Quantum Shell – obliczenia kwantowe

Jakub Pilch

Poniższy dokument jest pierwszym z dwóch stanowiących pełną dokumentację projektu Quantum Shell. Drugim z nich jest *Quantum Shell – emulator komputera kwantowego* zawierający opis praktycznego podejścia do stworzenia aplikacji.

I Wprowadzenie

Pojęcia i intuicje związane z obliczeniami kwantowymi często stoją w sprzeczności z tymi związanymi z klasyczną teorią obliczeń i złożoności obliczeniowej. Aby nieco rozjaśnić temat postanowiłem na samym początku wprowadzić kilka pojęć i zasad, które mogą okazać się niezbędne do zrozumienia idei obliczeń kwantowych (a tym bardziej do stworzenia emulatora obliczeń kwantowych na „zwykłe" procesory oparte o krzem).

1. Silna Teza Churcha – Turinga

Aby zacząć rozważania o obliczalności i roli obliczeń kwantowych w informatyce, musimy zdefiniować podstawową maszynę obliczeniową.

Maszyna Turinga

Abstrakcyjna maszyna, składająca się z:

- nieskończonej taśmy podzielonej na pola zawierające pojedyncze znaki z alfabetów wejściowego i roboczego,
- głowicy, mogącej naraz odczytać jeden znak z taśmy lub zapisać jeden znak na taśmie oraz przesuwać się o jedno pole w lewo lub prawo,
- funkcji przejścia, jednoznacznie określającej (w przypadku deterministycznej maszyny Turinga) zachowanie maszyny w danej konfiguracji (tj. przy głowicy nad danym znakiem, co ma zostać zapisane na taśmie i w którą stronę głowica ma się przesunąć).

Swoje szczególne miejsce w teorii obliczeń maszyna Turinga zawdzięcza następującej tezie:

Teza Churcha – Turinga:

Każdy intuicyjnie obliczalny problem jest rozwiązywalny przez maszynę Turinga.

Oczywiście teza ta nie jest weryfikowalna (choćby z powodu użytego w niej sformułowania „intuicyjnej obliczalności"), jednak od lat traktowana jest jako jedna z podstaw informatyki. Należy zwrócić uwagę, że traktuje ona o obliczalności ogółem, nie wspominając przy tym o złożoności obliczeń.

Mianem uniwersalnej maszyny Turinga nazywamy taką maszynę, która potrafi zasymulować dowolną inną poprzez wczytanie konfiguracji symulowanej maszyny na początku swojego działania. Jak można się łatwo domyślić, taka symulacja nie obejdzie się bez kosztów i samo obliczenie może zająć wielomianowo lub nawet wykładniczo więcej czasu niż gdyby było przeprowadzone na dedykowanej do niego maszynie Turinga.

Z punktu widzenia obliczalności UMT jest równoważna deterministycznej maszynie Turinga.

Wyobraźmy sobie teraz następujące uogólnienie maszyny Turinga: poza podstawowe elementy wymienione w definicji wyposażymy ową maszynę w generator binarnych liczb losowych (czyli np. maszynkę rzucającą monetą i mogącą sprawdzić jej stan). Taką maszynę nazywamy probabilistyczną maszyną Turinga. Podobnie jak w przykładzie wyżej, można wykazać równoważność PMT z klasyczną, deterministyczną maszyną Turinga z punktu widzenia obliczalności. Jeżeli jednak rozpatrzymy złożoność obliczeniową problemów to pojawią się znaczące różnice. Istnieją problemy, dla których znamy efektywne (wielomianowe) algorytmy na maszyny probabilistyczne, lecz najszybsze algorytmy deterministyczne potrzebują na ich rozwiązanie czasu wykładniczego. Mając powyższą wiedzę możemy wprowadzić kolejną definicję:

Klasyczna Silna Teza Churcha – Turinga:

Probabilistyczna maszyna Turinga może efektywnie symulować dowolny realistyczny model obliczeń.

Istotną różnicą w stosunku do wcześniejszej, podobnej definicji jest sformułowanie „realistyczny model obliczeń". Można budować deterministyczne maszyny dedykowane do konkretnych problemów, które będą potrzebowały istotnie (nawet wykładniczo) mniejszej liczby operacji do rozwiązania danego problemu niż uniwersalna maszyna probabilistyczna. Należy jednak zwrócić uwagę, że sama liczba wykonanych operacji może nie być tutaj właściwą miarą złożoności, ponieważ dedykowane maszyny mogą wykorzystywać pewne fizyczne czynności wymagające nawet ponadwielomianowych zasobów. Mówiąc o silnej tezie Churcha – Turinga mamy na myśli istotnie *realistyczny* model obliczeń, tj. zgodny z obowiązującymi prawami fizyki i uwzględniający *wszystkie* zasoby używane do obliczeń.

Klasyczna silna teza Churcha – Turinga wprowadza jednak pewien problem – ograniczenia fizyczne. Jak wiadomo, klasyczne zasady fizyki niezbyt dobrze nadają się do opisu zjawisk kwantowych. W tak małej skali konieczne jest stosowanie zasad fizyki kwantowej aby adekwatnie opisywać rzeczywistość. Tym samym powyższa teza nie uwzględnia pewnego realistycznego modelu obliczeń jakim jest komputer kwantowy. Aby zacząć opisywać obliczenia kwantowe musimy pójść o krok dalej i zapisać jeszcze jedną, nieco uogólnioną i istotną z naszego punktu widzenia tezę:

Kwantowa Silna Teza Churcha – Turinga:

Kwantowa maszyna Turinga może efektywnie symulować dowolny realistyczny model obliczeń.

Wprowadzenie powyższych pojęć stanowi dobrą podstawę do uzasadnienia wartości badania obliczeń kwantowych. Skoro kwantowa maszyna Turinga może *efektywnie* symulować dowolne realistyczne obliczenia, to pozwala w akceptowalnym (czyli zazwyczaj wielomianowym) czasie rozwiązywać problemy bardzo czasochłonne dla klasycznej, deterministycznej maszyny Turinga.

Aktualnie (2014r.) nie ma dowodu na to, że dowolnego obliczenia kwantowego nie da się efektywnie zasymulować na maszynach deterministycznych. Istnieją za to problemy, dla których najszybsze znane algorytmy deterministyczne potrzebują czasu wykładniczego

(lub nawet ponadwykładniczego), a znane są dla nich algorytmy kwantowe działające w czasie wielomianowym.

2. Układowy model obliczeń

W celu prowadzenia dalszych rozważań o obliczeniach kwantowych (zwłaszcza w kontekście budowy emulatora) pomocnym będzie wprowadzenie jeszcze jednego modelu obliczeń – modelu układowego (Circuit Model of Computation). Zamiast o maszynach Turinga będziemy mówić o kompletnych zestawach prostych układów odwracalnych. Zdefiniujmy zatem co mamy na myśli mówiąc „układ":

Układ

Układem nazywamy urządzenie złożone z przewodów mogących przekazywać binarne wartości do bramek wykonujących operacje logiczne na dostarczonych bitach.

Do naszych rozważań potrzebujemy aby nasze układy spełniały pewne wymogi. Po pierwsze, bierzemy pod uwagę wyłącznie układy acykliczne, więc takie, w których informacje zapisane w bitach przechodzą tylko w jedną stronę i nie istnieją pętle powrotne. Po drugie, od zestawu bramek wymagamy uniwersalności:

Uniwersalność bramek

Zestaw bramek nazywamy uniwersalnym dla klasycznych obliczeń jeżeli dla dowolnych dodatnich liczb całkowitych n, m i funkcji $f: \{0,1\}^n \to \{0,1\}^n$ można stworzyć układ obliczający f korzystający wyłącznie z bramek z tego zestawu.

Po trzecie: ustalamy, że w jednej jednostce czasu dany bit informacji może wejść do co najwyżej jednej bramki (a operacja wykonywana przez pojedynczą bramkę zajmuje jedną jednostkę czasu). Po czwarte i ostanie, wymagamy aby bramki wykonywały operacje odwracalne.

Odwracalność bramek

Bramka jest odwracalna (wykonuje operację odwracalną) jeżeli dla zestawu danych na wyjściu bramki możemy jednoznacznie określić jakie dane były na wejściu.

Typowym przykładem uniwersalnych bramek dla obliczeń klasycznych są bramki NAND i FANOUT – z ich pomocą możemy zbudować układy obliczające wyniki dowolnych obliczalnych funkcji. Jeśli jednak ograniczymy się wyłącznie do bramek odwracalnych sprawa nieco się komplikuje – nie możemy wtedy osiągnąć uniwersalności używając wyłącznie bramek operujących na jednym lub dwóch bitach. Wprowadźmy zatem nową, 3-bitową bramkę:

Bramka Toffoli'ego

Zasada działania: jeżeli (i tylko wtedy gdy) pierwsze dwa wejściowe bity są ustawione na 1 to neguje ona trzeci bit. W pozostałych przypadkach nie zmienia na wyjściu nic.

Zestaw składający się wyłącznie z bramki Toffoli'ego (która jest odwracalna) jest kompletny dla obliczeń klasycznych (pod warunkiem możliwości użycia generatorów sygnału 0 lub 1).

O ile w przypadku maszyn Turinga mówiąc o złożoności obliczeniowej mieliśmy na myśli czas lub pamięć potrzebne do obliczeń, o tyle w układowym modelu mówić będziemy o ilości bramek użytych w układzie oraz o tzw. głębokości układu (czyli ilości kolejnych rzędów równoległych bramek).

Podobnie jak wcześniej dla maszyny Turinga, dla układowego modelu obliczeń również możemy wprowadzić probabilistyczny element: generator losowego bitu, który w jednej jednostce czasu generuje bit o losowej wartości binarnej (niezależnie od bitów wejściowych).

Aby zapisać aktualny stan maszyny deterministycznej moglibyśmy wypisać wszystkie wartości bitów w danym fragmencie układu. W maszynie probabilistycznej będziemy rozpatrywać wektory prawdopodobieństw powiązanych z wystąpieniami konkretnych wartości w danym miejscu układu.

$$\begin{pmatrix} p_0 \\ p_1 \end{pmatrix}$$

Powyższy zapis należy interpretować następująco: w badanym fragmencie układu bit może mieć wartość 0 z prawdopodobieństwem p_0 i wartość 1 z prawdopodobieństwem p_1.

Skoro używamy wektorów do zapisu wartości bitów w układzie, chcielibyśmy móc zapisywać operacje wykonywane przez bramki w formie pozwalającej na ich zaaplikowanie do wektora wartości. Z tego powodu bramki w algebraicznej reprezentacji układowego modelu obliczeń będziemy zapisywać jako macierze. Prostym przykładem może być bramka NOT:

$$NOT\begin{pmatrix} p_0 \\ p_1 \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} p_1 \\ p_0 \end{pmatrix}$$

Wykonuje ona operację odwracalną, więc spełnia nasze wymaganie.

Wreszcie chcąc zapisać stan wielobitowego układu probabilistycznego w danym miejscu musimy zaprezentować wszystkie możliwe kombinacje bitów w układzie wraz z towarzyszącymi im prawdopodobieństwami wystąpienia. Przykładowo przyjmijmy, że chcemy opisać stan układu zawierającego wyłącznie dwa przewody. Prawdopodobieństwo wystąpienia wartości 1 wynosi p_1 dla pierwszego przewodu i q_1 dla drugiego. Odpowiednio prawdopodobieństwa wystąpienia wartości 0 wynoszą p_0 i q_0. Istnieją więc cztery możliwe stany takiego układu, co zapisujemy w postaci wektora:

$$\begin{pmatrix} p_0 q_0 \\ p_0 q_1 \\ p_1 q_0 \\ p_1 q_1 \end{pmatrix}$$

Skoro możemy reprezentować stany składające się z więcej niż pojedynczego bitu, możemy również macierzowo zapisywać bramki wykonujące operacje na większej liczbie bitów. Jako przykład przedstawmy prostą bramkę (której kwantowa wersja okaże się kluczowa dla obliczeń kwantowych):

Bramka *CNOT*

Przyjmuje dwa wejściowe bity, kontrolny i docelowy. Jeżeli bit kontrolny ustawiony jest na 1, bit docelowy jest negowany. W przeciwnym wypadku na wyjściu nie zmienia się nic.

Bramka *CNOT* (controlled NOT) może być zapisana macierzowo w następujący sposób:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Z powyższych rozważań widać, że układ obliczeniowy możemy zapisać algebraicznie, za pomocą wektorów prawdopodobieństw wystąpień konkretnych wartości bitów i macierzy reprezentujących bramki logiczne.

Układowy model obliczeń jest istotnym krokiem w stronę zrozumienia idei i działania obliczeń kwantowych.

3. Stan systemu kwantowego - Qubity

W latach trzydziestych XX wieku świat fizyki przeszedł pewnego rodzaju rewolucję, kiedy okazało się, że klasyczne zasady znane od lat nie wystarczają do opisu zjawisk które zaczęto badać. Rozpoczęto tworzenie niejako nowego modelu fizyki, który z założenia miał być zgodny z obserwacjami eksperymentów i tym samym pozwalać na ich lepsze zrozumienie.

Podobnie w informatyce, chcąc opisywać i przetwarzać informacje „zapisane kwantowo" jesteśmy zmuszeni dostosować się do pewnych zasad rządzących mechaniką kwantową, a tym samym zmienić nieco nasze dotychczasowe podejście do informacji ogółem. Można powiedzieć, że informatykę czeka transformacja podobna do tej, którą fizyka przeszła około połowy ubiegłego wieku.

