量子計算入門

量子力学の復習

波動関数・量子状態

- 電子のような量子的な物理系の状態は**波動関数** $\psi(x)$ で表される。
- 波動関数の物理的な意味：位置 x に電子が存在する確率密度が $|\psi(x)|^2$。
- 波動関数は**規格化条件** $\int|\psi(x)|^2dx = 1$ を必ず満たす。
- 波動関数は適当な正規直交基底関数 $\phi_k(x)$ によって以下のように展開できる。

$$\psi(x) = \sum_k c_k\phi_k(x)$$

規格化条件は $\sum_k|c_k|^2 = 1$ に翻訳される。

- 例：ある位置 y にだけ電子が存在する状態 $\delta(x-y)$ によって展開する。

$$\psi(x) = \int dy\, \psi(y)\delta(x-y)$$

- 係数 c_k を並べたベクトルは $\psi(x)$ と同一視できて、$\psi(x)$ は関数空間でのベクトルとみなせる。このベクトルを $|\psi\rangle$ と書く。(読み方：ケットプサイ/プサイケット)

$$|\psi\rangle = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \end{pmatrix}$$ ※ ϕ_k を基底とする場合

ケットベクトル/ブラベクトル

- $|\psi\rangle$ の双対ベクトル (複素共役転置) を $\langle\psi|$ と書く。(読み方：ブラプサイ/プサイブラ)

$$|\psi\rangle = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \end{pmatrix} \Leftrightarrow \langle\psi| = \begin{pmatrix} c_1^* & c_2^* & \cdots \end{pmatrix}$$

- $|a\rangle = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \end{pmatrix}$ と $|b\rangle = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \end{pmatrix}$ の内積は $\langle a|b\rangle$ と書く。

$$\langle a|b\rangle = \begin{pmatrix} a_1^* & a_2^* & \cdots \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \end{pmatrix}$$

- $|a\rangle\langle b|$ は演算子 (行列) になる。

$$|a\rangle\langle b| = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \end{pmatrix} \begin{pmatrix} a_1^* & a_2^* & \cdots \end{pmatrix} = \begin{pmatrix} b_1 a_1^* & b_1 a_2^* & \cdots \\ b_2 a_1^* & b_2 a_2^* & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

物理量の測定

- 量子系に対して物理量の観測を行うと、$|\psi\rangle$ とその物理量に対応する演算子 A によって決まる確率分布から**ランダムに結果が得られる。**
- 観測結果は必ず A の固有値 a_i のどれかになる。
- a_i が得られる確率 p_i は、対応する固有ベクトル $|a_i\rangle$ を使って以下のように計算される。

$$p_i = |\langle a_i|\psi\rangle|^2$$

Remark: $|\psi\rangle$ を $e^{i\theta}|\psi\rangle$（**グローバル位相**を付加）としても物理量の観測には影響しない。

- 物理量 A の期待値 $\langle A\rangle$ は、普通の確率論と同じように定義される。

$$\langle A\rangle = \sum_i a_i p_i$$

演習（レポート）：$|\psi\rangle$ に対する物理量 A の期待値 $\langle A\rangle$ が $\langle A\rangle = \langle\psi|A|\psi\rangle$ と計算できることを示してください。

演算子について

- A の複素共役転置を $A^\dagger$ と書く。

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots \\ A_{21} & A_{22} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \Leftrightarrow A^\dagger = \begin{pmatrix} A_{11}^* & A_{21}^* & \cdots \\ A_{12}^* & A_{22}^* & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

- $A = A^\dagger$ を満たすとき、A **はエルミート**であるという。

Remark: エルミート演算子の固有値は実数 → 物理量はエルミート演算子によって表現される。

- $U^\dagger = U^{-1}$ を満たすとき、U **はユニタリー**であるという。

Remark1: ユニタリー演算子はベクトルの大きさを変えない。$\||\psi\rangle\|^2 = \langle\psi|\psi\rangle$ として、

$$\||\psi\rangle\|^2 = \|U|\psi\rangle\|$$

規格化条件はユニタリ変換で不変

Remark2: ユニタリー演算子の固有値の大きさは 1 → 固有値は $e^{i\phi}$ の形になる。

- 演算子のブラケット表示

$$A = \sum_{ij} A_{ij}|i\rangle\langle j| = \sum_{ij} a_i|a_i\rangle\langle a_i|$$

対角化した表現

量子状態の時間発展

- (孤立)量子系の時間発展は、エネルギーに対応する演算子であるハミルトニアン H を使って、以下の**シュレディンガー方程式**によって決定される。

$$\frac{d}{dt}|\psi(t)\rangle = -iH|\psi(t)\rangle$$

- 初期状態が $|\psi(0)\rangle$ のとき、方程式の解は

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$$

※ $e^{A} = I + A + \frac{1}{2}A^2 + \cdots$

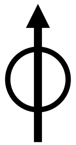
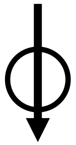
- e^{-iHt} はユニタリ演算子 → **(孤立) 量子系の時間発展は常にユニタリ。**

H をデザインして量子状態を制御し、量子力学を計算に活用するのが量子計算

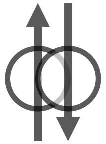
量子ビット

量子ビットとは

- 量子力学的な 2 準位系を**量子ビット**と呼ぶ。
- 代表例は S=1/2 のスピン。

 $= |0\rangle$  $= |1\rangle$

- 量子ビットは量子力学的な重ね合わせ状態を取れる。

 $= \alpha|0\rangle + \beta|1\rangle$

- 数学的には2次元の複素ベクトルと等価。

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

パウリ演算子

- 量子ビットに対する以下の 3 つの演算子を**パウリ演算子**と呼ぶ。

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

※ $|0\rangle, |1\rangle$ 基底での行列表示

- 物理的には、S=1/2 のスピンの x, y, z 方向の磁気モーメントに対応する。
- パウリ演算子は**エルミートかつユニタリ**。
- 単位行列 $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ を含めることもある。

演習：

(1)　X, Y の固有ベクトルを計算して、$|0\rangle, |1\rangle$ の和で表してください。

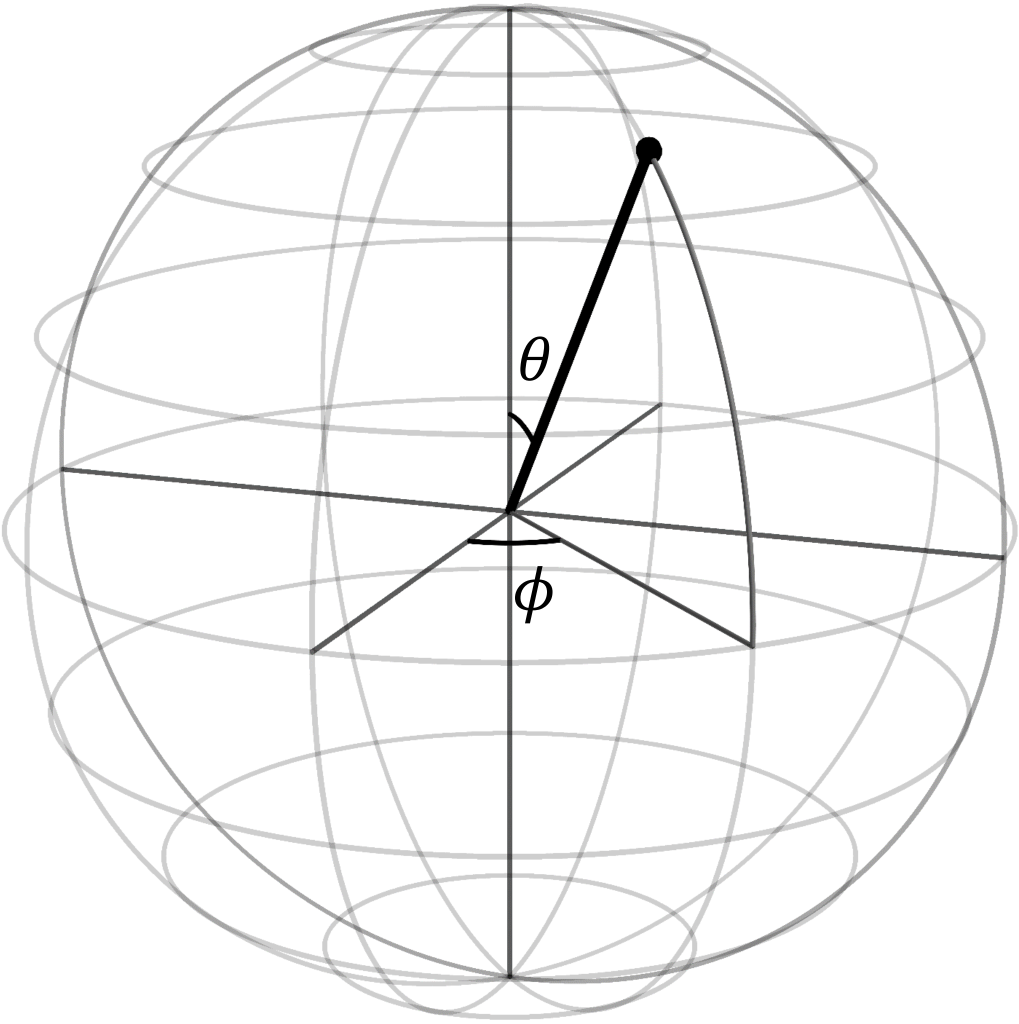
(2)　パウリ演算子が $X^2 = Y^2 = Z^2 = I$ を満たすことを示してください。

ブロッホ球

- １量子ビットの任意の状態ベクトルは、グローバル位相を除いて以下のように書ける。

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

- (θ, ϕ) は球面上の一点に対応付けられる。
- このような表現方法を**ブロッホ球**表現と呼ぶ。



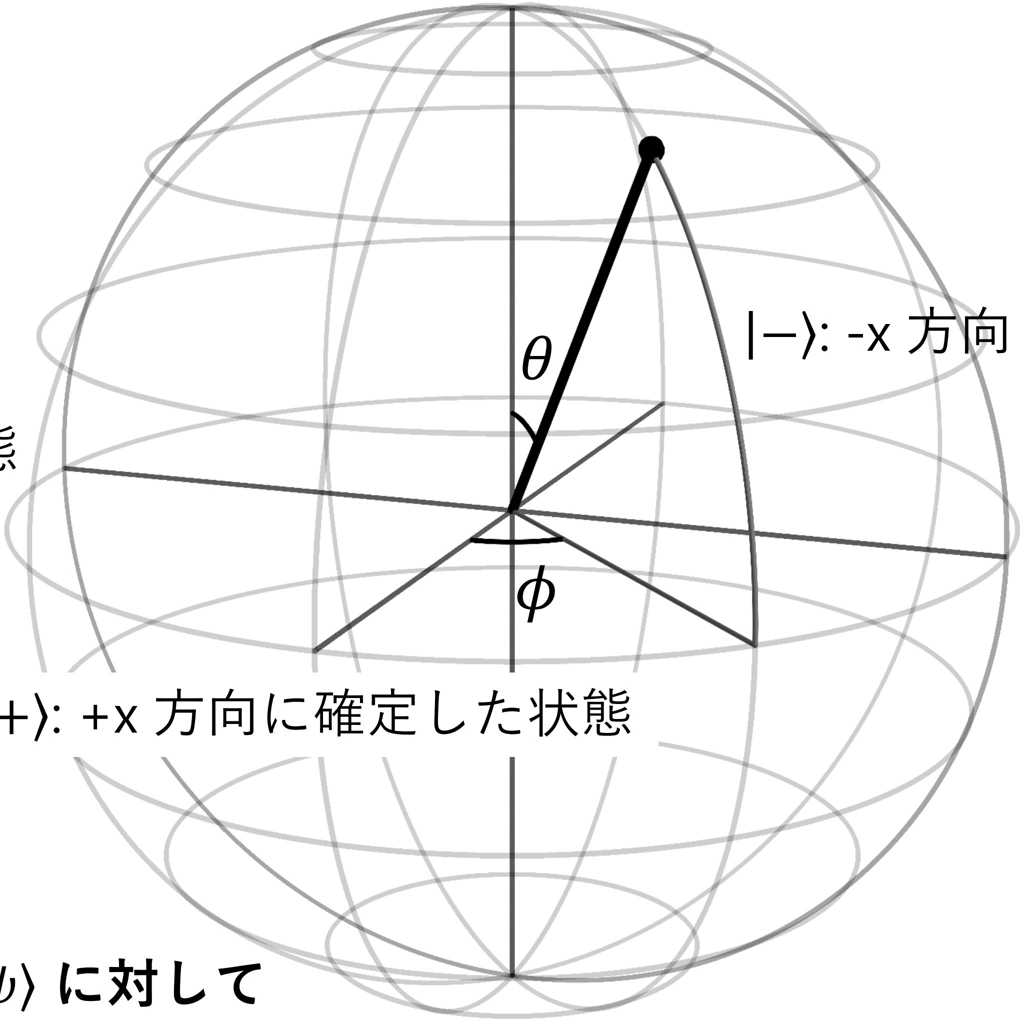
演習：

X, Y, Z の固有ベクトルをブロッホ球上にプロットしてください。

ブロッホ球上の量子状態の物理的意味

S=1/2 のスピンの言葉に翻訳すると

$|0\rangle$: +z 方向に確定した状態



$|-\rangle$: -x 方向に確定した状態

$|-i\rangle$: -y方向に確定した状態

$|+i\rangle$: +y 方向に確定した状態

$|+\rangle$: +x 方向に確定した状態

$|1\rangle$: -z 方向に確定した状態

特に 1 量子ビットの状態 $|\psi\rangle$ に対して $(\langle X\rangle, \langle Y\rangle, \langle Z\rangle)$ はブロッホ球上の座標になる！

パウリ演算子による１量子ビット操作

パウリ演算子はどのように作用するか？

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- パウリ X は量子ビットを反転する。**=NOTゲート**

$$X|0\rangle = |1\rangle, \qquad X|1\rangle = |0\rangle$$

- パウリ Y も量子ビットを反転するが、位相をつける。

$$Y|0\rangle = i|1\rangle, \qquad Y|1\rangle = -i|0\rangle$$

- パウリ Z は $|1\rangle$ に位相をつける。

$$Z|0\rangle = |0\rangle, \qquad Z|1\rangle = -|1\rangle$$

演習：

Z を X の固有ベクトルに作用させるとどうなるか計算してください。

回転ゲート

以下の 3 **つのユニタリをそれぞれ** x, y, z **回転ゲートと呼ぶ。**

$$R_x(\theta) = e^{-i\theta X/2} \quad R_y(\theta) = e^{-i\theta Y/2} \quad R_z(\theta) = e^{-i\theta Z/2}$$

演習：x 軸回転ゲートについて、以下の関係式を示してください。

$$e^{-i\theta X/2} = I\cos\frac{\theta}{2} - iX\sin\frac{\theta}{2}$$

ブロッホ球上での回転ゲート

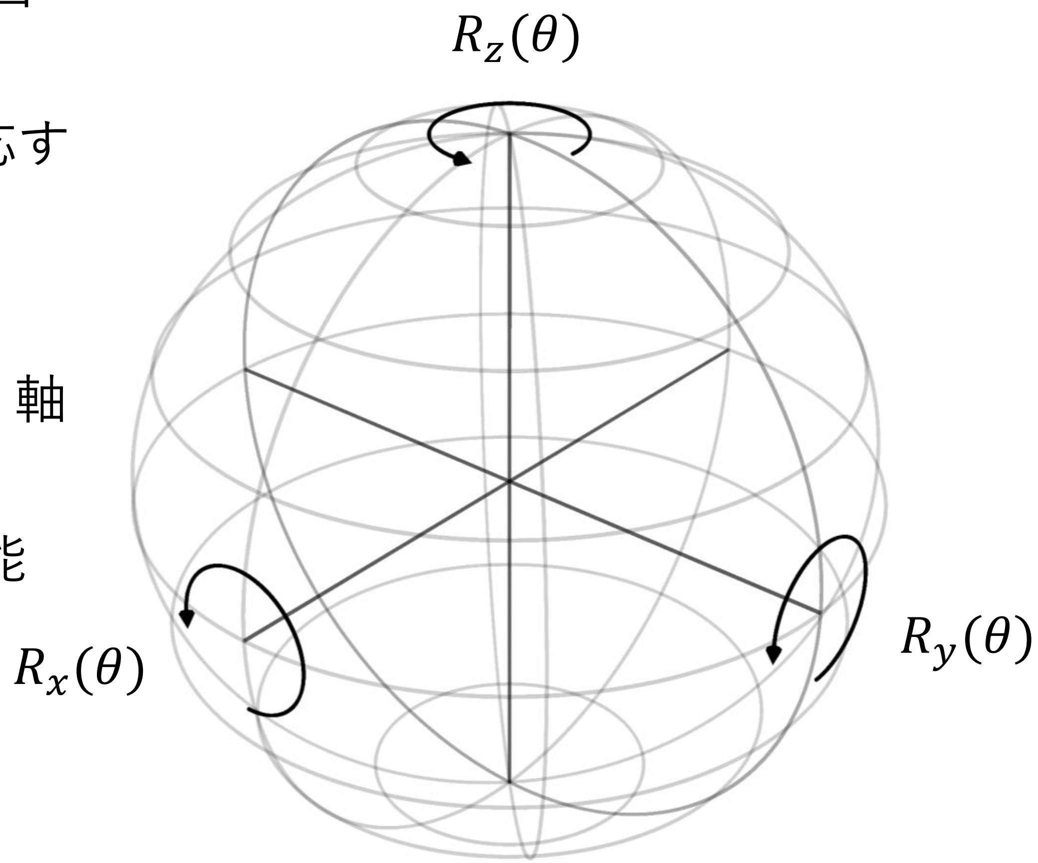
- x, y, z 回転ゲートはブロッホ球上でみるとx, y, z 軸の回転に対応する。
- 1 量子ビットの任意の状態はブロッホ球上に1対1対応するので、

1量子ビットのユニタリ変換 = ブロッホ球上の回転

- 任意の 3 次元空間の回転は z 軸回転 → x 軸回転 → z 軸回転で表せる。

→ 任意の 1 量子ビットユニタリは、次のように分解可能

$$U = e^{i\alpha} R_z(\theta_3) R_x(\theta_2) R_z(\theta_1)$$



その他の重要なゲートとその性質

- **アダマールゲート**

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

- **S ゲート**

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

- **T ゲート**

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/8} \end{pmatrix}$$

演習：

(1) $H|0\rangle = |+\rangle$, $H|1\rangle = |-\rangle$ を示してください。

(2) $H^2 = I$, $HXH = Z$, $HZH = X$ を示してください。

(3) $SXS^\dagger = Y$, $S^\dagger YS = X$ を示してください。

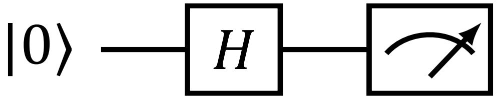
量子ビットの測定/量子回路

測定：

- $|\psi\rangle$ を**測定**すると、$|0\rangle$ or $|1\rangle$ がランダムに得られる。
- それぞれの確率は $|\langle 0|\psi\rangle|^2$ と $|\langle 1|\psi\rangle|^2$
- 観測結果に応じて、量子ビットは $|0\rangle$ か $|1\rangle$ に収束する。**(射影測定)**

量子回路：

- 量子ビットに対する操作を視覚的にわかりやすくする。
- 下の例は、$|0\rangle$ に初期化 → H ゲートを作用 → 測定 のシーケンスを表す量子回路。



演習

演習 (レポート)：

(1) 状態 $|\psi\rangle$ に用意した量子ビットを N 回 $|0\rangle, |1\rangle$ 射影測定したら、0 が N_0 回、1 が N_1 回得られた。この状態 $|\psi\rangle$ に対する Z の期待値 $\langle\psi|Z|\psi\rangle$ を推定してください。

(2) 状態 $|\psi\rangle$ が与えられたとき、1量子ビットの任意のエルミート演算子 H に対して、$\langle\psi|H|\psi\rangle$ を推定したい。量子ビットに対する操作として、

・任意の1量子ビットゲート

・$|0\rangle, |1\rangle$ の射影測定

が許されているとき、どのように推定すればよいでしょうか？

複数量子ビット

複数の量子系

- 2 つの量子系 A, B があるとき、その状態はそれぞれの状態ベクトル空間の**テンソル積**によって記述される。
- A の基底を $|i\rangle_A$, B の基底を $|i\rangle_B$ とするとき、複合系の状態は一般に

