $p_{s,0}$ un $p_{s,1}$ varbūtību, ka algoritms Π, atrodoties stāvokli s, minēs bitu $0, 1$ respektīvi. 6. att.. Skaidrs ka $p_{s,0}+p_{s,1} = 1$ un

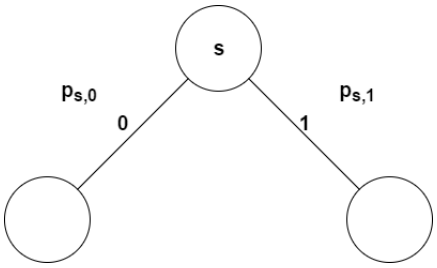
$$G \geq E_\Pi[\text{nepareizo minējumu skaits uz ieejas } x] = \sum_{A(x) \text{ apmeklē } s_i} p_{s_i,\neg x_i}$$

Definēsim algoritmu Π_{det} ar sekojošām varbūtībām:

$$p'_{s,0} = \begin{cases} 0, \text{ ja } p_{s,0} \leq p_{s,1} \\ 1, \text{ ja } p_{s,0} > p_{s,1} \end{cases}$$
$$p'_{s,1} = 1 - p'_{s,0}$$

Tagad Π_{det} ir determinēts algoritms un

$$E_{\Pi_{det}}[\text{nepareizo minējumu skaits uz ieejas } x] = \sum_{A(x) \text{ apmeklē } s_i} p'_{s_i,\neg x_i} \leq$$
$$\leq \sum_{A(x) \text{ apmeklē } s_i} 2p_{s_i,\neg x_i} \leq 2G$$



6. att. Pārejas un tos varbūtības

Tagad, ja mēs izmainām koka T_A pārēju vērtības tā, lai pāreja $x_i = 1$ sakrīt ar algoritma Π_{det} neuzminēšanu, tad jaunais koks un līdz ar to arī koks T_A ir nepilns funkcijas TH_T^{2G} lēmumu koks. Un tad, lietojot teorēmu 7, sanāk $TW(T_A) = O(\sqrt{TG})$ □

Darbā [7] ir aprakstīts veids ka ar bumbas meklēšanas modeli atrisināt sekojošus uzdevumus:

- Dota Būla funkcija $f : \{0,1\}^N \to \{0,1\}$. Un $f(x) = 1$, ja eksistē 2 vienīnieki attālumā $< K$. Šai funkcija var uzkonstruēt kvantu vaicājošu algoritmu ar sarežģītību $Q_2(f) = O(\frac{N}{\sqrt{K}})$ lietojot teorēmu 8.
- Dota Būla funkcija $f : \{0,1\}^N \to \{0,1\}$. Un $f(x) = 1$, ja eksistē K vienīnieki pēc kārtās. Šai funkcijai var uzkonstruēt kvantu vaicājošu algoritmu ar sarežģītību $Q_2(f) = O(\frac{N}{\sqrt{K}})$, lietojot teorēmu 8.

Tātad tos pašus rezultātus var iegūt arī ar čaulu programmām. (Piezīme, raksta [7] algoritms A augstāk definētiem uzdevumiem ir determinēts)
Tālāk autors nodemonstrē, kā ar čaulu programmām var atrisināt grafa sakarības un grafa divdaļības problēmas, kuras iepriekš pētītas [2, 9] ar algoritmiem, kas iegūti citā veidā.

Teorēma 11. *Dota $N \times N$ matrica $x_{i,j} \in \{0,1\}$. Šķautne starp grafa virsotnēm i un j eksistē tad un tikai tad, ja $x_{i,j} = 1$. Būla funkcijām:*

(a) Grafa sakarība

(b) Grafa divdaļīgums

var uzkonstruēt čaulu programmas ar liecinieka izmēriem $O(N\sqrt{N})$

Pierādījums:
Pieradījumam pielietosim teorēmu 9.

(a) Vaicājam šķautnes, kuras pašlaik nepieder vienai komponentei, minot $x_{i,j} = 0$. Viegli saprast, ka $T \leq N^2$ un $G \leq N - 1$.

