Keywords you should know. 第112回

知っておきたいキーワード

量子コンピューティング

出崎善久†

† 茨城大学 工学部
"Quantum Computing" by Yoshihisa Desaki (Faculty of Engineering, Ibaraki University, Ibaraki)

キーワード：量子コンピューティング，量子ビット，量子回路，重ね合わせ，Deutsch-Jozsaのアルゴリズム

1. まえがき

ムーアの法則が提唱されてからおよそ半世紀が経ち，集積回路の微細化とは異なる方向で計算機の能力向上を目指した研究が近年盛んに行われています．量子力学の考え方を計算に取り入れた量子計算機もその一つです．本稿では，代表的な計算モデルである量子回路に基づく量子コンピューティングについて説明します．

2. 量子ビット

皆さんよくご存じのとおり，古典計算機では情報を2進数で扱います．普段パソコンやスマートフォンを使っている時には2進数を意識することはありませんが，ディスプレイに表示されている文字も，計算機の内部では2進数として扱われています．古典計算機では，2進数の1桁をビット（bit）と呼んで計算の最小単位としています．これに対して，量子計算で計算の最小単位となるのは，量子ビット（quantum bit，またはqubit）と呼ばれるものです．1ビットを0，1で表すのと同様に，1量子ビットを$|0\rangle$，$|1\rangle$と表します（Diracの記法と呼ばれています）．

量子計算機と古典計算機

古典計算機とは，量子計算機と対比して通常の計算機のことを指す用語で，量子コンピューティングの分野で広く使用されている用語です（言葉の頭に「古典」がつくと，量子力学的効果を考えないという意味になります）．

古典計算のビットは，物理的には電圧の値の高低等と結びつけて解釈されます．量子ビットの場合も同様に，物理系の量子力学的な状態を表したものであると解釈します．量子ビットは，以下のようにベクトルで表現することができます．

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

次に，以下のような線形結合を考えてみましょう．

$$\psi=c_0|0\rangle+c_1|1\rangle=\begin{pmatrix}c_0\\c_1\end{pmatrix}$$

これもまた，1量子ビットが取りうる値となります．上の式中のc_0，c_1は確率振幅（probability amplitude）と呼ばれる複素数で，以下の式を満たすものとします．

$$|c_0|^2+|c_1|^2=1$$

これは，$|0\rangle$，$|1\rangle$に対応する状態を，係数c_0，c_1で重ね合わせた状態であると解釈できます．古典ビットは0と1の2通りの状態しか表現できません．これに対して，係数c_0，c_1が（2乗和が1であれば）任意の複素数でよいことから，1量子ビットが表現できる値は無限個あることがわかります．ただし，量子ビットを実現した物理系を ☞

映像情報メディア学会誌 Vol. 70，No. 4，pp. 632～636（2016）

☞ 実際に測定すると，$|0\rangle$または$|1\rangle$のどちらか一つが得られ，各ビットが得られる確率は，$|c_0|^2$, $|c_1|^2$となります．

1量子ビットは，それに対応する物理系の量子力学的な状態を表すものでした．2量子ビット（量子ビット2桁）では二つの物理系の状態を表すことになるので， 以下の4通りの状態を考える必要があります．

$|0\rangle$と$|0\rangle$, $|0\rangle$と$|1\rangle$, $|1\rangle$と$|0\rangle$, $|1\rangle$と$|1\rangle$

これは数学的には直積なので，以下のように表すことができます．

$|0\rangle\otimes|0\rangle, |0\rangle\otimes|1\rangle, |1\rangle\otimes|0\rangle, |1\rangle\otimes|1\rangle$

量子ビットの数を3以上に増やした場合も同様に，以下のように書くことができます．

$|0\rangle\otimes|0\rangle\otimes|0\rangle, |0\rangle\otimes|0\rangle\otimes|1\rangle, \cdots, |1\rangle\otimes|1\rangle\otimes|1\rangle$

ただし，この書き方だと見た目が煩雑になるので，通常は以下の省略記法を使用します．

$$\overbrace{|000\cdots0\rangle}^{n\text{桁の量子ビット}} = \overbrace{|0\rangle\otimes|0\rangle\otimes\cdots\otimes|0\rangle}^{n\text{個の}|0\rangle\text{の直積}}$$

