第4章
ショアのアルゴリズム
完全理解

2025/02/21(金)

千野 浩輔

佐藤研
SATOH Lab.

はじめに
結論（ショアのアルゴリズムを一言で）
位数発見問題とは
位数発見の方法
具体的な手順（量子回路）
アルゴリズムの計算量解析
実際に動かしてみる

はじめに（この講義の目標の共有）

結論（ショアのアルゴリズムを一言で）

結論

- 素因数分解問題を位数発見問題に帰着.
- 位数発見には位相推定サブルーチンを使用.

たったのこれだけ． 以上.

はじめに
結論（ショアのアルゴリズムを一言で）
位数発見問題とは
位数発見の方法
具体的な手順（量子回路）
アルゴリズムの計算量解析
実際に動かしてみる

位数発見問題とは

位数発見問題とは

$$x^r \bmod \mathrm{N} = 1$$

N：素因数分解したい整数

x：Nと互いに素な整数

上記を満たす，最小の整数rを見つけること.

この整数rを位数（order）と呼ぶ.

佐藤研 SATOH Lab.

位数発見問題とは

位数rが見つかると，

$$x^r \bmod \mathrm{N} = 1 \tag{1}$$

$$(x^r - 1) \bmod \mathrm{N} = 0 \tag{2}$$

$$(x^{r/2} + 1)(x^{r/2} - 1) \bmod \mathrm{N} = 0 \tag{3}$$

より，$(x^{\frac{r}{2}} + 1)$または$(x^{\frac{r}{2}} - 1)$のどちらかがNの倍数.

もしくは，　$(x^{\frac{r}{2}} + 1)$，　$(x^{\frac{r}{2}} - 1)$が合わさってNの倍数.

佐藤研 SATOH Lab.

位数発見問題とは

つまり,

$$p = \gcd(x^{r/2} + 1, N) \tag{4}$$
$$q = \gcd(x^{r/2} - 1, N) \tag{5}$$

が整数Nの因数になっている.

上記を繰り返せば，Nを小さな因数へと分解できる.

位数発見問題とは

具体例（$N = 21, x = 11$）

$$x^r \bmod \mathrm{N} = 1 \tag{6}$$

$$11^r \bmod 21 = 1 \implies r = 6 \tag{7}$$

$$\because 11^6 = 1771561 = 84360 \times 21 \cdots 1 \tag{8}$$

より，位数は$r = 6$であることが分かる．ここで，p, qを計算すると

$$p = \gcd(11^{6/2} + 1, 21) = \gcd(1332, 21) = 3 \tag{9}$$

$$q = \gcd(11^{6/2} - 1, 21) = \gcd(1330, 21) = 7 \tag{10}$$

となり，$21 = 3 \times 7$と素因数分解を計算できる．

佐藤研 SATOH Lab.

ここまでのまとめ

・位数を見つけることができる
　＝素因数分解ができる.

・実は，その位数は偶数.

