Kvantni algebrajski učinki

Strah, mentor: doc. dr. Matija Pretnar

28. februar 2022

Povzetek

Kvantni programski jeziki predstavljajo nove probleme za teorijo programskih jezikov, kot so linearnost tipov, (kvantni) fizikalni pojavi, in še mnogi drugi. V tej nalogi se bomo posvetili tema dvema. Naš cilj je razumeti, kako se kvantne programe, in dober način je razumevanje enakosti programov, tj. kdaj sta dva programa enaka?

Najprej bomo predstavili algebrajsko teorijo za kvantne programe; ta je zgrajena na unitarnih vratih in meritvah ter ima linearne parametre. Nato bomo dokazali, da lahko s to teorijo predstavimo vse programe (polnost) in nato iz nje izpeljali pravila za enakost kvantnih programov.

1 Kvantna mehanika

Ta del je povzet po [**ess-qc**]. Vsebuje osnovne definicije (in primere) matematičnih osnov kvantne mehanike, ki jih potrebujemo, da definiramo želene operacije nad kubiti. Te se nahajajo na prvih nekaj straneh,

Oznake

Skozi ta del bomo uporabljali naslednje oznake:

- $\mathbb{N} = \{0, \dots\}, \mathbb{N}_+ = \{1, \dots\}$,
- $n, m \in \mathbb{N}_+$, ki mu bomo pravili število kubitov,
- $j, k, \dots \in \{0, \dots, 2^n - 1\}$,
- a_j j-ta komponenta vektorja a,
- $j = j_1 \dots j_n$ binarni zapis števila j.

1.1 Kvantni vektorji

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

Primer. $\mathbf{B}_2 = \{00, 01, 10, 11\}$.

Opomba. 1 in 01 predstavljata različna vektorja.

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

Definicija 3 (Braket notacija)**.** Naj bo $j \in \{0, \dots, 2^n - 1\}$, ter $\hat{\jmath} \in \mathbf{B}_n$ pripadajoč vektor v binarnem zapisu. Potem je $|j\rangle = |\hat{\jmath}\rangle := e_j$.

Opomba. Po definiciji je torej $\mathbf{H}_n = \mathcal{L}_{\mathbb{C}}(\{|j\rangle \mid j \in \mathbf{B}_n\})$.

Primer ($n = 1$ in $n = 2$)**.**

$$a = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = a_0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + a_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = a_0 \left|0\right\rangle + a_1 \left|1\right\rangle,$$

$$a = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} = a_{00} \left|00\right\rangle + a_{01} \left|01\right\rangle + a_{10} \left|10\right\rangle + a_{11} \left|11\right\rangle.$$

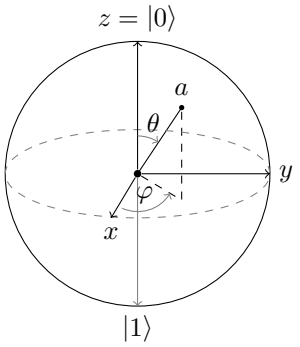
Primer (Hadamardov vektor)**.**

$$\mathbf{h} \coloneqq \rho \left(\left|0\right\rangle + \left|1\right\rangle\right), \quad \mathbf{h}_n \coloneqq \rho^n \sum_{j \in \mathbf{B}_n} \left|j\right\rangle, \quad \rho \coloneqq \frac{1}{\sqrt{2}}.$$

1.2 Blochova sfera

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* $\mathbf{P}\mathbb{C}^2 \cong \mathbb{S}^2$*:*

$$a = \cos\frac{\theta}{2} \left|0\right\rangle + e^{i\varphi} \sin\frac{\theta}{2} \left|1\right\rangle, \quad \varphi \in [0, 2\pi), \theta \in [0, \pi].$$