$|00\cdots0\rangle$を$|0\rangle|0\rangle\cdots|0\rangle$と書くこともあります．1量子ビットが変数や式で表現されている場合は，$|x, x\oplus y\rangle$のように書きます．

$|00\cdots0\rangle$, $|00\cdots1\rangle$, $|11\cdots1\rangle$は，n量子ビットの計算基底（computational basis）と呼ばれるものです．1量子ビットの場合と同様に，n量子ビットもまた計算基底の線形結合で表すことができ，確率振幅の2乗和は1となります．

3. 量子回路

ここでは，量子ビットに対する計算のモデルとして広く知られている量子回路についてごく簡単に紹介します．

3.1 Hadamardゲート

古典ビットに対する操作は，ANDやOR等の論理演算です．計算機として実装する際には，ANDゲートやORゲートといった論理ゲートを電子回路として実現します．量子回路モデルにおいて論理ゲートに相当するものは，量子ゲートと呼ばれています．まずは，1量子ビットに対する代表的な量子ゲートであるHadamardゲートを取り上げてみましょう．

Hadamardゲート（Hadamard gate）

1量子ビットに対してHadamard変換Hを施すゲートです．$\psi = c_0|0\rangle + c_1|1\rangle$に$H$を作用させて$\psi' = c'_0|0\rangle + c'_1|1\rangle$が得られます．

$$H\begin{pmatrix}c_0\\c_1\end{pmatrix} = \begin{pmatrix}c'_0\\c'_1\end{pmatrix},\ H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}$$

論理ゲートを論理関数で表現したのと同様に，量子ゲートは行列で表現できます．回路図では，**図1**のように四角い箱の中にゲートの種類を表す情報を書いて表現します．箱に接続されている線は量子ビットを表し，古典計算機の論理ゲートと同様に，ゲートの左側の線を入力，右側の線を出力と解釈します．

ここで，Hadamardゲートの性質について少し考えてみましょう．$|0\rangle$にHadamardゲートを作用させると，以下のようになります．

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$$
$$= \frac{1}{\sqrt{2}}\left\{\begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix}\right\} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$|1\rangle$にHadamardゲートを作用させた場合も同様に，

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

以上の結果をまとめて，任意の1量子ビット$|x\rangle$にHadamardゲートを作用させて得られる式を与えておきます．

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

n量子ビット$|x\rangle = |x_1 x_2 \cdots x_n\rangle$に対してHadamardゲートを並列に作用させると，以下のようになります[1) 2)]．

$$H^{\otimes n}|x\rangle = H^{\otimes n}|x_1 x_2 \cdots x_n\rangle$$
$$= H|x_1\rangle\otimes H|x_2\rangle\otimes\cdots\otimes H|x_n\rangle$$
$$= \sum_{z_1\in\{0,1\}} \frac{(-1)^{x_1 z_1}}{\sqrt{2}}|z_1\rangle\otimes \sum_{z_2\in\{0,1\}} \frac{(-1)^{x_2 z_2}}{\sqrt{2}}|z_2\rangle\otimes\cdots\otimes$$
$$\sum_{z_n\in\{0,1\}} \frac{(-1)^{x_n z_n}}{\sqrt{2}}|z_n\rangle$$
$$= \sum_{z_1\in\{0,1\}}\sum_{z_2\in\{0,1\}}\cdots\sum_{z_n\in\{0,1\}} \frac{(-1)^{x_1 z_1 + x_2 z_2 + \cdots + x_n z_n}}{\sqrt{2^n}}|z_1 z_2 \cdots z_n\rangle$$
$$= \sum_{z\in\{0,1\}^n} \frac{(-1)^{xz}}{\sqrt{2^n}}|z\rangle$$

