Posledica
definition
Primer Primer Primeri
remark Opomba
[qpl]ocamlescapeinside=||, autogobble
draft.bib

Kvantni algebrajski učinki

Strah, mentor: doc. dr. Matija Pretnar

1. marec 2022

Povzetek

Kvantno računalništvo temelji na veliko modernih konceptih v teoriji programskih jezikov, kot na primer linearnost tipov, (kvantnimi) fizikalnimi pojavi, in še mnogo drugimi. V diplomski nalogi se bomo posvetili tema dvema, v tem članku pa zgolj drugemu. Naš cilj je razumeti, kako se kvantni programi obnašajo, in dober način je razumevanje enakosti programov.

1 Kvantna mehanika

Ta del je povzet po [**?**]. Vsebuje osnovne definicije (in primere) matematičnih osnov kvantne mehanike, ki jih potrebujemo za definicije želenih operacij nad kubiti.

Oznake

Skozi ta del bomo uporabljali naslednje oznake:

- $ = \{0, \ldots\}$, $ _+ = \{1, \ldots\}$, $ = \{0, \ldots, 2 - 1\}$,
- $n, m \in \ _+$, ki mu bomo pravili število kubitov,
- $j, k, \ldots \in \ $,
- a j-ta komponenta vektorja a,
- $j = j \ldots j$ binarni zapis števila j.

1.1 Kvantni vektorji

Definicija 1 *Binarni vektorji so elementi prostora* $\mathbf{B}_n\{0, 1\}$ *in jih pišemo kot nize v binarnem zapisu. Za nas predstavljajo svet v katerem se odvijajo klasični programi.*

$\mathbf{B}_2 = \{00, 01, 10, 11\}$. 1 in 01 predstavljata različna vektorja.

Definicija 2 (Hilbertov prostor) *Elementom prostora* $\mathbf{H}_n C^2$ *pravimo kvantni vektorji, elementom* $\mathbf{HH}_1$ *pa kubiti. Prostoru* $\mathbf{H}_n$ *torej pravimo prostor kvantnih vektorjev reda* n*, njegovo standardno bazo pa označimo z* $\{e\}$*. Tu se izvajajo kvantni programi.*

Definicija 3 (Braket notacija) *Naj bo* $j \in$, *ter* $\widehat{j} \in \mathbf{B}_n$ *pripadajoč vektor v binarnem zapisu. Potem je* $j = \widehat{j}e$.

Po definiciji je torej $\mathbf{H}_n = \mathcal{L}_C(\{j \mid j \in \mathbf{B}_n\})$.
$[n = 1 \text{ in } n = 2]$ a = a
a = a 1
0 + a 0
1 = a 0 + a 1,
a = a
a
a
a = a
a
a
a = a 00 + a 01 + a 10 + a 11.
[Hadamardov vektor]

$$\mathbf{h}\,(0+1)\,, \quad \mathbf{h}_n \sum_{j \in \mathbf{B}_n} j, \quad \frac{1}{\sqrt{2}}.$$

1.2 Blochova sfera

[baseline=(current

Trditev 1 *V fizičnem svetu sta dva kubita, ki se razlikujeta zgolj za (kompleksen) faktor, enaka. Matematično to pomeni, da stanja kubitov (nadaljnje tudi kubiti) živijo v* $\mathbf{P}C \cong S$*:*

$$a = \cos\tfrac{}{2}0 + e^{i}\sin\tfrac{}{2}1, \quad \in [0,2),\ \in [0,].$$

Naj bo $a = a0 + a1 = re^{i}0 + re^{i}1$. Označimo $r\sqrt{r+r}, -, 2\arccos\frac{r}{r}$. Potem je $a = \widehat{a}\frac{a}{re^{i}} = \frac{r}{r}0 + \frac{r}{r}e^{i}1 = \cos\frac{}{2}0 + e^{i}\sin\frac{}{2}1$.
bounding box.north)]
(0,0) node[circle,fill,inner sep=1] (orig) – (7/2/6,7/2/4) node[circle,fill,inner sep=0.7,label=above:a] (a) ; [dashed] (orig) – (7/2/6,-7/2/10) node (phi) – (a);
(orig) circle (7/2/2); [dashed, gray] (orig) ellipse (7/2/2 and 7/2/6);
[->] (orig) – ++(-7/2/10,-7/2/6) node[below] (x) x; [->] (orig) – ++(7/2/2,0) node[right] (y) y; [->] (orig) – ++(0,7/2/2) node[above] (z) $z = 0$; [->, draw=gray] (orig) – ++(0,-7/2/2) node[below] (s) 1;
[draw=gray,->,"",angle eccentricity=0.6] angle=x–orig–phi; [draw=gray,<-,"",angle eccentricity=1.4] angle=a–orig–z;

1.3 Tenzorski produkt

Definicija 4 (Tenzorski produkt) *Tenzorski produkt prostorov* $\mathbf{H}_n$ *in* $\mathbf{H}_m$ *je enak* $\mathbf{H}_{n+m}$. *Pišemo* $\mathbf{H}_n\mathbf{H}_m$. *Če sta* $a \in \mathbf{H}_n$ *in* $b \in \mathbf{H}_m$ *je* $ab \in \mathbf{H}_n\mathbf{H}_m$.