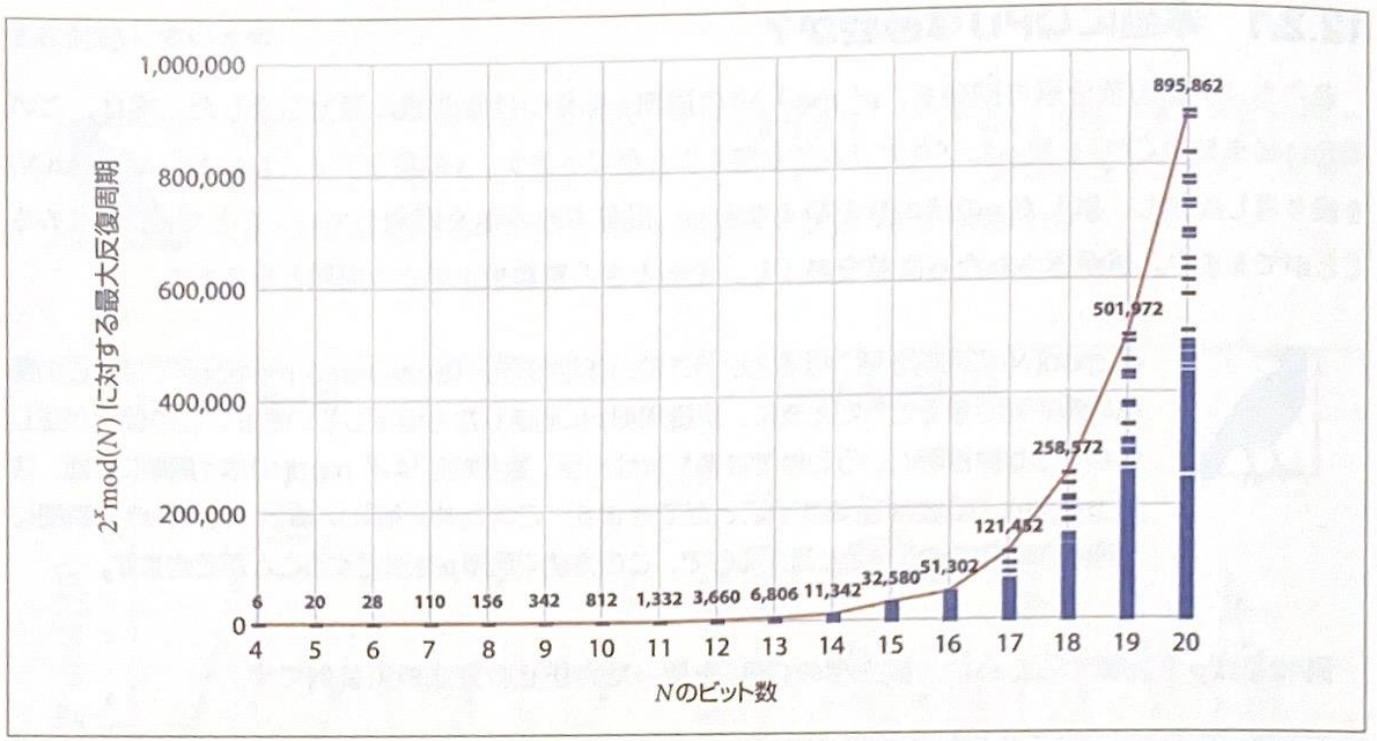
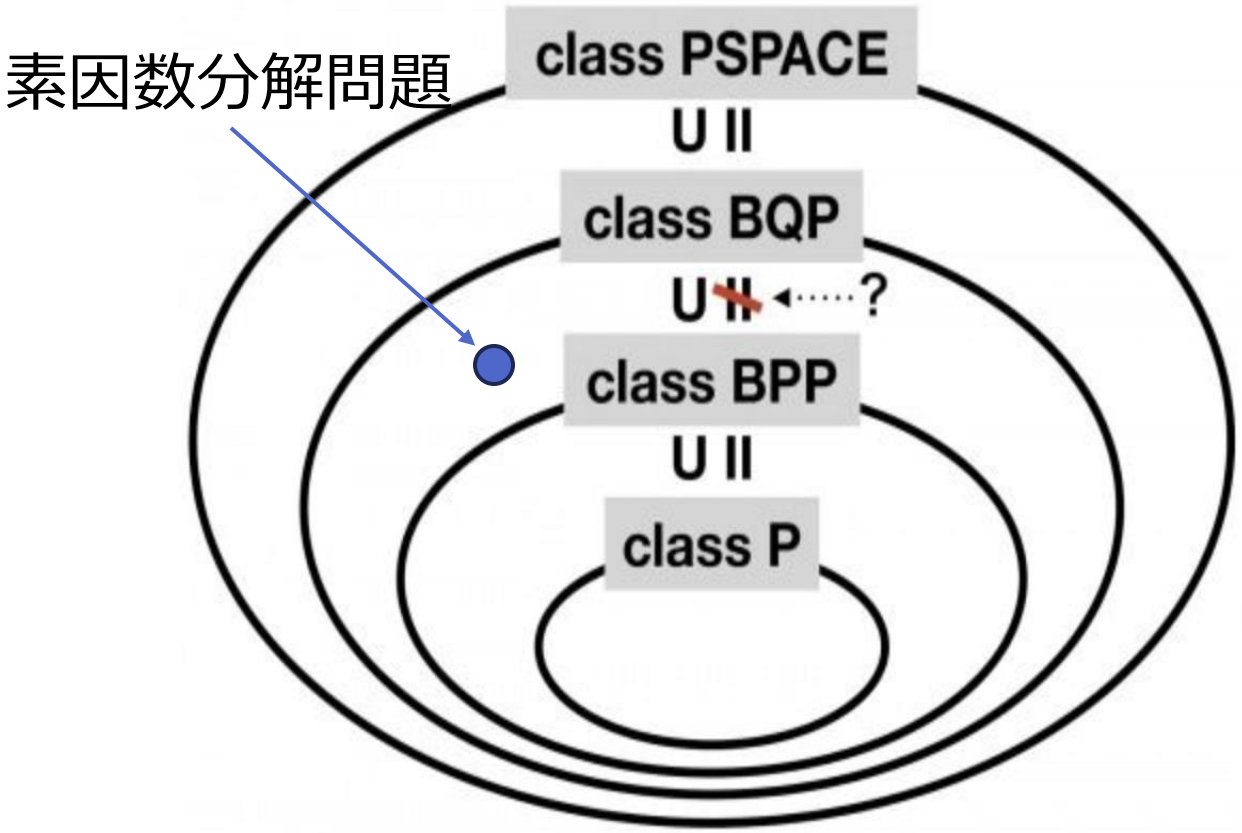
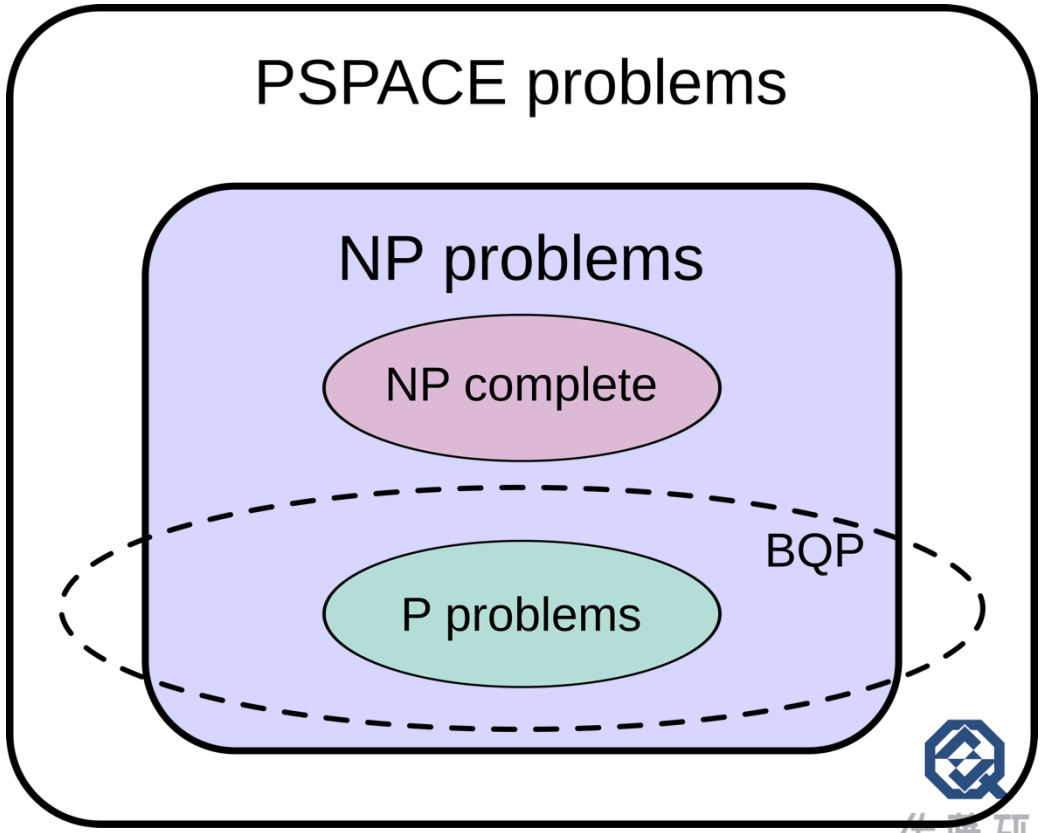


図12-2　長さNのビットストリングで表現される整数の反復周期を見つけるために必要なループの最大数。それぞれの棒は、長さNのビット列で表される整数の反復周期の分布を示すヒストグラムを表す。

参考文献[1]. P.236 図12-2を引用

余談：複雑性クラス


素因数分解問題
class PSPACE
U ||
class BQP
U ||
?
class BPP
U ||
class P


PSPACE problems
NP problems
NP complete
BQP
P problems

はじめに
結論（ショアのアルゴリズムを一言で）
位数発見問題とは
位数発見の方法
具体的な手順（量子回路）
アルゴリズムの計算量解析
実際に動かしてみる

位数発見の方法（ここからショアのアルゴリズム）

位数発見の方法

さて，ではどうやって位数を発見するのか．

位相推定サブルーチンを用いる．

位相推定とは，

$$U|u\rangle = e^{i2\pi\phi}|u\rangle$$

のとき，$\phi(0 \leq \phi < 1)$を求める．

位数発見の方法

今回，着目するユニタリ演算子は，以下で定義されるものである.

$$U_{x,N}\,|y\rangle = |xy\ (\mathrm{mod}\ \mathrm{N})\rangle$$

積の剰余を求めるユニタリ演算子.

佐藤研 SATOH Lab.

位数発見の方法

なぜ，このようなユニタリ演算子を考えるのか．

理由①：$U_{x,N}$をa回作用すると，$|x^a \bmod N\rangle$となるから．

$$U^a|1\rangle = U^{a-1}|x \bmod N\rangle = U^{a-2}|x^2 \bmod N\rangle = \cdots = |x^a \bmod N\rangle$$

位数発見の方法

なぜ，このようなユニタリ演算子を考えるのか.

理由②：この演算子の固有値が都合の良い形だから.

→固有値の位相部分に位数の情報が現れる.

