量子コンピュータ

宮田　俊敬
02RP034

目 次

はじめに

量子コンピュータと一言で言ってもその内容は大きすぎるだろう。だからといって、内容を一つに絞りすぎては深い理解につながらない。そういった理由もあるので、ここでは量子コンピュータの大まかな仕組み、それを実現するためにどういった装置を作るかを調べ、またそれを理想的にできるものと仮定したときに、量子コンピュータの威力を発揮できるアルゴリズムをいくつか具体的に調べたいと思う。

さらに付録として量子コンピュータに関係する話として、量子暗号などを付け加えたい。

1　量子計算機

1.1　序章

量子コンピュタとは、量子力学の原理を用いた、まったく新しい仕組みによるコンピュータである。その概念は１９８２年、ファインマンが量子系の物理学を古典コンピュータでシミュレーションすることに関する根本的な困難について指摘し、その困難を乗り越えるためには量子コンピュータが必要であろう。と、そう述べたあたりからである。しかし、ここで扱っていく量子コンピュータは、その主な性質は平行計算であり、そういった意味ではドイチュ（もともと平行宇宙論を研究しており、平行計算を最初に指摘した人物）が発案者となるだろう。

1.2　古典計算機のモデル（テューリングマシン）

そもそも（古典）計算機とはどういった仕組みで動いているのであろうか?実際は違うのだがそのモデルとしてテューリングマシンがしばしば使われる。その仕組みは０と１を書き込んだテープ（作業テープ）に新たに動作命令を与えるテープ（入力テープ）を用意し、命令作業を終えた後で作業テープを計算結果として見るものである。もちろん入力テープはあらかじめプログラムされている。こう見ると、「計算とは状態間の遷移」と言えそうである。というのも量子力学などで何度もやってきた、波動関数に演算子をかけ、演算後の波動関数を読み取る過程と似通っている。実際量子計算とは、この演算子として常にユニタリーなものを使いうまくその性質を利用したものと言える。簡単な例を挙げよう、入力テープとして

$$\Sigma =\vdash 1001 \dashv \tag{1}$$

と、ある固定したものを選ぶ。作業テープには

$$\Gamma =\vdash BBB... \tag{2}$$

としたものを用意する。ここで$\vdash$ は左端、$\dashv$ は右端、B はブランクを示すものである。($\vdash$、 $\dashv$、B は０，１で表されてないではないかと思うかもしれないが、例えば$\vdash = 010$、 $\dashv = 010$、 $B = 000$ とでもしておけばよい。つまり全ての入出力は０と１からなることにも注意)

ここで計算過程において少し付け加えおいておく必要がある。この作業テープ、入力テープは十分離れており重なることはないが、ただし２つの間にはお互いの状態（０か１）を見て行動を起こす”ヘッド”と言うものがある。先程入力テープが命令を与えると言ったが、本当はこのヘッドである。このヘッドは、状況に応じて作業テープの０と１をひっくり返したり、テープ上を移動したりする。

話をまた戻す。次に命令として

$$Q = \{q_0, q_1\} \tag{3}$$

$$\begin{gathered} q_0 = \\ \vdash\vdash\rightarrow\vdash RRq_0 \\ 0B \rightarrow 0RRq_0 \\ 1B \rightarrow 1RRq_0 \\ \dashv B \rightarrow 0SRq_1 \end{gathered} \tag{4}$$

$$\begin{gathered} q_1 = \\ \dashv B \rightarrow A \end{gathered} \tag{5}$$

を準備する。ここで出てきた R は、ヘッドを右に動かすことを意味し（L は左に）S、A は、前者は入力テープのヘッドの動きを止め、後者は計算結果を受理して停止することを意味する。

実はこの操作は、一度ブランクの作業テープに入力テープをコピーし、その後２倍する。という計算をしている（入力を 1001 とすれば、式　$1001 \rightarrow 10010$ が出る）。似たようなものを

$$Q = \{q_0, q_1, q_2\} \tag{6}$$

$$\begin{gathered} q_0 = \\ \vdash\vdash\rightarrow\vdash RRq_0 \\ 0B \rightarrow 0RRq_0 \\ 1B \rightarrow 1RRq_0 \\ \dashv B \rightarrow BLLq_1 \end{gathered} \tag{7}$$

$$\begin{gathered} q_1 = \\ 00 \rightarrow 1RRq_2 \\ 11 \rightarrow 1LLq_1 \end{gathered} \tag{8}$$

$$\begin{gathered} q_2 = \\ 11 \rightarrow 1LLq_1 \\ \dashv B \rightarrow A \end{gathered} \tag{9}$$

として作れば、今度は 1 を足すテューリングマシンが完成する。

もちろん、繰り返しになるが普通の計算機が文字通りテューリングマシンの原理で動いてるわけではない。実際にはハードウェアに依存した動作をしているが、テューリングマシンの動作に置き換えて考えることは可能とされている。

テューリングマシンは、詳細は触れないが、計算可能性あるいは計算量の定義に便利であり、そこでの定義が普通の計算機でも成り立つことが示されているので、これが計算機の仕組みとしてテューリングマシンが使われる理由となる。[1]

1.3 量子テューリングマシン

計算とはプログラムされた状態間の遷移であるという点では、量子計算機にも違いはない。量子計算機と古典計算機と違う点は、量子計算機においては状態として 0 と 1 だけでなく、それらの重ね合わせも許す点である。すなわち、$|\alpha|^2 + |\beta|^2 = 1$　を満たす複素数として、

$$\Psi = \alpha 0 + \beta 1 \tag{10}$$

なる状態もテープに書き込める（今後状態であることを強調するため、0 と 1 を $|0\rangle, |1\rangle$ と書くことにする）これが物理的に可能な状態であることは、例えば磁気モーメントを持つスピン $\frac{1}{2}$ の粒子を考えればよい。スピン $\frac{1}{2}$ の粒子のスピンは、量子化の方向を Z 方向に指定すれば、＋か－のどちらかで、それに一定時間一様な磁場をかければよい。スピンと Z 軸のなす角を θ とすれば　$\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$ を得ることができる。

またここで、今まで 0 と 1 と呼んでいたものを”ビット”と呼ぶことにし、上で言った重ね合わせも許すビットのことをキュービットと呼ぶことにする。このキュービットを仮に N こ持ってくれば、2^N 個もの状態を重ね合わせを用意することができる。例えば N=3 として見よう。すると 0 から 7 までの数を同じ重みで表現できる。キュービットを先程述べたスピンとして、3 つのキュービット全てを上向きにセットしておく（状態は $|0\rangle|0\rangle|0\rangle$）。わかりいいように図で書いておこう

$$\uparrow \quad \uparrow \quad \uparrow \tag{11}$$

これに磁場をかけると（破線で磁場を表す）

$$\cdots\uparrow\cdots\uparrow\cdots\uparrow\cdots \rightharpoonup$$
$$=\updownarrow\updownarrow\updownarrow$$

ここで

$$\updownarrow\updownarrow\updownarrow = \frac{1}{2\sqrt{2}} \begin{cases} \uparrow\uparrow\uparrow \\ \uparrow\uparrow\downarrow \\ \uparrow\downarrow\uparrow \\ \uparrow\downarrow\downarrow \\ \downarrow\uparrow\uparrow \\ \downarrow\uparrow\downarrow \\ \downarrow\downarrow\uparrow \\ \downarrow\downarrow\downarrow \end{cases} \tag{12}$$

なんといっても量子コンピュータの強みはこの重ね合わせである。しかしこの現象をテューリングマシンで表そうとすると、テューリングマシンのテープが次から次えと増えてしまうことになり扱いにくい。そこでテューリングマシンに代わる中間のステップ、論理回路が必要になってくる。

2　量子論理ゲート

2.1　古典計算機における論理ゲート

古典計算機における代表的な論理ゲートとして、NOT OR AND XOR などを紹介しておく。

NOT：論理否定

$$|0\rangle \to |1\rangle \tag{13}$$
$$|1\rangle \to |0\rangle$$

OR：入力に $|1\rangle$ が含まれてさえいれば $|1\rangle$

$$|0\rangle|0\rangle \to |0\rangle \tag{14}$$
$$|0\rangle|1\rangle \to |1\rangle$$
$$|1\rangle|0\rangle \to |1\rangle$$
$$|1\rangle|1\rangle \to |1\rangle$$

AND：掛け算

$$|0\rangle|0\rangle \to |0\rangle \tag{15}$$
$$|0\rangle|1\rangle \to |0\rangle$$
$$|1\rangle|0\rangle \to |0\rangle$$
$$|1\rangle|1\rangle \to |1\rangle$$

XOR：2 を法とする足し算

$$|0\rangle|0\rangle \to |0\rangle \tag{16}$$
$$|0\rangle|1\rangle \to |1\rangle$$
$$|1\rangle|0\rangle \to |1\rangle$$
$$|1\rangle|1\rangle \to |0\rangle$$

この論理ゲートがあると、テューリングマシンに代わるものができる。テューリングマシンは簡単に言うと、作業ビット、入力ビットをヘッドが読み込んで、作業ビットの書き換えを行っているにすぎない。そんな場合の数は

$$|0\rangle|0\rangle \to |0\rangle or |1\rangle \tag{17}$$
$$|0\rangle|1\rangle \to |0\rangle or |1\rangle$$
$$|1\rangle|0\rangle \to |0\rangle or |1\rangle$$
$$|1\rangle|1\rangle \to |0\rangle or |1\rangle$$

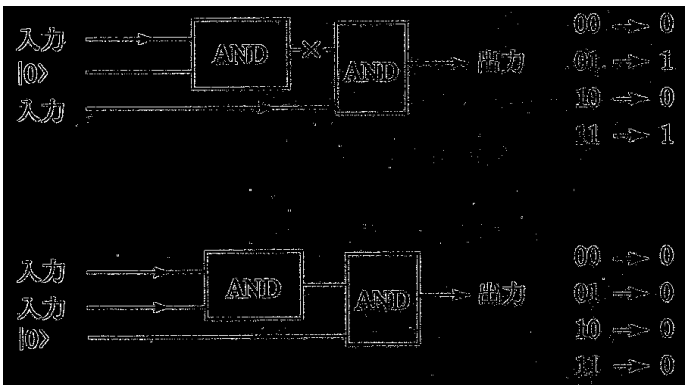


図 1: 例

の $2^4 = 16$ 通りある。つまりそんな 16 通りすべてを論理ゲートで組み合わせることが出来ればいいわけである。これを実際作ることは出来て、例えば AND と NOT を使って実現できる（万能テューリングマシン）。証明は明確で、しかし力ずくのものではあるが、16 通り全ての場合を AND と NOT を用いて表現すればよい。

2.2 量子計算機における論理ゲート

しかし量子計算機の強みは何度も強調したとおり重ね合わせである。重ねあわせを実現するには、NOT、OR、AND の様に 2 つの入力から 1 つの出力を出すような非可逆な変換を量子計算機のゲートとして使うのは好ましくない。前に述べたとおり量子計算機において、古典計算機における 0 と 1 の書き換えは複素 2 次元のユニタリー変換にあたる。（量子計算がユニタリである以上 $UU^\dagger = U^\dagger U = 1$）そこで新しいゲートを要請する。

制御 NOT は、古典計算における XOR の働きもできる 2 キュービットゲートで、量子計算において重要な役割を果たす。2 つの入力ビットのうち一方を制御ビット、他方を標的ビットと呼ぶ。制御ビットを区別するために普通黒丸を打ってある。図の左側から入力され、右側に出力される。制御ビットが $|0\rangle$ のときは遷移を起こさないが、制御ビットが $|1\rangle$ のときは NOT ゲートとして働く。言い換えると、制御ビット $|a\rangle$ と標的ビット $|b\rangle$ の入力があれば、標的ビットに $|(a+b) \bmod 2\rangle$ の出力があるので、確かに XOR の働きをしている。2 つの入力に対し、明確な 2 つの出力があるのだから、逆があることは明らかだろう。

