量子アルゴリズムの提案

03-130997 松本卓朗

2014 年 7 月 28 日

1　序論

私は昨年度、量子コンピュータに関する講義を受講したが、受講する前は私は無知であり、量子コンピュータが時間が進むにつれて計算できる量が増えていくような計算機ではないかと思い込んでいたのであるが、そのようなコンピュータではなく、n qubit あったら、2^n 通りを同時に計算できる計算機であった。私の研究はここから始まることになる。講義を行っていた長谷川准教授は私を量子コンピュータ製作に誘ってくれたので、私は量子コンピュータが自分の手に届くものであると感じ、量子アルゴリズムの開発を試みるようになった。この報告書において、そのアルゴリズムを詳しく解説する。

2　プロジェクトの目的

量子コンピュータに関する講義は半学期しか与えられず、その深い理解には到底不十分である。そこで、このプロジェクトを通して、長い時間をかけて、その理解を試みる。そして、さらにそこで得た知識を用いて、未だに知られていない（もしくは意味があるか不明な）アルゴリズムの開発の可能性を追求しようと考えている。また私はこの研究によって、先生方に推薦状を書いていただき、アメリカ留学をしようと考えている。

3　プロジェクトの実施内容

私は量子アルゴリズムに関するアイデアをいくつか思いついたので、それに関する研究を行うのであるが、そのアイデアが実現可能であるかどうかは、アイデアが複雑であり、かついまだに量子コンピュータでそれを計算できる性能のものがないために、まだよくわかっていない。そこで、私はそれを実現するために、”Quantum Information and Quantum Computation”を中心とする本をよく読みながら、このアイデアの否定を試み、さらに改良を加えていくというのが、このプロジェクトの内容である。その工程を時系列的に、また素人にもわかりやすく、かつ論理的にまとめたのが次の「議論」の章である。

4　議論

4.1　最初のアイデア

前述したとおり、量子コンピュータは 2^n 通りを同時に計算できる。これを聞いて、私はいくつかのアルゴリズの原案を考え付いた。まず、一つ目は自動定理証明である。これはゴールドバッハ予想問題のような大き

な未解決問題に対して、あっているかあっていないかわからない証明を 2^n 通り与える。証明があっているかあっていないかを並列的に計算し、あっているものがあれば、それを抜き出し、証明とする。この方法を用いると、n bit 以内の証明を持つものなら証明が一瞬でできてしまうことになり、非常に有用である。次に考え付いたのが作曲である。2^n 通りの曲を入力して、脳のシミュレーションを用いた評価関数でよい曲かどうかを計算し、最もいい曲を抜き出すと、すばらしい曲が作曲できるはずである。脳のシミュレーションも、脳細胞が並列的に動いているだけなので、量子コンピュータで一瞬に計算できてしまうと考えた。この原理で成功すれば、これ以外にも作画、発明などと夢が広がる。また最後にもう一つのアイデアとして、細胞の設計のようなインテリジェントデザインができてしまうのではないかとも考えている。しかし、これは非常に未来のことであるので、あまり深くは追わない。問題はこの n がどれくらいの数になるのかということであるが、現在は n=512 ぐらいが最大で、Rose's law[5] によれば、これが 1 年に 2 倍くらいのスピードで増えていくとのことである。$n = 10^6$ ぐらいになれば、このような分野でよい成果が残せるのはないかと考えた。では次節からこのアイデアの実らせるために、否定にとりかかる。

4.2 量子コンピュータの原理

4.2.1 状態の指数関数的増加

2^n 通りを同時に計算することのできる量子コンピュータとはどのようなものであるだろうか。まずその原理の理解から試みる。それにはまず、量子力学から理解しなければならない。量子力学は小さな世界に関する運動の法則を追うときに、大きな世界での運動の法則が通用しないために、別の法則を考えようというものである。その法則の一つが、重ね合わせである。小さな世界の物質である「量子」は 2 つの状態を同時にとることができる。その状態がどちらかであるかを観測しようとすると、そのどちらかだけが観測される。ただし、これは観測される前から 2 つの状態のどちらかに決まっていたということではなく、2 つの状態が同時にあると考えて、観測によってどちらかになると考えると、整合性がとれるのである。ここで量子とはたとえば、光の送る粒子である光子であり、状態は、その光子の振動の度合いといった類のものである。2 つの状態があるとき、その状態を $|0\rangle$、$|1\rangle$ と記述するとすれば、それぞれが観測される確率の平方根（後の計算の都合上、平方根を用いる）を α、β とすることにより、

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

と記述することができる。すなわち、この量子を観測することによって $|0\rangle$ を観測する確率は $|\alpha|^2$、$|1\rangle$ を観測する確率は $|\beta|^2$ である。合計で確率は 1 とならなければならないので、$|\alpha|^2 + |\beta|^2 = 1$ を満たす。たとえば、

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \tag{2}$$

とすれば、観測を行うと、$|0\rangle$ と $|1\rangle$ が等確率で観測される。量子の重ね合わせで記述できる状態は 2 つだけではない。たとえば、この量子が 2 つあれば,

$$|\psi\rangle = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \tag{3}$$

$$= \frac{1}{2}|0\rangle|0\rangle + \frac{1}{2}|0\rangle|1\rangle + \frac{1}{2}|1\rangle|0\rangle + \frac{1}{2}|1\rangle|1\rangle \tag{4}$$

$$= \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle \tag{5}$$

$$= \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle + \frac{1}{2}|2\rangle + \frac{1}{2}|3\rangle \tag{6}$$

のように 4 つの状態を作ることができる。すなわち、この量子 2 つを観測すると、$|00\rangle$、$|01\rangle$、$|10\rangle$、$|11\rangle$ を等確率で観測することができる。量子の数を増やして、このような操作を行うと、状態の個数は、この量子の数、すなわち qubit の数に対して、指数関数的になり、2^n 個となる。これを用いて、計算を行うのである。

4.2.2 ゲート

2^n 通りの状態を表す方法を述べ終えたので、次はこれらを計算する方法について述べていく。この報告書では、まず初めに 3 つの基本的なゲートを紹介する。紹介するゲートには、1qubit のみに作用するものと 2qubit に作用するものがある。まず、1qubit のゲートに関して説明する。$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ から $|\psi'\rangle = \alpha'|0\rangle + \beta'|1\rangle$ に変化させるゲートがあったとき、2 次正方行列 A を用いて、

$$\begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} = A \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{7}$$

と記述することにする。このように記述したとき、2 次正方行列の部分が、

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{8}$$

となる変換ができるゲートを Hadamard ゲート,

$$T = \begin{pmatrix} 1 & 0 \\ 0 & exp(\frac{i\pi}{4}) \end{pmatrix} \tag{9}$$

となる変換ができるゲートを $\frac{\pi}{8}$ ゲートと呼び、これらは物理的に実装することが可能と言われているものである。物理的実装方法に関しては後の節で紹介する。ゲートは複雑なゲート群もわかりやすくするため、左側を入力する状態、右側を出力する状態として、たとえば図 1 のように記述する。

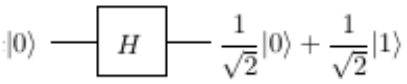

図 1　Hadamard ゲート

次に 2qubit を操作するゲートについて説明する。$\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$ から $\psi\rangle = \alpha'|00\rangle + \beta'|01\rangle + \gamma'|10\rangle + \delta'|11\rangle$ に変化させるゲートがあったとき、4 次正方行列 A を用いて、

$$\begin{pmatrix} \alpha' \\ \beta' \\ \gamma' \\ \delta' \end{pmatrix} = A \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix} \tag{10}$$

のように推移状態を記述することにする。このように記述したとき、4 次正方行列が、

$$U_{CN} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{11}$$

となるゲートを制御 NOT ゲートと言い、数学的には排他的論理和を表す。1qubit 目が $|0\rangle$ のときは、2qubit 目は何も変化がなく、1qubit 目が $|1\rangle$ のときは、2qubit 目は反転する。1qubit 目を上の線としたときの例を図 2 に示す。このゲートも物理的に構成できることが知られている。これで基本的なゲート 3 つを全て述べたので、次節からは実践的な計算例を示す。

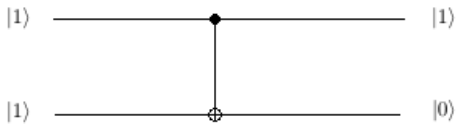

図 2　制御 NOT ゲート

4.2.3　並列計算

この節では私の報告書でよく用いられる並列計算手法について示す。$T^\dagger = (T^*)^{\mathrm{T}} = T^7$（すなわち、T の複素共役）を定義して、次のゲートを導入する。

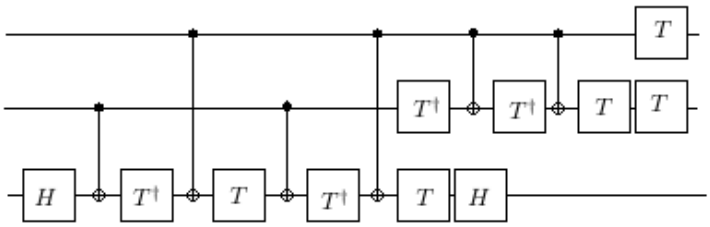

図 3　Toffoli ゲート

上から qubit の線を a, b, c としたとき、この関数を先ほどの定義にしたがって計算すると、以下の図のようになる。上からの線が、左からの列に対応する。

入力			出力		
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

図 4　Toffoli ゲートの入出力

(a, b, c) → (a, b, c⊕ab) となる関数であることがわかる。ここで ⊕ は排他的論理和である。すなわち (a,

b)=(1, 1) のときのみ、c が反転する。よく用いるゲートなので、次の図のように簡略化して記述することにする。

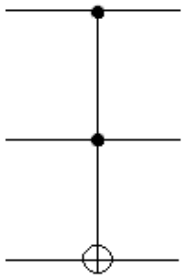

図 5　Toffoli ゲート

前述した c のところを 0 とすれば ab を出力する AND 回路、1 とすれば ab の否定を返す NAND 回路を作ることができる。これを用いて、並列計算の思考実験を行うために次のような回路を作る。

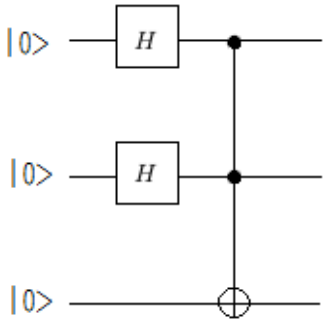

図 6　並列計算

このように計算すると、まず 1 段階目のステップで、

$$|\psi\rangle = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)|0\rangle \tag{12}$$

$$= (\frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle)|0\rangle \tag{13}$$

$$\tag{14}$$

となり、2^2 通りが、Toffoli ゲートに入る準備をすることになる。次に 2 段階目で、Toffoli ゲートは 3 つ目が 0 のとき、AND 回路として動くので、

$$|\psi\rangle = \frac{1}{2}|00\rangle|0\rangle + \frac{1}{2}|01\rangle|0\rangle + \frac{1}{2}|10\rangle|0\rangle + \frac{1}{2}|11\rangle|1\rangle \tag{15}$$

$$\tag{16}$$

のように計算でき、2^2 通りが同時に AND を計算したことになる。しかし、ここで注意したいのは、これは理論上を追っているだけで、わかるとおり、これを観測すると、4 通りのいずれかの入力にたいして、答えが一つ出てくるだけで、普通に計算したときと変わらないばかりかランダムで使いにくいことになる。しかし、後に示す Grover のアルゴリズムを用いると、並列計算を実際に行ったかのような結果を得ることができる。

4.2.4　任意の関数の作成

前節では 2qubit から 1qubit に出力する AND 計算を行った。しかし、この出力結果は自明なものである。さらに複雑な関数を作って、並列計算させることができるのだろうか。答えは YES である。n qubit から n qubit へのデジタル回路的な任意の関数は、時間を気にしなければ、必ず作ることができる。

