量子コンピュータの基礎と検証

参考文献:

- Phys. Rev. X **8**, 021060 (2018)
- npj Quantum Inf. **5**, 27 (2019)
- New J. Phys. **21**, 093060 (2019)
- Quantum **6**, 758 (2022)
- Phys. Rev. A **106**, L010601 (2022)

竹内 勇貴

NTT コミュニケーション科学基礎研究所
NTT 理論量子情報研究センタ

2024年3月4日 09:40 – 10:40

日本応用数理学会
第20回研究部会連合発表会

TAKE-HOMEメッセージ

量子コンピュータは古典コンピュータ**以上**の
計算能力が理論的に証明されている

しかし、

現在の量子コンピュータの実装・運用方法では
エラー確率が高い

量子計算中のエラーの有無は
「**量子計算の検証**」で効率的に判定可能

TAKE-HOMEメッセージ

1st Part

量子コンピュータは古典コンピュータ**以上**の
計算能力が理論的に証明されている

しかし、

現在の量子コンピュータの実装・運用方法では
エラー確率が高い

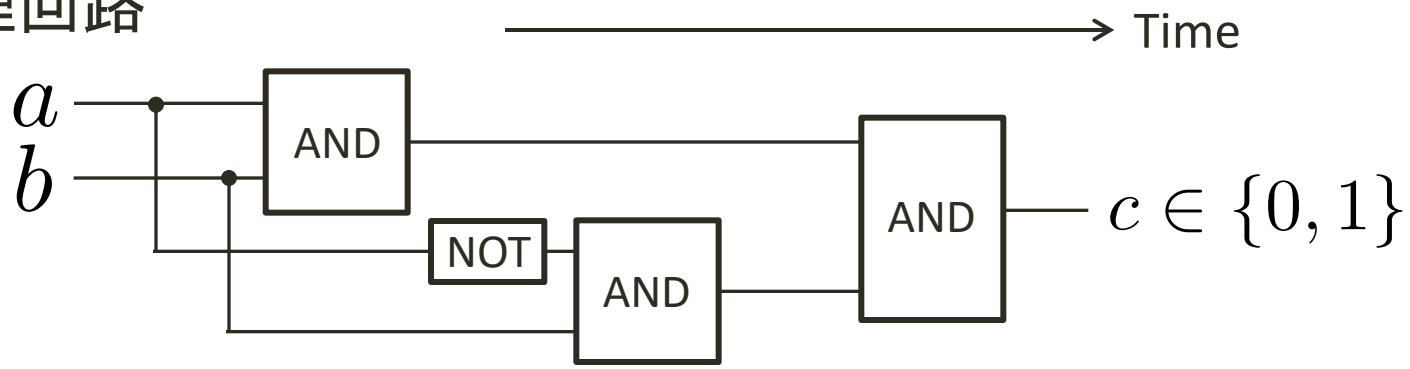
2nd Part

量子計算中のエラーの有無は
「**量子計算の検証**」で効率的に判定可能

1. 量子コンピュータの基礎

古典コンピュータ

- 論理回路

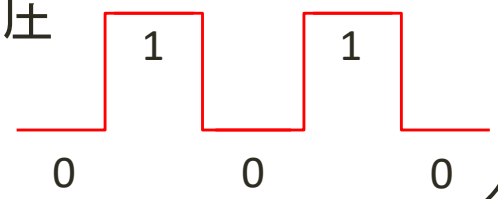


入力 (ビット)　　論理(古典)ゲート　　出力 (ビット)

ビット: 古典情報の最小単位

$$a, b \in \{0,1\}$$

例) 電圧



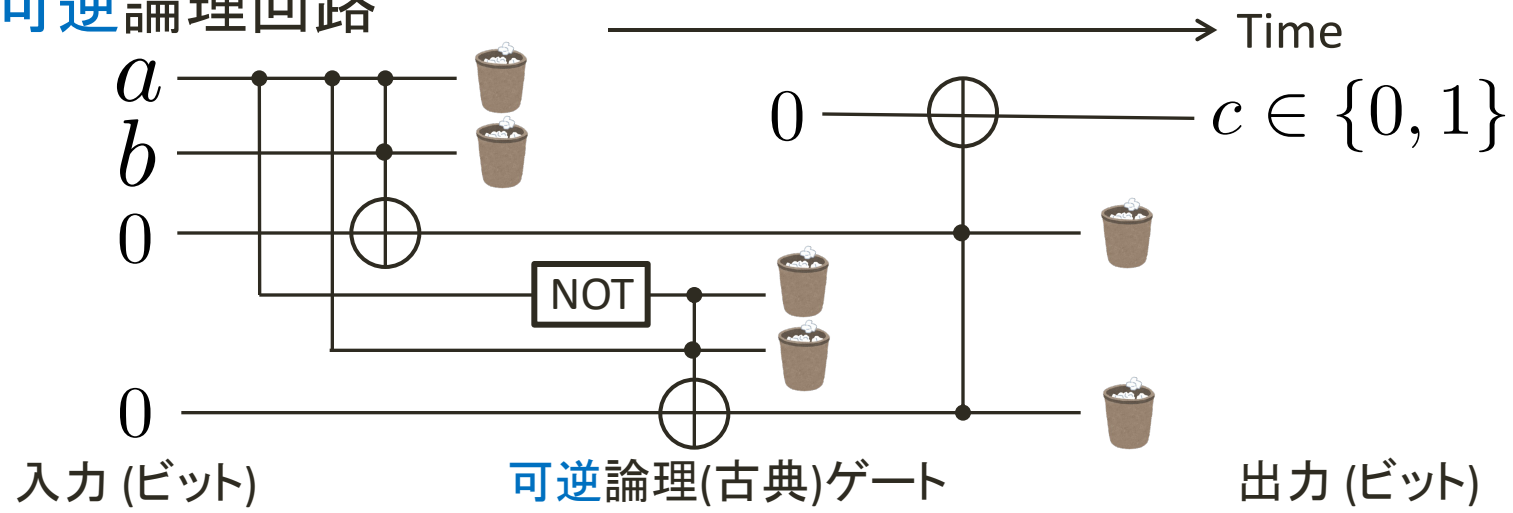
ユニバーサルゲートセット

入力 a	出力 $\neg a$
0	1
1	0

入力 (a, b)		出力 $a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

古典コンピュータ

- 可逆論理回路



可逆ユニバーサルゲートセット

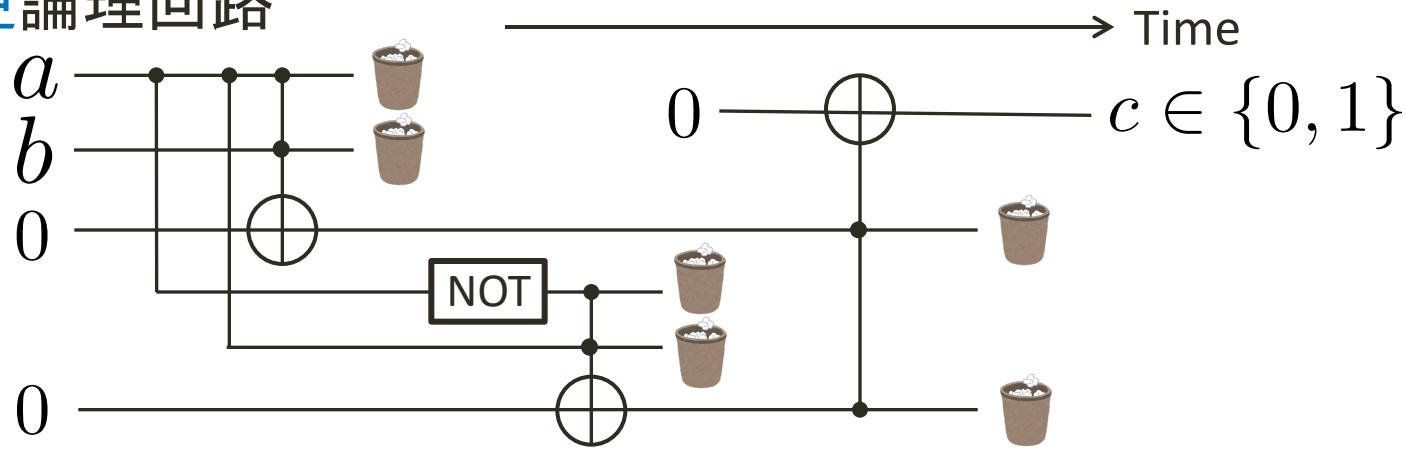
NOTゲート + トフォリゲート

入力 a	出力 $\neg a$
0	1
1	0

入力 (a, b, c)			出力
0	0	0	000
0	0	1	001
0	1	0	010
0	1	1	011
1	0	0	100
1	0	1	101
1	1	0	111
1	1	1	110

古典コンピュータ

- 可逆論理回路



可逆ユニバーサルゲートセット

NOTゲート + トフォリゲート

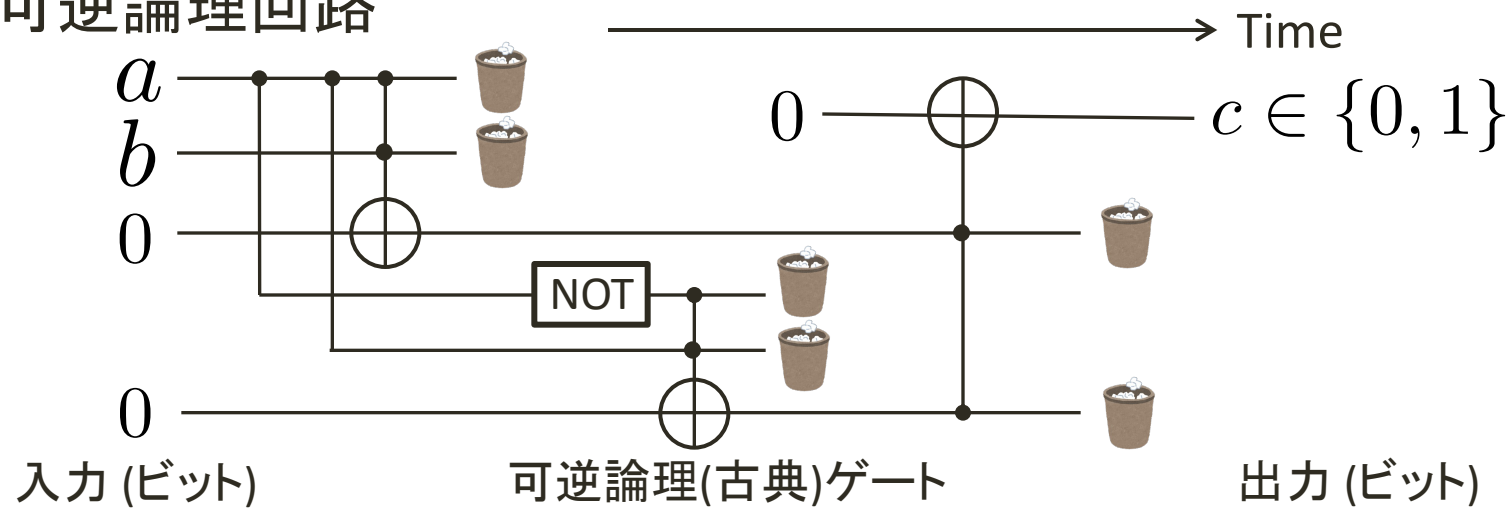
入力 a	出力 $\neg a$
0	1
1	0

AND

入力 (a, b, c)			出力
0	0	[redacted]	[redacted]0
0	0	1	001
0	1	[redacted]	[redacted]0
0	1	1	011
1	0	[redacted]	[redacted]0
1	0	1	101
1	1	[redacted]	[redacted]1
1	1	1	110

量子コンピュータ

- 可逆論理回路



量子ビット
(2次元正規ベクトル)

量子ゲート
(ユニタリ行列)

古典・量子コンピュータの違い①

2つのコンピュータでは、情報の表現方法が異なる

- 古典コンピュータ: 古典ビット

  0 または 1

- 量子コンピュータ: 量子ビット

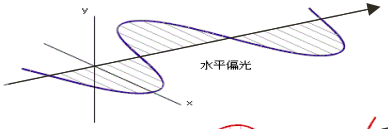
  0と1の重ね合わせ
  ･･･0と1を“同時”に表現

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

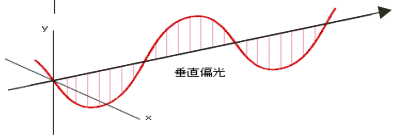
$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\alpha, \beta \in \mathbb{C} \ \text{s.t.}\ |\alpha|^2 + |\beta|^2 = 1$$

