量子計算機入門

東京工業大学・理学部　細谷　暁夫

1　量子計算機とは何か

量子計算機について語る前に、普通の計算機（古典計算機と呼ぼう）がどう働いているか知る必要があるだろう。古典計算機は、原理的にテューリングマシンという概念的な計算機に帰着される。テューリングマシンはテープとプロセッサーから成り立っており、テープには0と1の羅列が書き込まれている。プロセッサーにはヘッドが付いていてテープに書いてある数字を読んだり書き換えたりしながら、テープを前後に移動する（図1）。その動きはあらかじめプログラムされている。簡単な例をあげれば、2を掛けるには、掛けられる数をテープに2進法で表しておいて、その末尾に0を書き足せばよい。1を足すには、一番最後の0を1に、それより下の桁の1を0に書き換えればよい。　一般に、はじめにテープに書かれていた0と1の羅列を初期状態と見なし、書き換えられた結果のテープの0と1の羅列を終状態と見なしたとき、計算とはそれらの状態間の遷移であるということができる。計算が可能であるとはテューリングマシンの動きがいつかは停止することであり、近代の論理学の採用しているクライテリオンである。計算が複雑であるとは上記のヘッドの逐次的な動きの回数が多いということである。正確な定義等は、文献[1]をみられたい。

量子計算機が古典計算機と違う点は、量子計算機においては可能な状態として$|0>$と$|1>$だけではなく、（状態であることを強調するためにケットベクトルを導入した。）それらの重ね合わせも許すことである。すなわち、αとβを規格化条件、$|\alpha|^2+|\beta|^2=1$を満たす複素数として

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

なる状態もテープに書き込める。これが物理的に可能であることは、たとえば磁気モーメントをもつスピン1/2の粒子に磁場をかけることを考えればよい。すなわち、はじめにスピンがz軸方向を向いていたとして、それに対して斜めに磁場をかけてスピンの向きを変えてやればよい。ここではスピンのz成分が$+1/2$の状態を$|1>$、$-1/2$の状態を$|0>$と見なす。磁場とz軸のなす角度をθとすれば、例えば$cos\theta/2|0>+sin\theta/2|1>$を得ることができる。

古典計算機における0と1の「書き換え」は量子計算機においては複素2次元空間のユニタリー変換にあたる。「読み出し」については、もっと本質的な違いが起こる。量子力学の公理によれば、スピンのz成分 の観測を行うと$|0>$か$|1>$の状態に遷移し、その確率はおのおの$|\alpha|^2$と$|\beta|^2$で与えられる。

$|0>$と$|1>$の重ね合わせの状態をとりうるものをキュービット（qubit）と呼ぶ。キュービットをN個用意すれば、2^N個の状態の重ね合わせを実現することができる。例えば、N個のキュービットをいっせいに回転して、0から2^N-1までを2進法でラベルされた状態$|a)$を等しい重みでたしあげた重ね合わせ状態が実現できる。式で示せば

$$|0>|0>\ldots|0> \quad \rightarrow((|0>+|1>)/\sqrt{2})^N \tag{2}$$

$$= \frac{1}{\sqrt{2^N}}\sum_{a=0}^{2^N-1}|a) \tag{3}$$

となる。

ドイチは[2]、古典テューリングマシンで計算可能なものは量子テューリングマシンでも計算可能であり、（テューリングマシンがいずれ停止するという意味での）計算可能性に関しては両者は一致することを示している。しかし、計算の速さについては、両者には本質的な差があることが、本稿のテーマである。

量子計算は2^N次元のユニタリー変換を、ゲートと呼ばれる基本的なユニタリー変換の組み合わせで実行する。このユニタリー変換は当然ながら2^N個の状態を同時に変換するので2^N個の並列計算とみなすことができる。