前節までにおいて、制御 NOT を用いれば、NOT を作れることを示し、Toffoli ゲートを用いれば、AND を作れることを示した。これを用いれば、ド・モルガン則より、OR を作ることができる。式で書けば、

$$a \vee b = \neg(\neg a \wedge \neg b) \tag{17}$$

である。図で描くと、

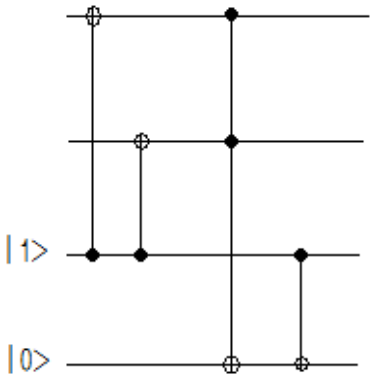

図 7　OR

のようになる。

n qubit から n qubit への任意の変換の議論にうつる。可能かどうかに関して言えば、出力の n qubit は独立に計算できるので、n qubit から 1qubit への任意の変換について議論すれば十分である。作りたい関数を f とし、入力を n qubit の x、出力を 1qubit の y として、y=f(x) について考える。x は qubit の 2 進法表示で、$x_1x_2\cdots x_n$ とする。f(x)=1 となるような全ての $x_1x_2\cdots x_n$ に対して、次のような操作をする。それぞれの i に対して、x_i=1 のときは、x_i を使い、x_i=0 のときは、$\neg x_i$ を用いて、積とする。たとえば、f(010)=1 ならば、$\neg x_1 \wedge x_2 \wedge \neg x_3$ とすることを行う。そして、このように操作した x それぞれの操作結果の和をとる。たとえば、x=001, 010 のときに限って、y=1 となるならば、$(\neg x_1 \wedge \neg x_2 \wedge x_3) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3)$ であり、これは求めたい関数になり、かつ、全て AND、OR、NOT で表現できるので、これを量子ゲートで置き換えればよい。

この手法を用いれば、全ての関数を作成することができるのだが、複雑な関数をこの方法で構成しようとすると、並列計算したときに、そうしないときに比べてかえって時間がかかってしまうということが起こりうる。しかし、古典コンピュータも基本はこのような回路から構成されているので、このような説明によって、量子コンピュータが古典コンピュータと同等の計算能力を持つことを示すとともに、関数作成の仕方によっては良いアルゴリズムを作ることができる可能性をまだ残しているということを示すことができた。

4.2.5 その他

また、この章の最後に、量子計算で用いられる並列計算手法は、前述した手法だけではないことも述べておく。詳しくは Shor のアルゴリズムなどを参照されたい。また、一般的な話として、後の議論で用いられる量子ゲートのユニタリ性についても述べておく。ユニタリ性とは、$UU^{\dagger} = I$ となるような性質であり、全ての量子ゲートはこの性質をもち、さらにユニタリ性をもつならば、量子ゲートとして記述することができる。証明は簡単に説明すると、前述した全てのゲートは全てユニタリ行列であり、ユニタリ行列同士の積もユニタリ行列なので、どのようなゲートもユニタリ行列になる。また、任意のユニタリ行列を指定した場合、それを近似的に表すゲート群を構成する方法が存在する。その証明は長いので割愛することにする。

4.3 Grover のアルゴリズム

4.3.1 はじめに

私はこの報告書の初めに自動定理証明と作曲という 2 つのアルゴリズムを考え付いたが、その複雑な関数については前章で示したように古典コンピュータと同等の計算能力を兼ね備えており、時間をかければ必ず計算できる。ここで議論したいのは、自動定理証明では、様々な合っているか合っていないかわからない証明から合っているものを抜き出す操作が必要であり、作曲においても、あらゆる曲からいい曲を抜きだすことが必要であるということである。たとえば、自動定理証明について、前章と合わせて考えてみる。様々な証明を、その証明が合っていれば 1 を返し、間違っていれば 0 を返す関数にいれて、そのままの状態で観測を行うと、ランダムな合ってるかわからない証明に関して、その証明があっているかだけがだけが返り、基本的に何も証明することはできない。しかし、Grover のアルゴリズムを使うと 1 を返した証明を抜き出すことができるのである。しかし、そのアルゴリズムの計算には夢のように短い時間でできるというわけではなく、その計算時間について議論することにした。

4.3.2 アルゴリズム

まず、Grover のアルゴリズムがどのようなアルゴリズムなのかを説明する。n qubit について考える。前章のデジタル回路的考えを用いれば、N=2^n 通りの状態をとることができる。その n qubit の重ね合わせになっている一つの状態 x について、答えが 0 か 1 になる関数 f(x) を考える。問題は f(x)=1 となるような x を求めることであるが、それには次の操作を行えばよい。

1. x になる n qubit と計算結果用の 1qubit を用意して、初期化を行う。状態は、$|0\rangle^{\otimes n}|0\rangle$ となる。ここで $\otimes n$ は n 個あることを示す。
2. 最初の n qubit 全てに Hadamard 変換を適用し、最後の 1 qubit は、NOT 変換を行ったあと、Hadamard 変換を適用する。すると、

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{18}$$

という状態が得られる。

3. 次に示す Grover iteration を繰り返す。この回数は後に議論するが、$O(\sqrt{2^n})$ 回程度である。
   （a）$|x\rangle$ について、関数 f を計算し、計算された 0 か 1 の値を用いて最後の 1qubit との排他的論理和を制御 NOT を用いて計算し、その 1qubit に入れる。
   （b）最初の全ての n qubit に対して、Hadamard 変換を適用する。
   （c）最初の n qubit に対して、$|x\rangle$ が $|0\rangle$ でないなら、結果を 1、そうでないなら 0 として、前述した手法と同じように、これを用いて、最後の 1qubit との排他的論理和を制御 NOT を用いて計算し、その 1qubit に入れる。この操作は前述した任意の関数の作成方法により、短いステップで簡単に実現できる。
   （d）最初の全ての n qubit に対して、Hadamard 変換を適用する。
4. 最初の n qubit を観測すると、高い確率で f(x)=1 となるときの x が手に入る。

このアルゴリズムで注意したいのは、Grover iteration を $O(\sqrt{2^n})$ 回程度繰り返すことが必要な点である。$O(\sqrt{2^n})$ 回より速いアルゴリズムはないことが知られているのだが、その時点で、1 秒間に 10^9 回計算でき、10^6 秒ほどの計算時間があったとしても、$10^{30} \simeq 2^{100}$ 通りの中ぐらいの解ぐらいしか拾ってこれないことになり、前述した手法を用いても、100bit 程度までの自動定理証明や作曲しかできないことになる。また、細胞の設計といったインテリジェントデザインは、簡単な手法を用いる方法では我々の生きている時間では不可能になる。

4.3.3 図による証明

$O(\sqrt{2^n})$ という回数が具体的にどれくらいかも見るため、具体的な図を用いた証明を書く。まず、前述の 2 を終えた時点で、最初の n qubit は、状態全てが同じ確率で存在し、$N=2^n$ とすると、

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \tag{19}$$

これを次のように分解する。

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{\{x|f(x)=0\}} |x\rangle \tag{20}$$

$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{\{x|f(x)=1\}} |x\rangle \tag{21}$$

として、

$$|\psi\rangle = \sqrt{\frac{N-M}{N}}|\alpha\rangle + \sqrt{\frac{M}{N}}|\beta\rangle \tag{22}$$

と考える。ここで

$$cos\frac{\theta}{2} = \sqrt{\frac{N-M}{N}} \tag{23}$$

と置くと、

$$|\psi\rangle = cos\frac{\theta}{2}|\alpha\rangle + sin\frac{\theta}{2}|\beta\rangle \tag{24}$$

と書ける。後に証明するが、ここで前述の Grover iteration を k 回行うと、

$$G^k|\psi\rangle = \left(cos\frac{2k+1}{2}\theta\right)|\alpha\rangle + \left(sin\frac{2k+1}{2}\theta\right)|\beta\rangle \tag{25}$$

のようになるため、$sin((2k+1)\theta)/2 \simeq 1$ となるまで、Grover iteration を繰り返せばよい。図で表すと次のようになる。

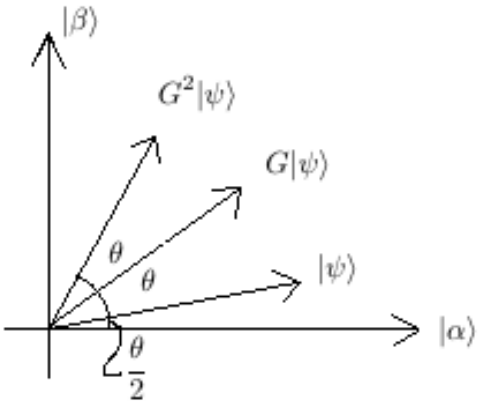

図 8　幾何学的証明

証明の大まかな流れがわかったところで、数式 (25) を証明する。アルゴリズムの 3 の (a),(c) において、計算した値を用いて排他的論理和を行っているが、これは、最後の 1qubit は

$$\frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{26}$$

の形を持っているので、これと 0 との排他的論理和は変化がなく、1 との排他的論理和は、

$$\frac{|1\rangle - |0\rangle}{\sqrt{2}} = -\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \tag{27}$$

ということを考えると、これは状態の正負の反転を意味する。

また、ここでテンソル積を定義する。$A = (a_1, a_2, \ldots, a_{n)}$, $B = (b_1, b_2, \ldots, b_{m)}$ としたとき、$A \otimes B$ は行列であり、$A_{ij} = a_i b_j$ である。また、A, B ではなく、$|\psi\rangle$、$|\phi\rangle$ のような表示を用いたとき、そのテンソル積は、$|\psi\rangle\langle\phi|$ と書く。

これらを用いると、Grover iteration を別の表記で次のように簡単に記述できる。

1. $|x\rangle \to (-1)^{f(x)}|x\rangle$
2. $H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n}$ を左からかける。

ここで、$2|0\rangle\langle 0| - I$ の 0 は n 個の全ての qubit において、$|0\rangle$ になっているという意味であり、$2^n \times 2^n$ 行列によって、

$$\begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & -1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & -1 \end{pmatrix} \tag{28}$$

となるので、確かに前述の (c) と一致する。また、(19) を用いると、

$$H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n} = 2H^{\otimes n}|0\rangle\langle 0|H^{\otimes n} - H^{\otimes n}IH^{\otimes n} \tag{29}$$

$$= 2(H^{\otimes n}|0\rangle)(\langle 0|H^{\otimes n}) - H^{\otimes n}H^{\otimes n} \tag{30}$$
$$= 2|\psi\rangle\langle\psi| - I \tag{31}$$

ここで、

$$H\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{32}$$
$$= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{33}$$

$$H^2 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{34}$$
$$= I \tag{35}$$

を用いた。

また、

$$(2|\psi\rangle\langle\psi| - I)\sum_k \alpha_k|k\rangle = (\frac{2}{N}\begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix})\sum_k \alpha_k|k\rangle \tag{36}$$
$$= \sum_k \left(\frac{2}{N}\sum_l \alpha_l - \alpha_k\right)|k\rangle \tag{37}$$

であるので、これを用いて (25) を数学的帰納法によって示す。k=0 のときは明らかなので、k のとき成り立つと仮定する。すなわち、

$$G^k|\psi\rangle = cos\frac{(2k+1)\theta}{2}|\alpha\rangle + sin\frac{(2k+1)\theta}{2}|\beta\rangle \tag{38}$$