演算子の固有状態

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp(-2\pi i \frac{s}{r} k) \, |x^k \bmod \mathrm{N}\rangle$$
$$0 \leq s \leq r-1$$

演算子の固有値

$$\exp(2\pi i \frac{s}{r})$$

位数発見の方法

(証明)

$$U_{x,N}\left|u_s\right\rangle = U_{x,N}\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\exp(-2\pi i\frac{s}{r}k)\left|x^k \bmod \mathrm{N}\right\rangle \tag{11}$$

$$= \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\exp(-2\pi i\frac{s}{r}k)\left|x^{k+1} \bmod \mathrm{N}\right\rangle \tag{12}$$

$$= \exp(2\pi i\frac{s}{r})\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\exp(-2\pi i\frac{s}{r}(k+1))\left|x^{k+1} \bmod \mathrm{N}\right\rangle \tag{13}$$

$$= \exp(2\pi i\frac{s}{r})\left|u_s\right\rangle \tag{14}$$

$$\therefore U_{x,N}\left|u_s\right\rangle = \exp(2\pi i\frac{s}{r})\left|u_s\right\rangle$$

ただし，$\exp(-2\pi i\frac{s}{r}\cdot 0) = \exp(-2\pi i\frac{s}{r}\cdot r)$，$x^0 \pmod N = x^r \pmod N$を利用.

位数発見の方法

なぜ，このようなユニタリ演算子を考えるのか．

理由③：固有状態を重ね合わせると，$|1\rangle$になるから．
→位相推定を行う前に，位相を知っていなければならない問題を解消．

演算子の固有状態

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp(-2\pi i \frac{s}{r} k) \, |x^k \bmod \mathrm{N}\rangle$$

重ね合わせると

重ね合わせ状態

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

位数発見の方法

(証明)

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle = \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\exp(-2\pi i\frac{s}{r}k)\,|x^k \bmod N\rangle \tag{20}$$

$$= \frac{1}{r}\sum_{k=0}^{r-1}(\sum_{s=0}^{r-1}\exp(-2\pi i\frac{s}{r}k))\,|x^k \bmod N\rangle \tag{21}$$

$$= \frac{1}{r}\cdot r\,|x^0 \bmod N\rangle \tag{22}$$

$$= |1\rangle \tag{23}$$

ただし，　下記の性質を利用.

$$\sum_{s=0}^{r-1}\exp(-2\pi i\frac{s}{r}k) = \begin{cases} r, & k=0 \\ 0, & k\neq 0 \end{cases}$$

佐藤研 SATOH Lab.

位数発見の方法

改めて，なぜこのユニタリを考えるのか．

・複数回作用させることで，問題設定の状態を作れる．

・求めたい位数が，固有値の位相に現れる．

・ユニタリの固有状態を予め知っている必要がない．

→位相を知る術は，位相推定サブルーチンがある．

$$U_{x,N}\,|y\rangle = |xy\ (\mathrm{mod\ N})\rangle \xrightarrow{\text{固有値}} \exp(2\pi i\frac{s}{r})$$

ここまでのまとめ

・当初，素因数分解を位数発見問題に帰着．

・さらに，位数発見問題を$U_{x,N}$の固有値（位相）に発見に帰着．
つまり，$U_{x,N}$に適用された位相推定問題を解く問題と言える．

素因数分解→位数発見→位相推定

佐藤研 SATOH Lab.

はじめに
結論（ショアのアルゴリズムを一言で）
位数発見問題とは
位数発見の方法
具体的な手順（量子回路）
アルゴリズムの計算量解析
実際に動かしてみる

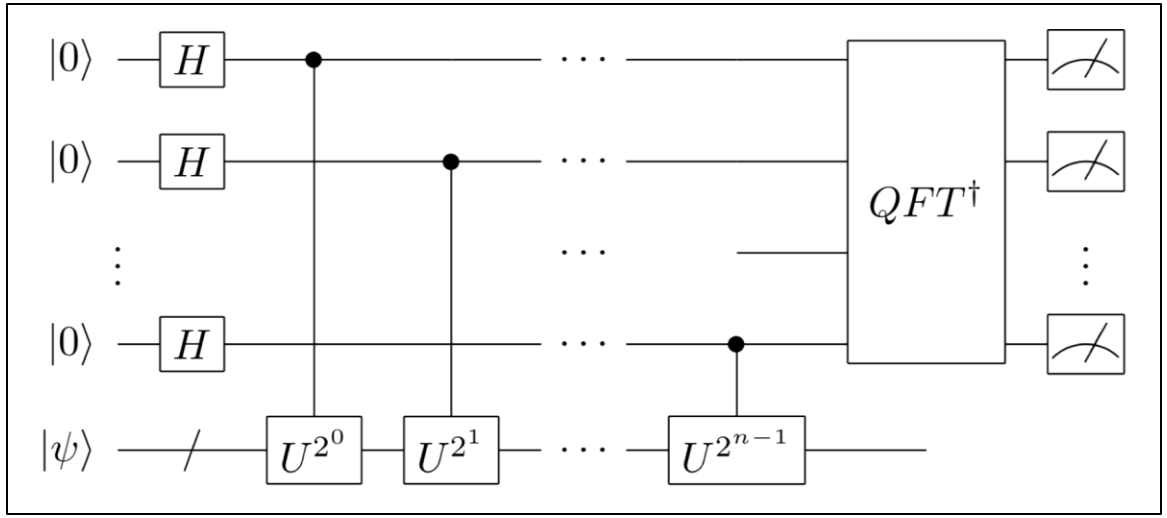
具体的な手順（量子回路）

具体的な手順（量子回路）

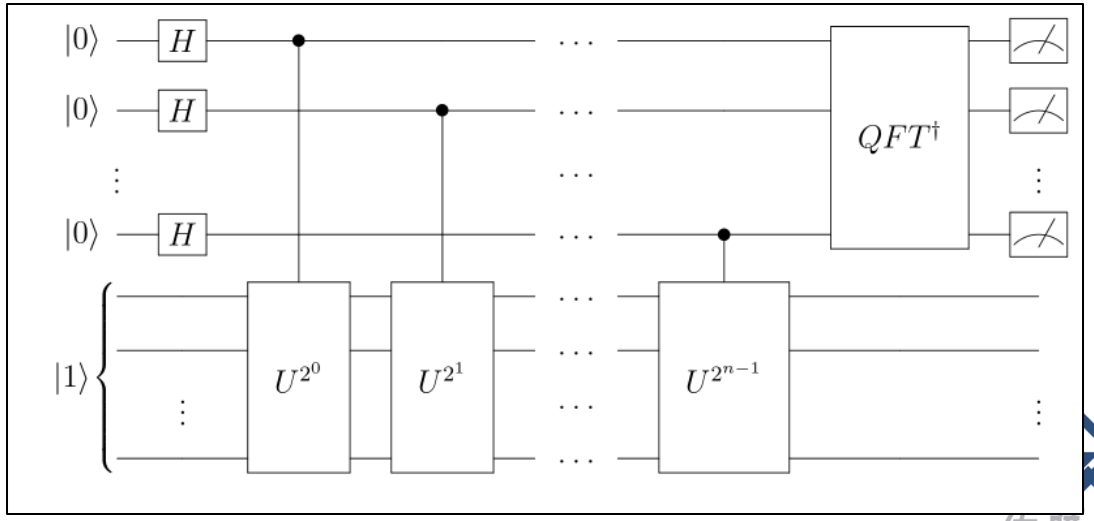
ステップ1

測定レジスタ：$t = 2L + 1 + \log(3 + \frac{1}{2\varepsilon})$個のqbitを$|0\rangle$に初期化.

作業レジスタ：$L = \log N$個のqbitを状態$|1\rangle$に初期化.