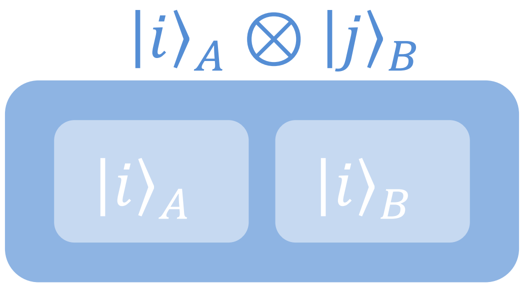
$$\sum_{i,j} c_{ij} \, |i\rangle_A \otimes |j\rangle_B$$

$\otimes$：ベクトルのテンソル積を表す記号

  になる。
- 特に、A が $|\psi\rangle_A$ の状態で、 B が $|\phi\rangle_B$ の状態にあるとき、複合系の状態は

$$|\psi\rangle_A \otimes |\phi\rangle_B$$

  となる。
- この状態を $|i\rangle_A, |i\rangle_B$ 基底で測定すると、$\left|c_{ij}\right|^2$ の確率で A に i が、B に j が観測される。



テンソル積の演算規則など

- 普通の掛け算のように計算して OK！

$$\left(\sum_i a_i|i\rangle\right)\otimes\left(\sum_j b_j|j\rangle\right)=\sum_{i,j} a_i b_j|i\rangle\otimes|j\rangle$$

- 内積

$$(\langle a|\otimes\langle b|)(|c\rangle\otimes|d\rangle)=\langle a|c\rangle\langle b|d\rangle$$

- $\otimes$ を毎回書くのは面倒なので、よく次のように省略する。

$$|i\rangle\otimes|j\rangle=|i\rangle|j\rangle=|ij\rangle$$

- ベクトル表記でのテンソル積の計算方法

$$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}\otimes\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}=\begin{pmatrix} a_1\begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \\ a_2\begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \\ \vdots \end{pmatrix}=\begin{pmatrix} a_1 b_1 \\ a_1 b_2 \\ \vdots \\ a_n b_m \end{pmatrix}$$

テンソル積について演習問題

演習：

(1) $a|0\rangle + b|1\rangle$ と $c|0\rangle + d|1\rangle$ の状態にある 2 つの量子ビットがある。この複合系の状態をテンソル積を使って計算してください。

(2) 各ビット列 00, 01, 10, 11 が観測される確率は、直感的なものとあっているでしょうか？

(3) n 量子ビットの状態は、何次元のベクトルでしょうか？

演算子のテンソル積

- 複合系に作用する演算子は、演算子のテンソル積で記述される。
- 演算子 $A = \sum_{i,j} A_{ij} |i\rangle\langle j|$ と $B = \sum_{i,j} B_{ij} |i\rangle\langle j|$ のテンソル積は以下のように計算できる。

$$
\begin{aligned}
A \otimes B &= \left(\sum_{i,j} A_{ij} |i\rangle\langle j| \right) \otimes \left(\sum_{k,l} B_{kl} |k\rangle\langle l| \right) \\
&= \sum_{ijkl} A_{ij} B_{kl} (|i\rangle\langle j|) \otimes (|k\rangle\langle l|) \\
&= \sum_{ijkl} A_{ij} B_{kl} (|i\rangle \otimes |k\rangle)(\langle j| \otimes \langle l|) \\
&= \sum_{ijkl} A_{ij} B_{kl} |ik\rangle\langle jl|
\end{aligned}
$$

- $A \otimes B$ は1番目の系に A を作用させ、2番目の系に B を作用させるという演算子

演算子のテンソル積 (行列表示)

- 行列表記でのテンソル積の計算方法

$$
\mathrm{A} \otimes B = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{pmatrix} \otimes \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1m} \\ B_{21} & B_{22} & \cdots & B_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{m2} & \cdots & B_{mm} \end{pmatrix}
$$

$$
= \begin{pmatrix} A_{11}\begin{pmatrix} B_{11} & \cdots & B_{1m} \\ \vdots & \ddots & \vdots \\ B_{m1} & \cdots & B_{mm} \end{pmatrix} & A_{12}\begin{pmatrix} B_{11} & \cdots & B_{1m} \\ \vdots & \ddots & \vdots \\ B_{m1} & \cdots & B_{mm} \end{pmatrix} & \cdots \\ A_{21}\begin{pmatrix} B_{11} & \cdots & B_{1m} \\ \vdots & \ddots & \vdots \\ B_{m1} & \cdots & B_{mm} \end{pmatrix} & A_{22}\begin{pmatrix} B_{11} & \cdots & B_{1m} \\ \vdots & \ddots & \vdots \\ B_{m1} & \cdots & B_{mm} \end{pmatrix} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}
$$

$$
= \begin{pmatrix} A_{11}B_{11} & A_{11}B_{12} & \cdots & A_{1n}B_{1m} \\ A_{11}B_{21} & A_{11}B_{22} & \cdots & A_{1n}B_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1}B_{m1} & A_{n1}B_{m2} & \cdots & A_{nn}B_{mm} \end{pmatrix}
$$

行列のテンソル積について演習問題

演習：

(1) $X \otimes I, X \otimes X, I \otimes Z$ の行列表示を計算してください。

(2) 2量子ビットの状態

$$a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

に対して上で求めた行列を作用させてみてください。

(3) $A \otimes B$ という演算子が、各系に A, B をそれぞれ作用させる演算子となっていることを、上の例で確認しましょう。

(4) n 量子ビットに作用する演算子は、何次元の行列でしょうか？

2量子ビットゲート

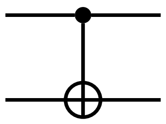
代表的な 2 量子ビットゲート

➢ **Controlled-NOT (CNOT) ゲート**
control bit が 1 のときのみ、target bit を反転する。

$$\mathrm{CNOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$$

回路記号

Control
Target



入出力対応

入力	出力
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|01\rangle$
$\|10\rangle$	$\|11\rangle$
$\|11\rangle$	$\|10\rangle$

行列表記

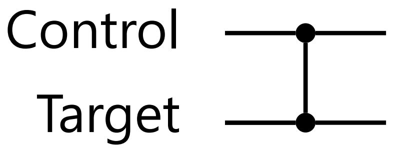
$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

代表的な 2 量子ビットゲート

➢ **Controlled-Z (CZ) ゲート**
control bit が 1 のときのみ、target bit に Z ゲートをかける。

$$\mathrm{CZ} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z$$

回路記号



入出力対応

入力	出力
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|01\rangle$
$\|10\rangle$	$\|10\rangle$
$\|11\rangle$	$-\|11\rangle$

行列表記

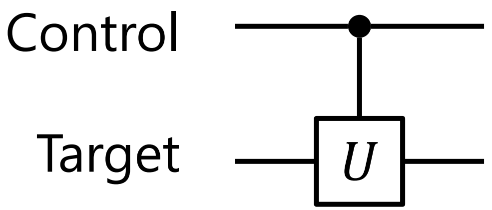
$$\mathrm{CZ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

代表的な 2 量子ビットゲート

- **Controlled-U ゲート**
  control bit が 1 のときのみ、target bit に U をかける。

$$\mathrm{C}(U) = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U$$

回路記号



入出力対応

入力	出力
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|01\rangle$
$\|10\rangle$	$\|1\rangle \otimes U\|0\rangle$
$\|11\rangle$	$\|1\rangle \otimes U\|1\rangle$

行列表記

$$\mathrm{C}(U) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{11} & U_{12} \\ 0 & 0 & U_{21} & U_{22} \end{pmatrix}$$

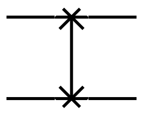
代表的な 2 量子ビットゲート

- **SWAP ゲート**
  2 量子ビットの状態を交換する。

$$\mathrm{SWAP} = |00\rangle\langle 00| + |01\rangle\langle 10| + |10\rangle\langle 01| + |11\rangle\langle 11|$$

回路記号

Control
Target



入出力対応

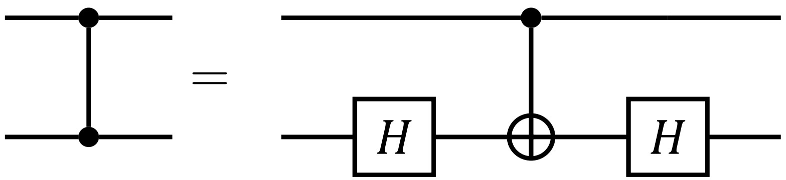
入力	出力
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|10\rangle$
$\|10\rangle$	$\|01\rangle$
$\|11\rangle$	$\|11\rangle$

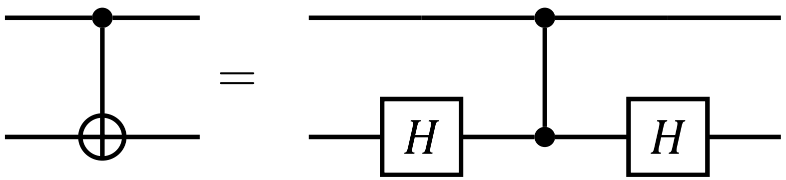
行列表記

$$\mathrm{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

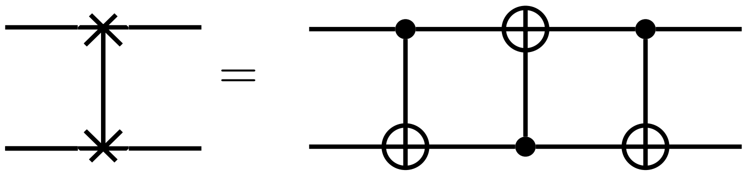
代表的な 2 量子ビットゲート間の関係

- CZ ゲート ⇔ CNOT ゲート ($HZH = X$ の関係式による。)





- CNOT ゲート ⇒ SWAP ゲート

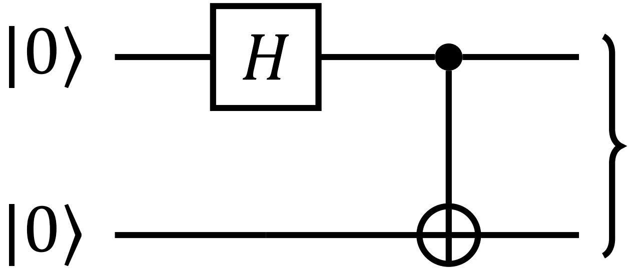


演習：これらの関係が正しいことを示してください

ちょっと脇道：
エンタングルメント/量子テレポーテーション

エンタングルメント

- **積状態** (product state)：
  $|a\rangle \otimes |b\rangle$ の形に書ける状態のこと。それぞれの量子ビットが独立に $|a\rangle, |b\rangle$ の状態にある状況に対応する。
- **エンタングル状態** (entangled state)：
  $|a\rangle \otimes |b\rangle$ の形で書けない状態のこと。「それぞれの量子ビットがある特定の状態にいる」という考え方が破綻する。
- もつれ状態の例：**ベル状態**



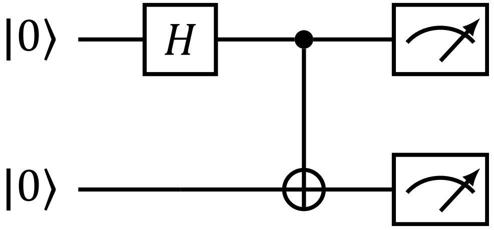
$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

演習：

ベル状態 $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ がエンタングルしていることを示してください。

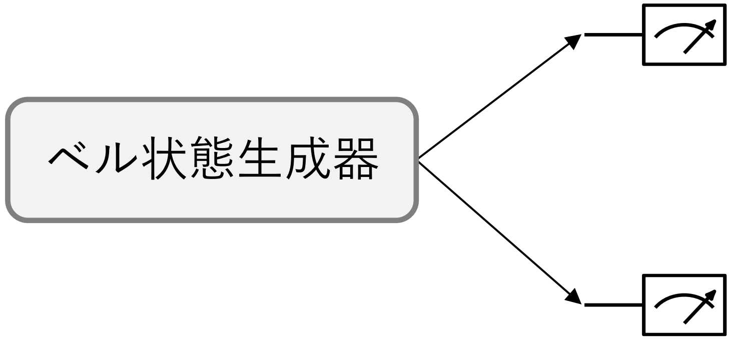
エンタングルメントと古典相関

- エンタングルメントに関してよくある説明：

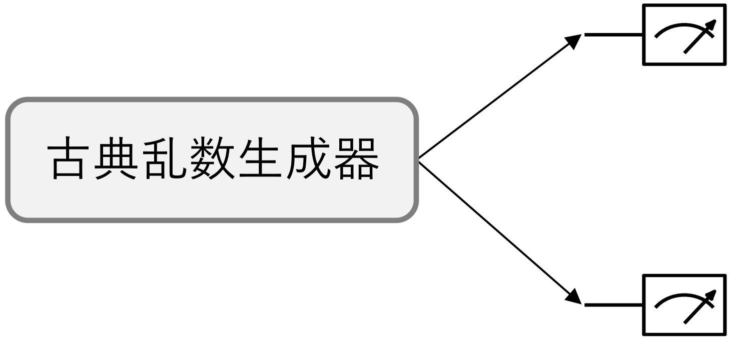


00 or 11 しか出ない！
片方を知るともう片方がわかる → エンタングルしている！

- でもこれは古典相関でも可能。00 or 11 しか出ない乱数生成器は簡単に作れる。



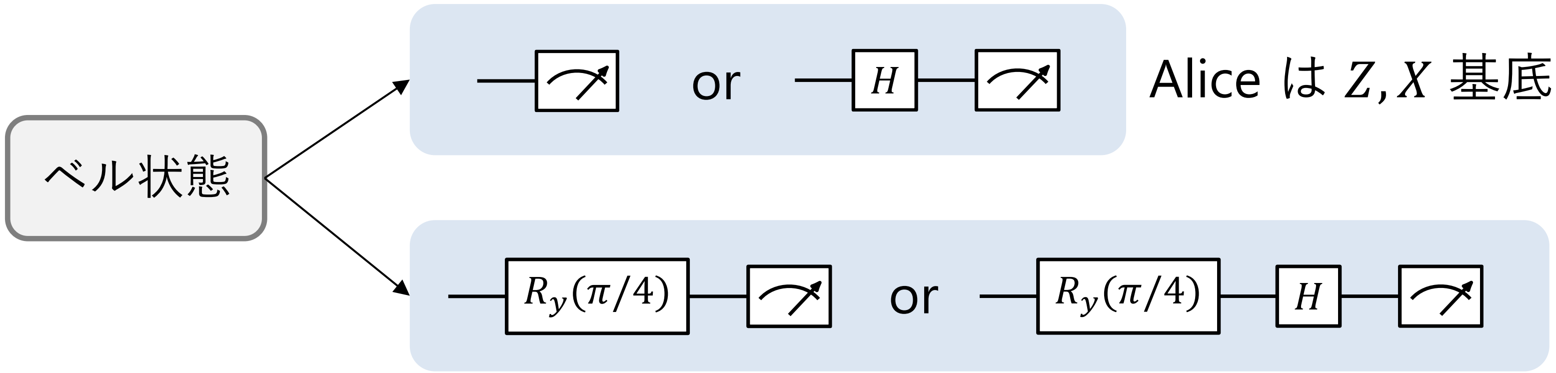
=



どちらも 00 or 11 しか出ない。最後に観測する人からすればどちらも等価。

量子相関

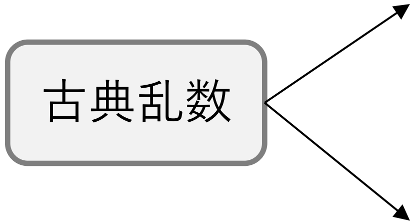
- 送られてきた情報の複数の側面を見なければ、古典と同じ。
- 下のようなセットアップで得られる確率分布は、**古典情報の送受信では絶対に実現できない**ことが知られている。**(CHSH 不等式)**



Alice は Z, X 基底でランダムに測定

Bob は $(Z \pm X)/\sqrt{2}$ 基底でランダムに測定

どのような A_1, A_2, B_1, B_2 を持ってきても再現不可能

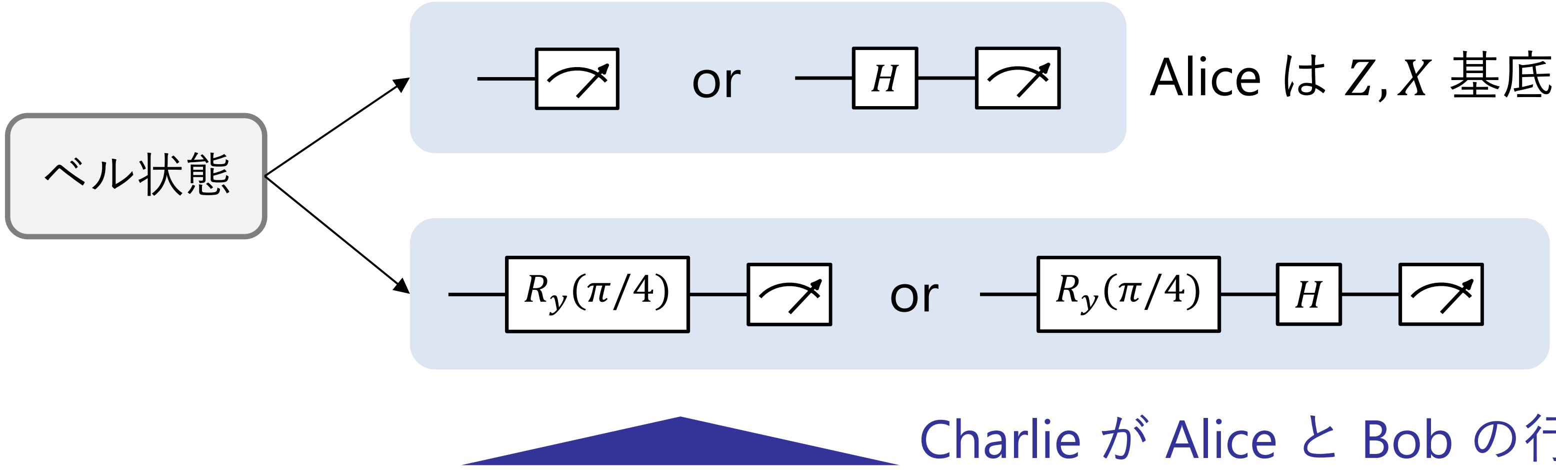


Alice は送られてきた乱数に対してブール関数 A_1, A_2 を計算する。

Bob は送られてきた乱数に対してブール関数 B_1, B_2 を計算する。

量子相関

- 送られてきた情報の複数の側面を見なければ、古典と同じ。
- 下のようなセットアップで得られる確率分布は、**古典情報の送受信では絶対に実現できない**ことが知られている。**(CHSH 不等式, Bell 不等式)**
- ただし、乱数を発生させる人が、各試行で Alice と Bob が何を計算するかあらかじめ知っているときは実現可能。