前述の重ね合わせ状態（3）において、奇数のラベルをもつ状態にマイナス符号をつける変換：$\frac{1}{\sqrt{2^N}}\sum_{a=0}^{2^N-1}|a) \rightarrow \frac{1}{\sqrt{2^N}}\sum_{a=0}^{2^N-1}(-1)^a|a)$ を考えよう。古典計算だと、状態一つ一つに当たる作業が必要なので、2^N個の計算を要する。量子計算だと単にN番目のキュービットにおいてユニタリー変換：$|0>\rightarrow|0>,|1>\rightarrow -|1>$ を行うだけでよい。2^N個の状態を操作しているにもかかわらず、計算は一回のユニタリー変換で済むのである。

量子計算では、はじめに簡単に用意できる積による状態を準備する。つぎにあるプログラムされたユニタリー変換によって、重ね合わせの中に望ましい状態を含んでいるものを作る。最後に何回か観測をしてその望ましい状態に波束の収縮をさせて、そこに書き込まれている情報を読みとる。

当然、望ましくない状態に波束が収縮することも起こりうるが、検算をしてそれを排除する。こういう意味で、量子計算は量子力学の本質的な部分だけを用いているということができると思う。したがって、この量子計算に向いている計算とそうでないものがある。例えば、数値積分などは検算のしようがないので、おそらく向かないだろう。これから述べる、因数分解とか、まだできてはいないが巡回セールスマン問題[11]などに向いている。おおまかに言えば、解くには組み合わせ論的に複雑であるが、検算が容易な問題に適している。

2 量子論理ゲート

古典計算機における代表的な論理ゲートとしては、NOT, OR, AND, EXCLUSIVE-OR(XOR)[1] などがある。例えば、 AND と NOT で万能のテューリングマシンができる。しかし上に述べたように、量子計算はユニタリー変換のことであるから、逆変換のあるゲートしか使えない。NOT は $|0>$ を $|1>$に $|1>$を $|0>$に「天の邪鬼」の遷移をさせるので逆を持つが、他の古典ゲートは逆を持たない。[2]

量子計算のダイアグラムの見方を説明しよう（図2参照)。横線はキュービットをあらわし、そのうえに $|0>$と $|1>$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。縦線はゲートと呼ばれるキュービット間の相互作用をあらわし、ゲートを通過するたびに、状態は指定されたユニタリー変換を受ける。

量子回路と初めに述べた量子テューリングマシンとの関係は、大ざっぱには以下の通りである。まず、量子テューリングマシンの升目の一個にあたっているのがキュービットの横線一本で、量子テューリングマシンのヘッドの動きひとつがゲート一個に対応する。量子回路では、ゲートが隣のキュービット以外ともつながるが、それも隣と次々と繋いだものと解することもできる。同等性の厳密な証明は Yao[3] による。

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。図2にあるように、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。制御ビットを明示するために黒丸を打ってある。図の左側から入力され、右側に出力される。制御ビットが $|0>$の時には標的ビットは遷移をおこさないが、制御ビットが $|1>$のときは NOT ゲートとして働く。言い替えると、制御ビット $|a>$と標的ビット $|b>$の入力があれば、 標的ビットに $|a+b \bmod 2>$の出力があるので、確かに XOR の働きをしている。逆があることは明らかであろう。

2.1 エンタングルド状態

以上は、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できる。量子計算の特徴は、前にも述べたように、重ね合わせ状態を許すことである。量子ゲートの出力は入力に対して線形である。

例えば制御ビットに重ね合わせ状態 $|0> + |1>$を選ぶと、標的ビットが

[1] 二つの入力のうち一方だけが YES のとき、YES と出力し、他の場合は NO と出力する。

[2] AND OR はそもそも入力ビットが2個で出力ビットが1個だから可逆でないのは明らか。ただし、可逆な古典計算機というものもあり得るが、別のゲートを用いる。[5] そのために、万能の量子テューリングマシンを作るには新たなゲートを必要とする。