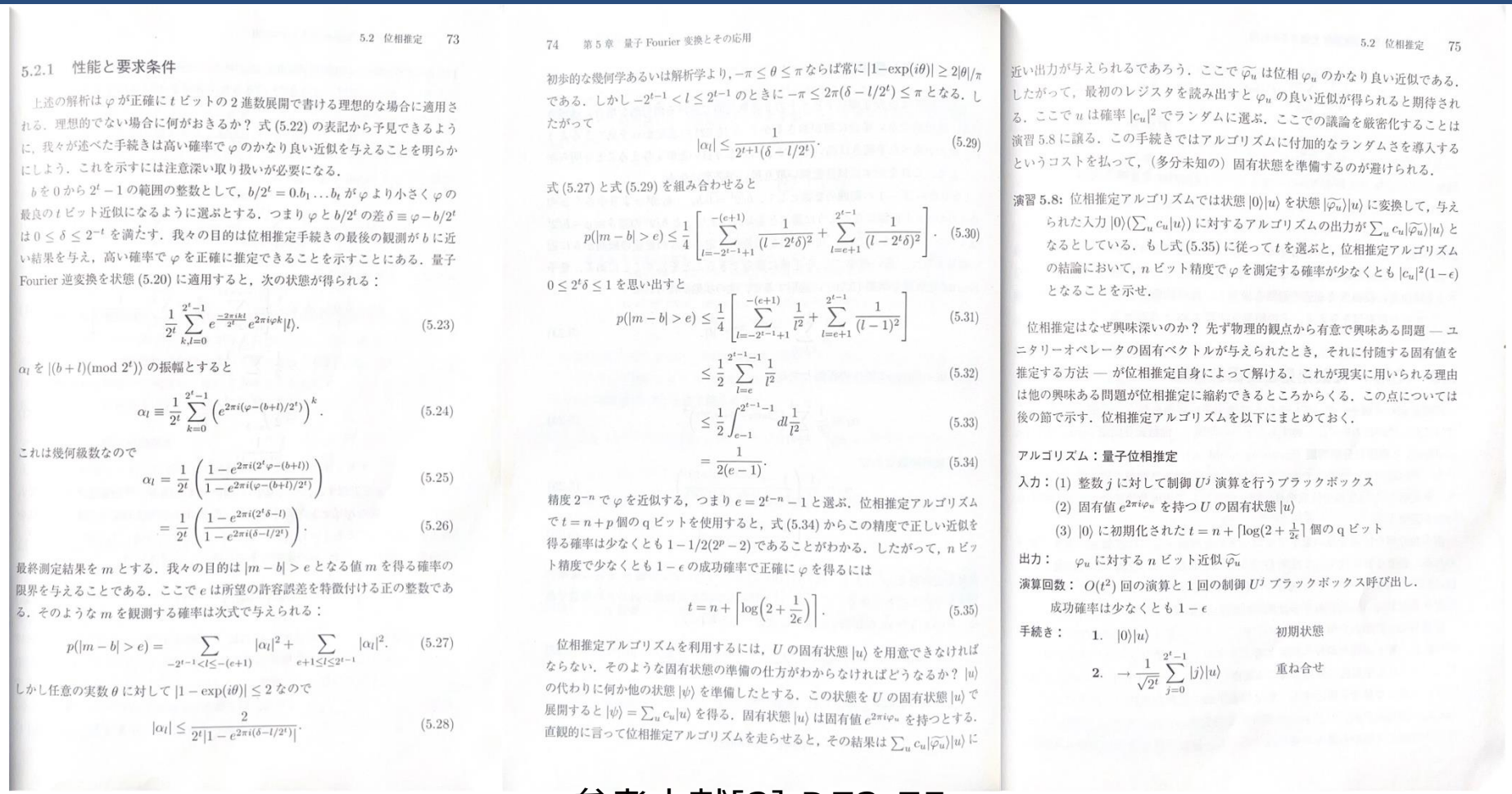
通常の位相推定サブルーチンの回路



問題としている位相推定サブルーチンの回路

具体的な手順（量子回路）

(補足)



5.2 位相推定 73

5.2.1 性能と要求条件

上述の解析は φ が正確に t ビットの 2 進数展開で書ける理想的な場合に適用される．理想的でない場合に何がおきるか？ 式 (5.22) の表記から予見できるように，我々が述べた手続きは高い確率で φ のかなり良い近似を与えることを明らかにしよう．これを示すには注意深い取り扱いが必要になる．

b を 0 から 2^t-1 の範囲の整数として，$b/2^t = 0.b_1 \ldots b_t$ が φ より小さく φ の最良の t ビット近似になるように選ぶとする．つまり φ と $b/2^t$ の差 $\delta \equiv \varphi - b/2^t$ は $0 \le \delta \le 2^{-t}$ を満たす．我々の目的は位相推定手続きの最後の観測が b に近い結果を与え，高い確率で φ を正確に推定できることを示すことにある．量子 Fourier 逆変換を状態 (5.20) に適用すると，次の状態が得られる：

$$\frac{1}{2^t}\sum_{k,l=0}^{2^t-1} e^{\frac{-2\pi i k l}{2^t}} e^{2\pi i \varphi k}|l\rangle. \tag{5.23}$$

α_l を $|(b+l)(\mathrm{mod}\ 2^t)\rangle$ の振幅とすると

$$\alpha_l \equiv \frac{1}{2^t}\sum_{k=0}^{2^t-1}\left(e^{2\pi i(\varphi-(b+l)/2^t)}\right)^k. \tag{5.24}$$

これは幾何級数なので

$$\alpha_l = \frac{1}{2^t}\left(\frac{1-e^{2\pi i(2^t\varphi-(b+l))}}{1-e^{2\pi i(\varphi-(b+l)/2^t)}}\right) \tag{5.25}$$

$$= \frac{1}{2^t}\left(\frac{1-e^{2\pi i(2^t\delta-l)}}{1-e^{2\pi i(\delta-l/2^t)}}\right). \tag{5.26}$$

最終測定結果を m とする．我々の目的は $|m-b|>e$ となる値 m を得る確率の限界を与えることである．ここで e は所望の許容誤差を特徴付ける正の整数である．そのような m を観測する確率は次式で与えられる：

$$p(|m-b|>e) = \sum_{-2^{t-1}<l\le-(e+1)}|\alpha_l|^2 + \sum_{e+1\le l\le 2^{t-1}}|\alpha_l|^2. \tag{5.27}$$

しかし任意の実数 θ に対して $|1-\exp(i\theta)| \le 2$ なので

$$|\alpha_l| \le \frac{2}{2^t|1-e^{2\pi i(\delta-l/2^t)}|}. \tag{5.28}$$

74 第 5 章 量子 Fourier 変換とその応用

初歩的な幾何学あるいは解析学より，$-\pi \le \theta \le \pi$ ならば常に $|1-\exp(i\theta)| \ge 2|\theta|/\pi$ である．しかし $-2^{t-1} < l \le 2^{t-1}$ のときに $-\pi \le 2\pi(\delta - l/2^t) \le \pi$ となる．したがって

$$|\alpha_l| \le \frac{1}{2^{t+1}(\delta - l/2^t)}. \tag{5.29}$$

式 (5.27) と式 (5.29) を組み合わせると

$$p(|m-b|>e) \le \frac{1}{4}\left[\sum_{l=-2^{t-1}+1}^{-(e+1)}\frac{1}{(l-2^t\delta)^2} + \sum_{l=e+1}^{2^{t-1}}\frac{1}{(l-2^t\delta)^2}\right]. \tag{5.30}$$