Alice は Z, X 基底でランダムに測定

Bob は $(Z \pm X)/\sqrt{2}$ 基底でランダムに測定

Charlie が Alice と Bob の行動を事前に知っていれば再現可能

Charlie

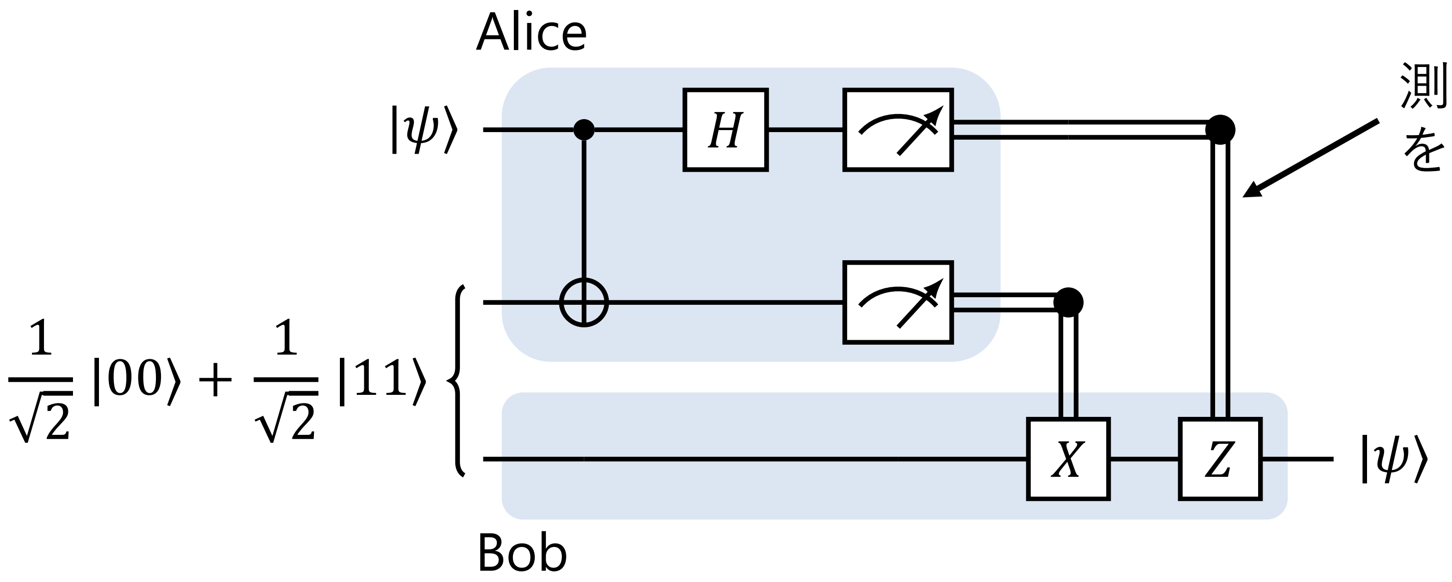
古典乱数

Alice は送られてきた乱数に対してブール関数 A_1, A_2 を計算する。

Bob は送られてきた乱数に対してブール関数 B_1, B_2 を計算する。

エンタングルメントの応用例

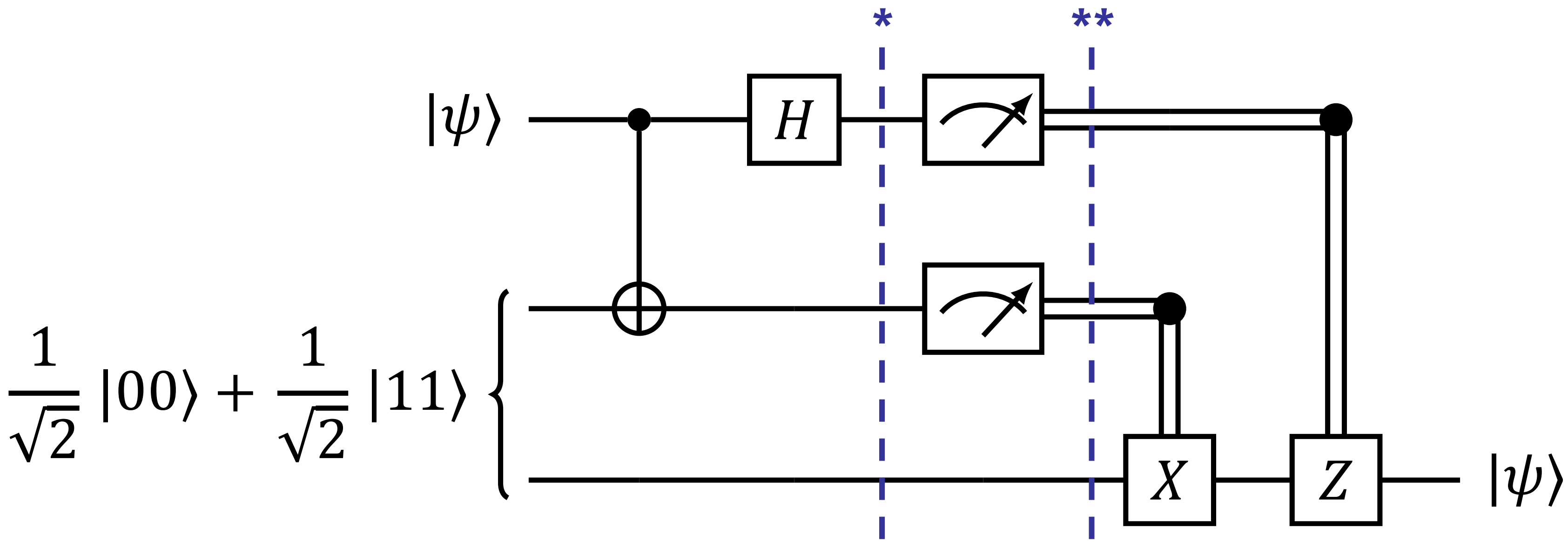
- **量子テレポーテーション：**
ベル状態を使って、1 量子ビットの状態 $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ を送信する。



測定で 1 が出たときだけゲート
を作用させるという記号

エンタングルメントの応用例

➢ **量子テレポーテーション：**
ベル状態を使って、1 量子ビットの状態 $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ を送信する。



演習（レポート）：

(1) * での量子状態を計算してください。

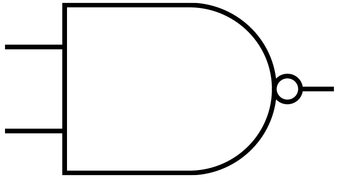
(2) Bob の量子ビットから $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ が出力されることを確認してください。

(3) (発展) ** 時点で(Alice の測定結果を知らない状態で)、Bob が適当な物理量 O の期待値を測定すると、その値は α, β に依存しないことを示してください。

古典計算 $\subseteq$ 量子計算

古典計算

- 古典計算ができるべきタスク：
  ビット列 $b_1 \cdots b_n$ を受け取って、ビット b を返す関数 $f(b_1 \cdots b_n)$ を計算する。
- NAND 演算は**万能** (NAND の組み合わせでどんな関数 $f(b_1 \cdots b_n)$ も計算できる。)



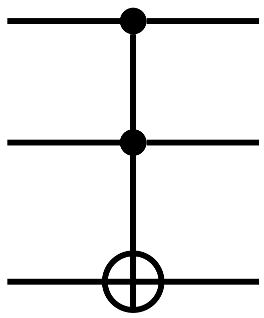
NAND 回路記号

入出力関係

入力	出力
00	1
01	1
10	1
11	0

可逆古典計算

- 古典計算ができるべきタスク：
  ビット列 $b_1 \cdots b_n$ を受け取って、ビット b を返す関数 $f(b_1 \cdots b_n)$ を計算する。
- 可逆な演算のみを使うのが、**可逆古典計算。**
- **トフォリゲート（CCNOTゲート）が万能。**(NAND を実現できる。)



トフォリゲート回路記号

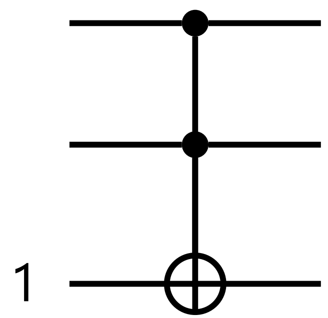
入出力関係

入力	出力
000	000
001	001
010	010
011	011
100	100
101	101
110	**111**
111	**110**

入出力が1対1対応
→可逆

トフォリゲートで NAND を作る

- ターゲットビットを 1 に初期化しておくと、NAND が実現できる。
  → 多数のビットが必要になるが、どんなブール関数でも計算可能。

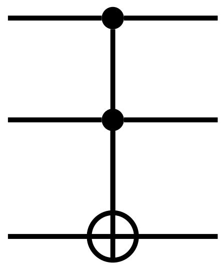


制御ビットの入力	ターゲットビットの出力
00	1
01	1
10	1
11	0

演習 (レポート)：NOT, CNOT, Toffoli ゲートを使って、半加算器で必要となる論理式 $S = b_1 \oplus b_2$, $C = b_1 \cdot b_2$ を構成してください。

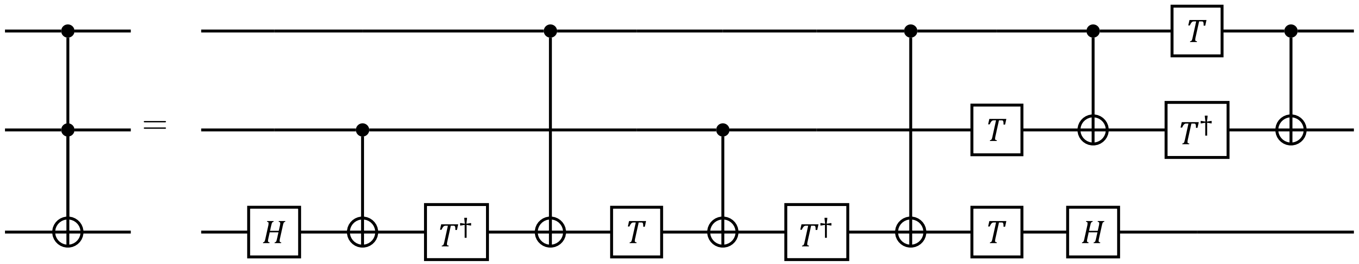
古典計算 ⊆ 量子計算

- トフォリゲートはユニタリーとして実現できる。



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

- 実は CNOT + 1量子ビットゲートによって次のように分解可能。



From Nielsen-Chuang textbook.

量子計算は任意の古典計算を実行可能

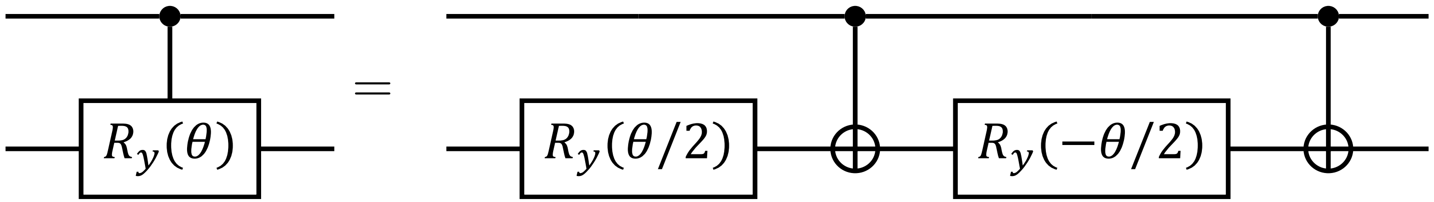
CNOT + 1 量子ビットゲートで量子ゲートを合成する

制御回転ゲート

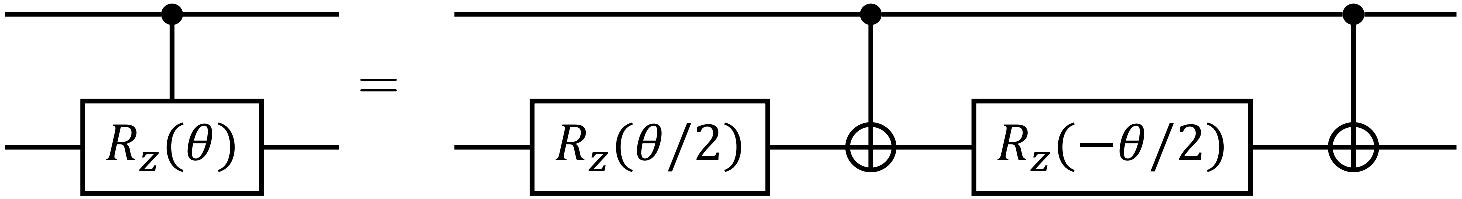
- Controlled-$R_y(\theta)$ ゲートを CNOT と 1 量子ビット回転で作りたい。
- 次の等式を使う。

$$XYX = -Y \Rightarrow XR_y(\theta)X = R_y(-\theta)$$

- 以下のように分解できることがわかる。



- $XZX = -Z, ZXZ = -X$ を使えば同様に Controlled-$R_z(\theta), R_x(\theta)$ゲートも分解可能。



演習（レポート）： Controlled-$R_x(\theta)$ ゲートを CNOT と 1 量子ビット回転に分解してください

制御 U ゲート

- 1 量子ビットの任意のユニタリ U について、Controlled-U ゲートを CNOT と 1 量子ビット回転で作りたい。
- 1 量子ビットのユニタリは次のように分解できるのだった。

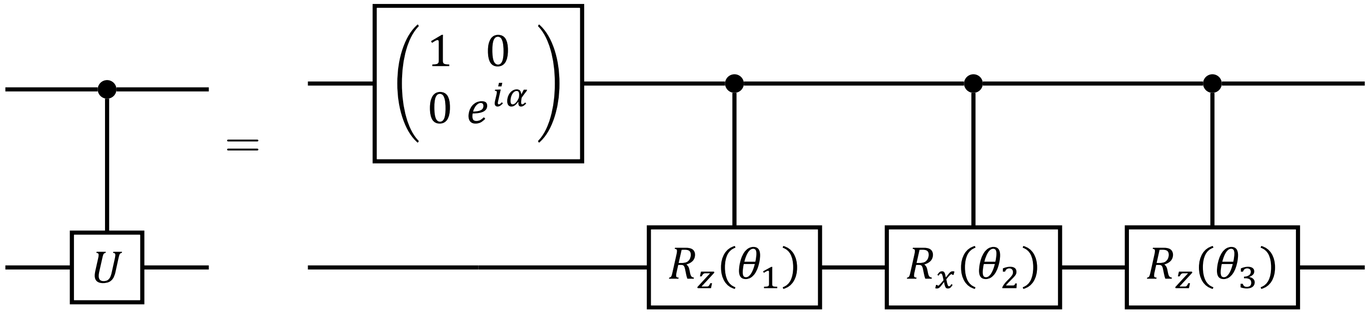
$$U = e^{i\alpha} R_z(\theta_3) R_x(\theta_2) R_z(\theta_1)$$

- それぞれを controlled 化すれば、controlled-U が得られる。
- このとき $e^{i\alpha}$ は 1 のときのみターゲット量子ビットに位相 $e^{i\alpha}$ をつけるゲート

$$|0\rangle_{\mathrm{ctrl}}|\psi\rangle_{\mathrm{targ}} \to |0\rangle_{\mathrm{ctrl}}|\psi\rangle_{\mathrm{targ}}, \qquad |1\rangle_{\mathrm{ctrl}}|\psi\rangle_{\mathrm{targ}} \to |1\rangle_{\mathrm{ctrl}} e^{i\alpha}|\psi\rangle_{\mathrm{targ}} = e^{i\alpha}|1\rangle_{\mathrm{ctrl}}|\psi\rangle_{\mathrm{targ}}$$

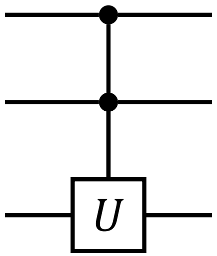
になる。これは制御量子ビットの $|1\rangle$ に位相をつけるゲートと等価。

- したがって、 Controlled-U は次のように分解できる。

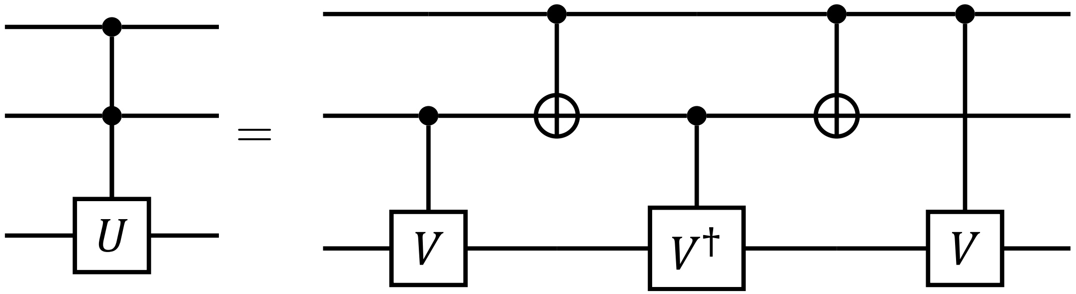


制御ビットが２つある量子ゲート

- 図のような C^2-U ゲートを作りたい。



- Controlled-R_y を作ったときと同じアイデアで、$V^2 = U$ となるような V を使って次のように分解する。

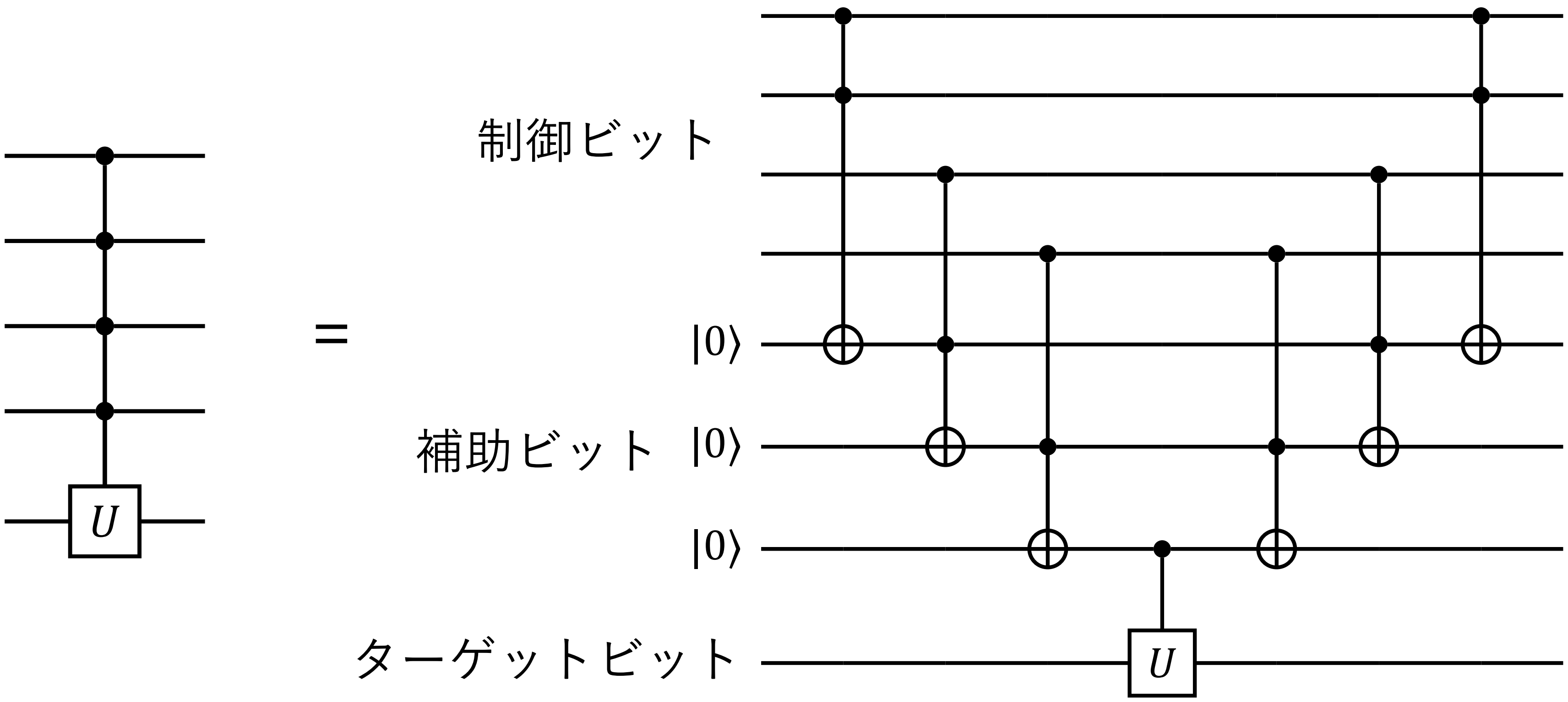


トフォリゲートも同様に分解可能。

演習： 制御ビットに 00, 01, 10, 11 を入力した場合を考えて、上記の分解が正しいことを確かめてください。

制御ビットが n 個ある量子ゲート

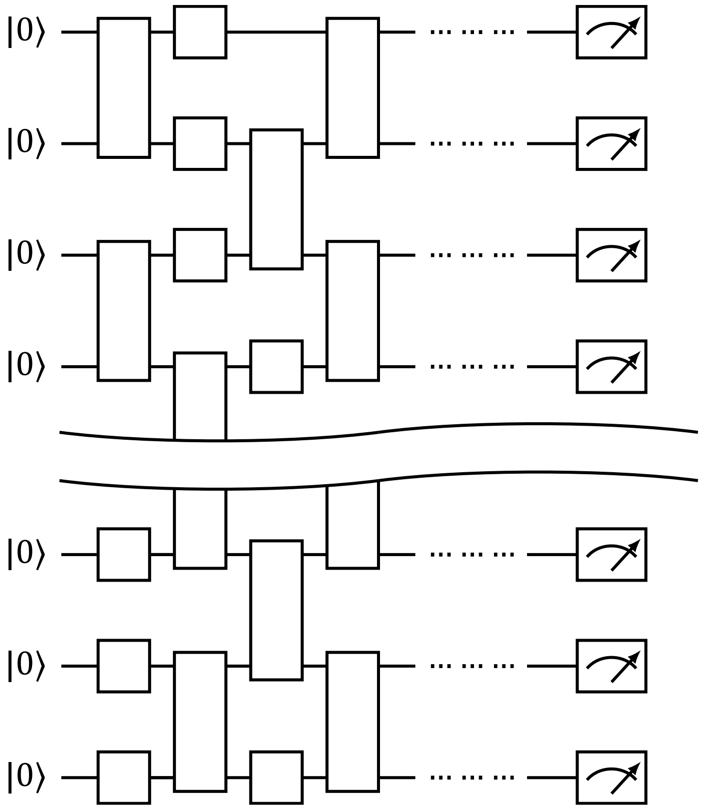
➢ 例えば 4 つの制御ビットを持つ制御 U ゲートは以下のように作れる。



万能量子計算

万能量子計算機

- n 量子ビットに対して、任意の $2^n \times 2^n$ ユニタリー行列を作用させられる計算機を、**万能量子計算機 (universal quantum computer)**と呼ぶ。
- 任意の 1 量子ビットゲートと CNOT を持つ量子コンピュータは万能になる。



