量子科学入門ゼミ　第 1 章から第 3 章まで

東工大物理学系 B4 松本侑真

2023 年 10 月 4 日

概要

量子科学入門の第 1 章から第 3 章の終わりまでについて、本の行間などをまとめた。第 1 章では量子ビットを記述するための簡単な線形代数の基礎と、Dirac のブラケット記法の準備を行う。第 2 章では古典回路モデルと量子回路モデルの基礎を扱い、特に量子回路がユニタリ演算子によって構成されることを見る。第 3 章では量子アルゴリズムの代表例として、

- Deutsch-Jozsa のアルゴリズム（定数関数/バランス関数判定問題）
- Glover のアルゴリズム（探索問題）
- Shor のアルゴリズム（素因数分解問題）

について扱う。（Glover と Shor は本資料では紹介しない。）また、第 0 章のお話はまとめていない。

目次

1　第 1 章

1.1　Dirac の表記法

Dirac の表記法とは、ベクトルや行列を簡潔に表すことのできる表記法である。LaTeX では、braket パッケージや physics パッケージを用いると簡単に書くことができる。自分は physics パッケージを用いている。縦ベクトルをブラ、横ベクトルをケットで表すことで、行列やベクトルの演算を直感的に操作できるようになる。量子情報で良く出てくる記号として、

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix} \in \mathbb{C}^2 \tag{1.1}$$

の計算基底（z 基底）がある。計算 “基底” と言っているのは、任意の $\mathbb{C}^2$ のベクトルは、$a, b \in \mathbb{C}$ として、

$$\begin{pmatrix}a\\b\end{pmatrix} = a\,|0\rangle + b\,|1\rangle \tag{1.2}$$

と展開できるためである。すなわち、1 つの量子ビットの状態は、計算基底の線形結合によって表される。そして、計算基底は正規直交基底となっている。$i, j \in \{0, 1\}$ として

$$\langle i|j\rangle = \delta_{ij} \tag{1.3}$$

が成立する。ここで出てくる記号

$$\langle\cdot|\cdot\rangle \tag{1.4}$$

は 2 つのベクトルの内積を表す。例えば、

$$\langle 0|1\rangle = \begin{pmatrix}1\\0\end{pmatrix}\cdot\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}1 & 0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = 0 \tag{1.5}$$

である。すなわち、ブラベクトルとは、ケットベクトルの共役転置（Hermite 共役）である：

$$\langle x| = |x\rangle^\dagger \ 。 \tag{1.6}$$

他にも、Hadamard 基底（x 基底）

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{1.7}$$

や、円基底（y 基底）

$$|i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle), \quad |i-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i\,|1\rangle) \tag{1.8}$$

などが良く出てくる。これらは、計算基底に特定のユニタリ演算子を作用させることで変換することができる。

ブラとケットをその順番に並べたものは内積を意味するのであった。ケットとブラの順番に並べたものは行列を意味する。例えば、

$$|0\rangle\langle 1| = \begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}0 & 1\end{pmatrix} = \begin{pmatrix}0 & 1\\0 & 0\end{pmatrix} \tag{1.9}$$

である。この演算（ケットブラ）は慣れるまで分かりづらいかもしれないが、行列とは最終的になんらかのベクトル $|\psi\rangle = (a \quad b)^\top \in \mathbb{C}^2$ に作用するものなので、

$$(|0\rangle\langle 1|)\,|\psi\rangle = |0\rangle\,(\langle 1|\psi\rangle) = \langle 1|\psi\rangle\,|0\rangle \tag{1.10}$$

と理解する方が簡単である。この計算を行列の成分表示に翻訳すると、

$$\begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}0 & 1\end{pmatrix}\begin{pmatrix}a\\b\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}(0\times a + 1\times b) = b\begin{pmatrix}1\\0\end{pmatrix} \tag{1.11}$$

となる。行列の成分表示での計算は面倒であるが、ブラケット記法に慣れてくると一瞬で計算することができる。

$$|\psi\rangle = \begin{pmatrix} a \\ b \end{pmatrix} = a\,|0\rangle + b\,|1\rangle \tag{1.12}$$

であるため、

$$|0\rangle\langle 1|\psi\rangle = |0\rangle\langle 1|\,(a\,|0\rangle + b\,|1\rangle) = b\,|0\rangle \tag{1.13}$$

といった要領である。ここで、$|0\rangle, |1\rangle$ の正規直交性を頭の中で用いた。ブラケット記法のまま計算できないと、量子計算の意味を捉えにくいため、早めにこの記法に慣れておこう。大事なことは、ブラケット演算はスカラーとなり、ケットブラ演算は行列になるということである。また、

$$\sum_{i=0}^{1} |i\rangle\langle i| = I\ (\text{単位行列}) \tag{1.14}$$

となることも重要な性質であり、式変形で良く出てくる。これは、$|0\rangle, |1\rangle$ が基底であることから得られる性質である。

1.2 量子ビット系

量子ビット系とは、2 つの基底 $|\phi_0\rangle, |\phi_1\rangle$ の線形結合により表される状態からなる系のことである。状態 $|\psi\rangle$ を基底 $\{|\phi_0\rangle, |\phi_1\rangle\}$ の元で測定すると、$|\phi_0\rangle$ もしくは $|\phi_1\rangle$ の状態が得られる。その確率はそれぞれ

$$P_0 \propto |\langle\phi_0|\psi\rangle|^2, \quad P_1 \propto |\langle\phi_1|\psi\rangle|^2 \tag{1.15}$$