$0 \le 2^t\delta \le 1$ を思い出すと

$$p(|m-b|>e) \le \frac{1}{4}\left[\sum_{l=-2^{t-1}+1}^{-(e+1)}\frac{1}{l^2} + \sum_{l=e+1}^{2^{t-1}}\frac{1}{(l-1)^2}\right] \tag{5.31}$$

$$\le \frac{1}{2}\sum_{l=e}^{2^{t-1}-1}\frac{1}{l^2} \tag{5.32}$$

$$\le \frac{1}{2}\int_{e-1}^{2^{t-1}-1} dl\frac{1}{l^2} \tag{5.33}$$

$$= \frac{1}{2(e-1)}. \tag{5.34}$$

精度 2^{-n} で φ を近似する，つまり $e = 2^{t-n}-1$ と選ぶ．位相推定アルゴリズムで $t = n+p$ 個の q ビットを使用すると，式 (5.34) からこの精度で正しい近似を得る確率は少なくとも $1-1/2(2^p-2)$ であることがわかる．したがって，n ビット精度で少なくとも $1-\epsilon$ の成功確率で正確に φ を得るには

$$t = n + \left\lceil \log\left(2+\frac{1}{2\epsilon}\right)\right\rceil. \tag{5.35}$$

位相推定アルゴリズムを利用するには，U の固有状態 $|u\rangle$ を用意できなければならない．そのような固有状態の準備の仕方がわからなければどうなるか？ $|u\rangle$ の代わりに何か他の状態 $|\psi\rangle$ を準備したとする．この状態を U の固有状態 $|u\rangle$ で展開すると $|\psi\rangle = \sum_u c_u|u\rangle$ を得る．固有状態 $|u\rangle$ は固有値 $e^{2\pi i\varphi_u}$ を持つとする．直観的に言って位相推定アルゴリズムを走らせると，その結果は $\sum_u c_u|\widetilde{\varphi_u}\rangle|u\rangle$ に

5.2 位相推定 75

近い出力が与えられるであろう．ここで $\widetilde{\varphi_u}$ は位相 φ_u のかなり良い近似である．したがって，最初のレジスタを読み出すと φ_u の良い近似が得られると期待される．ここで u は確率 $|c_u|^2$ でランダムに選ぶ．ここでの議論を厳密化することは演習 5.8 に譲る．この手続きではアルゴリズムに付加的なランダムさを導入するというコストを払って，(多分未知の) 固有状態を準備するのが避けられる．

演習 5.8: 位相推定アルゴリズムでは状態 $|0\rangle|u\rangle$ を状態 $|\widetilde{\varphi_u}\rangle|u\rangle$ に変換して，与えられた入力 $|0\rangle(\sum_u c_u|u\rangle)$ に対するアルゴリズムの出力が $\sum_u c_u|\widetilde{\varphi_u}\rangle|u\rangle$ となるとしている．もし式 (5.35) に従って t を選ぶと，位相推定アルゴリズムの結論において，n ビット精度で φ を測定する確率が少なくとも $|c_u|^2(1-\epsilon)$ となることを示せ．

位相推定はなぜ興味深いのか？ 先ず物理的観点から有意で興味ある問題 — ユニタリーオペレータの固有ベクトルが与えられたとき，それに付随する固有値を推定する方法 — が位相推定自身によって解ける．これが現実に用いられる理由は他の興味ある問題が位相推定に縮約できるところからくる．この点については後の節で示す．位相推定アルゴリズムを以下にまとめておく．

アルゴリズム：量子位相推定

入力：(1) 整数 j に対して制御 U^j 演算を行うブラックボックス
(2) 固有値 $e^{2\pi i\varphi_u}$ を持つ U の固有状態 $|u\rangle$
(3) $|0\rangle$ に初期化された $t = n + \lceil \log(2+\frac{1}{2\epsilon})\rceil$ 個の q ビット

出力： φ_u に対する n ビット近似 $\widetilde{\varphi_u}$

演算回数： $O(t^2)$ 回の演算と 1 回の制御 U^j ブラックボックス呼び出し．
成功確率は少なくとも $1-\epsilon$

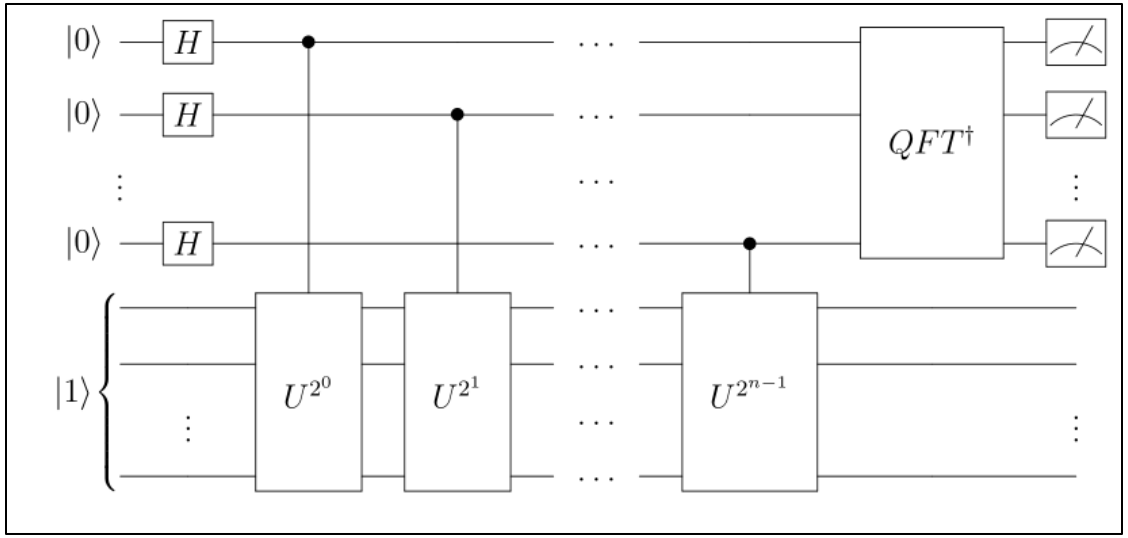
手続き：
1. $|0\rangle|u\rangle$ 初期状態
2. $\rightarrow \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle|u\rangle$ 重ね合せ

参考文献[2] P.73-75

具体的な手順（量子回路）

ステップ2

測定レジスタすべてに，Hゲートを作用.



$$|0\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle \xrightarrow{H} \frac{1}{\sqrt{2^t}}(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes \cdots \otimes (|0\rangle + |1\rangle) \quad (15)$$

$$= \frac{1}{\sqrt{2^t}}(|00\cdots0\rangle + |00\cdots1\rangle + \cdots + |11\cdots1\rangle) \quad (16)$$

$$= \frac{1}{\sqrt{2^t}}(|0\rangle + |1\rangle + \cdots + |2^t - 1\rangle) \quad (17)$$