Dokaz. Naj bo $a = a_0 \left|0\right\rangle + a_1 \left|1\right\rangle = r_0 e^{i\varphi_0} \left|0\right\rangle + r_1 e^{i\varphi_1} \left|1\right\rangle$. Označimo $r \coloneqq \sqrt{|a_0|^2 + |a_1|^2}, \varphi \coloneqq \varphi_1 - \varphi_0, \theta \coloneqq 2 \arccos\frac{a_0}{r}$.
Potem je $a = \hat{a} \coloneqq \frac{a}{re^{i\varphi_0}} = \frac{a_0}{r} \left|0\right\rangle + \frac{a_1}{r} e^{i\varphi} \left|1\right\rangle$. □

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 $\mathbf{H}_n \otimes \mathbf{H}_m$. Če sta $a \in \mathbf{H}_n$ in $b \in \mathbf{H}_m$ je $a \otimes b \in \mathbf{H}_n \otimes \mathbf{H}_m$.

Opomba. Operator $\otimes$ je res tenzorski produkt.

Primer (Tenzorski produkt baznih vektorjev)**.**

$$\left|j\right\rangle \otimes \left|k\right\rangle = \left|j_1 \ldots j_n k_1 \ldots k_m\right\rangle = \left|j\right\rangle \left|k\right\rangle = \left|jk\right\rangle,$$

Primer (Splošni tenzorski produkt)**.**

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \otimes \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{bmatrix}, \quad a \otimes b = \sum_{\substack{j \in \mathbf{B}_n, \\ k \in \mathbf{B}_m}} a_j b_k \left|jk\right\rangle.$$

Primeri (Tenzorski eksponent)**.**

$$\mathbf{h}_n = \mathbf{h}^{\otimes n} = \rho^n \underbrace{\left(\left|0\right\rangle + \left|1\right\rangle\right) \otimes \cdots \otimes \left(\left|0\right\rangle + \left|1\right\rangle\right)}_{n},$$

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

Definicija 5. Če lahko $a \in \mathbf{H}_n$ zapišemo kot $\bigotimes_{j=1}^{n} a_j$ za neke $a_j \in \mathbf{H}$ 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 $\mathbf{U}_n \coloneqq \mathbf{U}(2^n)$, prostor unitarnih $2^n \times 2^n$ matrik. Tenzorski produkt vrat $U \otimes V \coloneqq [u_{jk} V]_{j,k}$ uporabljen na $a \otimes b$ je enak $Ua \otimes Vb$.

Primer (Tenzorski produkt unitarnih vrat)**.**

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \otimes B = \begin{bmatrix} a_{00}B & a_{01}B \\ a_{10}B & a_{11}B \end{bmatrix}.$$

Definicija 7. Za vrata $U_0, \dots, U_s$ označimo njihovo bločno-diagnoalno matriko z $D(U_0, \dots, U_s)$.

Izrek 1 (No cloning)**.** *Ne obstajajo vrata reda* 2*, ki vsak vektor* $a \otimes |0\rangle \in \mathbf{H} \otimes \mathbf{H}$ *slika v* $a \otimes a$*.*

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

$$U(\rho(|00\rangle + |10\rangle)) = \begin{cases} \rho^2(|00\rangle + |01\rangle + |10\rangle + |11\rangle), \\ \rho U|00\rangle + U|10\rangle = \rho(|00\rangle + |11\rangle), \end{cases}$$

kar je protislovje. □

Primer (Paulijeve matrike)**.** To so matrike zrcaljenja okrog osi na Blochovi sferi:

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Velja $X^2 = Y^2 = Z^2 = I_2$. Preslikavi X pravimo negacija, saj je $X|0\rangle = |1\rangle$ in $X|1\rangle = |0\rangle$.

Primer (Hadamardova matrika)**.**

$$\mathtt{Had} = \rho \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \mathtt{Had}\,|0\rangle = \mathbf{h}, \quad \mathtt{Had}^{\otimes n}\,|\mathbf{0}^n\rangle = \mathbf{h}_n.$$

Primer (Fazni zamik)**.**

$$S_\alpha = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{bmatrix}, \text{ posebej označimo } S := S_{\pi/2}, T := S_{\pi/4},$$
$$S_\alpha\,(a_0\,|0\rangle + a_1\,|1\rangle) = a_0\,|0\rangle + a_1 e^{i\alpha}\,|1\rangle\,.$$

