Algorytmy Kwantowe

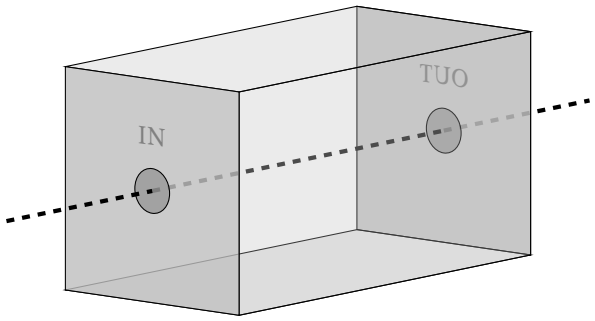
Lista 1

1. Superpozycje

Zadanie 1

Trudność: łatwe Punktów: 1

Na lotnisku znaleziono prostopadłościenne pudło z dwoma oszklonymi otworami na przeciwległych ściankach. Nad jednym otworem jest napisane IN, a nad przeciwległym OUT. Ściany pudła są nieprzezroczyste.



Pudło może być jednym z dwojga:

(ATRAPA) Pudło jest puste, a okienka są ze zwykłego szkła.

(BOMBA) Za szybką IN umieszczono *dzielnik wiązki polaryzacyjnej* (polarizing beam splitter)[1]. Wiązka spolaryzowana poziomo przechodzi do OUT. Wiązka pionowa aktywuje detonację bomby.

Chcemy ustalić, czy pudło jest atrapą czy bombą, nie wysadzając jednocześnie lotniska w powietrze. Wskaż procedurę, która używa jednego polaryzatora liniowego, dla atrapy zawsze powie ATRAPA, a dla bomby powie BOMBA z prawdopodobieństwem $\frac{3}{4}$, ale spowoduje wybuch tylko z prawdopodobieństwem $\frac{1}{2}$[2].

Zadanie 2

Trudność: średnie Punktów: 3

Dla dowolnie małego $\varepsilon > 0$ wskaż procedurę, która odróżnia bombę od atrapy bezbłędnie i doprowadza do wybuchu z prawdopodobieństwem $< \varepsilon$. Możesz korzystać z różnych polaryzatorów oraz luster.

Wskazówka: $\forall x \in \mathbb{R}_+ \ \sin x < x$.

[1] Można myśleć, że dzielnik dokonuje pomiaru na fotonie, a następnie wysyła nowy, odpowiednio spolaryzowany foton (H/V) w odpowiednim kierunku

[2] Gdy bomba wybucha, uważamy ją za wykrytą.

2. Paradoks EPR

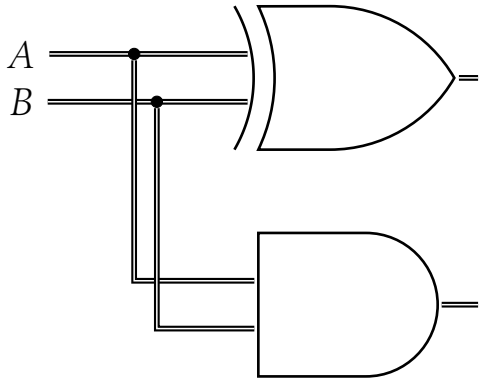
Zadanie 3 Punktów: 2

Wytłumacz eksperyment zaproponowany przez Bella i jego konsekwencje.

Uwaga: Dla mnie dość ciekawy był ten filmik: https://www.youtube.com/watch?v=7zfnvGXpy-g. Żeby go zrozumieć trzeba wiedzieć, że spin jest taką samą własnością kwantową jak polaryzacja, tyle że stany bazowe to ↑/↓, a nie H/V. Dobrym ćwiczeniem byłoby wyjaśnienie tego samego ale na polaryzacji.

3. Obwody logiczne

Na następnym wykładzie będziemy konstruować algorytmy w postaci obwodów logicznych, takich jak półsumator na rysunku. Obwód jest DAG-iem, którego wierzchołki źródłowe to wejście (bity). Wierzchołki wewnętrzne (o stałym stopniu) to bramki logiczne, a w ujściach jest wynik obliczenia.



Rysunek 1: Obwód realizujący półsumator. Użyto w nim bramek COPY, XOR i AND.