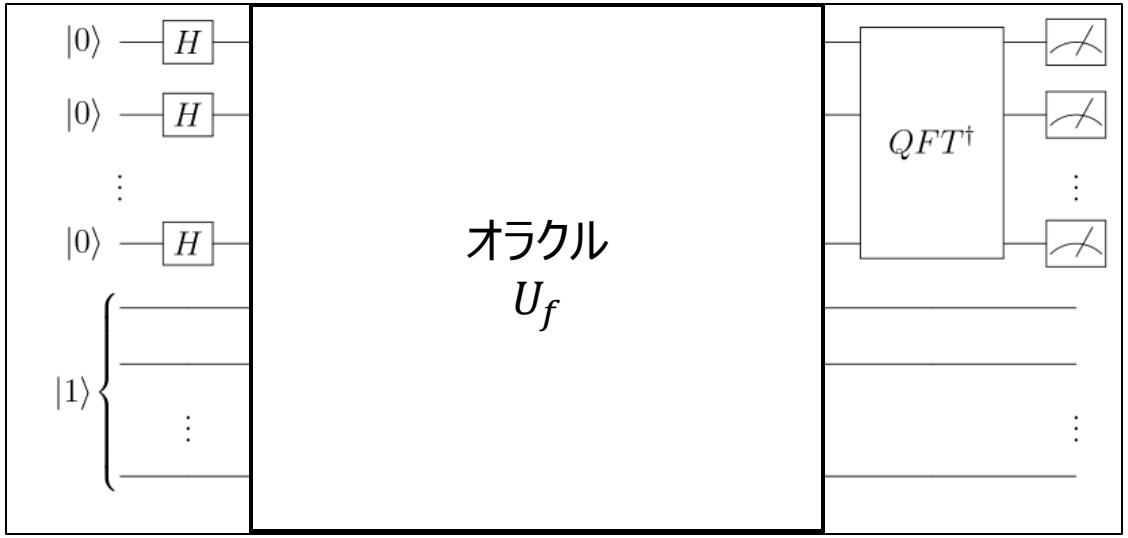
$$= \frac{1}{\sqrt{2^t}} \sum_{a=0}^{2^t-1} |a\rangle \quad (18)$$

↑関数$x^a \bmod N$に渡す引数aを
重ね合わせで作っている.

具体的な手順（量子回路）

ステップ3

制御ユニタリゲート$U_{x,N}$を作用.



$$\frac{1}{\sqrt{2^t}}\sum_{a=0}^{2^t-1}|a\rangle|1\rangle \xrightarrow{U_f} \frac{1}{\sqrt{2^t}}\sum_{a=0}^{2^t-1}|a\rangle|x^a \bmod N\rangle \tag{24}$$

$$= \frac{1}{\sqrt{2^t}}\sum_{a=0}^{2^t-1}\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}e^{\frac{2\pi isa}{r}}|a\rangle|u_s\rangle \tag{25}$$

↑測定レジスタを制御（引数）として，作業レジスタに計算結果を重ね合わせとして保持

具体的な手順（量子回路）

(補足)

$$\frac{1}{\sqrt{2^t}}\sum_{a=0}^{2^t-1}|a\rangle\,|x^a \bmod N\rangle = \frac{1}{\sqrt{2^t}}\sum_{a=0}^{2^t-1}\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}e^{\frac{2\pi isa}{r}}\,|a\rangle\,|u_s\rangle$$

直感的な説明

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

より，第1レジスタを制御部とする$U_{x,N}$を掛けると，

$$U_{x,N}\,|1\rangle = \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}U_{x,N}\,|u_s\rangle \tag{30}$$

$$= \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}e^{\frac{2\pi isa}{r}}\,|u_s\rangle \tag{31}$$

となる．ただし，固有値・固有状態の定義式を利用．

$$U_{x,N}\,|u_s\rangle = e^{\frac{2\pi isa}{r}}\,|u_s\rangle$$

式変形による証明

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}e^{\frac{2\pi isa}{r}}\,|u_s\rangle = \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}e^{\frac{2\pi isa}{r}}\cdot\frac{1}{\sqrt{r}}\sum_{a'=0}^{r-1}e^{\frac{2\pi isa'}{r}}\,|x^{a'} \bmod N\rangle \tag{26}$$

$$= \frac{1}{r}\sum_{s=0}^{r-1}\sum_{a'=0}^{r-1}e^{-\frac{2\pi is(a'-a)}{r}}\,|x^{a'} \bmod N\rangle \tag{27}$$

$$= \sum_{a'=0}^{r-1}\delta_{aa'}\,|x^{a'} \bmod N\rangle \tag{28}$$

$$= |x^a \bmod N\rangle \tag{29}$$

ただし，$|u_s\rangle$が以下で表されること，等比数列の和の公式を利用．

$$|u_s\rangle = \frac{1}{\sqrt{r}}\sum_{a'=0}^{r-1}e^{\frac{2\pi isa'}{r}}\,|x^{a'} \bmod\rangle$$

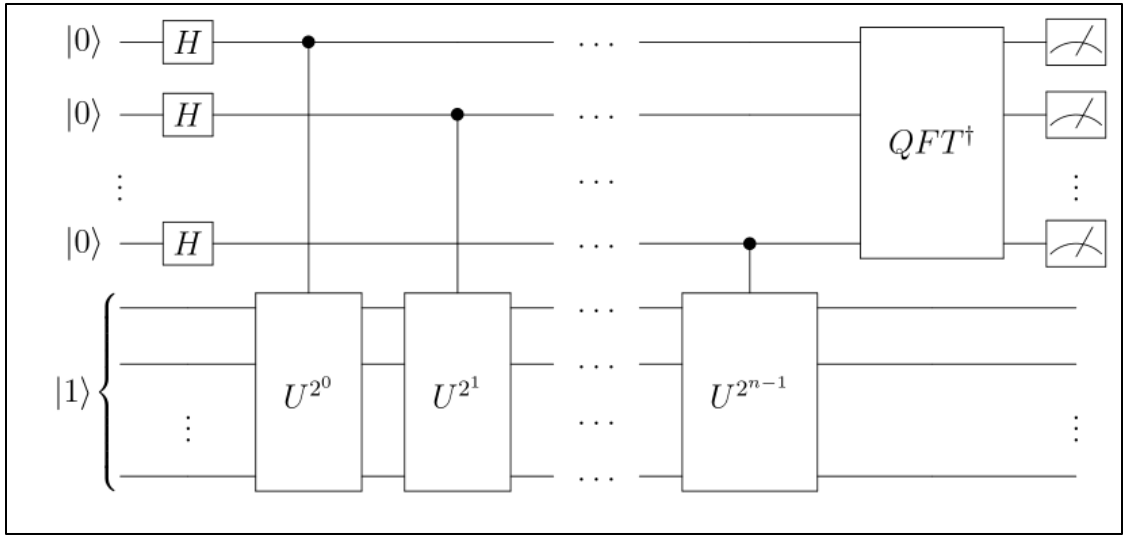
$$\sum_{s=0}^{r-1}e^{-\frac{2\pi is(a'-a)}{r}} = \begin{cases} r & (a=a') \\ 0 & (a\neq a') \end{cases}$$

佐藤研 SATOH Lab.

具体的な手順（量子回路）

ステップ4

測定レジスタに$QFT^{\dagger}$を作用.