Operator je res tenzorski produkt. [Tenzorski produkt baznih vektorjev]

$$jk = j \ldots jk \ldots k = jk = jk,$$

[Splošni tenzorski produkt]

$$aabb = abababab, \quad ab = \sum_{j \in \mathbf{B}_n, k \in \mathbf{B}_m} abjk.$$

[Tenzorski eksponent] $\mathrm{h}_n = \mathbf{h}^n = \underbrace{(0+1)\ldots(0+1)}_{n}$,

$\mathbf{H}_n = \mathbf{H}^n = \underbrace{\mathbf{H}\ldots\mathbf{H}}_{n}$.

Definicija 5 *Če lahko $a \in \mathbf{H}_n$ zapišemo kot $\bigotimes_{j=1}^{n} a$ za neke $a \in \mathbf{H}$ pravimo, da je enostaven ali separabilen, sicer je pa sestavljen oziroma kvantno prepleten.*

1.4 Kvantne preslikave

Definicija 6 *Prostor unitarnih vrat reda n je $\mathbf{U}_n\mathbf{U}(2)$, prostor unitarnih 2×2 matrik. Tenzorski produkt vrat $UV[u_{jk}V]_{j,k}$ uporabljen na ab je enak $UaVb$.*

[Tenzorski produkt unitarnih vrat]

$$aaaaB = aBaBaBaB.$$

Definicija 7 *Za vrata $U, \ldots, U$ označimo njihovo bločno-diagnoalno matriko z $D(U, \ldots, U)$.*

Izrek 1 (No cloning) *Ne obstajajo vrata reda 2, ki vsak vektor $a0 \in \mathbf{HH}$ slika v aa.*

Naj bo U tak, da za vsak $a \in \mathbf{H}$ velja $U(a0) = aa$.
Potem za $\mathbf{h}0 = (00 + 10)$ velja:

$$U((00+10)) = \{\ (00 + 01 + 10 + 11), U00 + U10 = (00 + 11),$$

kar je protislovje.

[Paulijeve matrike] To so matrike zrcaljenja okrog osi na Blochovi sferi:

$$I = 1001, \quad X = 0110, \quad Y = 0 - ii0, \quad Z = 100 - 1.$$

Velja $X = Y = Z = I$. Preslikavi X pravimo negacija, saj je $X0 = 1$ in $X1 = 0$.

[Hadamardova matrika]

$$\mathtt{Had} = 111 - 1, \quad \mathtt{Had}0 = \mathbf{h}, \quad \mathtt{Had}^n\mathbf{0} = \mathbf{h}_n.$$

[Fazni zamik] $\mathrm{S} = 10$
$0e^i, posebejoznaimoSS_{/_2}, TS_{/_4}$,
$S\,(a0 + a1) = a0 + ae^i 1$.

1.5 Kvantna meritev

V klasičnem računalništvu poznamo pogojne stavke. To lahko na kubite posplošimo na dva načina, prvi z direktno meritvijo kubita (in uporabo klasičnih pogojnih stavkov), drugi pa z uporabo kvantne prepletenosti. Izkaže se, da če na koncu zmerimo kubite, se drugi način obnaša enako kot prvi.

Definicija 8 (Kvantna meritev) *Meritev kubita* $a = a_0 0 + a_1 1$ *označimo* $M(a)$ *in je* 0 *z verjetnostjo* $|a_0|^2$ *in* 1 *z verjetnostjo* $|a_1|^2$. *To „uniči“ kubit* a.

Definicija 9 (Kontrola) *Za* $r, s \in N$ *in* $U \in \mathbf{U}_1$ *definiramo* $C_{r,s}(U)$ *in* $\overline{C}_{r,s}(U)$ *s predpisoma*

$$C_{r,s}(U)j = \{\ j; j = 0j \ldots Uj \ldots j; j = 1 \qquad \overline{C}_{r,s}(U)j = \{\ j; j = 1j \ldots Uj \ldots j; j = 0$$

Takim vratom pravimo kontrolirana („na ena“ in „na nič“). Posebej za $U \in \mathbf{U}_1$ *označimo*

$$\texttt{cU}C_{1,2}(U) = D(\texttt{I}, \texttt{U}), \quad \bar{\texttt{c}}\texttt{U}\overline{C}_{1,2}(U) = D(\texttt{U}, \texttt{I}).$$

[Prepleteni pari kubitov] Kontrolirana vrata prepletejo pare kubitov. Na primer
$apply_{\texttt{cX}}(a, b)$ se obnaša kot if $|measure(a) = 0|$ then $|(a, b)|$ else $|(a, \neg b)|$. Seveda vemo, da drugi izraz ni veljaven (ker meritev uniči kubit a), ampak je zato kontrola ravno tisto orodje, s katerim želimo nadomestiti pogojne stavke.