上の式において，$|z\rangle = |z_1 z_2 \cdots z_n\rangle$であり， ☞

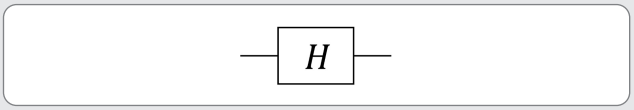

図1 Hadamardゲート

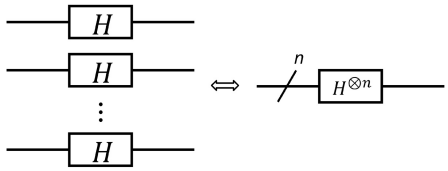

図2　Hadamardゲートの並列作用

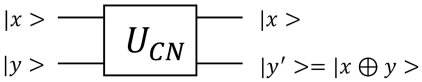

図3　制御NOTゲート

☞　$xz = x_1z_1 + x_2z_2 + \cdots + x_nz_n$としました．また，$n$量子ビットに対するHadamardゲートの並列作用を$H^{\otimes n}$で表すことにします（**図2**）．

3.2　制御NOTゲート

2量子ビットに対する代表的な量子ゲートとして，制御NOTゲート（**図3**）を紹介しておきます．

制御NOTゲート（Controlled NOT gate），あるいはCNOTゲート

制御NOTゲートは，以下の写像で定義された2量子ビットに対して作用するゲートであり，古典計算機のXORに相当します．制御ビット$|x\rangle$はそのまま出力され，標的ビット$|y\rangle$は$|y'\rangle = |x \oplus y\rangle$（⊕は排他的論理和）となって出力されます．

$$U_{CN} : |x,y\rangle \rightarrow |x, x \oplus y\rangle$$

複数の入出力線をもつ量子ゲートでは，上の入力を上位ビット，下の入力を下位ビットであると解釈します．2量子ビットの計算基底に制御NOTゲートを作用させると，以下のように制御ビット（上位ビット）の値により標的ビット（下位ビット）の値が変化します．

$$U_{CN}|00\rangle = |00\rangle,\ U_{CN}|01\rangle = |01\rangle,$$
$$U_{CN}|10\rangle = |11\rangle,\ U_{CN}|11\rangle = |10\rangle$$

ここで，重ね合わせた状態を含んだ2量子ビットψに制御NOTゲートを作用させると何が起こるでしょうか．例えば，以下のようなψを考えてみます．

$$\psi = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |0\rangle = \frac{|00\rangle + |10\rangle}{\sqrt{2}}$$

これに制御NOTゲートを作用させると，

$$\beta_{00} = U_{CN}\psi = U_{CN}\frac{|00\rangle + |10\rangle}{\sqrt{2}}$$
$$= U_{CN}\frac{|00\rangle}{\sqrt{2}} + U_{CN}\frac{|10\rangle}{\sqrt{2}}$$
$$= \frac{|00\rangle}{\sqrt{2}} + \frac{|11\rangle}{\sqrt{2}} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

こうして得られたβ_{00}はBell状態（Bell state）と呼ばれ，1量子ビットの直積では表現することができません（$\beta_{00} = (c_{00}|0\rangle + c_{01}|1\rangle) \otimes (c_{10}|0\rangle + c_{11}|1\rangle)$を満たす確率振幅の組$c_{00}$，$c_{01}$，$c_{10}$，$c_{11}$が存在しません）．また，1ビット目が0であれば2ビット目も0，というようにビット間に相関が生じています．これを，量子もつれ（entanglement）と呼び，古典計算では生じない量子計算固有の概念となっています．

量子ゲートを表す図では，古典計算の論理ゲートにならって入出力のビット数に応じて独立に線を書いていますが，このようにもつれた状態の入出力となる場合もあるため注意が必要です．

3.3　量子並列化回路

ここで，以下のような写像で定義された$n+1$量子ビットに対する量子ゲートU_fについて考えてみましょう．

$$U_f : |x,y\rangle \rightarrow |x, y \oplus f(x)\rangle$$
（$f(x)$はnビット入力1ビット出力の関数）

複数の量子ゲートを接続して得られる量子回路の例を**図4**に示します．この場合，回路図の左から右に向かって時間が流れると解釈し，一番左の量子ビット（回路の入力）にいくつかの量子ゲートを作用させて，計算結果となる量子ビット（回路の出力）が得られます．この量子回路に$\psi_0 = |00\rangle$を入力した場合，どのような出力ψ_2が得られるでしょうか．