である。

1.2.1 状態の規格化とグローバル位相について

α を 0 でない任意の複素数として、状態 $|\psi\rangle$ と状態 $\alpha\,|\psi\rangle$ は同じ状態とみなす。そのため、$\langle\psi|\psi\rangle = 1$ となるように規格化をしているものとする。一般の量子力学の問題では規格化を考えないことも考えることもあるが、量子コンピュータの文脈では必ず規格化していることに注意。規格化条件を課してもなお状態に不定性が残る。なぜなら、状態 $|\psi\rangle$ に絶対値 1 の複素数 β をかけてもノルムは 1 だからである。また、絶対値 1 の複素数 β は、実数 ϕ を用いて

$$\beta = e^{i\phi} = \cos\phi + i\sin\phi \tag{1.16}$$

と表すことができる。この実数を位相 (phase) と呼ぶ。それぞれの基底の観測確率は変わらないため、位相が状態にかかったものは元と同じ状態とみなすことができる。このような不定性を、グローバル位相による不定性と呼び、普通は $\phi = 0$ として考える。しかし、“グローバル” でない位相のかかり方をしているものは異なることに注意。すなわち、

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle \tag{1.17}$$

と、

$$|\psi'\rangle = a\,|0\rangle + e^{i\phi}b\,|1\rangle \tag{1.18}$$

は異なる状態である。確かに $|0\rangle$ と $|1\rangle$ を観測する確率は変化してないが、異なる基底に変換した際に、その基底で観測する確率は $|\psi\rangle$ と $|\psi'\rangle$ で異なる。これは、密度行列が異なるため（$|\psi\rangle\langle\psi| \neq |\psi'\rangle\langle\psi'|$）とも理解できる。

1.2.2 ブロッホ球について

別プリントを参照

1.3　基本的な量子回路と、状態の時間発展

量子力学では、状態ベクトル $|\psi\rangle$ で記述される系の時間発展は Schrödinger 方程式で表される：

$$i\hbar\frac{\mathrm{d}}{\mathrm{d}t}|\psi(t)\rangle = \hat{H}|\psi(t)\rangle \ 。 \tag{1.19}$$

ここで、$\hat{H}$ は量子系の Hamilton 演算子（古典的にはエネルギーに対応するもの）である。したがって、初期条件を $t=0$ で $|\psi(t)\rangle = |\psi(0)\rangle$ と置くと、一般解は

$$|\psi(t)\rangle = e^{-i\hat{H}t/\hbar}|\psi(0)\rangle \tag{1.20}$$

と求まる。ここで、指数関数の肩に載っているのは演算子（例えば $\hat{H} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}$ とか）であるため、通常の意味での指数関数ではないことに注意。[*1] この演算子を時間発展演算子と呼び、ユニタリ演算子である：

$$UU^\dagger = U^\dagger U = I \quad (U = e^{-i\hat{H}t/\hbar}) \ 。 \tag{1.22}$$

ユニタリ演算子によって時間発展が行われるということは、内積が不変であるということである。物理的には、確率が保存していることに対応している。また、$t=0$ における $\hat{H}$ の固有値問題

$$\hat{H}|\phi\rangle = E|\phi\rangle \tag{1.23}$$

が解けたとして、固有値と固有ベクトルの組として

$$(E_0, |\phi_0\rangle), (E_1, |\phi_1\rangle), \ldots, (E_n, |\phi_n\rangle) \tag{1.24}$$

が得られたとしよう。このとき、固有ベクトルの組は完全性を為しているものとする。すると、任意の状態は固有ベクトルと複素数 C_i を用いて

$$|\psi(0)\rangle = \sum_{i=1}^{n} C_i|\phi_i\rangle \tag{1.25}$$

と展開できる。また、固有ベクトルに対する時間発展は

$$|\phi_i(t)\rangle = e^{-i\hat{H}t/\hbar}|\phi_i(0)\rangle = e^{-iE_it/\hbar}|\phi_i(0)\rangle \tag{1.26}$$

と計算することができる。つまり、右辺の指数関数がスカラーとなることが嬉しい。これを用いると、一般の状態の時間発展は、

$$|\psi(t)\rangle = \sum_{i=1}^{n} C_i e^{-iE_it/\hbar}|\phi_i(0)\rangle \tag{1.27}$$

と求めることができる。時刻 $t=0$ における固有値問題さえ解いておけば、状態の時間発展を計算することができる。この考え方は院試でも頻出なので良く理解しておこう。量子コンピュータの文脈でもこの考え方を用いる。量子ビット系で実現される時間変化は、とあるユニタリ演算子 $\hat{U}$ によって記述される：

$$|\psi\rangle' = \hat{U}|\psi\rangle \ 。 \tag{1.28}$$

このような時間変化を、状態のユニタリ発展と呼ぶ。1 つの量子ビット系は普通 $\mathbb{C}^2$ のベクトルで記述されるため、このユニタリ演算子は 2×2 の複素行列で表される。

[*1] 一般に、演算子 $\hat{A}$ が肩に載った指数関数演算子 $e^{\hat{A}}$ は、

$$e^{\hat{A}} = \sum_{n=0}^{\infty}\frac{1}{n!}\hat{A}^n \tag{1.21}$$

で定義される。そのため、2 つの演算子 $\hat{A}$, $\hat{B}$ の交換関係が 0 でない場合 ($[\hat{A},\hat{B}] \neq 0$) は、指数関数演算子も非可換となる。

1.3.1 良く出てくるユニタリ行列

- 単位行列と Pauli 行列 σ_x, σ_y, σ_z

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{1.29}$$

  - Pauli 行列は、$\sigma_x = \sigma_1$, $\sigma_y = \sigma_2$, $\sigma_z = \sigma_3$ とも表される。2 次の単位行列を σ_0 とすることもある。
  - Pauli 行列はエルミートかつユニタリであり、トレースレスで反交換関係が 0 になり、2 乗すると単位行列になるような 2×2 行列として特徴づけられる。

$$\operatorname{Tr} \sigma_i = 0 \quad (i = 1, 2, 3) \tag{1.30}$$

$$\{\sigma_1, \sigma_2\} = \{\sigma_2, \sigma_3\} = \{\sigma_3, \sigma_1\} = 0 \tag{1.31}$$