2 Kvantno računalništvo ter algebrajski učinki in diagrami

2.1 Kvantna vezja

Kvantne programe lahko predstavimo kot diagrame vezja. Škatle predstavljajo unitarna vrata, črte med njimi pa žice; po enojnih žicah tečejo kubiti, po dvojnih pa klasični biti (0 ali 1). Pike na žici (in potem navpična žica ven) pomenijo kontrolo; prazna pika kontrolira „na nič“, polna pa „na ena“. Taka vezja beremo od leve proti desni. Natančnejši opis lahko najdete v [**?**].

Spodaj sta dva primera kvantnih programov, opisana z besedami in diagrami, ki ju bomo srečali tudi še kasneje.

[Projekcija na z-os] Najprej zmerimo a in nato glede na rezultat svež kubit bodisi negiramo bodisi ne. Na Blochovi sferi to zgleda približno kot projekcija na z-os (edina kubita na z-osi sta 0 in $1 = X0$).

$$@C = 1em@R = .7em0\texttt{X}ba{-}1$$

[Naključna rotacija faze] Meritev Hadamardovega vektorja simulira pravičen met kovanca, vrata Z pa rotirajo fazo, torej bomo v polovici primerov kubitu a rotirali fazo.

$$@C = 1em@R = .7ema\texttt{Z}a0\texttt{Had}{-}1$$

2.2 Algebrajski učinki

Z računskimi učinki se med programiranjem pogosto srečamo: globalno stanje spremenljivk, vhodno/izhodne naprave, naključnost, izjeme, nedeterminizem, ipd.

Definicija 10 (Računski učinki) *Če ima funkcija ali operacija še kak navzven viden učinek poleg vrnjene vrednosti, slednjemu pravimo računski učinek (učinek računanja).*

Definicija 11 (Algebrajski učinki) *Računskim učinkom, ki jih lahko predstavimo s kašno algebrajsko teorijo, pravimo algebrajski učinki.*

3 Programski jezik

V našem jeziku[**?**] imamo navadne osnovne konstrukte, npr. tipe, `let` ter `if` stavke, itd. Poleg tega imamo pa še elemente kvantnega računalništva: tip kubitov `qubit` in tip prepletenih parov AB za vsaka dva tipa A in B. Zaradi narave kubitov ne moremo neposredno dostopati do notranjega stanja pomnilnika, imamo pa naslednje funkcije dostopanja:

- *new*: dodeli nov kubit, z začetno vrednostjo 0,
- $apply_{\mathtt{U}}$: uporabi vrata U na danem vektorju,
- *measure*: izvede meritev na kubitu, vrne element tipa `bit`.

3.1 Pretvorba v algebrajske izraze

Konstruktom v programskem jeziku priredimo naslednje algebrajske izraze ter uvedemo še strnjeno obliko, za lažjo manipulacijo na papirju.

Kvantni programski jezik	Algebrajski izrazi	Matematični simboli
let $\lvert a \leftarrow new() \rvert$ in $\lvert x(a) \rvert$	$new(a.\, x(a))$	$\nu a.\, x(a)$
$\lvert apply_{\mathtt{U}}(a) \rvert; \lvert x(a) \rvert$	$apply_{\mathtt{U}}(a.\, x(a))$	$\mathtt{U}_a(x(a))$
if $\lvert measure(a) = 0 \rvert$ then $\lvert t \rvert$ else $\lvert u \rvert$	$measure(a.\, t; u)$	$t \mathbin{?_a} u$
$\lvert discard(a) \rvert; \lvert t \rvert$	$discard(a.\, t)$	$disc_a(t)$

[Projekcija na z-os]

1. if $\lvert measure(a) = 0 \rvert$ then $\lvert new() \rvert$ else $\lvert apply_{\mathtt{X}}(new()) \rvert$
2. $measure(a.\, new(b.\, x(b)); new(b.\, apply_{\mathtt{X}}(b.\, x(b))))$
3. $a \nu b.\, x(b) \nu b.\, \mathtt{X}_b(x(b))$

[Naključna rotacija faze]