最初に$|00\rangle$の上位ビット（左側のビット）にHadamardゲートを作用させて，以下の状態ψ_1が得られます．

$$\psi_1 = H|0\rangle \otimes |0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |0\rangle = \frac{|00\rangle + |10\rangle}{\sqrt{2}}$$

これにU_fを作用させると，

$$\psi_2 = U_f\psi_1 = U_f\frac{|00\rangle + |10\rangle}{\sqrt{2}}$$
$$= U_f\frac{|00\rangle}{\sqrt{2}} + U_f\frac{|10\rangle}{\sqrt{2}} = \frac{|0, 0 \oplus f(0)\rangle}{\sqrt{2}} + \frac{|1, 0 \oplus f(1)\rangle}{\sqrt{2}}$$
$$= \frac{|0, f(0)\rangle}{\sqrt{2}} + \frac{|1, f(1)\rangle}{\sqrt{2}} = \frac{|0, f(0)\rangle + |1, f(1)\rangle}{\sqrt{2}}$$

得られたψ_2は$|0,f(0)\rangle$と$|1,f(1)\rangle$の重ね合わせになっており，関数$f(x)$の任意の入力に対する結果を並列に求めていることになります．古典計算機で関数の並列処理を行う場合，入力の数だけプロセッサを並べる必要があるのに対し，量子計算機では量子ビットの性質を利用して並列処理を実現しているのです．

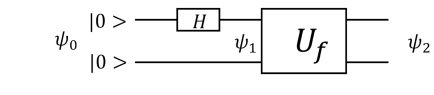

図4　量子並列化回路

4. Deutsch-Jozsaのアルゴリズム

量子計算機が古典計算機に対して明確に優位であることを示したアルゴリズムの例として，Deutsch-Jozsaのアルゴリズムについて説明します．まず，以下のような関数の推測問題について考えてみましょう．

関数の推測問題

nビット入力，1ビット出力の関数$f(x)$は，定数型か分布型であるとする．$f(x)$がどちらの型であるかを特定するアルゴリズムを示せ．

定数型：すべてのxに対して，$f(x)=0$，または$f(x)=1$．

分布型：ちょうど半分のxに対して$f(x)=0$，残り半分のxに対して$f(x)=1$（xの分布は任意）．

古典計算機で上の問題を解く場合のアルゴリズムの擬似コードを以下に示します．

```
x = f (0);
for (i = 0; i < 2^ {n -1}; i ++) {
  y = f (i +1);
  if (y ! = x) {
    break ;
  }
  x = y;
}
if (i == 2^ {n-1}) {
  println ("定数型") ;
} else {
  println ("分布型") ;
}
```

容易にわかるように，2^{n-1}個目の入力xまで$f(x)$の値がすべて同じ場合，$2^{n-1}+1$個目の入力xに対する結果を求めるまでは，$f(x)$がどちらの型かを特定できません．この場合，$f(x)$の計算を$2^{n-1}+1$回行う必要が生じます．したがって，関数の推測問題を古典計算機で解く場合の計算量は$O(2^n)$となります．一般に，計算量が指数関数で表された問題は，計算量が多いという意味で古典計算機では解くのが困難な問題であるとみなされます．

ランダウの記号

計算機科学の分野では，nビットで表現されたアルゴリズムの入力に対して，計算量をnの関数$f(n)$で表します．計算量は$n\to\infty$とした場合の漸近的な挙動で評価するため，$f(n)$の係数等は通常は無視されます．例えば，$f(n)=an^2+bn+c$の場合は$O(n^2)$と表記し，n^2の係数aや2次の項よりも小さい項$bn+c$については考慮しません．

古典計算機で解くのが困難な問題を，量子計算機ではどのようにして少ない計算量で解くのでしょうか．関数の推測問題に対するアルゴリズムを実現する量子回路を**図5**に示します．

アルゴリズムの処理の経過を量子回路の途中の状態を調べることで追ってみましょう（詳細は文献[1)2)]参照）．入力を$\psi_0=|00\cdots0\rangle\otimes|1\rangle$としたとき，$\psi_1$は以下のようになります．