  - 良く使う関係式

$$\sigma_i \sigma_j = i\varepsilon_{ijk}\sigma_k + \delta_{ij} I \tag{1.32}$$

$$(\boldsymbol{\sigma} \cdot \boldsymbol{a})(\boldsymbol{\sigma} \cdot \boldsymbol{b}) = (\boldsymbol{a} \cdot \boldsymbol{b})\sigma_0 + i(\boldsymbol{a} \times \boldsymbol{b}) \cdot \boldsymbol{\sigma} \tag{1.33}$$

Pauli 行列は量子ビットの回転にも用いられる。$\boldsymbol{n} = (n_x, n_y, n_z)$ をノルム 1 の方向ベクトルとすると、以下の行列指数関数

$$e^{-i(\boldsymbol{n}\cdot\boldsymbol{\sigma})\theta} = \sum_{n=0}^{\infty} \frac{(-i)^n}{n!} (\boldsymbol{n} \cdot \boldsymbol{\sigma})^n \theta^n = \cos\theta - i \sin\theta (\boldsymbol{n} \cdot \boldsymbol{\sigma}) \tag{1.34}$$

が量子ビットの回転を司る演算子となる。“量子ビットの回転” とは、量子ビットの Bloch 球表現において、Bloch 球上の対応する座標を指すベクトルが回転しているという意味である。すなわち、Bloch 球上の極座標表示 $(\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta)$ に対応する状態ベクトルは

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle \tag{1.35}$$

であった。ここで、$R_{\boldsymbol{n}}(\theta) \coloneqq e^{-i(\boldsymbol{n}\cdot\boldsymbol{\sigma})\theta/2}$ と定義すると、

$$R_{\boldsymbol{n}}(\alpha)|\psi\rangle \tag{1.36}$$

という変換は、Bloch 球上のベクトルを方向ベクトル $\boldsymbol{n}$ の周りに α 回転させる変換に対応している。例えば、$\boldsymbol{n} = (0, 0, 1)$ とすると、

$$R_{\boldsymbol{n}}(\alpha)|\psi\rangle = \left(\cos\frac{\alpha}{2} - i\sin\frac{\alpha}{2}\sigma_z\right)\left(\cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle\right) \tag{1.37}$$

$$= \left(\cos\frac{\alpha}{2} - i\sin\frac{\alpha}{2}\right)\cos\frac{\theta}{2}|0\rangle + \left(\cos\frac{\alpha}{2} + i\sin\frac{\alpha}{2}\right)e^{i\varphi}\sin\frac{\theta}{2}|1\rangle \tag{1.38}$$

$$= e^{-i\alpha/2}\left(\cos\frac{\theta}{2}|0\rangle + e^{i(\varphi+\alpha)}\sin\frac{\theta}{2}|1\rangle\right) \sim \cos\frac{\theta}{2}|0\rangle + e^{i(\varphi+\alpha)}\sin\frac{\theta}{2}|1\rangle \tag{1.39}$$

となるため、確かに z 軸回りに α 回転する変換となっている。(グローバル位相だけ違う状態は同じ量子状態である。) 1 量子ビットの変換は、全て $R_{\boldsymbol{n}}(\theta)$ 演算子によって記述することができる。そのうち、特別な変換には特別な記号が与えられている：

$$X = \sigma_x = iR_{(1,0,0)}(\pi), \quad Y = \sigma_y = iR_{(0,1,0)}(\pi), \quad Z = \sigma_z = iR_{(0,0,1)}(\pi) \quad (\text{Pauli ゲート}), \tag{1.40}$$

$$H = iR_{(1/\sqrt{2},0,1/\sqrt{2})}(\pi) \quad (\text{Hadamard ゲート}), \tag{1.41}$$

$$S = e^{i\pi/4}R_{(0,0,1)}(\pi/2), \quad T = e^{i\pi/8}R_{(0,0,1)}(\pi/4)。 \tag{1.42}$$

X, Y, Z は直感的にわかりやすい変換であると思う。それぞれの固有ベクトルは、Hadamard 基底、円基底、計算基底である：

$$X\left|+\right\rangle = \left|+\right\rangle, X\left|-\right\rangle = -\left|-\right\rangle, \quad Y\left|i\right\rangle = \left|i\right\rangle, Y\left|i-\right\rangle = -\left|i-\right\rangle, \quad Z\left|0\right\rangle = \left|0\right\rangle, Z\left|1\right\rangle = -\left|1\right\rangle \text{ 。} \tag{1.43}$$

Pauli ゲートと呼ばれるのは、これらの変換を行列表示した際に各 Pauli 行列と一致しているためである。全体に i 倍のズレは生じるが、Pauli 行列を量子状態に作用させると、それは各座標軸回りに π 回転するようなものになっていると理解できる。また、量子計算において大事なのは、計算基底 $\left|0\right\rangle, \left|1\right\rangle$ がどのように変換するかということである。そのため、各 Pauli ゲートによって計算基底がどのように変化するかをまとめておく。なお、$a = \{0, 1\}$ とする：

$$X\left|a\right\rangle = \left|\bar{a}\right\rangle, \quad Y\left|a\right\rangle = (-1)^a i\left|\bar{a}\right\rangle, \quad Z\left|a\right\rangle = (-1)^a\left|a\right\rangle \text{。} \tag{1.44}$$

X と Y が作用すると $\left|0\right\rangle \leftrightarrow \left|1\right\rangle$ の変換が生じることは、Bloch 球での回転を考えたらわかる。ただし、係数が -1 倍されたりされなかったり、i 倍されたりされなかったりするのは覚えるしかない。X を作用させるときに Y や Z と同じように -1 倍の相対位相がつかない理由としては、Hadamard ゲート H が

$$H = \frac{1}{\sqrt{2}}(X + Z) \tag{1.45}$$

と表されるためだと考えれば良い。H が上のように $(X + Z)/\sqrt{2}$ によって表される Hermite 演算子であることと、

$$H\left|0\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle + \left|1\right\rangle) = \left|+\right\rangle, \quad H\left|1\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle) = \left|-\right\rangle \tag{1.46}$$