ここでも詳細についてはふれないが、制御ノットと 1 ビットのユニタリー変換の組み合わせでせべてのユニタリー変換が実行できることがすでに示されている。

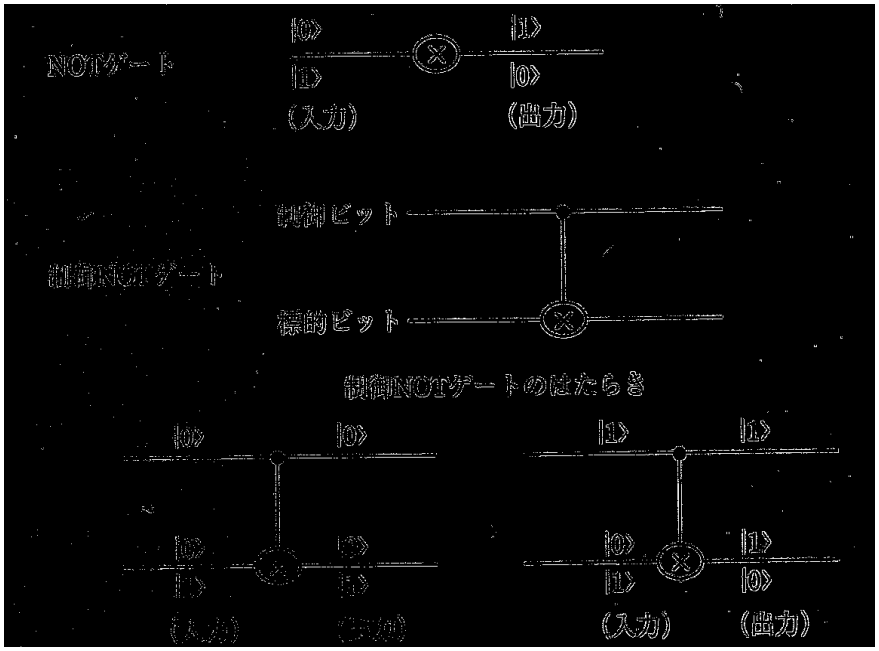


図 2: 制御 NOT ゲート

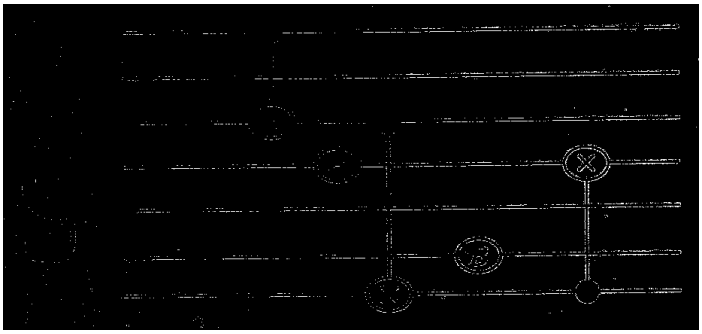


図 3: 制御 NOT ゲート

上の図に書いた量子計算のダイアグラムの見方は、横線はキュービットを表しその上に $|0\rangle$ と $|1\rangle$ の重ね合わせ状態が置かれる。ダイアグラムを楽譜のように左から右に見ていく（当然本によっては右から読ませるものもある）。縦線はゲートと呼ばれるキュービット間の相互作用を表し、ゲートを通過するたびに、状態は指定されはユニタリー変換を受ける。

2.3 エンタングルド状態

今までは入力として 0 か 1、つまり制御 NOT の古典的な機能であるが、量子的にはもっといろいろな働きが出来る。例えば制御ビットに重ね合わせ状態 $|0\rangle + |1\rangle$ を選ぶと、標的ビットが 1 のとき、入力状態は直積で

$$(|0\rangle + |1\rangle)|1\rangle = |0\rangle|1\rangle + |1\rangle|1\rangle \tag{18}$$

である。これに制御 NOT を働かせると、出力としてエンタングルド状態（絡み合った状態）

$$|0\rangle|1\rangle + |1\rangle|0\rangle \tag{19}$$

を作ることが出来る。式を見ればすぐ分かるが、これは直積の形でかけない。制御ビットが $|0\rangle$ なら標的ビットは必ず $|1\rangle$ の状態であり、逆に制御ビットが $|1\rangle$ なら標的ビットは必ず $|0\rangle$ の状態である（これが絡み合っていることを意味する）。用は観測によって生じる相関の事である。

2.4 量子複製不可能定理

後に（量子暗号等で）出てくると思うのでここで重ね合わせはコピーできないことについて触れよう。

証明は背理法による。

任意の状態がコピーできると仮定して、また $|a\rangle, |b\rangle$ を独立なベクトルとする。

$$\begin{aligned} C|a\rangle|0\rangle &= |a\rangle|a\rangle \\ C|b\rangle|0\rangle &= |b\rangle|b\rangle \end{aligned} \tag{20}$$

従って、

$$C\left(\alpha|a\rangle + \beta|0\rangle\right)|0\rangle = \alpha|a\rangle|a\rangle + \beta|b\rangle|b\rangle \tag{21}$$

しかしこれは $\alpha|a\rangle + \beta|0\rangle$ の複製

$$\left(\alpha|a\rangle + \beta|0\rangle\right)\left(\alpha|a\rangle + \beta|0\rangle\right) \tag{22}$$

とは $\alpha = 1(0), \beta 0(1)$ の場合以外は異なる。

証明終了。

3　量子回路

量子論理ゲートには 2 種類あり、一つは 1 ビットのユニタリー変換でもう一つは制御 NOT である、とりあえず今後、この 2 つの回路は作れるものとして話を進める。また、それらの組み合わせで全ての量子計算が可能になる

3.1　ユニタリー変換の構成

ドイチたちは、任意のユニタリー変換が、2 キュービットの制御 NOT と 1 キュービットのユニタリー変換の組み合わせで作れることを示している。

まずそのための準備として制御 U なるものを作る（制御 NOT はその特別な形である）。

任意の SU(2) 行列 U に対して A,B,C を ABC=I、AXBXC ＝ U となるように選ぶことが出来て、例えば

$$U = R_z(\alpha)R_y(\beta)R_z(\gamma) \tag{23}$$

と回転行列 R を用いて分解し、3 つのユニタリー行列 A,B,C を

$$A = R_z(\frac{\alpha-\gamma}{2}), \quad B = R_z(-\frac{\alpha+\gamma}{2})R_y(-\frac{\beta}{2}), \quad C = R_y(\frac{\beta}{2})R_z(\gamma) \tag{24}$$

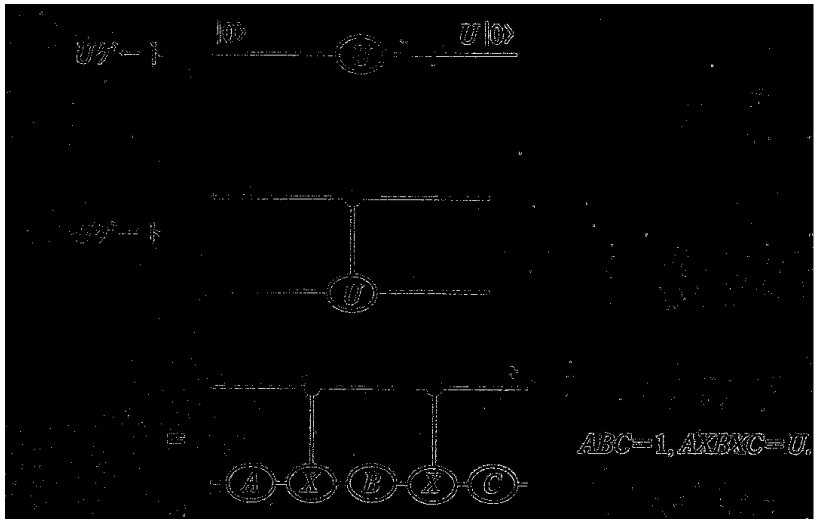


図 4: U ゲート、制御 U ゲート

次にこの制御 U 等を使って任意のユニタリー変換が出来ることを示す。証明の仕方は以下の通りで、任意の状態（$\sum_{a,b=0,1} C_{ab}|a\rangle|b\rangle$）から 4 つの基底

$|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle$　各々へ変換できることを示す。そうすれば、ユニタリー変換は逆があるのだから、各基底から任意の状態へ変換できることになる。3 ビット以上に拡張することはその応用である。

ためしにひとつ、

$$\sum_{a,b=0}^{1} C_{ab}|a\rangle|b\rangle \to |1\rangle|1\rangle \tag{25}$$

を考えてみる。しかしその前に各ゲートの行列表示を記しておく必要があると思われる。まず基底 $|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle$　は行列表記で

$$|0\rangle|0\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, |0\rangle|1\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, |1\rangle|0\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, |1\rangle|1\rangle = \begin{pmatrix}1\\0\\0\\1\end{pmatrix} \tag{26}$$

と定義しておく。第 1 ビットに制御ビット、第 2 ビットを標的ビットにする制御 U は

$$U = \begin{pmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & u_{11} & u_{12}\\0 & 0 & u_{21} & u_{22}\end{pmatrix} \tag{27}$$

また逆の、第 1 ビットを標的ビット、第 2 ビットを制御ビットとした制御 W は

$$W = \begin{pmatrix}1 & 0 & 0 & 0\\0 & u_{11} & 0 & u_{12}\\0 & 0 & 1 & 0\\0 & u_{21} & 0 & u_{22}\end{pmatrix} \tag{28}$$

更に第 1 ビットを NOT、第 2 ビットを恒等変換（ゲートなし）としたものは

$$\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix} \bigotimes \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} = \begin{pmatrix}0 & 0 & 1 & 0\\0 & 0 & 0 & 1\\1 & 0 & 0 & 0\\0 & 1 & 0 & 0\end{pmatrix} \tag{29}$$

これを使って下のような回路を作れば直ちに $|1\rangle|1\rangle$ を得られる。まず最初のゲートで

$$\begin{pmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & u_{11} & u_{12}\\0 & 0 & u_{21} & u_{22}\end{pmatrix} \begin{pmatrix}c_{00}\\c_{01}\\c_{10}\\c_{11}\end{pmatrix} \to \begin{pmatrix}c_{00}\\c_{01}\\0\\\sqrt{c_{10}^2 + c_{11}^2}\end{pmatrix} \text{(ユニタリー性より可能)} \tag{30}$$

同様の計算を NOT、制御 V(式 (27) の変数を変えたもの)、NOT と立て続けに行うと

$$\begin{pmatrix} 0 \\ \sqrt{c_{00}^2 + c_{011}^2} \\ 0 \\ \sqrt{c_{10}^2 + c_{11}^2} \end{pmatrix} \tag{31}$$

を得て、最後に制御 W で $|1\rangle|1\rangle$ を取り出せばよい。

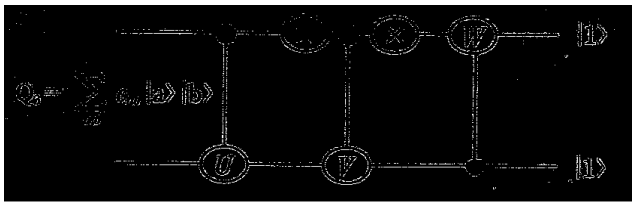


図 5: $|1\rangle|1\rangle$ を取り出す回路

このようにして全ての基底を任意の状態から導くことが出来る。

次に各基底にたいする位相を与える回路を作ろう。まず簡単のため $|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle$ を $|\phi_0\rangle, |\phi_1\rangle, |\phi_2\rangle, |\phi_3\rangle$ とする。すると数式的な演算子は $S = \sum_{n=0}^{3} e^{i\sigma_n} |\phi_n\rangle\langle\phi_n|$ として表せる。このゲートを作りたい。そこで $S = \prod_{n=0}^{3} G_n^{-1} X_n G_n$ を考える。G_{n} は前述の吐き出して基底を取り出す演算子で、これが制御 NOT、1 ビットのユニタリー変換で作れることはすでに示した。X_n は

$$\begin{aligned} X_0 &= e^{i\sigma_0}|0\rangle\langle 0| + |1\rangle\langle 1| + |2\rangle\langle 2| + |3\rangle\langle 3| \\ X_1 &= |0\rangle\langle 0| + e^{i\sigma_1}|1\rangle\langle 1| + |2\rangle\langle 2| + |3\rangle\langle 3| \\ X_2 &= |0\rangle\langle 0| + |1\rangle\langle 1| + e^{i\sigma_2}|2\rangle\langle 2| + |3\rangle\langle 3| \\ X_3 &= |0\rangle\langle 0| + |1\rangle\langle 1| + |2\rangle\langle 2| + e^{i\sigma_3}|3\rangle\langle 3| \end{aligned}$$