とし、$|x\rangle \to (-1)^{f(x)}|x\rangle$ の操作を O とすると、

$$G^{k+1}|\psi\rangle = (2|\psi\rangle\langle\psi| - I)OG^k|\psi\rangle \tag{39}$$
$$= (2|\psi\rangle\langle\psi| - I)O\left(cos\frac{(2k+1)\theta}{2}|\alpha\rangle + sin\frac{(2k+1)\theta}{2}|\beta\rangle\right) \tag{40}$$
$$= (2|\psi\rangle\langle\psi| - I)\left(cos\frac{(2k+1)\theta}{2}|\alpha\rangle - sin\frac{(2k+1)\theta}{2}|\beta\rangle\right) \tag{41}$$
$$= (2|\psi\rangle\langle\psi| - I)\Bigg(cos\frac{(2k+1)\theta}{2}\frac{1}{\sqrt{N-M}}\sum_{\{x|f(x)=0\}}|x\rangle \tag{42}$$
$$-sin\frac{(2k+1)\theta}{2}\frac{1}{\sqrt{M}}\sum_{\{x|f(x)=1\}}|x\rangle\Bigg) \tag{43}$$
$$= \Bigg(\frac{2}{N}(cos\frac{(2k+1)\theta}{2}\sqrt{N-M} - sin\frac{(2k+1)\theta}{2}\sqrt{M}) \tag{44}$$
$$-cos\frac{(2k+1)\theta}{2}\frac{1}{\sqrt{N-M}}\Bigg)\sum_{\{x|f(x)=0\}}|x\rangle \tag{45}$$
$$+\Bigg(\frac{2}{N}(cos\frac{(2k+1)\theta}{2}\sqrt{N-M} - sin\frac{(2k+1)\theta}{2}\sqrt{M}) \tag{46}$$

$$+sin\frac{(2k+1)\theta}{2}\frac{1}{\sqrt{M}}\Bigg)\sum_{\{x|f(x)=1\}}|x\rangle \tag{47}$$

$$=\left((1-\frac{2M}{N})cos\frac{(2k+1)\theta}{2}-\frac{2\sqrt{M(N-M)}}{N}sin\frac{(2k+1)\theta}{2}\right)\frac{1}{\sqrt{N-M}}\sum_{\{x|f(x)=0\}}|x\rangle \tag{48}$$

$$+\left(\frac{2\sqrt{M(N-M)}}{N}cos\frac{(2k+1)\theta}{2}+(1-\frac{2M}{N})cos\frac{(2k+1)\theta}{2}\right)\frac{1}{\sqrt{M}}\sum_{\{x|f(x)=1\}}|x\rangle \tag{49}$$

$$=\left((1-\frac{2M}{N})cos\frac{(2k+1)\theta}{2}-\frac{2\sqrt{M(N-M)}}{N}sin\frac{(2k+1)\theta}{2}\right)|\alpha\rangle \tag{50}$$

$$+\left(\frac{2\sqrt{M(N-M)}}{N}cos\frac{(2k+1)\theta}{2}+(1-\frac{2M}{N})cos\frac{(2k+1)\theta}{2}\right)|\beta\rangle \tag{51}$$

ここで、(23) より、

$$cos\theta=2cos^2\frac{\theta}{2}-1 \tag{52}$$

$$=1-\frac{2M}{N} \tag{53}$$

また、$0<\theta<\frac{\pi}{2}$ としてよいので、

$$sin\theta=\sqrt{1-cos^2\theta} \tag{54}$$

$$=\frac{2\sqrt{M(N-M)}}{N} \tag{55}$$

より、(49) は、

$$\left(cos\theta cos\frac{(2k+1)\theta}{2}-sin\theta sin\frac{(2k+1)\theta}{2}\right)|\alpha\rangle+\left(sin\theta cos\frac{(2k+1)\theta}{2}+cos\theta cos\frac{(2k+1)\theta}{2}\right)|\beta\rangle \tag{56}$$

$$=cos\frac{(2(k+1)+1)\theta}{2}|\alpha\rangle+sin\frac{(2(k+1)+1)\theta}{2}|\beta\rangle \tag{57}$$

となり、証明が終わる。

4.3.4 例

上述の説明だけではわかりにくいと思うので、具体的な例を用いて実際にこのように計算できることを実証する。$N=2^3$ とし、$x=0,1,\cdots,7$ となる f(x) に対して、x=2 のとき、f(x)=1、$x\neq 2$ のとき、f(x)=0 となる関数について考え、この 2 を計算結果として出すことを考える。上述の Grover のアルゴリズムで、最後の 1qubit は正負の反転を行う操作なので、これをもとにして、最初の n qubit のみについて考えればよい。最初の初期化で、その n qubit が、

$$|\psi\rangle=\frac{1}{\sqrt{8}}\begin{pmatrix}1\\1\\1\\1\\1\\1\\1\\1\end{pmatrix} \tag{58}$$

となる。

$$2|\psi\rangle\langle\psi| - I = (\frac{2}{8}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}) \tag{59}$$

$$= \begin{pmatrix} -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{3}{4} \end{pmatrix} \tag{60}$$

よって、

$$G|\psi\rangle = (2|\psi\rangle\langle\psi| - I)O\frac{1}{\sqrt{8}}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \tag{61}$$

$$= (2|\psi\rangle\langle\psi| - I)\frac{1}{\sqrt{8}}\begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \tag{62}$$

$$= \frac{1}{\sqrt{8}}\begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{5}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \tag{63}$$

この計算でわかるように x=2 のときのみ $f(x) = \frac{5}{2\sqrt{8}}$ となり、$x \neq 3$ のとき、$f(x) = \frac{1}{2\sqrt{8}}$ となり少しずつ起き上がってくる。さらに計算すると、

$$G^2|\psi\rangle = \frac{1}{\sqrt{8}} \begin{pmatrix} -\frac{1}{4} \\ -\frac{1}{4} \\ \frac{11}{4} \\ -\frac{1}{4} \\ -\frac{1}{4} \\ -\frac{1}{4} \\ -\frac{1}{4} \\ -\frac{1}{4} \end{pmatrix} \tag{64}$$

のようになり、この状態で観測を行うと、x=2 という結果が、$(\frac{11}{4\sqrt{8}})^2 = \frac{121}{128}$ の確率で計算できる。

4.3.5 計算量

この節では Grover iteration を何回ぐらい適用しなければいけないかについて議論する。前節でやったとおり、Grover iteration の意味は回転であった。

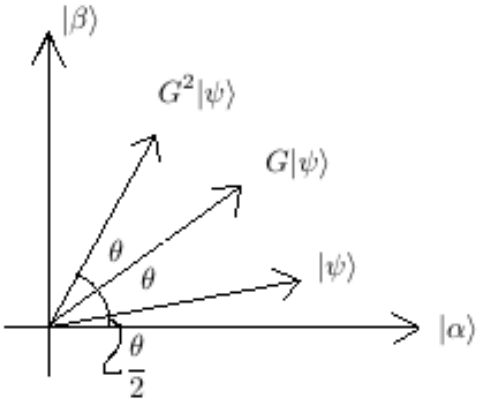

図 9　幾何学的証明

$|\psi\rangle$ が最も起き上がったときの Grover iteration を適用した回数を R として、これがどれくらいになるかを求める。まず、

$$R \leq \lceil \frac{\frac{\pi}{2}}{\theta} \rceil \tag{65}$$

$$= \lceil \frac{\pi}{2\theta} \rceil \tag{66}$$

となり、また、

$$\frac{\theta}{2} \geq sin\frac{\theta}{2} \tag{67}$$

$$= \sqrt{\frac{M}{N}} \tag{68}$$

であるので、

$$R \leq \lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \rceil \tag{69}$$

となる。これは M がわかっていなければ計算できないことにも注意したい。M がわからないときの計算方法については、あとで記すことにする。ここでは、Grover のアルゴリズムより速いアルゴリズムがないかについて議論する。しかし、この操作は最低でも関数 f の計算を $O(\sqrt{N})$ 回ほど計算しなければならないことが知られている。これはこの報告書において、新たな量子アルゴリズムを現実的に作成する重要なポイントであるため、証明を全て書くことにする。教科書には証明は M=1 のときのみしか載っていなかったが、$M > 1$ のときも、これから述べる証明を用いて背理法により証明できると考えられる。

証明のおおまかな道筋は、

$$O_x = I - 2|x\rangle\langle x| \tag{70}$$

$$|\psi_k^x\rangle = U_k O_x U_{k-1} O_x \cdots U_1 O_x |\psi\rangle \tag{71}$$

$$|\langle x|\psi_k^x\rangle|^2 \geq \frac{1}{2} \tag{72}$$

すなわち、O_x を値が x のときのみビットを負へ反転させるオラクル関数、$|\psi_k^x\rangle$ は、U をユニタリ変換として欲しい状態を起き上がらせることを考え、欲しい結果が 1/2 以上の確率で手に入るときのことを考える。これを仮定すると、

$$k \geq \sqrt{\frac{0.42N}{4}} \tag{73}$$

となる。具体的にこれを証明していく。まず、

$$|\psi_k\rangle = U_k U_{k-1} \cdots U_1 |\psi\rangle \tag{74}$$

$$D_k = \sum_x ||\psi_k^x - \psi_k||^2 \tag{75}$$

とすると、$D_k \leq 4k^2$ となることを数学的帰納法で示す。k=0 のときは明らかなので、k のとき成り立つとすると、

$$\begin{aligned}
D_{k+1} &\leq \sum_x ||O_x\psi_k^x - \psi_k||^2 && (76)\\
&= \sum_x ||O_x(\psi_k^x - \psi_k) + (O_x - I)\psi_k||^2 && (77)\\
&= \sum_x ||O_x(\psi_k^x - \psi_k) - 2\langle x|\psi_k\rangle|x\rangle||^2 && (78)\\
&\leq \sum_x \left(||O_x(\psi_k^x - \psi_k)||^2 + 4||O_x(\psi_k^x - \psi_k)||\langle x|\psi_k\rangle|x\rangle| + 4|\langle x|\psi_k\rangle|x\rangle|^2\right) && (79)\\
&= \sum_x ||(\psi_k^x - \psi_k)||^2 + 4\sum_x ||(\psi_k^x - \psi_k)||\langle x|\psi_k\rangle| + 4\sum_x |\langle x|\psi_k\rangle|^2 && (80)\\
&= D_k + 4\sum_x ||\psi_k^x - \psi_k||\langle x|\psi_k\rangle| + 4 && (81)\\
&\leq D_k + 4\left(\sum_x ||\psi_k^x - \psi_k||^2\right)^{\frac{1}{2}}\left(\sum_x |\langle x|\psi_k\rangle|^2\right)^{\frac{1}{2}} + 4 \ (Cauchy - Schwarz \text{ の不等式}) && (82)\\
&= D_k + 4\sqrt{D_k} + 4 && (83)\\
&= 4k^2 + 8k + 4 && (84)\\
&= 4(k+1)^2 && (85)
\end{aligned}$$

より成り立つ。また、

$$D_k = \sum_x ||(\psi_k^x - x) + (x - \psi_k)||^2 \tag{86}$$

$$\geq \sum_x ||\psi_k^x - x||^2 - 2\sum_x ||\psi_k^x - x||||x - \psi_k|| + \sum_x ||x - \psi_k||^2 \tag{87}$$

$$\geq \sum_x ||\psi_k^x - x||^2 - \sqrt{\sum_x ||\psi_k^x - x||^2 \sum_x ||x - \psi_k||^2} + \sum_x ||x - \psi_k||^2 \tag{88}$$

$$(Cauchy - Schwarz \text{ の不等式}) \tag{89}$$

$$= \left(\sqrt{\sum_x ||x - \psi_k||^2} - \sqrt{\sum_x ||\psi_k^x - x||^2}\right)^2 \tag{90}$$