Zadanie 4

Trudność: trudne Punktów: 5

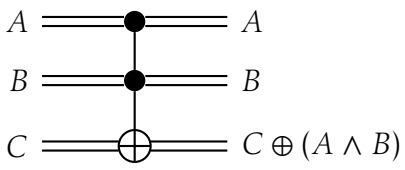
Pokaż, że dla każdego języka $\mathcal{L} \in \mathsf{P}$ istnieje rodzina $\{C_n\}_{n\in\mathbb{N}}$ obwodów logicznych zbudowanych z bramek COPY, AND, OR i NOT, że C_n przyjmuje n bitów wejścia i rozstrzyga, czy słowo należy do języka $\mathcal{L}$, a rozmiary (liczbę bramek) obwodów z rodziny można ograniczyć wielomianem.

Wskazówka: Warto zastanowić się, jak przerobić maszynę Turinga, by jej spacer po taśmie nie zależał od wejścia.

Zadanie 5

Trudność: łatwe Punktów: 1

Bramka Toffoliego (CCNOT, *Controlled-Controlled-NOT*) działa jak na rysunku:



Jak zbudować za jej pomocą półsumator **nie rozdzielając kabelków** (używamy tylko bramek CCNOT, nie używamy bramek COPY)? Do wejścia można dołożyć pulę bitów jedynkowych.

Zadanie 6

Trudność: łatwe Punktów: 1

Pokaż, że każdy obwód logiczny zbudowany z bramek COPY, AND, OR i NOT można zrealizować wyłącznie za pomocą bramek CCNOT (możemy dołożyć pulę bitów jedynkowych na wejściu oraz śmieci na wyjściu).

Zadanie 7

Trudność: łatwe Punktów: 1

Pokaż, że obliczenia realizowane przez obwód logiczny zbudowany z bramek CCNOT są odwracalne.

Zadanie 8

Trudność: łatwe Punktów: 1

Pokaż, że każdy obwód zbudowany z bramek CCNOT można przerobić na równoważny, planarny obwód z bramek CCNOT.

3.1 Odśmiecanie

Jak przekonaliśmy się dzięki poprzednim zadaniom, dla dowolnej funkcji $f : \{0,1\}^n \to \{0,1\}^m$ realizowanej przez obwód klasyczny zbudowany z bramek AND, OR i NOT można stworzyć odwracalny obwód złożony wyłącznie z bramek CCNOT realizujący tę samą funkcję — ale poszerzoną o wejście pomocnicze (*ancilla*) oraz wyjście-śmieci. Gdy zgodzimy się na bramki CCNOT i NOT, można założyć, że wszystkie bity pomocnicze mają wartość 0. To znaczy, nasz obwód C będzie przyjmował $n+a$ bitów wejścia i zwracał $m+b$ bitów wyjścia ($n+a=m+b$, gdyż obwód jest odwracalny). Gdy bity pomocnicze będą zainicjalizowane zerami, będzie on obliczał funkcję f:

$$C(x_1,\ldots,x_n,\overbrace{0,\ldots,0}^{a \text{ ancilla}}) = [f(x)_1,\ldots,f(x)_m,\overbrace{g(x)_1,\ldots,g(x)_b}^{b \text{ bitów śmieciowych}}]$$

My jednakowoż chcielibyśmy zredukować użycie bitów śmieciowych.

Zadanie 9

Trudność: łatwe Punktów: 2

Jak przekształcić obwód C na odwracalny obwód C' o następującej specyfikacji: Gdy C' dostanie na wejściu wektor $[x_1,\ldots,x_n,0^n,0^a]$ (w sumie $2n+a$ bitów), to zwróci $[x_1,\ldots,x_n,f(x)_1,\ldots,f(x)_m,g(x)_1,\ldots,g(x)_b]$ ($n+m+b$ bitów)?

Zadanie 10

Trudność: łatwe Punktów: 2

Jak przekształcić obwód C' na odwracalny obwód C'' o następującej specyfikacji: Gdy C'' dostanie na wejściu wektor $[x_1,\ldots,x_n,0^n,0^a,y_1,\ldots,y_m]$ (czyli $2n+a+m$ bitów; y jest dowolnym wektorem boolowskim), to zwróci $[x_1,\ldots,x_n,0^n,0^a,y\oplus f(x)]$.