$|1>$のとき入力状態は直積$(|0>+|1>)|1>=|0>|1>+|1>|1>$である。制御NOTは2つの項を各々$|0>|1>\rightarrow|0>|1>$、$|1>|1>\rightarrow|1>|0>$と遷移させるので、出力としてエンタングルド状態$|0>|1>+|1>|0>$を作ることができる。これは、直積で表せない状態である。制御ビットが$|0>$なら標的ビットは必ず$|1>$の状態にあり、逆に制御ビットが$|1>$なら標的ビットは必ず$|0>$の状態にある。すなわち、一方の状態を観測すれば他方が一意的に決まっている。この事情は、有名なアインシュタイン、ポドルスキー、ローゼンのパラドックスと全く同じである。

エンタングルド状態は、「場合分け」に用いることができる。例えば、遷移、$|1>|0>|0>\rightarrow|1>|1>|0>+|1>|0>|1>$ははじめ1番目のスロットにあったボールが、次には2番目かあるいは3番目のスロットに行くという可能性を重ね合わせ状態で表現している。

量子計算のプログラムはある意味で、簡単に用意できる直積の状態から望ましいエンタングルド状態を作ることである。そのエンタングルド状態で、ある量を観測して波束の収縮を引き起こし、別の知りたい量をほとんど確率1で得てしまう。

実際、エンタングルド状態は量子暗号のエッセンスであり、量子誤り訂正の基本的な原理になっている。[4]

2.2 万能量子テューリングマシン

はじめに述べたように、量子計算はユニタリー変換と観測からなりたっている。与えられたキュービットに対して、任意のユニタリー変換が実行できるゲートを持つテューリングマシンを万能量子テューリングマシンと呼ぶことにしよう。前に述べたように、D.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と呼ばれる基本的なゲートと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[7]。理由を簡単に説明しよう。まず、任意の2キュービットのユニタリー変換ができれば、それを組み合わせて、一般のNキュービットのユニタリー変換ができることは簡単な線形代数からわかるだろう。

文献7の9人組の論文によれば、任意の$SU(2)$行列U（2行2列のユニタリー行列で行列式が1のもの）に対して、制御NOTを2個と1キュービットの3個の$SU(2)$の行列A, B, Cを使えば一般の制御-Uが作れる（図3参照）。制御-Uは制御ビットが$|1>$のときにのみ標的ビットが変換Uを受ける。NOTは$|0>,|1>$の基底ではパウリ行列$X=\begin{pmatrix}0&1\\1&0\end{pmatrix}$で表すことができるので、制御NOTは制御-Uの特別な場合である。

任意の2キュービットのユニタリー変換を制御-Uを組み合わせてつくるに

は、 ある制御-U と、それの制御ビットと標的ビット入れ替えた制御-V、1キュービットのユニタリー変換2個、A,B があれば十分であることも2行2列の行列にたいする初等的な線形代数からわかる。

以上の議論から制御ビットと標的ビットが複数ある時には、制御-U をいろいろなキュービットに対して使うことにより、任意の 2^N 次元ユニタリー変換を構成することができることがわかる。万能量子テューリングマシンの概念構築はできたのである。

この節をまとめると、表示を標準的なものに定めて、いくつものゲートを通して状態の遷移を左から右へつぎつぎに行っていくダイアグラムを書くことが量子プログラムということになる。物理屋の言葉でいえば、S行列そのものを一種のファインマンダイアグラムで表していると言ってよいだろう。

3 量子論理ゲートの実験

他に詳しい解説があるので, ここでは制御 NOT に絞って、それを実験的に作る手順を解説しよう[8]。まず2個のキュービットを、2個の2レベル状態を持つ物理系で実現する。それらを $|0>|0>, |0>|1>, |1>|0>, |1>|1>$ と書こう。(例えば、$|1>|0>$ は制御ビットが左に書いた状態 $|1>$ にあり、標的ビットが右に書いた状態 $|0>$ にあることを表す。) 実はこのほかにもう一つ補助的な状態 $|a>$ が必要であるが、その役割はあとで述べよう。それらのエネルギーレベルは模式的に図4 に与えておく。

例えば、状態がはじめに $|1>|1>$ にあったとして、図5にあるエネルギー差 E にあたる振動数の光を照射すると遷移が起きて $|1>|1>$ と $|1>|0>$ の重ね合わせになり、その割合は時間とともに周期的に変化する。照射する時間をうまく選ぶとちょうど半々にすることができる。