Klasyczna informatyka przyzwyczaiła nas do binarnej reprezentacji danych. Zgodnie ze słowami twórcy teorii informacji Claude'a E. Shannon'a, niemal każdą informację można

zapisać w postaci skończonego ciągu zer i jedynek. Pojedynczy bit może przyjmować jeden z dwóch stanów: prawdę (1) lub fałsz (0), a ciąg takich bitów może opisywać dowolny klasyczny system informacyjny.

Kwantowe systemy opisywane są za pomocą …kwantów informacji, czyli po prostu pewnych dyskretnych stanów. Dzięki swojej dyskretności stan może być opisany za pomocą pewnych wektorów:

Postulat przestrzeni stanów

Każdy układ fizyczny może być całkowicie opisany przez jednostkowy wektor stanu należący do przestrzeni Hilberta $\mathcal{H}$.

Wymiar przestrzeni Hilberta z powyższego postulatu zależy od stopni swobody wybranego systemu. Z punktu widzenia informatyki jesteśmy zainteresowani systemami z dwoma możliwymi stanami (odpowiadającymi 1 lub 0). Przykładowym modelem fizycznym spełniającym takie wymagania jest spin cząstki, który można opisać wektorem jednostkowym z przestrzeni Hilberta. Spin-up może być reprezentowany jako $|0\rangle$, a spin-down jako $|1\rangle$ (wektorowa notacja Diraca).

Z powyższego opisu mogłoby się wydawać, że kwantowy opis stanu niczym nie różni się od klasycznego – dla najprostszej paczki informacji mamy dostępne dwa stany, prawdę lub fałsz. Jednak w przypadku stanu kwantowego możemy mieć do czynienia z *superpozycją* stanów 1 i 0, co wynika z faktu, że nigdy do końca nie znamy wszystkich wartości opisujących nasz system kwantowy w danym momencie.

Aby nabrać nieco intuicji: wyobraźmy sobie pewien „probabilistyczny bit", którego wartości nie możemy określić dokładnie. Wiemy jedynie, że może on znajdować się w jednym ze stanów 0 lub 1 z prawdopodobieństwami odpowiednio p_0 i p_1, gdzie $p_0 + p_1 = 1$ (gdyż nasz „bit" na pewno przyjmuje 0 lub 1). Ogólny stan systemu opartego o „probabilistyczny bit" moglibyśmy więc zapisać jako $p_0 + p_1$.

Superpozycja dwóch możliwych stanów kwantowych wymaga od nas pójścia o krok dalej i uwzględnia, że współczynniki (tzw. amplitudy) przy wektorach opisujących stan mogą być zespolone[1,2]. Dodatkowo zauważmy, że zespoloną amplitudę a możemy zdekomponować jako $a = e^{i\theta}|a|$, oraz że zgodnie z Postulatem przestrzeni stanów stan systemu opisuje znormalizowany wektor z przestrzeni Hilberta. Ponieważ z fizycznego punktu widzenia globalna faza (odchylenie w kierunku osi liczb urojonych) nie ma znaczenia, możemy przyjąć, że współczynnik przy $|0\rangle$ jest liczbą rzeczywistą (pozbywając się tym samym jego zespolonej części) i rozpatrywać tylko fazę relatywną. Tym samym stan pojedynczego „bitu kwantowego" możemy najprościej zapisać jako:

Uogólniony zapis stanu pojedynczego bitu kwantowego (qubitu)

$$\cos\left(\frac{\theta}{2}\right)|0\rangle + e^{-i\varphi} sin\left(\frac{\theta}{2}\right)|1\rangle$$

W powyższym zapisie $|0\rangle$ i $|1\rangle$ to wektory bazowe naszej przestrzeni, θ jest kątem odchylenia wektora stanu w osi stanów podstawowych (czyli jest szerokością na sferze

Blocha[1,2]), a φ kątem relatywnej fazy wektora stanu (czyli długością na sferze Blocha). We wzorze zapisujemy $\left(\frac{\theta}{2}\right)$ zamiast θ, ponieważ chcemy traktować wektory |0⟩ oraz |1⟩ jako prostopadłe z kartezjańskiego punktu widzenia (kąt pomiędzy |0⟩ i |1⟩ jest równy π, dlatego w powyższym zapisie wektor |1⟩ będziemy rozpatrywać jako kartezjańsko prostopadły do |0⟩). „Kwantowe bity", których stany opisujemy jak powyżej nazywamy qubitami (z ang. quantum bits).

Pewna intuicja dotycząca bitów, bitów probabilistycznych i qubitów jest następująca:

- **stan klasycznego bitu deterministycznego** możemy graficznie przedstawić jako oznaczony jeden z dwóch punktów (0 lub 1),
- **stan klasycznego bitu probabilistycznego** możemy przedstawić jako punkt na odcinku jednostkowym, zakończonym stanami 0 i 1, odległy o p_1 od stanu 1 i o p_0 od stanu 0,
- **stan bitu kwantowego (qubitu)** możemy przedstawić jako wektor wodzący punktu na trójwymiarowej sferze (zwanej sferą Blocha).

Możemy stwierdzić, że w swojej istocie qubit przyjmuje nieskończenie wiele stanów[7] (co nawiązuje do nieskończonej liczby punktów na sferze), lecz podczas odczytu zawsze ujawnia stan 0 lub 1.

4. Systemy zamknięte i splątanie qubitów

Postulat ewolucji stanu zamkniętego systemu kwantowego

Ewolucja stanu zamkniętego systemu kwantowego w czasie może być opisana przez operator unitarny (taki, którego złożenie z jego operatorem sprzężonym jest identycznością).

Z powyższego postulatu[1] bezpośrednio wynika, że dla dowolnej ewolucji (ciągu przekształceń) zamkniętego systemu kwantowego istnieje operator unitarny U taki, że jeżeli początkowy stan systemu reprezentujemy jako $|v_1\rangle$, a końcowy jako $|v_1\rangle$ to $|v_2\rangle = U|v_1\rangle$. Mówiąc o obliczeniach kwantowych, unitarny operator działający na pojedynczym qubicie będziemy nazywać **jedno-qubitową bramką kwantową.** Warto również zauważyć, że używanie operatorów unitarnych zapewnia nas o odwracalności przekształceń.

Zgodnie z tym, co zostało napisane wcześniej (w części 2., o układowym modelu obliczeń), bramki możemy przedstawiać w postaci macierzy. Konkretnie, bramkę dla pojedynczego qubita (w przestrzeni Hilberta o wymiarze 2) reprezentujemy jako macierz 2 x 2. Każda bramka określona jest jako zestaw przekształceń dla każdego możliwego wejścia. Działanie bramki dla konkretnego wejścia możemy zapisać w postaci iloczynu stanu wejściowego systemu (w postaci wektora kolumnowego) z macierzą reprezentującą bramkę (zgodnie z opisem dla układowego modelu obliczeń).

Aż do tej pory nie został poruszony w tym opracowaniu temat zachowania układu złożonego z wielu qubitów. Na podstawie informacji podanych wyżej można opisać system złożony z wielu odizolowanych qubitów, jednak taki model nie jest zbyt praktyczny do obliczeń. Aby rozważać obliczenia na potrzebujemy sposobu na opisanie ich zachowania kiedy mogą wchodzić ze sobą w interakcje. Z pomocą przychodzi nam następujący postulat[1]:

Postulat kompozycji systemów

Traktując dwa fizyczne systemy jako jeden połączony system, stan takiej kombinacji jest produktem tensorowym przestrzeni systemów składowych. Jeśli $|v_1\rangle$ to stan pierwszego systemu, a $|v_2\rangle$ drugiego z nich, to stan połączonych systemów jest postaci $|v1\rangle \otimes |v2\rangle$**.**

Powyższy postulat może być skalowany przez indukcję do systemu stworzonego z połączenia *n* podsystemów.

Jednak nie każdy 2-qubitowy system daje się przedstawić w postaci produktu tensorowego. Dwa odizolowane qubity tworzą dwa osobne zamknięte systemy, dlatego ich złożenie może być zapisane jak wyżej. Jeżeli jednak qubity nie są od siebie izolowane, to mogą wchodzić ze sobą w interakcje i tworzyć stany niemożliwe do zapisania w postaci produktu tensorowego. W takiej sytuacji mówi się o ***splątaniu qubitów*****.** Przykładem może być następujący stan 2-qubitowego systemu:

$$|\boldsymbol{v}\rangle = \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{11}\rangle$$

Można pokazać, że nie istnieją współczynniki takie, żeby powyższy stan zapisać jako produkt tensorowy dwóch stanów[1,8]. Podany przykład to tzw. Para EPR, od nazwisk Einstein, Podolsky, Rosen, którzy rozważali ją w swoich badaniach mechaniki kwantowej.

Z chwilą, w której dokonujemy pomiaru qubitu, system przestaje być zamknięty, a tym samym jego stan zostaje zniszczony. Zachowanie systemu kwantowego przy pomiarze opisuje postulat[9]:

Postulat pomiaru stanu kwantowego

Pomiar stanu kwantowego może być reprezentowany przez zbiór operatorów $\{M_m\}$ operujących na przestrzeni stanów systemu. Prawdopodobieństwo *p* odczytania wartości *m* podczas pomiaru stanu $|v\rangle$ wynosi

$$p(m) = \langle v|M_m^t M|v\rangle$$

Po odczycie system zostaje pozostawiony w stanie

$$|v'\rangle = \frac{M_m|v\rangle}{\sqrt{p(m)}}$$

Z powyższego postulatu wynika m.in. fakt, że przy ponownym odczycie stanu $|v\rangle$ (następującym przed jakąkolwiek zmianą stanu) odczyt tej samej wartości co za pierwszym razem jest zdarzeniem pewnym.

5. Implementacja qubitu

Zgodnie z wcześniejszym opisem wiemy, że od qubitu wymagamy kilku własności:

- Możliwości odczytu z niego informacji 0 lub 1
- Możliwości wejścia w superpozycje stanów 0 i 1
- Możliwości wchodzenia w stan splątania z innymi qubitami

Jednak oprócz wymagań teoretycznych implementacja qubitu musi spełniać pewne stricte praktyczne wymagania:

- Możliwość manipulacji (ustawienie wejścia, resetowanie stanu)
- Stabilność
- Wykonalność

Choć początkowo mogą one brzmieć nieco absurdalnie, problemy implementacyjne wcale nie są oczywiste, a różne podejścia napotykają różne przeszkody.

Jedną z możliwych reprezentacji fizycznych qubitu jest foton, którego polaryzacja może reprezentować stan qubitu. Inną, często rozpatrywaną implementacją jest spin elektronu, który również może być opisany przez wektor dwuwymiarowej przestrzeni Hilberta. Jeszcze innym podejściem jest wybranie cząstek, w których różnica energetyczna między dwoma pierwszymi poziomami jest niewielka w stosunku do różnicy wobec poziomów kolejnych (elektron może znajdować się w stanie ekscytacji lub nie). Rozważa się także kierunek przepływu prądu w nadprzewodniku oraz zwrot mikroskopijnego pola magnetycznego wytwarzanego w maleńkich obwodach nadprzewodzących.

Najbardziej palącym problemem w kwestii implementacji qubitu jest jego stabilność. Ponieważ zjawiska kwantowe zachodzą jedynie w nano-skali, nawet najmniejsze cząsteczki czy też fale elektromagnetyczne mogą zakłócać działanie qubitu, wywołując jego dekoherencję (tj. sprowadzenie do jednego ze stanów podstawowych), a tym samym – utratę informacji. Co więcej, zgodnie z postulatem z początku podrozdziału I-4, należy upewnić się, że qubity wchodzące w skład procesora kwantowego nie będą splątane z innymi cząsteczkami (zwłaszcza z zewnątrz procesora). W przypadku zaniechania tego kroku może się okazać, że stworzony system nie jest zamknięty, co czyni go bezużytecznym do obliczeń (ponieważ operatory unitarne nie będą działać zgodnie z założeniami).

Z powyższych powodów procesory kwantowe, które próbuje się budować obecnie, opakowywane są w ołowiane osłony oraz potężne wymienniki ciepła, chłodzące rdzenie do temperatury nawet poniżej 1K (co samo w sobie nie jest zadaniem łatwym z inżynierskiego punktu widzenia).

Firma D-Wave, która słynie z badań z dziedziny obliczeń kwantowych, wykorzystuje ostatnią z ww. Implementacji qubitu[6]. Należy zwrócić jednak uwagę, iż procesory D-Wave zostały stworzone do wykonywania pewnego konkretnego algorytmu zwanego *Quantum Annealing* (analogicznie do *Simulated Annealing* jest to algorytm wyszukiwania globalnego minimum funkcji), oraz że przy ich obecnej implementacji napotykanych jest wiele problemów inżynieryjnych (m.in. aktualnie nie stworzono procesora mogącego spowodować splątanie więcej niż 8. qubitów).

Firma Microsoft od kilkunastu lat prowadzi badania w kierunku innym niż pozostałe[5]. Pod kierownictwem Michael'a Freedman'a poszukują oni pewnej odmiany qubita, zwanej qubitem topologicznym. Sama idea może być tematem osobnej rozprawy, natomiast warto wspomnieć, że jeżeli badania zakończyłyby się sukcesem, oznaczałoby to

stworzenie qubitu bardzo stabilnego i odpornego na zakłócenia zewnętrzne (w przeciwieństwie do innych znanych implementacji). Aktualnie prowadzone są poszukiwania cząstek zwanych Anyonami (ang. *non-Abelian Anyons*), których istnienie stanowi podstawę rozważań nt. qubitów topologicznych.