ここで、

$$\sum_x ||\psi_k^x - x||^2 = \sum_x (||\psi_k^x||^2 + ||x||^2 - 2|\langle x|\psi_k^x\rangle|) \tag{91}$$

$$\leq \sum_x (2 - \sqrt{2}) \tag{92}$$

$$= (2 - \sqrt{2})N \tag{93}$$

また、

$$\sum_x ||x - \psi_k||^2 = \sum_x (||x||^2 + ||\psi_k||^2 - 2|\langle x|\psi_k\rangle|) \tag{94}$$

$$= \sum_x (2 - 2|\langle x|\psi_k\rangle|) \tag{95}$$

$$= 2N - 2\sum_x |\langle x|\psi_k\rangle| \tag{96}$$

$$\geq 2N - 2\left(\sum_x |\langle x|\psi_k\rangle|^2\right)^{\frac{1}{2}} \quad (Cauchy - Schwarz \text{ の不等式}) \tag{97}$$

$$= 2N - 2\sqrt{N} \tag{98}$$

よって、

$$D_k \geq \left(\sqrt{2N - 2\sqrt{N}} - \sqrt{(2 - \sqrt{2})N}\right)^2 \tag{99}$$

$$= \left(\sqrt{2 - \frac{2}{\sqrt{N}}} - \sqrt{2 - \sqrt{2}}\right)^2 N \tag{100}$$

$$\to \left(\sqrt{2} - \sqrt{2 - \sqrt{2}}\right)^2 N \ \ (N \to \infty) \tag{101}$$

$$\approx 0.42N \tag{102}$$

ゆえに、

$$4k^2 \geq D_k \geq 0.42N \tag{103}$$

$$k \geq \sqrt{\frac{0.42N}{4}} \tag{104}$$

4.3.6 実際の計算法

前述した手法では、f(x)=1 となる x の個数である M がわかっていなければ、x を計算することは難しかった。教科書には量子計数という手法が載っており、M の数を数えることができる。まず、この手法を用いて、実際の計算がどれくらい時間がかかるのかという計算を行った。計算量についてだけ議論したいので、その具体的な計算方法は割愛するが、これも Grover iterator を繰り返し計算する必要がある。計算に失敗する確率を ϵ とし、

$$t = m + \lceil log(2 + \frac{1}{2\epsilon}) \rceil \tag{105}$$

とすると、M の値が、

$$|\Delta M| < \left(\sqrt{2MN} + \frac{N}{2^{m+1}} \right) 2^{-m} \tag{106}$$

の誤差で計算するには、Grover iteration を $2^t - 1$ 回計算しなければならない。そこで、具体的な値を設定し、この計算量について考えた。たとえば、

$$m = \lceil \frac{n}{2} \rceil + 1 \tag{107}$$

$$N = 2^n \tag{108}$$

$$\epsilon = \frac{1}{6} \tag{109}$$

とすると、

$$\left(\sqrt{2MN} + \frac{N}{2^{m+1}} \right) 2^{-m} \tag{110}$$

$$= \sqrt{2M} 2^{\frac{n}{2} - \lceil \frac{n}{2} \rceil + 1} + 2^{n - 2\lceil \frac{n}{2} \rceil - 1} \tag{111}$$

$$\geq \sqrt{2M} 2^{\frac{1}{2}} + 2^{-2} \tag{112}$$

$$= 2\sqrt{M} + \frac{1}{4} \tag{113}$$

また、

$$2^t - 1 = 2^{\lceil \frac{n}{2} \rceil + 3} - 1 \tag{114}$$

$$\geq 8 \cdot 2^{\frac{n}{2}} - 1 \tag{115}$$

とすると、誤差が大きいにもかかわらず、Grover のアルゴリズムの計算量をはるかに上回る計算を行わなければならない。そこでこの方法については考えないことにした。そこで考えついたのが、M がわからなくても、近似的に計算する方法である。具体的に書いてみる。

1. $M = 1, 2, \cdots, 7$ として次のことを行う。
   (a) $R = \lceil \frac{\pi}{4} \sqrt{\frac{N}{M}} \rceil$ とする。
   (b) この R を Grover iteration を繰り返す回数として、Grover のアルゴリズムを計算する。
   (c) 出てきた結果に対して、古典コンピュータ的手法を用いて、求めたい結果かどうかを計算し、求めたい結果であるならば、計算を終了する。

この手法を用いれば、明らかに、解の数が $1, 2, \cdot, 7$ のときに関しては、ほぼ 1 の確率で、答えを出すことができる。実は、解の数が 8 以上のときも、高確率で計算することができる。それは、Grover iteration が回転を表すからである。仮に解の個数が 100 個だったとして、M の数を適当 ($M < 100$) に決めて計算を行うと、図的には、$\frac{\pi}{4}\sqrt{\frac{N}{100}}$ 回の計算で 90 度回転するので、傾きは 90 度を越えて、$\frac{\pi}{2}\sqrt{\frac{100}{M}}$ の傾きほど回転することになり、大体円状に一様の確率でどこかに存在することになる。これを観測すると、それなりの確率で正しい計算結果が計算できることが推測される。具体的にどれくらいの確率になるのか計算してみる。

まず、解の個数が $1, 2, \cdots, 7$ のときについて考える。図において垂直に起き上がるほどの回数の Grover iteration を計算すれば、多少のズレはあるが、答えの出る確率はほぼ 1 になる。そのズレを計算すると、前述の図において、垂直に起き上がっているときより、θ だけ傾いているときのが、ズレの最大となるので、M が解の個数のときに、答えの出る確率は、

$$sin^2\left(\frac{\pi}{2} - \theta\right) = cos^2\theta \tag{116}$$

$$= \left(1 - \frac{2M}{N}\right)^2 \tag{117}$$

$$< \frac{1}{2} \qquad \left(\frac{M}{N} < \frac{1}{4}\right) \tag{118}$$

また、解の個数が 8 個以上の場合は、一回計算し終わったときに、図において円状に一様に分布していると仮定すると、正しい答えが得られる確率は

$$\frac{1}{2\pi}\int_0^{2\pi} sin^2 x dx = \frac{1}{2} \tag{119}$$

となるので、7 回の計算を繰り返すと、解の個数がいくつであるかに関わらず、

$$1 - \left(1 - \frac{1}{2}\right)^7 > 0.99 \tag{120}$$

の確率で正しい答えを得ることができる。

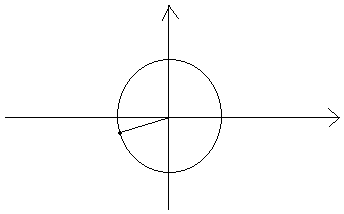

図 10　図

また、このとき計算する Grover iteration の数は、(69) より、

$$R' \leq \sum_{M=1}^{7} \lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \rceil \tag{121}$$

$$\leq \sum_{M=1}^{7} \frac{\pi}{4}\sqrt{\frac{N}{M}} + 7 \tag{122}$$

$$\leq 3.16\sqrt{N} + 7 \tag{123}$$

となる。ゼータ関数のような形をしてるのも面白い。

私は $M = 1, 2, \cdots, 7$ として、計算を行ったが、解の個数の分布がすでにわかっている場合には、さらにいい手法がある可能性がある。私のアルゴリズムにおいては、N の数をどんどん増やしながら計算していく方法が有効ではないかと考えられるため、解の個数が少ない場合に計算できることが重要であり、このような手法を用いた。

4.3.7 最適解の求め方

考え方としては難しくないが、一応、私の最適解の求め方を書いておく。変数には整数型と浮動小数点数型があるが、議論を簡単にするために、整数型として議論を進めるが、浮動小数点数に関しても、大きく問題が変わるわけではない。m qubit を用いて、2^m 通りの整数の値を取れる評価値を入れる変数を用意する。ここで考えたい問題は、n qubit と Hadamard 変換を用いて、2^n 通りの状態に対して、あるアルゴリズムを動作させ、m qubit の評価値を計算し、その 2^n 通りの m qubit のうち、最も値の高い状態を結果として出力することを考える。私が考えるアルゴリズムは次のようなものである。

1. 閾値用の変数を 3 つ用意し T, T_{min},T_{max} とし、T_{min} に評価値の最小値、T_{max} に評価値の最大値、T に $(T_{max} + T_{min})/2$ を入れる。
2. Grover のアルゴリズムを適用し、Grover iteration を繰り返すのであるが、関数のところは次のように計算する。
   （a）解説してきた手法を用いて、2^n 通りの評価値を m qubit に入れる。
   （b）別の 1qubit に、評価値が閾値 T 以上ならば 1, 閾値未満なら 0 を入力する。
3. Grover のアルゴリズムの結果が正しいか検算をする。閾値以上の状態でなければ、閾値以上の状態が存在しないと仮定して、$T_{max} \leftarrow T$、閾値以上なら、閾値以上の状態が存在するので、$T_{min} \leftarrow T$ として、また $T \leftarrow (T_{max} + T_{min})/2$ とする。
4. $T_{max} = T_{min}$ でないなら、(2) へ戻る。

この手法を用いれば、$O(m\sqrt{2^n})$ の計算時間で最適解を求めることができる。さらに速い計算はできないのだろうか。しかし、量子探索が最低でも $O(\sqrt{2^n})$ の計算時間がかかるとすると、もし、たとえば最適解を $O(logn)$ などの時間で求められるとすると、m=1 のときの最適解を計算する手法を用いることで、量子探索がさらに高速に計算できるため、背理法的に最適解の計算は基本的にこれ以上のスピードアップは見込めない。一応これらを定理としてまとめておく。

定理 1　量子探索と量子最適解探索の計算量のオーダーは同じである。

4.3.8 もっと速く計算できないか

前の節にて、量子探索には情報の数 2^n 個に対して、$O(\sqrt{2^n})$ ぐらいの時間がかかることを証明した。これが $O(log(2^n))$ ならば、様々な面白いソフトウェアを作ることができる可能性があったのだが、この結果は現実的である。私は量子探索の解釈の違いや、証明に間違いがある可能性を考え、さらに高速に量子探索を行う方法を考えることを試みたところ、次のような量子探索アルゴリズムを思いついた。基本的なアイデアはトーナメント方式にして、計算量を対数時間にまで下げるというものである。とりあえず、一番単純な例である 2 段階勝ち抜き方式について記述する。

1.（a）入力用の n qubit を 2 つにわけ、$n = n_1 + n_2$ という規定のもと、変数 in0 を $n_0$qubit と変数 in1 を $n_1$qubit を設定する。
（b）in0、in1 に Hadamard 変換で全ての状態を入力して関数 f を計算し、out に計算結果を出力する。
（c）in1, out に関して、量子計数を適用し、out が $|1\rangle$ になっている個数を count に入力する。
（d）count を用いて、Grover のアルゴリズムを適用し、それを in1' に入れる。（ここで in0, in1' を観測すると、ランダムな in0 に対して解があるときは、in1' に解が入っているという感じになっている。）
2.（a）in0 と in1' を入力して関数 f を計算し、out に計算結果を出力する。

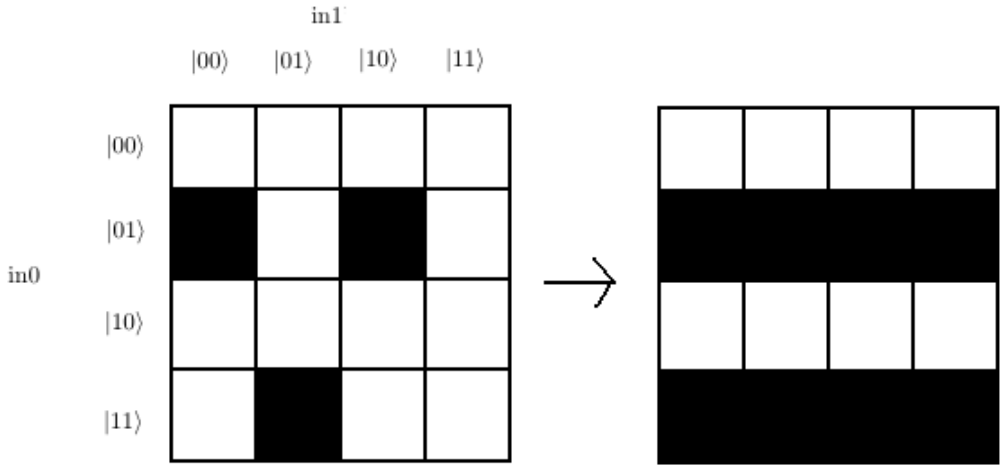

図 11　イメージ図：塗りつぶしてあるところが out が $|1\rangle$ のところ。in1' 計算後は、解が存在するとき、in1 がなんであるかに関わらず、out は 1 となる。