任意の n qubit ユニタリの構成 (1)

- 2^n 次元の空間のうち、2 次元の部分空間にのみ作用するユニタリ (two-level unitary gates) は万能
- 3 × 3 行列の例を見るとわかる。以下のユニタリを作りたいとする。

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{pmatrix}$$

- $U_3U_2U_1U = I$ となる two level unitaries U_1, U_2, U_3を以下の手順で見つける。
  - $U_1 = \begin{pmatrix} u_{11} & u_{12} & 0 \\ u_{21} & u_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix}$ をかけて、

$$U_1U = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{pmatrix}$$

任意の n qubit ユニタリの構成

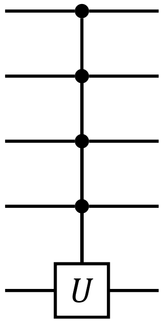
- 2^n 次元の空間のうち、2 次元の部分空間にのみ作用するユニタリ (two-level unitary gates) は**万能**
- 3 × 3 行列の例を見るとわかる。以下のユニタリを作りたいとする。

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{pmatrix}$$

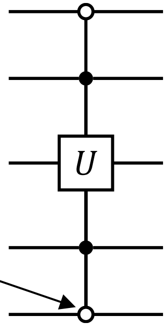
- $U_3U_2U_1U = I$ となる two level unitaries U_1, U_2, U_3を以下の手順で見つける。
  - $U_1 = \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{pmatrix}$をかけて、$U_1U = \begin{pmatrix} u'_{11} & u'_{12} & u'_{13} \\ 0 & u'_{22} & u'_{23} \\ u'_{31} & u'_{32} & u'_{33} \end{pmatrix}$の形にできる。
  - $U_2 = \begin{pmatrix} a' & 0 & b' \\ 0 & 1 & 0 \\ c' & 0 & d' \end{pmatrix}$をかけて、$U_2U_1U = \begin{pmatrix} u''_{11} & u''_{12} & u''_{13} \\ 0 & u''_{22} & u''_{23} \\ 0 & u''_{32} & u''_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & u''_{22} & u''_{23} \\ 0 & u''_{32} & u''_{33} \end{pmatrix}$の形にできる。
  - $U_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & a'' & b'' \\ 0 & c'' & d'' \end{pmatrix}$をかけて、 $U_3U_2U_1U = I$

Two-level ユニタリの合成 (1)

- 2^n 次元の空間のうち、2つのビット列 $|b_1 b_2 \cdots b_n\rangle$ と $|b_1' b_2' \cdots b_n'\rangle$ で張られる空間でのユニタリはどのように作れるか？
- 複数制御ビットを持つ Controlled-U ゲートは $|11110\rangle$ と $|11111\rangle$ の空間に U をかける。



- 一般化すると、1ビットだけ異なるビット列で張られる部分空間に U を作用させられる。



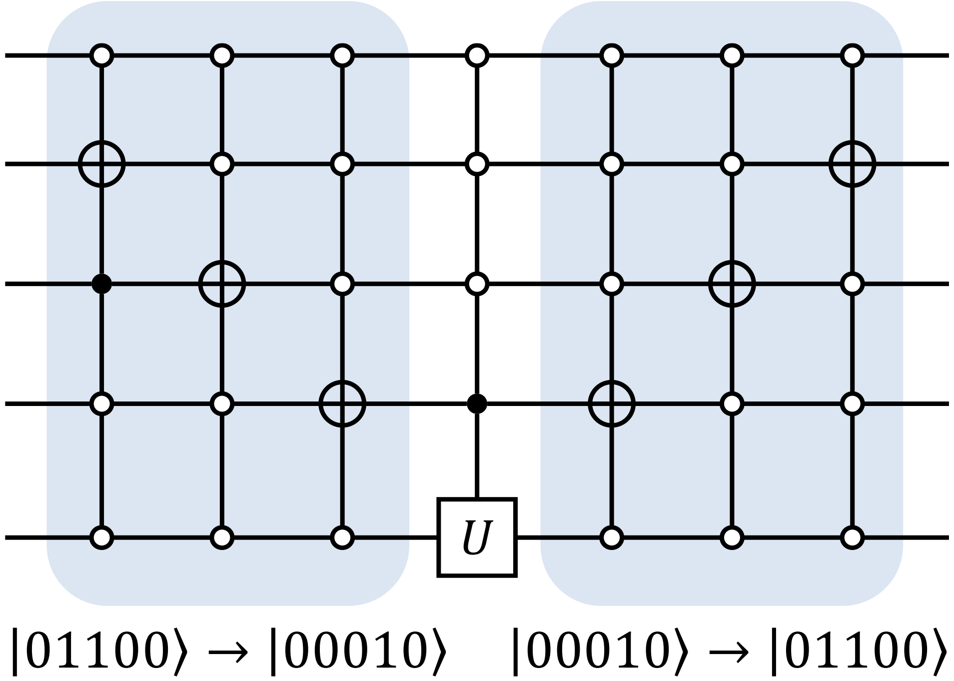
：$|01010\rangle$ と $|01110\rangle$ に U をかける

白丸は $|0\rangle$ のときに U を作用させる制御を示す。

Two-level ユニタリの合成 (2)

- $|b_1 b_2 \cdots b_n\rangle$ と $|b'_1 b'_2 \cdots b'_n\rangle$ の間に U をかけたいときには、以下のようにする。

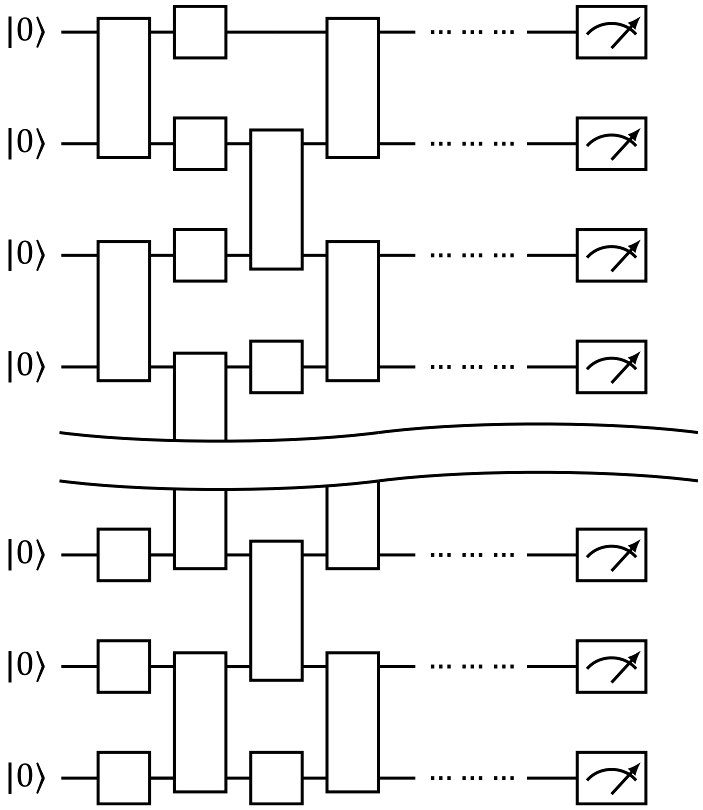
例： $|01100\rangle$ と $|00011\rangle$ に U をかける。



演習： n qubit に対する two-level ユニタリには最大何個の複数制御 CNOT ゲートが必要でしょうか？

万能量子計算機

- 複数制御 U ゲートは 1 量子ビットゲートと CNOT で作れたので....
- **任意の 1 量子ビットゲートと CNOT を持つ量子コンピュータは万能になる。**



量子アルゴリズム

量子コンピュータの応用先

- 量子コンピュータが古典に対して優位性を持つ代表的な応用先/アルゴリズムは、次の 4 つ。
  - 量子力学のシミュレーション
  - 素因数分解
  - データベース探索/組み合わせ最適化
  - 行列計算

- 「量子計算が古典計算に対して優位」
  ⇒ (量子アルゴリズムの計算量オーダー) < (古典ベストアルゴリズムの計算量オーダー)

計算量オーダー

- あるアルゴリズムが必要とする計算量は、問題の入力サイズ n の関数になる。
- 入力サイズが大きい極限での漸近的な計算量を表すのには、オーダー記号が便利。
  例：
  - $12n^2+5n+1=O(n^2)$
  - $2^n+n=O(2^n)$
  - $\log n+\log\log n=O(\log n)$

  $O(n\log n)=\tilde{O}(n)$ と書くことも。
- 多項式オーダーを $\mathrm{poly}\,n$ とかく。($\mathrm{poly}\,n=O\left(n^k\right)$ for all k).
- 一般に、「**ある問題が効率的に計算できる。**」⇔「 $\mathrm{poly}\,n$ **時間のアルゴリズムが存在する。**」
- 古典コンピュータが効率的に計算可能な (決定) 問題のクラス：**P**
- 量子コンピュータが効率的に計算可能な (決定) 問題のクラス：**BQP**

演習：$n\times n$ の 2 つの行列の積を計算するには、どのくらいのオーダーの計算が必要でしょうか？

量子アルゴリズムの計算量

- 量子コンピュータが古典に対して優位性を持つ代表的な応用先の計算量
  - n 個の量子力学的スピン 1/2 の時間発展 [S. Lloyd, Science, **273**, 1073-1078 (1996)]

    $$O(2^n) \to \mathrm{poly}\, n$$

  - n bit 整数の素因数分解 [P. W. Shor, Proceedings 35th Annual Symposium on Foundations of Computer Science, 124-134 (1994)]

    $$O\left(e^{1.9 n^{1/3} (\log n)^{2/3}}\right) \to O(n^2 \log n \log\log n)$$

  - N 個のデータを持つデータベースの検索 [L. K. Grover, Proceedings, 28th Annual ACM Symposium on the Theory of Computing, 212-219, (1996)]

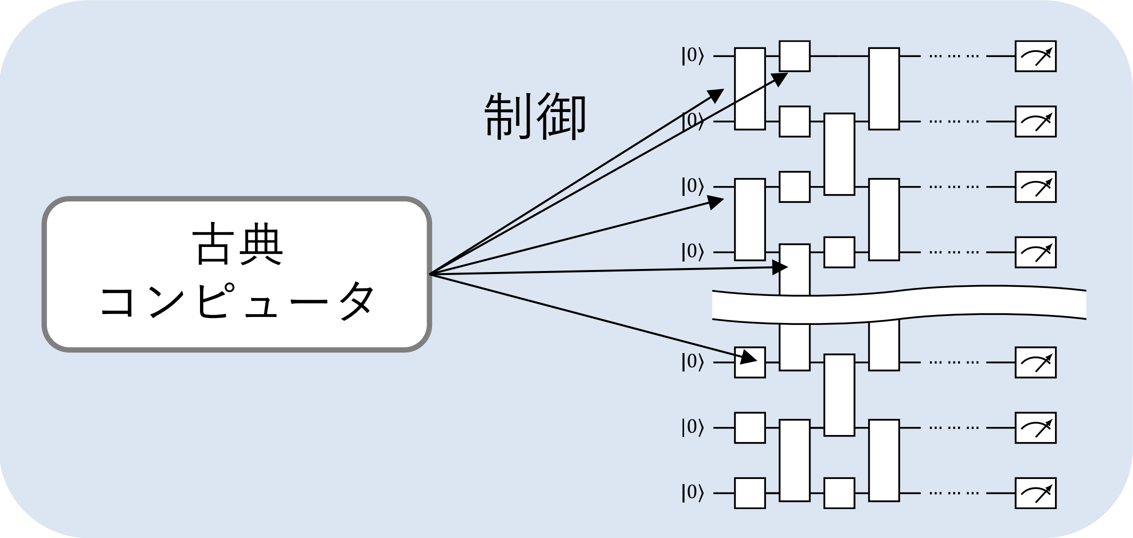
    $$O(N) \to O(\sqrt{N})$$

  - N 次元疎行列の逆行列計算 (疎性 s, 条件数 κ, 精度 $1/\epsilon$) [A. Harrow et al., PRL, **103**, 150502 (2009)]

    $$O(Ns\sqrt{\kappa}\log 1/\epsilon) \to \tilde{O}(\log N\, s^2\kappa^2/\epsilon)$$

計算量オーダーの罠

- **O 記法で小さい計算量オーダーのアルゴリズムが見つかったからといって、それが実用上高速であるとは限らない。**
- Q： $n \times n$ の2つの行列の積の計算を
$$O(n^{2.3728596})$$
で実行できるアルゴリズム [arXiv: 2010.05846] が知られているが、使われていない。なぜ？
- 実用的には O 記法で隠れてしまう定数係数が非常に重要。
- 量子コンピュータでは？
- **量子コンピュータのクロック**
<<<< 古典コンピュータのクロック
- 量子コンピュータが実用的な計算ができるかは定数係数まで考える必要あり。



量子コンピュータの制御は古典コンピュータで行う。

量子コンピュータの応用先

- 量子コンピュータが古典に対して優位性を持つ代表的な応用先/アルゴリズムは、次の 4 つ。
  - 量子力学のシミュレーション
  - 素因数分解
  - データベース探索/組み合わせ最適化
  - 行列計算

量子シミュレーション on 量子コンピュータ

物質シミュレーションの重要性

- **ハーバーボッシュ法** = 高温・高圧でアンモニアを合成する手法
- アンモニアの8割は化学肥料へ。**「空気からパンを作る」手法**
- アンモニアの合成に全世界の **1～2% のエネルギー**を消費している
  （イギリス1国のエネルギー消費に相当）

物質シミュレーションの重要性 (2)

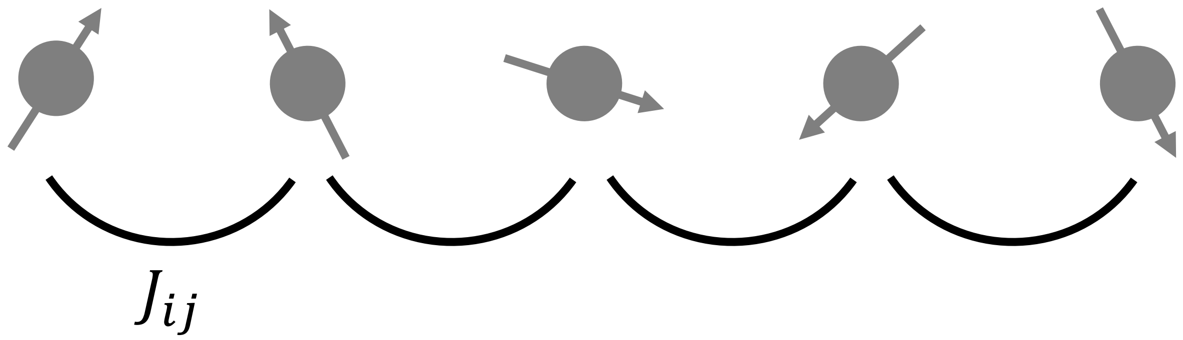
- マメ科の根の窒素固定酵素 FeMocoは常温・常圧で空気中の窒素を集める。
- FeMocoの原理を量子コンピュータを使って解析 → 大量合成に耐えられる触媒を設計できるかも？

ハイゼンベルグ模型のエネルギースペクトル

- 簡単な例として、n 個のスピン S=1/2 ハイゼンベルグ模型を考える。

$$H = \sum_{i,j} J_{ij}\left(X_i X_j + Y_i Y_j + Z_i Z_j\right)$$

- H のエネルギースペクトルを知りたい。
- 古典コンピュータで計算するなら、$2^n \times 2^n$ 行列の対角化が必要となる。



時間発展からエネルギースペクトルを知る

- H の固有値・固有ベクトルをそれぞれ $E_i, |E_i\rangle$ とする。
- アルゴリズム：
  1. 適当な初期状態 $|\psi(0)\rangle$ を決める。
  $$|\psi_0\rangle = \sum_i a_i |E_i\rangle$$
  2. e^{-iHt} を作用させる。
  $$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$$
  3. 内積 $g(t) = \langle\psi(t)|\psi(0)\rangle$ を評価する。
  4. $g(t)$ をフーリエ変換する。
  5. ピークを読み取る。

古典コンピュータでは
$O(2^n)$ 時間かかる

演習(レポート)：$g(t) = \sum_i |a_i|^2 e^{iE_i t}$ となることを示してください。

量子シミュレーションのための量子回路

- e^{iHt} を実行できる量子回路を作りたい。
- トロッター展開

$$e^{(A+B)t} \approx \left(e^{At/m}e^{Bt/m}\right)^m$$

により簡単なユニタリーに分解する。

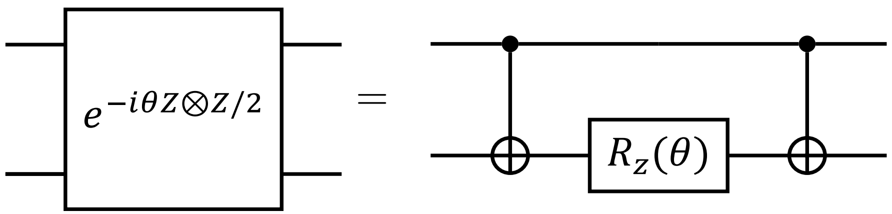
- H は XX, YY, ZZ の和だったので、

$$e^{iX\otimes X\Delta t}, e^{iY\otimes Y\Delta t}, e^{iZ\otimes Z\Delta t}$$

を作れば OK.

- 有用な公式：

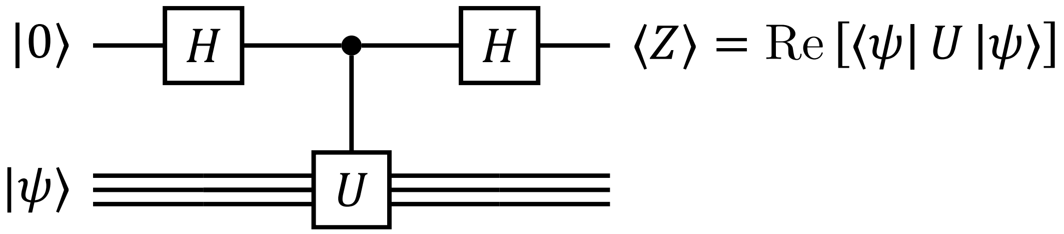
$$\mathrm{CNOT}(I \otimes Z)\mathrm{CNOT} = Z \otimes Z$$

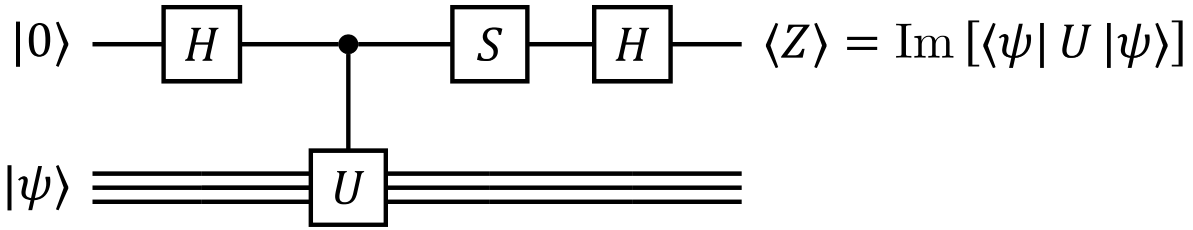


演習：$\mathrm{CNOT}(I \otimes Z)\mathrm{CNOT} = Z \otimes Z$ を示してください。また、$e^{i\theta X\otimes X}, e^{i\theta Y\otimes Y}$ を実現する量子回路を書いてください。

内積を計算する量子回路

- $\langle\psi(t)|\psi(0)\rangle = \langle\psi(0)|e^{iHt}|\psi(0)\rangle$ のような内積の計算には、**アダマールテスト**と呼ばれる量子回路が使える。





演習：上記の回路の出力が $\langle Z\rangle = \mathrm{Re}[\langle\psi|U|\psi\rangle]$ になることを示してください。

制御 U ゲートは、制御ビットの位相に固有値の情報を書き込む。

時間発展からエネルギースペクトルを知る

- アルゴリズム：
  1. 適当な初期状態 $|\psi(0)\rangle$ を決める。
  2. e^{-iHt} を作用させる。
  $$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$$
  3. 内積 $g(t) = \langle\psi(t)|\psi(0)\rangle$ for $0 \leq t \leq T$ を評価する。