例) 光子の偏光 $|0\rangle$



$|1\rangle$



古典・量子コンピュータの違い①

2つのコンピュータでは、情報の表現方法が異なる

- 古典コンピュータ: 古典ビット

0 または 1

$(\alpha, \beta) = (0, 1)$の特別ケース

$(\alpha, \beta) = (1, 0)$の特別ケース

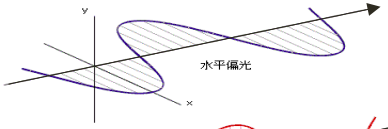
- 量子コンピュータ: 量子ビット

0と1の重ね合わせ
・・・0と1を“同時”に表現

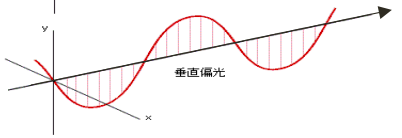
$$\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

$$\left(\begin{array}{l} |0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \ |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ \alpha, \beta \in \mathbb{C} \ \text{s.t.} \ |\alpha|^2 + |\beta|^2 = 1 \end{array} \right)$$

例) 光子の偏光 $|0\rangle$



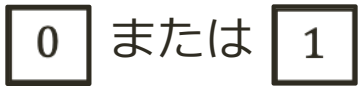
$|1\rangle$



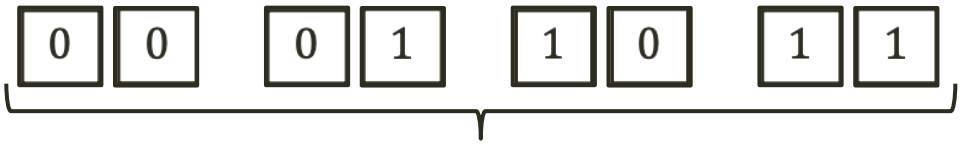
古典・量子コンピュータの違い①

2つのコンピュータでは、情報の表現方法が異なる

- 古典コンピュータ: 古典ビット

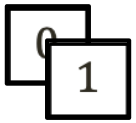


4種類の数字を同時に表現するなら、



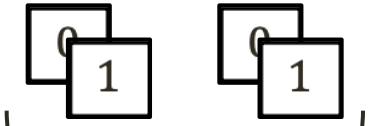
2×4 = 8ビット必要!!

- 量子コンピュータ: 量子ビット



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

4種類の数字を同時に表現するなら、



2量子ビットで十分!!

$$\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$$

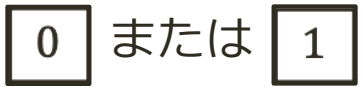
$$= (\alpha, \beta, \gamma, \delta)^T$$

$$\left[\forall i, j \in \{0, 1\},\ |ij\rangle \equiv |i\rangle \otimes |j\rangle\right]$$

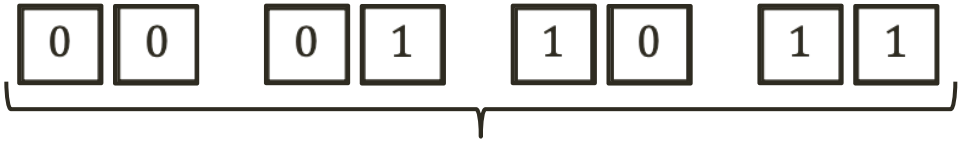
古典・量子コンピュータの違い①

2つのコンピュータでは、情報の表現方法が異なる

- 古典コンピュータ: 古典ビット

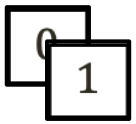


4種類の数字を同時に表現するなら、



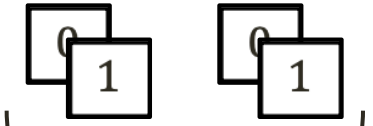
2×4 = 8ビット必要!!

- 量子コンピュータ: 量子ビット



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

4種類の数字を同時に表現するなら、



2量子ビットで十分!!

2^n個の数字を同時に表現するなら、
$n2^n$ビット
必要☹

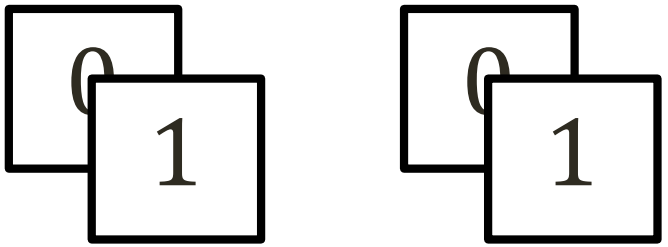
→ 指数的に少ない →

2^n個の数字を同時に表現するなら、
n量子ビット
必要☺

"並列"計算が得意

古典・量子コンピュータの違い①

Remark: **エンタングル状態**



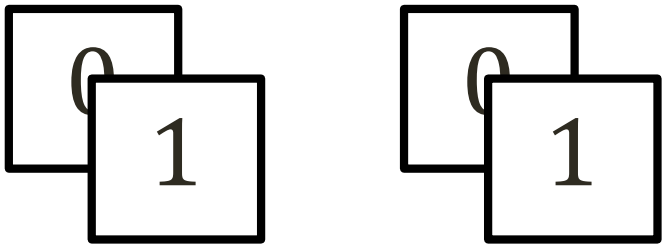
$$\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle = \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix}$$

古典ビットと同じ？

$$(\alpha|0\rangle + \beta|1\rangle) \otimes (\alpha'|0\rangle + \beta'|1\rangle)$$

古典・量子コンピュータの違い①

Remark: **エンタングル状態**



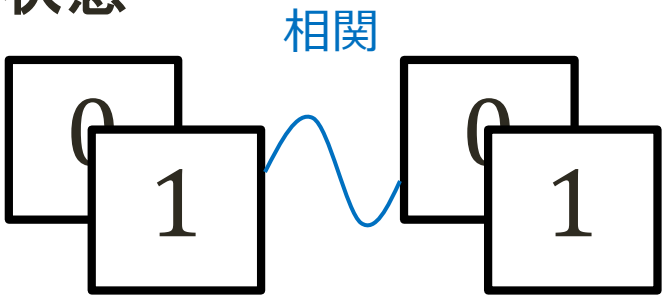
$$\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle = \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix}$$

古典ビ

$$(\quad \beta|1\rangle) \otimes (\alpha\,|0\rangle \quad\quad \rangle)$$

古典・量子コンピュータの違い①

Remark: **エンタングル状態**



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

$(\alpha|0\rangle + \beta|1\rangle) \otimes (\alpha'|0\rangle + \beta'|1\rangle)$ の形に書くためには

$$\alpha\beta' = \beta\alpha' = 0 \quad \text{かつ} \quad \alpha\alpha' = \beta\beta' = \frac{1}{\sqrt{2}}$$

が要求されるが、この条件を満たす$(\alpha, \beta, \alpha', \beta')$は存在しない

古典・量子コンピュータの違い②

量子コンピュータの操作(量子ゲート)はユニタリ行列で表現できる

- シュレディンガー方程式

量子計算は、量子力学の原理に基づいた計算モデル
→可能な操作は量子力学の原理によって決まる

Erwin Schrödinger
[https://en.wikipedia.org/wiki/Erwin_Schrödinger]

$$i\frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle \quad \Rightarrow \quad |\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$$

ハミルトニアン
(エルミート行列)

古典・量子コンピュータの違い②

量子コンピュータの操作(量子ゲート)はユニタリ行列で表現できる

- シュレディンガー方程式

量子計算は、量子力学の原理に基づいた計算モデル
→可能な操作は量子力学の原理によって決まる

Erwin Schrödinger
[https://en.wikipedia.org/wiki/Erwin_Schrödinger]

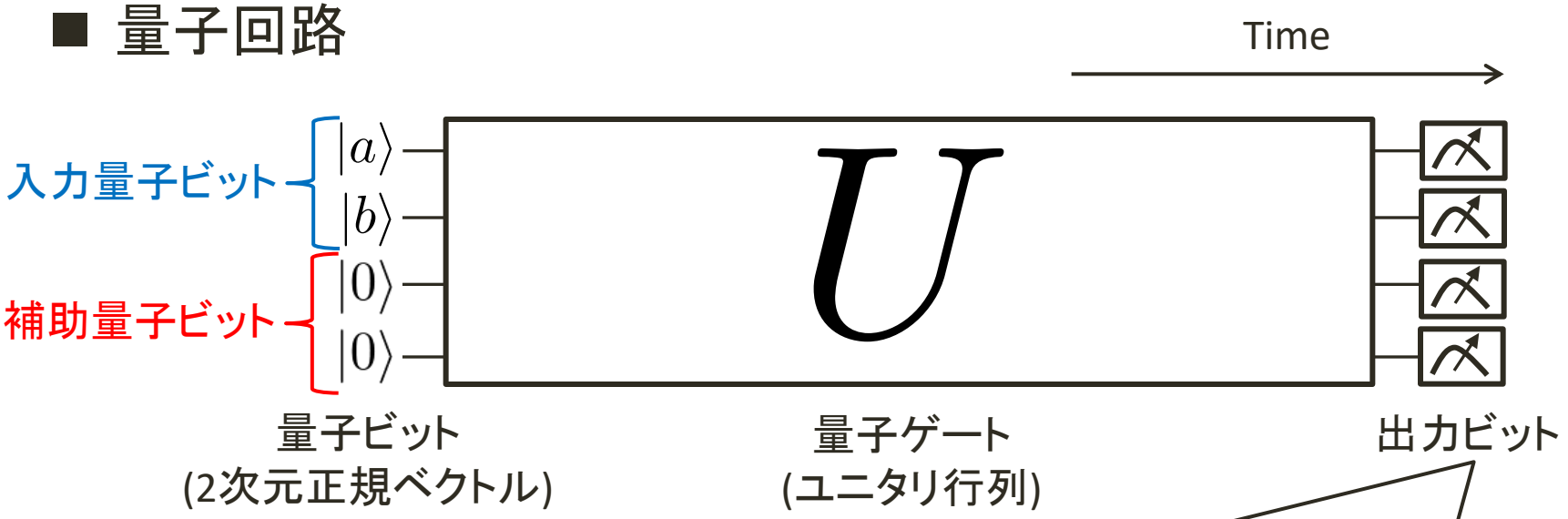
$$i\frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle \quad \Rightarrow \quad |\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$$

ハミルトニアン
(エルミート行列)

量子ゲートはユニタリ行列で書ける

量子コンピュータ

■ 量子回路



➢ (計算基底)**測定**

量子ビット(ベクトル)をビットに確率的に変換する操作

$$\alpha|0\rangle + \beta|1\rangle \longrightarrow \begin{cases} |\alpha|^2\text{の確率でビット0} \\ |\beta|^2(=1-|\alpha|^2)\text{の確率でビット1} \end{cases}$$

量子コンピュータ

- ユニバーサルゲートセット

Solovay-Kitaevの定理 [C. M. Dawson and M. A. Nielsen, Quantum Inf. Comput. **6**, 81 (2006)]

アダマールゲート

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

Tゲート

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

CNOTゲート

$$CX = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

の組み合わせ(掛け算とテンソル積)で、任意のユニタリ行列を近似できる。

量子コンピュータ

- ユニバーサルゲートセット

Solovay-Kitaevの定理 [C. M. Dawson and M. A. Nielsen, Quantum Inf. Comput. **6**, 81 (2006)]

アダマールゲート、Tゲート、CNOTゲートの組み合わせ(掛け算とテンソル積)で、任意のユニタリ行列を近似できる。

アダマールゲートの役割: 重ね合わせ状態の生成

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

Tゲートの役割: 重ね合わさり具合の調整

$$T\frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{i\pi/4}|1\rangle}{\sqrt{2}}$$

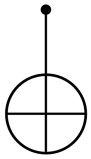
CNOTゲートの役割: エンタングル状態の生成

$$CX\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |0\rangle\right) = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

量子コンピュータ vs. 古典コンピュータ

量子ユニバーサルゲートセット

CNOT



アダマールゲート

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

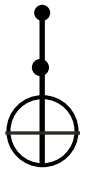
T gate

$$T \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

vs.

古典ゲートセット

トフォリゲート



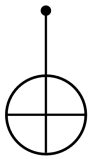
NOT

NOT

量子コンピュータ vs. 古典コンピュータ

量子ユニバーサルゲートセット

CNOT



アダマールゲート

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

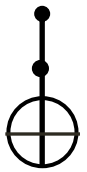
T gate

$$T \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

vs.