（b）in0, out に関して、量子計数を適用し、out が $|1\rangle$ になっている個数を count に入力する。
（c）count を用いて、Grover のアルゴリズムを適用し、それを in0' に入れる。
3.（a）in0 に in0'、in1 に全ての状態を入力して関数 f を計算し、out に計算結果を出力する。
（b）in1, out に関して、量子計数を適用し、out が $|1\rangle$ になっている個数を count に入力する。
（c）count を用いて、Grover のアルゴリズムを適用し、それを in1' に入れる。
（d）in0'、in1' を最適解とする。

この手法が実現すれば、量子探索は $O(\sqrt{2^{n_1+n_2}})$ ではなく、$O(\sqrt{2^{n_1}} + \sqrt{2^{n_2}})$ となり、スピードアップが望める。これが基本的な原理である。しかし、これは確率的な手順である量子計数を含んでおり、まだ曖昧でこれだけでは上手く計算できるかわからない。そこでこれが二段階ではなく、変数を変えて、$n_1$qubit を固まりを、n_2 段階の操作としたとき（$n = n_1 n_2$）の一般的なことも含めて、実際に確率的に計算できるか評価を行った。ϵ を量子計数一回の失敗確率とし、Grover のアルゴリズムが計算できる確率を p とすると、一般的なアルゴリズムに関しては、量子計数と Grover のアルゴリズムの操作を $1 + 2 + 3 + \cdots + n_2$ 回計算する必要があるため、最終的な計算できる確率は、

$$P = ((1-\epsilon)p)^{\frac{n_2(n_2+1)}{2}} \tag{124}$$

となる。これを変形すると、

$$\frac{1}{\epsilon} = \frac{1}{1 - exp(-logp + \frac{2logP}{n_2(n_2+1)})} \tag{125}$$

ここで p について深く考える。前の節からやった内容からすれば、図において最も起き上がったときから Grover iteration の θ だけ傾いたときが最もエラーの大きい状態である。また突然ではあるが、M は十分小さいと仮定してある（私の問題においてはこのように設定してもそんなに問題はない。）

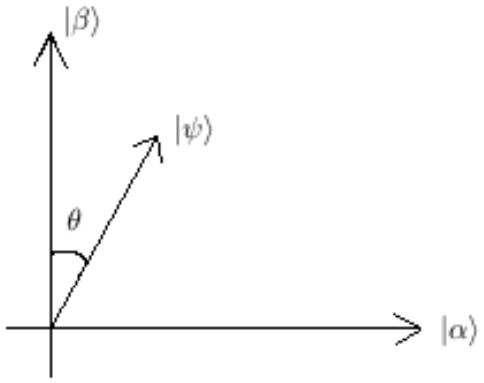

図 12　エラーの大きいとき

よって、

$$p = sin^2\left(\frac{\pi}{2} - \theta\right) \tag{126}$$
$$= cos^2\theta \tag{127}$$
$$= \left(1 - \frac{2M}{N}\right)^2 \tag{128}$$
$$= 1 - \frac{M}{N} + \frac{4M^2}{N^2} \tag{129}$$
$$= 1 - \frac{M}{2_1^n} + \frac{4M^2}{2^{2n_1}} \tag{130}$$

となり、ここで $n_2 = log_2 n_1$ とすると、

$$\frac{1}{\epsilon} = \frac{1}{1 - exp\left(-log(1 - \frac{M}{2^{n_1}} + \frac{4M^2}{2^{2n_1}}) + \frac{2logP}{n_2(n_2+1)}\right)} \tag{131}$$
$$= \frac{-log(1 - \frac{M}{2^{n_1}} + \frac{4M^2}{2^{2n_1}}) + \frac{2logP}{n_2(n_2+1)}}{1 - exp\left(-log(1 - \frac{M}{2^{n_1}} + \frac{4M^2}{2^{2n_1}}) + \frac{2logP}{n_2(n_2+1)}\right)} \tag{132}$$
$$\cdot \frac{1}{-log(1 - \frac{M}{2^{n_1}} + \frac{4M^2}{2^{2n_1}}) + \frac{2logP}{n_2(n_2+1)}} \tag{133}$$
$$\to \frac{-1}{-log(1 - \frac{M}{2^{n_1}} + \frac{4M^2}{2^{2n_1}}) + \frac{2logP}{n_2(n_2+1)}} \quad (n_2 \to \infty) \tag{134}$$
$$= \frac{-1}{\frac{-\left(-\frac{M}{2^{n_1}} + \frac{4M^2}{2^{2n_1}}\right) log(1 - \frac{M}{2^{n_1}} + \frac{4M^2}{2^{2n_1}})}{-\frac{M}{2^{n_1}} + \frac{4M^2}{2^{2n_1}}} + \frac{2logP}{n_2(n_2+1)}} \tag{135}$$

$$\to \frac{-1}{\frac{M}{2^{n_1}} - \frac{4M^2}{2^{2n_1}} + \frac{2logP}{n_2(n_2+1)}} \quad (n_2 \to \infty) \tag{136}$$

$$\to -\frac{n_2(n_2+1)}{2logP} \quad (n_2 \to \infty) \tag{137}$$

(105) の表記を用いると、量子計数一回に用いる計算量は、

$$2^t - 1 = 2^{m+\lceil log(2+\frac{1}{2\epsilon})\rceil} - 1 \tag{138}$$

$$\simeq 2^m \left(2 + \frac{1}{2\epsilon}\right)^{log2} \tag{139}$$

$$\simeq \frac{2^m}{(-4logP)^{log2}} n_2^{2log2} \tag{140}$$

ここで誤差を規定する変数 m を前述で議論したときと同じように $\frac{n_2}{2}$ で近似すると、

$$\frac{2^{\frac{n_2}{2}}}{(-4logP)^{log2}} n_2^{2log2} = \frac{1}{(-4logP)^{log2}} 2^{\frac{n_2}{2}} n_2^{2log2} \tag{141}$$

$$= \frac{1}{(-4logP)^{log2}} n_1^{\frac{1}{2}} (log_2 n_1)^{2log2} \tag{142}$$

となるので、全体の計算量は、A,B を定数として、

$$(A\sqrt{2^{n_2}} + B n_1^{\frac{1}{2}} (log_2 n_1)^{2log2}) \frac{n_1(n_1+1)}{2} \tag{143}$$

$$= (A n_1^{\frac{1}{2}} + B n_1^{\frac{1}{2}} (log_2 n_1)^{2log2}) \frac{n_1(n_1+1)}{2} \tag{144}$$

$$\simeq O((log_2 n_1)^{2log2} n_1^{\frac{5}{2}}) \tag{145}$$

ここで全体の並列度と対応をつけるため、並列度 $N = g(n_1)$ として計算を行うと、

$$g(n_1) = 2^{n_1 n_2} \tag{146}$$

$$= n_1^{n_1} \tag{147}$$

$$> 2^{n_1} \quad (n_1 \to \infty) \tag{148}$$

$$log_2 g(n_1) > n_1 \tag{149}$$

よって計算量は、

$$O((log_2 n_1)^{2log2} n_1^{\frac{5}{2}}) < O((log_2 log_2 N)^{2log2} (log_2 N)^{\frac{5}{2}}) \tag{150}$$

となり、平方根の時間ではなく、対数時間となった。

私は、このアルゴリズムを開発して、もしかしたらもっと速くなるのではないかと考え、もう一つ高速量子探索アルゴリズムを考えるに至った。それは次のようなものである。

1. 入力用の n qubit 変数である in と、出力用の 1qubit 変数である out について考える。
2. in に Hadamard 変換で全ての状態を入力し、out を計算する。
3. in の n qubit を一つづつそれぞれに対して、次の操作を行う。
   （a）すでに後の操作で解が $in_0, \cdots, in_{i-1}$ についてはわかっているとしたら、in_i の解を 0 として in と解の排他的論理和の否定を計算し、その i+1qubit の計算結果の積を計算する。最後のそれと out の AND を計算して out2 にいれる。

（b）out2 に 1 のものがあるかを考えるため、次の操作を行う。

（c）in の全ての qubit に対して、次の操作を行う。選んだ 1qubit に対して値が 0 のときと 1 のときの out2 の和を計算し、out2 に入れる。

（d）ここで out2 を観測すると、1 のものがあったかの結果がわかるので、あったならば、in_i の解を 0 とする。ないならば解を 1 とする。

4. 最終的に全ての解が出力される。

図で示すと以下のようになる。

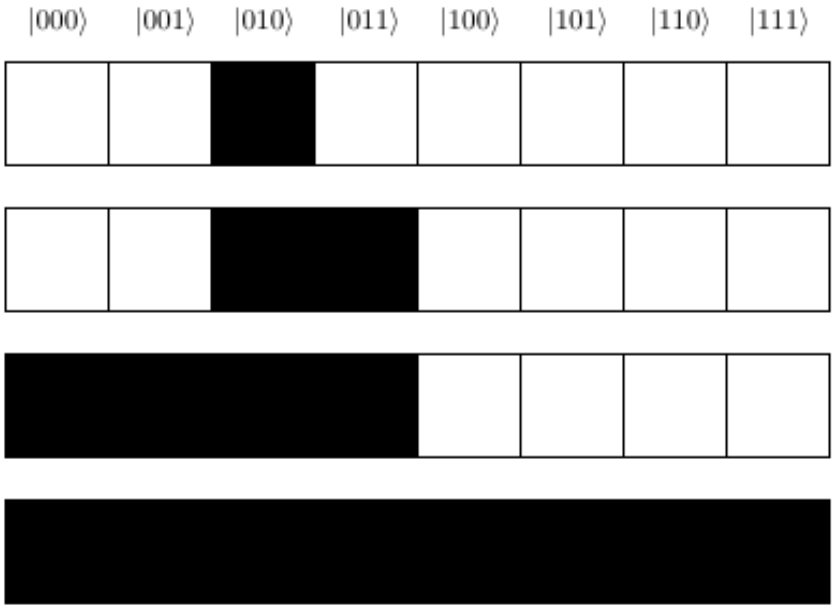

図 13　3(c) のイメージ図

計算量を考えると、3 の操作を n 回、3(c) の操作を n 回繰り返す必要があるので、$N = 2^n$ として、おおよそ $O(n^2) = O((logN)^2)$ となり、この手法を用いても対数時間で量子探索を行えることになる。一体、何が正しいのだろうか。そこで私は証明と自作のアルゴリズムのあらをくまなく間違いを探すことにしたところ、私のアルゴリズムに怪しいところがあることを発見することになった。まず、後者のアルゴリズに関してだが、3(c) のところで和をとっているのだが、これは 4.2.4 で説明した OR とは異なるものであるということである。具体的に書くと、4.2.4 での OR は、

$$|x_1\rangle|x_2\rangle|y\rangle \rightarrow |x_1\rangle|x_2\rangle|x_1 or x_2\rangle \tag{151}$$

であるが、このアルゴリズムでは、

$$(|0\rangle|x_1\rangle + |1\rangle|x_2\rangle)|y\rangle \rightarrow (|0\rangle|x_1\rangle + |1\rangle|x_2\rangle)|x_1 or x_2\rangle \tag{152}$$