とかけるものだが、この X_n はゲートで表すと事が出来る。

図 X_0, X_1, X_2, X_3

まず、$G_0^{-1} X_0 G_0$ の働きを見てみる。

$$G_0^{-1} X_0 G_0 |\phi_0\rangle = e^{i\sigma_0}|\phi_0\rangle \tag{32}$$

は定義から明らかで、$|\phi_n\rangle, (n \neq 0)$ に対しては

$$G_0^{-1}X_0G_0|\phi_n\rangle = G_0^{-1}(e^{i\sigma_0}|0\rangle\langle 0| + |1\rangle\langle 1| + |2\rangle\langle 2| + |3\rangle\langle 3|)G_0|n\rangle$$

だが、$\langle\phi_0|G_0|\phi_n\rangle = 0$ より

$$\text{与式} = G_0^{-1}(|1\rangle\langle 1| + |2\rangle\langle 2| + |3\rangle\langle 3|)G_0|n\rangle$$

この丸カッコ内に $|0\rangle\langle 0|$ を入れても 0 を入れるだけだから代わらない, 従って、

$$= G_0^{-1}(|0\rangle\langle 0| + |1\rangle\langle 1| + |2\rangle\langle 2| + |3\rangle\langle 3|)G_0|n\rangle$$
$$= G_0^{-1}G_0|\phi_n\rangle = |\phi_n\rangle$$

このことから一般に

$$G_m^{-1}X_mG_m|\phi_m\rangle = e^{i\sigma_m}|\phi_m\rangle,$$
$$G_n^{-1}X_nG_n|\phi_m\rangle = |\phi_m\rangle, (n \neq m)$$

が成り立つことがわかる。このことは、S を $|\phi_m\rangle$ 二演算するときに、積、$\prod_{n=0}^{3} G_n^{-1}X_nG_n$ のうち $n = m$ のところに位相 $e^{i\sigma_m}$ を与え、他は恒等変換として働くことを示している。従って、

$$S|\phi_m\rangle = e^{i\sigma_m}|\phi_m\rangle, m = 0, 1, 2, 3, \tag{33}$$

が示された。これは $S = \sum_{n=0}^{3} e^{i\sigma_n}|\phi_n\rangle\langle\phi_n|$ に他ならない。

3.2 量子計算の簡単な例

ここで簡単な例 $(a + b) \mod 2$ を計算してみるよう。a,b という入力に対し、出力が $(a + b) \mod 2$ となればいいわけだが下のような回路

図 6: $(a + b) \mod 2$

を組めば、$|a\rangle|b\rangle|0\rangle \rightarrow |a\rangle|b\rangle|(a + b) \mod 2\rangle$ が出力として得ることが読み取れる。しかしこのままでは古典計算となんら変わりはない。a,b に重ね合

わせ状態を与えることで初めて量子計算になる。この状態に持っていくには a,b をはじめ 0 でセットし各々のキュービットにウォルシュ・アダマール変換と呼ばれる

$$|0\rangle \to \frac{|0\rangle + |1\rangle}{\sqrt{2}} \qquad (34)$$
$$|1\rangle \to \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

ゲートを用意すればいい。その後は上で言った回路に従えばいいわけで、結局まとめると

$$|0\rangle|0\rangle|0\rangle \stackrel{H}{\to} (\frac{1}{\sqrt{2}})^2 \sum_{a,b=0}^{1} |a\rangle|b\rangle|0\rangle \stackrel{U}{\to} \sum_{a,b=0}^{1} |a\rangle|b\rangle|(a+b) \mod 2\rangle \qquad (35)$$

という出力を得る。これを見ると、4 つの計算を平行して同時にしていることが分かる。もちろんこの計算を測定しても $|a\rangle|b\rangle|(a+b) \mod 2\rangle$ の出力のうちランダムな計算結果を得るだけであまり意味がないものと思われる。例えば $(0+1) \mod 2$ をしたいのであれば、回路に（何も量子回路でなくて良い）に $a=1, b=0$ を入力したほうが良いだろう。こういった理由もあり、量子コンピューターは当初、まともな計算が出来るとは思われてなかった。

3.3 論理回路

古典計算は NOT,AND,OR,XOR などの基本的ゲートをうまく組み合わせることによって様々な計算を行うものであった。それに対し量子計算は、ユニタリー変換をすることで状態間を遷移させるものであった。しかし当然、量子回路でも NOT,AND,OR,XOR は作ることは出来る。一応示しておこう。

基本的な道具（といっても極わずかなアルゴリズムを作るためのだが）は揃ったとおもわれる。ここからついに、量子コンピューターの威力を発揮すると言われる有名なアルゴリズム、因数分解、グローバーのアルゴリズム、ドイチ・ジョサのアルゴリズム等についてふれて行きたい。

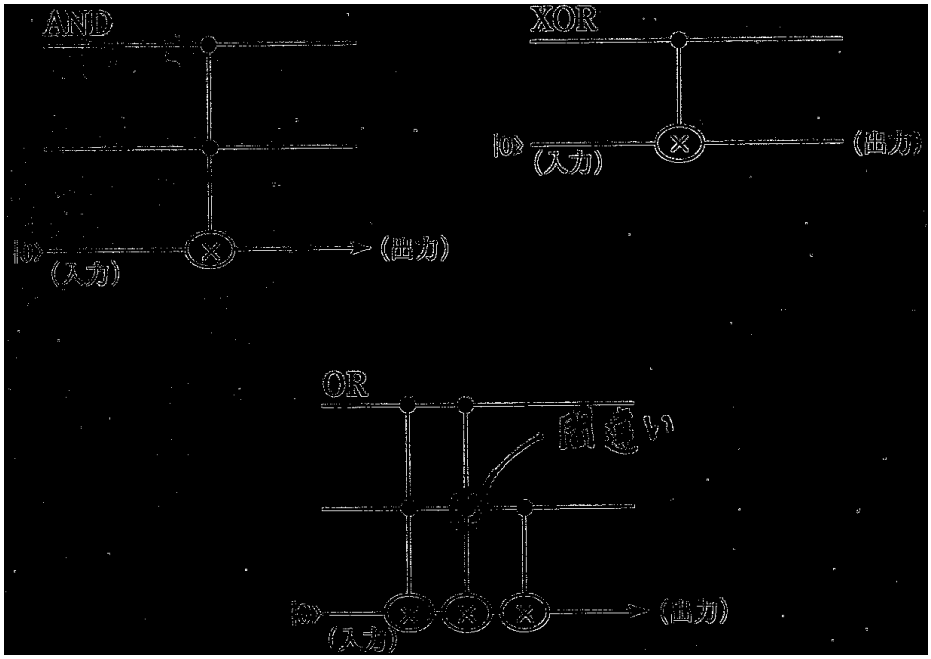


図 7: 論理ゲート

4 離散積分変換

4.1 離散積分変換

一般に 2 つの実数値変数 x,y の関数 K に対して

$$\tilde{f}(y) = \int_a^b K(y,x)f(x)dx \tag{36}$$

によって、関数 f を $\tilde{f}$ に変換することを K を核とした積分変換とよぶ。例えば

$$K(y,x) = e^{-ixy}, a = -\infty, b = \infty \tag{37}$$

ととった場合はいわゆるフーリエ変換と呼ばれ、最もよく使われる積分変換の一つである。この変換において、変数のとる値を離散化し積分を和に置き換えたもの離散型積分変換という。

離散積分変換

$$\tilde{f}(y) = \sum_{x=0}^{N-1} K(y,x)f(x) \tag{38}$$

ここでひとつ後々のため命題を証明しておく。[2]

命題

$N = 2^n$ とし、K を N × N のユニタリ行列とする。このとき量子コンピュータ U が

$$U|x\rangle = \sum_{y=0}^{N-1} K(y,x)|y\rangle \tag{39}$$

を満たせば、U は K を核とする離散積分変換

$$\tilde{f}(y) = \sum_{x=0}^{N-1} K(y,x)f(x)$$

を計算する。すなわち

$$U\,(\sum_{x=0}^{N-1} f(x)|x\rangle) = \sum_{y=0}^{N-1} \tilde{f}(y)|y\rangle \tag{40}$$

が成り立つ

証明

$$U\,(\sum_{x=0}^{N-1} f(x)|x\rangle) = \sum_{x=0}^{N-1} f(x)U|x\rangle \tag{41}$$

$$= \sum_{x=0}^{N-1} f(x)\sum_{y=0}^{N-1} K(y,x)|y\rangle = \sum_{y=0}^{N-1}(\sum_{x=0}^{N-1} K(y,x)f(x))|y\rangle$$

$$= \sum_{y=0}^{N-1} \tilde{f}(y)|y\rangle \tag{42}$$

となり、式（39）と一致する。

4.2 ウォルシュ・アダマール変換

今までも何度か話してきた変換でなんとなく分かるものと思うが少し厳密に、離散積分変換の一つとして再考する。前にも述べたように 1 キュービットに対してウォルシュ・アダマール変換は

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

であったが、これを $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$ と、行列表示したときウォルシュ・アダマール変換は

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \tag{43}$$

となる。では n ビットに対してはどうなるか？

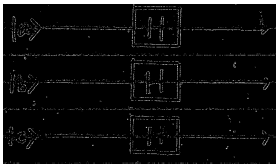


図 8: ウォルシュ・アダマール変換

まず、分かりいいように回路を上に書いておく

入力を図の通り（正直に図の通りに書くと $|a\rangle \bigotimes |b\rangle \bigotimes |c\rangle$ だが）$|i_0\rangle \bigotimes |i_1\rangle \bigotimes ... \bigotimes |i_{n-1}\rangle \equiv |i)$ として $(i_k = 0, 1)$ ウォルシュ・アダマール変換 H は

$$H|i) = \Pi_{k=0}^{n-1} \bigotimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} |i_k\rangle \tag{44}$$

(38) 式は $\sum_{j_k=0}^{1} (-1)^{i_k \cdot j_k} |j_k\rangle$ が

$$i_k = 0 \text{ のとき } \rightarrow |0\rangle + |1\rangle$$
$$i_k = 1 \text{ のとき } \rightarrow |0\rangle - |1\rangle$$

であることから

$$\Pi_{k=0}^{n-1} \bigotimes \sum_{j_k=0}^{1} \frac{1}{\sqrt{2^n}} (-1)^{i_k \cdot j_k} |j_k\rangle \tag{45}$$

と書くことが出来る。更にこの式をいじくってやると

$$= \frac{1}{\sqrt{2^n}} (\Sigma_{j_0=0}^{1} (-1)^{i_0 \cdot j_0} |j_0\rangle) \bigotimes (\sum_{j_1=0}^{1} (-1)^{i_1 \cdot j_1} |j_1\rangle) \bigotimes (\sum_{j_2=0}^{1} (-1)^{i_2 \cdot j_2} |j_2\rangle) \bigotimes ...$$

$$= \frac{1}{\sqrt{2^n}} \sum_{j=0}^{N-1} (-1)^{i_0 \cdot j_0 + i_1 \cdot j_1 + ... + i_{n-1} \cdot j_{n-1}} |j)$$

$$(\because \sum_{j=0}^{1} \sum_{j=0}^{1} ... \sum_{j=0}^{1} = \sum_{j=0}^{2^n - 1})$$

$$= \frac{1}{\sqrt{2^n}} \sum_{j=0}^{N-1} (-1)^{i \cdot j} |j)$$

となることがわかる。まとめるとウォルシュ･アダマール変換とは核を $\frac{1}{\sqrt{2^n}} (-1)^{i \cdot j}$ とした離散積分変換

$$H|i) = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{N-1} (-1)^{i \cdot j} |j) \tag{46}$$

ただしここここで $i \cdot j = i_0 \cdot j_0 + i_1 \cdot j_1 + ... + i_{n-1} \cdot j_{n-1}, N = 2^n$ としてある。

4.3　フーリエ変換

同様にフーリエ変換についても記述しておく。フーリエ変換の場合、核を $\exp(\frac{2\pi i}{N}xy)$ としたときの

$$\tilde{f}(y) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \exp(\frac{2\pi i}{N}xy) f(x) \tag{47}$$

を行う変換である。例えば、$f(1)$ にこの変換を施すと

$$\tilde{f}(1) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \exp(\frac{2\pi i}{N}x) f(x) \tag{48}$$