(b) Vaicājam šķautnes, kuras pašlaik nepieder vienai komponentei vai veido ciklu nepāra garumā, minot $x_{i,j} = 0$. Viegli saprast, ka $T \leq N^2$ un $G \leq N$

Piezīme, gadījumu (a) var paplašināt gadījumam, ja grafs ir orientēts un vajag noskaidrot, vai no dotas virsotnes v var nonākt līdz jebkādai citai virsotnei. □

3. Funkcijas ar svariem (1, 1, $\sqrt{2}$) analīze

3.1. Iepriekšējie rezultāti

Šajā nodaļā aplūkojam Būla funkciju $ITE(x_1, x_2, x_3 \vee x_4)$. Šai funkcijai ar datoru pārlasi ir atrasta čaulu programma ar liecinieka izmēru ≈ 2.228, un arī ar datoru pārlasi ir atrasta precīza apakšēja robeža liecinieku izmēram ≈ 2.208. [5]

Savā kursa darbā autoram izdevās atrast čaulu programmu ar liecinieka izmēru $\sqrt{\frac{1+\sqrt{2}+\sqrt{7+2\sqrt{2}}}{2}} \approx 2.228$:

$$
\begin{aligned}
&x_1 = 0 \to \sqrt{\frac{\sqrt{2}-1+\sqrt{7-2\sqrt{2}}}{2}} e_1 \\
&x_1 = 1 \to \sqrt{\frac{-\sqrt{2}+1+\sqrt{7-2\sqrt{2}}}{\sqrt{2}}} e_2 \\
&x_2 = 1 \to e_1 \\
&x_3 = 1 \to e_2 \\
&x_4 = 1 \to e_2 \\
&t = e_1 + \sqrt[4]{2} e_2
\end{aligned}
$$

kur e_1 un e_2 ir bāzes vektori.
Kursa darbā autors ir arī pierādījis sekojošu teorēmu:

Teorēma 12. *Ja P_f ir čaulu programma kas rēķina funkciju $ITE(x_1, x_2, x_3 \vee x_4)$ un vērtībām $x_2 = 0$, $x_3 = 0$ un $x_4 = 0$ neatbilst neviens vektors, tad $WSIZE(P_f) \geq \sqrt{\frac{1+\sqrt{2}+\sqrt{7+2\sqrt{2}}}{2}}$*

3.2. Optimālas čaulu programmas meklēšana

Savā bakalaura darbā autors turpina meklēt optimālu čaulu programmu $ITE(x_1, x_2, x_3 \vee x_4)$ funkcijai. Autors meklē čaulu programmu forma, kur katrai bitu vērtībai atbilst viens vektors no $\mathbb{C}^3$:

$$
\begin{aligned}
&x_1 = 0 \to v_1 \\
&x_1 = 1 \to v_2 \\
&x_2 = 0 \to v_3 \\
&x_2 = 1 \to v_4 \\
&x_3 = 0 \to v_5 \\
&x_3 = 1 \to v_6 \\
&x_4 = 0 \to v_7 \\
&x_4 = 1 \to v_8
\end{aligned}
\tag{1}
$$

Teorēma 13. *Ja čaulu programma rēķina $ITE(x_1, x_2, x_3 \vee x_4)$ un ir forma 1, tad tā atbilst vienai no:*

1)	2)	3)
$x_1 = 0 \to z_1 e_1$	$x_1 = 0 \to z_1 e_1$	$x_1 = 0 \to z_1 e_1$
$x_1 = 1 \to z_2 e_2$	$x_1 = 1 \to z_2 e_2$	$x_1 = 1 \to z_2 e_2$
$x_2 = 0 \to z_3 e_3$	$x_2 = 0 \to \emptyset$	$x_2 = 0 \to z_3 e_1 + z_4 e_2 + z5 e_3$
$x_2 = 1 \to z_4 e_1 + z_5 e_3$	$x_2 = 1 \to z_3 e_1 + z_4 e_3$	$x_2 = 1 \to z_6 e_3$
$x_3 = 0 \to z_6 e_3$	$x_3 = 0 \to z_5 e_3$	$x_3 = 0 \to \emptyset$
$x_3 = 1 \to z_7 e_2 + z_8 e_3$	$x_3 = 1 \to z_6 e_1 + z_7 e_2 + z_7 e_3$	$x_3 = 1 \to z_7 z_3 e_1 + z_8 e_2 + z_7 z_5 e_3$
$x_4 = 0 \to z_9 e_3$	$x_4 = 0 \to \emptyset$	$x_4 = 0 \to \emptyset$
$x_4 = 1 \to z_{10} e_2 + z_{11} e_3$	$x_4 = 1 \to z_8 e_2$	$x_4 = 1 \to z_7 z_3 e_1 + z_8 e_2 + z_7 z_5 e_3$
$t = e_1 + e_2 + e_3$	$t = e_1 + e_2 + e_3$	$t = e_2 + e_3$

kur e_i ir bāzes vektori un z_i ir reāli skaitļi.