となることは常識にしておく必要がある（Bloch 球上で $(1, 0, 1)$ 方向の π 回転を考えればわかる）のだが、もしも $X\left|a\right\rangle = (-1)^a\left|\bar{a}\right\rangle$ と表されるとしたら、$H\left|0\right\rangle = H\left|1\right\rangle$ となってしまうため、$X\left|a\right\rangle = \left|\bar{a}\right\rangle$（NOT 演算）となるのだなと考えれば良い。NOT 演算という見方と対比させると、Z は相対位相の反転のみ、Y は（グローバル位相の変化を除いて）NOT 演算と位相反転を行う演算子と考えることもできる。実際に、X ゲートを NOT ゲート、Z ゲートを位相反転ゲートとも呼ぶ。同様に、S ゲートは $SS = Z$ であるため、z 方向の $\pi/2$ の位相シフトゲート、T は $TT = S$ より z 方向の $\pi/4$ の位相シフトゲートと考えれば良い。（つまり、Z ゲートと同じように、$\left|1\right\rangle$ にだけそれぞれのゲートに対する位相が付く。）

1.4 多量子ビット系

いままでは全て 1 量子ビット系を考えてきた。多数の量子ビットを考えるような系ももちろん存在する。例えば、1 つの粒子の状態を 1 量子ビットの状態に対応させて考えたとしよう。この粒子が 2 つあるような系は 2 量子ビット系に対応させて考える必要ができる。すなわち、粒子 1 の状態と粒子 2 の状態の組が系の状態を指定するということである。このように考えると、n 個の粒子が存在する系の取りうる状態は、2^n 個存在することがわかると思う。一般に、n ビット系では 2^n 次元のベクトル（各成分は 0 か 1）を用いてビットの状態を区別する。これを数学的に表す方法として、テンソル積というものを用いる。テンソル積の具体的な定義の説明は難しいが、具体的に扱うのはとても簡単である。例として、2 量子ビット系の表現方法を考えてみる。量子ビット 1 と、量子ビット 2 が $\left|\psi_1\right\rangle, \left|\psi_2\right\rangle$ と表されるとする：

$$\left|\psi_1\right\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}, \quad \left|\psi_2\right\rangle = \begin{pmatrix}\gamma\\ \delta\end{pmatrix}\text{。} \tag{1.47}$$

このとき、量子ビット 1 と量子ビット 2 の全体系の量子ビット状態 $\left|\psi\right\rangle$ はテンソル積 $\otimes$ を用いて

$$\left|\psi\right\rangle = \left|\psi_1\right\rangle \otimes \left|\psi_2\right\rangle = \begin{pmatrix}\alpha\left|\psi_2\right\rangle\\ \beta\left|\psi_2\right\rangle\end{pmatrix} = \begin{pmatrix}\alpha\gamma\\ \alpha\delta\\ \beta\gamma\\ \beta\delta\end{pmatrix} \tag{1.48}$$

と表される。一般に行列同士のテンソル積というものも同じように考えられる。例えば、

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \tag{1.49}$$

として、B を適当な $m \times n$ 行列とすると、

$$A \otimes B = \begin{pmatrix} A_{11}B & A_{12}B \\ A_{21}B & A_{22}B \end{pmatrix} \tag{1.50}$$

と計算されて、$2m \times 2n$ 行列となる。イメージとしては、テンソル積の左にある行列が、テンソル積をした後の行列全体の形を決定し、右の行列が局所的な形を決定しているという感じである。そのため、

$$I_2 \otimes \sigma_1 = \begin{pmatrix} 1 \times \sigma_1 & 0 \times \sigma_1 \\ 0 \times \sigma_1 & 1 \times \sigma_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{1.51}$$

と計算できる。ただ、量子計算においてはテンソル積の行列表示を書き下すことはない。なぜなら、テンソル積同士の積は、それぞれの場所における行列積のテンソル積になるためである：

$$(A \otimes B)(C \otimes D) = AC \otimes BD \text{ 。} \tag{1.52}$$

つまり、1 量子ビットにおける演算をマスターしておけば、多量子ビット系では単純に各量子ビットの演算をした後にテンソル積を作れば良い。例として、$|0\rangle \otimes |0\rangle$ の状態の 1 量子ビット目に X ゲート、2 量子ビット目に H ゲートを作用させた場合、

$$(X \otimes H)\,|0\rangle \otimes |0\rangle = X\,|0\rangle \otimes H\,|0\rangle = |1\rangle \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) = \frac{1}{\sqrt{2}}(|1\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle) \tag{1.53}$$

と計算できる。また、いちいち $\otimes$ を付けるのがめんどくさいので、

$$|i\rangle \otimes |j\rangle = |i\rangle\,|j\rangle = |ij\rangle \tag{1.54}$$

のように書くことが多い。

一般の多量子ビット状態について

量子状態の線形結合もまた量子状態となる。そのため、テンソル積をした量子状態の線形結合もまた量子状態となる。すなわち、

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{1.55}$$

のようなものも 2 量子ビットの状態として考えることができる。そして、これは 1 つのテンソル積にまとめて表すことができない。このようなものをエンタングル状態と呼び、量子テレポーテーションなどで大事な役割を果たす。これに対比させて、1 つのテンソル積で表せる状態は積状態と呼ぶ。結局のところ、

$$\text{「}n \text{ 個の 1 量子ビットのテンソル積」} \Rightarrow \text{「}n \text{ 量子ビット状態」} \tag{1.56}$$