II Kwantowy model obliczeń

Zrozumienie wspomnianego wcześniej układowego modelu obliczeń pozwala o wiele szybciej zrozumieć model kwantowy. Oczywiście pierwszy z nich nawiązywał do obliczeń klasycznych (a więc rządzących się nieco innymi zasadami niż kwantowe), jednak pozwolił wprowadzić m.in. pojęcie odwracalności obliczenia, które z punktu widzenia obliczeń kwantowych jest bardzo istotne.

Kwantowy model obliczeń przedstawia wizję bardzo zbliżoną do modelu układowego: na „wejściu" układu ustawiamy qubity, które następnie przetwarzane są przez kolejne bramki. Zwróćmy jednak uwagę, że qubity nie popłyną między bramkami przez przewody (ze względu na implementacje, np. z użyciem spinu elektronu). W zamian można sobie to wyobrazić jako wykonywanie kolejnych operacji na pewnym rejestrze trzymającym qubity, aż do uzyskania wyniku (coś w rodzaju „przykładania bramek" do wspomnianego rejestru).

Chcąc wykorzystać kwantowy układ do obliczeń, ostatecznie musimy odczytać wynik w pewnej klasycznej, interpretowalnej postaci. Tutaj pojawia się problem – odczytanie informacji zapisanej kwantowo spowoduje zniszczenie wektora stanu poprzez sprowadzenie go do 0 lub 1 (ukryte informacje, współczynniki dot. prawdopodobieństwa, zostaną całkowicie zniszczone). Dodatkowo dla wielu algorytmów wynik operacji przeprowadzonych na układzie kwantowym jest poprawny jedynie z pewnym prawdopodobieństwem. Z tego powodu wiele algorytmów uruchamia się wielokrotnie, co zwiększa prawdopodobieństwo poprawności odczytanego wyniku.

Poniższy rozdział przedstawia koncepcje ściśle związane z obliczeniami kwantowymi oraz opisuje ich implementacje i działanie.

1. Jedno-qubitowe bramki kwantowe

Bramki kwantowe mają za zadanie wykonać określone przekształcenia na wejściowym stanie przy określonych warunkach. Jak wiadomo z wcześniejszej części opracowania, stan systemu kwantowego opisujemy przy pomocy wektorów reprezentujących qubity. Operacje wykonywane przez bramki kwantowe mogą być więc postrzegane jako zmiana parametrów wektora reprezentującego stan.

Przyjrzyjmy się bliżej bramkom jedno-qubitowym. Zgodnie z intuicją na koniec podrozdziału I-3 stan qubitu można przedstawić jako wektor wodzący punktu na sferze zwanej sferą Blocha. Weźmy przykładową bramkę kwantową G przetwarzającą jeden qubit. Przekształcenie stanu wejściowego dokonywane przez taką bramkę może być postrzegane jako obrót wektora reprezentującego stan wejściowy w sferze Blocha. Jeżeli za bramkę G przyjmiemy bramkę kwantową NOT, to przekształcenie przez nią dokonywane możemy rozpatrywać jako rotację wektora stanu wejściowego względem osi x.

Możemy wyróżnić cztery istotne jedno-qubitowe bramki kwantowe, zwane bramkami Pauliego:

$I \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
$Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	$Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

Zwróćmy uwagę, że każda z nich nawiązuje do pewnej rotacji wektora stanu. Bramki te możemy więc zapisać jako funkcje (uwzględniając przy tym współczynniki reprezentujące kąt):

$$\boldsymbol{R_x(\theta)} \equiv \boldsymbol{e^{\frac{-i\theta X}{2}}}$$

$$\boldsymbol{R_y(\theta)} \equiv \boldsymbol{e^{\frac{-i\theta Y}{2}}}$$

$$\boldsymbol{R_z(\theta)} \equiv \boldsymbol{e^{\frac{-i\theta Z}{2}}}$$

Znając powyższe funkcje możemy zapisać twierdzenie:

Dowolną jedno-qubitową bramkę kwantową możemy przedstawić jako sekwencję rotacji względem osi sfery Blocha:

$$\boldsymbol{U = e^{-i\propto} R_z(\beta) R_y(\gamma) R_z(\delta)}$$

Oczywiście nie znaczy to, że wszystkie użyteczne dla nas jedno-qubitowe bramki kwantowe będziemy zapisywać w postaci sekwencji rotacji (choć każdą z nich można tak interpretować). Kolejne dwie bramki zajmują pewne szczególne miejsca w kwantowym modelu obliczeń.

Bramka Hadamarda (zwykle oznaczana ***H***) mapuje wektory bazowe w następujący sposób:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Warto zaznaczyć, że bramką odwrotną do bramki Hadamarda jest ona sama ($H = H^{-1}$).

Bramka $\frac{\pi}{8}$ (zwykle oznaczana ***T***) mapuje wektory bazowe następująco:

$$T|0\rangle = |0\rangle$$

$$T|1\rangle = e^{i\frac{\pi}{4}}|1\rangle$$

$$T = \begin{bmatrix} e^{-i\frac{\pi}{8}} & 0 \\ 0 & e^{i\frac{\pi}{8}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}$$

Powyższa równoważność zapisów macierzowych zachodzi z dokładnością do globalnej fazy (która z punktu widzenia obliczeń nie ma znaczenia).

2. Kontrolowane bramki kwantowe

Podobnie jak w modelu układowym, w kwantowym modelu obliczeń chcemy używać bramek kontrolowanych, tj. wykonujących pewne przekształcenia tylko gdy spełnione są dane wymagania. Przedstawiona w podrozdziale I-2 bramka *CNOT*[1,4] jest przykładem bramki kontrolowanej, która neguje drugi z podanych na wejście bitów wtedy i tylko wtedy gdy pierwszy z nich jest w stanie wysokim (1).

W przypadku kwantowego modelu obliczeń kontrolowana bramka operuje na stanie kwantowym będącym superpozycją wejściowych stanów kwantowych. Mając daną jedno-qubitową bramkę kwantową G możemy zdefiniować odpowiadającą jej kontrolowaną, dwu-qubitową bramkę *c-G* w następujący sposób:

$$c\text{-}G|0\rangle|v\rangle = |0\rangle|v\rangle$$
$$c\text{-}G|1\rangle|v\rangle = |1\rangle G|v\rangle$$

3. Splątujące bramki kwantowe

Bramki splątujące

Dwu-qubitową bramkę kwantową nazywamy splątującą jeżeli dla pewnego stanu wejściowego $|v\rangle|u\rangle$ będącego produktem tensorowym stan wyjściowy nie jest produktem tensorowym (qubity w stanie wyjściowym są splątane).

Przykładem splątującej bramki kwantowej jest bramka *CNOT*, co można stosunkowo łatwo zaobserwować[8]:

Weźmy stan kwantowy $(\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle = \alpha|00\rangle + \beta|10\rangle$, gdzie $\alpha, \beta \neq 0$. Zauważmy, że taki stan uzyskujemy na wejściu kwantowej bramki *CNOT* jeśli jako qubit kontrolny podamy qubit w superpozycji $(\alpha|0\rangle + \beta|1\rangle)$, a jako qubit właściwy podamy $|0\rangle$.

Przypomnijmy macierzową postać bramki *CNOT*:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Stąd możemy zapisać $CNOT(\alpha|00\rangle + \beta|10\rangle) = \alpha|00\rangle + \beta|11\rangle$. Wyjściowy stan jest stanem splątanym, co możemy pokazać nie wprost:

Załóżmy, że $\alpha|00\rangle + \beta|11\rangle$, $\alpha, \beta \neq 0$, nie jest stanem splątanym, czyli da się go przedstawić jako produkt tensorowy dwóch stanów. Można więc zapisać:

$$\alpha|00\rangle + \beta|11\rangle = (k|0\rangle + l|1\rangle) \otimes (m|0\rangle + n|1\rangle) = km|00\rangle + kn|01\rangle + lm|10\rangle + ln|11\rangle$$

Z powyższego wynika, że $\{km \neq 0, kn = 0, lm = 0, ln \neq 0\}$. Spełnienie wszystkich tych warunków jest niemożliwe, czyli powyższe równanie jest sprzeczne, co należało dowieść.

4. Uniwersalny zestaw bramek kwantowych

Podobnie jak w przypadku obliczeń klasycznych, w kwantowym modelu obliczeń interesuje nas możliwość wykonywania skomplikowanych algorytmów (operujących na *n* qubitach). Jednocześnie nie możemy w nieskończoność konstruować bramek przyjmujących coraz większe ilości qubitów. Analogicznie do modelu klasycznego, podczas konstruowania kwantowego modelu obliczeń chcemy znaleźć skończony zbiór bramek, z których można zbudować potencjalnie dowolny układ obliczeniowy.

Dopasujmy teraz naszą definicję uniwersalnego zestawu bramek do kwantowego modelu obliczeń.

Uniwersalny zestaw bramek kwantowych

Zestaw bramek kwantowych nazywamy uniwersalnym jeżeli dla dowolnej liczby naturalnej $n > 0$ dowolny n-qubitowy operator unarny może być przybliżony z ustaloną dokładnością przez układ kwantowy złożony wyłącznie z bramek z tego zestawu.

Wcześniejsze twierdzenie (z podrozdziału II-1) o przedstawianiu jedno-qubitowych bramek jako sekwencji rotacji można nieco uogólnić, aby dotyczyło bramek innych niż bramki Pauliego:

Zestaw dwóch jedno-qubitowych bramek jest uniwersalny dla jedno-qubitowych bramek jeśli:

1. Osie, wokół których dane bramki wykonują rotacje nie są równoległe,
2. Bramki wykonują rotacje o kąty α i β takie, że $\alpha, \beta \in [0, 2\pi]$ i $\frac{\beta}{\pi}$ oraz $\frac{\alpha}{\pi}$ są niewymiernymi liczbami rzeczywistymi.

Można pokazać, że para bramek HTHT oraz THTH spełnia powyższe twierdzenie, z czego wynika, że zestaw {H, T} jest uniwersalnym zestawem bramek jedno-qubitowych.

Ponieważ potrzebujemy móc przybliżyć wyniki działań dowolnych n-qubitowych bramek, nasz uniwersalny zestaw musi zawierać przynajmniej jedną bramkę operującą

na dwóch lub więcej qubitach. Przykładem takiej bramki jest *CNOT* operująca na dwóch qubitach.

Znając powyższe twierdzenia i definicje możemy zapisać następujące twierdzenie:

Zestaw bramek kwantowych zawierający wszystkie bramki 1-qubitowe i dowolną 2-qubitową bramkę splątującą jest uniwersalny.

Na mocy dwóch twierdzeń z tego podrozdziału oraz twierdzenia o bramkach splątujących z podrozdziału II-3 możemy stwierdzić, że zestaw bramek kwantowych {*CNOT, H, T*} jest uniwersalny.

5. Stany Bella

Istotnymi z punktu widzenia zastosowań kwantowego modelu obliczeń są tzw. Stany Bella. Jest to grupa 4. stanów kwantowych tworzących ortonormalną bazę:

$\|\beta_{00}\rangle = \frac{1}{\sqrt{2}}\|00\rangle + \frac{1}{\sqrt{2}}\|11\rangle$	$\|\beta_{01}\rangle = \frac{1}{\sqrt{2}}\|01\rangle + \frac{1}{\sqrt{2}}\|10\rangle$
$\|\beta_{10}\rangle = \frac{1}{\sqrt{2}}\|00\rangle - \frac{1}{\sqrt{2}}\|11\rangle$	$\|\beta_{11}\rangle = \frac{1}{\sqrt{2}}\|01\rangle - \frac{1}{\sqrt{2}}\|1\rangle$

Każdy z powyższych stanów jest parą EPR. Aby zmienić bazę dwóch qubitów *i, j* na bazę Bella β_{ij} należy przetworzyć qubit *i* za pomocą bramki Hadamarda, a następnie wykonać operację *CNOT* na parze qubitów *i, j*, gdzie *i* jest qubitem kontrolnym. Wykonanie powyższego przekształcenia w odwrotnej kolejności przynosi odwrotny efekt (tj. powoduje zmianę bazy qubitów z bazy Bella na standardową bazę obliczeniową).

III Protokoły kwantowe

Z punktu widzenia informatyki cała teoria obliczeń kwantowych nabiera znaczenia dopiero przy jej zastosowaniu. Celem, dla którego ludzie od lat próbują zrozumieć możliwości obliczeń kwantowych oraz wydają miliony dolarów na próby stworzenia komputera kwantowego są pewne właściwości algorytmów i protokołów kwantowych pozostające poza zasięgiem dowolnego klasycznego modelu.

Poniższy rozdział ma za zadanie przedstawić najbardziej znane protokoły kwantowe wraz z towarzyszącymi im problemami i potencjalnymi zastosowaniami. Choć nie mają one bezpośredniego związku z tworzeniem emulatora komputera kwantowego, uznałem, że warto je zamieścić ze względu na ich wartość merytoryczną i pomoc w zrozumieniu istoty kwantowego modelu obliczeń.

1. Kodowanie supergęste

Przesyłanie informacji odgrywa kluczową rolę w technologii informacyjnej. Od prostych rozmów telefonicznych, przez zakupy w Internecie aż po bankowość elektroniczną – informacje przesyłane są często i w dużych ilościach. Jednymi z podstawowych

problemów telekomunikacji są zabezpieczenie danych przed niepowołanym dostępem oraz jak najoszczędniejsze przesyłanie informacji.