を行わなければならない。この操作はできないのだろうか。私はこの操作の具体的な手法について考えるため、次のように問題を変えた。3qubit ではなく、2qubit に縮小し、

$$|0\rangle|x_1\rangle + |1\rangle|x_2\rangle \to |0\rangle|x_1 or x_2\rangle + |1\rangle|x_1 or x_2\rangle \tag{153}$$

を考えると、

$$|00\rangle + |10\rangle \to |00\rangle + |10\rangle \tag{154}$$
$$|00\rangle + |11\rangle \to |01\rangle + |11\rangle \tag{155}$$
$$|01\rangle + |10\rangle \to |01\rangle + |11\rangle \tag{156}$$
$$|01\rangle + |11\rangle \to |01\rangle + |11\rangle \tag{157}$$

というのを変換できるユニタリ行列を考えればよく、その存在の可能性によって、このアルゴリズムが機能するかという話に帰着することができる。この行列は 4×4 行列であり 16 個の変数がある。またユニタリ行列であるため、実行列と考えると、10 個の式で制約されており、さらに、さきほどの 4 つの式で 8 つの式で制約される。この時点で $16 < 10 + 8$ であまり望みが無さそうであるが、この方程式を解くということを考えてみた。ノートに計算してみたのだが、どうも上手くいかず、コンピュータを用いて、最急降下法のような手法で近似的にその解の可能性を探ってみた。具体的に書いてみる。

1. 求めたい行列の変数の初期値を（たとえば全部 0）に設定する。
2. 以下を評価関数が収束するまで何回も繰り返す。
   （a）変数それぞれに対して、変数の値をわずかに上げたときと下げた場合について評価関数を計算する。評価関数は、前述の 18 個の式を「=0」の形にしたときの、18 個の左辺の和である。
   （b）評価関数が下がったなら、変数の値を変更後の値にする。

これを行ったところ、パラメータの設定を変更しても評価関数は 0 には収束せず、望みの行列を見つけ出すことはできなかった。また 3qubit の操作についても同様のことを行ったが結果は収束しなかった。私はこの結果より、私のアルゴリズムが間違っていて、量子探索は私の作ろうとしてるアルゴリズムの解決に関しても $O(\sqrt{2^n})$ 以上のスピードアップは見込めないと判断した。結果は失敗であるが、一つの帰結を得ることができる。私のアルゴリズムが可能である、すなわち、前述したような OR の計算ができると仮定すると、量子探索がより高速に行え、前述した証明に矛盾してしまう。背理法により、その OR の不可能性が証明できたことになる。定理としてまとめておくと、

定理 2

$$(|0\rangle|x_1\rangle + |1\rangle|x_2\rangle)|y\rangle \to (|0\rangle|x_1\rangle + |1\rangle|x_2\rangle)|x_1 or x_2\rangle \tag{158}$$

は計算不能である。

また、AND が計算できると仮定すると、OR を計算することができてしまうために、AND も計算することができない。

前者のアルゴリズムはどうして対数時間で計算できるのだろうか。それは 1 でいうと、(c)(d) に隠されている。このアルゴリズムは、関数を計算したあと、その関数が 1 になるものを強く観測できるように起き上がらせているが、Grover のアルゴリズムの説明を読んでみると、Grover iteration で関数を計算するときに、計

算しおえた関数の結果を使って計算できるかについては何も言っておらず、そこをこのアルゴリズムの怪しい点とした。具体的に数式で書くならば、

$$|x_j\rangle \left(\sum_i |x_i\rangle|y_i\rangle \right) |z\rangle \to |x_j\rangle \left(\sum_i |x_i\rangle|y_i\rangle \right) |y_j\rangle \tag{159}$$

ができるかが怪しいということである。結局、量子探索には $\sqrt{2^n}$ の計算量が必要だから、これが不可能であるということもできるが、これができると定理 2 の計算不能な計算ができることを示して、強く証明することにする。

(159) の x, y, z それぞれを 1qubit として考える。x_j に 0 を入力したときと、1 を入力したとき、それぞれに関して計算を行い、計算された最後の qubit の和をとると、定理 2 の計算が行える。これで、これが計算可能だと定理 2 の計算が可能だということが示される。定理 2 は計算不能性を示したものであり、また対偶を考えると、この計算が、実験的な手法によっても不可能であることが示された。よって、定理としてまとめると、

定理 3

$$|x_j\rangle \left(\sum_i |x_i\rangle|y_i\rangle \right) |z\rangle \to |x_j\rangle \left(\sum_i |x_i\rangle|y_i\rangle \right) |y_j\rangle \tag{160}$$

は計算不能である。

これらの定理は現実的な成果であるが、現実的に量子ソフトウェアを開発するときに使用する概念としてはかかせないのではないかと考えている。また、前述したときの概算では、Grover iteration に時間がほとんどかからないと仮定し、2^{100} 通りが並列に計算できると論じたが、この定理から、Grover iteration 内で計算したい関数をいちいち計算する必要があることがわかり、私のアルゴリズムにおいて、実際の並列度は、2^{40} ぐらいではないかと結論できる。

4.4 量子ソフトウェアの可能性

4.4.1 作曲の可能性

これから具体的な可能性の議論に入らさせていただくが、作曲に関していえば、脳がシミュレーションできることを前提にしている。脳は 2019 年にはシミュレーションできるようになるという記事 [2] も存在し、現在できないからだめというわけではなく、将来的な可能性として論じている。脳の仕組みはまだわかっていないことだらけなのであるが、基本はニューロン細胞と呼ばれる神経細胞の集まりでできている。

電気信号が細長いところを伝わっていくという構造となっている。具体的に電圧を観測すると、下図のようになり、パルス波が何回も来るような形であることがわかる。

また、神経細胞間の電位の伝播の基本的なモデルとしてニューラルネットワークというものが知られており、これは下図のように、信号が丸のところで足しあわされたあと、非線形な変換を行い、さらに伝播を続けるというものである。

実際の神経回路はさらに複雑なものであると予想されるが、物理法則に則っているので、物理シミュレータを用いれば、計算できるはずである。ここでは、格子法と、粒子法と、ネットワーク法というよく使われる 3 つの方法に関して、量子コンピュータが並列性を生かして、これらの計算を行おうことができるのかについて考える。

私が最初に思いついたのは格子法である。脳を 2^n 個のサイコロになるように豆腐を切るように分解し計算

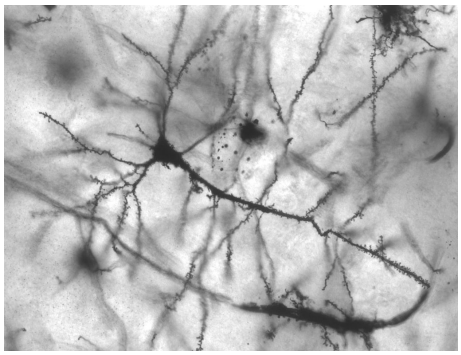

図 14　ニューロン細胞

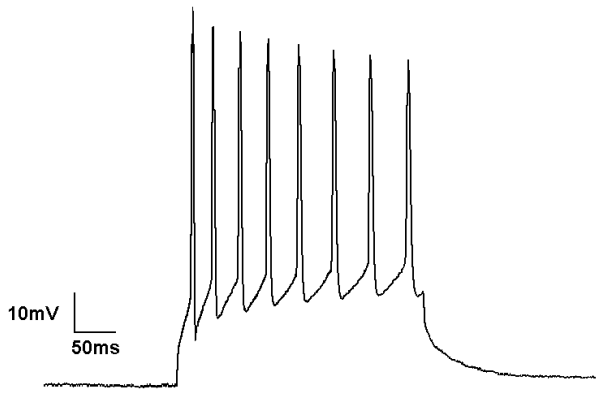

図 15　ニューロン細胞の電位

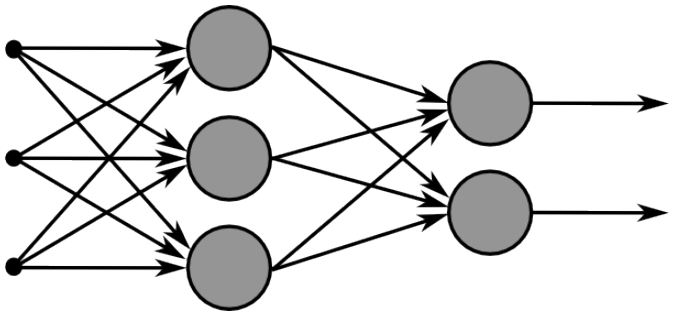

図 16　ニューラルネットワーク

を行えば、脳の長さの対数の qubit で計算が行えるのではないかと考えた。つまり、位置を (x,y,z)、そこにある脳のデータを data としたとき、

$$\sum_i |x_i\rangle|y_i\rangle|z_i\rangle|data_i\rangle \tag{161}$$

を順次更新していく操作について考える。しかし、定理 2 より、2 つの位置におけるデータの相互作用は AND、OR すら計算することはできないので、格子法を量子コンピュータで計算するにあたって、その並列性を生かした計算方法は存在しないことになる。

粒子法についてであるが、これは格子法と違って、位置 $|0\rangle$ と $|1\rangle$ におけるデータでの関数を計算するという手法ではなく、位置から距離を計算して近いもの同士の相互関係を計算するというものである。しかし、粒子の位置が格子だとすると、格子法の計算を効率的に行うことができるようになってしまうため、粒子法に関しても並列性を生かした計算方法は存在しない。

最後にネットワーク法であるが、これは脳のニューロン細胞がネットワーク構造であることを利用して、具体的な位置を気にせず、繋がっている関係だけで記述する方法である。具体的には、$x \to y$ として、

$$\sum_i \left(|x_i\rangle \sum_j |y_{ij}\rangle \right) \tag{162}$$

として記述する。ここで $|x_k\rangle$ を入力して、そこからつながっている細胞を探したいのだが、これも定理 3 によって不可能であることがわかる。

以上より、脳 1 つを量子コンピュータで計算するときにおいて、量子コンピュータの並列性を生かして計算することはできないと判断する。では、この作曲を実現させようと思うと、現実的にはどのようなことになるのか。2^{40} ほどの並列度なら、複数の脳を並列に計算できる可能性についてはまだ否定できないので、私は、脳を 3 次元プリンタのような装置をを用いて、量子コンピュータとしてプリントアウトできれば、まだ高速に脳を動かす可能性はあると考えている。作曲に関して言えば、全ての曲を検索するのではなく、ある程度曲を一般のプログラミングの手法に従って作曲し、できた 2^{40} 通りの曲を検索すれば、現実的とはいえ、新しいジャンルの作曲を行える可能性がある。また、単純に 1 度検索をかけるのではなく、問題を分割して、何度も繰り返し検索するような手法により、作曲に関して大きな可能性があると考えている。

4.4.2 自動定理証明の可能性

自動定理証明に関しても、前述した脳のシミュレーションの手法が使えると考えられるが、完全な論理を計算してくれるシミュレーションはまだ先の話と考えられるため、効率の観点からも量子で、現在のコンピュータと同様の計算手法で計算できないかと考えた。現在のコンピュータを並列に動かすことができれば、2^{40} 通りの証明を入力する方法によって、現在のコンピュータより高速に自動定理証明を行える可能性が見えてくる。現在のゲート式量子コンピュータが高々 10qubit 程度であることを考え、私はまず、量子アドレスによる量子 CPU を考えた。具体的には、

$$\sum_i |address_i\rangle|data_i\rangle \tag{163}$$

となるような機械語列を計算してくれるコンピュータである。しかし、この手法では定理 2 で示したとおり、コンピュータの基本演算である AND や OR も計算することができないし、定理 3 で示したとおり、アドレス参照も不可能であるため、自動定理証明という複雑なプログラムはこの手法では書けないと考えられる。そこ