  量子コンピュータなら$\mathrm{poly}\, n$

  4. $g(t)$ をフーリエ変換する。 → 古典コンピュータ上で。FFT はデータ点数 M に対して $\mathrm{poly}\, M$。
  5. ピークを読み取る。
- フーリエ変換のピークの幅は一般に $1/T$
  → エネルギーの推定誤差を ϵ にしたければ、$T = O(1/\epsilon)$ と取ればよい。

時間発展からエネルギースペクトルを知る

- アルゴリズム：
  1. 適当な初期状態 $|\psi(0)\rangle$ を決める。
  2. e^{-iHt} を作用させる。
     $$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$$
  3. 内積 $g(t) = \langle\psi(t)|\psi(0)\rangle$ for $0 \leq t \leq T$ を評価する。

  （1.〜3.）量子コンピュータなら$\mathrm{poly}\, n$

  4. **$g(t)$ をフーリエ変換する。** → 古典コンピュータ上で。FFT はデータ点数 M に対して $\mathrm{poly}\, M$。
  5. ピークを読み取る。
- フーリエ変換も量子でできないか？

量子フーリエ変換と
量子位相推定アルゴリズム

量子フーリエ変換

- データ $\{x_k\}$ の離散フーリエ変換 $\{y_k\}$ は以下のように定義される。

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \mathrm{e}^{i2\pi kj/N} x_j$$

- 量子状態の振幅にデータを書き込むことを考え、

$$|\boldsymbol{x}\rangle = \sum_{j=0}^{N-1} x_j |j\rangle$$

$$|\boldsymbol{y}\rangle = \sum_{k=0}^{N-1} y_k |k\rangle = \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} \mathrm{e}^{i2\pi kj/N} x_j |k\rangle$$

を定義する。

※ j, k は $j_n \cdots j_2 j_1, k_n \cdots k_2 k_1$ というビット列 (2進整数)

※ $N = 2^n$

- $|\boldsymbol{x}\rangle \to |\boldsymbol{y}\rangle$ の変換には、

$$U|j\rangle = \sum_{k=0}^{N-1} e^{i2\pi kj/N} |k\rangle$$

という変換が実現できれば良い。

量子フーリエ変換の量子回路に向けて

- $U|j\rangle = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{i2\pi kj/N}|k\rangle$ という変換の右辺は、要するに
  「周波数 j/N で振動する波を、2 進整数 $k = k_n \cdots k_2 k_1$ 上に準備しなさい」
  ということ。
- 最下位ビット k_1 が $0 \to 1$ になったら、整数 k としては 1 進むので、位相は $e^{i2\pi j/N}$ 進める。
- 2番目のビット k_2 が $0 \to 1$ になったら、整数 k としては 2 進むので、位相は $e^{i4\pi j/N}$ 進める。
- 3番目のビット k_3 が $0 \to 1$ になったら、整数 k としては 4 進むので、位相は $e^{i8\pi j/N}$ 進める。
- …
- 最上位ビットが k_n が $0 \to 1$ になったら、整数 k としては 2^{n-1} 進むので、位相は $e^{i2^n\pi j/N}$ 進める。
- つまり

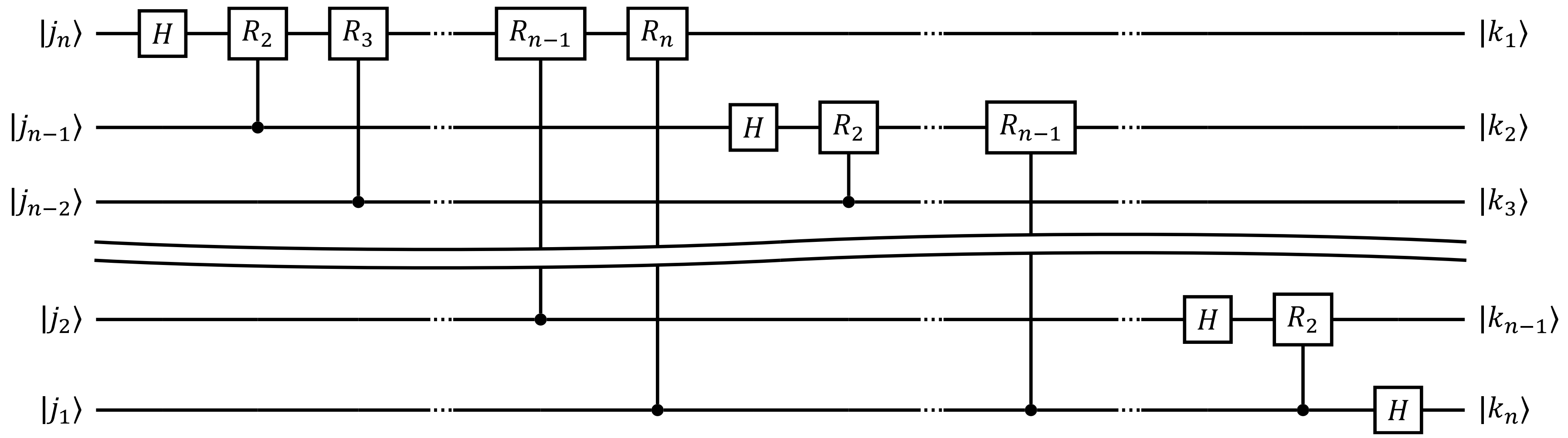
$$\sum_{k=0}^{N-1} e^{i2\pi kj/N}|k\rangle = \left(|0\rangle + e^{i2^n\pi j/N}|1\rangle\right)\cdots\left(|0\rangle + e^{i4\pi j/N}|1\rangle\right)\left(|0\rangle + e^{i2\pi j/N}|1\rangle\right)$$

量子フーリエ変換の量子回路

- $j/N = j/2^n = 0.j_n j_{n-1} \cdots j_2 j_1$ (2進小数表示) であることを使うと

$$\sum_{k=0}^{N-1} e^{i2\pi kj/N}|k\rangle$$
$$= \left(|0\rangle + e^{i2\pi 0.j_1}|1\rangle\right)\left(|0\rangle + e^{i2\pi 0.j_1 j_2}|1\rangle\right)\cdots\left(|0\rangle + e^{i2\pi 0.j_{n-1}\cdots j_2 j_1}|1\rangle\right)\left(|0\rangle + e^{i2\pi 0.j_n\cdots j_2 j_1}|1\rangle\right)$$

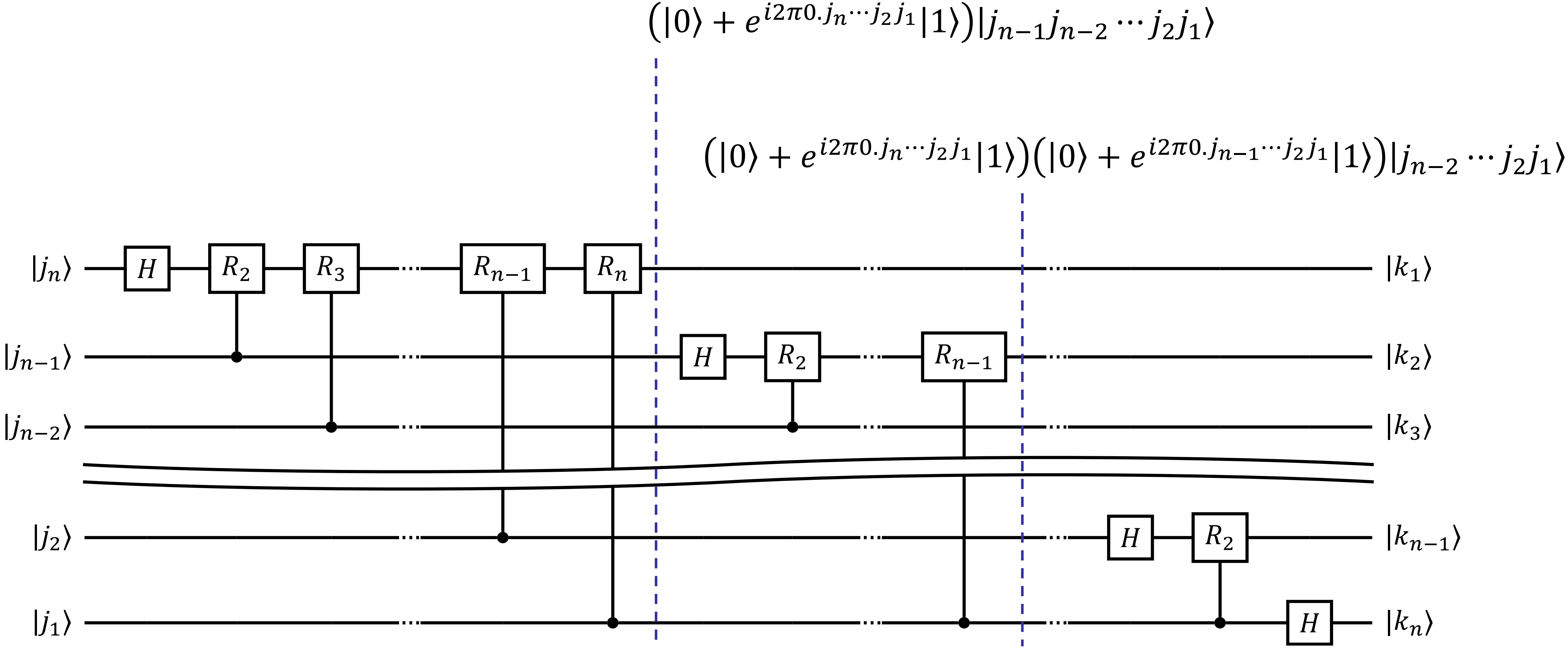
- この表現をもとに、 $U|j\rangle = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{i2\pi kj/N}|k\rangle$ を実現するのが以下の回路。



※ $R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{i2\pi/2^k} \end{pmatrix}$

$O(n^2) = O\left((\log N)^2\right)$ でフーリエ変換を実現！ (古典 FFT は $O(N\log N)$)

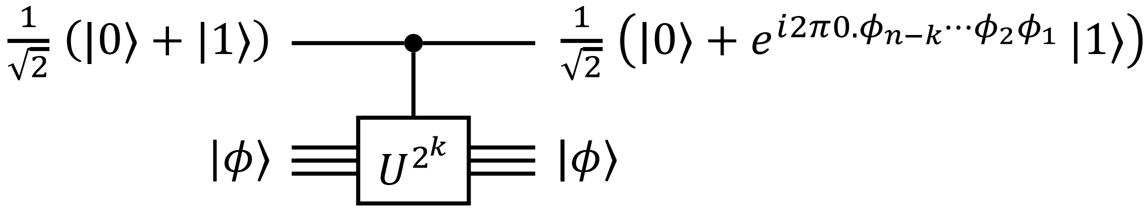
量子フーリエ変換の量子回路



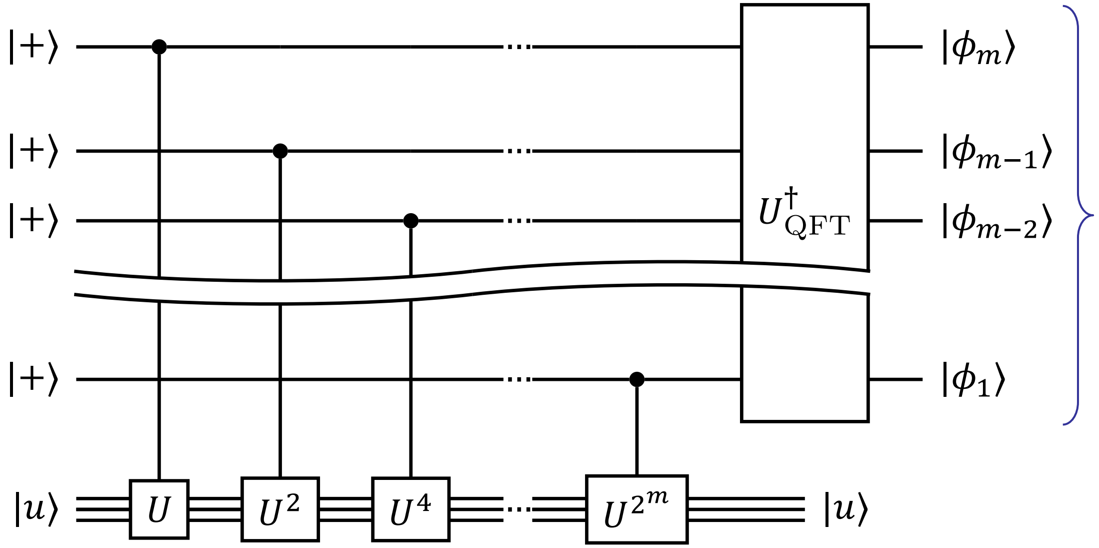
量子位相推定アルゴリズム

- ユニタリー U の固有値・固有ベクトルを $e^{i2\pi\phi} = e^{i2\pi 0.\phi_m \cdots \phi_2 \phi_1}$ ・$|u\rangle$ とする。

演習：以下の回路の動作を計算してください。



- **量子位相推定アルゴリズム**

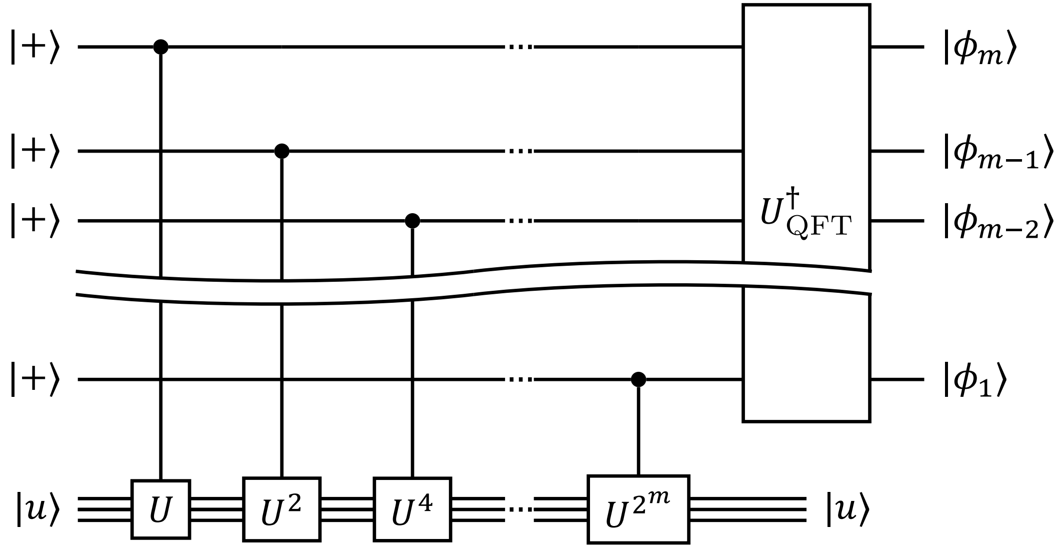


ϕ の2進数表示
今後 $|b(\phi)\rangle$ と書く。

- 量子でフーリエ変換まで完結できた！

量子位相推定アルゴリズムの性質

- 入力が固有ベクトル $|u_i\rangle$ (固有値 $\mathrm{e}^{i2\pi\phi_i}$) の重ね合わせ状態 $|\psi\rangle = \sum_i a_i\,|u_i\rangle$ のとき、出力は
$$\sum_i a_i|u_i\rangle|b(\phi_i)\rangle$$
となる。
- $|b(\phi)\rangle$ のレジスタを測定すると、$|a_i|^2$ の確率で ϕ_i の2進表現が得られる。
- ϕ_i が測定されたとき、$|\psi\rangle$ は $|u_i\rangle$ に射影される。



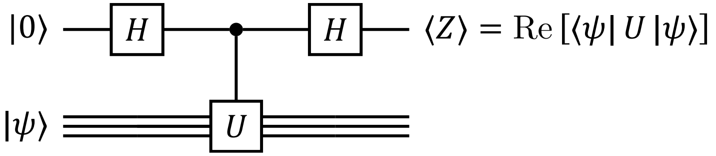
量子シミュレーションへの応用

- ハミルトニアン H の固有値 E_i・固有ベクトル $|E_i\rangle$ を知りたい。
- $e^{i2\pi H}$ の位相推定を、入力 $|\psi\rangle = \sum_i a_i|E_i\rangle$ に対して行えば、出力は
$$\sum_i a_i|E_i\rangle|b(E_i)\rangle$$
となる。
- $|b(E_i)\rangle$ のレジスタを測定すると、$|a_i|^2$ の確率で E_i の2進表現が得られる。
- E_i が測定されたとき、$|\psi\rangle$ は $|E_i\rangle$ に射影される。

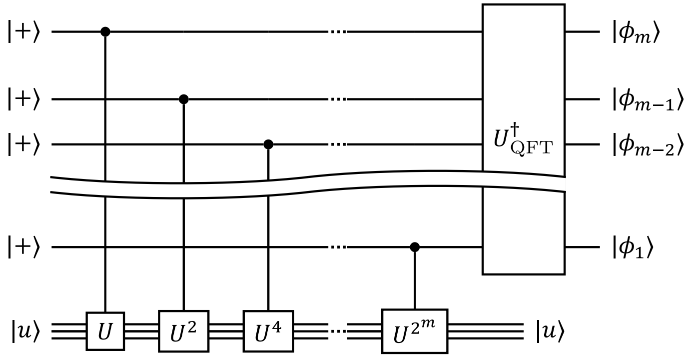
位相推定によって、$E_i, |E_i\rangle$ をサンプリングできる。

いったんまとめ

- 量子コンピュータは、トロッター展開によって時間発展演算子 e^{iHt} を作用させられる。
- アダマールテストは、状態間の内積を計算するための量子回路。



- 量子フーリエ変換を使って、固有値・固有ベクトルをサンプリングできるのが、量子位相推定。



- 応用先として、ハミルトニアン H の固有値計算が考えられる。

演習(レポート)：位相推定による固有値推定において、U を使う回数と推定精度はどのような関係になっているでしょうか？

量子コンピュータの応用先

- 量子コンピュータが古典に対して優位性を持つ代表的な応用先/アルゴリズムは、次の 4 つ。
  - 量子力学のシミュレーション
  - 素因数分解
  - データベース探索/組み合わせ最適化
  - 行列計算

Shor の素因数分解アルゴリズム

素因数分解 → 位数計算

- 整数 M の素因数分解をしたい。
- この問題は、ランダムに選んだ整数 $a < M$ に対して、
$$a^r \equiv 1 \bmod M$$
となる整数 r (**位数**) を計算できれば解ける。

簡単な説明：

- r が偶数なら、上の式は
$$\left(a^{r/2}+1\right)\left(a^{r/2}-1\right) \equiv 0 \bmod M$$
を導く。
- このとき $a^{r/2}+1$ と $a^{r/2}-1$ のどちらかは、M と同じ約数を持つはず。
- $a^{r/2}+1$ と M の最大公約数、 $a^{r/2}-1$ と M の最大公約数を計算すれば、M の約数が求まる。
- 最大公約数の計算はユークリッドの互除法によって $(\log M)^2$ 時間で可能。
- (天下りですが) ランダムに持ってきた a について r が偶数である確率は $O(1)$。

最後の定理は https://people.eecs.berkeley.edu/~vazirani/f04quantum/notes/lec9.pdf の Lemma 9.3 などを参照。

位数計算のアルゴリズム

- $U_a|x\rangle = |xa \bmod M\rangle$ となるような演算 U_a を考える。
  - これは単なる古典計算なので、トフォリゲートの組み合わせで実装可能
- U_a の固有値は $u_s = e^{i2\pi s/r}$ (s：整数, $0 \le s \le r-1$)。 対応する固有ベクトル u_s は

$$|u_s\rangle = \frac{1}{\sqrt{r}}\left(|1\rangle + u_s|a\rangle + u_s^2|a^2 \bmod M\rangle + \cdots + u_s^{r-1}|a^{r-1} \bmod M\rangle\right)$$