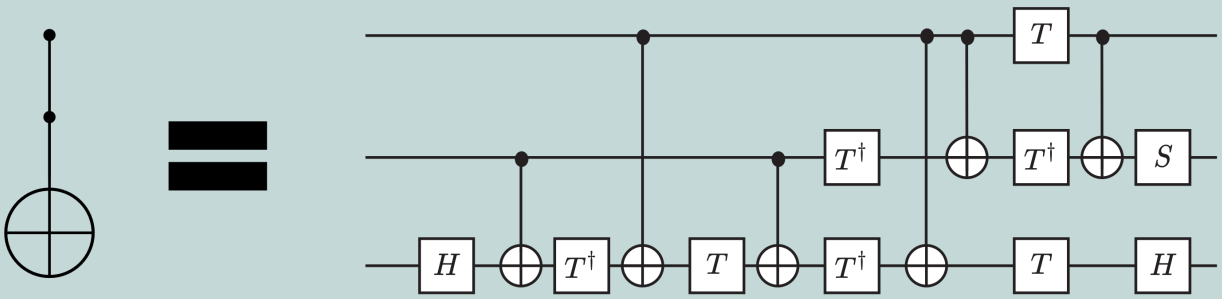
古典ゲートセット

トフォリゲート



NOT

※ $T^\dagger = T^7, S = T^2$



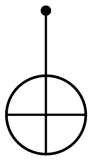
$$NOT = X$$
$$= HT^4H$$

[M. Rahman and G. Paul, IEEE Trans. on Quant. Engineering **3**, 3100809 (2022)]

量子コンピュータ vs. 古典コンピュータ

量子ユニバーサルゲートセット

CNOT



アダマールゲート

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

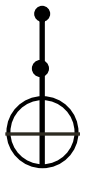
T gate

$$T \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

vs.

古典ゲートセット

トフォリゲート



NOT

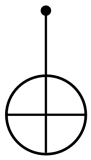
NOT

量子コンピュータは、古典コンピュータ以上の計算能力がある

量子コンピュータ vs. 古典コンピュータ

量子ユニバーサルゲートセット

CNOT



アダマールゲート

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

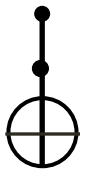
T gate

$$T \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

vs.

古典ゲートセット

トフォリゲート



NOT

NOT

量子コンピュータは、古典コンピュータ以上の計算能力がある

逆に、量子コンピュータにしか出来ないことはあるの？

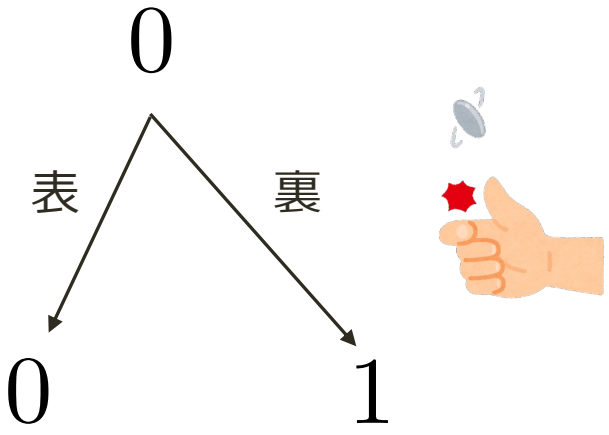
量子コンピュータ vs. 古典コンピュータ

コインフリップ

量子コンピュータの場合

$$|0\rangle \xrightarrow{H} \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

古典コンピュータの場合



どちらの場合でも、
50%の確率で1が出る

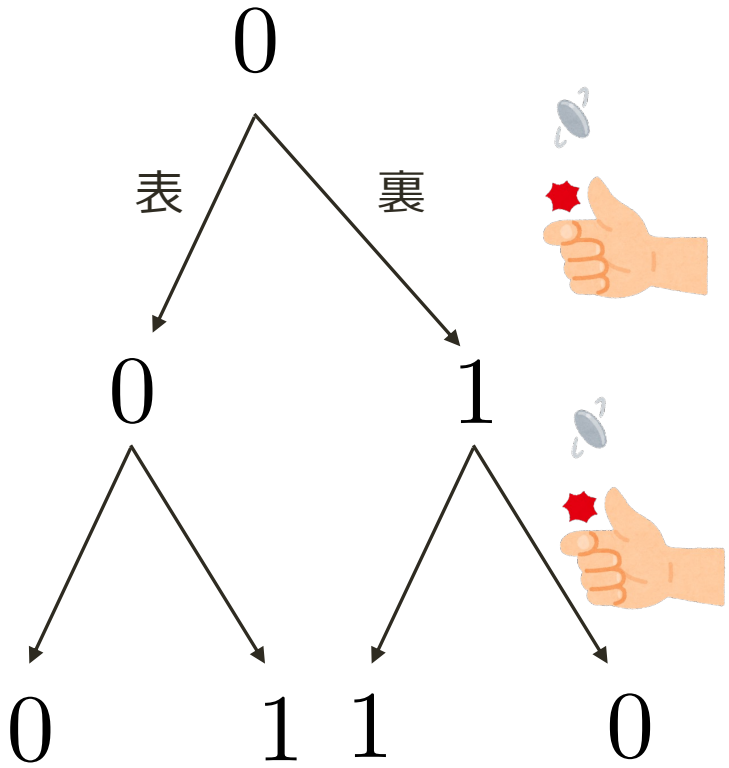
量子コンピュータ vs. 古典コンピュータ

コインフリップ

量子コンピュータの場合

$$|0\rangle \xrightarrow{H} \frac{|0\rangle + |1\rangle}{\sqrt{2}} \xrightarrow{H} |0\rangle$$

古典コンピュータの場合



量子コンピュータでは、
不都合な結果(i.e., 1)を消せる!!

量子コンピュータの優位性

量子重ね合わせを利用することで、以下の問題が
(既知の)最良の古典アルゴリズムよりも効率よく解ける:

- 素因数分解 [P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).]
- 離散対数の発見 [P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).]
- 隠れ部分群問題 [A. Y. Kitaev, arXiv:quant-ph/9511026 (1995).]
- Jones多項式の近似 [D. Aharonov, V. Jones, and Z. Landau, Algorithmica **55**, 395 (2009).]
- 分配関数の近似 [A. Matsuo, K. Fujii, and N. Imoto, Phys. Rev. A **90**, 022304 (2014).]
- (構造のない)データベース探索 [L. K. Grover, Phys. Rev. Lett. **79**, 325 (1997).]
- 可解群の位数の計算 [J. Watrous, in *Proc. of the 33rd STOC* (ACM, Crete, 2001), p. 60.]
- 周期発見 [D. R. Simon, SIAM J. Comput. **26**, 1474 (1997).]
- 連立方程式 [A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).]
- 衝突発見 [G. Brassard, P. HØyer, and A. Tapp, in *Proc. of the LATIN'98* (Springer, Campinas, 1998), p. 163.]

etc...

量子コンピュータの優位性

量子重ね合わせを利用することで、以下の問題が
(既知の)最良の古典アルゴリズムよりも効率よく解ける:

- 素因数分解 [P. W. Shor, SIAM J
- 離散対数の発見 [P. W. Shor,
- 隠れ部分群問題 [A. Y. Kitaev
- Jones多項式の近似 [D. Aha
- 分配関数の近似 [A. Matsuo,
- (構造のない)データベース探索 [L. K. Grover, Phys. Rev. Lett. **79**, 325 (1997).]
- 可解群の位数の計算 [J. Watrous, in *Proc. of the 33rd STOC* (ACM, Crete, 2001), p. 60.]
- 周期発見 [D. R. Simon, SIAM J. Comput. **26**, 1474 (1997).]
- 連立方程式 [A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).]
- 衝突発見 [G. Brassard, P. HØyer, and A. Tapp, in *Proc. of the LATIN'98* (Springer, Campinas, 1998), p. 163.]

etc...

より詳細を知りたい方は、
Quantum Algorithm Zoo
https://quantumalgorithmzoo.org
Qmedia (上記の和訳版[Y. Suzuki, YT, *et al.*])
https://www.qmedia.jp/algorithm-zoo/

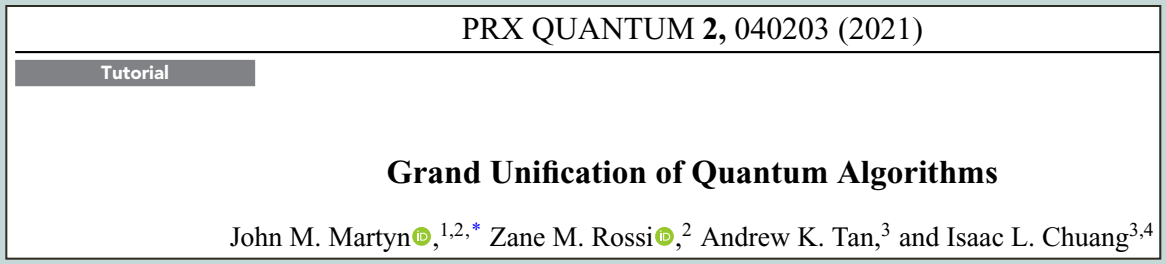
量子コンピュータの優位性

量子重ね合わせを利用することで、以下の問題が
(既知の)最良の古典アルゴリズムよりも効率よく解ける:

- 素因数分解 [P. W. Shor, SIAM J
- 離散対数の発見 [P. W. Shor,
-

より詳細を知りたい方は、
Quantum Algorithm Zoo
algorithmzoo.org
和訳版[Y. Suzuki, YT, *et al.*])
dia.jp/algorithm-zoo/

量子特異値変換(QSVT)


PRX QUANTUM **2,** 040203 (2021)

Tutorial

Grand Unification of Quantum Algorithms

John M. Martyn,[1,2,*] Zane M. Rossi,[2] Andrew K. Tan,[3] and Isaac L. Chuang[3,4]

電子情報通信学会誌2021年11月号:
早川さん、森前先生の解説

情報処理学会誌2022年6月号:
藤井先生の解説

Lett. **79**, 325 (1997).]

CM, Crete, 2001), p. 60.]

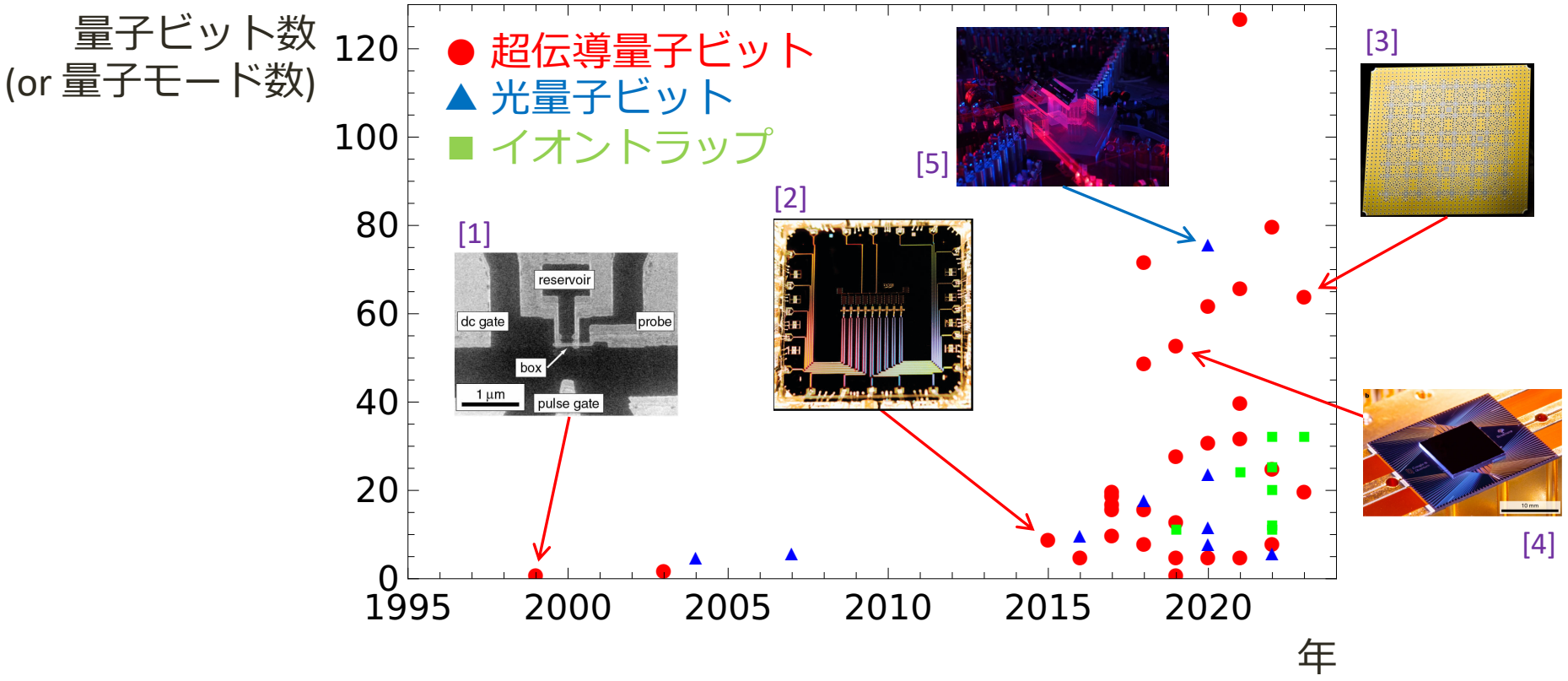
tt. **103**, 150502 (2009).]