となり, この考え方で $N = 2^3 = 8$, 入力：$|1\rangle = |0\rangle|0\rangle|1\rangle$ としたとき出力は

$$\begin{aligned} |1\rangle \to U|1\rangle &= \frac{1}{\sqrt{8}} \sum_{x=0}^{N-1} \exp(\frac{2\pi i}{8}y)|y\rangle \\ &= \frac{1}{\sqrt{8}}\{|0\rangle + e^{\frac{2\pi i}{8}}|1\rangle + e^{\frac{4\pi i}{8}}|2\rangle + ...\} \end{aligned} \tag{49}$$

となる。ちなみにこれは入力に重ね合わせの状態 $\{\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\}^3$ を下の回路のように組んだとき同じ出力を得られる。これはフーリエ変換の特殊な形となるが、回路が非常に簡単で分かりやすい。

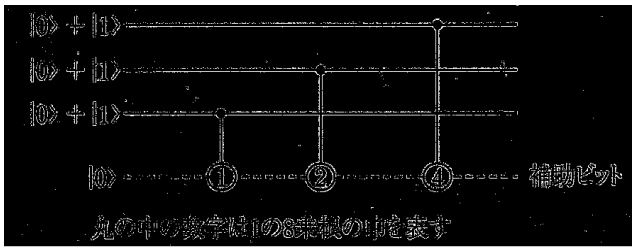


図 9: 8 項の離散的フーリエ変換

また、このフーリエ変換は因数分解のアルゴリズムとして多用されており、次のグラフはその因数分解で使う

$$\sum_{a,c=0}^{N-1} \exp\left(\frac{2\pi i}{N}ac\right)|a\rangle|c\rangle \tag{50}$$

を出力する絡んだフーリエ展開である。これが式（50）を満たすことはダイアグラムをたどれば直ちに確認できる。

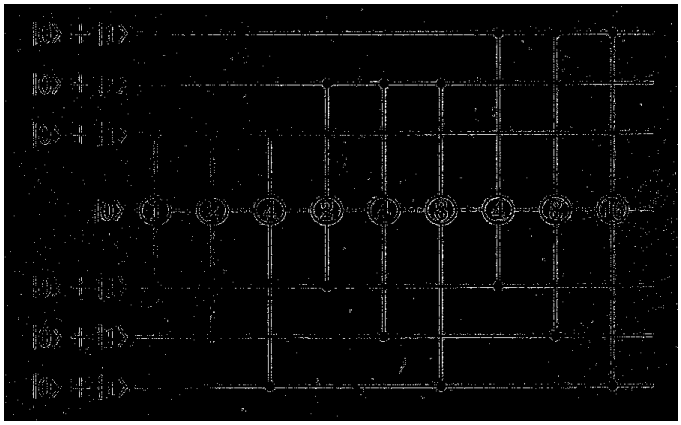


図 10: 絡んだフーリエ変換

4.4 選択的回転変換

核 K を

$$K(x,y) = e^{i\theta_x}\delta_x y \tag{51}$$

ととったときの離散積分変換

$$\tilde{f}(y) = \sum_{x=0}^{N-1} e^{i\theta_x}\delta_{xy} f(x) = e^{i\theta_y} f(y) \tag{52}$$

を選択的回転変換として定義される。ここに、θ_y は x によって定まる実数である。

この変換は具体例を見ると分かりやすい。$n = 1$ および 2 のときの K はそれぞれ

$$K_1 = \begin{pmatrix} e^{i\theta_0} & 0 \\ 0 & e^{i\theta_1} \end{pmatrix}, \qquad K_2 = \begin{pmatrix} e^{i\theta_0} & 0 & 0 & 0 \\ 0 & e^{i\theta_1} & 0 & 0 \\ 0 & 0 & e^{i\theta_2} & 0 \\ 0 & 0 & 0 & e^{i\theta_3} \end{pmatrix} \tag{53}$$

で与えられる。

選択的回転変換を計算する量子コンピュータは原理的に次のようにして構

成できる。例えば $n = 3$ の場合核は、基底を

$$|0,0,0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |0,0,1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |0,1,0\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} ... \tag{54}$$

ととり、

$$K_3 = \begin{pmatrix} e^{i\theta_0} & & & & & & & \\ & e^{i\theta_1} & & & & & & \\ & & e^{i\theta_2} & & & & & \\ & & & e^{i\theta_3} & & & & \\ & & & & e^{i\theta_4} & & & \\ & & & & & e^{i\theta_5} & & \\ & & & & & & e^{i\theta_6} & \\ & & & & & & & e^{i\theta_7} \end{pmatrix} \tag{55}$$

という形をした対角行列であるが、これは

$$K_3 = A_0 A_1 A_2 A_3 \tag{56}$$

の形に分解できる。ここに

$$A_0 = \begin{pmatrix} e^{i\theta_0} & & & & & & & \\ & e^{i\theta_1} & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{pmatrix} \quad A_1 = \begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & e^{i\theta_2} & & & & & \\ & & & e^{i\theta_3} & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{pmatrix} \tag{57}$$

$$A_2 = \begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & e^{i\theta_4} & & & \\ & & & & & e^{i\theta_5} & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{pmatrix} \quad A_3 = \begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & e^{i\theta_6} & \\ & & & & & & & e^{i\theta_7} \end{pmatrix} \tag{58}$$

である。一方このような回路は制御 U で下のような回路を組めば行える事が容易に分かる。ここで。

$$U_0 = \begin{pmatrix} e^{i\theta_0} & 0 \\ 0 & e^{i\theta_1} \end{pmatrix}, \quad U_1 = \begin{pmatrix} e^{i\theta_2} & 0 \\ 0 & e^{i\theta_3} \end{pmatrix}, \quad U_2 = \begin{pmatrix} e^{i\theta_4} & 0 \\ 0 & e^{i\theta_5} \end{pmatrix}, \quad U_3 = \begin{pmatrix} e^{i\theta_6} & 0 \\ 0 & e^{i\theta_7} \end{pmatrix} \tag{59}$$

である

実際、A_0 は式（58）から分かるとおり

$$A_0|0,0,0\rangle = e^{i\theta_0}, \quad A_0|0,0,1\rangle = e^{i\theta_1}|0,0,1\rangle \tag{60}$$

を満たし、そのほかの入力 $|a.b.c\rangle$ に対しては $A_0|a,b,c,\rangle = |a,b,c\rangle$ となる。結局この変換は任意の基底に位相を与える変換であり、うまく θ を選べば任意の基底にマイナスをつける事も出来る（後のグローバーのアルゴリズムで使う）

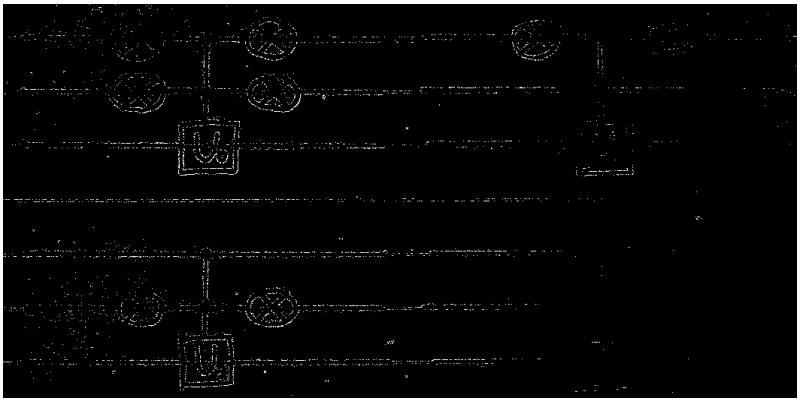


図 11: 回路 A_0, A_1, A_2, A_3

5 因数分解

因数分解は一見簡単そうに見えるが、桁数の増大とともに指数関数的に計算時間が増大することが知られており、例えば 1 万桁の整数を因数分解するには、現在最高の計算機を使っても、1000 億年以上の時間が必要だとされる。ところがショアは量子コンピューターを用いれば、桁数にせいぜい比例する程度の時間で計算できることを発見した。大雑把な試算では数時間で 1 万桁の因数分解が行えることになる。ただこの量子アルゴリズムはかなり込み入ってるので、まずはじめに因数分解のアウトラインを例を交えながら紹介することにする。

因数分解

簡単のため $N = pq, (p, q$ は素数) を因数分解することにする。

1: 集合 $\{1, 2, ..., N-1\}$ の中から一つの数 x をランダムに選ぶ。

2: $\gcd(x, N)$ を求める。

3: $\gcd(x, N) = 1$ ならば ” 4 ” へ行く。$\gcd(x, N) \neq 1$ ならば ” 8 ” へ行く。

4: x の mod$N$ に関する位数 r を求める。

5: r が偶数なら ” 6 ” へ行く。奇数ならば ” 8 ” へ。

6: $p' = \gcd(x^{\frac{r}{2}+1}, N)$ と $q' = \gcd(x^{\frac{r}{2}-1}, N)$ を求める。

7: p’q’ のいずれかが N ならば ” 8 ” へ行く。

8: 因数分解に失敗したので ” 1 ” に行きやり直す。

そうでなければ、それが p か q になる。

ここで出てきた位数とは、$x^r = 1(\mathrm{mod}N)$ を満たす r の事であり, これが因数になる事はつぎのようにしてわかる。今言ったように求めた r が $x^r = 1(\mathrm{mod}N)$

を満たすのだから、k をある自然数として

$$\begin{aligned} & x^r = 1 + kN \\ \Longrightarrow & x^r - 1 = kN \\ \Longrightarrow & (x^{\frac{r}{2}} - 1)(x^{\frac{r}{2}} + 1) = kN = kpq \end{aligned}$$

を得るが、当然 p,q=1 の解も考えられる。そのとき、仮に p=1 とすると、

$$(x^{\frac{r}{2}} + 1)(x^{\frac{r}{2}} - 1) = kq$$

となり（この場合 q=N となっている事に注意）

$$\begin{aligned} x^{\frac{r}{2}} + 1 &= k \quad (= q) \\ x^{\frac{r}{2}} - 1 &= q \quad (= k) \\ \therefore \gcd(x^{\frac{r}{2}} + 1, N) &= 1 \quad (= N) \\ \gcd(x^{\frac{r}{2}} - 1, N) &= N \quad (= 1) \end{aligned}$$

従って、当たり前の事だがこのケースでは因数分解できてない。

次に $p, q \neq 1$ のとき同様に

$$\begin{aligned} (x^{\frac{r}{2}} + 1)(x^{\frac{r}{2}} - 1) &= kpq \\ (x^{\frac{r}{2}} + 1)(x^{\frac{r}{2}} - 1) &= (hp)(lq) \quad (\text{ただし } k = hl) \\ \therefore x^{\frac{r}{2}} + 1 &= hp \quad (= lq) \\ x^{\frac{r}{2}} - 1 &= lq \quad (= hq) \end{aligned}$$

この式は

$$\begin{aligned} \gcd(x^{\frac{r}{2}+1}, N) &= \gcd(hp, pq) = p \quad (= q) \\ \gcd(x^{\frac{r}{2}+1}, N) &= \gcd(lq, pq) = q \quad (= p) \end{aligned}$$

である事を意味する。証明終了。

実はこのアルゴリズムは昔からしられたものであったのだが、ステップ”4”を求めるうまい方法が見つかっていなかったためにお蔵入りとなっていたものである。ショアはこの部分を量子コンピュータがあれば高速に解けるということを発見した。またこのアルゴリズムで多少気になる問題となるのは”2”、”6”の計算であろう。しかし”2”、”6”のステップはユークリッド交除法と呼ばれるアルゴリズムを使って求めることが出来て、今の計算機でも高速に解くことが出来る。

ユークリッド交除法

厳密な証明ではないが、一般に $\gcd(a, b), a \geq b$ を見つけるには $a - b$ と b を比較して大きいほうから小さいほうを引く。これを繰り返して 0 になる直