$|1>|0>$ に対しても同様の遷移が起きるので、制御ビットがはじめに $|1>$ にある場合をまとめると、

$$|1>|0> \quad \rightarrow |1>(|0>+|1>)/\sqrt{2} \tag{4}$$

$$|1>|1> \quad \rightarrow |1>(-|0>+|1>)/\sqrt{2} \tag{5}$$

となる。

次に、右辺に含まれる状態のうち $|1>|1>$ の符号だけを変えるために、$|1>|1>$ を補助状態 $|a>$ にいったん遷移させて、またもとに戻す。すなわち、

$$|1>(|0>+|1>)/\sqrt{2} \quad \rightarrow \quad |1>(|0>-|1>)/\sqrt{2} \tag{6}$$

$$|1>(-|0>+|1>)/\sqrt{2} \quad \rightarrow \quad |1>(-|0>-|1>)/\sqrt{2} \tag{7}$$

最後に第一段階の逆を行う。始状態と終状態だけを見れば、

$$|1>|0> \quad \rightarrow -|1>|1> \tag{8}$$

$$|1>|1> \quad \rightarrow -|1>|0> \tag{9}$$

となり、標的ビットの状態はフリップしている。

他方、制御ビットの状態がはじめに$|0>$のときは、第二段階の符号の変化がないので、第三段階ではすっかり元に戻る。すなわち、

$$|0>|0> \quad \rightarrow |0>|0> \tag{10}$$

$$|0>|1> \quad \rightarrow |0>|1> \tag{11}$$

確かに、上記の３段階からなる操作により、制御 NOT が実現している。

実験例としては、イオントラップの方法で捕捉され、レーザーによって冷やされたベリリウムイオンの超微細レベルを標的ビットに使い、捕捉ポテンシャルによる基底状態と第一励起状態を制御ビットに使うものが報告されている [8]。この実験では、１イオンすなわち２キュービットのもっとも基本的な素子を実現しているが、キュービットが２以上の時にはどうするのか興味が起こるだろう。実は彼らの実験はシラクとゾラーによる理論的提案に基づいている [10]。シラクとゾラーは、捕捉された一価のイオンを一次元的に並べ、各々レーザーで制御する量子計算機の雛形を提案している。一次元的に並べられたベリリウムイオンは集団的な振動モードを持つので、それを制御ビットに用いれば、同時にいくつものイオンのスピンをコントロールできる。最近ではシラクとゾラーは捕捉されたイオンを約１０個一次元的に並べることが現在の技術で可能だとしている。

4　量子計算はなぜ速いのか

量子計算が古典計算に比べて圧倒的に速いことを、マトリックス M をベクトル v に演算するという例でより詳しく見てみよう。M を 2^n 行 2^n 列のマトリックスとし、v を 2^n 次元ベクトルとすると、$w = Mv$ の素朴な計算には $(2^n)^2$ 回のかけ算が必要になる。

M と v を n 個のテンソル積に分解しよう。

$$\begin{aligned} M &= S^{(1)} \otimes S^{(2)} \otimes \cdots S^{(n)} \\ v &= v^{(1)} \otimes v^{(2)} \otimes \cdots v^{(n)}. \end{aligned}$$

ここに、S^i は i 番目のビットにのみに働く２行２列のマトリックスである。$w = Mv$ を具体的に書けば、

$$w_{j_1 \ldots j_n} = \sum_{i_1 \ldots i_n} S^{(1)}_{j_1 i_1} \ldots S^{(n)}_{j_n i_n} v_{i_1 \ldots i_n}. \tag{12}$$

となる。$S^{(1)}$ の演算には、$i_2 \ldots i_n$ の可能な場合全てに対して一回ずつ行うので 2^{n-1} 個のかけ算が必要である。したがって、全部で $n2^{n-1}$ 回必要になる。