98 (Springer, Campinas, 1998), p. 163.]

etc...

量子コンピュータの開発状況

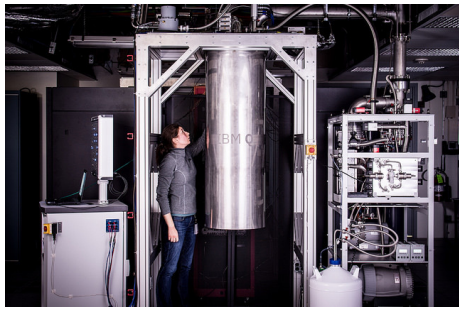
※全ての量子コンピュータが含まれている訳ではない



[1] Y. Nakamura, Y. A. Pashkin, and J. S. Tsai, Nature **398**, 786 (1999).
[2] J. Kelly *et al.*, Nature **519**, 66 (2015). [3] https://monoist.itmedia.co.jp/mn/articles/2304/07/news068.html
[4] F. Arute *et al.*, Nature **574**, 505 (2019). [5] https://www.nature.com/articles/d41586-020-03434-7

量子コンピュータの欠点

IBMの
量子コンピュータ [1]



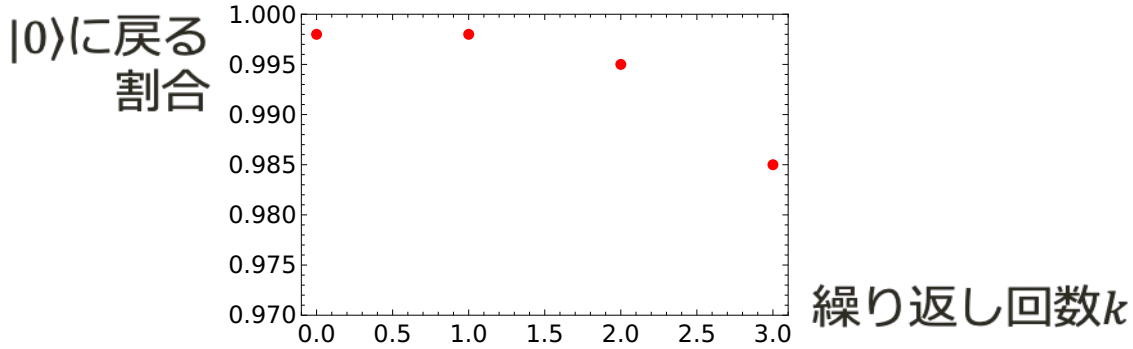
(現在の)実装方法の欠点

- サイズが大きい
- 超伝導量子ビットなどの場合、極低温(e.g., <50mK)が必要
- 定期的なメンテナンスが必要
- エラー確率が(古典コンピュータに比べて)高く、
  計算時間の増加に伴い不正な計算結果の割合が増えてしまう

27℃は300K

例) 量子コインフリップを量子コンピュータ実機(ibm_osaka)で実行

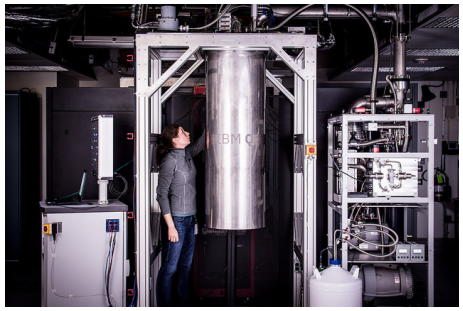
$$\left(\begin{array}{c} |0\rangle \\ H \uparrow \downarrow H \\ \dfrac{|0\rangle + |1\rangle}{\sqrt{2}} \end{array} \right) \times k$$



[1] https://www.popularmechanics.com/technology/a29105270/most-powerful-quantum-computer/

量子コンピュータの欠点

IBMの
量子コンピュータ [1]



(現在の)実装方法の欠点

- サイズが大きい
- 超伝導量子ビットなどの場合、極低温(e.g., <50mK)が必要
- 定期的なメンテナンスが必要
- エラー確率が(古典コンピュータに比べて)高く、
  計算時間の増加に伴い不正な計算結果の割合が増えてしまう

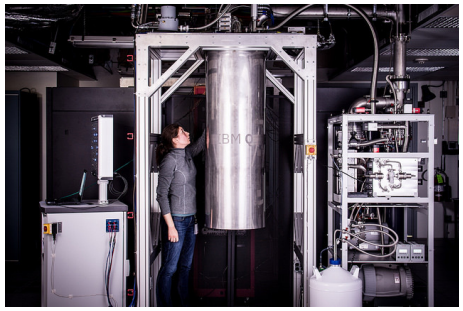
27℃は300K

個人での所有は困難 & 計算結果の正否判定が重要

[1] https://www.popularmechanics.com/technology/a29105270/most-powerful-quantum-computer/

量子コンピュータの欠点

IBMの
量子コンピュータ [1]



(現在の)実装方法の欠点

- サイズが大きい
- 超伝導量子ビットなどの場合、極低温(e.g., <50mK)が必要
- 定期的なメンテナンスが必要
- エラー確率が(古典コンピュータに比べて)高く、
  計算時間の増加に伴い不正な計算結果の割合が増えてしまう

27℃は300K

個人での所有は困難 & 計算結果の正否判定が重要

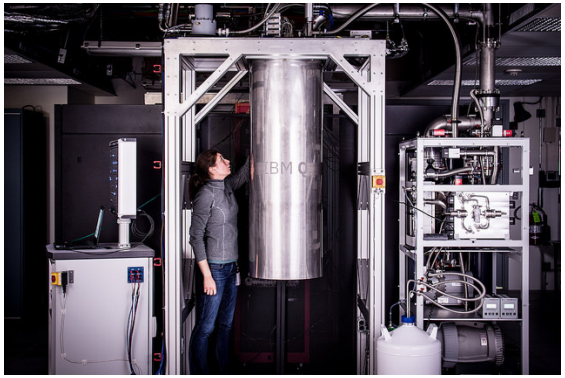
クラウド方式で解決可能！

[1] https://www.popularmechanics.com/technology/a29105270/most-powerful-quantum-computer/

クラウド量子コンピュータ

量子コンピュータを遠隔地のサーバに配置し、
ユーザは必要な時だけアクセスして使用する運用方法

IBM, AWS (IonQ, D-Wave), Xanadu, Origin Quantum等が実現済み



サーバ
(量子コンピュータ)

量子回路の記述(=解きたい問題)

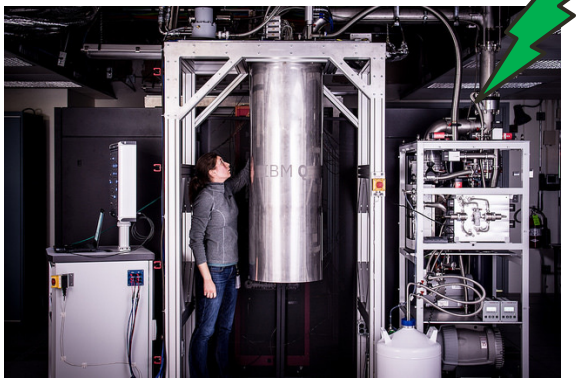
答え

ユーザ

クラウド量子コンピュータ

量子コンピュータを遠隔地のサーバに配置し、
ユーザは必要な時だけアクセスして使用する運用方法

IBM, AWS (IonQ, D-Wave), Xanadu, Origin Quantum等が実現済み



サーバ
(量子コンピュータ)

エラー

量子回路の記述(=解きたい問題)

答え

ユーザ

問題点

ユーザは答えが正しいか分からない

クラウド量子コンピュータ

量子コンピュータを遠隔地のサーバに配置し、
ユーザは必要な時だけアクセスして使用する運用方法

IBM, AWS (IonQ, D-Wave), Xanadu, Origin Quantum等が実現済み

不正なサーバ
(量子コンピュータ)

量子回路の記述(=解きたい問題)