前の数を答えとすればよい。例えば gcd(20, 12) の場合は

$$20 - 12 = 8$$
$$12 - 8 = 4$$
$$8 - 4 = 4$$
$$4 - 4 = 0$$

となり、答え 4 を得る。

ではこのアルゴリズムを用いて実際に簡単な因数分解をしてみたい。

N=15 の例

1: a=7 が選ばれたとする
2: $\gcd(7, 15) = 1$
3: $\gcd(a, N) = 1$ なので ” 4 ” へ行く。
4: ここは本来、量子コンピュータで解くが r =4 が出たとする。
5: r =4 は偶数なので ” 6 ” へ行く。
6: $p' = \gcd(a^{\frac{r}{2}+1}, N) = 5$ $q' = \gcd(a^{\frac{r}{2}-1}, N) = 3$ を得る。
7: p’q’ のいずれも N ではないから、求める因数は 5 と 3 である。

5.1 ショアのアルゴリズム

このアルゴリズムは因数分解のアルゴリズム、ステップ 4 のためのアルゴリズムで、結局 $x^r = 1 \mod N$ を満たす r を求めたいわけである。（答えを言うと、先に述べたフーリエ変換を主に使って解くのだが、）まずはじめに次の重ね合わせ状態を作る。

$$\frac{1}{q}\sum_{a=0}^{q-1}\sum_{c=0}^{q-1}\exp\left[\frac{2\pi i}{q}ac\right]|c\rangle|x^a \mod N\rangle \tag{61}$$

q は 2^n で表される。

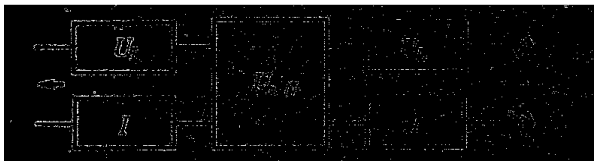


図 12: 位数を求めるのに使われる量子コンピュータ

図のような回路を用意する。ただし

$$U|x\rangle = \sqrt{\frac{1}{q^2}}\sum_{y=0}^{q-1}\exp\left[\frac{2\pi i}{q}xy\right]|y\rangle \tag{62}$$
$$U_{a,N}|x, 0\rangle = |x, a^x \mod N\rangle \tag{63}$$

と変換するものである。U についてはただのフーリエ変換だが、$U_{a,N}$ はかなり込み入った回路になるので具体的なゲートは後回しにして、とりあえずこのような変換のするゲートが組めるとする。ここからは図の通りの 3 段階の変換をいちいちやっていく事にしよう。まず最初の変換は

$$\begin{aligned}\left(U \bigotimes I\right)|0\rangle \bigotimes |0\rangle &= (U|0\rangle)|0\rangle \\ &= \sqrt{\frac{1}{q^2}} \sum_{y=0}^{q-1} \exp\left[\frac{2\pi i}{q} 0y\right] |y\rangle|0\rangle \\ &= \sqrt{\frac{1}{q^2}} \sum_{y=0}^{q-1} |y\rangle|0\rangle \\ &= \sqrt{\frac{1}{q^2}} \sum_{y=0}^{q-1} |y, 0\rangle \end{aligned} \tag{64}$$

この出力を持ったまま第二変換に移る。

$$\sqrt{\frac{1}{q^2}} U_{a,N} \left(\sum_{y=0}^{q-1} |y, 0\rangle \right) = \sqrt{\frac{1}{q^2}} \sum_{y=0}^{q-1} |y, a^y \mod N\rangle \tag{65}$$

これに、再びフーリエ変換をほどこしてやると

$$\begin{aligned} U\left(\sqrt{\frac{1}{q^2}} \sum_{y=0}^{q-1} |y, a^y \mod N\rangle \right) &= \sqrt{\frac{1}{q^2}} \sum_{y=0}^{q-1} U|y\rangle \bigotimes |a^y \mod N\rangle \\ &= \frac{1}{q} \sum_{y=0}^{q-1} \sum_{c=0}^{q-1} \exp\left[\frac{2\pi i}{q} yc\right] |c\rangle|x^y \mod N\rangle \end{aligned} \tag{66}$$

ここで量子数 c ,$x^k \mod N$ を得る確率を、量子力学より計算しよう。ただしここで注意が必要である。$x^a \mod N = x^k \mod N$ を満たす a は複数あり、上記の確率を求めるにはその a 群の和をとってやらないといけない。これは、b を整数として、$a = br + k$ で表す事の出来る a の和をとってやると解決される。 r は $x^r = 1 \mod N$ である事を思い出すと

$$\begin{aligned} x^{br+k} \mod N &= (x^r)^b x^k \mod N \\ &= (1 + lN)^b x^k \mod N \ (l \text{ は自然数}) \\ &= \left[\{(1 + lN) \mod N\}^b x^k \mod N\right] \mod N \\ &\text{だが、}(1 + \mod N) = 1 \text{ より} \\ &= (x^k \mod N) \mod N \\ &= x^k \mod N \\ &\text{ただしここの途中計算で } ab \mod N = [(a \mod N)(b \mod N)] \mod N \text{ となる事を用いた。} \end{aligned}$$

である事より示される。つまり $P(c, x^k \mod N) \equiv P(c, k)$ は

$$\frac{1}{q^2} \left| \sum_{a=0,\ \text{ただし } a = br+k \text{ を満たす a}}^{q-1} \exp\left[\frac{2\pi i}{q} ac\right] \right|^2 \tag{67}$$

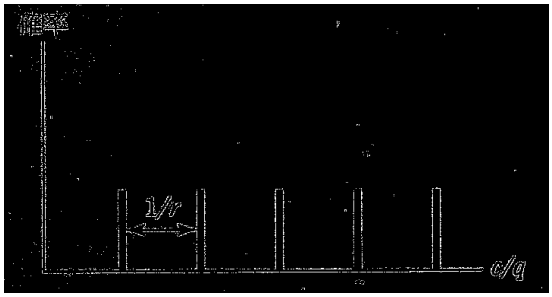
となり、これを少し変形すると

$$\frac{1}{q^2}\left|\sum_{b=0}^{\lfloor\frac{q-k-1}{r}\rfloor}\exp\left[\frac{2\pi i}{q}(br+k)c\right]\right|^2$$
$$=\frac{1}{q^2}\left|\sum_{b=0}^{\lfloor\frac{q-k-1}{r}\rfloor}\exp\left[\frac{2\pi i}{q}brc\right]\right|^2 \tag{68}$$

を得る。この和が q の倍数に近いところで鋭いピークを持つ事は納得でき、厳密にも示せるので、測定した c の値とはじめから用意した q から求める量 r を割り出す事が出来る。式で書けば

$$\frac{c}{q}=\text{整数}\times\frac{1}{r} \tag{69}$$

となる。



5.2 連分数を使って位数を求める

今やったことを大雑把に言えば、$\frac{c}{q}$ を多数回プロットして位数 r を求めるというものだった。今回のやり方は $q > N^2$ であれば一回 c を測定しただけで r を求める事が出来るといったものである。ただし、出た答えが正しくない可能性もあるのだが、その事については後で述べる。この方法を行ううえで最も核となる命題を証明しておく。

命題

$\left|\frac{d}{r}-\frac{c}{q}\right|\leq\frac{1}{2q}$ を満たす $\frac{d}{r}$ は高々1つしかない。ただし d は $\gcd(r,d)=1$ を満たす

証明

もし2つあったとして、それらを $\frac{d_1}{r_1},\frac{d_2}{r_2}$ とすると

$$\left|\frac{d_1}{r_1}-\frac{d_2}{r_2}\right|=\left|\frac{d_1}{r_1}-\frac{c}{q}+\frac{c}{q}-\frac{d_2}{r_2}\right|\leq\left|\frac{c}{q}-\frac{d_1}{r_1}\right|+\left|\frac{c}{q}-\frac{d_2}{r_2}\right|\leq\frac{1}{2q}+\frac{1}{2q}=\frac{1}{q}\leq\frac{1}{N^2} \tag{70}$$

最後の不等号で $q > N^2$ を使っている事に注意。ここで左端と右端を比べると $\frac{1}{N^2} \geq \left|\frac{d_1}{r_1} - \frac{d_2}{r_2}\right|$ を得、さらに

$$\frac{1}{N^2} \geq \left|\frac{d_1}{r_1} - \frac{d_2}{r_2}\right| = \frac{|d_1 r_2 - d_2 r_1|}{r_1 r_2} > \frac{|d_1 r_2 - d_2 r_1|}{N^2} \tag{71}$$

すなわち

$$|d_1 r_2 - d_2 r_1| < 1 \tag{72}$$

となるが、少し考えればそのような $|d_1 r_2 - d_2 r_1| < 1$ は $d_1 r_2 - d_2 r_1 = 0$ しかないことがわかるので、従って $\frac{d_1}{r_1} = \frac{d_2}{r_2}$

ただしここで $N \geq r$(それゆえ $N^2 > r_1 r_2$) である事を用いているが、この証明は省略する。

以上より $q > N^2$ であるとき、測定値（$\frac{c}{q}$）を一つ出せば、うえの不等式を満たす $\frac{d}{r}$ は一つしかないことから、$\frac{c}{q}$ を連分数近似してその $\frac{c}{q}$ を探し出せば同時に r が求まるわけである。これが c を 1 回測定しただけで r を知ることのできる理由となる。少し例を挙げて考えよう

$N = 15, q = 256(< N^2), x = 4$ の問題に対し $c = 18$ が観測されたとする

$\frac{18}{256}$ を連分数近似

$$\frac{18}{256} = \frac{18}{14 \times 18 + 4} = \frac{1}{14 + \dfrac{4}{18}} = \frac{1}{14 + \dfrac{4}{4 \times 4 + 2}} = \frac{1}{14 + \dfrac{1}{4 + \dfrac{1}{2}}} \tag{73}$$

$$= [14, 4, 2] \tag{74}$$

と表される。ここで 0 次、つまり $\frac{d}{r} = \frac{1}{14}$ で計算してみると、これは $\left|\frac{c}{q} - \frac{d}{r}\right| < \frac{1}{2q}$ を満たすことが分かる。よって位数 $r = 14$ を得て、これを因数分解をアルゴリズムに入れてやれば、因数 3，5 を得る。ただし得られた位数が欲しくない因数 1，15 を出す事もあるがその場合は運が悪かったと思ってもう一度やり直す。

同じ問題で $c = 48$ を得た場合

$$\frac{48}{256} = \frac{3}{16} = \frac{1}{5 + \dfrac{1}{3}} = [5, 3] \tag{75}$$

0 次、$\frac{1}{5}$ をとると、これは $\left|\frac{c}{q} - \frac{d}{r}\right| < \frac{1}{2q}$ を満たさない。1 次、$\frac{3}{16}$ では（残りはこれしかないので必ずうまくいくのだが）$\left|\frac{c}{q} - \frac{d}{r}\right| < \frac{1}{2q}$ を満たし、$r = 16$ を得るが、これは N > r に反する。つまりこの観測量では正しくない位数が求まってしまうわけである。

まとめると結局

$$C:\left\{\exists d\in\{1,2,...,r-1\},\quad \gcd(r,d)=1,\quad \left|\frac{c}{q}-\frac{d}{r}\right|<\frac{1}{2q}\quad ,r<N\right\} \tag{76}$$

を満たす c を観測できればいいわけである。先ほどの 2 番目の例では r < N を満たさない r なので上の条件には入れないわけである。

次に、そのような観測量 c ∈C が得られる確立を計算してみる（わざわざこんな計算をやるのだから当然答えは高い確率になる）前に計算したとおり P（c,k）は $\frac{1}{q^2}\left|\sum_{b=0}^{\lfloor\frac{q-k-1}{r}\rfloor}\exp\left[\frac{2\pi i}{q}brc\right]\right|^2$ で与えられるのでこれを等比級数の和とみれば

$$\frac{1}{q^2}\left|\sum_{b=0}^{\lfloor\frac{q-k-1}{r}\rfloor}\exp\left[\frac{2\pi i}{q}brc\right]\right|^2=\frac{1}{q^2}\left|\frac{1-\exp\left\{(\frac{2\pi i}{q}rc)\times\lfloor\frac{q-k-1}{r}\rfloor\right\}}{1-\exp(\frac{2\pi i}{q}rc)}\right|^2$$