だが、

$$\text{「}n \text{ 量子ビット状態」} \Rightarrow \text{「}n \text{ 個の 1 量子ビットのテンソル積」} \tag{1.57}$$

ではないということである。積状態とエンタングル状態の本質的な違いは、とある 1 つの量子ビットの測定結果がその他の量子ビットの測定結果に影響を与えるか否かである。積状態であれば、各量子ビットが独立しているとみなせるため、とある量子ビットを測定しても、他の量子ビットの状態は何も変わらない。しかし、例えば式 (1.55) のエンタングル状態であれば、1 量子ビット目の観測を行った結果 $|0\rangle$ であれば、2 量子ビット目の状態は $|0\rangle$ に確定する。なぜなら、式 (1.55) のエンタングル状態には $|01\rangle$ という状態が存在しないためである。

2　第 2 章

2.1　古典回路

定理 2.1（回路計算量の上界）
任意の $n \in \mathbb{N}$ に対して、任意の n ビット入力 1 ビット出力の論理関数 $f\colon \{0,1\}^n \to \{0,1\}$ の回路計算量はたかだか $5 \cdot 2^{n-1} - 4$ である。

証明. n ビットの入力に関する論理関数 f について

$$f(x_1, \ldots, x_{n-1}, x_n) = (\neg x_n \wedge f(x_1, \ldots, x_{n-1}, 0)) \vee (x_n \wedge f(x_1, \ldots, x_{n-1}, 1)) \tag{2.1}$$

が成立する。これは、$x_n = 0$ のときの論理関数の出力が $f(x_1, \ldots, x_{n-1}, 0)$ であり、$x_n = 1$ のときの論理関数の出力が $f(x_1, \ldots, x_{n-1}, 1)$ であることを表している。この式を用いると、n ビット入力 1 ビット出力の論理関数 f を計算する回路を構成する素子は、

$$2 \times (5 \cdot 2^{n-2} - 4) + 4 = 5 \cdot 2^{n-1} - 4 \tag{2.2}$$

個であることがわかる。これは、$n-1$ ビット入力 1 ビット出力の論理関数を計算する回路を構成する素子の個数に等しい。$n = 1$ のときは明らかに成立するため、帰納法より示された。 □

この証明でよくわからないのが、

$$f(x_1, \ldots, x_{n-1}, 0) \tag{2.3}$$

を構成する素子が

$$f(x_1, \ldots, x_{n-1}) \tag{2.4}$$

と等しいのは自明なのかがわからなかった。$x_n = 0$ なだけで、n 変数入力の論理関数ではないんだろうか？ 例えば、

$$f(x_1, \ldots, x_{n-1}, 0) = (g(x_1, \ldots, x_{n-1}) \vee 0) \wedge (h(x_1, \ldots, x_{n-1}) \wedge 1) \tag{2.5}$$

を計算するための素子の個数は $f(x_1, \ldots, x_{n-1})$ と等しいのだろか？？ そもそも、g, h を $5 \cdot 2^{n-1} - 4$ 個の回路計算量を持つ n 変数古典論理回路として、$f = g \wedge h$ としたら、f は n 変数古典論理回路であり、必要な素子数は $2 \times (5 \cdot 2^{n-1} - 4) + 1$ 個にならないのか。

この疑問への回答

現在考えているのは、論理関数 f を構成する素子の上界を求めていることであり、f を構成する素子の個数を求めているわけではない。そのため、$f = g \wedge h$ と表されたとしても、他の構成方法によって f を構成できればよい。帰納法の仮定により、$n-1$ 変数の論理関数を構成する素子の個数はたかだか $5 \cdot 2^{n-2} - 4$ 個であることがわかっている。そのため、$n-1$ 入力論理関数 f を構成する素子の上界は $5 \cdot 2^{n-1} - 4$ 個と考えて良く、帰納法が正しいことから、この仮定も正しい。まとめると、

$$f(x_1, \ldots, x_{n-1}, 0) \tag{2.6}$$

は $x_1, \ldots, x_{n-1}$ を入力とした $n-1$ 入力論理関数であり、その素子の個数は $5 \cdot 2^{n-2} - 4$ 個以下である。

定理 2.2（回路計算量の下界）
十分大きな任意の $n \in \mathbb{N}$ に対して、ある n 入力 1 出力の論理関数 $f\colon \{0,1\}^n \to \{0,1\}$ が存在して、その f の回路計算量は少なくとも $2^n/2n$ である。

証明. 任意の n 入力 1 出力の論理関数の総数は 2^{2^n} である。これは、入力が 2^n 通り存在し、それぞれの入力に対して出力が 2 通り存在するためである。次に、s 個以下の素子からなる回路で構成することのできる論理回路の個数を求める。まず、回路が s 個の素子から構成されているとする。s 個の素子それぞれの接続の仕方を考えると、各素子はたかだか 2 つの入力線をもち、その入力線は他の $s-1$ 個の素子の出力もしくは、n 個の入力線と接続されている。つまり、1 つの素子の接続方法はたかだか

$$\binom{n+s-1}{2} \tag{2.7}$$

通りである。これが s 個全ての素子で成立するため、s 個の素子で構成で構成できる回路の個数はたかだか

$$\binom{n+s-1}{2}^s \tag{2.8}$$

通りとなる。さらに、素子の種類としては $\wedge, \vee, \neg$ もしくは、何も操作をしない素子 I で構成される。例えば、I が $t(\leq s)$ 個回路中に存在すると、それは $s-t$ 個の素子からなる論理回路とみなすことができる。したがって、s 個以下の素子からなる論理回路で表現できる論理関数はたかだか

$$\binom{n+s-1}{2}^s \cdot 4^s = \frac{(n+s-1)^s(n+s-2)^s}{2^s} \cdot 4^s = (2\cdot(n+s-1)(n+s-2))^s \tag{2.9}$$