ワザと不正な答えを送信

セキュリティ的モチベーション

ユーザ

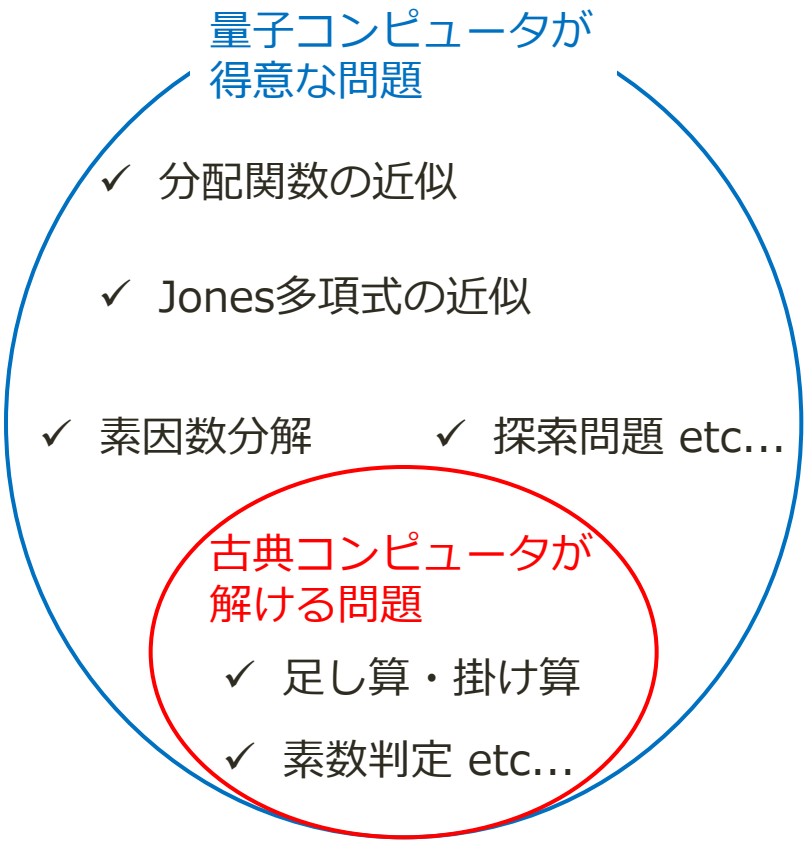
問題点

ユーザは答えが正しいか分からない

2. 量子計算の検証

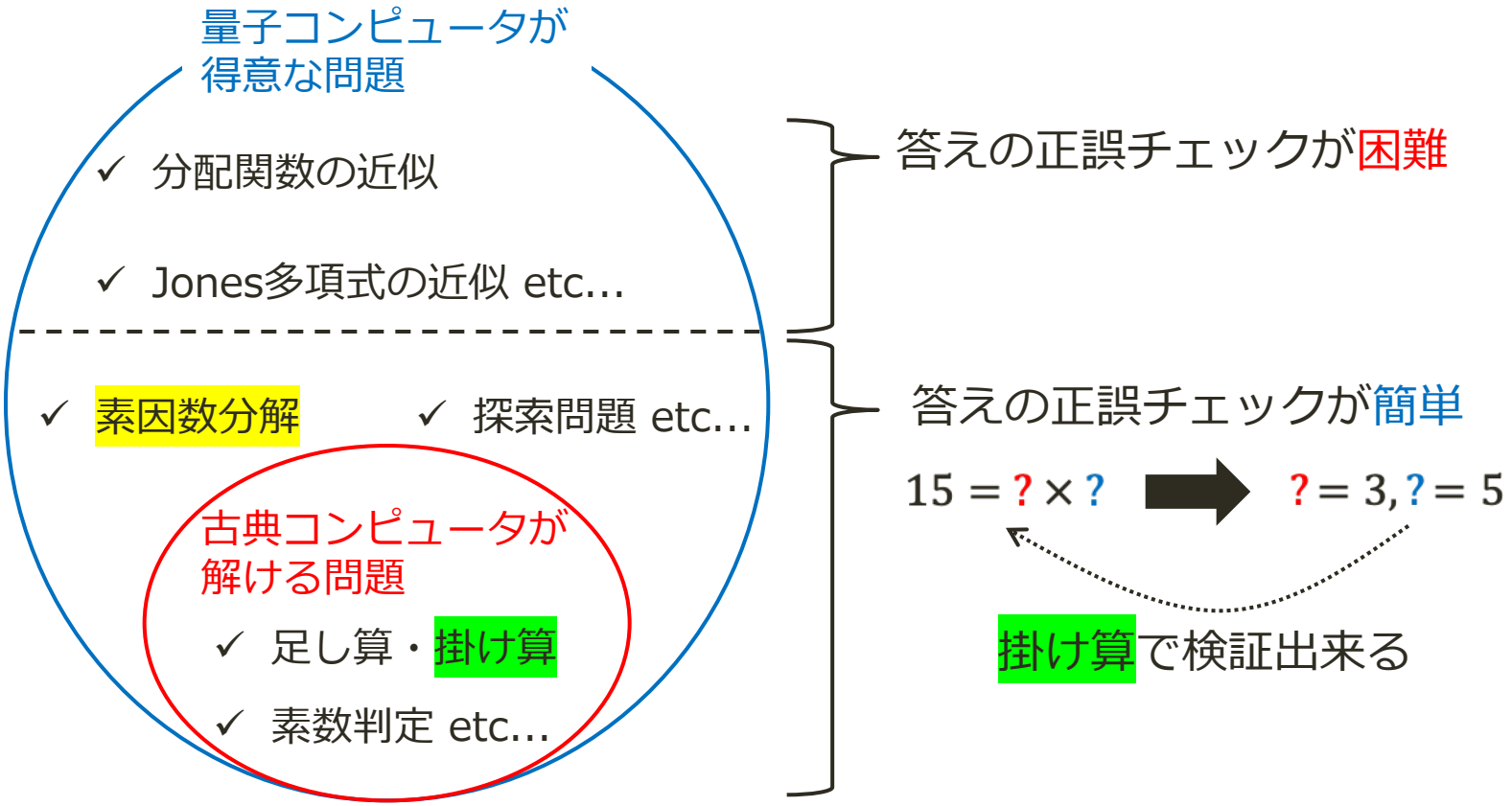
量子計算の検証が困難な理由①

量子コンピュータが解ける問題の中には逆算が難しい問題が含まれている。(より厳密には、NPに含まれなさそうな問題がある。)



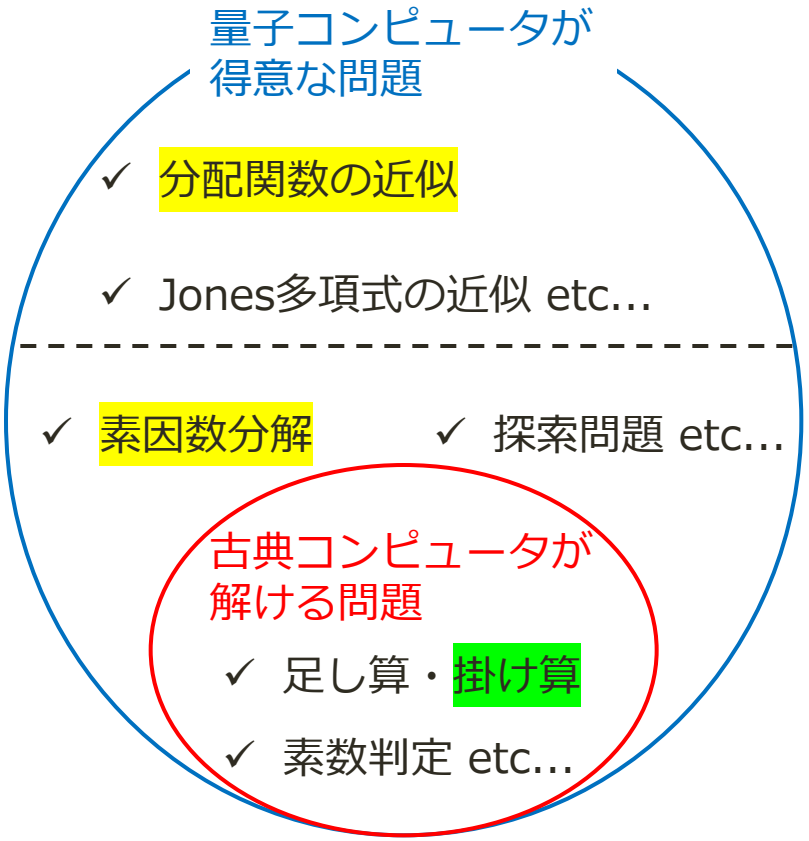
量子計算の検証が困難な理由①

量子コンピュータが解ける問題の中には逆算が難しい問題が含まれている。(より厳密には、NPに含まれなさそうな問題がある。)



量子計算の検証が困難な理由①

量子コンピュータが解ける問題の中には逆算が難しい問題が含まれている。(より厳密には、NPに含まれなさそうな問題がある。)



答えの正誤チェックが困難

$$\mathrm{Tr}\left[\exp\left(-\beta H\right)\right] \simeq 0.1$$

エルミート行列

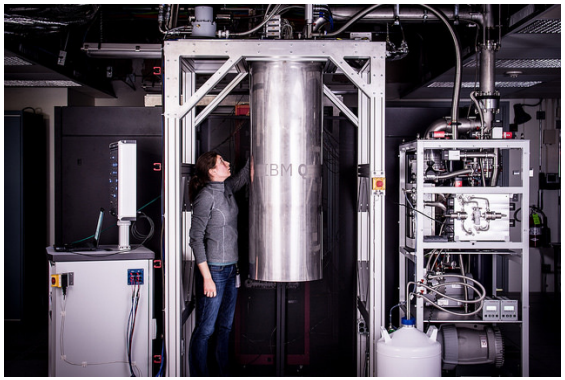
答えの正誤チェックが簡単

$15 = ? \times ?$ ➡ $? = 3, ? = 5$

掛け算で検証出来る

量子計算の検証が困難な理由②

- 検算



サーバ
(量子コンピュータ)

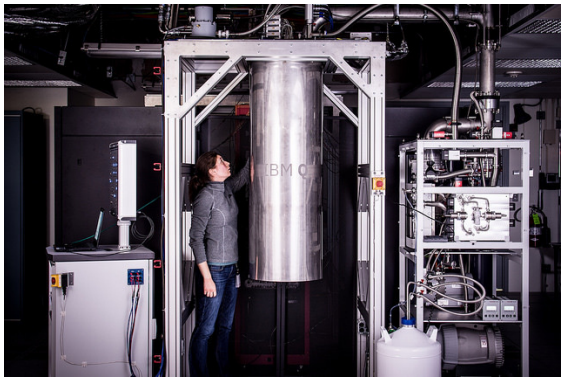
$$\mathrm{Tr}\left[\exp\left(-\beta H\right)\right] \simeq 0.1$$

ユーザ

量子コンピュータの計算を、
古典コンピュータ上で再現してみよう

量子計算の検証が困難な理由②

- 検算: **スケーラブルではない**



サーバ
(量子コンピュータ)

$$\mathrm{Tr}\left[\exp\left(-\beta H\right)\right] \simeq 0.1$$

ユーザ

量子コンピュータの計算を、
古典コンピュータ上で再現してみよう

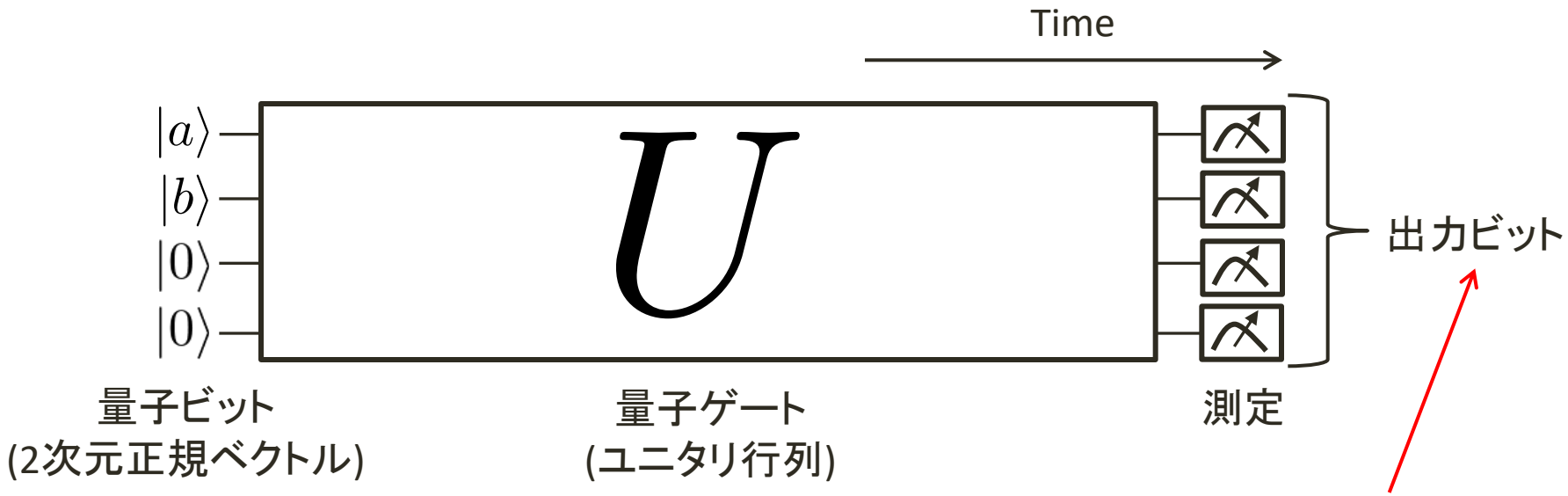
↳ n量子ビットの重ね合わせを
表現するのに、$\Omega(2^n)$ビット必要☹

理由

量子重ね合わせを古典コンピュータで再現するのは困難

量子計算の検証

- 解決策: ビットに変換される直前で検証を行う

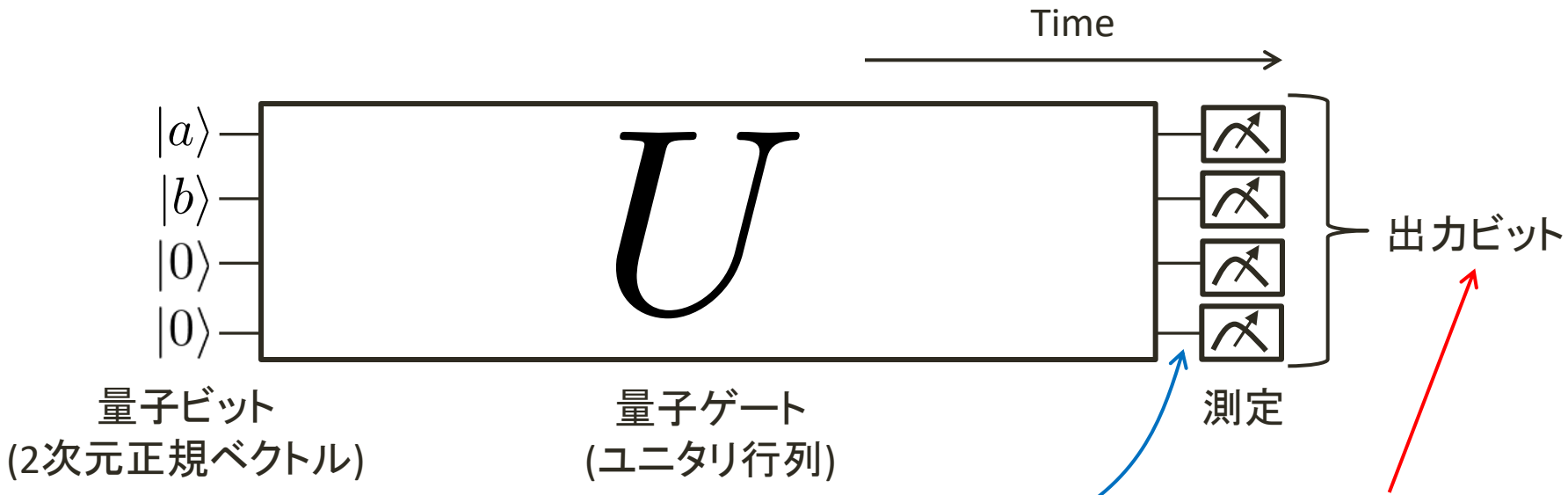


検証が難しい

e.g., [D. Hangleiter, M. Kliesch, J. Eisert, and C. Gogolin, Phys. Rev. Lett. **122**, 210502 (2019).]

量子計算の検証

- 解決策: ビットに変換される直前で検証を行う



検証が難しい

量子ビットの時点で
検証を行えば良い！
(量子状態の検証)

e.g., [D. Hangleiter, M. Kliesch, J. Eisert, and C. Gogolin, Phys. Rev. Lett. **122**, 210502 (2019).]