ここで

$$\theta=\frac{2\pi rc}{q} \tag{77}$$

$$\alpha_k=\lfloor\frac{q-k-1}{r}\rfloor \tag{78}$$

と定義して、$|1-\exp(i\theta)|^2=2(1-\cos\theta)$ に注意すると $P(c,k)$ は

$$\frac{1}{q^2}\frac{1-\cos\alpha_k\theta}{1-\cos\theta}$$

となる。また $\left|\frac{c}{q}-\frac{d}{r}\right|<\frac{1}{2q}$ より θ について次の条件式が導ける。

$$\begin{aligned}&\left|\frac{c}{q}-\frac{d}{r}\right|<\frac{1}{2q}\qquad(\text{両辺 }2\pi r\text{ をかけて})\\&\Leftrightarrow|2\pi d-\theta|<\frac{\pi r}{q}\\&\Leftrightarrow 2\pi d-\frac{\pi r}{q}\le\theta\le 2\pi d+\frac{\pi r}{q}\\&\therefore-\frac{\pi r}{q}\le\theta\le\frac{\pi r}{q}\end{aligned}$$

ところが $r\ll N$ であれば（そう定義する）θ の絶対値はとても小さいという事がわかる。そこで $\cos\theta\sim 1-\frac{\theta^2}{2}$ とすれば

$$P(c,k)\sim\frac{2\alpha_k^2}{q^2}\frac{1-\cos\alpha_k\theta}{(\alpha_k\theta)^2} \tag{79}$$

と近似できる。次に α_k は k に依存するので、$\alpha_k\theta$ のとり得る値について考

える。$-\frac{\pi r}{q} \leq \theta \leq \frac{\pi r}{q}$ より、

$$\begin{aligned} |\alpha_k \theta| &\leq \frac{\pi r}{q} \times \quad \lfloor \frac{q-k-1}{r} \rfloor \\ &\leq \frac{\pi r}{q} \times \frac{q-k-1}{r} = \pi \left(1 - \frac{k+1}{q} \right) \end{aligned}$$

ここで $k < r, r \ll q$ より $\frac{k+1}{q} \sim 0$ だから　$|\alpha_k \theta|$ について拘束条件

$$|\alpha_k \theta| \leq \pi \tag{80}$$

を得る。ここで一つ補題を用意しておく

補題

$$f(x) = \begin{cases} \frac{1-\cos x}{x^2} & if \quad x \neq 0 \\ \frac{1}{2} & if \quad x = 0 \end{cases} \tag{81}$$

は閉区間 $[-\pi, \pi]$ において $x = \pi$ で最小となる。

この証明は簡単で、見た目から明らかなので省略する。

この補題を用いれば $P(c,k)$ は $|\alpha_k \theta| = \pi$ で最小だから、

$$P(c,k) \geq \frac{2\alpha_k^2}{q^2} \frac{1-\cos \pi}{\pi^2} = \frac{4\alpha_k^2}{\pi^2 q^2} \tag{82}$$

と下から評価される。更に

$$\alpha_k \lfloor \frac{q-k-1}{r} \rfloor \geq \frac{q-k-1}{r} = \frac{q}{r} - 1 \tag{83}$$

だから、最終的に

$$P(c,k) \geq \frac{4}{\pi^2 q^2} \left(\frac{q}{r} - 1 \right)^2 \sim \frac{4}{\pi^2 r^2} \tag{84}$$

と評価される。

以上のことからさらに次の命題が示せる。

命題

量子コンピュータ U によって、x の　mod N に関する正しい位数が得られる確立は

$$P_r \geq \frac{\tau}{\log \log N} \tag{85}$$

と、下から押さえられる。

証明

先ほど出した確率を使って、

$$P_r \geq \sum_{c \in C} P(c,k) = \sum_{c \in C} \sum_{k=0}^{r-1} P(c,k) \geq \sum_{c \in C} \frac{4}{\pi^2 r} \tag{86}$$

を得る。ここで $\left|\frac{c}{q}-\frac{d}{r}\right|<\frac{1}{2q}$ から、

$$\frac{qd}{r}-\frac{1}{2}\leq c\leq\frac{qd}{r}+\frac{1}{2}$$

とでき、この式からどんな d に対しても少なくとも（式を満たす c が）一つは存在する事が見て取れる。ところで $\gcd(r,d)=1$ を満たす d の個数を $\phi(r)$ とすると（この関数の事を Euler のファイ関数という）、上の c はどんな d に対しても一つ以上（最高で 2 つ）ペアがあるので $\phi(r)$ を下回る事はない。さらに $\phi(r)$ に関して

$$\lim_{r\to\infty}\frac{\phi(r)\log\log r}{r}=e^{-\gamma}\qquad(\text{ただし }\gamma\text{ は Euler の定数})\tag{87}$$

という事実が知られているので、これを使えば大きい r に対して

$$\phi(r)\geq\frac{e^{-\gamma}r}{\log\log r}\geq\frac{e^{-\gamma}r}{\log\log N}\tag{88}$$

が成立する事になる。これを (86) 式に使ってやると結局

$$P_r\geq\sum_{c\in C}\frac{4}{\pi^2 r}\geq\frac{4}{\pi^2 r}\phi(r)\geq\frac{4e^{-\gamma}}{\pi^2\log\log N}\tag{89}$$

証明終了。

以上より量子コンピュータ U による計算を $O(\log\log N)$ 回実行し、そのつど連分数を使って r を出せば、ほぼ 1 の確率で正しい位数が求められる事になる。

6　グローバーのアルゴリズム

N 個のファイルがありその中に一個だけ「正しいファイル」があるとしよう。これから相手にする問題は、その「正しいファイル」を見つける早いアルゴリズムを見つけてくことである。これを古典計算 (オーソドックスにランダムに一つずつ選ぶやり方) でやろうとすると、N 個のファイルから 1 個のファイルを見つけるには平均で $\frac{N}{2}$ 回かかるが、量子計算でこのアルゴリズムを用いると $\sqrt{N}$ 回のステップで計算が行える。

因数分解のときのように多項式時間（$\log N$ のべき程度）のアルゴリズムでないが、量子計算の特徴が見えやすく、その過程はとても面白いものである。

6.1　グローバーによるアルゴリズム（ファイル検索）

問題を再定義しておこう。古典で言う N 個の中、というのは等しい重みの状態が N 個あるということであるので、問題は

等しい重みの状態が N 個あり、その中から特定の状態 $|w\rangle$ を求める。

としておこう。すると始状態は

$$|\Psi\rangle = |s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \tag{90}$$

でスタートできる。アルゴリズムの概要は、この重ね合わせ状態に 2 つのあるユニタリー変換を、交互に行う事によって（この操作に $\sqrt{N}$ のステップがいる）欲しいファイルの確率振幅を増幅させほぼ 1 の確率でファイルを抜き取る、といったものである。始めにその 2 つのユニタリ変換とはどういったものか調べたい。2 つの変換は

$$U_1 = 1 - 2|w\rangle\langle w| \qquad U_2 = 2|s\rangle\langle s| - 1 \tag{91}$$

で定義される。まずこの 2 つがユニタリである事はちょっとした計算で確かめる事は出来るだろう。ではさっそく U_1、U_2 を解析していく。

$U_1 = 1 - 2|w\rangle\langle w|$

簡潔に言うと全状態の中から $|w\rangle$ の符号を変える変換であるのだが、これをまず行列で表記しておこう。

$$U_1 = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & -1 & & & \\ & & & & \ddots & & \\ & & & & & 1 & \\ & & & & & & 1 \end{pmatrix} \tag{92}$$

となるが（ただし-1 は w 行 w 列目）、これは $f(x)$ を

$$f(x) = 0 \quad (\forall a \neq w), \qquad f(w) = 1 \tag{93}$$

としたもの（オラクル関数と呼ぶ）を使って、

$$U_1 = (-1)^{f(x)}\delta_{xy} \tag{94}$$

としたものと同じである。ところでこれは以前定義した選択的回転変換を $\theta_x = 0 \quad \theta_w = \pi$ をした形である事に気づく。何が言いたかったかというと、このように特定の状態の符号を変える変換の回路は制御 U と NOT で作る事が可能だということである。

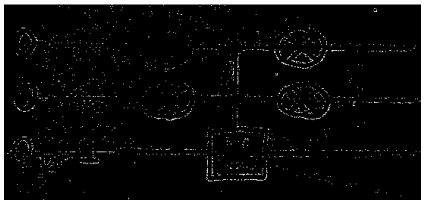


図 13: 選択的回転変換の簡単な例

図の例で考えてみよう。この回路の場合 $|0\rangle|0\rangle|0\rangle, |0\rangle|0\rangle|1\rangle$ のときのみ U_2 がかかる。例えば $|0\rangle|0\rangle|1\rangle$ だけにマイナスをつけたいなら

$$U_0 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

とすれば、

$$|0\rangle|0\rangle|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |0\rangle|0\rangle|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

としたとき

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

と首尾よく $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ を得る。

$U_2 = 2|s\rangle\langle s| - 1$

詳しくは後でまたふれるので、この変換は式どおり

$$(2|s\rangle\langle s| - 1)|x\rangle = 2\sqrt{\frac{1}{N}}|s\rangle - |x\rangle \quad (\because \langle s||x\rangle = \sqrt{\frac{1}{N}}\sum_{x'=0}^{N-1}\langle x'||x\rangle = \sqrt{\frac{1}{N}}) \tag{95}$$

を行うということだけ示しておく。

以上を踏まえた上で、$U = U_2U_1$ が、ある 2 次元空間を回転させている事が分かる。その事を示そう。まず、そのある 2 次元空間とは

$$|\alpha\rangle \equiv \frac{1}{\sqrt{N-1}}\sum_{x=0,x\neq w}^{N-1}|x\rangle \tag{96}$$

$$\frac{1}{\sqrt{N}}|w\rangle \tag{97}$$

と定義したものである。すると始状態は

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle = \frac{\sqrt{N-1}}{\sqrt{N-1}} \frac{1}{\sqrt{N}} \left\{ \sum_{x=0, x\neq w}^{N-1} |x\rangle + |w\rangle \right\} \tag{98}$$

$$= \sqrt{\frac{N-1}{N}} |\alpha\rangle + \sqrt{\frac{1}{N}} |w\rangle \tag{99}$$

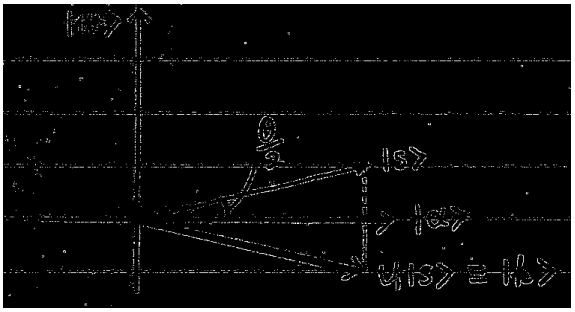
と表せる。ここで便宜上

$$\cos\frac{\theta}{2} \equiv \sqrt{\frac{N-1}{N}} \quad , \sin\frac{\theta}{2} \equiv \sqrt{\frac{1}{N}}$$

としておこう。これに U_1 をかけると、

$$U_1|s\rangle \to \cos\frac{\theta}{2}|\alpha\rangle - \sin\frac{\theta}{2}|w\rangle \tag{100}$$

これを図示すると下のようになる。



次に U_2 がこの空間で何をするか、言葉では説明しづらいので図で見ていく。まず

$$(2|s\rangle\langle s| - 1)|k\rangle = 2\langle s||k\rangle|s\rangle - |k\rangle \tag{101}$$

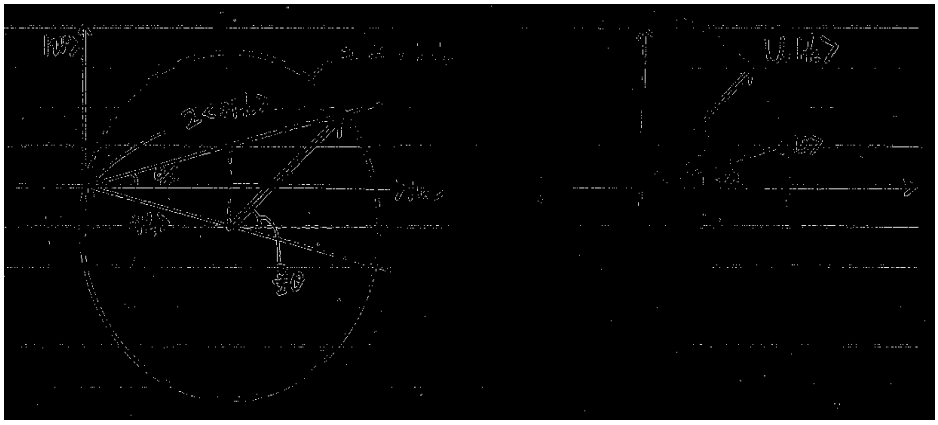
であるから（$|k\rangle$ は U_1 を施した後のベクトル）U_2 を施した後のベクトルをユニタリ性に注意しながら図で表すと下図のようになる。