W klasycznej informatyce możemy powiedzieć, że informacje przesyłane są w sposób oszczędny, kiedy wybrane kodowanie nie daje zbyt wielkiego narzutu informacji. Można stwierdzić, że przesłanie informacji bez żadnego narzutu, a z gwarancją poprawności byłoby doskonałym kodowaniem. Wykorzystanie mechaniki kwantowej otwiera przed nami nowe możliwości, umożliwiając przesłanie dwóch klasycznych bitów poprzez przesłanie tylko jednego qubitu kanałem komunikacji kwantowej – tym właśnie jest kodowanie supergęste.

Posługując się konwencją typową dla kryptografii, załóżmy, że Alicja chce przesłać do Boba dwa klasyczne bity informacji. Chcąc wykorzystać mechanikę kwantową wymagamy, aby odbiorca i nadawca byli połączeni tzw. kwantowym kanałem komunikacyjnym (który może przyjmować różne postaci, w zależności od implementacji qubitu - dla fotonów może to być np. światłowód), oraz aby każde z nich posiadało po jednym qubicie ze splątanej pary w stanie Bella $|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ [9]. (Należy oczywiście zwrócić uwagę, że nie są to wymagania trywialne – samo splątanie dwóch qubitów musiałoby być przeprowadzone wcześniej, w odpowiedniej izolacji itp).

Chcąc przesłać dwa klasyczne bity, Alicja przesyła do Boba jedną z czterech wartości: 0, 1, 2 lub 3. Aby wysłać 0, Alicja nie robi ze swoim qubitem nic (z innego punktu widzenia: wykonuje na nim operację identyczności *I*). Aby wysłać 1 – wykonuje na swoim qubicie operację X Pauliego. Aby wysłać 2 – wykonuje operację Z. Aby wysłać 3 – wykonuje Z•X (czyli najpierw wykonuje X, a następnie Z). Ponieważ qubity Alicji i Boba są splątane, powyższe przekształcenia należy zapisać jako operacje na parze qubitów:

Do wysłania	**Przekształcenie**	**Stan $|\beta_{00}\rangle$ przechodzi w:**
0d = 00b	$I \otimes I$	$\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle) = \|\beta_{00}\rangle$
1d = 01b	$X \otimes I$	$\frac{1}{\sqrt{2}}(\|01\rangle + \|10\rangle) = \|\beta_{01}\rangle$
2d = 10b	$Z \otimes I$	$\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle) = \|\beta_{10}\rangle$
3d = 11b	$Z \cdot X \otimes I$	$\frac{1}{\sqrt{2}}(\|01\rangle + \|10\rangle) = \|\beta_{11}\rangle$

Po wykonaniu stosownego przekształcenia Alicja przesyła do Boba swój qubit, dzięki czemu ma on dwa splątane qubity w jednym z czterech stanów Bella. Wystarczy więc, że Bob zbada wartość qubitów w bazie Bella aby otrzymał dwa klasyczne bity informacji.

2. Kwantowa teleportacja

Biorąc pod uwagę ogromną ilość informacji niesioną przez qubit (nie mówiąc o splątanej parze qubitów) może nie być aż tak dużym zaskoczeniem, że przesyłając pojedynczy qubitu możemy przesłać dwa klasyczne bity. Odwrotna operacja, czyli przesłanie informacji zawartej w qubicie za pomocą przesłania dwóch bitów klasycznych, wydaje się niemal niemożliwa. Skoro nadawca sam nie zna dokładnie informacji zawartej w qubicie (pomiar powoduje jej nieodwracalne zniszczenie), jak może ją przekazać odbiorcy nie

przesyłając całego qubita? Co więcej, nawet jeśli znalibyśmy informację reprezentowaną przez qubit, to liczba klasycznych bitów potrzebna do jej przesłania byłaby potencjalnie nieskończona.

Kwantowa teleportacja to właśnie sposób na dokonanie rzeczy pozornie niemożliwej. Z pomocą tego protokołu Alicja może sprawić aby Bob uzyskał stan qubita identyczny jak u niej przesyłając mu tylko dwa klasyczne bity. Innymi słowy, protokół kwantowej teleportacji umożliwia przesyłanie informacji kwantowej pomiędzy odbiorcami niepołączonymi kwantowym kanałem komunikacyjnym.

Podobnie jak w przypadku kodowania supergęstego, również kwantowa teleportacja wymaga aby zarówno nadawca (Alicja) jak i odbiorca (Bob) posiadali po jednym qubicie ze splątanej pary $|\beta_{00}\rangle$. Aby przesłać do Boba stan pewnego qubita $|v\rangle = \alpha|0\rangle + \beta|1\rangle$ Alicja musi przeprowadzić pomiar w dziedzinie Bella dla $|v\rangle$ oraz swojego qubita ze splątanej pary współdzielonej z Bobem. Tym samym otrzyma ona dwa klasyczne bity a i b, a trzy przetwarzane qubity po pomiarze znajdą się w jednym ze stanów (z prawdopodobieństwem $\frac{1}{4}$ dla każdego):

$$|\beta_{00}\rangle|v\rangle$$
$$|\beta_{01}\rangle(X|v\rangle)$$
$$|\beta_{10}\rangle(Z|v\rangle)$$
$$|\beta_{11}\rangle(XZ|v\rangle)$$

Mając te wiedzę Bob może wykonać odpowiednie operacje (w zależności od klasycznych bitów a i b) aby transformować jego stan do stanu qubita $|v\rangle$:

a	**b**	**Operacja**
0	0	I
0	1	X
1	0	Z
1	1	$Z \bullet X$

Tym samym wartość qubita Alicji została niejako teleportowana do Boba.

Poprawność powyższych zapisów można łatwo wykazać[9]. Pomiar w dziedzinie Bella odpowiada (pod względem zwracanych wartości klasycznych) wykonaniu $(CNOT \otimes I)$, a następnie $(H \otimes I)$ na produkcie tensorowym $|v\rangle \otimes \beta_{00}$. Powyższe przekształcenia doprowadzą nas do stanu opisanego wektorem:

$$\frac{1}{2}(\alpha|000\rangle + \beta|001\rangle + \beta|010\rangle + \alpha|011\rangle + \alpha|100\rangle - \beta|101\rangle - \beta|110\rangle + \alpha|111\rangle) =$$

$$\frac{1}{2}\big(|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)\big)$$

Po wykonaniu pomiaru qubitów w dziedzinie obliczeniowej stan pierwszych dwóch qubitów zostanie ustalony i tym samym trzeci qubit przyjmie stan reprezentowany przez jeden z poniższych zapisów:

$$(\alpha|0\rangle + \beta|1\rangle)$$

$$(\alpha|1\rangle + \beta|0\rangle)$$

$$(\alpha|0\rangle - \beta|1\rangle)$$

$$(\alpha|1\rangle - \beta|0\rangle)$$

Powyższe stany przekształcone odpowiednio przez *I*, *X*, *Z* lub *Z•X* zostaną sprowadzone do stanu $\alpha|0\rangle + \beta|1\rangle$, czyli do stanu qubita $|v\rangle$. Alicji istotnie udało się teleportować kwantowy stan do Boba przesyłając wyłącznie dwa klasyczne bity.

IV Algorytmy kwantowe

Celem poniższego rozdziału jest przedstawienie wykorzystania kwantowego modelu do prowadzenia realnych obliczeń, rozwiązujących konkretne problemy obliczeniowe. Dodatkowa wstępna część ma za zadanie porównać obliczenia probabilistyczne wykonywane na klasycznym modelu z ich kwantowymi odpowiednikami, aby ułatwić zrozumienie istotnych różnic pomiędzy tymi modelami. Pojawiają się również podrozdziały zawierające niezbędne podstawy teoretyczne pozwalające wprowadzić bardziej zaawansowane algorytmy kwantowe w dalszych częściach opracowania.

1. Obliczenia kwantowe a probabilistyczne

Z powodu często powtarzanego w kontekście obliczeń kwantowych słowa „niedeterminizm" można pomyśleć, że w istocie model kwantowy jest modelem probabilistycznym. Wrażenie to jest jednak mylne, ponieważ mechanika kwantowa dostarcza pewnych zachowań, których nie można zaobserwować dla probabilistycznych obliczeń klasycznych. Splątanie kwantowe, superpozycja czy zachowanie qubita wobec pomiaru jego stanu znacząco odbiegają od zjawisk znanych z klasycznego modelu probabilistycznego.

Różnicę pomiędzy wspomnianymi modelami można przedstawić na prostym przykładzie. Weźmy układ złożony z dwóch bramek Hadamarda uruchomiony na wejściu $|0\rangle$. Jeśli dokonamy pomiaru stanu po każdej z bramek, to po pierwszym odczycie stan systemu będzie równy $|0\rangle$ lub $|1\rangle$, każdy z prawdopodobieństwem $\frac{1}{2}$. Tym samym po drugiej bramce Hadamarda, tuż przed drugim pomiarem, stan układu będzie równy $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ lub $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, każde z prawdopodobieństwem $\frac{1}{2}$.

Jeżeli nie dokonamy odczytu po pierwszej bramce Hadamarda (czyli odczytamy wyłącznie stan końcowy systemu), to ostatecznie stan systemu można przedstawić jako:

$$|v_{końcowy}\rangle = H(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle) = \frac{1}{\sqrt{2}}H|0\rangle + \frac{1}{\sqrt{2}}H|1\rangle$$

$$= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) + \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)$$

$$= \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle + \frac{1}{2}|0\rangle - \frac{1}{2}|1\rangle = |0\rangle$$

Ujemna amplituda zniosła amplitudę dodatnią, co całkowicie zmieniło wynik obliczenia. Nawet tak prosty przykład ilustruje sytuację, w której dzięki interferencji i superpozycji system kwantowy zachowuje się zupełnie inaczej, niż system probabilistyczny.

2. Algorytm Deutscha

Poniższy algorytm zostaje wprowadzony w tym opracowaniu (podobnie jak protokoły kwantowe) ze względu na potencjalnie duży wkład w zrozumienie potencjału obliczeń kwantowych. Należy jednak zaznaczyć, że problem Deutscha nie przekłada się na zbyt wiele rzeczywistych problemów.

Daną wejściową w problemie Deutscha jest pewna funkcja *f(x)* odwzorowująca $\{0,1\} \rightarrow \{0,1\}$. Nie mamy wzoru *f(x)*, a jedyne co o niej wiemy, to że jest albo stała (zawsze zwraca 0 lub 1), albo nie (zwraca rezultat inny dla argumentu wejściowego 0, a inny dla 1). Chcemy sprawdzić, czy *f(x)* jest stała, czy nie.

Z punktu widzenia klasycznego modelu obliczeń potrzebujemy wywołać *f* dwukrotnie, z argumentem 0 oraz 1. Jeżeli *f(0) XOR f(1) = 0* to znaczy, że *f* jest stała (jeśli 1 – przeciwnie). Kwantowy model obliczeń udostępnia nam mechanizmy pozwalające zdobyć odpowiedź wywołując *f* tylko raz.

Zdefiniujmy kwantową bramkę D_f operującą na dwóch qubitach, której działanie można opisać za pomocą operatora unitarnego:

$$|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$$

gdzie $\oplus$ oznacza operację alternatywy wykluczającej (XOR). Tym samym jeśli ustawimy $|y\rangle = |0\rangle$, to dla pierwszego qubita $|x\rangle = |0\rangle$ bramka ustawi drugi qubit na $|0 \oplus f(0)\rangle = |f(0)\rangle$. W przeciwnym razie drugi qubit zostanie ustawiony na $|0 \oplus f(1)\rangle = |f(1)\rangle$.

Jeżeli ustawimy pierwszy z qubitów na superpozycję $|x\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, a drugi z qubitów pozostawimy w stanie $|y\rangle = |0\rangle$ to stanem wejściowym naszej bramki będzie $\frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|0\rangle$. Bramka D_f przekształci go następująco:

$$D_f\left(\frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|0\rangle\right) = \frac{1}{\sqrt{2}}|0\rangle|f(0)\rangle + \frac{1}{\sqrt{2}}|1\rangle|f(1)\rangle$$

Można powiedzieć, że w pewien sposób wartości *f(0)* i *f(1)* zostały obliczone jednocześnie. Nie ma to jednak znaczenia dopóki nie będziemy mogli ich odczytać (a jak wiadomo pomiar stanu kwantowego zniszczy ów stan i uniemożliwi odczytanie jednej z wartości).

Możemy skonstruować jednak prosty układ kwantowy, który umożliwi nam poprawną interpretację wyniku zwróconego przez bramkę D_f. Ustawmy qubit $|x\rangle$ początkowo na

$|0\rangle$, a qubit $|y\rangle$ na $|1\rangle$. Następnie na obu qubitach wykonajmy przekształcenie przez bramkę Hadamarda. Tak przekształcone qubity skierujmy do bramki D_f, a po wyjściu jeszcze raz wykonajmy przekształcenie H na qubicie $|x\rangle$. Pierwsze dwa kroki można zapisać w postaci obliczeń:

$$H|0\rangle H|1\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = |v_1\rangle$$

$$D_f|v_1\rangle = \left(\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$= (-1)^{f(0)}\left(\frac{|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = |v_2\rangle$$

Przekształcenie w drugiej linijce wynika z następujących zależności:

- Dla $f(x) = 0$ zachodzi $D_f|y\rangle = D_f\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = \left(\frac{|0\oplus f(x)\rangle - |1\oplus f(x)\rangle}{\sqrt{2}}\right) = \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$
- Dla $f(x) = 1$ zachodzi $D_f|y\rangle = D_f\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = \left(\frac{|0\oplus f(x)\rangle - |1\oplus f(x)\rangle}{\sqrt{2}}\right) = \left(\frac{|1\rangle - |0\rangle}{\sqrt{2}}\right)$

Stąd w ogólności można zapisać:

$$\frac{|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}} = (-1)^{f(x)}\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Powyższą równoważność przekształceń nazywamy *przerzuceniem fazy*.