量子状態の近さ

■ 忠実度(Fidelity)

[M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (2000).]

2つのn量子ビット状態$|\psi\rangle$と$|\phi\rangle$の忠実度$0 \leq F \leq 1$は

$$F \equiv |\langle\phi|\psi\rangle|^2$$

2つのベクトルの内積

- Fが大きい → 2つの量子状態は近い
- Fが小さい → 2つの量子状態は遠い

量子状態の近さ

■ 忠実度(Fidelity)

[M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (2000).]

2つのn量子ビット状態$|\psi\rangle$と$|\phi\rangle$の忠実度$0 \leq F \leq 1$は

$$F \equiv |\langle\phi|\psi\rangle|^2$$

2つのベクトルの内積

- Fが大きい → 2つの量子状態は近い
- Fが小さい → 2つの量子状態は遠い

$$|\psi\rangle \xrightarrow{\text{測定}} \{p_z\}_{z\in\{0,1\}^n}$$

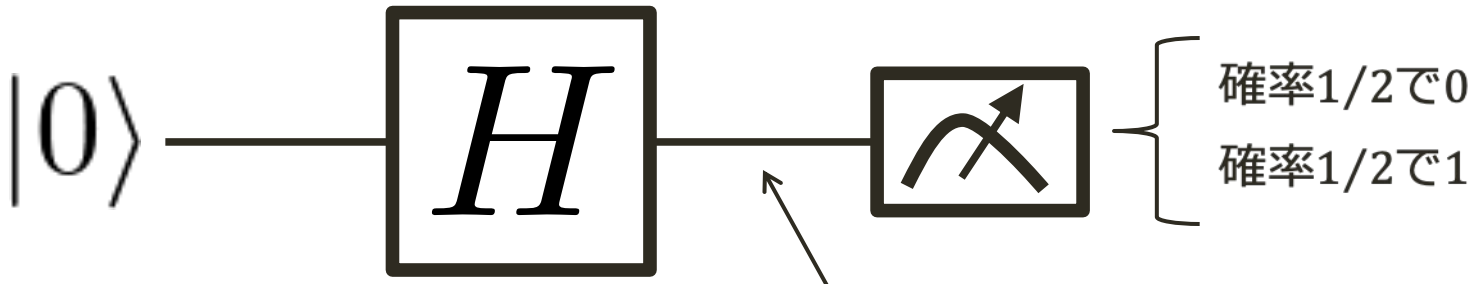
$$|\phi\rangle \xrightarrow{\text{測定}} \{q_z\}_{z\in\{0,1\}^n}$$

Fuchs-van de Graaf不等式

$$\sum_{z\in\{0,1\}^n} |p_z - q_z| \leq 2\sqrt{1-F}$$

量子状態の検証の具体例: 乱数検証

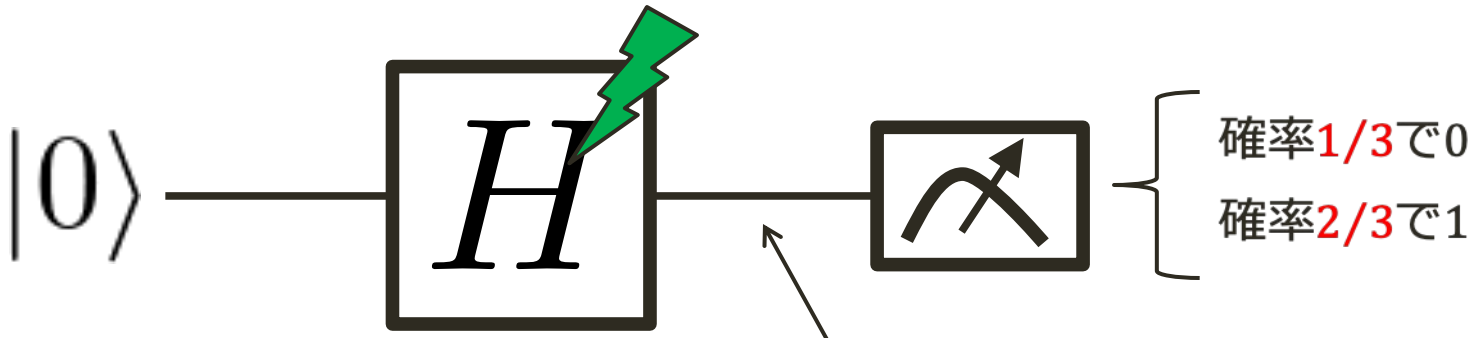
以下の量子回路で1ビットの乱数を生成出来る



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

量子状態の検証の具体例: 乱数検証

以下の量子回路で1ビットの乱数を生成出来る



$$|\psi\rangle = \frac{|0\rangle + \sqrt{2}|1\rangle}{\sqrt{3}}$$

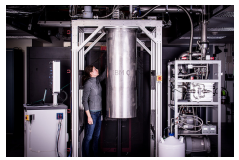
正しい乱数とエラーが発生した乱数の確率分布の距離は

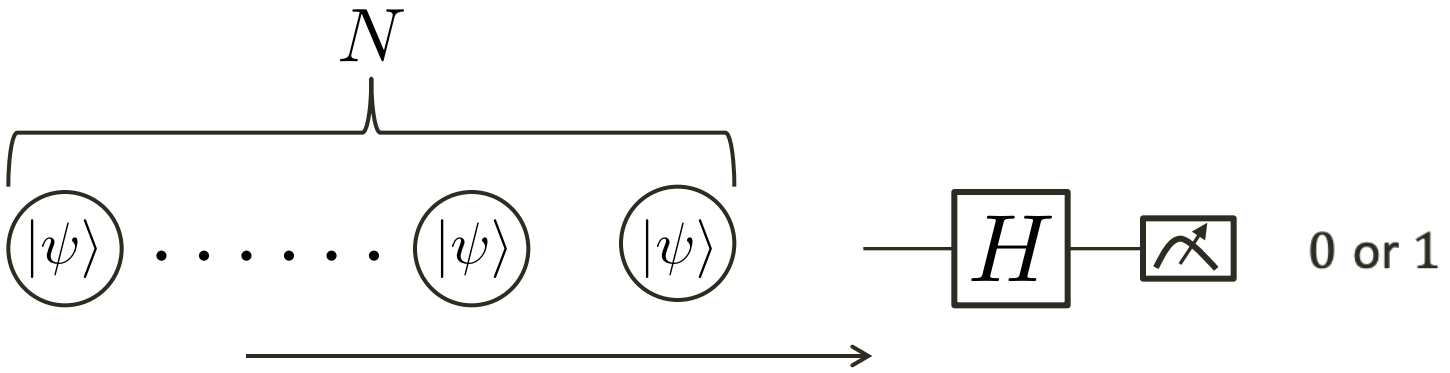
$$\left|\frac{1}{2} - \frac{1}{3}\right| + \left|\frac{1}{2} - \frac{2}{3}\right| = \frac{1}{3} \leq 2\sqrt{1 - F} \simeq 0.97$$

量子状態の検証の具体例: 乱数検証

忠実度の測り方

Hをかけた時に$|0\rangle$に戻る割合M_0を計算する
(量子状態に対する検算)



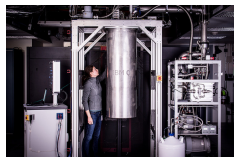


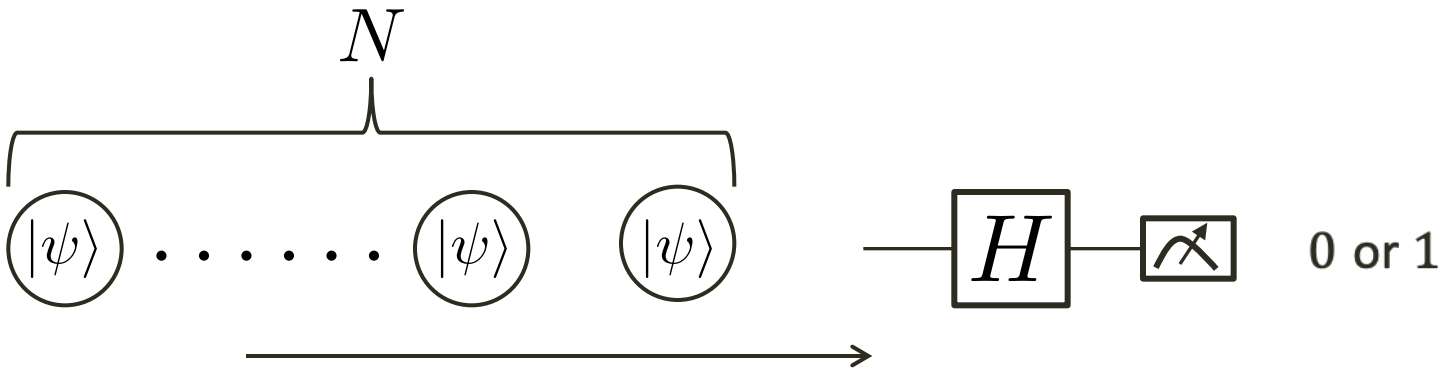
$$\lim_{N\to\infty} M_0 = F$$

量子状態の検証の具体例: 乱数検証

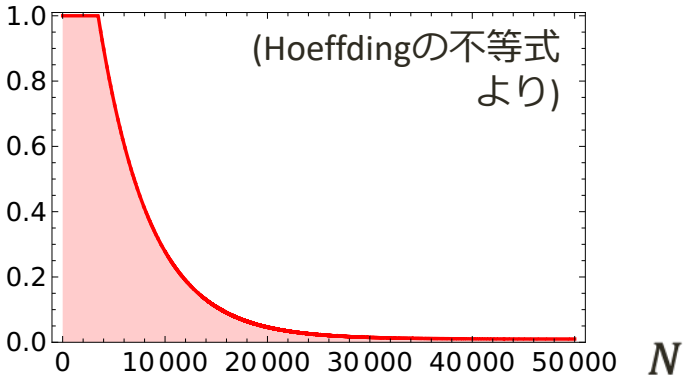
忠実度の測り方

Hをかけた時に$|0\rangle$に戻る割合M_0を計算する
(量子状態に対する検算)





$|F - M_0|$の平均の上限

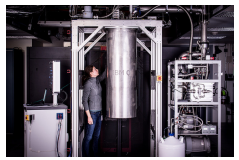


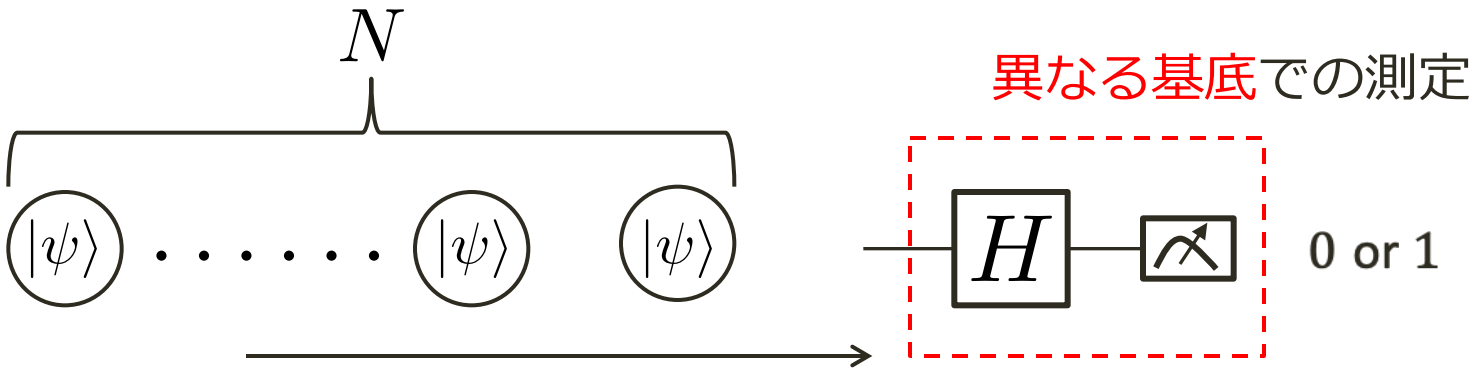
$$\lim_{N\to\infty} M_0 = F$$

量子状態の検証の具体例: 乱数検証

忠実度の測り方

Hをかけた時に$|0\rangle$に戻る割合M_0を計算する
(量子状態に対する検算)





計算結果を得るための測定基底と異なる基底を用いれば
効率的な検証が行える

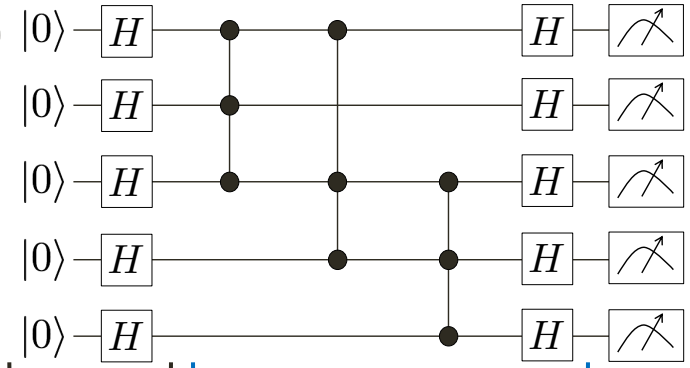
我々の結果: 量子乱数の検証

[YT and T. Morimae, Phys. Rev. X **8**, 021060 (2018).]
[M. Hayashi and YT, New J. Phys. **21**, 093060 (2019).]