これは、テンソル積を用いるとそうでない場合と比べて、約2^n倍速くなることを意味しており、古典計算における高速フーリエ変換のポイントになっている。量子計算では、$S^{(1)}$の演算が$i_2 \ldots i_n$に対して並列的に行うことができるので実は一回のステップで実行できる。結局全部でn回程度で$w = Mv$を実行できることになる。

5 因数分解

古典計算機で大きな整数を因数分解しようとすると、大変時間がかかる。このことを逆手に取ったものが素数を鍵に用いる公開鍵方式の暗号システムであることはよく知られている。因数分解のアルゴリズムの詳細 [6] は原論文を見ていただくことにして、ここではアイデアだけ述べよう。

整数Nを因数分解するには、まずその因数を一つ見つけ、Nをそれで割り、あとはこれを繰り返す。因数を見つけるために、Nと互いに素であるようなNより小さい整数xを選んで

$$x^r = 1 \mod N \tag{13}$$

を満たす整数rを捜す。(xがNと互いに素なので、この方程式は解を持つ。)
rが偶数ならば、上の式を少し変形して、

$$(x^{r/2}+1)(x^{r/2}-1) = \text{整数} \times N \tag{14}$$

を得るので、$gcd(x^{r/2}+1, N)$ か $gcd(x^{r/2}-1, N)$ のどちらかが欲しい因数を与える。rが奇数ならば、別のxを選んで偶数が出てくるまで続ければよい。大ざっぱには、確率５０パーセントでrは偶数になるので、この試行はすぐ終わる。

さて、$x^r = 1 \mod N$を量子計算機で解くのである。離散的なフーリエ変換とすでに知られている巾計算 $|a) \to |x^a \bmod N)$ に対するアルゴリズム [9] を用いると、次の重ね合わせ状態を高速で作ることができる。

$$\sum_{a,c=0}^{q-1} exp[\frac{2\pi i}{q}ac]|c)|x^a \bmod N). \tag{15}$$

ここにqは充分大きな２の巾乗にとっておく。二つの状態の量子数を各々測定して、それぞれcとx^kを得たとしよう。その確率は、量子力学の公理により、

$$|\sum_{a=0,\, x^a = x^k \bmod N}^{q-1} exp[\frac{2\pi i}{q}ac]|^2 = |\sum_{b=0}^{(q-k-1)/r} exp[\frac{2\pi i}{q}brc]|^2. \tag{16}$$

で与えられる。ここで、$x^r = 1 \mod N$を思い出して、拘束条件$x^a = x^k \bmod N$はbを整数として$a = br + k$と解けることを用いた。

この和がrcがqの倍数に近いところでのみピークを持つことは容易に納得できるし、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を割り出すことができる。式で書けば、c/q = 整数$\times \frac{1}{r}$. rcがqの倍数に近くないところでは、干渉効果のために確率は小さくなり実際上そのようなrは観測されない。qを大きくとっておくのは、鋭いピークが欲しいからである。

例として、分解すべき数$N = 6$に対して、x $= 5$、$q = 2^7$と選ぶ。cを観測すると、たとえば$c = 2^5 \times 3$を高い確率で得る。つぎに、同じ量子計算を行って、またcを観測すると、今度は$c = 2^5 \times 5$を得るかもしれない。これを何回か繰り返して、「実験データ」を、c/qに対してプロットすれば、図6のようになるだろう。分布の最小の周期が$1/r$なので、$rc = q\times$整数を満足するrをみつけることができる。今の場合は、$r = 4$。従って、$gcd(5^{4/2}+1, 6) = 2$と$gcd(5^{4/2}-1, 6) = 6$を得るので、前者を採用して因数2を発見できたことになる。

ここで、量子計算というよりは数論的な注が必要になる。はじめに、Nと互いに素の数xを選んだり、Nと別の数の最大公約数(gcd)を計算のためにNの因数分解が必要ではないか？、というもっともな疑問が湧くかも知れない。実は、最大公約数を見つけるには因数分解をする必要がなく、ユークリッドの互除法という普通の計算機でできる速いアルゴリズムがある。[3]