Przerzucenie fazy

Zjawisko zachodzące dla kontrolowanej bramki kwantowej realizującej przekształcenie U uruchomionej na docelowym qubicie będącym w stanie reprezentującym wektor własny $|v\rangle$ przekształcenia U. Faza będąca wartością własną przekształcenia U może zostać przerzucona do qubitu kontrolnego.

W trzeciej linijce wykorzystywany jest fakt, że $(-1)^{f(0)}(-1)^{f(1)} = (-1)^{f(0)\oplus f(1)}$.

Po przekształceniach opisanych jak powyżej można wykonać trzeci, ostatni krok rozwiązania problemu Deutscha, wykonując przekształcenie przez bramkę Hadamarda na qubicie $|x\rangle$. Zwróćmy uwagę, że jeśli funkcja f jest stała, to $f(0) \oplus f(1)$ jest równe 0, a w przeciwnym wypadku 1. Zatem jeśli f jest stała to:

$$|v_2\rangle = (-1)^{f(0)}\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Tym samym bramka Hadamarda operująca na pierwszym qubicie przekształci stan systemu do postaci:

$$|v_{końcowy}\rangle = (-1)^{f(0)}|0\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Jeżeli *f* nie jest stała, to:

$$|v_2\rangle = (-1)^{f(0)}\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

W tym wypadku końcowy stan systemu po przekształceniu pierwszego qubita przez bramkę Hadamarda będzie prezentował się następująco:

$$|v_{końcowy}\rangle = (-1)^{f(0)}|1\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Tym samym jeśli funkcja *f* jest stała to mamy pewność, że odczytując wartość qubita $|x\rangle$ odczytamy $|0\rangle$. W przeciwnym wypadku odczytamy $|1\rangle$. Jak widać możemy rozwiązać problem Deutscha wywołując *f* jedynie raz.

3. Algorytm Deutscha – Jozsy

O ile poprzedni algorytm działa szybciej od jego klasycznej wersji o stałą liczbę wywołań pewnej funkcji *f*, o tyle jego modyfikacja daje już o wiele bardziej wyraźny zysk. Problem Deutscha – Jozsy (oraz rozwiązujący go algorytm) również polega na stwierdzeniu czy pewna funkcja *f* jest stała czy zbalansowana (dla połowy dziedziny daje wynik 1, a dla drugiej połowy – 0). Tym razem jednak funkcja *f* odwzorowuje $\{0,1\}^n \rightarrow \{0,1\}$, a więc dziedziną są wszystkie n-bitowe liczby binarne.

Aby sprawdzić czy *f* jest stała czy zbalansowana klasyczny algorytm deterministyczny musiałby w najgorszym wypadku wykonać funkcję *f* $2^{n-1} + 1$ razy. Kwantowy algorytm Deutscha – Jozsy umożliwia obliczenie wyniku za pomocą tylko *jednego* wywołania funkcji *f*.

Układ bramek kwantowych obliczający rozwiązanie problemu Deutscha – Jozsy jest analogiczny do tego rozwiązującego problem Deutscha: jego kluczowym elementem jest pewna bramka DJ_f. Jednak w tej wersji bramka ta nie jest sterowana pojedynczym qubitem kontrolnym $|x\rangle$, lecz *n-elementowym rejestrem qubitów*, który będziemy oznaczać jako $|\boldsymbol{x}\rangle$.

Wejście do naszego układu stanowi n qubitów kontrolnych, każdy ustawiony na $|0\rangle$, (czyli możemy zapisać $|\boldsymbol{x}\rangle = |0\rangle^{\otimes n}$), oraz qubit $|y\rangle = |1\rangle$. Pierwszym krokiem jest przekształcenie wszystkich qubitów wejściowych przez bramki Hadamarda. Tym samym dane wejściowe po pierwszej fazie algorytmu są następującej postaci:

$$|v_1\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|\boldsymbol{x}\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Wykorzystujemy tu mechanizm dostępny jedynie dla obliczeń kwantowych – rejestr $|\boldsymbol{x}\rangle$ znajduje się w superpozycji wszystkich możliwych n-bitowych ciągów binarnych.

Drugi krok algorytmu polega na uruchomieniu bramki DJ_f na qubicie $|y\rangle$. Stan układu po tym przekształceniu prezentuje się następująco:

$$|v_2\rangle = \frac{1}{\sqrt{2^n}} DJ_f \left(\sum_{x\in\{0,1\}^n} |\boldsymbol{x}\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \right) = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |\boldsymbol{x}\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

Podobnie jak w algorytmie Deutscha powyższy zapis zawiera tzw. przerzucenie fazy.

Trzeci i ostatni krok algorytmu Deutscha – Jozsy polega na przekształceniu wszystkich qubitów rejestru $|\boldsymbol{x}\rangle$ przez bramki H. Aby znacząco ułatwić zapis wprowadzimy następujące równości dla przekształceń przez bramki Hadamarda:

$$H|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^x|1\rangle) = \frac{1}{\sqrt{2}} \sum_{z\in\{0,1\}} (-1)^{xz}|z\rangle$$

$$H^{\otimes n}|\boldsymbol{x}\rangle = H|x_1\rangle H|x_2\rangle \dots H|x_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} (-1)^{\boldsymbol{xz}}|\boldsymbol{z}\rangle$$

Wykorzystując powyższy zapis, stan układu po trzecim kroku algorytmu możemy przedstawić następująco:

$$|v_3\rangle = \left(\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} (-1)^{\boldsymbol{xz}}|\boldsymbol{z}\rangle \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

$$= \frac{1}{2^n} \sum_{z\in\{0,1\}^n} \left(\sum_{x\in\{0,1\}^n} (-1)^{f(x)+\boldsymbol{xz}} \right) |\boldsymbol{z}\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

Aby odczytać wynik, badamy rejestr qubitów $|\boldsymbol{z}\rangle$ (czyli ten sam, który początkowo oznaczaliśmy $|\boldsymbol{x}\rangle$). Należy zwrócić uwagę, że amplituda dla stanu $|\boldsymbol{z}\rangle = |0\rangle^{\otimes n}$ jest postaci:

$$\frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)}$$

Jeżeli *f* jest funkcją stałą, to amplituda ta przyjmie jedną z dwóch wartości:

- $\frac{(-1)\times 2^n}{2^n} = (-1)$ dla *f(x) = 1*
- $\frac{(-1)\times 2^n}{2^n} = 1$ dla *f(x) = 0*

W przeciwnym razie *f* jest funkcją zbalansowaną, czyli amplitudy zniosą się wzajemnie (w sumie dla wszystkich wartości $x \in \{0,1\}^n$ wystąpi tyle samo liczb 1 co -1), a tym samym amplituda rejestru $|\boldsymbol{z}\rangle = |0\rangle^{\otimes n}$ wyniesie 0, czyli $|\boldsymbol{z}\rangle \neq |0\rangle^{\otimes n}$ (w rejestrze tym występuje przynajmniej jedna jedynka).

Mamy zatem gwarancję: jeśli *f* jest funkcją stałą to odczytując rejestr qubitów odczytamy same wartości 0. W przeciwnym razie odczytamy przynajmniej jedną jedynkę. Jesteśmy zatem w stanie określić charakter funkcji *f* wywołując ją tylko jeden raz. Algorytm kwantowy działa dla tego problemu w czasie $O(1)$, a klasyczny algorytm deterministyczny – $O(2^{n-1})$ (rozpatrując liczbę wywołań *f* jako główny składnik kosztu obliczeń).

4. Algorytm Simona

Algorytm Simona skonstruowany został w celu rozwiązania następującego problemu: mając pewną funkcję f przekształcającą $\{0,1\}^n \to X$ (gdzie X jest pewnym skończonym zbiorem) i wiedząc, że istnieje pewien ciąg symboli **s** = $s_1 s_2 \ldots s_n$ taki, że $f(x) = f(y)$ wtedy i tylko wtedy gdy $x = y$ lub $x = y \oplus \boldsymbol{s}$, należy znaleźć **s**. Dziedzinę funkcji f będziemy rozpatrywać jako przestrzeń wektorową Z_2^n nad Z_2. Przyjmujemy również, że $X \subseteq \{0,1\}^n$ (co jest dość rozsądnym założeniem z punktu widzenia informatyki).

Dowolny klasyczny algorytm (zarówno deterministyczny jak i probabilistyczny) potrzebuje czasu co najmniej wykładniczego względem n (długości **s**) aby znaleźć rozwiązanie. Algorytm Simona rozwiązuje powyższy problem przy liniowej względem n oczekiwanej liczbie wywołań f oraz oczekiwanym użyciu $O(n^3)$ elementarnych bramek kwantowych.

W podrozdziale IV-3 (o algorytmie Deutscha – Jozsy) zostało pokazane jak n bramek Hadamarda przekształca rejestr n qubitów będących w jednym ze stanów bazowych. Dla stanów będących superpozycją (o równych amplitudach) dwóch stanów bazowych n-qubitowa transformacja Hadamarda może być przedstawiona następująco:

$$H^{\otimes n}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|s\rangle\right) = \frac{1}{\sqrt{2^{n+1}}} \sum_{z\in\{0,1\}^n} |\boldsymbol{z}\rangle + \sum_{z\in\{0,1\}^n} (-1)^{\boldsymbol{sz}} |\boldsymbol{z}\rangle$$

$$= \frac{1}{\sqrt{2^{n+1}}} \sum_{z\in\{0,1\}^n} (1 + (-1)^{\boldsymbol{sz}})|\boldsymbol{z}\rangle$$

Zdefiniujmy zbiór $S^{\perp} = \{\boldsymbol{z} \in \{0,1\}^n | \boldsymbol{sz} = 0\}$. Zauważmy, że $S^{\perp}$ jest podprzestrzenią wektorową Z_2^n. Zauważmy również, że jeżeli $\boldsymbol{sz} = 1$, to **z** znika z równania, a w przeciwnym wypadku pozostaje z amplitudą $\frac{1}{\sqrt{2^{n+1}}} \times 2 = \frac{1}{\sqrt{2^{n-1}}}$. Z tą definicją możemy przepisać wcześniejszy zapis transformacji Hadamarda:

$$H^{\otimes n}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|s\rangle\right) = \frac{1}{\sqrt{2^{n-1}}} \sum_{z\in\{s\}^{\perp}} |\boldsymbol{z}\rangle$$

Co więcej, przyjmując $a, b \in \{0,1\}^n$, $s = a \oplus b$ możemy zapisać:

$$H^{\otimes n}\left(\frac{1}{\sqrt{2}}|a\rangle + \frac{1}{\sqrt{2}}|b\rangle\right) = H^{\otimes n}\left(\frac{1}{\sqrt{2}}|a\rangle + \frac{1}{\sqrt{2}}|a \oplus s\rangle\right) = \frac{1}{\sqrt{2^{n-1}}} \sum_{z\in\{s\}^{\perp}} (-1)^{\boldsymbol{az}} |\boldsymbol{z}\rangle$$

Przyjmijmy, że mamy pewną odwracalną bramkę $S_f: |\boldsymbol{x}\rangle|\boldsymbol{y}\rangle \to |\boldsymbol{x}\rangle|\boldsymbol{y} \oplus f(x)\rangle$. Układ potrzebny do wykonania kwantowej części algorytmu Simona przekształca rejestr wejściowy $|\boldsymbol{x}\rangle$ przez bramki H, następnie bramka S_f operuje na $(H^{\otimes n}|\boldsymbol{x}\rangle)|\boldsymbol{0}\rangle$ po czym rejestr $|\boldsymbol{x}\rangle$ jest ponownie przekształcany przez bramki Hadamarda. Na końcu następuje pomiar $|\boldsymbol{x}\rangle$.

Pseudokod algorytmu Simona wygląda następująco[3]:

1. Licznik i = 1
2. Przygotuj stan wejściowy $\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle|\mathbf{0}\rangle$ (czyli przekształć $|\boldsymbol{x}\rangle = |0\rangle^{\otimes n}$ przez $H^{\otimes n}$).
3. Wykonaj operację bramki S_f - stan układu zostanie przekształcony na $\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle|f(x)\rangle$.
4. Przekształć pierwszy rejestr przez $H^{\otimes n}$.
5. Wykonaj pomiar pierwszego rejestru i zapisz otrzymany wynik w_i.
6. Jeśli wymiar podprzestrzeni liniowej rozpiętej przez $\{w_i\}$ jest równy n – 1 to przejdź do kroku 7. W przeciwnym razie zwiększ licznik i, a następnie idź do kroku drugiego.
7. Rozwiąż układ równań $Ws^T = \mathbf{0}^T$.
8. Zwróć niezerowe rozwiązanie układu równań.

W celu analizy powyższego algorytmu zauważmy, że zbiór $\{0,1\}^n$ można podzielić na 2^{n-1} par słów postaci $\{x, x \oplus s\}$ (dla każdego z 2^n wyrazów określamy jego „bliźniaka" poprzez wykonanie operacji XOR ze słowem s – stąd par będzie dwa razy mniej niż wszystkich wyrazów). Przyjmijmy pewien zbiór I będący podzbiorem $\{0,1\}^n$ zawierający po jednym reprezentancie z każdej pary $\{x, x \oplus s\}$. Stan systemu po przekształceniu przez bramkę S_f możemy zapisać jako[12]:

$$\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle|f(x)\rangle = \frac{1}{\sqrt{2^{n-1}}}\sum_{x\in I}\frac{1}{\sqrt{2}}(|x\rangle + |x\oplus s\rangle)|f(x)\rangle$$

Powyższa równość jest prawdziwa, ponieważ fakt, że z jednakowym prawdopodobieństwem może zostać odczytana każda z 2^n wartości jest równoważny temu, że z jednakowym prawdopodobieństwem może zostać odczytana każda z dwóch wartości z jednej z 2^{n-1} par.