- U_a の固有値 u_s を位相推定によって求めれば r が決まる。
- でも $|u_s\rangle$ を準備できないとそれもできない？しかし上の式を見れば明らかに、

$$\sum_{s=0}^{r-1}|u_s\rangle = |1\rangle$$

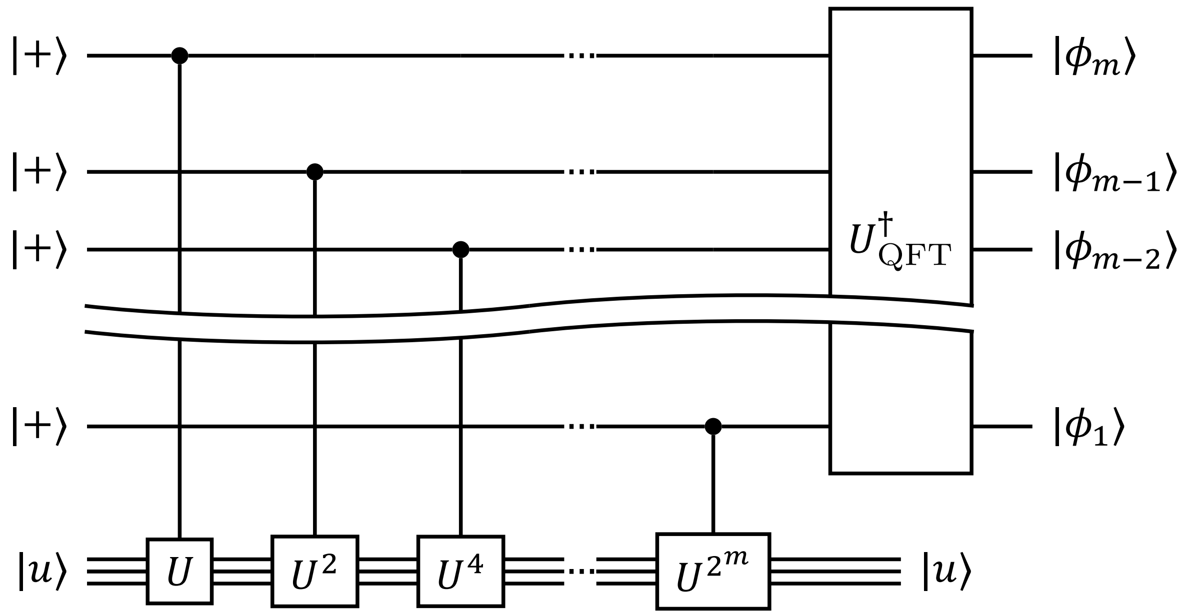
である。

$|1\rangle$ を位相推定に入力すれば、s/r が求められる！

※ 正確には、 s/r を近似するビット列が出力されるので、連分数展開で求める必要あり。

位相推定アルゴリズム

- m bit の精度の位相推定アルゴリズムには、指数回の演算 (U^{2^m}) が必要？



- **$U_a|x\rangle = |xa \bmod M\rangle$ に関しては、$U_a^{2^m}$ を簡単に作用させることが可能！**
- なぜなら $U_a^{2^m}|x\rangle = \left|xa^{2^m} \bmod M\right\rangle = U_{a^{2^m} \bmod M}|x\rangle$ だから。先に $a^{2^m} \bmod M$ を古典コンピュータ上で計算して、 $U_{a^{2^m} \bmod M}$ に必要な回路を設計すればよい。
- $a^{2^m} \bmod M$ は、$a \to a^2 \to a^4 \to \cdots \to a^{2^m}$ と計算すれば $O(m)$ で計算できる。

素因数分解アルゴリズム

タスク： n bit 整数 M の素因数分解

1. 整数 $1 < a < M$ をランダムに選ぶ。
2. 位相推定によって $a^r \equiv 1 \bmod M$ となるような整数 r を計算する。
3. r が奇数 → 1 へ戻る。
   r が偶数 → $a^{r/2} - 1$ ・ $a^{r/2} + 1$ と M の最大公約数 c を求める。
4. $M = cM'$ と因数分解する。
5. (必要があれば) c, M' をさらに因数分解する

量子コンピュータの応用先

- 量子コンピュータが古典に対して優位性を持つ代表的な応用先/アルゴリズムは、次の 4 つ。
  - 量子力学のシミュレーション
  - 素因数分解
  - データベース探索/組み合わせ最適化
  - 行列計算

逆行列計算（HHL アルゴリズム）

逆行列計算

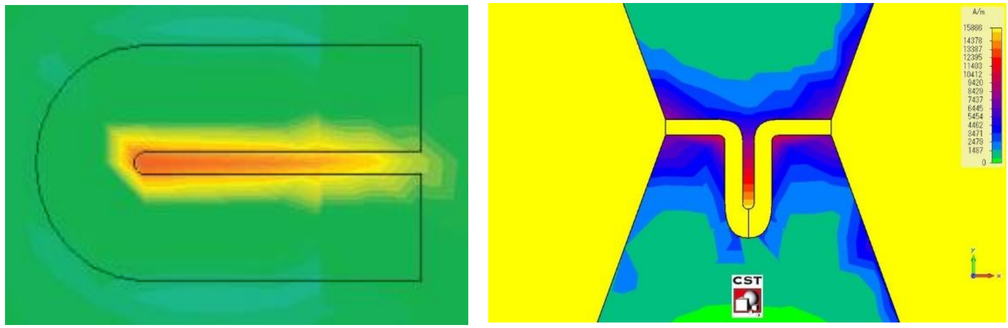
- 次の方程式を解きたい。

$$A\boldsymbol{x} = \boldsymbol{b}$$

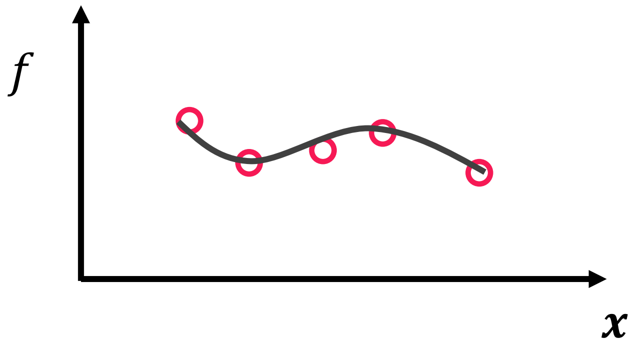
- 解は

$$\boldsymbol{x} = A^{-1}\boldsymbol{b}$$

- 実はこの計算タスクも（ある意味において）量子コンピュータで加速できる。
- 連立方程式は至るところで使われており、応用先は数え切れない。



電磁場のシミュレーション
(微分方程式を有限要素法で解く。)



機械学習

量子で線形方程式を解く

- ベクトルの振幅エンコーディング

$$|\boldsymbol{b}\rangle = \sum_{j=0}^{N-1} b_j |j\rangle$$

- **量子線形システム問題 (Quantum linear systems problem)** を次のような計算タスクとして定義。
  - 入力：行列 A と量子状態 $|\boldsymbol{b}\rangle$
  - 出力：ベクトル $A^{-1}\boldsymbol{b}$ の振幅エンコーディング $\sum_j (A^{-1}\boldsymbol{b})_j |j\rangle$
- **HHL (Harrow-Hassidim-Lloyd)** アルゴリズムはこの問題を $\mathrm{poly}\log N$ 時間で解く。
- 注意しなければならないこと：
  - ✓ $|\boldsymbol{b}\rangle = \sum_j b_j |j\rangle$ を準備する方法が自明でない。
  - ✓ 出力が $\sum_j (A^{-1}\boldsymbol{b})_j |j\rangle$ という量子状態である。
    → ベクトル $A^{-1}\boldsymbol{b}$ の各成分を読み出すには、結局 $O(N)$ 回の測定が必要になる。

HHL アルゴリズム

1. $|\boldsymbol{b}\rangle = \sum_j b_j|j\rangle$ を準備する。

2. e^{iA} の位相推定を $|\boldsymbol{b}\rangle$ を入力として行う。
   A の固有値を λ_i 対応する固有ベクトルを $|\boldsymbol{a}_i\rangle$ とし、
   $$|\boldsymbol{b}\rangle = \sum_i \beta_i|\boldsymbol{a}_i\rangle$$
   とするとき、このステップの出力は
   $$\sum_i \beta_i|\boldsymbol{a}_i\rangle|b(\lambda_i)\rangle$$
   $b(\lambda_i)$ は固有値 λ_i のビット列表現

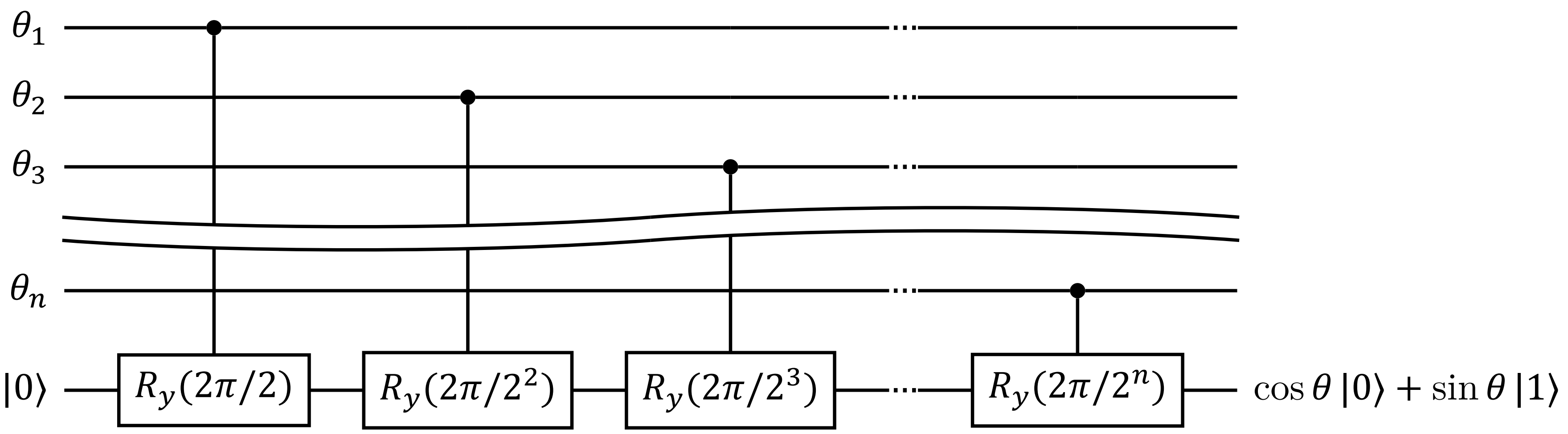
3. $|b(\lambda_i)\rangle$ をもとに、$|b(\cos^{-1}(C/\lambda_i))\rangle$ を計算する。
   トフォリゲートなどを使ったブール関数の計算で計算可能。C は定数で $C/\lambda < 1$ となるように設定する。

4. 補助量子ビットを一つ足し、制御ゲートによって以下の状態を生成する。(次スライド参照)
   $$\sum_i \beta_i|\boldsymbol{a}_i\rangle|b(\lambda_i)\rangle\left(\frac{C}{\lambda_i}|0\rangle + \sqrt{1-\frac{C^2}{\lambda_i^2}}|1\rangle\right)$$

5. 補助量子ビットを測定して、$|0\rangle$ が出たら成功。位相推定の逆変換で $|b(\lambda_i)\rangle$ を $|0\rangle$ に戻す。

回転角を振幅へ書き込む

- HHL アルゴリズムのステップ 4 では、ビット列として保存された角度 $\theta = \cos^{-1} C/\lambda$ を補助量子ビットの振幅に書き込む、という操作を行う。
- θ のビット列表現 $b(\theta)$ は、$\theta = 2\pi \times 0.\theta_1\theta_2 \cdots \theta_m$ となるような2進小数 $0.\theta_1\theta_2 \cdots \theta_m$ であるとする。
- これは以下の量子回路で実現できる。



演習：この回路が所望の動作をすることを確かめてください。

演習 (レポート)

1. HHL アルゴリズムに現れる状態

$$\sum_i \beta_i |\boldsymbol{a}_i\rangle |b(\lambda_i)\rangle \left(\frac{C}{\lambda_i}|0\rangle + \sqrt{1 - \frac{C^2}{\lambda_i^2}}|1\rangle \right)$$

で補助ビットが $|0\rangle$ に測定される確率 p_{accept} を求めてください。

2. $C = \min|\lambda_i|$ と選べたとします。このとき

$$p_{\mathrm{accept}} \geq \left(\frac{\min|\lambda_i|}{\max|\lambda_i|} \right)^2$$

を示してください。

グローバーのアルゴリズム

やりたいこと

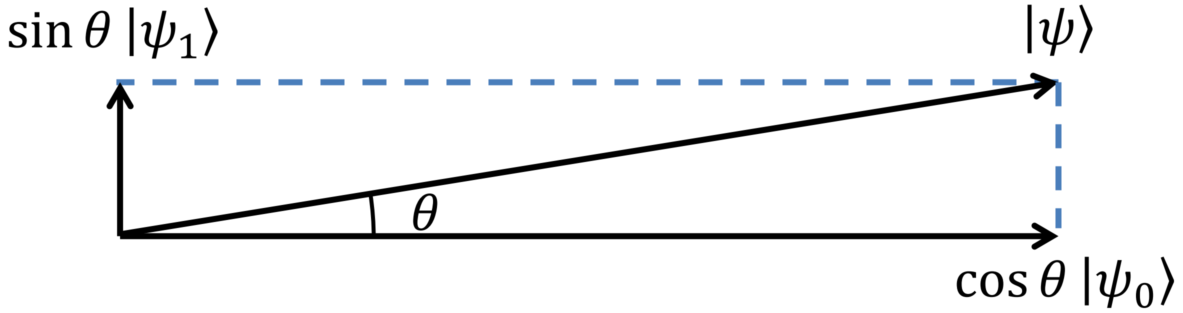
- 均等な重ね合わせ状態

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle$$

から、適当な条件 $f(k) = 1$ を満たすビット列 k を見つけたい。

- 準備：
$f(k) = 1$ となる部分 $|\psi_1\rangle = C_1 \sum_{f(k)=1} |k\rangle$ と $f(k) = 0$ となる部分 $|\psi_0\rangle = C_0 \sum_{f(k)=0} |k\rangle$ に分ける
C_1, C_0 は規格化定数。

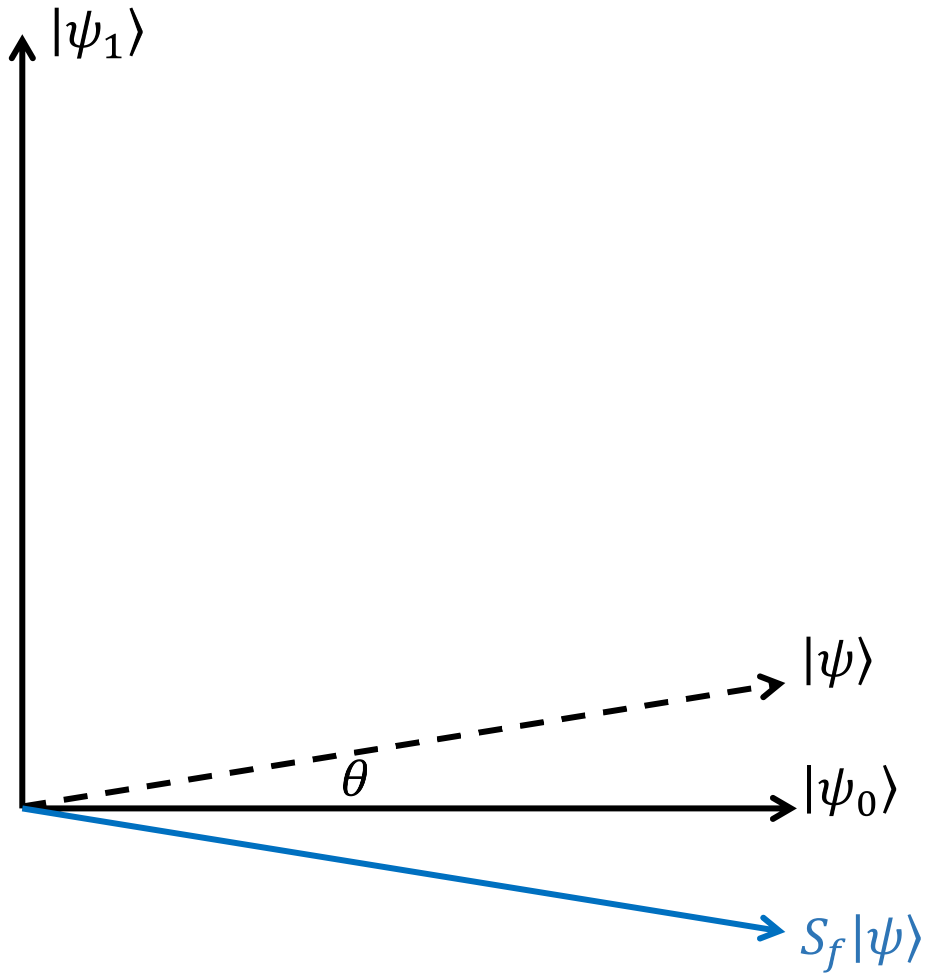
- $|\psi_1\rangle$ の振幅を大きくするようなユニタリ演算子を設計したい！



アルゴリズム

天下り的ですが…

1. $|\psi_0\rangle$ に対する反転演算 $S_f = I - 2|\psi_0\rangle\langle\psi_0|$ を作用させる。



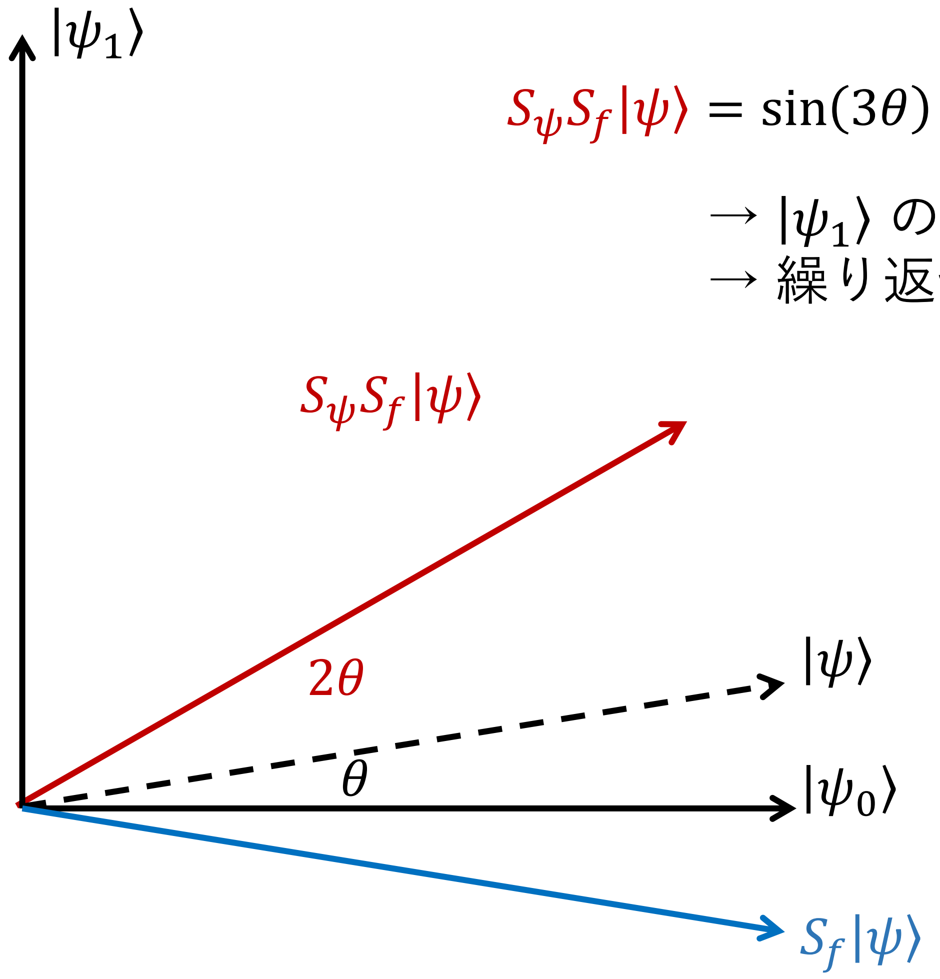
アルゴリズム

天下り的ですが…

2. $|\psi\rangle$ に関して反転する。

$$S_\psi S_f|\psi\rangle = \sin(3\theta)\,|\psi_1\rangle + \cos(3\theta)\,|\psi_0\rangle$$

→ $|\psi_1\rangle$ の振幅が大きくなった！
→ 繰り返せば $\theta, 3\theta, 5\theta, \ldots$ と大きくなる。