QFTの定義式

$$|j\rangle \xrightarrow{QFT} \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} e^{\frac{2\pi ijk}{2^t}} |k\rangle$$

$$\frac{1}{\sqrt{2^t}} \sum_{a=0}^{2^t-1} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{\frac{2\pi isa}{r}} |a\rangle |u_s\rangle \xrightarrow{QFT^{\dagger}} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |\frac{s}{r}\rangle |u_s\rangle$$

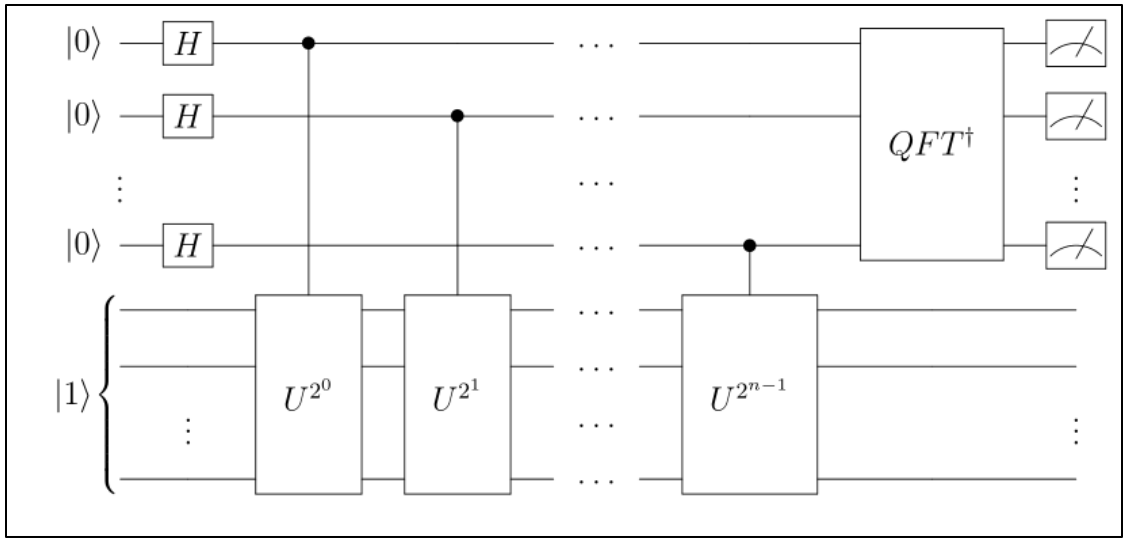
↑測定レジスタの状態に求めたい位数rの情報がある.

(sが邪魔…)

具体的な手順（量子回路）

ステップ5

測定レジスタを測定.



$$|\frac{s}{r}\rangle = |0.j_0 j_1 \cdots j_{t-1}\rangle$$

測定レジスタの各qbitからは
$\frac{s}{r}$を小数表示したときの
各位の値(0,1)が測定される

→つまり，$j_0 j_1 \cdots j_{t-1}$の
ビット列が得られる

佐藤研 SATOH Lab.

具体的な手順（量子回路）

ステップ6

連分数アルゴリズム（実数を分数に書き直す）の適用.

測定した結果，得られたのは

$$j_0 j_1 \cdots j_{t-1(2)} = s'_{(10)}$$

$$\left|\frac{s}{r}\right\rangle = |0.j_0 j_1 \cdots j_{t-1}\rangle$$

$\frac{s'}{2^t}$を連分数アルゴリズムにより，

分数に書き直す.

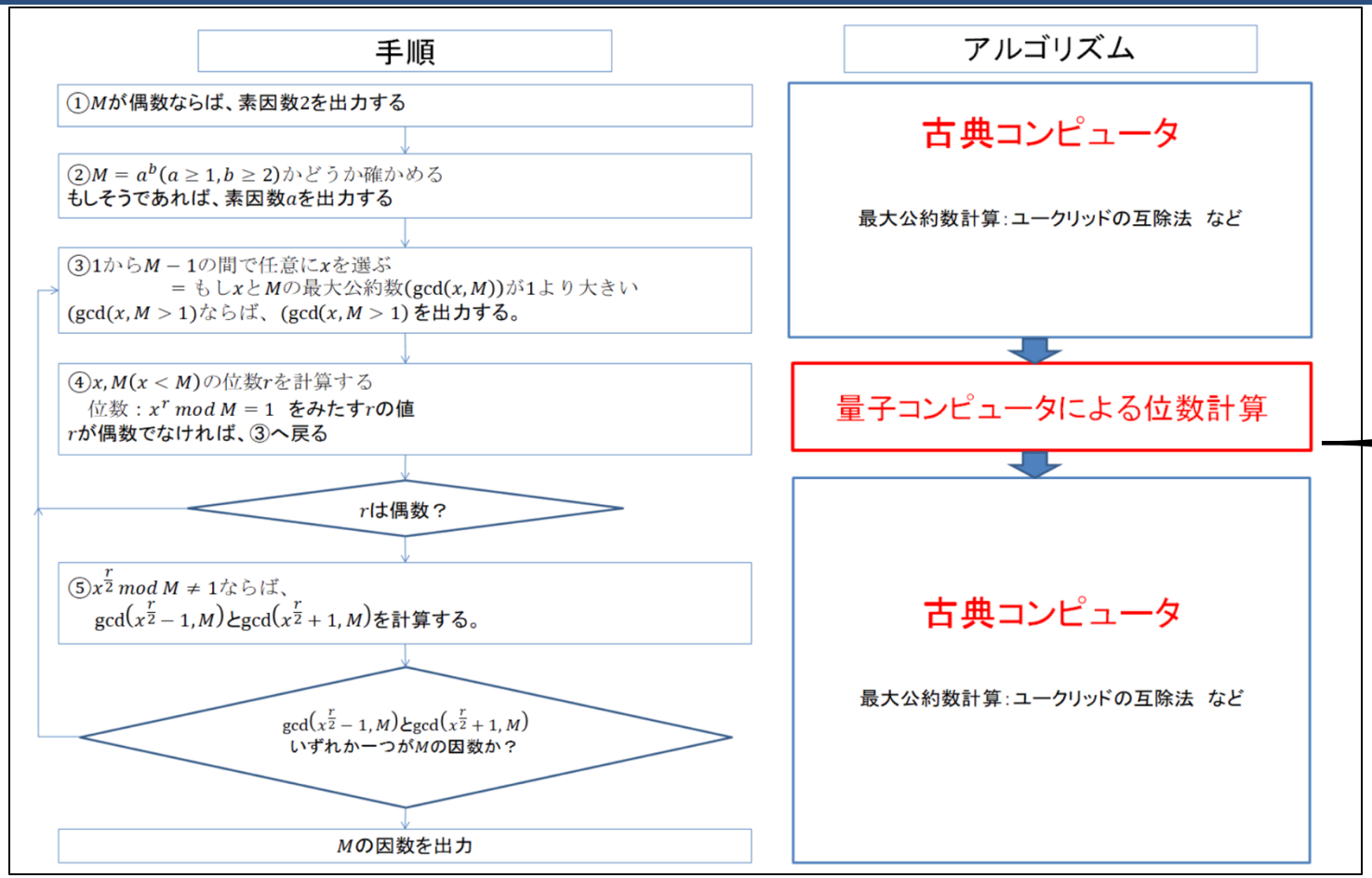
つまり，

$$\frac{s}{r} = \frac{s'}{2^t} \xrightarrow{\text{分数化}} \frac{d'}{r'}$$

で得られるr'が求めたい位数.

佐藤研 SATOH Lab.

ここまでのまとめ



https://plasmaima.com/量子コンピュータ/ショアのアルゴリズム

量子部分の操作

1. レジスタ初期化.
2. 重ね合わせを作成.
3. 制御ユニタリを作用.
4. $QFT^{\dagger}$を作用.
5. 測定.
6. 連分数アルゴリズム適用.