Z wcześniejszego zapisu z tego podrozdziału wiemy, że:

$$H^{\otimes n}(\frac{1}{\sqrt{2}}|x\rangle + \frac{1}{\sqrt{2}}|x\oplus s\rangle) = \frac{1}{\sqrt{2^{n-1}}}\sum_{z\in\{s\}^\perp}(-1)^{\boldsymbol{xz}}|\boldsymbol{z}\rangle$$

Czyli powyższa transformacja sprawia, że pierwszy rejestr jest w superpozycji wszystkich stanów z $s^\perp$ o równoważnych amplitudach. Tym samym wartości s_i odczytywane w 5. kroku algorytmu są w istocie losowo wybranymi wektorami z $s^\perp$. Potrzebujemy n – 1 niezależnych liniowo wektorów z $s^\perp$ aby móc poprawnie określić **s** (dlatego sprawdzamy wymiar podprzestrzeni liniowej rozpinanej przez znalezione wektory).

Innymi słowy, po powyższej transformacji wiemy, że pierwszy rejestr zawiera superpozycję wszystkich wektorów **z** takich, że $\boldsymbol{zs} = 0$. Mając n – 1 różnych wektorów **z** (czyli każdy różni się przynajmniej na jednej pozycji od pozostałych) możemy określić każdą z pozycji (tu: każdy z bitów) wektora **s**, co robimy w krokach 7. i 8. (Odrzucamy oczywiście trywialne rozwiązanie **0**).

Rozwiązanie układu n – 1 równań liniowych można obliczyć w czasie liniowym względem n (korzystając np. z eliminacji Gaussa). Do znalezienia n liniowo niezależnych wektorów

z potrzebujemy n – 1 wywołań f (przy założeniu, że za każdym razem odczytamy inną z możliwych wartości **z**). Tym samym oczekiwana liczba wywołań funkcji f wynosi n – 1.

5. Określanie fazy stanu kwantowego

Z punktu widzenia wielu algorytmów kwantowych (a szczególnie tych opartych o kwantową transformatę Fouriera) istotna jest możliwość określenia lub przybliżenia fazy qubitu. Wynika to z faktu, że niejednokrotnie podczas obliczeń kwantowych istotne informacje zostają niejako *zakodowane w relatywnej fazie* stanu kwantowego.

Podstawowym koderem i dekoderem informacji zapisanej w fazie stanu jest bramka Hadamarda. Spójrzmy na jej działanie z nieco innej niż dotychczas strony, dla pewnego stanu bazowego $|x\rangle, x \in \{0,1\}$:

$$H|x\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{(-1)^x}{\sqrt{2}}|1\rangle = \frac{1}{\sqrt{2}}\sum_{y\in\{0,1\}}(-1)^{xy}|y\rangle$$

Można zauważyć, że bramka H niejako *koduje informacje* o $|x\rangle$ w relatywnej fazie między stanami podstawowymi $|0\rangle$ i $|1\rangle$. Ponieważ bramka H jest jednocześnie swoją odwrotnością, możemy również z jej pomocą odkodować informacje zapisane w relatywnej fazie:

$$H(\frac{1}{\sqrt{2}}|0\rangle + \frac{(-1)^x}{\sqrt{2}}|1\rangle) = |x\rangle$$

Co więcej, możemy uogólnić powyższe stwierdzenia na n-qubitowe stany kwantowe. n-qubitowa bramka Hadamarda działa na rejestrze n-qubitów $|\boldsymbol{x}\rangle$ w następujący sposób:

$$H^{\otimes n}|\boldsymbol{x}\rangle = \frac{1}{\sqrt{2^n}}\sum_{\boldsymbol{y}\in\{0,1\}^n}(-1)^{\boldsymbol{xy}}|\boldsymbol{y}\rangle$$

Również to przekształcenie możemy rozpatrywać jako zakodowanie informacji o stanie z rejestru $|\boldsymbol{x}\rangle$ w relatywnych fazach $(-1)^{\boldsymbol{xy}}$ stanów bazowych $|\boldsymbol{y}\rangle$. Oczywiście ponowne przekształcenie przez tą samą bramkę Hadamarda przywróci stan pierwotny (czyli niejako odkoduje informacje zapisane w relatywnych fazach):

$$H^{\otimes n}\left(\frac{1}{\sqrt{2^n}}\sum_{\boldsymbol{y}\in\{0,1\}^n}(-1)^{\boldsymbol{xy}}|\boldsymbol{y}\rangle\right) = H^{\otimes n}(H^{\otimes n}|\boldsymbol{x}\rangle) = H^{\otimes n}H^{\otimes n}|\boldsymbol{x}\rangle = I|\boldsymbol{x}\rangle = |\boldsymbol{x}\rangle$$

Fazy postaci $(-1)^{xy}$ są oczywiście przypadkami szczególnymi. W ogólności potrzebujemy bardziej precyzyjnego opisu:

Relatywna faza stanu kwantowego

Relatywna faza stanu kwantowego jest liczbą zespoloną postaci $e^{2\pi i\omega}$ dla dowolnej liczby rzeczywistej $\omega \in (0,1)$.

Jak można zauważyć, bramka Hadamarda nie jest w stanie odkodować bardziej skomplikowanej informacji, zapisanej w zespolonej relatywnej fazie. Istnieje jednak mechanizm określania fazy, który jest pewnego rodzaju rozszerzeniem podstawowego dekodera/enkodera w postaci bramki H.

Załóżmy, że z pewnego powodu chcemy poznać relatywną fazę ω pewnego stanu kwantowego zapisanego następująco:

$$\frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1} e^{2\pi i\omega y}|y\rangle$$

gdzie $|y\rangle$ traktujemy w tym wypadku jako liczbę całkowitą z przedziału $[0, 2^n)$. Zauważmy, że ω można zapisać binarnie jako:

$$\omega = 0.x_1x_2x_3\ldots$$

Przy powyższym zapisie łatwo zauważyć:

$$2^k\omega = x_1x_2\ldots x_k.x_{k+1}x_{k+2}\ldots$$

Wiadomo również, że dla dowolnej liczby całkowitej k zachodzi $e^{2\pi ik} = 1$, dzięki czemu możemy zapisać:

$$e^{2\pi i(2^k\omega)} = e^{2\pi i(x_1x_2\ldots x_k)}e^{2\pi i(0.x_{k+1}x_{k+2}\ldots)} = e^{2\pi i(0.x_{k+1}x_{k+2}\ldots)}$$

Dla jedno-qubitowego stanu kwantowego o pewnej relatywnej fazie $\omega = 0.x_1$ poprawny jest zapis (korzystający z formy zapisanej powyżej):

$$\frac{1}{\sqrt{2}}\sum_{y=0}^{1} e^{2\pi i(0.x_1)y}|y\rangle = \frac{1}{\sqrt{2}}\sum_{y=0}^{1} e^{2\pi i\left(\frac{x_1}{2}\right)y}|y\rangle = \frac{1}{\sqrt{2}}\sum_{y=0}^{1} e^{\pi i(x_1y)}|y\rangle = \frac{1}{\sqrt{2}}\sum_{y=0}^{1}(-1)^{x_1y}|y\rangle$$

$$= \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_1}|1\rangle)$$

Jak pokazaliśmy wcześniej, x_1 możemy odczytać za pomocą bramki Hadamarda:

$$H(\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_1}|1\rangle)) = |x\rangle$$

Ponieważ $\omega = 0.x_1$ to odczytując x_1 odczytaliśmy ω. Jednak jak odczytać fazę jeżeli jest ona bardziej skomplikowana?

Przedstawmy pewną użyteczną równość:

$$\frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1} e^{2\pi i\omega y}|y\rangle =$$

$$= \left(\frac{|0\rangle + e^{2\pi i\left(2^{n-1}\omega\right)}|1\rangle}{\sqrt{2}}\right)\otimes\left(\frac{|0\rangle + e^{2\pi i\left(2^{n-2}\omega\right)}|1\rangle}{\sqrt{2}}\right)\otimes\ldots\otimes\left(\frac{|0\rangle + e^{2\pi i(\omega)}|1\rangle}{\sqrt{2}}\right) =$$

$$= \left(\frac{|0\rangle + e^{2\pi i(0.x_nx_{n+1}\ldots)}|1\rangle}{\sqrt{2}}\right)\otimes\left(\frac{|0\rangle + e^{2\pi i(0.x_{n-1}x_n\ldots)}|1\rangle}{\sqrt{2}}\right)\otimes\ldots\otimes\left(\frac{|0\rangle + e^{2\pi i(0.x_1x_2\ldots)}|1\rangle}{\sqrt{2}}\right)$$

Weźmy pewien dwu-qubitowy stan $\frac{1}{\sqrt{2^2}}\sum_{y=0}^{2^2-1} e^{2\pi i\omega y}|y\rangle$. Tym razem przyjmijmy, że $\omega = 0.x_1x_2$. Korzystając z równości opisanej powyżej możemy zapisać podany stan jako:

$$\frac{1}{\sqrt{2^2}}\sum_{y=0}^{2^2-1} e^{2\pi i(0.x_1x_2)y}|y\rangle = \left(\frac{|0\rangle + e^{2\pi i(0.x_2)}|1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + e^{2\pi i(0.x_1x_2)}|1\rangle}{\sqrt{2}}\right)$$

Jak widać, x_2 można zbadać od razu, przekształcając pierwszy qubit przez bramkę H i dokonując pomiaru. Jeśli odczytana wartość wynosi 0, to aby odczytać x_1 wystarczy przekształcić drugi z qubitów przez bramkę Hadamarda i również dokonać pomiaru. W przeciwnym razie stan drugiego qubita można zapisać jako $\frac{|0\rangle+e^{2\pi i(0.x_1 1)}|1\rangle}{\sqrt{2}}$. Aby odczytać wartość x_1 musimy przekształcić ten stan do postaci, która nam to umożliwi.

W tym celu zdefiniujmy jedno-qubitową bramkę rotacji fazy:

Jedno-qubitowa bramka rotacji fazy

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{bmatrix}$$

gdzie *k* oznacza pozycję 2^{-k} fazy ω qubitu.

W naszym przypadku interesuje nas „uwolnienie" x_1 od x_2, które znajduje się na 2. pozycji po przecinku. W tym celu możemy wykorzystać bramkę R_2, a dokładniej – jej odwrotność:

$$R_2 = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i(0.01)} \end{bmatrix}$$

$$R_2^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & e^{-2\pi i(0.01)} \end{bmatrix}$$

Zaobserwujmy teraz efekt przekształcenia drugiego qubita z naszego przykładu przez R_2^{-1}:

$$R_2^{-1}\left(\frac{|0\rangle + e^{2\pi i(0.x_1 1)}|1\rangle}{\sqrt{2}}\right) = \frac{|0\rangle + e^{2\pi i(0.x_1 1-0.01)}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{2\pi i(0.x_1)}|1\rangle}{\sqrt{2}}$$

Tym samym otrzymaliśmy stan, z którego po przekształceniu przez bramkę Hadamarda odczytamy x_1. Mając wartości x_1 i x_2 znamy $\omega = 0.x_1x_2$, czyli odczytaliśmy relatywną fazę danego dwu-qubitowego stanu.

Zauważmy, że rotacja fazy jest wymagana tylko wtedy, kiedy pierwszy z qubitów ($|x_2\rangle$) ma wartość 1. Tym samym nasz układ mógłby zawierać kontrolowaną bamkę R_2^{-1}, gdzie pierwszy qubit jest qubitem kontrolnym, a na końcu każdy z qubitów przekształcany byłby przez bramkę Hadamarda i odczytywany.

Powyższe rozwiązanie stosunkowo łatwo skaluje się w górę. Przykładowo, jeżeli dołożymy do naszego systemu kolejny, trzeci qubit, to jego stan postaci $\frac{|0\rangle+e^{2\pi i(0.x_1x_2x_3)}|1\rangle}{\sqrt{2}}$ możemy przekształcić przez bramki R_3^{-1} kontrolowaną przez pierwszy qubit oraz R_2^{-1} kontrolowaną przez drugi qubit. Postępowanie wobec dwóch pierwszych qubitów jest analogiczne do dwu-qubitowego układu, a ostatecznie wszystkie qubity są przekształcane („rozkodowywane") przez bramki Hadamarda.

Zauważmy, że podany algorytm określa fazę ω postaci $0.x_1x_2\ldots$. Można jednak udowodnić, że dla faz ω niedających się zapisać w powyższej postaci algorytm określania fazy stanu kwantowego zwraca oszacowanie x_{approx} spełniające:

$$\left|\frac{x_{approx}}{2^n} - \omega\right| \leq \frac{1}{2^n}$$

z prawdopodobieństwem wynoszącym przynajmniej $\frac{8}{\pi^2}$.

6. Kwantowa transformata Fouriera

Układ realizujący algorytm opisany w poprzednim (tj. IV-5) podrozdziale pozwala na określenie relatywnej fazy $\omega = 0.x_1x_2\ldots x_n$ stanu kwantowego postaci $\frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1} e^{2\pi i\omega y}|y\rangle$. Innymi słowy, wymaga on aby ω była postaci $\frac{x}{2^n}$ dla pewnej całkowitej liczby x (a dla ω niemożliwej do zapisania w ten sposób algorytm określania fazy daje najbliższe przybliżenie z wysokim prawdopodobieństwem). Tym samym układ ten realizuje przekształcenie:

$$\frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1} e^{2\pi i(\frac{x}{2^n})y}|y\rangle \rightarrow |x\rangle$$

Oczywiście qubity wyjściowe są w odwrotnej kolejności (tj. pierwszy z qubitów przechowuje wynikową wartość ostatniego, drugi – przedostatniego itd.), jednak jest to wyłącznie kwestia indeksacji.