個である。$s \leq 2^n/2n$ とすると、

$$\begin{aligned}
(2\cdot(n+s-1)(n+s-2))^s &\leq (2\cdot(n+2^n/2n-1)(n+2^n/2n-2))^{2^n/2n} \\
&= \left\{2\left[\left(n^2+2^n+\left(\frac{2^n}{2n}\right)^2\right)-3\left(n+\frac{2^n}{2n}\right)+2\right]\right\}^{2^n/2n} \\
&= \left[2\left(\frac{2^n}{2n}\right)^2\right]^{2^n/2n}\left\{\left[1+\left(\frac{2n}{2^n}\right)^2\left(n^2+2^n-3\left(n+\frac{2^n}{2n}\right)+2\right)\right]\right\}^{2^n/2n} \\
&\underbrace{\to}_{n\to\infty}\left[2\left(\frac{2^n}{2n}\right)^2\right]^{2^n/2n} < \left(2\frac{2^{2n}}{2}\right)^{2^n/2n} = 2^{2^n}
\end{aligned} \tag{2.10}$$

と評価できる。すなわち、$s \leq 2^n/2n$ のとき、s 個以下の素子からなる論理回路で表現できる論理関数の個数は 2^{2^n} 個よりも少ない。そのため、回路計算量の下界が $2^n/2n$ となるような論理関数が存在する。 □

定理 2.3（回路計算量が高い関数の割合）
任意の $n \in \mathbb{N}$ に対して全ての 2^{2^n} 個の n ビット入力 1 ビット出力関数のうち、$(1-2^{-(\log n/n)2^n})2^{2^n}$ 個以上の関数の回路計算量は少なくとも $2^n/2n$ である。つまり、

$$\Pr\left\{\mathcal{C}(f) \geq \frac{2^n}{2n}\right\} \geq 1-2^{-(\log n/n)2^n} \tag{2.11}$$

となる。

証明. $s = 2^n/2n$ として、

$$P = \frac{2^s(n+s-1)^s(n+s-2)^s}{2^{2^n}} \tag{2.12}$$

とおく。これはランダムに論理回路を選んだ時に、$\mathcal{C}(f) \leq s$ となる確率を表しているため、$P \leq 2^{-(\log n/n)2^n}$ を示せばよい。$2^n = 2ns$ なので、

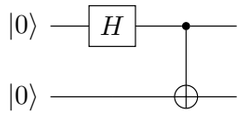
$$P = \left(\frac{2(n+s-1)(n+s-2)}{2^{2n}}\right)^s = \left(\frac{2(n+2^n/2n-1)(n+2^n/2n-2)}{2^{2n}}\right)^{2^n/2n} \tag{2.13}$$

$$\to \left(\frac{2(2^n/2n)^2}{2^{2n}}\right)^{2^n/2n} = \left(\frac{2^{2n-1}}{2^{2n}n^2}\right)^{2^n/2n} = \left(\frac{1}{2^{2\log n+1}}\right)^{2^n/2n} \leq 2^{-(\log n/n)2^n} \tag{2.14}$$

を得る。ただし、一般の n についての証明はよくわからない。（大きな n でしか問題にならないので考えなくても大丈夫な気がする。） □

2.2 量子回路

1 量子ビットに作用する量子回路は第 1 章で紹介した。複数量子ビットに作用する演算子も存在する。代表的なものとして、CNOT ゲートがある。CNOT ゲートが作用する例は以下のようなものである：

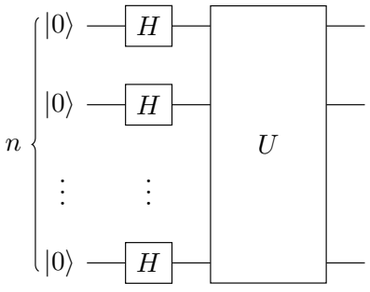


CNOT ゲートは、1 量子ビット目を制御ビット、2 量子ビット目をターゲットビットとして、制御ビットが $|1\rangle$ のときにのみターゲットビットに X ゲートを作用させるものである。したがって、

$$\mathrm{CNOT}H_0 |00\rangle = \mathrm{CNOT}\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{2.15}$$

と状態が変化する。この状態は Bell 状態と呼ばれ、エンタングルメントしている。なお、$H_0 = H \otimes I$ である。多量子ビットを制御ビットとする CNOT ゲートも考えることができる。2 つの量子ビットを制御ビットとする CNOT ゲートは、Toffori ゲートもしくは CCNOT ゲートと呼ばれる。

また、量子アルゴリズムで良く使われる量子回路として、



というものがある。U が作用する前の状態は、

$$H^{\otimes n}|0\rangle^{\otimes n} = H|0\rangle \otimes \cdots \otimes H|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \tag{2.16}$$

という状態である。これは、あらゆる n ビットの状態を等しい振幅で重ね合わせた状態であり、量子アルゴリズムの初期状態として良く用いられる。

古典論理関数の量子回路における計算可能性

古典論理関数の量子回路における計算可能性について考える。まず、CCNOT が任意の $x_1, x_2, x_3 \in \{0,1\}$ に対して

$$\mathrm{CCNOT}(x_1, x_2, x_3) = (x_1, x_2, (x_1 \wedge x_2) \oplus x_3) \tag{2.17}$$

という変換を行うことに注目する。$x_1 = x_2 = 1$ のとき、CCNOT は

$$\mathrm{CCNOT}(1, 1, x_3) = (1, 1, x_3 \oplus 1) = (1, 1, \neg x_3) \tag{2.18}$$

という変換を行う。さらに、$x_3 = 0$ を入力した場合、

$$\mathrm{CCNOT}(x_1, x_2, 0) = (x_1, x_2, x_1 \wedge x_2) \tag{2.19}$$

という変換を行う。これは、AND ゲートを表している。したがって、CCNOT ゲートを用いて AND ゲートを作ることができる。OR ゲートはこれらの組み合わせで作ることができる。これによって、以下の定理を得ることができる：