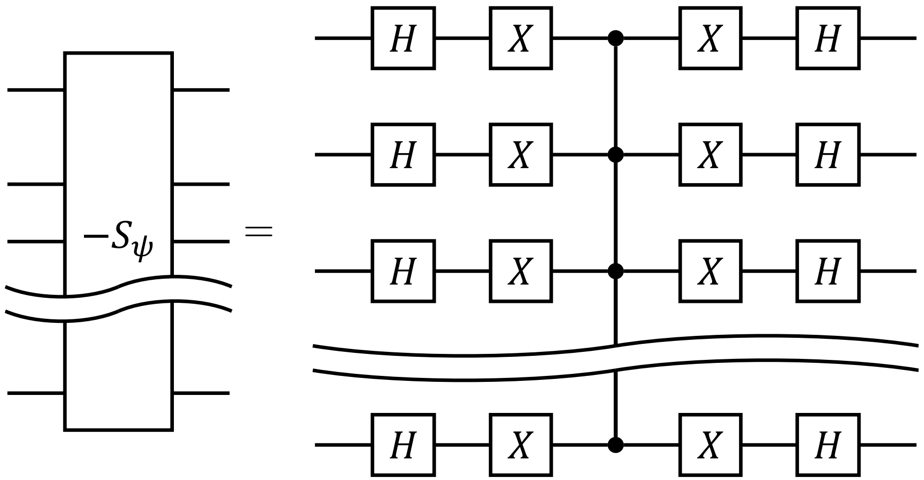
アルゴリズムの計算量

- $f(k) = 1$ を満たす k が 1 つしかない状況を考える。
- 初期状態 $|\psi\rangle = \frac{1}{\sqrt{N}}\sum_k |k\rangle$ だったので、1 つのビット列の振幅は $1/\sqrt{N}$ 。つまり
$$\sin\theta = \frac{1}{\sqrt{N}} \Rightarrow \theta \approx \frac{1}{\sqrt{N}}$$
- m 回の繰り返しで振幅は $\sin(2m+1)\theta$ まで増幅される。
- ほぼ確実に目的のビット列を見つけられるためには、$\sin(2m+1)\theta \approx 1$ となれば良い。つまり
$$(2m+1)\theta \approx \frac{\pi}{2}$$
を満たすように m を見つければ OK。
- よって
$$m = O\left(\sqrt{N}\right)$$

※ 古典計算では $f(k)$ に特段の構造が無い限り総当りを迫られる → 古典計算量 $O(N)$

反転演算の構成

- $|\psi\rangle$ に対する反転演算 S_ψ は
$$S_\psi = 2|\psi\rangle\langle\psi| - I$$
とかける。
- $|\psi\rangle = H^{\otimes n}|0\rangle$ なので、
$$S_\psi = H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n}$$
- $2|0\rangle\langle 0| - I$ は $|0\rangle$ 以外に対して位相 -1 をつけるゲート。CZ ゲートがすべての量子ビットが 1 のときのみ位相 -1 をつけることを思い出せば、以下の回路で実現できる。



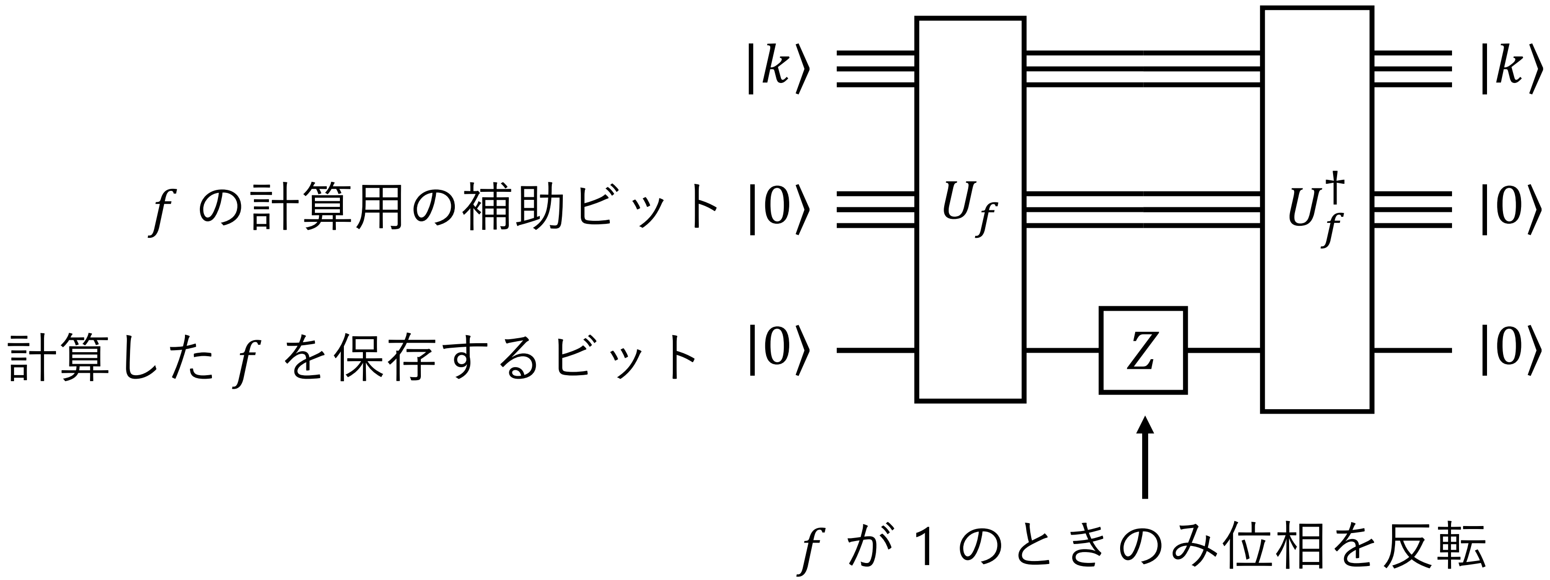
反転演算の構成 (2)

- $|\psi_0\rangle$ に対する反転演算 S_f は

$$S_f|k\rangle = \begin{cases} |k\rangle \text{ if } f(k) = 0 \\ -|k\rangle \text{ if } f(k) = 1 \end{cases}$$

 という演算。

- これは次のような回路で可能。
 U_f は f を計算するための古典可逆計算回路。



演習 (1,2 をレポート)

1. 演算子 $G = S_\psi S_f$ を $|\psi_0\rangle, |\psi_1\rangle$ に作用させた結果を計算し、G の作用は $|\psi_0\rangle, |\psi_1\rangle$ で張られる2次元空間内で閉じていることを示してください。
2. $|\psi_0\rangle, |\psi_1\rangle$ の2次元空間内における、G の固有値・固有ベクトルを計算してください。
3. （**量子振幅推移アルゴリズム**）$|\psi\rangle = \frac{1}{\sqrt{N}}\sum_k |k\rangle$ を入力状態として、ユニタリー G の位相推定をすると、何が得られるでしょうか？($f(k) = 1$ となる k が複数個ある状況を考えます。)

量子アルゴリズムのまとめ

- 量子コンピュータが古典に対して優位性を持つ代表的な応用先について説明した。
  - n 個の量子力学的スピン 1/2 の時間発展 [S. Lloyd, Science, **273**, 1073-1078 (1996)]

$$O(2^n) \to \mathrm{poly}\, n$$

  - n bit 整数の素因数分解 [P. W. Shor, Proceedings 35th Annual Symposium on Foundations of Computer Science, 124-134 (1994)]

$$O\left(e^{1.9 n^{1/3} (\log n)^{2/3}}\right) \to O(n^2 \log n \log\log n)$$

  - N 個のデータを持つデータベースの検索 [L. K. Grover, Proceedings, 28th Annual ACM Symposium on the Theory of Computing, 212-219, (1996)]

$$O(N) \to O\left(\sqrt{N}\right)$$

  - N 次元疎行列の逆行列計算 (疎性 s, 条件数 κ, 精度 $1/\epsilon$) [A. Harrow et al., PRL, **103**, 150502 (2009)]

$$O(Ns\sqrt{\kappa}\log 1/\epsilon) \to \tilde{O}(\log N\, s^2 \kappa^2/\epsilon)$$

量子誤り訂正

誤り訂正技術の必要性

現在の量子ビットのエラー率 ~ **0.1%** [Arute et. al., Nature (2019)]

⇕

現在の古典ビットのエラー率 ~ **10^{-17} %** [Oliveira et al, SC17 (2017)]

※クロック数からFITをエラー率に換算

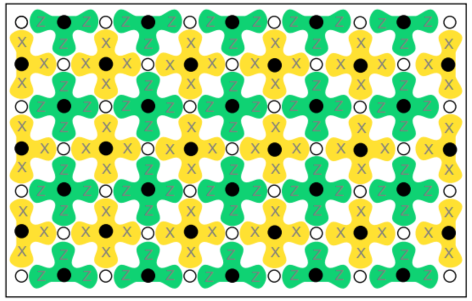
「まともな」計算をするには誤り訂正技術が必須

多数決符号 000 ──→ 010 ──→ 000

エラーが発生

多数決で正しい
状態に戻す

表面符号



~1000 qubit で 1 qubit を作る

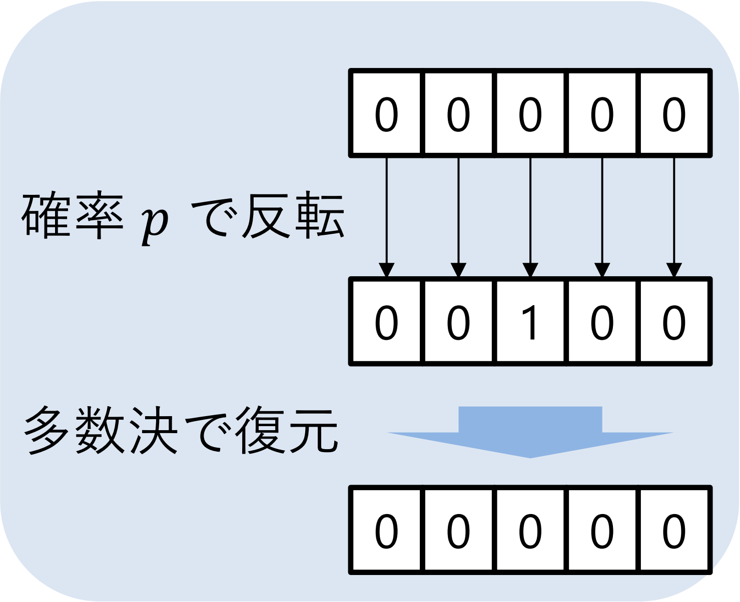
[Phys. Rev. A **86**, 032324 (2012)]

量子ビットに対するノイズ

- **ビット反転エラー**：
  ある確率 p で、量子ビットに X ゲートがかかってしまうというノイズモデル。普通古典ビットに対するノイズはこれ。
  $$|0\rangle \to |1\rangle, |1\rangle \to |0\rangle$$
  $$\alpha|0\rangle + \beta|1\rangle \to \beta|0\rangle + \alpha|1\rangle$$
- **位相反転エラー**：
  ある確率 p で、量子ビットに Z ゲートがかかってしまうというノイズモデル。(古典での対応物はない)
  $$\alpha|0\rangle + \beta|1\rangle \to \alpha|0\rangle - \beta|1\rangle$$
- **Depolarizing (分極解消) エラー**：
  ある確率 p で、量子ビットに X, Y, Z ゲートのどれかがかかってしまうというノイズモデル。
- **過回転エラー**：
  回転ゲートの回転角がずれることによるエラー。

古典誤り訂正：反復符号

- 一定時間経つと、各ビットにビット反転エラーが確率 $p \ll 1$ で起こるようなメモリがあるとする。
- 情報を守るために、0 を 00000 で、1 を 11111 で表すと約束する。(**符号化**)
- 読みだしたとき、0 or 1 の多い方がもとの情報だったと推定できる。(多数決復号)
  - 2 ビットまでの反転ならもとの情報を復元可能。
  - 3 ビットの反転 (確率 p^3) が起こると復元不可能に。

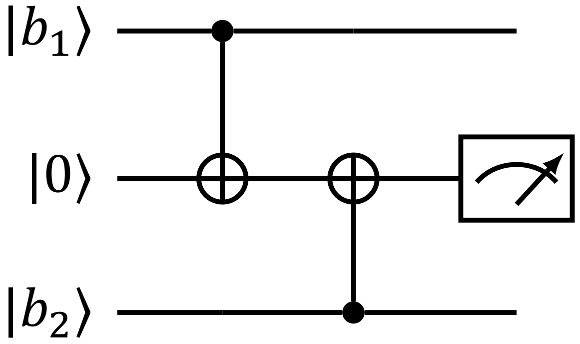


量子で反復符号を作れるか？

- 一定時間経つと、各量子ビットにビット反転エラーが確率 $p \ll 1$ で起こるとする。
- 情報を守るために、$|0\rangle$ を $|000\rangle$ で、$|1\rangle$ を $|111\rangle$ で符号化する。
- 量子情報を保護したければ
$$\alpha|000\rangle + \beta|111\rangle$$
という**重ね合わせ状態も保護したい**はず。
- 多数決を取ろうとして量子ビットを観測すると、重ね合わせ状態は壊れてしまう。
→ **直接的な多数決は不可能。**
- 量子ビットの誤り訂正では、補助ビットを用いて間接的に誤り検出を行う必要あり。

量子反復符号のエラー検出

- $|0\rangle$ を $|000\rangle$ で、$|1\rangle$ を $|111\rangle$ で表すと約束する。
- エラー検出を、「隣り合うビット同士が同じか違うか (パリティーチェック)」で行う。
- 2 ビットのパリティを測定するには以下のような回路が使える。

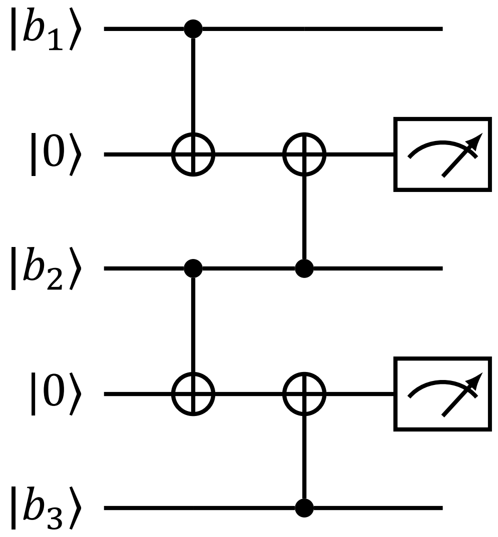


$b_1 = b_2$ のとき 0

$b_1 \neq b_2$ のとき 1

量子反復符号のエラー検出

- パリティチェックによって、3 bit 反復符号のエラー検出をする回路：



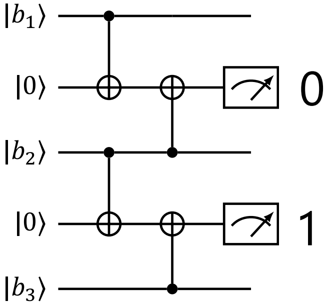
※ 測定結果を**シンドローム**と呼ぶ

- $|000\rangle$ か $|111\rangle$ しか含まれていないはずなので、エラーが無いときは補助ビットは 2 つとも 0

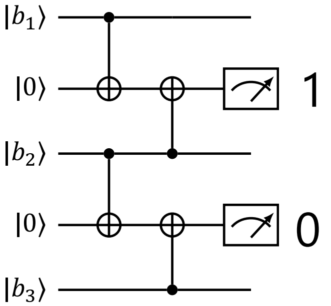
演習：$|b_1\rangle, |b_2\rangle, |b_3\rangle$ の量子ビットに $\alpha|000\rangle + \beta|111\rangle$ を入力したとき、上記の測定はこの状態を壊さないことを示してください。

量子反復符号のエラー検出

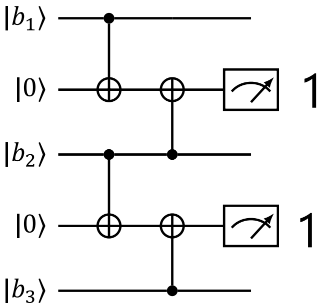
➢ 残りの 3 パターンを考えると…



- ~~b_1, b_2 に同時にエラーが起きた？~~ (確率 p^2)
- b_3 にエラーが起きた。 (確率 p)



- ~~b_2, b_3 に同時にエラーが起きた？~~ (確率 p^2)
- b_1 にエラーが起きた。 (確率 p)



- ~~b_1, b_3 に同時にエラーが起きた？~~ (確率 p^2)
- b_2 にエラーが起きた。(確率 p)

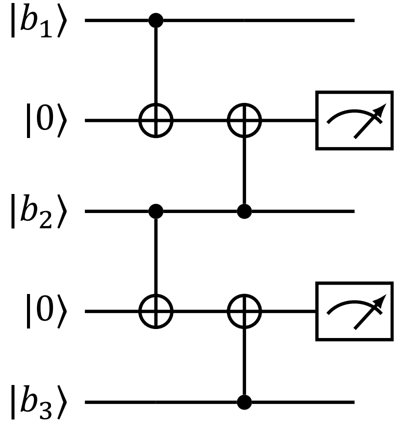
誤り検出・訂正可能

回転エラーも訂正可能

- 量子状態は連続的なものなので、$\alpha|000\rangle + \beta|111\rangle$ に対して作用するエラーは、
$$\alpha|000\rangle + \beta|111\rangle \to \alpha|100\rangle + \beta|011\rangle$$
のような離散的なものだけではない。

- より一般に 1 番目の量子ビットを角度 θ だけ回転させてしまうエラー
$$\alpha|000\rangle + \beta|111\rangle \to \alpha(\cos\theta\,|0\rangle + i\sin\theta\,|1\rangle)|00\rangle + \beta(i\sin\theta\,|0\rangle + \cos\theta\,|1\rangle)|11\rangle$$
も起きうる。

- **しかし実はこのエラーも反復符号で訂正可能。**

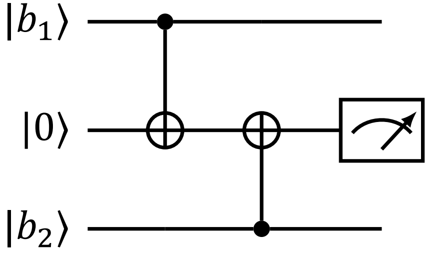


測定時に訂正可能な状態へ射影される。

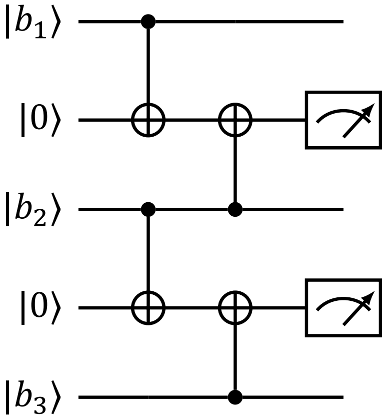
演習：$|b_1\rangle, |b_2\rangle, |b_3\rangle$ の量子ビットに $\alpha(\cos\theta\,|0\rangle + i\sin\theta\,|1\rangle)|00\rangle + \beta(i\sin\theta\,|0\rangle + \cos\theta\,|1\rangle)|11\rangle$ を入力したとき、あり得るシンドローム測定結果と、対応する測定後の状態を書き下してください。

パリティチェックと同値なオブザーバブル

- パリティチェックの測定は、$Z_1 Z_2$というオブザーバブルを射影測定していることに等しい。

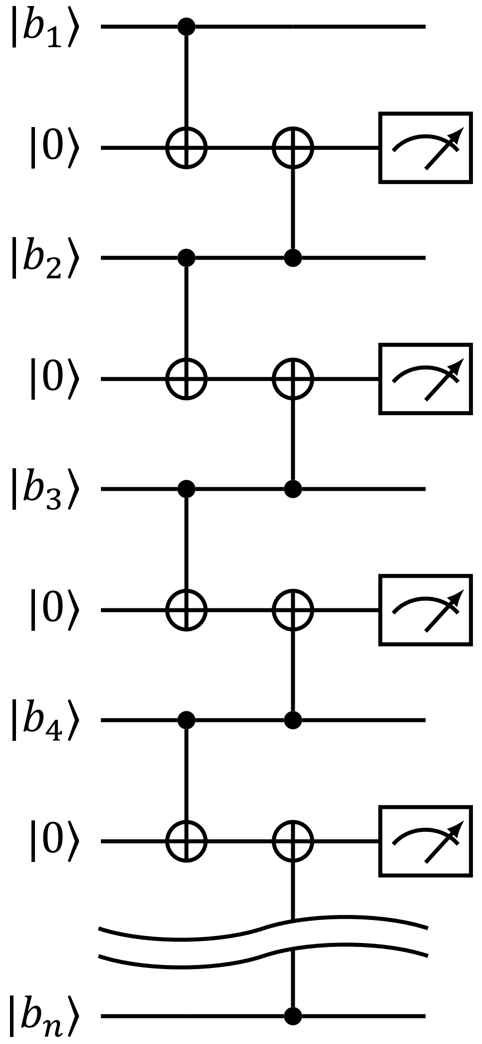


- $Z_1 Z_2$ は $b_1 = b_2$ のとき+1, $b_1 \neq b_2$ のとき-1 となるオブザーバブルだから。
- 3 量子ビットの反復符号のシンドローム測定は、$Z_1 Z_2$ と $Z_2 Z_3$ を射影測定していることに等しい。



