Reverzibilní výpočty

Každý klasický algoritmus je booleovskou funkcí

$$\{0,1\}^n \to \{0,1\},$$

která vstupní posloupnosti přiřadí výstupní bit. Je-li výstup delší, což je obvyklé, je algoritmus souborem takových funkcí.

Složitost algoritmu, díváme-li se na něj jako na booleovskou funkci, odpovídá velikosti nejmenšího booleovského obvodu, který danou funkci počítá. Obvod se může skládat z několika jednoduchých, obvykle jedno- nebo dvouhodnotových hradel. Konstrukce obvodu je tedy jakýmsi *rozkladem* booleovské funkce do nějaké vhodné množiny jednoduchých funkcí.

Na podobném základě je možné mluvit i o složitosti kvantových algoritmů: příslušnou unitární transformaci rozložíme na jednoduché transformace a ptáme se po velikosti rozkladu, tj. počtu použitých hradel.

K tomu je třeba zvolit nějakou vhodnou množinu základních operátorů, které můžeme v rozkladu používat a pomocí kterých lze zkonstruovat libovolný operátor. Takové množině říkáme *univerzální množina hradel*.

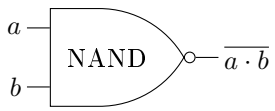
V klasickém případě tvoří přirozenou univerální množinou hradel funkce AND, OR a NOT, což jsou booleovské operátory $\wedge$, $\vee$ a $\neg$. Libovolnou funkci lze do ní přímočaře rozložit např. díky disjunktivní normální formě funkce f:

$$f(x_1, x_2, \ldots, x_n) = \bigvee_{f(\mathbf{z})=1} (y_1 \wedge y_2 \wedge \cdots \wedge y_n),$$

kde $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ probíhá $\{0,1\}^n$ a

$$y_i = \begin{cases} x_i, & \text{pokud } z_i = 1, \\ \overline{x_i}, & \text{pokud } z_i = 0. \end{cases}$$

Existuje ovšem také několik hradel, která jsou sama univerzální, např. NAND:



Univerzalita NAND plyne ze vztahů $\overline{a} = \mathrm{NAND}(a, a)$ a $a \vee b = \mathrm{NAND}(\overline{a}, \overline{b})$.

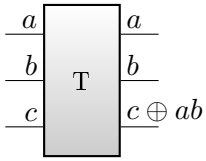
Pokud od kvantových obvodů očekáváme lepší výsledky než od klasických, měli bychom být přinejmenším schopni realizovat kvantově klasické algoritmy. K tomu by stačilo realizovat kvantový obvod NAND. To ale není bez dalšího možné, protože všechny kvantové operátory jsou reversibilní, zatímco NAND nikoli.

Tím se dotýkáme problematiky reverzibilního výpočtu booleovské funkce. Univerzální reverzibilní funkce musí být alespoň tříbitová, a je jí např. Toffoliho funkce definovaná jako

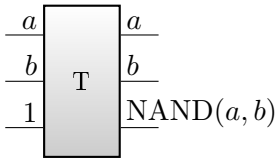
$$T : (a, b, c) \mapsto (a, b, c \oplus ab),$$

kde binární násobení odpovídá AND a binární sčítání $\oplus$ odpovídá logickému hradlu XOR, tedy "vylučovacímu nebo". Je to funkce reversibilní, je sama sobě inverzem.

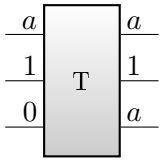
Příslušné hradlo budeme zobrazovat jako



a operátor NAND dostaneme pomocí Toffoliho hradla takto:



Pro kvantové výpočty je důležité, že můžeme pořizovat kopie vstupních bázových stavů:

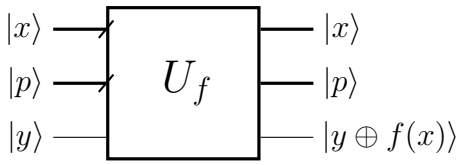


Poznamenejme, že takto kopírujeme pouze bázové stavy, takže věta o neklonování není porušena. Pro obecný stav $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$ dostáváme

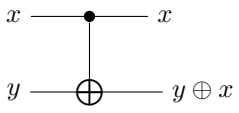
$$T|\varphi\rangle|1\rangle|0\rangle = \alpha \cdot T|010\rangle + \beta \cdot T|110\rangle = \alpha|010\rangle + \beta|111\rangle,$$

což je propletený stav, který se určitě nerovná $|\varphi\rangle|1\rangle|\varphi\rangle$.

Z obrázků je vidět, že výpočty složené z Toffoliho hradel budou potřebovat kromě vstupu ještě pomocné bity. Také jsme viděli, že pomocné bity se se vstupními proplétají, což je nežádoucí, protože je musíme brát v úvahu např. při odhadu výsledku měření. Kvantový výpočet booleovské funkce f by měl odpovídat podobě, kterou jsme viděli v Deutschově-Jozsově algoritmu. Měl by se skládat ze vstupního registru $|x\rangle$, který se výpočtem nezmění, z pomocného registru $|p\rangle$, který se rovněž nezmění, a z výstupního kubitu $|y\rangle$, do něhož se přičte výstup hodnota $f(x)$. Schematicky



přičemž pomocný registr přirozeně ze zápisu vynecháváme. Přičtení výsledku operace budeme realizovat následujícím obvodem



který se nazývá CNOT, neboli *kontrolovaná negace*, protože se vlastně jedná o instrukci: pokud x, neguj y.

Žádoucí podoby výpočtu je nyní možné dosáhnout následujícími kroky:

- reverzibilní kvantový výpočet funkce f (např. pomocí Toffoliho hradel) ve vstupním a pomocném registru;
- přičtení výsledku do výstupního kubitu pomocí funkce CNOT;
- inverzní výpočet ve vstupním a pomocném registru vedoucí k původním stavům.

Postup ilustruje následující schéma na výpočtu funkce $a \vee b = \overline{(\bar{a} \wedge \bar{b})}$.