1. if $\lvert measure(apply_{\mathtt{Had}}(new())) = 0 \rvert$ then $\lvert a \rvert$ else $\lvert apply_{\mathtt{Z}}(a) \rvert$
2. $new(b.\, apply_{\mathtt{Had}}(b.\, measure(b.\, x(a); apply_{\mathtt{Z}}(a.\, x(a)))))$
3. $\nu b.\, \mathtt{Had}_b(x(a) \mathbin{?_b} \mathtt{Z}_a(x(a)))$

3.2 Aksiomi

Aksiome za enakost programov lahko delimo na dva dela; prvih pet je glavnih, ostalih sedem pa bolj „administrativnih“ oziroma pomožnih. Slednji nam povejo zgolj, da se *apply* strinja s strukturo unitarnih matrik, ter da stvari komutirajo, kolikor vezanje spremenljivk (in vrstni red uporabe matrik) dopušča. Podrobnejši opis (z dokazom) najdete v [**?**].
Kvantna negacija pred meritvijo je negacija po meritvi:

Kvantna negacija pred meritvijo je negacija po meritvi:
Aksiom A. $\mathtt{X}_a(x \mathbin{?_a} y) = y \mathbin{?_a} x$.
Kvantna kontrola je po meritvi kot klasična kontrola:

Kvantna kontrola je po meritvi kot klasična kontrola:

Aksiom B. $D(\mathtt{U},\mathtt{V})_{a,b}(x(b)\ ?_a\ y(b)) = \mathtt{U}_b(x(b))\ ?_a\ \mathtt{V}_b(y(b))$.
Kvantna vrata uporabljena na zavrženih kubitih so odveč:

Kvantna vrata uporabljena na zavrženih kubitih so odveč:
Aksiom C. $\mathtt{U}_a(disc_a(t)) = disc_a(t)$.
Novi kubiti so 0 glede na meritev:

Novi kubiti so 0 glede na meritev:
Aksiom D. $\nu a.\, x\ ?_a\ y = x$.
Novi kubiti so 0 glede na kontrolo:

Novi kubiti so 0 glede na kontrolo:
Aksiom E. $\nu a.\, D(\mathtt{U},\mathtt{V})_{a,b}(x(a,b)) = \mathtt{U}_b(\nu a.\, x(a,b))$.
Spoštovanje simetrične grupe $\mathbf{U}_n$:

Spoštovanje simetrične grupe $\mathbf{U}_n$:
Aksiom F. $\mathtt{swap}_{a,b}(x(a,b)) = x(b,a)$,

Aksiom G. $\mathtt{I}_a(x(a)) = x(a)$,

Aksiom H. $\mathtt{UV}_a(x(a)) = \mathtt{V}_a(\mathtt{U}_a(x(a)))$,

Aksiom I. $\mathtt{UV}_{a,b}(x(a,b)) = \mathtt{U}_a(\mathtt{V}_b(x(a,b)))$.
Komutativnost:

Komutativnost:
Aksiom J. $au\ ?_b\ vx\ ?_b\ y = bu\ ?_a\ xv\ ?_a\ y$,

Aksiom K. $\nu a.\, \nu b.\, x(a,b) = \nu b.\, \nu a.\, x(a,b)$,

Aksiom L. $\nu a.\, x(a)\ ?_b\ y(a) = b\nu a.\, x(a)\nu a.\, y(a)$.

[Izpeljava enakosti projekcije na z-os in naključne rotacije faze] Izpeljava se zanaša na identiteti $\mathtt{cX.swap.cX}\dagger{=}\mathtt{swap.cX.swap}$ in $\mathtt{swap.cX.swap}\ddagger{=}(\mathtt{HadI}).\mathtt{cZ}.(\mathtt{HadI})$.
$\mathrm{a}\nu\mathrm{b}.\ \mathrm{x}(b)\ \nu b.\, \mathtt{X}_b(x(b))$
$= \nu b.\, x(b)\ ?_a\ \mathtt{X}_b(x(b))(\mathbf{??})$
$= \nu b.\, \mathtt{cX}_{a,b}(x(b)\ ?_a\ x(b))(\mathbf{??})$
$= \nu b.\, \mathtt{cX}_{a,b}(disc_a(x(b)))(def.)$
$= \nu b.\, \mathtt{cX}_{a,b}(\mathtt{cX}_{a,b}(disc_b(x(a))))(\dagger)$
$= \nu b.\, \mathtt{cX}_{a,b}(disc_b(x(a)))(\mathbf{??})$
$= \nu b.\, \mathtt{Had}_b(\mathtt{cZ}_{b,a}(\mathtt{Had}_b(disc_b(x(a)))))(\ddagger)$
$= \nu b.\, \mathtt{Had}_b(\mathtt{cZ}_{b,a}(disc_b(x(a))))(\mathbf{??})$
$= \nu b.\, \mathtt{Had}_b(x(a)\ ?_b\ \mathtt{Z}_a(x(a))).(\mathbf{??})$