後々便利な考え方です

n ビットの量子反復符号のエラー検出



反復符号における論理ゲート

- 符号化によって守られた量子状態に対して、その符号空間内のみで作用する演算子を**論理演算子**/**論理ゲート**と呼ぶ。
- $|0\rangle_L = |000\rangle$ と $|1\rangle_L = |111\rangle$ の空間だけに作用する論理ゲートは？
  - 論理 X 演算子：$X_L = X_1X_2X_3$

    $$X_L|0\rangle_L = |1\rangle_L$$

  - 論理 Z 演算子：$Z_L = Z_1Z_2Z_3$

    $$Z_L|1\rangle_L = -|1\rangle_L$$

    ※ $Z_L = Z_1, Z_2, Z_3$ としても、Z_L としての機能は満たせる。
- 他の論理演算子は、X_L と Z_L から作り出せば良い。

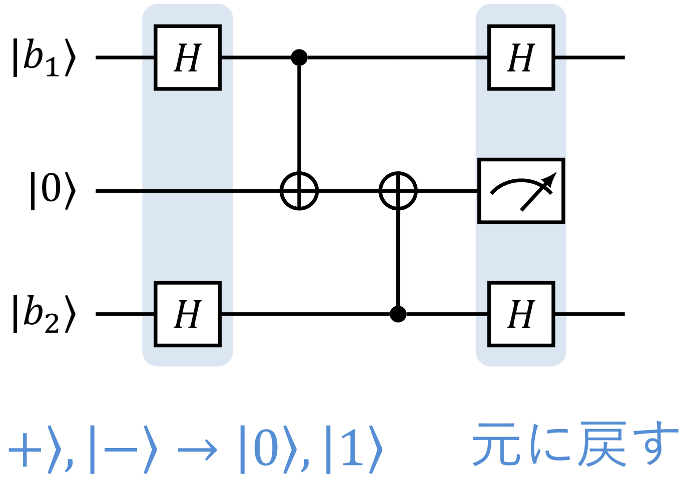
位相反転エラーも考える

- 一定時間経つと、各量子ビットにビット反転エラーが確率 $p \ll 1$ で起こるとする。
- $|0\rangle$ を $|000\rangle$ で、$|1\rangle$ を $|111\rangle$ で符号化することで、ビット反転エラーは訂正可能。
- しかし現実にはビット反転だけでなく、位相反転も起こる。
- アイデア：
  $|0\rangle, |1\rangle$ 基底の反復符号でビット反転エラーから守った論理量子ビットを、さらに位相反転エラーに対応できるような反復符号にする。

演習：$|0\rangle_L = |000\rangle, |1\rangle_L = |111\rangle$ としたとき、1つの物理量子ビットの位相反転エラーは、論理量子ビットの位相反転エラーと等価であることを示してください。

位相反転エラーに対する反復符号

- 位相反転エラーは、$|+\rangle, |-\rangle$ 基底でみるとビット反転エラーと等価。
- よって $|+\rangle, |-\rangle$ 基底で反復符号を構成すれば誤り検出・訂正できる。
- $|0\rangle_L = |{+}{+}{+}\rangle$, $|1\rangle_L = |{-}{-}{-}\rangle$ を使う。
- $|+\rangle, |-\rangle$ 基底でのパリティチェック回路



- X_1X_2 と X_2X_3 の射影測定をしていることに等しい。

Shor の 9 qubit code

- ビット反転エラーを訂正できる反復符号を構成する。

$$|0\rangle_{L'} = |000\rangle, |1\rangle_{L'} = |111\rangle$$

この時点では、1 つの物理量子ビットに対する位相反転エラー = 論理量子ビットに対する位相反転エラーとなり、位相反転エラーの訂正は不可。

- $|0\rangle_{L'}$ を 3 つ使って、位相反転エラーを訂正できる反復符号を構成する。

$$|0\rangle_L = |+\rangle_{L'}|+\rangle_{L'}|+\rangle_{L'}, |1\rangle_L = |-\rangle_{L'}|-\rangle_{L'}|-\rangle_{L'}$$

各 $|+\rangle_L$ に対する位相反転エラーが訂正可能に。

- できあがった論理ビットは

$$|0\rangle_L = \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle)^{\otimes 3}$$

$$|1\rangle_L = \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle)^{\otimes 3}$$

- **ビット/位相反転エラーのどちらにも対応できる。**

Shor の 9 qubit code のシンドローム測定

- $|0\rangle_{L'} = |000\rangle, |1\rangle_{L'} = |111\rangle$ の誤り検出には、

$$Z_1Z_2, \qquad Z_2Z_3$$

  を測定すれば OK.

- $|0\rangle_L = |+++\rangle_{L'}, |1\rangle_L = |---\rangle_{L'}$ の誤り検出には、

$$X_{L'} \otimes X_{L'} \otimes I, \qquad I \otimes X_{L'} \otimes X_{L'}$$

  を測定する。

- $X_{L'} = X_1X_2X_3$ とかけていたので、9 qubit code で測定すべき演算子は、以下の 8 個の演算子。

$$\begin{array}{c} ZZIIIIIII \\ IZZIIIIII \\ IIIZZIIII \\ IIIIZZIII \\ IIIIIIZZI \\ IIIIIIIZZ \\ XXXXXXIII \\ IIIXXXXXX \end{array}$$

演習：1つの補助ビットを使って、6 量子ビットの演算子 $X^{\otimes 6}$ を測定する回路を書いてください。

ビット/位相反転耐性 → 任意の1 qubit エラー耐性

- $iXZ = -iZX = Y$ なので、ビット反転 + 位相反転への耐性があれば Y エラーにも耐性。
- 先述の議論により、1 qubit の微妙な回転でも訂正可能。
- **Shor 9 qubit code**

$$|0\rangle_L = \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle)^{\otimes 3}$$
$$|1\rangle_L = \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle)^{\otimes 3}$$

は任意の 1 qubit エラーの訂正が可能。

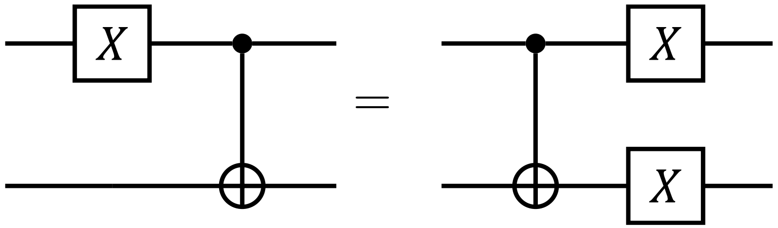
誤り耐性量子計算

- 1 qubit のエラーに耐性を持つ符号を構成することができた。
- 誤り耐性を持つ論理量子ビット $|0\rangle_L, |1\rangle_L$ によって量子計算 (**誤り耐性量子計算**) をしたい。
- 誤り耐性量子計算に必要な条件：
  - ✓ 論理ゲート $\alpha|0\rangle_L + \beta|1\rangle_L \to \alpha'|0\rangle_L + \beta'|1\rangle_L$ をかけたとき、訂正可能なエラーしか発生しない。
  - ✓ シンドローム測定のとき、訂正可能なエラーしか発生しない。

- Shor の 9 qubit code の論理ゲート：
  - 論理 X ：すべての qubit に Z をかける。(**transversal**)
  - 論理 Z ：すべての qubit に X をかける。
  - 論理 H ：~~すべての qubit に H をかける~~ これではできない。→ 2 qubit エラーが不可避...

エラーの伝搬

➢ 回路中に 2 qubit ゲートが存在すると、エラーが拡散する。



➢ 論理演算やシンドローム測定に 2 qubit ゲートを使いすぎると、エラーの訂正が不可能に。

➢ 1 qubit ゲートのテンソル積 = 論理演算 となっているのが好ましい。

一般化：スタビライザー符号

誤り耐性量子計算が可能な符号を求めて…

- Shor 9 qubit codeで測定していた演算子をもう一度見直す。

$$
\begin{gathered}
ZZIIIIIII \\
IZZIIIIII \\
IIIZZIIII \\
IIIIZZIII \\
IIIIIIZZI \\
IIIIIIIZZ \\
XXXXXXIII \\
IIIXXXXXX
\end{gathered}
$$

- これらの演算子は全部可換。
- 一般に、可換なパウリ演算子がなす群をスタビライザー群と呼ぶ。
- $|0\rangle_L$ と $|1\rangle_L$ (エラーのない状態) はこれらの演算子の +1 固有状態。
  = スタビライザー群の全ての演算子に対して +1 固有状態となっているのが $|0\rangle_L, |1\rangle_L$。
- スタビライザー群の+1固有状態となる状態を**スタビライザー状態**と呼ぶ。
- スタビライザー状態によって作る符号：**スタビライザー符号**

スタビライザー符号

- Shor 9 qubit code で測定していた演算子をもう一度見直す。

$$
\begin{gathered}
ZZIIIIIII \\
IZZIIIIII \\
IIIZZIIII \\
IIIIZZIII \\
IIIIIIZZI \\
IIIIIIIZZ \\
XXXXXXIII \\
IIIXXXXXX
\end{gathered}
$$

- $H^{\otimes 9}$ をかけるとスタビライザー群が変わってしまう。
  → $H^{\otimes 9} X_L H^{\otimes 9} = Z_L$ だが、$H^{\otimes 9}$ は状態を符号空間から出してしまう (論理エラー)。

- なぜ Shor 9 qubit code が 1 qubit エラーを全て訂正できたのか？
  - ✓ スタビライザー演算子が 8 個
    → 「全ての演算子が+1になる」という条件で、2 次元 (1 qubit 分) の部分空間を指定できた。
  - ✓ 全ての qubit が Z 演算子と X 演算子に見張られている。
    → 各量子ビットに X エラーか Z エラーが起きると、シンドロームが反転する。
    $XZX = -Z, ZXZ = -X$ なので

Steane 7 qubit code

- 以下のスタビライザーで定義される符号。
  ハミング符号から構成される。

$$
\begin{matrix}
I\,I\,I\,Z Z Z Z\\
I\,Z Z\,I\,I\,Z Z\\
Z\,I\,Z\,I\,Z\,I\,Z\\
I\,I\,I\,X X X X\\
I\,X X\,I\,I\,X X\\
X\,I\,X\,I\,X\,I\,X
\end{matrix}
$$

- 任意の 1 qubit エラーを訂正可能。かつ、
  - ✓ $X_L = X^{\otimes 7}$
  - ✓ $Z_L = Z^{\otimes 7}$
  - ✓ $H_L = H^{\otimes 7}$ → transversal アダマールゲートが可能！
  - ✓ $\mathrm{CNOT}_L = \mathrm{CNOT}^{\otimes 7}$ → transversal CNOT ゲートが可能！

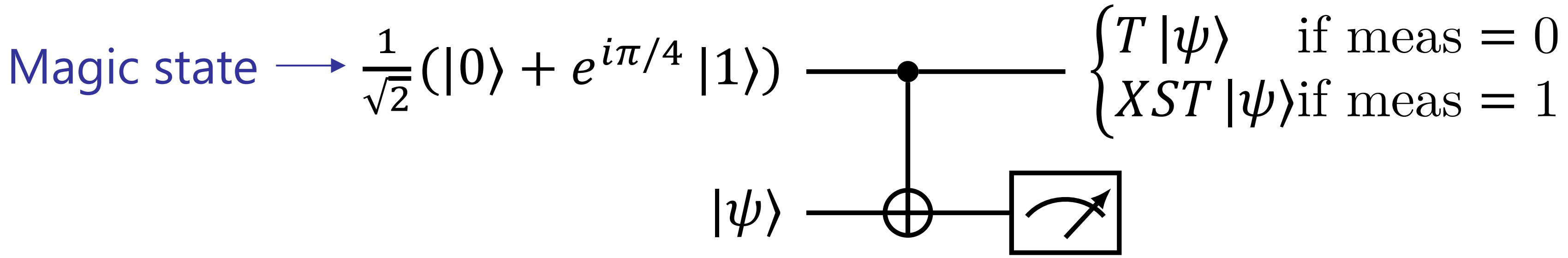
演習： $H^{\otimes 7}$ がスタビライザーを変化させないことを示してください。

任意の 1 qubit 論理ゲートを使うには

- スタビライザー符号は、パウリ演算子の固有状態として定義される。
- パウリ演算子をパウリ演算子に移すゲート (**クリフォードゲート**) なら、スタビライザーを変化させずに作用させられる。
- しかし例えば $R_z(\theta)$ は非クリフォードで
$$R_z(\theta) X R_z^\dagger(\theta) = \cos\theta\, X + \sin\theta\, Y$$
と変換してしまう。
- $R_{z,L}(\theta)$ を作るには、$X_L = X^{\otimes 7}$ を無理やり $\cos\theta\, X_L + \sin\theta\, Y_L$ に変換するしか無いのでは…？
でもこれには大量の 2 qubit ゲートが必要… 誤り耐性が失われてしまう…

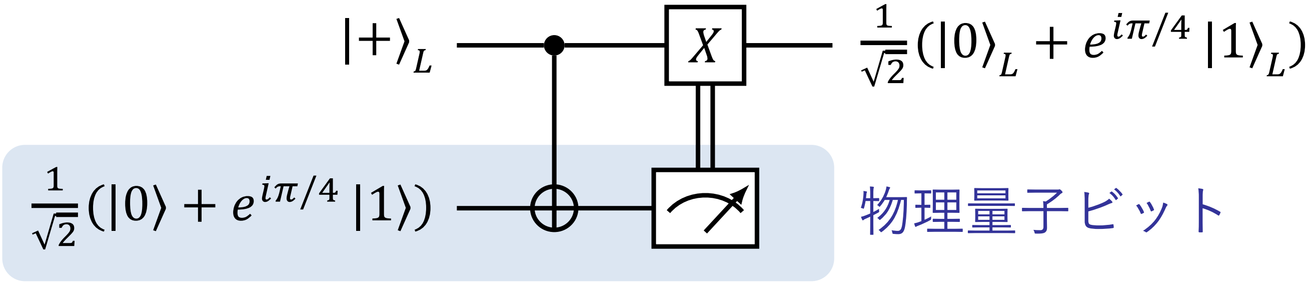
テレポーテーションによる T ゲート

- ゲートテレポーテーション



誤り訂正符号化した量子ビットで行えば、T ゲート (非クリフォード) を作用させられる。

- でもそもそも $\frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}|1\rangle)$ が作れないのでは？
  → 論理量子ビットと物理量子ビット間の量子テレポーテーションを使う。

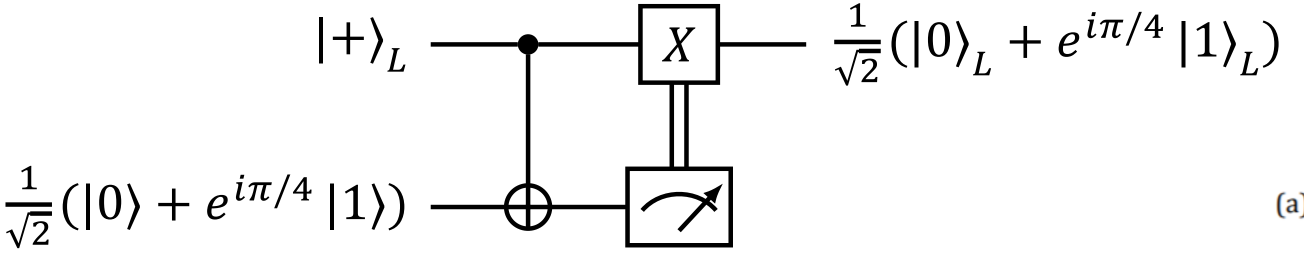


(Solovay-Kitaev のアルゴリズム)

- **T と H で任意の 1 qubit ゲートが構成できることが知られている。**

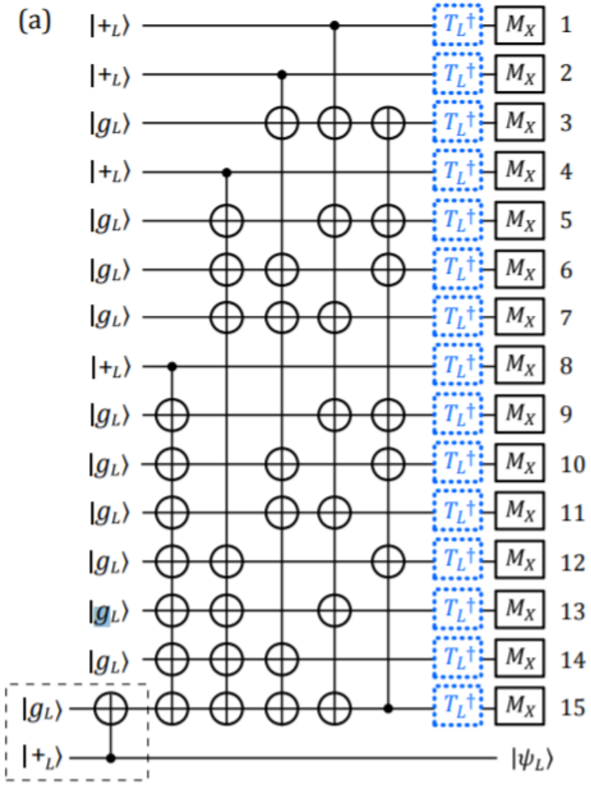
Magic state distillation

- テレポーテーションによって送られてくる論理 magic state はノイズを含む。



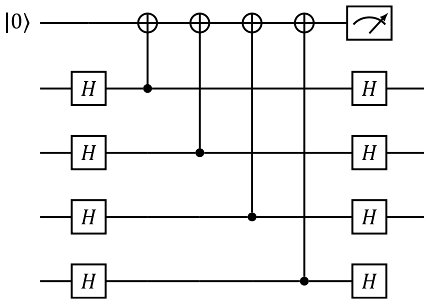
- クリフォード演算のみを使って、ノイズを含む magic state をきれいな magic state に “蒸留” (distill) できることが知られている。
- 右は 15 個の noisy T ゲートをきれいな magic state に蒸留する回路の例。

arXiv:1208.0928



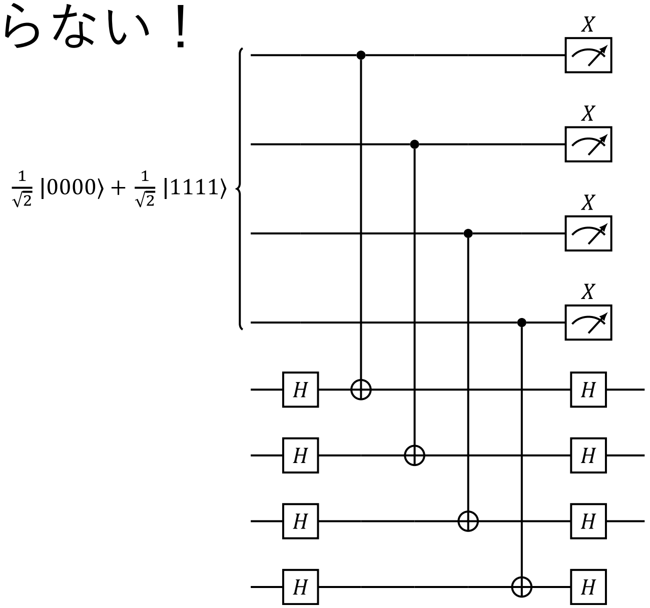
スタビライザー測定の誤り耐性

- Steane code のスタビライザー測定回路：



このままではエラーが拡散してしまう。

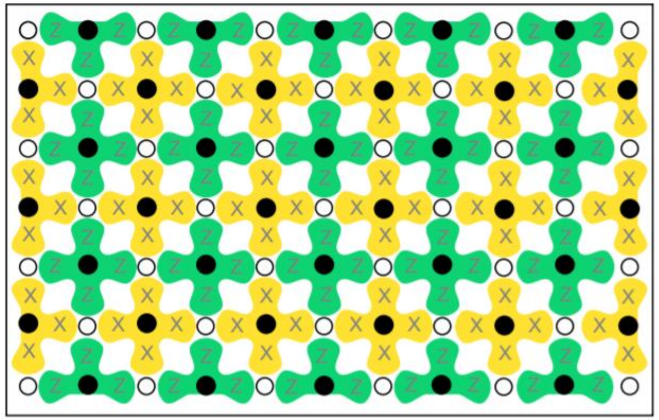
- 改良案：エラー拡散が起こらない！



誤り訂正まとめ

- 量子状態を冗長化することで、誤り耐性をもたせることができる。
- シンドローム測定によって、連続的エラーも離散的に訂正可能になる。
- 誤り耐性量子計算には、すべての物理ゲートで発生しうるエラーが訂正できるような符号を使う必要がある。
- T ゲートが transversal に使えないスタビライザー符号では、 magic state distillation とゲートテレポーテーションによってきれいな T ゲートを作る必要がある。
- **誤り耐性量子計算は可能。**
  エラーがある程度局所的であるという仮定のもとで。

最も実現しそうな誤り訂正符号は**表面符号** →
今回は説明しませんでしたが…



arXiv: 1208.0928