上ではc/q = 整数$\times \frac{1}{r}$.を用いて周期$1/r$を求めると簡単に述べたが、これについても数論的な注が必要になる。まず、「整数」なるものが、rと互いに素の数の時のみ正しいrを得ることができる。$c/q < 1$に注意すると、「整数」がrよりも小さいことが判る。rよりも小さい数で、rと互いに素の数の総数を数論ではオイラー関数と呼び、$\phi(r)$と書く。そのような数はけっこうたくさんあって、$e^{-\gamma}r/loglog(r)$より多いことが示されている。[4]いいかえると、$loglog(r)$回程度試行すれば一回ぐらいはrと互いに素の数に行き当たる。もちろん、$r < N$だから$loglog(r)$回以下の試行で正しい答えを得るだろう。

量子計算で因数分解を行う際にかかる時間は、ユニタリー変換のところ、検算のところ、最大公約数を求めるところおよび観測を繰り返すところであるが、いずれも多項式時間で計算実行可能である。

ショアのアルゴリズムにおいて、干渉効果が本質的だというならば、古典光学の干渉実験でできそうな気がするかもしれない。もちろん、できるのだが、両者の根本的な違いは、ビームの本数にある。2^n程度の数の因数分解を行うのに、量子計算ではnキュービットあれば可能であるのに、古典光学

[3] **ユークリッドの互除法**一般に$gcd(a,b), a > b$をみつけるには$a-b$とbを比較して大きい方から小さい方を引く。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、
$$\begin{pmatrix} 20 & -12 \\ 8 & / -4 \\ 4 & / -4 \\ 0 & \end{pmatrix}$$
となり、0の上の数字が答えの4である。

[4] γはオイラーの数

では2^n個程度のビームが必要になる。前者では、量子力学的な並列計算を実行していることからこの違いが生じている。

6 まとめ

量子計算は重ねあわせの原理を用いて、並列的に計算を行い、答えの候補の状態の重ね合わせを生成する。候補をその中から絞るには干渉効果などをもちいて正しくないものをかなり大幅に消しておく。そうして、少数の候補を観測しては検算し、正しいものに出会うまで繰り返す。量子計算機は、組み合わせ論的に複雑ではあるが、検算が容易な問題をほとんど数回の逐次的ステップ数で解く可能性を与える。

少し前まで、量子計算機の問題点として、

（1）コヒーレンスを計算機全体にたいして保持すること、（2）重ね合わせの係数が連続な複素数値をとりうることからくる、アナログ計算機と共通のエラーの累積という困難があるといわれてきたが、最近いろいろなエラーを回復するコードが研究されて[4]、この方面での進歩は著しい。

量子計算機の「実用化」について、今はまだ何か言える段階ではないと思われる。イオントラップを用いたものはあまりにも思考実験に忠実すぎて、将来本格的な素子になるとはとうてい考えられないので、日本の実験家がイオントラップ以外の方法を工夫する余地はまだあるような気がする。量子論理ゲートももっと実用に即したものがあるのかもしれない。ゲームは始まったばかりなのだ。最近のレビューとしては文献[12]がある。

参考文献

[1] 小林考次郎著、"計算の複雑さ"（ソフトウェア講座　３３）昭晃堂（昭和６３年）.

[2] D.Deutsch, *Proc.Roy.Soc.London* **Ser**.A 400　96(1985).

[3] A. Yao, in Proceeding of the 34th Annual Symposium on Foundation of Computer Science,　IEEE Computer Society Press, Los Alamits,CA, pp. 352(1993).

[4] D.P. DiVincenzo and P.W. Shor *Phys.Rev.Lett* **77**3260(1996).

[5] R. Landauer, *IBM J. Res.Dev.* **5** 183(1961); C.H. Bennett, ibid **32** 16(1988); T. Toffoli,*Math.Syst.Theory* **14** 13(1981)

[6] P.W.Shor, in Proceeding of the 35th Annual Symposium on Foundation of Computer Science, IEEE Computer Society Press, Los Alamits,CA, pp.116-123(1994).