で私は、このような効率的手法をやめ、現在のコンピュータと同じデータ格納方法について考える。すなわち、

$$|data_1\rangle|data_2\rangle\cdots|data_n\rangle \tag{164}$$

といった具合である。しかし、これが既存の手法と違うのは、2^{40} 通りほどのデータを一つの qubit で格納できることである。このような手法を使えば、NAND が計算できるので、コンピュータの計算部分は処理可能であるし、長谷川先生によると量子メモリは可能ということなので、このような量子 CPU を作成すれば、自動定理証明が高速に計算できると考えられる。

4.5　量子コンピュータ実現に向けて

4.5.1　はじめに

量子コンピュータを実現する手法としては、核磁気共鳴、量子光学、量子ドット、超伝導素子、イオントラップなどの手法が考えられている。ここでは、イオントラップと、量子光学による実現手法について取り上げる。

4.5.2　イオントラップによる実現方法

イオントラップとは、イオン数個を真空の装置内に浮かせる手法のことである。イオン 1 つ 1 つを qubit として量子計算を行うことができる。この手法の良いところは、量子計算が高速にできる (一つの操作が 10^{-14} 秒くらい) 可能性があるとともに、量子状態が崩壊するまでの時間、すなわちデコヒーレンスの時間が長い (10^{-1}秒くらい) ことである。

具体的に初期設定の仕方から説明する。まず、装置内の空気の抜かなければならないのだが、高い真空度が要求され、これには数日程度の日数を要する。次にカルシウムといったイオンにする物質を暖め、イオン化させると、電荷を持つので、電界を上手く操作すれば、イオンのみを動かすことができるようになり、それを利用して、装置内にある四重極と呼ばれるところまで移動させることができる。

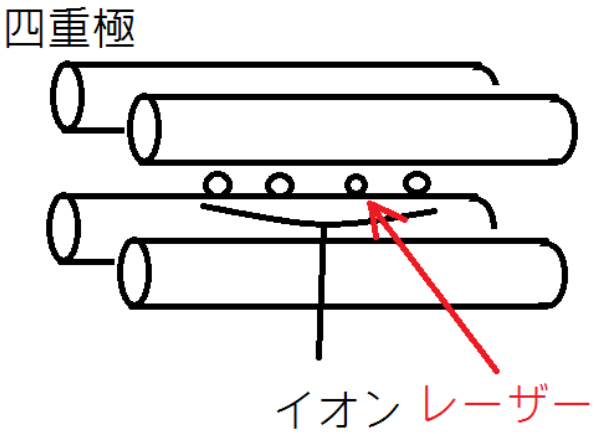

図 17　イオントラップ

四重極には正弦波的な電圧が、

$$\phi_0 = U_0 + V_0 cos(\omega t) \tag{165}$$

のように流れており、イオンを浮かせることができる。また、ポテンシャルを上手に動かす手法により、膨大な数のイオンを数個にまで落とすことができる。イオンの移動が完了したら、次はイオンを qubit として初期値 $|0\rangle$ にするために、冷却を行わなければならない。冷却には 2 種類あって、まず、ドップラー冷却というものを行う。これは動いているイオンに対し、レーザーを照射して、レーザーの方向にイオンの速度を変化させ、イオンを静止させる操作である。その次に行うのがサイドバンド冷却というもので、これはさらに量子力学レベルでの微小な速度を、同じくレーザーによって、イオンのエネルギー準位をあげたあと、別のエネルギー準位に遷移することを利用して、下げるというものである。

次に実際の量子計算、すなわち 1qubit に対する Hadamard 変換と、2qubit に対する制御 NOT を説明したいのだが、文献 [3] によると、かなり煩雑なため、大まかな手順のみを説明する。詳細はその文献を参照されたい。操作はレーザーをあてればいい。レーザーの強度、周波数、位相、イオンの位置、電荷を引数とした関数 l が存在し、それと、照射するイオン j と前述したレーザーの位相 ϕ_j を引数とするレーザーの操作 $A_j^l(\phi_j)$ と $B_j^{l,I}(\phi_j)$ と $B_j^{l,II}(\phi_j)$ が存在する。それを用いると、

$$A_j^{\frac{1}{2}}(0) \tag{166}$$

で、Hadamard 変換が実現し、

$$A_{m_2}^{\frac{1}{2}}(\pi) B_{m_1}^{1,I} B_{m_2}^{2,II} B_{m_1}^{1,I} A_{m_2}^{\frac{1}{2}}(0) \tag{167}$$

によって、イオン m_1 と m_2 の制御 NOT 操作を行うことができる。また、[3] は私の書いてきた手法とは違うが、

$$A_{m_{q+1}}^{\frac{1}{2}}(\pi) B_{m_1}^{1,I} \left(\prod_{j=2}^{q} B_{m_j}^{1,II} \right) B_{m_{q+1}}^{2,II} \left(\prod_{j=q}^{2} B_{m_j}^{1,II} \right) A_{m_{q+1}}^{\frac{1}{2}}(0) \tag{168}$$

とすれば、m_{q+1} と $m_1 \cdots m_q$ の積の排他的論理和が m_{q+1} に入り、NAND が計算できるため、私の論じてきたものがこの手法で全て計算できることになる。

4.5.3 量子光学による実現方法

続いて量子光学による実現手法について説明する。この手法においては、光を伝える粒子である光子を qubit として用いる。$|0\rangle$ か $|1\rangle$ かは、2 つのレーザーを用い、その光子の振動状態 $|0\rangle$ と $|1\rangle$ に対し、

$$|0_L\rangle = |0\rangle_1 \otimes |1\rangle_2 、 |1_L\rangle = |1\rangle_1 \otimes |0\rangle_2 \tag{169}$$

として定義を行う。この手法についても、1qubit の Hadamard 変換と 2qubit の制御 NOT について解説する。Hadamard 変換に関しては、ビームスプリッターというものに qubit として用いる光子をレーザー 2 つをあてて、出力を見ればよい。具体的には次のようなものである。

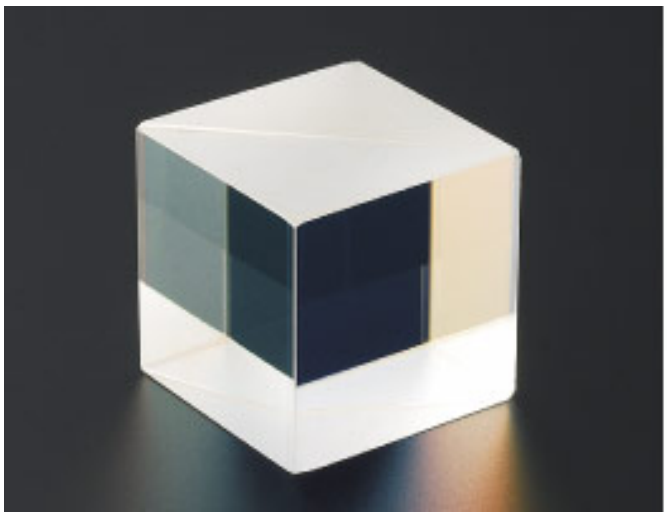

図 18　ビームスプリッター

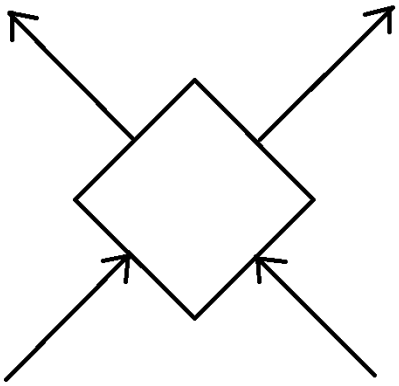

図 19　ビームスプリッターの入出力図

制御 NOT は全反射ミラーと光カー効果を用いて構成する。説明に用いる qubit の状態は先ほど説明したのと違い、

$$|0_L\rangle_1 = |0\rangle_1 \tag{170}$$
$$|1_L\rangle_1 = |1\rangle_1 \tag{171}$$
$$|0_L\rangle_2 = |0\rangle_2 \otimes |1\rangle_3 \tag{172}$$
$$|1_L\rangle_2 = |1\rangle_2 \otimes |0\rangle_3 \tag{173}$$

を用いると、

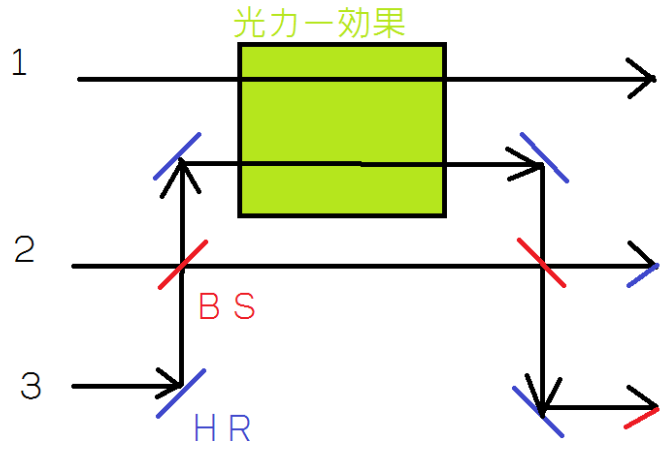

図 20　制御 NOT：BS はビームスプリッター、HR は全反射ミラー

のようにすれば、制御 NOT を計算できる。さらに位相シフターなども存在し、$\frac{\pi}{8}$ ゲートなども作れるため、全ての量子計算が行えることになる。詳細は [6] などを参照されたい。

4.5.4　誤り訂正

前述したように、量子状態はほかっておくとすぐ壊れてしまうので、長い計算はそのままでは行うことができない。壊れ方には、$|0\rangle$ から $|1\rangle$ もしくは、$|1\rangle$ から $|0\rangle$ に揺れ動いてしまう bit flip と、$|0\rangle$ が $-|0\rangle$ になってしまうような phase flip がある。このようなエラーを防ぐには、状態を保持したい qubit を複数複製しておき、計算したあと、複製した qubit で状態の多数決を行えばよい。こうすれば、少しデータが壊れたとしても復元を行うことができる。たとえば、次のようなものを用いればよい。E が計算である。

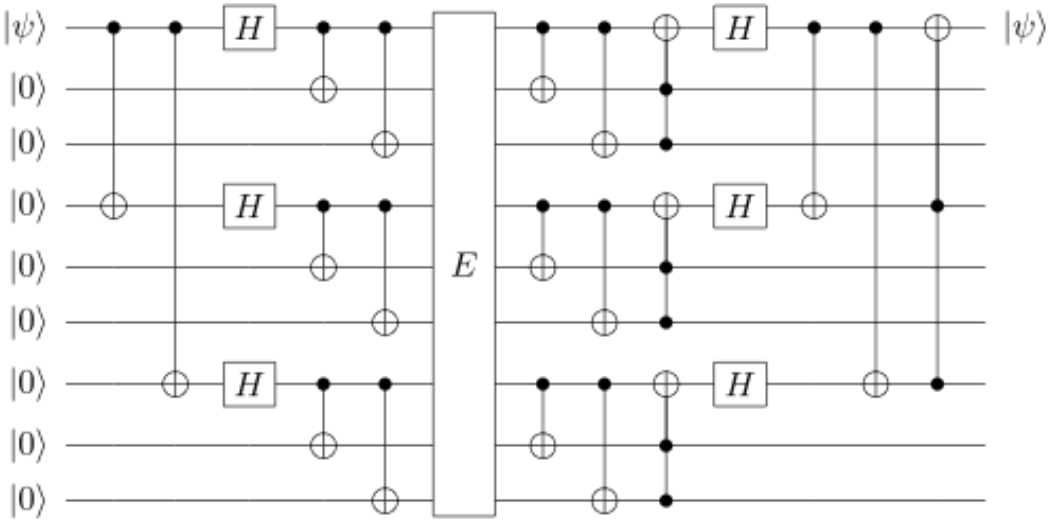

図 21　Shor code

4.5.5　量子焼きなまし法

近年、D-Wave などが量子焼きなまし法を用いて、512qubit の量子コンピュータを完成させたことが話題になっている。この量子コンピュータは私の説明してきたゲート式とは全く別の手法をとっており、私の求めているものが高速に計算できるのではないかとも考え、探ってみることにした。