Pierādījums:
Lai pierādījums nebūtu pārāk garš, paslēpsim daudzas detaļas uzrādot tikai pierādījuma galvenos apgalvojumus.

- $dim(span(v_5, v_7)) \leq 1$
- Ja $dim(span(v_5, v_7)) = 1$ tad $dim(span(v_1, v_2, v_5)) = 3$. Un tātad katru vektoru var pierakstīt caur v_1, v_2, v_5. Pēc tam var pamatot ka koeficienti pie vektoriem atbilstīs gadījumam 1. vai 2.
- $dim(span(v_5, v_7)) = 0$. Ja $dim(span(v_1, v_2, v_4)) = 2$ tad visi čaulu programmas vektori atrodas apakštelpā ar dimensiju 2 un kursa darba rezultātu uzlabot nevar. Ja $dim(span(v_1, v_2, v_4)) = 3$ tad čaulu programma atbilst gadījumam 3.

□

Katra no trim gadījumiem čaulu programmas liecinieka izmērs ir funkcija $g(z_1, ..., z_n)$ no vairākiem mainīgiem. Autors ir uztaisījis programmatūru kura spēj aprēķināt $g(z_1, ..., z_n)$ vērtību katra punkta. Lietojot g funkciju kā melno kasti, izstrādāta programma spēj samazināt funkcijas g vērtību. Lietojot izstrādāto programmu uz trim augstāk aprakstītiem gadījumiem, autoram neizdevās uzlabot kursa darba rezultātu.

Rezultāti

Šajā darbā ir atrasta metode, kā lietojot čaulu programmas, var pārveidot lēmumu koku par kvantu vaicājošo algoritmu. Rezultējošā kvantu algoritma sarežģītību $Q_2(f)$ var novērtēt caur lēmumu koka dziļumu d un virsotņu skaitu v.

1. $Q_2(f) = O(\sqrt{d \log_2 v})$
2. $Q_2(f) = O(\sqrt{d} \log_d v)$

Autors arī uzrāda šīs metodes apakšējo robežu $Q_2(f) = \Omega\left(\sqrt{\frac{d \log_2 v}{\log_2 \frac{d}{\log_d v}}}\right)$. Ar šīs metodes palīdzību autoram izdevās pamatot vājāko variantu bumbu meklēšanas algoritma konstrukcijai: teorēma 10.
Visbeidzot, autors ir izstrādājis rīku, ar kuras palīdzību mēģinājis atrast optimālo čaulu programmu funkcijai $ITE(x_1,\ x_2,\ x_3 \vee x_4)$.

Pateicības

Autors pateicas kursa darba vadītājam Andrim Ambainim par interesanto piedāvāto pētniecisko tēmu, kā arī par vērtīgiem padomiem un palīdzību darba izstrādes procesā.
Tāpat autors pateicas Kristapam Čivkulim par palīdzību bakalaura darba noformēšanā.

Literatūra

[1] N. Larka, "Kvantu algoritmu konstruēšana, izmantojot čaulu programmas," *Kursa darbs, LU datorikas fakultāte,* 2016.

[2] A. Āriņš, "Kvantu vaicājošie algoritmi čaulu programmu skaitļošanas modelī," *Maģistra darbs, LU Datorikas fakultāte,* 2013

[3] B. Reichardt, "Span programs and quantum query complexity: The general adversary bound is nearly tight for every boolean function," arxiv.org/abs/0904.2759 (2009)

[4] B. Reichardt, "Reflections for quantum query algorithms," arXiv:1005.1601v1 (2010)

[5] R. Špalek, "Designing efficient span programs," WQACT, http://www.ucw.cz/ robert/talks.html (2008)

[6] Cedric Yen-Yu Lin, Han-Hsuan Lin, "Upper bounds on quantum query complexity inspired by the Elitzur-Vaidman bomb tester," arxiv.org/abs/1410.0932 (2014)

[7] A. Kuzņecovs, "Kvantu algoritmi bumbu meklēšanas modelī", *Bakalaura darbs, LU datorikas fakultāte,* 2016

[8] M. Karchmer and A. Wigderson, "On span programs", In Proceedings of the IEEE 8th Annual Conference on Structure in Complexity Theory, pages 102–111, (1993)