可換量子回路(IQP回路)を用いて、
あらゆる古典コンピュータで生成不可能※な乱数が生成出来る
(※ 計算機科学で信じられているあるconjectureのもとで)

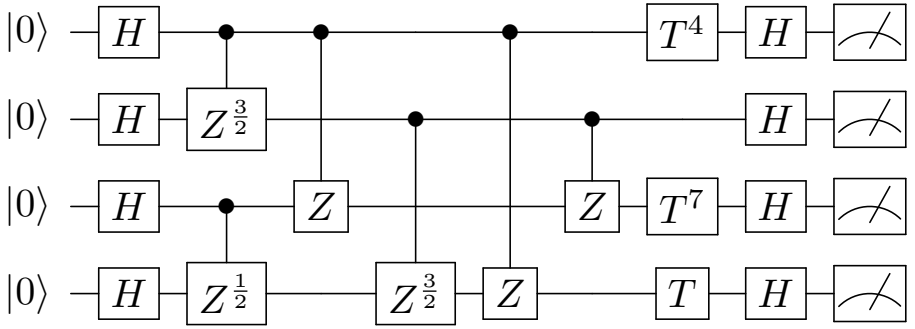
[M. J. Bremner, A. Montanaro, and D. J. Shepherd, Phys. Rev. Lett. **117**, 080501 (2016).]

(1)



前スライドの乱数生成　　量子優位性のための工夫

(2)



前スライドの乱数生成　　量子優位性のための工夫

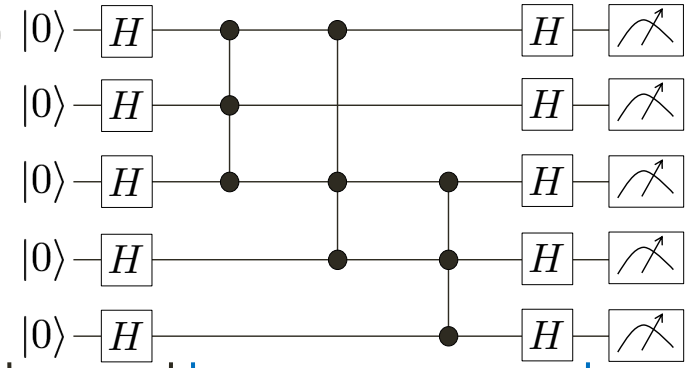
我々の結果: 量子乱数の検証

[YT and T. Morimae, Phys. Rev. X **8**, 021060 (2018).]
[M. Hayashi and YT, New J. Phys. **21**, 093060 (2019).]

可換量子回路(IQP回路)を用いて、
あらゆる古典コンピュータで生成不可能※な乱数が生成出来る
(※ 計算機科学で信じられているあるconjectureのもとで)

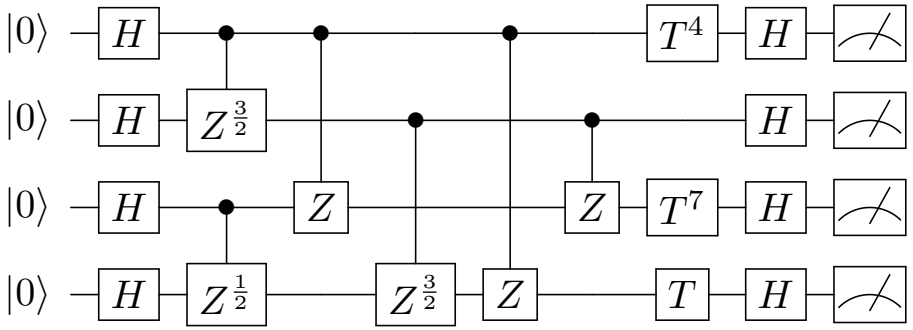
[M. J. Bremner, A. Montanaro, and D. J. Shepherd, Phys. Rev. Lett. **117**, 080501 (2016).]

(1)

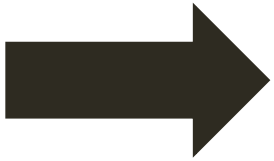


前スライドの乱数生成　　量子優位性のための工夫

(2)



前スライドの乱数生成　　量子優位性のための工夫



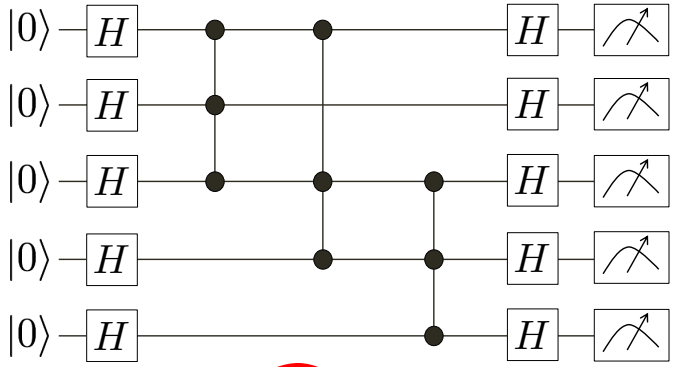
測定直前にエンタングル状態になるため、
1量子ビットずつ測定して忠実度を測ることが困難に

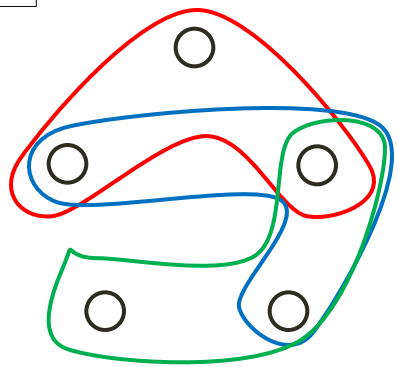
我々の結果: 量子乱数の検証

[YT and T. Morimae, Phys. Rev. X **8**, 021060 (2018).]
[M. Hayashi and YT, New J. Phys. **21**, 093060 (2019).]

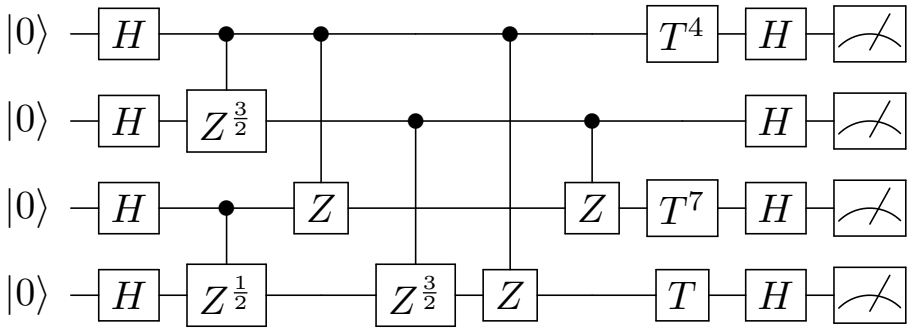
解決策: 量子ビット間の相関を
(1)ハイパーグラフ, (2)重み付きグラフ
として記述することで、1ビット乱数の検証に帰着

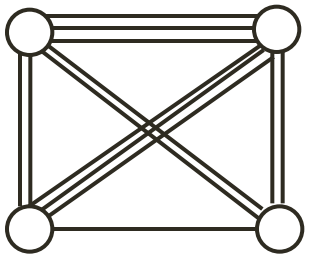
(1)





(2)



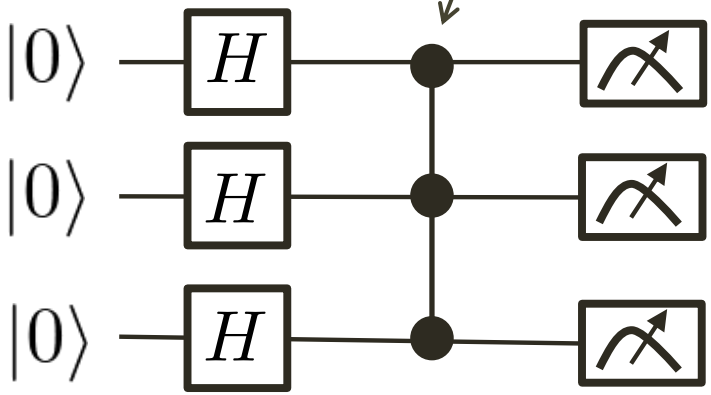


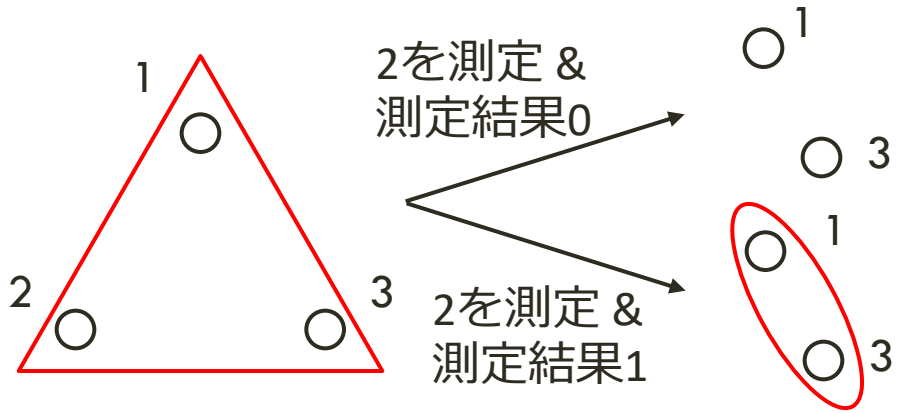
我々の結果: 量子乱数の検証

[YT and T. Morimae, Phys. Rev. X **8**, 021060 (2018).]
[M. Hayashi and YT, New J. Phys. **21**, 093060 (2019).]

具体例: 3量子ビットIQP回路(1)の場合

CCZ
$\equiv (I \otimes I \otimes H)\mathrm{Toffoli}(I \otimes I \otimes H)$





相関が無くなり、簡単に検証可能！

相関が残ってしまう☹

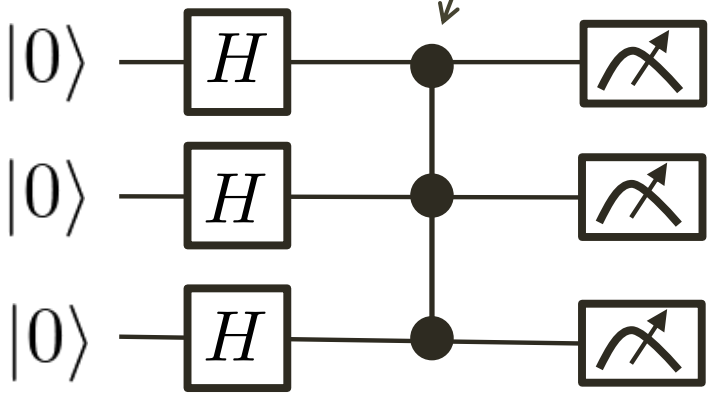
[2] M. Hayashi and T. Morimae, Phys. Rev. Lett. **115**, 220502 (2015).

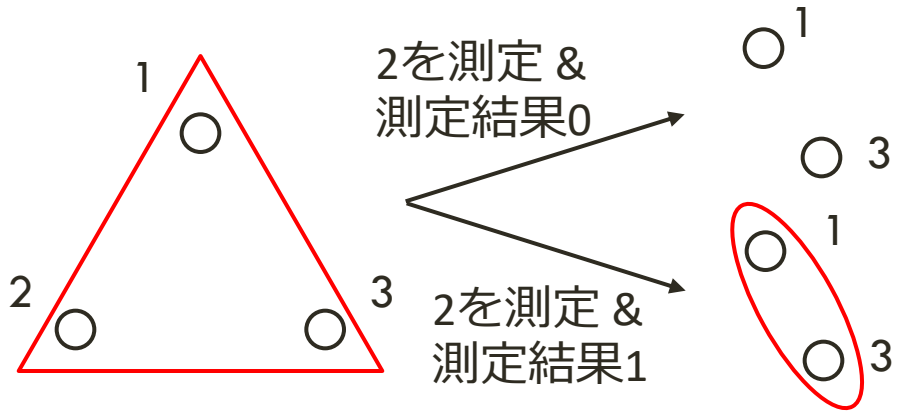
我々の結果: 量子乱数の検証

[YT and T. Morimae, Phys. Rev. X **8**, 021060 (2018).]
[M. Hayashi and YT, New J. Phys. **21**, 093060 (2019).]