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 $a = a_0\,|0\rangle + a_1\,|1\rangle$ 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 $r, s \in \mathbb{N}$ in $U \in \mathbf{U}_1$ definiramo $C_{r,s}(U)$ in $\overline{C}_{r,s}(U)$ s predpisoma

$$C_{r,s}(U)\,|j\rangle = \begin{cases} |j\rangle & ; j_r = 0 \\ |j_1 \dots\rangle\,|Uj_s\rangle\,|\dots j_n\rangle & ; j_r = 1 \end{cases} \qquad \overline{C}_{r,s}(U)\,|j\rangle = \begin{cases} |j\rangle & ; j_r = 1 \\ |j_1 \dots\rangle\,|Uj_s\rangle\,|\dots j_n\rangle & ; j_r = 0 \end{cases}$$

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

$$\mathtt{cU} := C_{1,2}(U) = D(\mathtt{I_2}, \mathtt{U}), \quad \mathtt{\bar{c}U} := \overline{C}_{1,2}(U) = D(\mathtt{U}, \mathtt{I_2}).$$

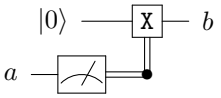
2 Kvantno računalništvo ter algebrajski učinki in diagrami

2.1 Kvantna vezja

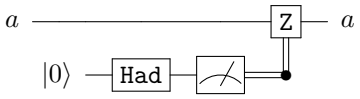
Kvantne programe lahko predstavimo kot diagrame vezja. Škatle predstavljajo unitarna vrata, črte med njimi pa žice; po enojnih žicah tečejo kubiti, po dvojnih pa klasični biti (0 ali 1). Pike na žici (in potem navpična žica ven) pomenijo kontrolo; polna pika kontrolira "'na nič"', prazna pa "'na ena"'.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

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

Primer (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 $|0\rangle$ in $|1\rangle = X|0\rangle$).



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



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[**algeff-lin-qpl**] 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 $A \otimes B$ za vsaka dva tipa A in B. Zaradi narave kubitov ne moremo neposredno dostopati do notranjega stanja pomnilnika, imamo pa naslednje funkcije dostopanja:

- $\underline{\mathsf{new}}$: dodeli nov kubit, z začetno vrednostjo $|0\rangle$,
- $\underline{\mathsf{apply}}_{\mathtt{U}}$: uporabi vrata U na danem vektorju,
- $\underline{\mathsf{measure}}$: 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
$\mathtt{let}\ a \leftarrow \underline{\mathsf{new}}()\ \mathtt{in}\ x(a)$	$\mathsf{new}(a.\,x(a))$	$\nu a.\,x(a)$
$\underline{\mathsf{apply}}_{\mathtt{U}}(a);\ x(a)$	$\mathsf{apply}_{\mathtt{U}}(a.\,x(a))$	$\mathtt{U}_a(x(a))$
$\mathtt{if}\ \underline{\mathsf{measure}}(a) = 0\ \mathtt{then}\ t\ \mathtt{else}\ u$	$\mathsf{measure}(a.\,t; u)$	$t\ ?_a\ u$
$\underline{\mathsf{discard}}(a);\ t$	$\mathsf{discard}(a.\,t)$	$\mathsf{disc}_a(t)$

Primer (Projekcija na z-os)**.**

1. **if** $\underline{\mathsf{measure}}(a) = 0$ **then** $\underline{\mathsf{new}}()$ **else** $\underline{\mathsf{apply}}_{\mathtt{X}}(\underline{\mathsf{new}}())$
2. $\mathsf{measure}\Big(a.\,\mathsf{new}(b.\,x(b)); \mathsf{new}\Big(b.\,\mathsf{apply}_{\mathtt{X}}(b.\,x(b))\Big)\Big)$
3. $(\nu b.\,x(b))\ ?_a\ (\nu b.\,\mathtt{X}_b(x(b)))$