定理 2.4（古典論理関数の量子回路における計算可能性）
論理関数 $f\colon \{0,1\}^n \to \{0,1\}$ を回路サイズ $s(n)$ の論理回路で計算可能な任意の関数とする。このとき、

$$U_f \left|x\right\rangle \left|0\right\rangle \left|011\right\rangle \left|0^{l(n)}\right\rangle = \left|x\right\rangle \left|f(x)\right\rangle \left|011\right\rangle \left|G(x)\right\rangle \tag{2.20}$$

を満たす CCNOT 素子のみからなる量子回路が存在する。ただし、$l(n) = \mathcal{O}(s(n) + n)$ であり、$G(x)$ は $l(n)$ ビットの補助ビットである。

この場合、$|G(x)\rangle$ は U_f による x の計算過程で生じる不要な量子ビットである。この情報は不要なものであり、できれば可逆計算の過程で消去したい。先ほどの U_f を 2 つと CNOT を 1 つだけ使用した量子回路 U_f' を用いて、

$$U_f' \left|x\right\rangle \left|0\right\rangle \left|0\right\rangle \left|011\right\rangle \left|0^{l(n)}\right\rangle = \left|x\right\rangle \left|f(x)\right\rangle \left|0\right\rangle \left|011\right\rangle \left|0^{l(n)}\right\rangle \tag{2.21}$$

が成立するものが存在する。具体的な構成手順は、

$$U_f \left|x\right\rangle \left|0\right\rangle \left|0\right\rangle \left|011\right\rangle \left|0^{l(n)}\right\rangle = \left|x\right\rangle \left|0\right\rangle \left|f(x)\right\rangle \left|011\right\rangle \left|G(x)\right\rangle \tag{2.22}$$

$$\mathrm{CNOT} \left|x\right\rangle \left|0\right\rangle \left|f(x)\right\rangle \left|011\right\rangle \left|G(x)\right\rangle = \left|x\right\rangle \left|f(x)\right\rangle \left|f(x)\right\rangle \left|011\right\rangle \left|G(x)\right\rangle \tag{2.23}$$

$$U_f^{-1} \left|x\right\rangle \left|f(x)\right\rangle \left|f(x)\right\rangle \left|011\right\rangle \left|G(x)\right\rangle = \left|x\right\rangle \left|f(x)\right\rangle \left|0\right\rangle \left|011\right\rangle \left|0^{l(n)}\right\rangle \tag{2.24}$$

を組み合わせて

$$U_f' = U_f^{-1} \mathrm{CNOT} U_f \tag{2.25}$$

とすればよい。

3　第 3 章

本章のアルゴリズムのうち、非常にわかりやすいアルゴリズムである Deutsch-Jozsa のアルゴリズムを詳しく紹介する。その他のアルゴリズムについては、ここでは紹介しない。

3.1　Deutsch-Jozsa のアルゴリズム

Deutsch-Jozsa のアルゴリズムは、与えられた関数が定数関数かバランス関数かを判定するアルゴリズムである。定数関数とは、すべての $x \in \{0,1\}^n$ について $f(x) = 0$ もしくは $f(x) = 1$ となる関数のことであり、バランス関数とは、$f(x) = 0$ となる x と $f(x) = 1$ となる x が同数存在する関数、つまり $|x \in \{0,1\}^n \colon f(x) = 0| = 2^n/2$ である関数のことである。

定数関数/バランス関数判定問題

- 入力：関数 $f \colon \{0,1\}^n \to \{0,1\}$
- 出力：f が定数関数ならば 0、バランス関数ならば 1

関数 f はブラックボックスとして与えられている。このブラックボックスは、質問 x に対してそれに対応する $f(x)$ を返してくれる。質問を通じてのみ f に関する情報を得ることができる。このようなブラックボックスとして与えられた関数をオラクルと呼ぶ。オラクルが与えられる計算の場合、オラクルへの質問回数が計算量の指標として良く考えられ、質問計算量と呼ばれる。古典計算の場合、決定性計算であればどのようなアルゴリズムに対しても正しく判定するためには質問計算量が $2^{n-1}+1$ の f が存在する。一方、確率的計算であっても、確率 1 で正しく判定するためには指数回の質問が必要である。しかし、量子計算の場合、Deutsch-Jozsa のアルゴリズムを用いることで、1 回の質問で f が定数関数かバランス関数かを判定することができる。

関数 f はオラクルとして与えられえているので、量子回路からこのオラクルに質問するためには、回路の素子としてこのオラクルを実装する必要がある。ここでは、n 量子ビットと 1 量子ビットの 2 つの量子ビット列に作用するユニタリ行列 U_f としてオラクルが実装されているとする：

$$U_f \left|x\right\rangle \left|b\right\rangle = \left|x\right\rangle \left|b \oplus f(x)\right\rangle \ 。 \tag{3.1}$$

この回路は初期量子ビット列 $\left|0^{n+1}\right\rangle = \left|0^n\right\rangle \left|0\right\rangle$ を入力として受け取り、回路が出力する量子ビット列を測定することで U_f に関する情報を得る。

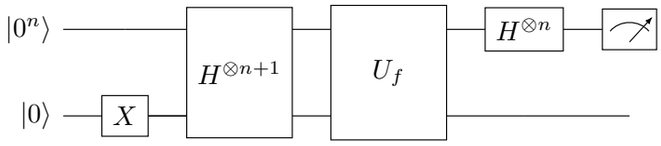
Deutsch-Jozsa のアルゴリズム

Deutsch-Jozsa のアルゴリズムは、以下のような構成である：

- $n+1$ ビット目のみに Pauli 行列 σ_1 を作用させる
- $n+1$ ビット全体に Hadamard 変換 H を作用させる
- $n+1$ ビット全体に U_f を作用させる
- 最初の n ビット全体に Hadamard 変換 H を作用させる
- 最初の n ビット全体を測定する