具体例: 3量子ビットIQP回路(1)の場合

CCZ
$\equiv (I \otimes I \otimes H)\mathrm{Toffoli}(I \otimes I \otimes H)$





相関が無くなり、簡単に検証可能！

相関が残ってしまう☹

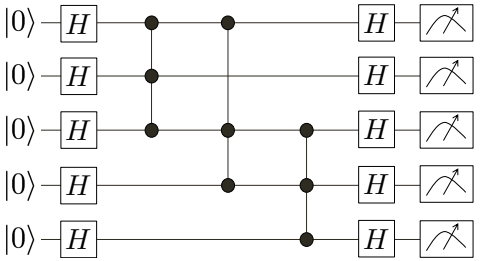
➡ 既存手法(e.g., [2])で検証可能！

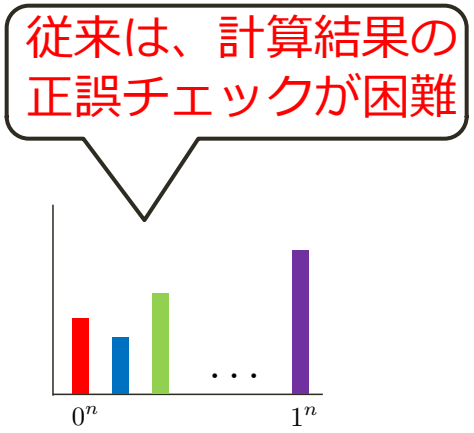
我々の結果: 量子乱数の検証

[YT and T. Morimae, Phys. Rev. X **8**, 021060 (2018).]
[M. Hayashi and YT, New J. Phys. **21**, 093060 (2019).]

■ 検証可能な量子乱数生成

1. 量子コンピュータで、量子乱数に変換される直前の量子状態を生成





2. 我々の手法で、測定直前の量子状態が正しいか検証

忠実度が高い場合　→　量子状態を測定し、計算結果を出力
忠実度が低い場合　→　量子状態を破棄 & 生成しなおす

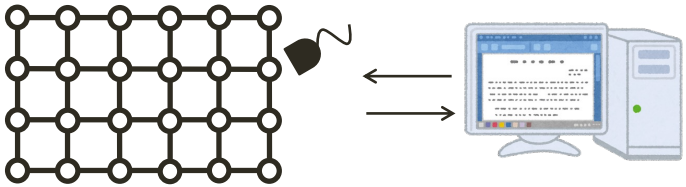
エラーの影響を受けない量子計算が可能に☺

その他の我々の結果

様々な量子計算モデル(量子コンピュータアーキテクチャ)
に対して検証手法を提案
(詳細は**NTT技術ジャーナル　2023年8月号**をご覧ください)

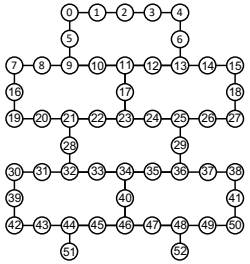
1. 測定型量子計算の検証手法を改善
   [YT, A. Mantri, T. Morimae, A. Mizutani, and J. F. Fitzsimons, npj Quantum Inf. **5**, 27 (2019).]

   光量子コンピュータに
   適したモデル



2. NISQ(ノイズがある小・中規模の量子)
   コンピュータの検証
   [YT, Y. Takahashi, T. Morimae, and S. Tani, Quantum **6**, 758 (2022).]

IBMの量子コンピュータ
チップの模式図



3. 誤り耐性量子コンピュータに重要な構成要素の検証
   [A. Mizutani, YT, R. Hiromasa, Y. Aikawa, and S. Tani, Phys. Rev. A **106**, L010601 (2022).]

[3] S. T. Flammia and Y.-K. Liu, Phys. Rev. Lett. **106**, 230501 (2011).

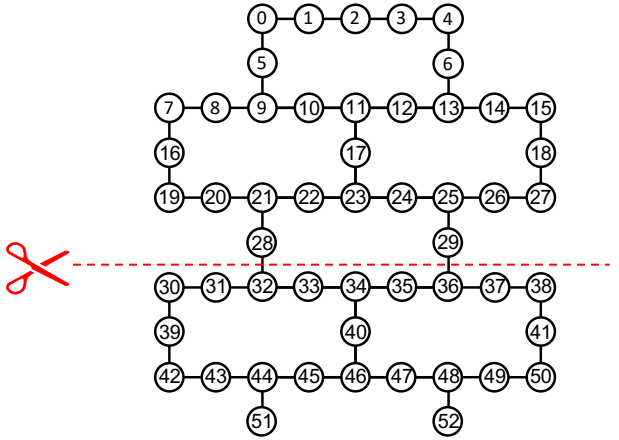
その他の我々の結果

- NISQ(ノイズがある小・中規模の量子)コンピュータの検証

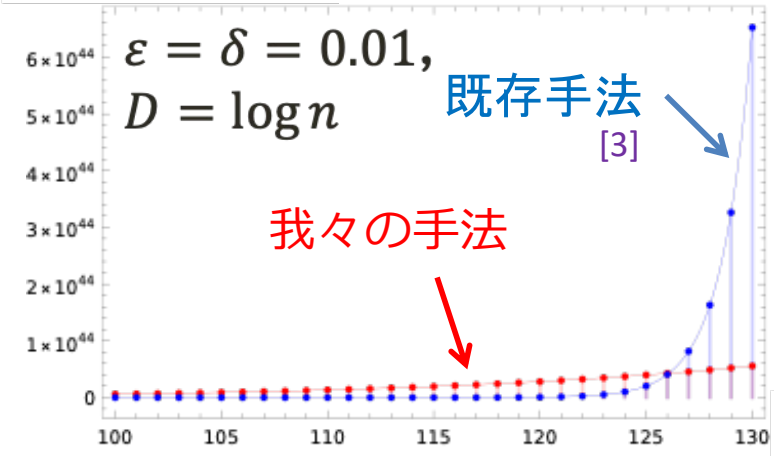
[YT, Y. Takahashi, T. Morimae, and S. Tani, Quantum **6**, 758 (2022).]

ε: 検証精度、 δ: 検証に失敗する確率
D: 2分割した出力にまたがる量子ゲートの数
NISQコンピュータの場合、$D = O(\log n)$

$$S = O\left(\frac{2^{12D}}{\varepsilon^6}\left(D + \log\frac{1}{\delta\varepsilon^4}\right)^3\right)$$



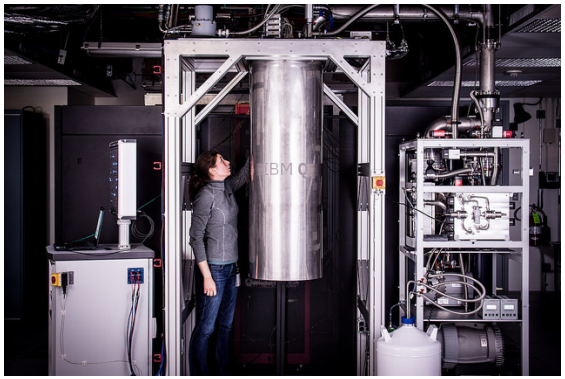
サンプル数S



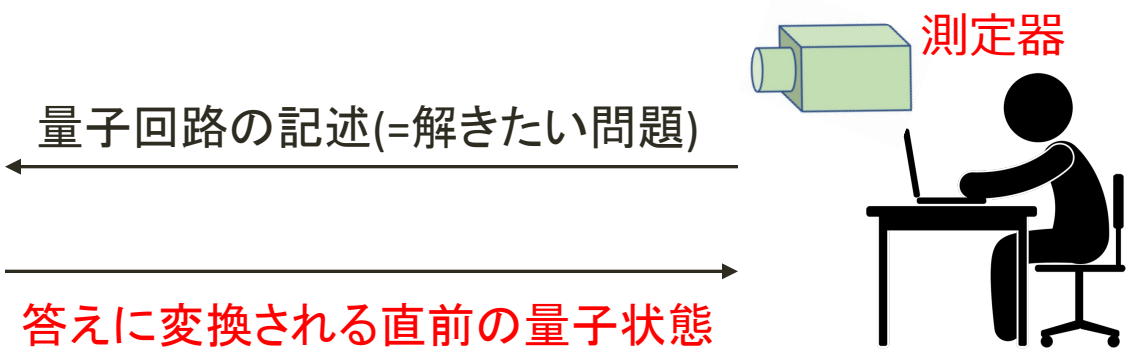
130量子ビットの場合、我々の手法は
800万倍以上少ないサンプル数で良い

まとめ

- 量子コンピュータは古典コンピュータ以上の計算能力を有することが数学的に示されている
- 「量子計算の検証」というソフトウェア技術を用いることで、信頼性が高いクラウド量子コンピュータシステムが実現出来る



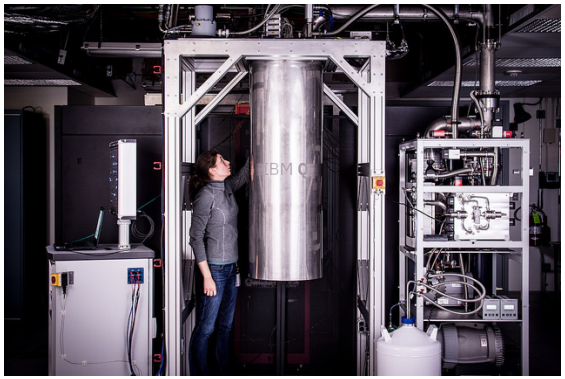
量子コンピュータサーバ



1. 測定器を用いて量子状態の検証を実行
2. 忠実度が高ければ、量子状態を測定し計算結果を出力
   低ければ、サーバの量子コンピュータが正しく動作していないと判定

今後の展望

古典通信のみで量子コンピュータを検証することは可能か？



量子コンピュータサーバ

量子回路の記述(=解きたい問題)

古典ビット？

ユーザ

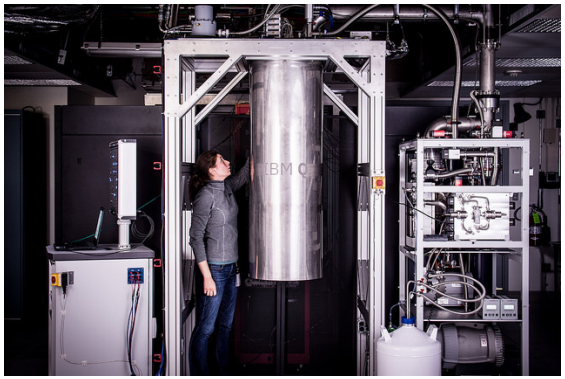
今後の展望

古典通信のみで量子コンピュータを検証することは可能か？

2018年のブレークスルー

計算量的な検証は古典通信のみで効率的に可能

Urmila Mahadev [4]



量子コンピュータサーバ
(多項式時間の量子計算とみなせる
エラーしか発生しない)

量子回路の記述(=解きたい問題)

古典ビット

ユーザ

[4] https://simons.berkeley.edu/people/urmila-mahadev

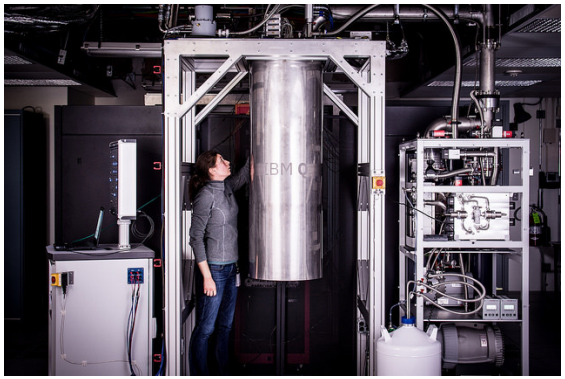
今後の展望

古典通信のみで量子コンピュータを検証することは可能か？

2018年のブレークスルー

計算量的な検証は古典通信のみで効率的に可能

Urmila Mahadev [4]



量子コンピュータサーバ

(多項式時間の量子計算とみなせる
エラーしか発生しない)

量子回路の記述(=解きたい問題)

古典ビット

ユーザ

分野の未解決問題

情報理論的な検証も古典通信で可能？

[4] https://simons.berkeley.edu/people/urmila-mahadev

ご静聴ありがとうございました☺