[7] D.Deutsch, A.Barenco and A. Eckert *Proc.Roy.Soc.London* **Ser.A** 474969(1995). A.Barenco, C.H. Bennett, R.Cleve, D.P..DiVincenzo, N.Margolus, P.Shor, T. Sleator,J. Smolin,and H. Weinfurter *Phys.Rev.A* **52** 3457 (1995).

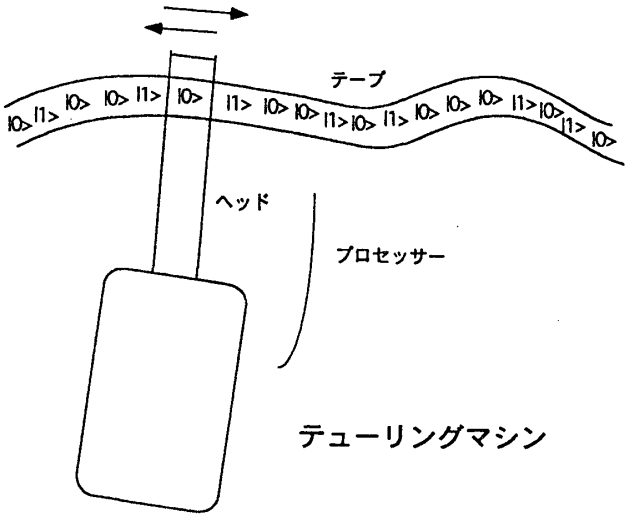
[8] C.Monroe, D.M.Meekhof, B.E.King, W.E.Itano, and D.I. Wineland, *Phys.Rev.Letters* **75** 4714 (1995).

[9] V. Vedral, A. Barenco ,and A. Ekert *Phys.Rev.A* **54**139(1996).

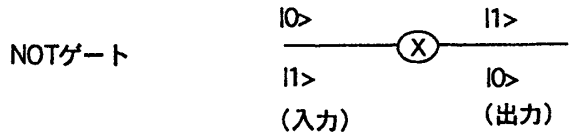
[10] J.I.Cirac and P. Zoller *Phys.Rev.Letters***74** 4091(1995) .

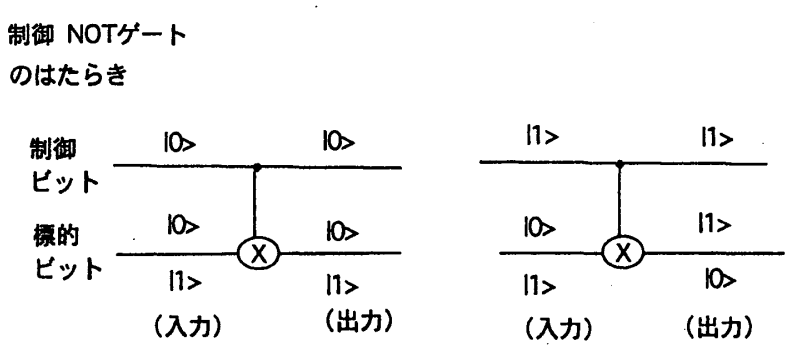
[11] E.L.Lawler et al. ed.,"The Traveling Salesman Problem" Wesley (1983)

[12] A. Ekert amd R. Jozsa *Rev. Mod. Phys.* **68** 733(1996).