[9] C. Cade, A. Montanaro and A. Belovs, "Time and Space Efficient Quantum Algorithms for Detecting Cycles and Testing Bipartiteness", arxiv.org/abs/1610.00581

Pielikumi

1. pielikums. Novērtējuma $O(\sqrt{dlog_2 v})$ pierādījums

Pamatosim nevienādību:

$$\forall x, y \geq 0 \wedge 2^d > x + y > 0 : f_{ITE}(\sqrt{8dlog_2(x)}, \sqrt{8dlog_2(y)}) \leq \sqrt{8(d+1)log_2(x+y+1)}$$

Piezīme: šeit mēs pieņēmām $log_2(0) = 0$
Pierādījums ar matemātiskiem pārveidojumiem:

$$2^d > x \geq y \geq 1 : f_{ITE}(\sqrt{8dlog_2(x)}, \sqrt{8dlog_2(y)}) \leq \sqrt{8(d+1)log_2(x+y)}$$
$$\frac{1}{2}\Big(\sqrt{8dlog_2(x)} + \sqrt{8dlog_2(y)} + \sqrt{\big(\sqrt{8dlog_2(x)} - \sqrt{8dlog_2(y)}\big)^2 + 4}\Big) \leq \sqrt{8(d+1)log_2(x+y)}$$
$$\sqrt{log_2(x)} + \sqrt{log_2(y)} + \sqrt{\big(\sqrt{log_2(x)} - \sqrt{log_2(y)}\big)^2 + \frac{1}{2d}} \leq 2\sqrt{\frac{(d+1)log_2(x+y)}{d}}$$
$$2(log_2(x) + log_2(y)) + \frac{1}{2d} + 2\sqrt{(log_2 x - log_2 y)^2 + \frac{(\sqrt{(log_2(x)} + \sqrt{log_2(y)})^2}{2d}} \leq \frac{4(d+1)log_2(x+y)}{d}$$
$$\frac{1}{4d} + \sqrt{\Big(log_2\Big(\frac{x}{y}\Big)\Big)^2 + \frac{(\sqrt{log_2(x)} + \sqrt{log_2(y)})^2}{2d}} \leq \frac{2log_2(x+y)}{d} + log_2\Big(1 + \frac{x}{y}\Big) + log_2\Big(1 + \frac{y}{x}\Big)$$

- $\frac{x}{y} \leq 2$

$$\frac{1}{4d} + \sqrt{\Big(log_2\Big(\frac{x}{y}\Big)\Big)^2 + \frac{(\sqrt{log_2(x)} + \sqrt{log_2(y)})^2}{2d}} \leq \frac{1}{4d} + \sqrt{1 + \frac{4d}{2d}} < 2 \leq$$
$$\leq log_2\Big(1 + \frac{x}{y}\Big) + log_2\Big(1 + \frac{y}{x}\Big)$$

- $\frac{x}{y} > 2$

$$\frac{1}{4d} + \sqrt{\Big(log_2\Big(\frac{x}{y}\Big)\Big)^2 + \frac{(\sqrt{log_2(x)} + \sqrt{log_2(y)})^2}{2d}} \leq$$
$$\leq \frac{1}{4d} + log_2\frac{x}{y} + \frac{(2\sqrt{log_2(x+y)})^2}{4d} \leq$$
$$\leq \frac{1}{4d} + log_2\frac{x}{y} + \frac{log_2(x+y)}{d} \leq$$
$$\leq log_2\frac{x}{y} + \frac{2log_2(x+y)}{d}$$

□

2. pielikums. Novērtējuma $O(\sqrt{d}log_d v)$ pierādījums

Pamatosim nevienādību:

$$\forall x, y \geq 0 \wedge 2^d > x + y > 0 : f_{ITE}(20\sqrt{d}log_d(x), 20\sqrt{d}log_d(y)) \leq 20\sqrt{d+1}log_{d+1}(x+y+1)$$

Piezīme: šeit mēs pieņēmām $log_d(0) = 0$
Pierādījums ar matemātiskiem pārveidojumiem:

$2^d > x \geq y \geq 1 : f_{ITE}(20\sqrt{d}log_d(x), 20\sqrt{d}log_d(y)) \leq 20\sqrt{d+1}log_{d+1}(x+y)$
$\frac{1}{2}\Big(20\sqrt{d}(log_d(x) + log_d(y)) + \sqrt{\big(20\sqrt{d}log_d(x) - 20\sqrt{d}log_d(y)\big)^2 + 4)}\Big) \leq 20\sqrt{d+1}log_{d+1}(x+y)$
$\sqrt{log_d\Big(\frac{x}{y}\Big)^2 + \frac{1}{100d}} \leq 2\sqrt{\frac{d+1}{d}}log_{d+1}(x+y) - log_d(xy)$
$\sqrt{log_d\Big(\frac{x}{y}\Big)^2 + \frac{1}{100d}} \leq 2\Big(\frac{1}{log_d(d+1)}\sqrt{\frac{d+1}{d}} - 1\Big)log_d(x+y) + log_d(\frac{x}{y} + \frac{y}{x} + 2)$
$\Big(\frac{1}{log_d(d+1)}\sqrt{\frac{d+1}{d}} - 1\Big) \geq \frac{1}{4d}$ Spēka pie $d \geq 56$, pārbaudīts ar Wolfram|Alpha
$\sqrt{log_d\Big(\frac{x}{y}\Big)^2 + \frac{1}{100d}} \leq \frac{log_d(x+y)}{2d} + log_d\Big(\frac{x}{y} + 2\Big)$

- $\frac{x}{y} \leq \frac{\sqrt{d}}{ln(d)}$

$$\frac{log_d(x+y)}{2d} + log_d\Big(\frac{x}{y} + 2\Big) \geq log_d\Big(\frac{x}{y} + 2\Big) \geq log_d\Big(\frac{x}{y}\Big) + \frac{2}{(\frac{x}{y} + 2)ln(d)} \geq$$
$$\geq log_d\Big(\frac{x}{y}\Big) + \frac{2}{\sqrt{d}} \geq \sqrt{log_d\Big(\frac{x}{y}\Big)^2 + \frac{1}{100d}}$$

- $\frac{x}{y} > \frac{\sqrt{d}}{ln(d)} \Rightarrow \log_d(\frac{\sqrt{d}}{ln(d)}) > \frac{1}{8}$

$$\sqrt{log_d\Big(\frac{x}{y}\Big)^2 + \frac{1}{100d}} \leq log_d(\frac{x}{y}) + \frac{1}{200dlog_d(\frac{x}{y})} \leq log_d(\frac{x}{y}) + \frac{1}{25d} \leq$$
$$\leq \frac{log_d(x+y)}{2d} + log_d\Big(\frac{x}{y} + 2\Big)$$

□

3. pielikums. Novērtējuma $O(\sqrt{NK})$ pierādījums

Pamatosim nevienādību:

$$f_{ITE}(\sqrt{5(N-1)K}, \sqrt{5(N-1)(K-1)}) \leq \sqrt{5NK}$$

Ar matemātiskiem pārveidojumiem:

$$\frac{1}{2}(\sqrt{5(N-1)K}+\sqrt{5(N-1)(K-1)}+\sqrt{\left(\sqrt{5(N-1)K}-\sqrt{5(N-1)(K-1)}\right)^2+4}\leq$$
$$\leq\sqrt{5NK}$$
$$\sqrt{K}+\sqrt{K-1}+\sqrt{\left(\sqrt{K}-\sqrt{K-1}\right)^2+\frac{4}{5(N-1)}}\leq 2\sqrt{K\frac{N}{N-1}}$$
$$2K-1+\frac{2}{5(N-1)}+\sqrt{1+\frac{4(\sqrt{K}+\sqrt{K-1})^2}{5(N-1)}}\leq 2K+\frac{2K}{N-1}$$
$$\sqrt{1+\frac{4(\sqrt{K}+\sqrt{K-1})^2}{5(N-1)}}\leq 1+\frac{2(K-\frac{1}{5})}{N-1}$$
$$\frac{4(\sqrt{K}+\sqrt{K-1})^2}{5(N-1)}\leq\frac{4(K-\frac{1}{5})}{N-1}$$
$$(\sqrt{K}+\sqrt{K-1})^2\leq 5K-1$$

□

Bakalaura darbs „Kvantu algoritmu konstruēšana, izmantojot čaulu programmas" izstrādāts LU Datorikas fakultātē.

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

Autors:

*Nikita Larka*_____ ___.05.2017.

Rekomendēju darbu aizstāvēšanai

Vadītājs: profesors, Dr. dat.

*Andris Ambainis*_____ ___.05.2017.

Recenzents: docents, Dr. *Aleksandrs Belovs*

Darbs iesniegts Datorikas fakultātē ___.05.2017.

Dekāna pilnvarotā persona:

vecākā metodiķe *Ārija Sproģe*_____

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

_____

Komisijas sekretāre*:*_____