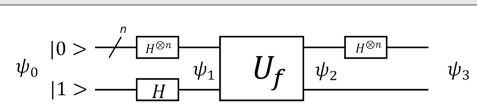

図5　Deutsch-Jozsaのアルゴリズムを実現する量子回路

$$\psi_1=H^{\otimes n}|00\cdots0\rangle\otimes H|1\rangle=\sum_{x\in\{0,1\}^n}\frac{|x\rangle}{\sqrt{2^n}}\otimes\left\{\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right\}$$

ψ_1にU_fを作用させると，以下のようになります．

$$\psi_2=U_f\psi_1=\sum_{x\in\{0,1\}^n}\frac{(-1)^{f(x)}|x\rangle}{\sqrt{2^n}}\otimes\left\{\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right\}$$

ψ_2の上位nビットにHadamardゲートを作用させると，

$$\psi_3=\sum_{z\in\{0,1\}^n}\sum_{x\in\{0,1\}^n}\frac{(-1)^{xz+f(x)}|z\rangle}{2^n}\otimes\left\{\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right\}$$

ここで，ψ_3の上位nビットが$|00\cdots0\rangle$（$z=00\cdots0$の場合）となっている項の係数について考えてみましょう．係数を表す式に$z=00\cdots0$を代入すると，

$$\sum_{x\in\{0,1\}^n}\frac{(-1)^{xz+f(x)}}{2^n}\Bigg|_{z=00\cdots0}=\sum_{x\in\{0,1\}^n}\frac{(-1)^{f(x)}}{2^n}$$

となり，具体的に係数の値を求めることができる式になりました．すなわち，

・$f(x)$が定数型の場合，係数の値は±1

・$f(x)$が分布型の場合，係数の値は0

となります．確率振幅に対する制約から，係数が±1の項があるということは，それ以外の項の係数が0になることを意味します．したがって，$f(x)$が定数型の場合は，上位nビットを測定すると必ず$|00\cdots0\rangle$が得られます．一方，$f(x)$が分布型の場合の$|00\cdots0\rangle$の係数は0なので，$|00\cdots0\rangle$を測定して得る確率は0となります．

以上をまとめると以下のようになります．

Deutsch-Jozsaのアルゴリズム

(1) 図5の量子回路に$|00\cdots0\rangle|1\rangle$を入力する．

(2) 出力された結果の上位nビットを測定する．

(3) 測定結果が$|00\cdots0\rangle$であれば定数型，そうでなければ分布型であると判断する．

古典計算機では関数$f(x)$を$2^{n-1}+1$回呼び出す必要がありましたが，量子計算機では量子ゲートU_fを1回作用させることで問題を解くことができます．関数の推測問題自体は実用的なものではありませんが，Deutsch-Jozsaのアルゴリズムは量子計算の優位性を初めて示したものであるという点で重要です（実用的なアルゴリズムとしては，Shorのアルゴリズムや Groverのアルゴリズム等が知られています）．

5. むすび

本稿では，量子回路に基づく量子コンピューティングの基礎について説明しました．量子コンピューティングを実現する計算モデルは，これまでに多数提案されています．その一つである量子アニーリングや量子断熱計算では，時間に伴う変化を表す方程式に従って物理系を変化させ，最適化問題の解を得る計算モデルです．広く知られている焼きなまし法の解を得る過程に，量子力学の考え方を持ち込んだのが特徴です[3]．

（2016年4月15日受付）

参 考 文 献

1) 木村達也訳：“量子コンピュータと量子通信Ⅰ－量子力学とコンピュータ科学－”，オーム社（2004）
2) 宮野健次郎，古澤明：“量子コンピュータ入門第2版”，日本評論社（2016）
3) 小林 聡ほか編：“ナチュラルコンピューティング・シリーズ第0巻－自然計算へのいざない－”，近代科学社（2015）

出崎 善久（でさき よしひさ） 1996年，大阪大学大学院基礎工学研究科物理系専攻博士後期課程退学．同年，東京都立大学工学部助手．2000年より，茨城大学に勤務．画像処理を対象としたハードウェアアルゴリズム，専用プロセッサの研究等に従事．

キーワード募集中

この企画で解説して欲しいキーワードを会員の皆様から募集します．ホームページ（http://www.ite.or.jp）の会員の声より入力可能です．また電子メール（ite@ite.or.jp），FAX（03-3432-4675）等でも受け付けますので，是非，編集部までお寄せください．（編集委員会）