Primer (Naključna rotacija faze)**.**

1. **if** $\underline{\mathsf{measure}}\Big(\underline{\mathsf{apply}}_{\mathtt{Had}}(\underline{\mathsf{new}}())\Big) = 0$ **then** a **else** $\underline{\mathsf{apply}}_{\mathtt{Z}}(a)$
2. $\mathsf{new}\Big(b.\,\mathsf{apply}_{\mathtt{Had}}\Big(b.\,\mathsf{measure}\Big(b.\,x(a); \mathsf{apply}_{\mathtt{Z}}(a.\,x(a))\Big)\Big)\Big)$
3. $\nu b.\,\mathtt{Had}_b(x(a)\ ?_b\ \mathtt{Z}_a(x(a)))$

3.2 Aksiomi

Osnovni aksiomi na kratko:

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

Kvantna kontrola je po meritvi kot klasična kontrola:
Aksiom B. $\mathtt{U}_b(x(b))\ ?_a\ \mathtt{V}_b(y(b)) = D(\mathtt{U},\mathtt{V})_{a,b}(x(b)\ ?_a\ y(b))$.

Kvantna vrata uporabljena na zavrženih kubitih so odveč:
Aksiom C. $\mathtt{U}_a(\mathsf{disc}_a(t)) = \mathsf{disc}_a(t)$.

Meritve novih kubitov so vedno 0:
Aksiom D. $\nu a.\,x\ ?_a\ y = x$.

Vrata kontrolirana z novimi kubiti se nikoli ne uporabijo:
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 $\mathbf{U}_n$:
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. $\mathtt{U} \otimes \mathtt{V}_{a,b}(x(a,b)) = \mathtt{U}_a(\mathtt{V}_b(x(a,b)))$.

Komutativnost:
Aksiom J. $(u\ ?_b\ v)\ ?_a\ (x\ ?_b\ y) = (u\ ?_a\ x)\ ?_b\ (v\ ?_a\ y)$,
Aksiom K. $\nu a.\,\nu b.\,x(a,b) = \nu b.\,\nu a.\,x(a,b)$,
Aksiom L. $\nu a.\,x(a)\ ?_b\ y(a) = (\nu a.\,x(a))\ ?_b\ (\nu a.\,y(a))$.

Primer (Izpeljava enakosti projekcije na z-os in naključno rotacijo faze)**.** Dokaz uporabi identiteti $\mathtt{cX.swap.cX} \overset{\dagger}{=} \mathtt{swap.cX.swap}$ in $\mathtt{swap.cX.swap} \overset{\ddagger}{=} (\mathtt{Had} \otimes I).\mathtt{cZ}.(\mathtt{Had} \otimes I)$

$$
\begin{aligned}
&(\nu b.\,x(b))\ ?_a\ (\nu b.\,\mathtt{X}_b(x(b))) \\
&\quad = \nu b.\,x(b)\ ?_a\ \mathtt{X}_b(x(b)) && (??)\\
&\quad = \nu b.\,\mathtt{cX}_{a,b}(x(b)\ ?_a\ x(b)) && (??)\\
&\quad = \nu b.\,\mathtt{cX}_{a,b}(\mathsf{disc}_a(x(b))) && (\text{def.})\\
&\quad = \nu b.\,\mathtt{cX}_{b,a}\big(\mathtt{cX}_{a,b}(\mathsf{disc}_b(x(a)))\big) && (\dagger)\\
&\quad = \nu b.\,\mathtt{cX}_{a,b}(\mathsf{disc}_b(x(a))) && (??)\\
&\quad = \nu b.\,\mathtt{Had}_b\big(\mathtt{cZ}_{b,a}(\mathtt{Had}_b(\mathsf{disc}_b(x(a))))\big) && (\ddagger)\\
&\quad = \nu b.\,\mathtt{Had}_b\big(\mathtt{cZ}_{b,a}(\mathsf{disc}_b(x(a)))\big) && (??)\\
&\quad = \nu b.\,\mathtt{Had}_b(x(a)\ ?_b\ \mathtt{Z}_a(x(a))). && (??)
\end{aligned}
$$

Kot vidimo, smo definirali jezik (in orodja) s katerimi lahko relativno enostavno dokazujemo enakost med programi.

,