つまり U_2 は $U_1|s\rangle$ を $|s\rangle$ について対称移動させていることになる。結局 U は

$$U = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \tag{102}$$

の働きをする。グローバーのアルゴリズムとは、この変換を適当な回数だけ行い、ベクトルを限りなく $|w\rangle$ に近づけた上で状態を観測しよう、というものである。その適当な回数は計算でき

$$U^k|s\rangle \to \cos\left(\frac{(2k+1)\theta}{2}\right)|\alpha\rangle + \sin\left(\frac{(2k+1)\theta}{2}\right)|w\rangle$$



N を大きくとって

$$\sqrt{\frac{1}{N}} = \sin\frac{\theta}{2} \overset{N\to\infty}{\to} \sqrt{\frac{1}{N}} \sim \frac{\theta}{2}$$
$$\therefore \theta \sim 2\sqrt{\frac{1}{N}}$$

とすれば、簡単な計算で $k \sim \frac{\pi}{4}\sqrt{N}$ 回で、ほぼ確率 1 で $|w\rangle$ を見つける事がわかる。始めに述べた $\sqrt{N}$ ステップと一致しているのがこれで見える。[3]

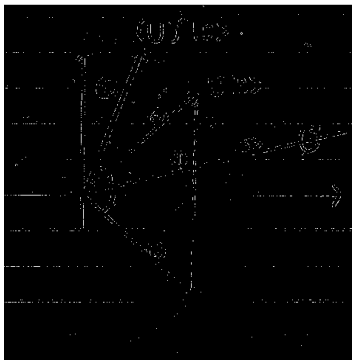


図 14: 2 回 U を施した場合

6.2 グローバーのアルゴリズムの局所性

局所性とは砕いて言うと回路が組めるか可能かということである。回路の可能とは一つは制御 NOT と 1 ビットのユニタリ変換で回路が組めるか可能かということと、その回路が少数で組めるかどうかという事である。U_1 についてはすでに述べた。U_2 はどうであろうか？

ここで関係ないように見えるかもしれないが-HRH がどういった変換をするかみる。ただし H はウォルシュ・アダマール変換、R は選択的回転変換の $|0\rangle$ の符号のみを変換するものである。

$$
\begin{aligned}
&H|x\rangle = \sqrt{\frac{1}{N}}\sum_{y=0}^{N-1}(-1)^{x\cdot y}|y\rangle \\
\Rightarrow &RH|x\rangle = -2\sqrt{\frac{1}{N}}|0\rangle + H|x\rangle \\
\Rightarrow &-HR|x\rangle = 2\sqrt{\frac{1}{N}}H|0\rangle - H^2|x\rangle = 2\sqrt{\frac{1}{N}}|s\rangle - |x\rangle \quad (\because H^2 = 1)
\end{aligned}
$$

だが、ここで

$$(2|s\rangle\langle s| - 1)|x\rangle = 2\sqrt{\frac{1}{N}}|s\rangle - |x\rangle$$

である事から、$U_2 = -HRH$ といえる事が分かる。つまり回路は少数で組めるわけである。

6.3 複数のファイル

次に N このファイルの中から t 個、取り出したいファイルがある問題を考える。しかし基本的には一個の場合と何も変わりはない。

一つのときのような 2 次元の空間があると便利であった事はいうまでもないだろう、したがって今回も類似した次のような 2 つの空間を定義したい、しかしそのまえに次の基底を用意しておこう

$$|\tilde{w_j}\rangle = \sqrt{\frac{1}{t}}\sum_{a=0}^{t-1}\exp\left[\frac{2\pi i}{t}ja\right]|w_a\rangle \quad (j = 0, 1, ...t-1) \tag{103}$$

これはフーリエ変換を用いれば可能だろう。これをつかって、ここに 2 つの空間を定義する

$$|\tilde{w_0}\rangle \equiv \sqrt{\frac{1}{t}}\sum_{a=0}^{t-1}|w_a\rangle \tag{104}$$

$$|r\rangle \equiv \sqrt{\frac{1}{N-t}}\sum_{a=0, a\neq\{w_0, w_1 ... w_{t-1}\}}^{t-1}|a\rangle \tag{105}$$

一見基底が違うので驚くかもしれないが、もとの基底で考えれば線形独立で

ある事は容易に分かる。これを使って始状態は前回同様

$$|s\rangle = \sqrt{\frac{1}{N}}\sum_{a=0}^{t-1} = \sqrt{\frac{N-t}{N-t}}\sqrt{\frac{1}{N}}\left\{\sum_{a=0,a\neq\{w_0,w_1\ldots w_{t-1}\}}^{t-1}|a\rangle + \sum_{a=0}^{t-1}|w_a\rangle\right\}$$
$$= \sqrt{\frac{N-t}{N}}|r\rangle + \sqrt{\frac{1}{N}}\sum_{a=0}^{t-1}|w_a\rangle$$
$$= \sqrt{\frac{N-t}{N}}|r\rangle + \sqrt{\frac{t}{N}}|\tilde{w_0}\rangle \tag{106}$$

と変形でき、2 つのユニタリ演算子は

$$U_2 = 2|s\rangle\langle s| - 1 \tag{107}$$
$$U_1 = 1 - 2\sum_{j=0}^{t-1}|\tilde{w_j}\rangle\langle\tilde{w_j}| \tag{108}$$

で定義する。すると実際

$$U_1\left\{\sqrt{\frac{N-t}{N}}|r\rangle + \sqrt{\frac{t}{N}}|\tilde{w_0}\rangle\right\}$$
$$= \sqrt{\frac{N-t}{N}}|r\rangle + \sqrt{\frac{t}{N}}|\tilde{w_0}\rangle - 2\sqrt{\frac{N-t}{N}}\sum_{j=0}^{t-1}\langle\tilde{w_j}||r\rangle|\tilde{w_j}\rangle - 2\sqrt{\frac{t}{N}}\sum_{j=0}^{t-1}\langle\tilde{w_j}||\tilde{w_0}\rangle|\tilde{w_j}\rangle$$

ここで $\tilde{w_j}$ は正しいファイルの重ね合わせ、$|r\rangle$ は正しくないファイルの重ね合わせであることから

$$\langle\tilde{w_j}||r\rangle = 0 \tag{109}$$

さらに 4 項目は、$j = 0$ のとき

$$\langle\tilde{w_0}||\tilde{w_0}\rangle = 1 \tag{110}$$

$j \neq 0$ のとき

$$\left\{\sum_{a=0}^{t-1}\exp\left[-\frac{2\pi i}{t}ja\right]\langle w_a|\right\}\sum_{a=0}^{t-1}|w_a\rangle = \sum_{a=0}^{t-1}\exp\left[-\frac{2\pi i}{t}ja\right] = 0 \tag{111}$$

であるから結局

$$U_1|s\rangle = \sqrt{\frac{N-t}{N}}|r\rangle - \sqrt{\frac{t}{N}}|\tilde{w_0}\rangle \tag{112}$$

これに U_2 をかければ、これも前回同様、幾何学的に $U_2(U_1|s\rangle) = (2|s\rangle\langle s| - 1)(U_1|s\rangle) = A|s\rangle - U_q|s\rangle$　(A は規格化定数) だから

あとは

$$\cos\frac{\theta}{2} = \sqrt{\frac{N-t}{N}}, \quad \sin\frac{\theta}{2} = \sqrt{\frac{t}{N}} \tag{113}$$

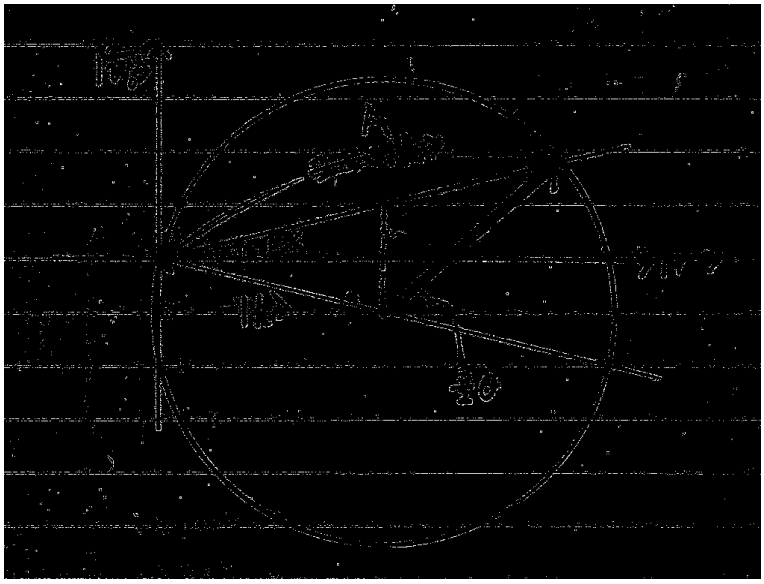


図 15: 複数のファイルの場合

とすれば U は $|r\rangle$　　$|\tilde{w_0}\rangle$ 基底で、回転

$$U = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \tag{114}$$

を表す。何度も恐縮だがこれも前回と同じようなやり方で適当な回数 k を計算でき結果だけ記すと

$$k \sim \frac{\pi}{4}\sqrt{\frac{N}{t}} \tag{115}$$

を得る。ここで注意して欲しいのは、このアルゴリズムでファイルの取り出しに成功した場合、一まとめに取り出すという事である。取り出す状態が $|\tilde{w_0}\rangle = \sqrt{\frac{1}{t}}\sum_{a=0}^{t-1}|w_a\rangle$ に注意すれば分かるだろう。こういった意味で、古典的に複数のファイルを取り出すとは少し意味が違うかもしれない。

6.4　量子勘定

今やったことは正しいファイルの個数 t が既知の時計算ステップは $\sqrt{\frac{N}{t}}$ 回という事だった。そこで次に t いくつかが分からない、といった場合の t を求めてみよう。まず状態に U を m 回演算してやる。すると始状態は

$$|s\rangle \to \sin(2m+1)\theta|g\rangle + \cos(2m+1)\theta|b\rangle \tag{116}$$

となる事はもういいだろう。ただしここでは

$$\sin\theta = \sqrt{\frac{t}{N}}$$
$$|g\rangle = \sqrt{\frac{1}{t}} \sum_{good} |a\rangle$$
$$|b\rangle = \sqrt{\frac{1}{N-t}} \sum_{bad} |a\rangle$$
$$U_1 = 1 - 2 \sum_{good} |a\rangle\langle a| \tag{117}$$

としてある。ここで m 回を一周期をとして、その回数から t を求められないかというのが今回の目標である。そんな周期の求め方は実はすでに、間接的だがやっている。ショアのフーリエ変換を用いたアルゴリズムである。

周期を求める

初期状態 $\sqrt{\frac{1}{N}} \sum_{a=0}^{N-1} |a\rangle$ に $\sqrt{\frac{1}{P}} \sum_{m=0}^{P-1} |m\rangle$ を付け足す。　（N,P は 2 のべき乗）つまり

$$\sqrt{\frac{1}{P}} \sum_{m=0}^{P-1} |m\rangle \sqrt{\frac{1}{N}} \sum_{a=0}^{N-1} |a\rangle \tag{118}$$