はじめに
結論（ショアのアルゴリズムを一言で）
位数発見問題とは
位数発見の方法
具体的な手順（量子回路）
アルゴリズムの計算量解析
実際に動かしてみる

アルゴリズムの計算量解析

アルゴリズムの計算量解析

古典vs量子

古典

一般数体ふるい法

$$\exp((\frac{64}{9})^{1/3}(\log n)^{1/3}(\log\log n)^{2/3})$$

2020年に
829ビット（10進数で250桁）の
素因数分解に成功

量子

ショアのアルゴリズム

$$O((\log N)^3)$$

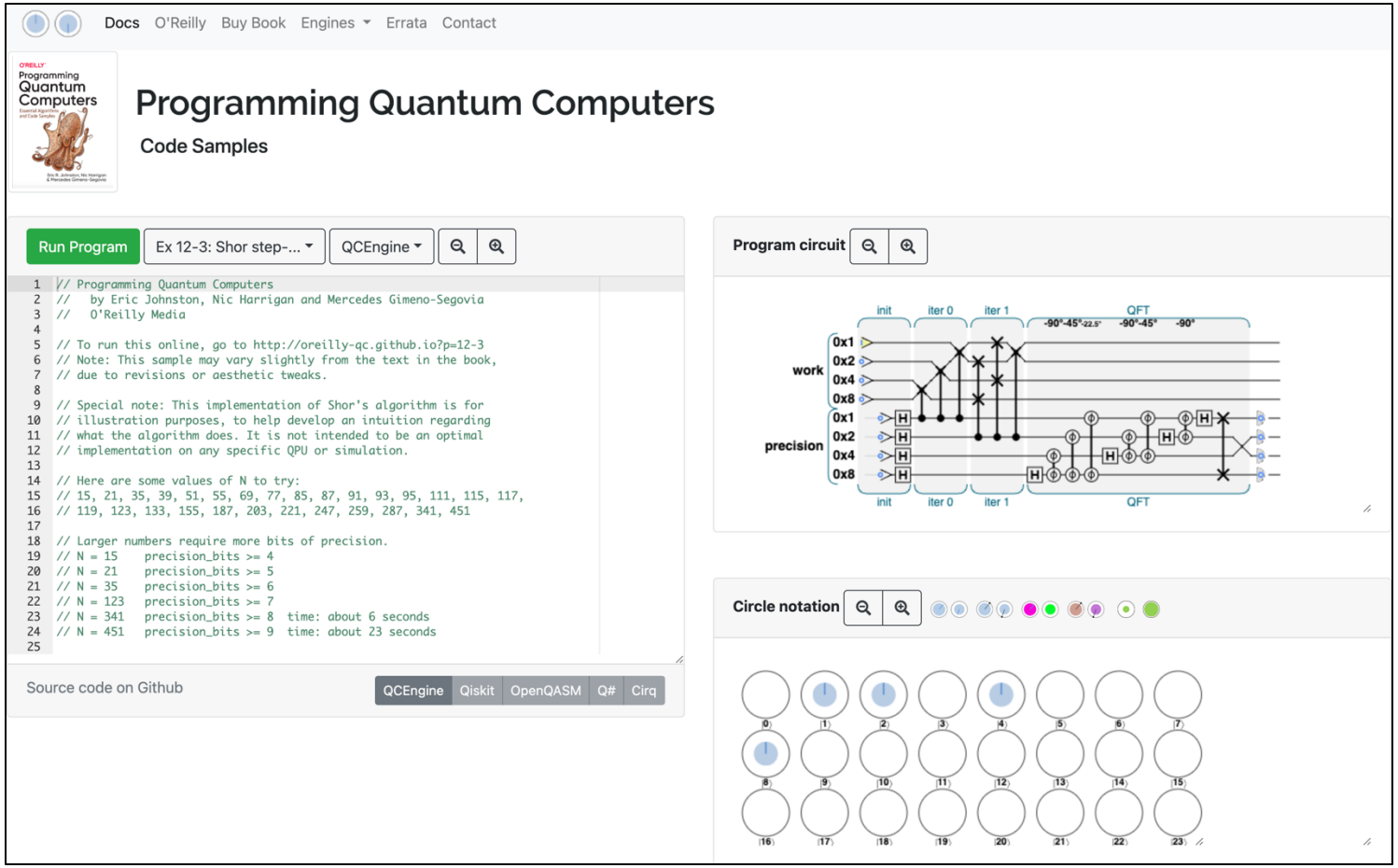
古典的な前処理：$O(\log N)$
量子的な処理：$O((\log N)^3)$
古典的な後処理：$O(\log N)$

佐藤研 SATOH Lab.

はじめに
結論（ショアのアルゴリズムを一言で）
位数発見問題とは
位数発見の方法
具体的な手順（量子回路）
アルゴリズムの計算量解析
実際に動かしてみる

実際に動かしてみる

実際に動かしてみる



https://oreilly-qc.github.io/?p=12-3

OREILLY,
動かして学ぶ
量子コンピュータ
プログラミング

のサンプルコードで
ショアのアルゴリズムの感覚を掴む.

実際に動かしてみる

5. 量子コンピュータの展望

日本総研 The Japan Research Institute, Limited

(参考) Shorのアルゴリズムに関する見解

- セキュリティの分野では、**Shorのアルゴリズム (RSA暗号を解読する量子アルゴリズム) によって、既存の暗号がいつ危殆化するのかが注目**されており、この予測に関連する研究が複数行われている。
- いずれの研究でも、2030年頃に期待される誤り訂正可能な量子コンピュータでは未だ暗号は危殆化せず、さらなるスケールアップが必要と想定されている。

暗号解読に必要なリソースの見積もり (Googleなど)

- 2,048ビットのRSA暗号の解読ができる規模で、Shorのアルゴリズムを動作させるために必要なハードウェアのリソースが見積られている (下図)。

論理量子ビット数	**14,000**
物理量子ビット数	**2,300万**
Toffoli + T ゲート数	**27億**

出所：Craig Gidney and Martin Eker, How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. Quantum 5, 2021, page 433

量子シミュレータによるRSA暗号の安全性評価 (富士通)

- 2023年1月に、富士通は自社開発した39量子ビットの量子コンピュータシミュレータ (以下、量子シミュレータ) を活用し、現在普及している2,048ビットのRSA暗号の安全性を評価。
- Shorのアルゴリズムを量子シミュレータ上に実装し、必要なリソースを定量的に評価した結果、2,048ビットのRSA暗号の解読には、約10,000論理量子ビットと、10^{12}～10^{14}(約2兆2,300億) の量子ゲートもの膨大な規模を有する誤り耐性型の量子コンピュータが必要なことが判明。
- これは、試算すると約104日の間、量子ビットを誤りなく保持する必要があり、現状、このような大規模かつ長時間にわたり安定稼働する量子コンピュータの実現は短期的には困難である。

出所：富士通プレスリリース, 量子シミュレータを活用したRSA暗号の安全性評価に成功, 2023

現状：

2023年12月
IBM「Condor」
1121物理量子ビット

佐藤研 SATOH Lab.

参考文献

[1] 2020. Eric R. Johnston, Nic Harrigan, Mercedes Gimeno-Segovia. 動かして学ぶ量子コンピュータプログラミング ーシミュレータとサンプルコードで理解する基本アルゴリズムー. オーム社.

[2] 2005. Michael A. Nielsen, Isaac L.Chuang. 木村達也訳. 量子コンピュータと量子通信Ⅱ 量子コンピュータとアルゴリズム. オーム社.

[3] 2020. 嶋田義皓. 量子コンピューティング 基本アルゴリズムから量子機械学習まで. オーム社.

佐藤研 SATOH Lab.