量子焼きなまし法は最適解を探すための手法である。その関数は私の述べてきたような手法ではなく、基本的なアイデアは、量子を動かすポテンシャルを関数とし、そのポテンシャル関数の小さな位置をみつけるというものである [7]。具体例で説明する。まず、下図の太線のように、ポテンシャルとして関数がわりあてられているとする。

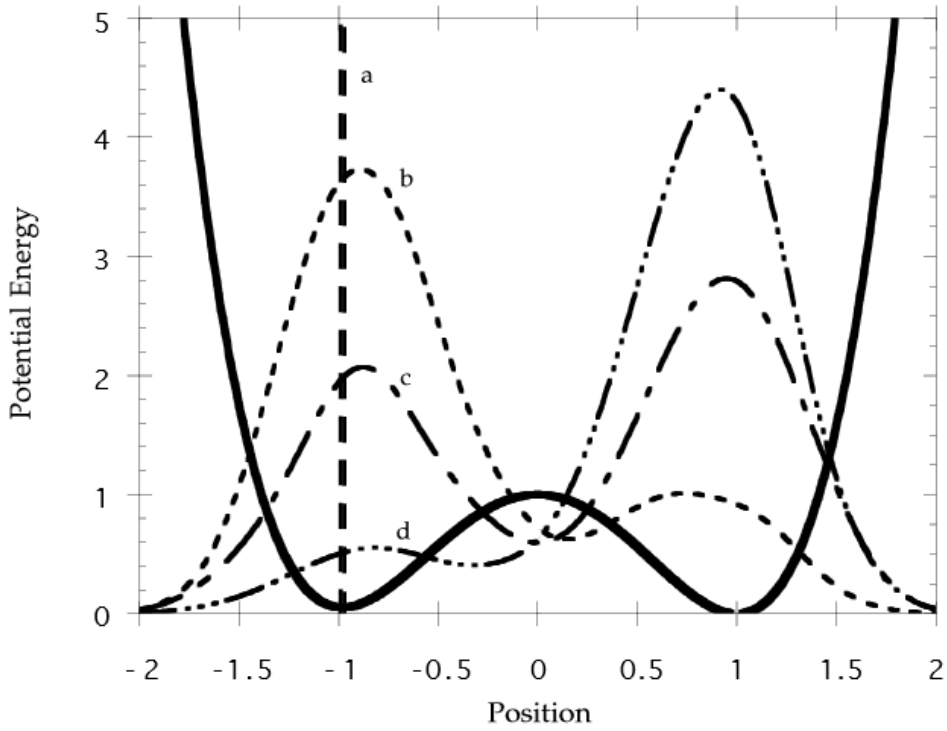

図 22　量子焼きなまし

これが量子ほど小さな世界でなくても、ボールをこの曲線上で転がし、最後に位置を観測すれば、局所最適解は探しだせるのがわかる。これに関しては量子のような小さな世界でも、ポテンシャルを V(x) として、

$$\frac{\partial \Psi}{\partial \tau} = \frac{\hbar^2}{2m}\frac{\partial^2 \Psi}{\partial x^2} - (V(x) - E_0)\Psi \tag{174}$$

というシュレディンガー方程式で動きは記述でき、やることは同じである。しかし、量子の世界は、そのボールに相当する量子の位置がはっきりしておらず、局所最適解として収束したあとも、この方程式に従うとすると、図の $a \to b \to c \to d$ というように遷移を続け、最終的に Position が 1 の大域最適解に染み出すように収

束させることができる。これを用いて、さらに複雑な問題を解決しようとするわけだが、その技術は難解なので、性能の結果だけを示すことにする。

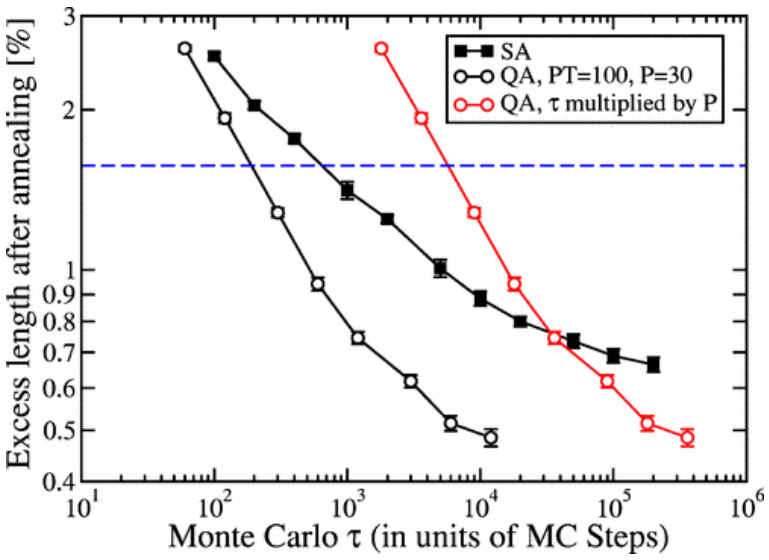

図 23　巡回セールスマン問題における性能評価

この図 [8] は、巡回セールスマン問題という、有名なシンプルな最適化問題に対して、モンテカルロ法的に量子焼きなまし法をシミュレーションにて行ったときの性能が表れている。縦軸が、最適解からのずれで、横軸が時間である。これを見ると、黒四角の古典コンピュータに比べて、白丸の量子焼きなまし法は、同じ性能の計算結果までに行き着くまでの時間がおおよそ「平方根」ぐらいとなっている。計算しおえた関数を用いて非ユニタリな高速量子最適解探索ができるのではないかという期待がさきほどの量子焼きなまし法の理解よりあったのだが、この図を見ると結局ユニタリな計算しかできないことがわかる。

4.6　非ユニタリな量子計算

私の論じてきた手法は量子計算のユニタリ性より制限を受けており、量子探索は $O(\sqrt{2^n})$ の計算量を必要とした。しかし、非ユニタリな量子計算の可能性を追ってもよいという文献 [9] も存在し、非ユニタリな量子計算によってさらに高速な計算ができるかどうか、格子法などが可能かどうかなどにについて考えてみることにした。

まず、本当にユニタリ以外の計算ができることがあるのだろうか。量子の物理法則は基本的に全てユニタリ変換だと言われている。しかし、ブラックホールの蒸発はユニタリ変換だけでは説明できない [10] らしく、まだ可能性を完全に否定できないと考えられる。非ユニタリにも様々なものがあるため、これから私の論述する手法が可能であるかはわからないが、反例としてあげておいた。

どのような非ユニタリ変換も可能ならば、量子探索は $O(1)$ で求まる。私は単純にこのような非現実的な議論ではなく、ありえそうな単純化された非ユニタリ変換を考えることにした。一つ目は、以下のような非ユニタリ変換である。

$$|0\rangle|0\rangle \to |0\rangle|0\rangle \tag{175}$$
$$|1\rangle|0\rangle \to |1\rangle|0\rangle \tag{176}$$
$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle \to \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|1\rangle \tag{177}$$

もし、これが可能だと、定理 2 の内容が計算可能になる。すなわち、

$$\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)|0\rangle \to \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)|0\rangle \tag{178}$$
$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)|0\rangle \to \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)|1\rangle \tag{179}$$
$$\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)|0\rangle \to \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)|1\rangle \tag{180}$$
$$\frac{1}{\sqrt{2}}(|01\rangle + |11\rangle)|0\rangle \to \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle)|1\rangle \tag{181}$$

を計算できる。計算手順は、次の通りである。

1. アドレスビットと OR を計算したいビットの排他的論理和ととる。
2. 計算したビットについて、前述の非ユニタリな操作を実行する。
3. その計算したビットの否定と OR を計算したいビットの和をとる。

これができれば、格子法などの計算も効率化できる可能性が見えてくることに注意したい。

もう一つ考えたモデルは、量子の独立的な複製である。量子状態は制御 NOT により複製できるのだが、それは、

$$(\alpha|0\rangle + \beta|1\rangle)|0\rangle \to \alpha|0\rangle|0\rangle + \beta|1\rangle|1\rangle \tag{182}$$

である。私が要求してるのは、

$$(\alpha|0\rangle + \beta|1\rangle)|0\rangle \to (\alpha|0\rangle + \beta|1\rangle)(\alpha|0\rangle + \beta|1\rangle) \tag{183}$$

である。これは量子複製不可能定理からもユニタリ変換を仮定するとできないといわれてるものである。これができると、前述の非ユニタリな操作が近似的に可能になることを手順によって示す。

1. 操作されるビットを何回も (100 回くらい) 複製する。
2. 複製されたビットの全ての積と、複製されたビットを全て否定したものの全ての積を、計算して出力する。
3. 出力された結果の和をとって、否定として出力する。

これを行うと、

$$(\alpha|0\rangle + \beta|1\rangle)|0\rangle \to (\alpha|0\rangle + \beta|1\rangle)(\sqrt{\alpha^{2l} + \beta^{2l}}|0\rangle + \sqrt{1 - \alpha^{2l} - \beta^{2l}}|1\rangle \tag{184}$$

が計算でき、 l が十分大きければ、現実的な範囲内で繰り返し使っても問題はない。すなわち、量子の独立的な複製が可能ならば、量子探索は、$O((logN)^2)$ ほどの時間ででき、私の考えるアルゴリズムがそのまま可能になる。

5　結論

私は初め、量子コンピュータを 2^n 通りが同時に計算できるコンピュータだと考えていたのだが、簡単にそういうわけではない。この報告書では述べなかったが、Shor のアルゴリズムは素因数分解の問題を対数時間ほどに落とし込むことができ、確かにスピードアップが望める。しかし、最適解探索問題などで重要な Grover のアルゴリズムに対しては対数時間までとはいかず、平方根時間ほどである。また、Grover のアルゴリズムなどにおいて用いられる関数については、古典コンピュータと同性能の計算を行うことができるが、量子コンピュータの並列性を利用して格子法や粒子法などで効率的に計算する手法は不可能である。よって、私の述べてきたアルゴリズムをそのまま計算することは不可能である。しかし、量子計算は現代の手法より高速であるために、３０年後ぐらいにはよく利用される手法になると考えている。最後に非ユニタリな量子力学の可能性を述べたが、仮に物理法則にあらがあって、量子が独立的に複製可能であるとすると、発明のスピードが著しく増加し、現代の人間社会は崩壊するであろう。私は人間にはそれが不可能であっても、宇宙人がすでにそのようなコンピュータを使っているのではないかと考えている。

参考文献

[1] Michael A.Nielsen, Isaac L.Chuang, Quantum Computation and Quantum Information

[2] http://www.gizmodo.jp/2011/11/ibm452019.html

[3] Marek Sasura, Vladmir Buzek, 2007, Cold trapped ions as quantum information processors, Journal of Modern Optics

[4] Peter Dayan, L.F.Abbott, THEORETICAL NEUROSCIENCE

[5] https://www.flickr.com/photos/jurvetson/8054771535/

[6] 古澤明, 量子光学と量子情報科学, 数理工学社

[7] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson and D. J. Doll (1994). “Quantum annealing: A new method for minimizing multidimensional functions”. Chem. Phys. Lett. 219: 343. doi:10.1016/0009-2614(94)00117-0.

[8] Roman Martok, Giuseppe E. Santoro, and Erio Tosatti, Quantum annealing of the traveling-salesman problem

[9] 大矢雅則, 牧二郎, 量子コンピュータの数理 (パリティ物理学コース—クローズアップ)

[10] Steffen Gielen, 2009, Does black-hole evaporation imply that physics is non-unitary, and if so, what must the laws of physics look like? An Essay.