このアルゴリズムの回路図は以下のようになる：[*2]

[*2] この回路図は、githubcopilot が作成してくれた。驚愕



まず、$n+1$ ビット目に σ_1 を作用させることで、

$$\sigma_1 |0^n\rangle |0\rangle = |0^n\rangle |1\rangle \tag{3.2}$$

となる。次に、$n+1$ ビット全体に Hadamard 変換を作用させることで、$N = 2^n$ として

$$H^{\otimes n+1} |0^n\rangle |1\rangle = \frac{1}{\sqrt{2N}} \sum_{x\in\{0,1\}^n} |x\rangle (|0\rangle - |1\rangle) \tag{3.3}$$

となる。この状態に U_f を作用させると、

$$U_f \frac{1}{\sqrt{2N}} \sum_{x\in\{0,1\}^n} |x\rangle (|0\rangle - |1\rangle) = \frac{1}{\sqrt{2N}} \sum_{x\in\{0,1\}^n} |x\rangle (|f(x)\rangle - |\neg f(x)\rangle) = \frac{1}{\sqrt{2N}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle) \tag{3.4}$$

を得る。ここで、任意の $x \in \{0,1\}^n$ に対して

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} (-1)^{\langle x,y\rangle} |y\rangle \tag{3.5}$$

が成立することを用いる。なお、

$$\langle x, y\rangle = \sum_{i=1}^{n} x_i y_i \tag{3.6}$$

である。これは、y を 2 進数表記の i 桁目の値 y_i が 1 であり、かつ x の 2 進数表記の i 桁目の値 x_i が 1 であるときに、符号を -1 倍することを表している。このように変換された状態のうち、最初の $|0^n\rangle$ の振幅に注目すると、

$$\Pr\{\text{測定結果が } |0^n\rangle \text{ である}\} = \left| \frac{1}{N} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} \right|^2 = \begin{cases} 1 & f \text{ が定数関数} \\ 0 & f \text{ がバランス関数} \end{cases} \tag{3.7}$$

となることがわかる。すなわち、f が定数関数であるとき、測定結果は $|0^n\rangle$ となり、f がバランス関数であるとき、測定結果は $|0^n\rangle$ でない状態が得られる。

3.2 Glover のアルゴリズム

Glover のアルゴリズムは、Deutsch-Jozsa のアルゴリズムより汎用的なアルゴリズムである。このアルゴリズムは、与えられた関数 $f\colon \{0,1\}^n \to \{0,1\}$ の充足解、すなわち $f(x_0) = 1$ となる x_0 を見つけるという一般的な探索問題に適用できる。このアルゴリズムにおいてカギを握るのが、Glover の拡散行列 D_N というものである。これは、$N = 2^n$ として $N \times N$ 次元の行列であり、

$$D_N = \begin{pmatrix} \frac{2}{N} - 1 & \frac{2}{N} & \cdots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \cdots & \frac{2}{N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{N} & \frac{2}{N} & \cdots & \frac{2}{N} - 1 \end{pmatrix} \tag{3.8}$$

と定義されるものである。まずは、この行列が

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x\in\{0,1\}^n} |x\rangle \tag{3.9}$$

として、

$$D_N = -I + 2\,|s\rangle\langle s| \tag{3.10}$$

と表されることを証明する。

証明. 数学的帰納法により示す。$n = 1$ のとき、

$$|s\rangle\langle s| = \frac{1}{2}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| + |1\rangle\langle 1|) = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \frac{1}{2}M_1 \tag{3.11}$$

である。なお、M_n は $2^n \times 2^n$ 次元の行列であり、

$$M_n = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix} \tag{3.12}$$

である。

$$\sum_{x\in\{0,1\}^n} |x\rangle = \sum_{x\in\{0,1\}^{n-1}} |x\rangle \otimes \sum_{i=\{0,1\}} |i\rangle \tag{3.13}$$

であることから、

$$|s\rangle\langle s| = \frac{1}{N}\left(\sum_{x\in\{0,1\}^{n-1}} |x\rangle \otimes \sum_{i=\{0,1\}} |i\rangle\right)\left(\sum_{x\in\{0,1\}^{n-1}} \langle x| \otimes \sum_{i=\{0,1\}} \langle i|\right) = \frac{1}{N}M_{n-1} \otimes M_1 = \frac{1}{N}M_n \tag{3.14}$$

と計算できる。したがって、

$$D_N = -I + 2\,|s\rangle\langle s| \tag{3.15}$$

となる。拡散行列が $|x\rangle$ に作用すると、

$$D_N\,|x\rangle = -\,|x\rangle + \frac{2}{\sqrt{N}}\,|s\rangle\langle s|x\rangle = -\,|x\rangle + \frac{2}{\sqrt{N}}\,|s\rangle \tag{3.16}$$

となるため、

$$D_N\left(-\,|x_0\rangle + \sum_{x\neq x_0} |x\rangle\right) = |x_0\rangle - \frac{2}{\sqrt{N}}\,|s\rangle + \sum_{x\neq x_0}\left(-\,|x\rangle + \frac{2}{\sqrt{N}}\,|s\rangle\right) = |x_0\rangle + \frac{2}{\sqrt{N}}(N-2)\,|s\rangle - \sum_{x\neq x_0} |x\rangle \tag{3.17}$$

$$= \left(1 + \frac{2}{N}(N-2)\right)|x_0\rangle + \left(\frac{2}{N}(N-2) - 1\right)\sum_{x\neq x_0} |x\rangle \tag{3.18}$$

$$= (3 - 4/N)\,|x_0\rangle + (1 - 4/N)\sum_{x\neq x_0} |x\rangle \tag{3.19}$$

と計算できる。 □

3.3 Shor のアルゴリズム