Przytoczmy ogólny zapis przekształcenia dyskretnej transformaty Fouriera[10, 11]:

$$X_k = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} e^{-2\pi i\frac{k}{N}n}x_n$$

Teraz rozważmy przekształcenie realizowane przez odwrotność układu określania fazy (możemy to zrobić ponieważ układy kwantowe z założenia są odwracalne):

Kwantowa transformata Fouriera – QFT

$$QFT_N\colon |x\rangle \rightarrow \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1} e^{2\pi i\frac{x}{N}y}|y\rangle$$

Podobieństwo nie jest przypadkowe – drugie z powyższych przekształceń nazywamy kwantową transformatą Fouriera (*QFT – Quantum Fourier Transform*) na jednym z N stanów bazowych (w zapisie powyżej zachodzi $N = 2^n$).

Ponieważ wiemy (z podrozdziału IV-5) jak zbudować układ realizujący badanie fazy stanu, potrafimy też zbudować jego odwrotność poprzez zamianę bramek na ich odwrotności i propagację układu wstecz. Tym samym mamy prosty sposób obliczania QFT dla dowolnego stanu kwantowego będącego superpozycją stanów bazowych.

QFT można również przedstawić jako macierz unitarną następującej postaci:

$$\frac{1}{\sqrt{N}}\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \dots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)(N-1)} \end{bmatrix};\ \omega = e^{\frac{2\pi i}{N}}$$

Warto również zaznaczyć, że z unitarności QFT wynika, że łatwo możemy policzyć również jej odwrotność (poprzez operator sprzężony).

Odwrotna kwantowa transformata Fouriera – QFT

$$QFT_N^{-1}: |x\rangle \to \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{-2\pi i \frac{x}{N} y} |y\rangle$$

7. Znajdowanie okresu stanu okresowego

Poniższy problem również nie wydaje się mieć istotnego związku z rzeczywistością, jednak stanowi on ważny krok wprowadzający do jednego z najbardziej praktycznych algorytmów kwantowych – algorytmu faktoryzacji liczb całkowitych, a także pokazuje zastosowanie kwantowej transformaty Fouriera.

Stan okresowy to stan kwantowy postaci:

$$|\phi_{r,b}\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |xr + b\rangle$$

gdzie r jest okresem, N – liczbą powtórzeń okresu, b – przesunięciem wybranym losowo z $\{0, 1, \dots, r-1\}$. Problem odnalezienia okresu polega na podaniu r mając dane Nr oraz maszynę generującą okresowe stany kwantowe o pewnych (nieznanych) parametrach.

Ponieważ nie znamy żadnego z parametrów zwykły pomiar stanu zwróci nam jedynie pewien losowy stan z przedziału $\{0, 1, \dots, Nr-1\}$, co nie niesie żadnej informacji o r. Z pomocą przychodzi nam jednak kwantowa transformacja Fouriera (a w zasadzie jej odwrotność). Można wykazać, że zachodzi:

$$QFT_{Nr}^{-1} |\phi_{r,b}\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i \frac{b}{r} k} |mk\rangle$$

Tym samym po przekształceniu wygenerowanego przez maszynę stanu okresowego przez odwrotną QFT otrzymamy stan, którego pomiar da nam pewną liczbę mk dla pewnego losowego $k \in \{0, 1, \dots, r-1\}$. Pozwala nam to obliczyć $\frac{mk}{mr} = \frac{k}{r}$ i wyrazić je za pomocą jak najmniejszych liczb, a tym samym – poznać r.

Jednym z efektywnych sposobów sprowadzania $\frac{k}{r}$ do jak najprostszej postaci jest wykorzystanie rozszerzonego algorytmu Euklidesa. Pozwala on efektywnie obliczać

największy wspólny dzielnik *k* i *r* (a najprostsza forma $\frac{k}{r}$ jest postaci $\frac{\frac{k}{NWD(k,r)}}{\frac{r}{NWD(k,r)}}$). Dodatkowo prawdziwe jest następujące twierdzenie:

> Niech *r* będzie pewną dodatnią liczbą całkowitą, a k_1 oraz k_2 dwoma niezależnymi liczbami wybranymi losowo z $\{0,1,\dots,r-1\}$. Niech c_1, c_2, r_1, r_2 będą pewnymi dodatnimi liczbami całkowitymi spełniającymi $\mathrm{NWD}(r_1, c_1) = \mathrm{NWD}(r_2, c_2) = 1$ oraz $\frac{k_1}{r} = \frac{c_1}{r_1}$ i $\frac{k_2}{r} = \frac{c_2}{r_2}$.
>
> Wtedy z prawdopodobieństwem $\frac{6}{\pi^2}$ zachodzi $r = NWW(r_1, r_2)$.

Innymi słowy, dwukrotne odczytanie wartości $\frac{k}{r}$ oraz wykorzystanie rozszerzonego algorytm Euklidesa sprawia, że możemy z wysokim prawdopodobieństwem wydajnie obliczyć *r*.

Istnieje pewne ryzyko, że *k* oraz *r* posiadają pewien wspólny dzielnik większy od 1. Tym samym podczas sprowadzania ułamka do postaci o najmniejszych liczniku i mianowniku podzielimy zarówno *k* jak i *r* przez ów dzielnik, a tym samym odczytamy niewłaściwą wartość *r*. Jednak prawdopodobieństwo względnej pierwszości dwóch liczb naturalnych[13] wynosi $\frac{6}{\pi^2} \approx 61\%$, więc oczekiwana liczba wykonań procedury potrzebna do odczytania *r* wynosi $\frac{2r}{5}$. Należy również zaznaczyć, że istnieją proste klasyczne metody pozwalające sprawdzić poprawność otrzymanego okresu *r* (zostaną one podane przy konkretnych przykładach wykorzystania algorytmu znajdowania okresu stanu okresowego).

Istnieje również nieco bardziej ogólna wersja opisywanego problemu, w której *mr* nie jest jedną z danych, a stany okresowe generowane przez maszynę są postaci $|\phi_{r,b}\rangle = \frac{1}{\sqrt{m_b}} \sum_{x:0 \le xr+b \le 2^n} |xr + b\rangle$, gdzie *n* jest dane, a $m_b \approx \frac{2^n}{r}$ gwarantuje normę stanu równą 1. W ramach omówienia podamy bez dowodu dwa pomocne twierdzenia: niech *x* oznacza wynik pomiaru $QFT_{2^n}^{-1}|\phi_{r,b}\rangle$. Można udowodnić, że dla każdego *x* spełniającego $\left|\frac{x}{2^n} - \frac{k}{r}\right| \le \frac{1}{2m_b r}$ dla pewnego *k* prawdopodobieństwo pomiaru *x* wynosi przynajmniej $\frac{m_b}{\pi^2 2^{n-2}}$. Można również dowieść, że jeżeli pomiar zwróci jedno z dwóch najdokładniejszych oszacowań $\frac{k}{r}$ oraz $2^n \ge 2r^2$ to można określić $\frac{k}{r}$.

8. Określanie wartości własnej przekształcenia

Podobnie jak w przypadku określania fazy stanu kwantowego, możliwość zbadania wartości własnej przekształcenia implementowanego przez układ kwantowy odgrywa istotną rolę w wielu algorytmach.

Zadaniem algorytmu określania wartości własnej przekształcenia jest jak najlepsze przybliżenie wartości ω mając dane operator *U*, wektor własny $|v\rangle$ oraz wartość własną postaci $e^{2\pi i\omega}$.

Jak pokazaliśmy we wcześniejszych podrozdziałach, rozpatrując kontrolowane bramki kwantowe uruchamiane na ich wektorach własnych możemy zaobserwować tzw. przerzucenie fazy (modyfikację fazy kontrolnego qubita w wyniku przekształcenia), które interpretowaliśmy też jako kodowanie informacji o stanie w fazach qubitów. W poniższym rozdziale zaprezentowane zostanie jeszcze jedno spojrzenie na relatywną fazę stanu.

Rozważmy operator unitarny U, o wektorze własnym $|v\rangle$ i wartości własnej $e^{2\pi i\omega}$, dla którego mamy efektywnie implementującą go bramkę kwantową. Kontrolowana wersja owej bramki może być rozpatrywana jako następująca operacja:

$$controlled - U|1\rangle|v\rangle = |1\rangle U|v\rangle = |1\rangle e^{2\pi i\omega}|v\rangle = e^{2\pi i\omega}|1\rangle|v\rangle$$

$$controlled - U|0\rangle|v\rangle = |0\rangle|v\rangle$$

Jeżeli bramka ta zostałaby uruchomiona na superpozycji $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ to efekt jej działania byłby następujący:

$$controlled - U(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle)|v\rangle = (\frac{1}{\sqrt{2}}|0\rangle + \frac{e^{2\pi i\omega}}{\sqrt{2}}|1\rangle)|v\rangle$$

Tym samym informacja o wartości własnej przekształcenia została zakodowana w relatywnej fazie qubitu kontrolnego. Zwróćmy również uwagę, że dla przekształcenia U^n, gdzie n jest dodatnią liczbą całkowitą, wektor własny pozostaje bez zmian ($|v\rangle$), a wartość własna jest postaci $\left(e^{2\pi i\omega}\right)^n = e^{2\pi in\omega}$. Tym samym zachodzi:

$$controlled - U^n(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle)|v\rangle = (\frac{1}{\sqrt{2}}|0\rangle + \frac{e^{2\pi in\omega}}{\sqrt{2}}|1\rangle)|v\rangle$$

Produkt tensorowy stanów wynikowych z powyższego przekształcenia dla $n = 2^x$gdzie $x \in \{0,1,\dots,X-1\}$ wygląda następująco:

$$\left(\frac{|0\rangle + e^{2\pi i(2^{X-1}\omega)}|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle + e^{2\pi i(2^{X-2}\omega)}|1\rangle}{\sqrt{2}}\right)\dots\left(\frac{|0\rangle + e^{2\pi i\omega}|1\rangle}{\sqrt{2}}\right) = \frac{1}{\sqrt{2^X}}\sum_{y=0}^{2^X-1} e^{2\pi i\omega y}|y\rangle$$

Jak wiadomo z podrozdziałów IV-5 i IV-6 (o określaniu fazy stanu kwantowego i kwantowej transformacie Fouriera) przekształcenie powyższego stanu przez QFT^{-1} spowoduje powstanie stanu, którego pomiar zwróci wartość którą możemy interpretować jako binarną liczbę całkowitą. Jeżeli oznaczymy zmierzoną wartość jako x to nasze oszacowanie wynosi $\omega_{approx} = \frac{x}{2^X}$. Paradoksalnie zawartość $|v\rangle$ praktycznie nas nie interesuje (z punktu widzenia opisywanego algorytmu może zostać odrzucona po przekształceniach).

Zamiast używać X kontrolowanych bramek U^{2^x} do utworzenia pożądanego przez nas stanu, można zbudować pojedynczą bramkę kontrolowaną U^X operującą na X qubitach (wykonującą przekształcenie $U^{2^{X-n-1}}$ na n-tym qubicie, dla $n \in \{0,1,\dots X-1\}$). X bramek

Hadamarda potrzebnych do utworzenia stanu wejściowego pożądanej postaci (X qubitów w stanie $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$) można zastąpić przez QFT (łatwo pokazać, że $QFT|0\rangle^{\otimes X} = \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)^{\otimes X}$).

Cała procedura określania wartości własnej wygląda zatem następująco:

1. Ustaw X-qubitowy rejestr kontrolny na stan $|0\rangle^{\otimes X}$, oraz przygotuj rejestr reprezentujący dany wektor własny $|v\rangle$.
2. Wykonaj QFT na rejestrze kontrolnym.
3. Uruchom $controlled - U^X$ na $|v\rangle$ kontrolowanym przez rejestr kontrolny.
4. Wykonaj QFT^{-1} na rejestrze kontrolnym.
5. Dokonaj pomiaru rejestru kontrolnego i zwróć odczytaną liczbę podzieloną przez 2^X.

9. Faktoryzacja

Rok 1994 przyniósł pewnego rodzaju rewolucję w zainteresowaniu obliczeniami kwantowymi, po tym jak Peter Shor przedstawił algorytm, który (w dużym uproszczeniu) umożliwia efektywne odnajdywanie czynników pierwszych dla liczb całkowitych. Stało się jasne, że klasyczna kryptografia oparta o trudność faktoryzacji (czyli np. bardzo popularne szyfrowanie RSA) jest bezużyteczna wobec odpowiednio dużego komputera kwantowego.

Problem faktoryzacji liczby całkowitej N polega na odnalezieniu dodatnich liczb całkowitych $p_1, p_2, \dots, p_n, r_1, r_2, \dots, r_n$ takich, że liczby p_i są różnymi liczbami pierwszymi oraz $N = p_1^{r_1} p_2^{r_2} \dots p_n^{r_n}$. Dodatkowo przyjmijmy, że interesują nas liczby N nieparzyste (ponieważ dla liczb parzystych jednym z czynników na pewno jest 2) oraz nie będące potęgami liczby pierwszej (można pokazać, że dla N niespełniających tego warunku łatwo jest odnaleźć ich rozkład na czynniki pierwsze). Zwróćmy uwagę, że N spełniające powyższe warunki ma przynajmniej dwa różne nieparzyste czynniki pierwsze.