を始めに準備する。

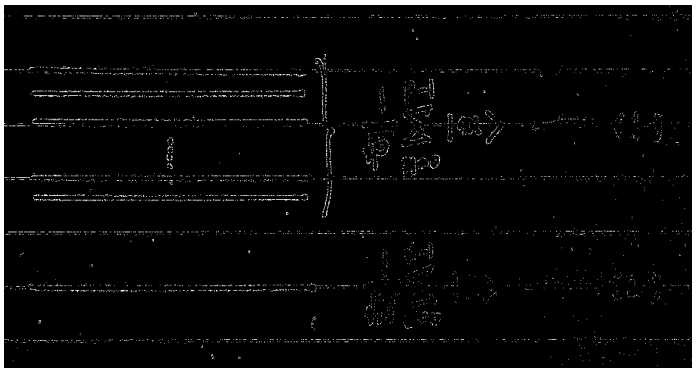


図 16: 周期を求める

これに、(2) の回路にだけ U を m 回（(1) の回路と一致するように）演算してやる。

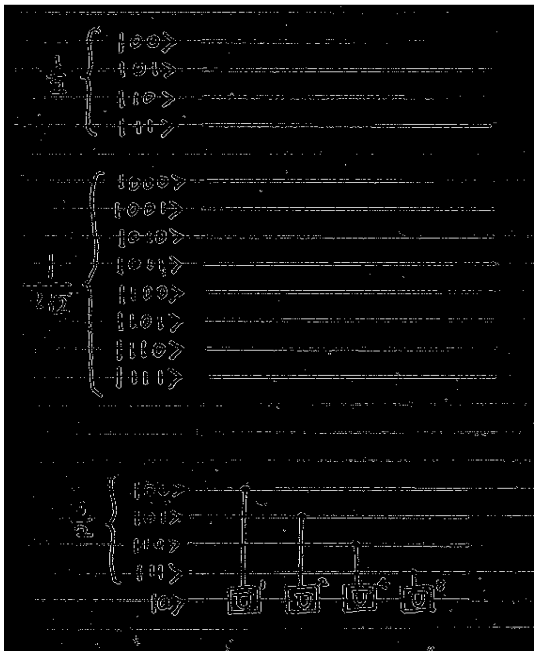


図 17: P=4 の例

式にすると

$$|s'\rangle \to |s''\rangle = \sqrt{\frac{1}{P}}\sum_{m=0}^{P-1}|m\rangle\sqrt{\frac{1}{N}}\sum_{a=0}^{N-1}U^m|a\rangle$$
$$= \sqrt{\frac{1}{P}}\sum_{m=0}^{P-1}|m\rangle\sqrt{\frac{1}{N}}\sum_{a=0}^{N-1}\left[\sin(2m+1)\theta|g\rangle + \cos(2m+1)\theta|b\rangle\right] \tag{119}$$

ここで $|s''\rangle$ をフーリエ変換して　$\left(|m\rangle \to \sqrt{\frac{1}{P}}\sum_{n=0}^{P-1}\exp\left[\frac{2\pi i}{P}nm\right]|n\rangle\right)$

$$|s''\rangle \to |s'''\rangle = \frac{1}{P}\sum_{m=0}^{P-1}\sum_{n=0}^{P-1}e^{\frac{2\pi inm}{P}}|n\rangle\left[\sin(2m+1)\theta|g\rangle + \cos(2m+1)\theta|b\rangle\right] \tag{120}$$

ここで m について和をとると、計算は省略するが

$$|s'''\rangle = \frac{1}{2}e^{\frac{-i\pi n}{P}}\sum_{n=0}^{P-1}|n\rangle\left[e^{i\pi f}\frac{\sin\pi(n+f)}{P\sin\frac{\pi(n+f)}{P}}(-i|g\rangle+|b\rangle) + e^{-i\pi f}\frac{\sin\pi(n-f)}{P\sin\frac{\pi(n-f)}{P}}(i|g\rangle+|b\rangle)\right] \tag{121}$$

を得る。　$\left(f = \frac{P\theta}{\pi}\right)$ この振幅は、$n = [f]$ と $[P-f]$ にピークを持つので、$|n\rangle$ を観測すれば、高い確率で f を得る。それから、$\frac{t}{N} = \sin^2\theta = \sin^2\left(\frac{\pi f}{P}\right)$ によって、角度 θ そして t を得る。

またその精度は

$$\frac{t}{N} = \sin^2\theta \quad \text{より}$$
$$\frac{\Delta t}{N} = \sin^2(\theta+\Delta\theta) - \sin^2\theta$$
$$= (\sin(\theta+\Delta\theta) + \sin\theta)\,|\sin(\theta+\Delta\theta) - \sin\theta|$$

ここで $|\sin(\theta+\Delta\theta) - \sin\theta| \leq |\Delta\theta|$

$$|\sin(\theta+\Delta\theta)| < \sin\theta + |\Delta\theta| \text{ だから}$$
$$\text{与式} < (2\sin\theta + |\Delta\theta|)\,|\Delta\theta|$$
$$\text{従って}\frac{\Delta t}{N} < (2\sin\theta + |\Delta\theta|)\,|\Delta\theta|$$
$$\Leftrightarrow \Delta t < \left(2\sqrt{\frac{t}{N}} + |\Delta\theta|\right)|\Delta\theta| \times N$$

$|\Delta\theta| \leq 2^{-m} \times \pi$ より

$$\Leftrightarrow \Delta t < \left(2\sqrt{\frac{t}{N}} + \frac{\pi}{2^m}\right)\frac{N}{2^m}\pi$$
$$\Leftrightarrow \Delta t < \left(2\sqrt{\frac{t}{N}} + \frac{\pi}{P}\right)\frac{N}{P}\pi$$

$$\therefore \Delta t < \frac{N\pi}{P}\left[\frac{\pi}{P} + 2\sqrt{\frac{t}{N}}\right] \tag{122}$$

となる。P を大きくすれば誤差はいくらでも小さくなるが、逆に計算ステップは多くなってしまう。

7　付録 (量子暗号)

ここからは話す内容はタイトル通り量子暗号で、私がこのテーマを選んだ理由の一つの単語である。量子暗号とは一言で言うと原理的に盗聴不可能な情報処理である。その柱となるのがやはり重ね合わせで、仮に盗聴者が情報を盗聴してしまうと観測によって重ね合わせの状態が収縮してしまうというものである。受信者は重ね合わせが収縮している事に気づくことが出来るので（後に示す）仮に盗聴されたら、されていたという事がわかる。これが間接的だが絶対に盗聴できないというからくりになっている、こういう意味で絶対に安全な暗号というべきかも知れない。

また、量子暗号は量子コンピュータに比べてはるかに実現の可能性が高いといわれている。

7.1　BB84 プロトクル

BB84 というのは 1984、ベネットとブラサードによって発見された量子暗号でプロトクルとは受信者、送信者のある種、決め事のようなものである。今やろうとしている事はこの二人の間で安全に情報の鍵を共有することである。とりあえずこの鍵さえ共有できれば、あとは安全に情報交換ができるとする。本題に入る前に、具体的な例をみながらが考えていったほうが分かりやすいと思われるので、軽い設定をしておこう。

・情報科の用語にのっとって送受信をアリスとボブ、盗聴者をイブする。

・通信手段はもっぱら電話と量子通信機に限る。ただし電話でのアリスとボブのやり取りは完全に盗聴されうる。

・イブは万能で、物理法則の許す限りなんでも出来る。

・一度鍵を共有できればアリスとボブは秘密に情報のやり取りが出来るとする。

・今アリスはボブに部屋の番号を教えたいものとする、ただし二人の関係についてはこれ以上関与しない。

では早速その鍵を共有するまでの順を、追って見ていく。

BB84

1

アリスがボブに電話で、部屋の番号を教えたいので量子通信機をセットしておくように指示する。当然この時点で、イブは自由に盗聴できている。

2

アリスは部屋の番号を教える前に鍵の基となるものを量子通信機を使ってボブに送る（当然ボブもその事は分かっている）ただしここから注意が必要である。この送る鍵の信号は 0 と 1 ではあるのだが 2 種類の 0 と 1 を送る。この 2 種類には実際には不可能であるが電子のスピンを使う、例えば z 方向の 0 か 1 と x 方向の 0 と 1 などである。

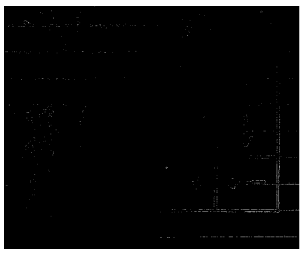


図 18: 2 種類の 0 と 1

その事を踏まえたうえで、アリスは今、鍵 100101 をそれぞれ zxxzzx 方向で電子を送る。このスピンの向き、1，0 はランダムに決めてよい。さらにボブは送られてきた情報を任意の方向で測定していく。

仮にボブが zxzzxx で測定したとすれば 10?1?1 が観測されるはずであるが、ここで?とは 0 か 1 どちらでも取り得るということである。今回は測定した結果 101101 を得たとする。

3

ボブがアリスに電話で受信方法を知らせる（zxzzxx）

4

それを聞いたアリスは、電話で一致している受信法が何番目かボブに伝える。（今回 1,2,4,6 番目ビット）

5

一致している｛0,1｝を鍵とする（今回 1,0,1,1）共有完了。

こうして出来た鍵は完全にランダムな文字列なので、例えば送りたい情報にこの鍵との排他的論理和をとれば送ろうとする情報はいったん完全にランダムになる。実は今、おもむろに排他的論理輪を使えばと言ったがこれには理由がある、次に書く例を見ればすぐ分かると思うが、これにもう一度同じ排他的論理和を加えてやれば（もちろん同じ鍵でないといけない）元の情報に復元されるのだ。こうしてボブは誰にも知られる事なくアリスから部屋番号を得て、無事ゴールに達するわけである。

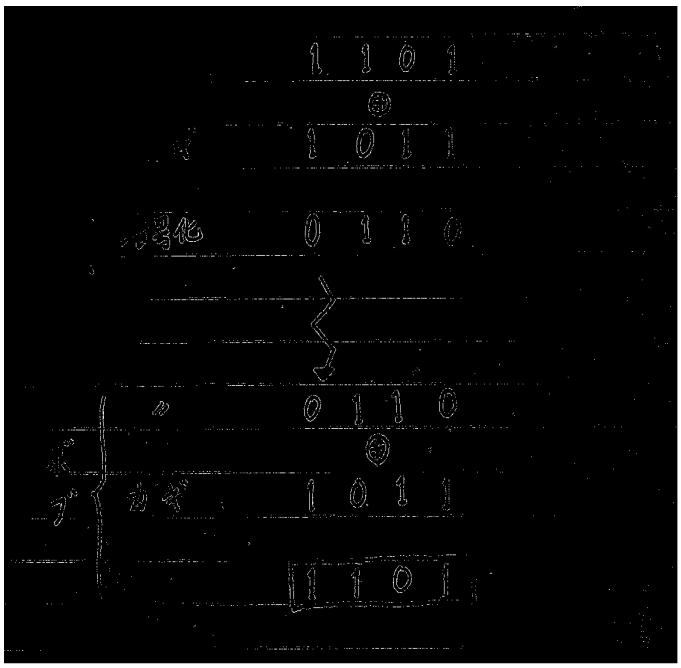


図 19: 暗号の復元

ではなぜこのシステムをとると盗聴されたかがわかるのか。それはアリスとボブとの情報の食い違いによって明らかになる。2.4 節でふれたように重ね合わせの状態を複製する事はできなかった事に注意する。イブは出来るだけ多くの鍵の情報を知りたいがためにアリスの情報を測定し、その情報をボブに送る。しかし測定した時点でアリスの作った重ね合わせを壊してしまう事に他ならない。従ってイブが盗聴した後でボブに情報を送っても、ボブがアリスに（電話で）測定方法の情報を交換した際、物理的にありえない結果が伴う可能性がある。例えばアリスがい i 番目のビットを z 方向で 1 を送り、ボブも i 番目のビットを z 方向で測定したとしよう。当然その場合それに対してイブは、半分の確率で、z 方向で観測したならうまくいくが x で観測してしまった場合はボブが観測するのはイブの作った x 方向の 0 か 1 である。つまり、このときボブはスタート地点から $\frac{1}{4}$ の確率で z 方向の 0 を測定する事になってしまう。これは物理的に誰かが盗聴していなければありえない結果だ。この事を確認するには、仮に 1000 ビット共有したビットがあったとしたら、そのうちの 50 個でもビット合わせをしてみればいいだろう。これだけで十分盗聴されたかされてないか分かるはずである、1 ビットあたりイブがうまくいく確立が $\frac{3}{4}$ であるから $(\frac{3}{4})^{50} \sim 6 \times 10^{-7}$ ほどの確率でしか盗聴できない。これはもう現実的に盗聴不可といって問題ない。

参考文献

[1] 細谷暁夫. 量子コンピュータの基礎. サイエンス社, 1999.

[2] 上坂吉則. 量子コンピュータの基礎数理. コロナ社, 2000.

[3] Isaac L. Chuang /木村達也 訳 Michael A. Nielsen.