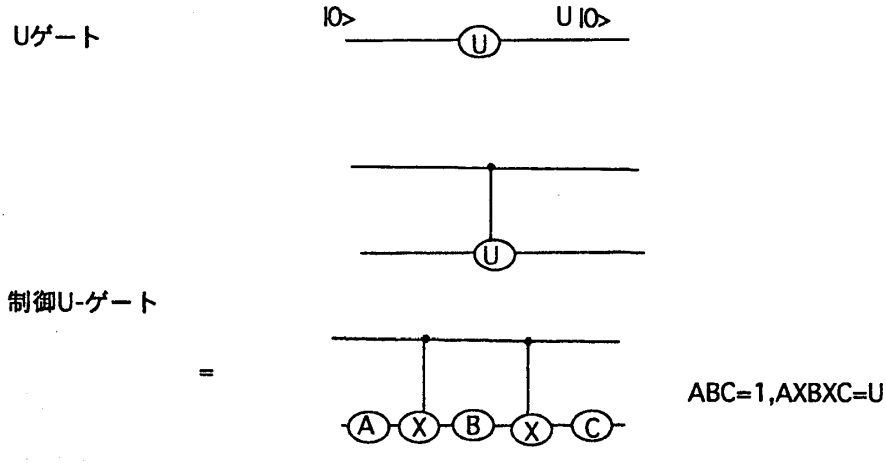
<図1>テューリングマシンは充分長いテープとプロセッサーから成り立っており、テープには $|0>$ と $|1>$ の羅列が書き込まれている。プロセッサーにはヘッダーが付いていてテープに書いてある数字を読んだり書き換えたりしながら、テープを前後に移動する。その動きは、あらかじめプログラムされている。そのプログラムは短いテープに書かれていてプロセッサーに入っているが図では省略している。





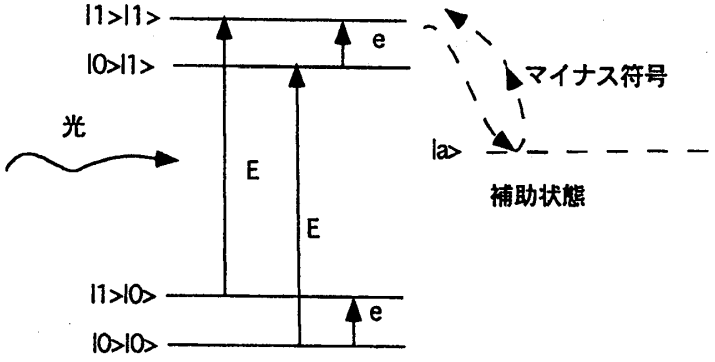
制御NOTゲート

＜図２＞NOT は１キュービットのゲートで、入力 |0 >を出力 |1 >に入力 |1 >を出力 |0 >に遷移をさせる。図では○の中に X を書いてそれを示している。制御 NOT は２キュービットゲートで、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。制御ビットには黒丸を打ってある。制御ビットについては、入力が |0 >か |1 >のいずれかであってそれらの重ねあわせでない限り、入力と出力が一致する。制御ビットの入力が |0 >の時には標的ビットは遷移をおこさないが、それが |1 >のときは NOT ゲートとして働く。



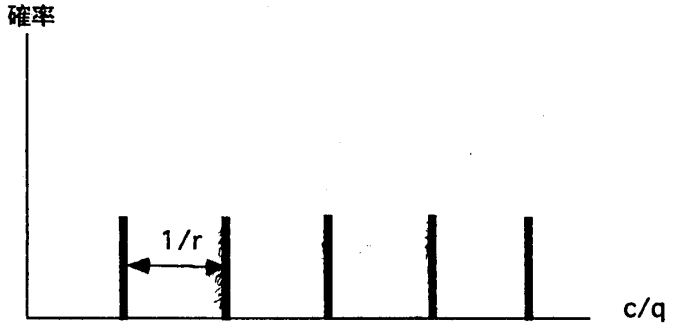
Uゲート

＜図３＞１ビットの U ゲートは入力の状態を U だけユニタリー変換をして出力する。制御 U ゲートは制御ビットの入力が |0 >の時には標的ビットは遷移をおこさないが、それが |1 >のときは U ゲートとして働く。



制御NOTゲートを物理的に実現した例のエネルギーレベルの模式図

＜図４＞制御 NOT のエネルギーレベルの模式図。E,e はレベル間のエネルギー差をあらわす。$|1>|1>$から補助状態$|a>$に遷移させてまた$|1>|1>$に戻すことによって、$-|1>|1>$を実現する物理系がある。



＜図５＞因数分解のアルゴリズムによって得られる「実験データ」。量子計算のあと c を観測するということを何回か繰り返して、「データ」をc/qに対してプロットする。分布の周期の最小値として$1/r$が求まる。