Faktoryzacja może być sprowadzona do następującego problemu (tzw. określania rzędu): mając dane dwie dodatnie względnie pierwsze liczby całkowite N i a należy określić rząd $a \bmod N$ (czyli taką dodatnią liczbę całkowitą r, że $a^r \bmod N = 1$). Warto zauważyć, że dla poprawnego wejścia zawsze istnieje rozwiązanie powyższego problemu, ponieważ jeśli $NWD(N, a) = 1$ to liczba 1 pojawi się na pewnym miejscu w ciągu $a \bmod N, a^2 \bmod N, a^3 \bmod N \dots$ (po czym jego wartości zaczną się okresowo powtarzać). To jak rozwiązanie powyższego problemu pomaga nam dokonać faktoryzacji N zapiszemy nieco później.

Przedstawmy pewien operator kwantowy U_a:

$$U_a: |s\rangle \to |sa \bmod N\rangle, s < N$$

$$|s\rangle \to |s\rangle, s \geq N$$

Wiemy, że istnieje odwrotność a modulo N (co wynika z względnej pierwszości a i N), czyli liczba a' taka, że $a'a \equiv aa' \equiv 1 \pmod N$. Z tego natychmiast wynika, że U_a jest operatorem unitarnym. Na podstawie powyższego oraz definicji problemu określania rzędu możemy stwierdzić:

$$U_a^r: |s\rangle \to |sa^r \bmod N\rangle = |s\rangle$$

Rozważmy teraz następujący stan kwantowy:

$$|u_k\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} |a^s \bmod N\rangle$$

Jego przekształcenie przez U_a wygląda następująco:

$$U_a|u_k\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} U_a|a^s \bmod N\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} |a^{s+1} \bmod N\rangle =$$

$$= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r}(s+1)} e^{2\pi i \frac{k}{r}} |a^{s+1} \bmod N\rangle = e^{2\pi i \frac{k}{r}} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r}(s+1)} |\, a^{s+1} \bmod N\rangle =$$

$$= e^{2\pi i \frac{k}{r}} |u_k\rangle$$

Ostatnia równość jest prawdziwa, ponieważ $e^{-2\pi i \frac{k}{r} r} |a^r \bmod N\rangle = e^{-2\pi i \frac{k}{r} 0} |a^0 \bmod N\rangle$ (pamiętając $e^{-2\pi i k} = 1$ dla dowolnego całkowitego k). Z tego wynika, że $|u_k\rangle$ jest wektorem własnym przekształcenia U_a z wartością własną $e^{2\pi i \frac{k}{r}}$.

Mając stan reprezentujący wektor własny przekształcenia U_a bylibyśmy w stanie określić $\frac{k}{r}$, korzystając z algorytmu określania wartości własnej przekształcenia. Jak wiemy z opracowania owego algorytmu, umożliwiłoby nam to podanie poprawnej wartości r z dużym prawdopodobieństwem. Tym samym bylibyśmy w stanie rozwiązać problem określania rzędu. Nie znamy jednak *r*, więc nie jesteśmy w stanie przygotować stanu $|u_k\rangle$.

Patrząc na problem z nieco szerszej perspektywy możemy zwrócić uwagę, że nie potrzebujemy odczytywać konkretnej wartości własnej $e^{2\pi i \frac{k}{r}}$ pewnego wektora $|u_k\rangle$. Pomocną byłaby nam dowolna z wartości własnych przekształcenia U_a, a taką możemy odczytać dla dowolnego z wektorów własnych $|u_k\rangle$ dla pewnego *k*. Tym samym możemy wykorzystać superpozycję o równym rozkładzie wszystkich wektorów własnych operatora U_a i odczytać wynik dla jednego, losowo wybranego z nich.

Pokażemy teraz, że jesteśmy w stanie przygotować taką superpozycję nawet nie znając *r*. Stan, który chcemy osiągnąć jest następujący:

$$\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |u_k\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} |a^s \bmod N\rangle$$

Wiemy, że $|a^s \bmod N\rangle = |1\rangle$ dla $s \equiv 0 \pmod r$. Ponieważ $s \in \{0, 1, \dots, r-1\}$ to jedyną wartością s przystającą $0 \pmod r$ jest 0. Stąd całkowita amplituda stanu $|1\rangle$ jest sumą wszystkich amplitud stanów dla których $s = 0$:

$$\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \frac{1}{\sqrt{r}} e^{-2\pi i \frac{k}{r} 0} = \frac{1}{r} \sum_{k=0}^{r-1} 1 = \frac{r}{r} = 1$$

Całkowita amplituda $|1\rangle$ wynosi 1, więc żaden inny stan nie może być odczytany (wszystkie pozostałe amplitudy muszą wynosić 0). Z tego wynika:

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|u_k\rangle = |1\rangle$$

Dlatego chcąc użyć równomiernie ważonej superpozycji wszystkich wektorów własnych U_a wystarczy użyć stanu $|1\rangle$. Potrafimy w takim razie utworzyć stan:

$$|0\rangle|1\rangle = |0\rangle\left(\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|u_k\rangle\right) = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|0\rangle|u_k\rangle$$

Teraz potrafimy wykorzystać algorytm określania wartości własnej przekształcenia, aby odczytać $\frac{k}{r}$ dla pewnego losowo wybranego $k \in \{0, 1, \dots, r-1\}$:

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|0\rangle|u_k\rangle \to |x\rangle|u_k\rangle$$

gdzie $\frac{x}{2^n}$ jest oszacowaniem $\frac{k}{r}$, które z dużym prawdopodobieństwem da nam możliwość dokładnego określenia r korzystając z algorytmu ułamków łańcuchowych (zgodnie z wstępnym opisem z podrozdziału IV-7 o znajdywaniu okresu stanu okresowego).

Algorytm ułamków łańcuchowych (CFA – *Continued Fractions Algorithm*)

Dla każdej liczby wymiernej postaci $\frac{x}{2^n}$ istnieje ciąg (o długości ograniczonej liniowo względem n) przybliżeń $\frac{a_1}{b_1}, \frac{a_2}{b_2}, \dots, \frac{a_m}{b_m}$, gdzie:

- $\frac{a_m}{b_m} = \frac{x}{2^n}$
- $a_1 < a_2 < \cdots < a_m$
- $b_1 < b_2 < \cdots < b_m$

Dodatkowo, jeśli istnieje pewien ułamek $\frac{k}{r}$ taki, że zachodzi $\left|\frac{x}{2^n} - \frac{k}{r}\right| \leq \frac{1}{2r^2}$, to $\frac{k}{r}$ występuje w ciągu przybliżeń $\frac{x}{2^n}$.

Listę wszystkich elementów ciągu można obliczyć w czasie wielomianowym względem n.

Algorytm określania rzędu (OFA – *Order Finding Algorithm*) można podsumować następująco:

1. Wybierz liczbę *n* spełniającą $2^n \geq 2r^2$ (np. $\lceil 2logN \rceil$).
2. Ustaw *n*-qubitowy rejestr kontrolny na $0^{\otimes n}$.
3. Ustaw *n*-qubitowy rejestr docelowy na $|000 \ldots 1\rangle = |1\rangle$.
4. Wykonaj *QFT* na rejestrze kontrolnym.
5. Wykonaj $c - U_a^x$ kontrolowane przez rejestr kontrolny na rejestrze docelowym.
6. Wykonaj *QFT*$^{-1}$ na rejestrze kontrolnym.
7. Odczytaj przybliżenie $\frac{x_1}{2^n}$ z rejestru kontrolnego.
8. Użyj CFA aby odnaleźć c_1, r_1 spełniające $\left|\frac{x_1}{2^n} - \frac{c_1}{r_1}\right| \leq \frac{1}{2^{\frac{n-1}{2}}}$. W razie niepowodzenia zwróć **FAIL**.
9. Powtórz kroki od 1-8 aby uzyskać oszacowanie $\frac{x_2}{2^n}$ oraz kolejną parę c_2, r_2 spełniającą $\left|\frac{x_2}{2^n} - \frac{c_2}{r_2}\right| \leq \frac{1}{2^{\frac{n-1}{2}}}$.
10. Oblicz $r = NWW(r_1, r_2)$
11. Jeśli $a^r \bmod N = 1$ zwróć r. W przeciwnym razie zwróć **FAIL**.

Można udowodnić, że algorytm określania rzędu zwróci poprawny wynik r będący rzędem a mod N z prawdopodobieństwem wynoszącym co najmniej $\frac{384}{\pi^6}$. W przeciwnym razie zwróci wielokrotność *r* lub wartość **FAIL**.

Potencjalnie problematycznym elementem z punktu widzenia złożoności jest wykonanie U_a^x. Dla $x = 2^j$ wykonanie $c - U_a^{2^j}$ wiąże się z wykonaniem operacji $c - U_a$ 2^j razy. Zachodzi jednak $c - U_a^{2^j} = c - U_{a^{2^j}}$ (iloczyn przez $a \bmod N$ wykonany 2^j razy jest równy iloczynowi przez $a^{2^j} \bmod N$). Ponieważ istnieją efektywne klasyczne metody obliczenia $a^{2^j} \bmod N$ możemy z ich pomocą dokonać wcześniejszych przeliczeń i zyskać tym samym wykładnicze przyspieszenie w stosunku do mnożenia przez $a \bmod N$ 2^j razy.

Algorytm faktoryzacji liczby *N* można streścić następująco:

1. Uruchom kwantowy algorytm określania rzędu dla liczby wejściowej N w celu znalezienia rzędu *r.*
2. Jeżeli uzyskaliśmy parzysty rząd zwróć dwie liczby:
   a. $a^{\frac{r}{2}} - 1$
   b. $a^{\frac{r}{2}} + 1$
3. Jeśli uzyskaliśmy nieparzysty rząd, powtórz całą procedurę od punktu 1 dla innego a.

Zwróćmy uwagę, że dla parzystego rzędu r zachodzi $a^r - 1 = (a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1)$. Dodatkowo wiemy, że skoro $a^r \equiv 1 \pmod N$ to $a^r - 1 \equiv 0 \pmod N$, czyli $a^r - 1$ jest podzielne przez *N*. Ponieważ *r* jest parzyste to $\frac{r}{2}$ jest dodatnią liczbą całkowitą, podobnie jak $a^{\frac{r}{2}} - 1$ oraz $a^{\frac{r}{2}} + 1$.

Kwantowy układ użyty w powyższym algorytmie wymaga jedynie $O\big((\log N)^2 \log\log(N) \log\log\log(N)\big)$ elementarnych bramek kwantowych. Najlepsze znane klasyczne algorytm deterministyczne wymagają użycia $e^{O((\log N)^{\frac{1}{2}}(\log\log N)^{\frac{1}{2}})}$ bramek, a najlepsze znane heurystyki $e^{O((\log N)^{\frac{1}{3}}(\log\log N)^{\frac{2}{3}})}$ bramek. Obserwujemy więc przyspieszenie z sub-wykładniczej do sub-liniowej złożoności dla problemu faktoryzacji.

Należy zwrócić zaznaczyć, że oryginalny algorytm zaprezentowany przez Petera Shor'a nieco różni się od podanego powyżej. Opisany w tym podrozdziale algorytm bazuje na możliwości określenia wektora własnego dla danego przekształcenia (tu: U_a), z czego wynikają istotne podobieństwa do algorytmu z podrozdziału IV-8. Jest to jednak procedura dokładnie równoważna pierwotnej, związanej z problemem znajdowania okresu stanu okresowego (opisanej w podrozdziale IV-7). Różnica, z której wynikają różne podejścia, polega na odmiennych bazach przestrzeni stanów. Peter Shor w swoim algorytmie stosował standardową bazę obliczeniową, a algorytm opisany tutaj operuje na bazie rozpiętej przez wektory własne przekształcenia U_a.

Bibliografia:

1. Kaye P., Laflamme R., Mosca M., *Introduction to Quantum Computing*, 1st edition, Oxford University Press 2007
2. Nielsen M. A., Chuang I. L., *Quantum Computation and Quantum Information*, 10th Anniversary Edition, Cambridge University Press 2011
3. Krzysztof Giaro, Marcin Kamiński, *Wprowadzenie do algorytmów kwantowych*, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2003
4. http://www.ibm.com/developerworks/library/l-quant/index.html (odczyt 20.10.2014r.)
5. http://www.technologyreview.com/featuredstory/531606/microsofts-quantum-mechanics/ (odczyt 28.11.2014r.)
6. http://www.dwavesys.com/tutorials/background-reading-series/introduction-d-wave-quantum-hardware (odczyt 29.11.2014r.)
7. http://qudev.ethz.ch/content/courses/QSIT07/qsit05_v1_2page.pdf (odczyt 29.11.2014r.)
8. http://www.stat.physik.uni-potsdam.de/~pikovsky/teaching/stud_seminar/Bell_EPR-1.pdf (odczyt 2.12.2014r.)
9. http://qudev.phys.ethz.ch/content/courses/QSIT09/QSIT09_V04_slides.pdf (odczyt 17.12.2014r.)
10. http://www.eecs.berkeley.edu/~luca/quantum/lecture06.pdf (odczyt 3.01.2015r.)
11. http://www.dsp.agh.edu.pl/_media/pl:dydaktyka:fft.pdf (odczyt 3.01.2015r.)
12. http://www.eecs.berkeley.edu/~luca/quantum/lecture07.pdf (odczyt 3.01.2015r.)
13. https://primes.utm.edu/notes/relprime.html (odczyt 4.01.2015r.)

Strona projektu:

- https://github.com/byakuya6/Quantum-Shell