Introduction to Quantum Computing
量子計算入門
Lecture 2: Quantum Algorithms

Rod Van Meter
rdv@tera.ics.keio.ac.jp
September 28-30, 2004
@会津大学
with numerous slides from E. Abe

Course Outline

- Lecture 1: Introduction
- Lecture 2: Quantum Algorithms
- Lecture 3: Quantum Computational Complexity Theory
- Lecture 4: Devices and Technologies
- Lecture 5: Quantum Computer Architecture
- Lecture 6: Quantum Networking
- Lecture 7: Wrapup

アウトライン

- Review of basics from yesterday
- Deutsch-Jozsa
- Shor's factoring algorithm
- Grover's search algorithm
- Brief look at other algorithms

量子計算とは?

- ひとつの量子は同時に二つの所にある。
  - 誰も見ていない時だけ!
  - 有名なgedankenexperiment: Schroedinger's cat
  - Superposition (重ね合わせ)
- その重ね合わせを使って、ちょう並列計算できるようになっている。

量子計算は何に使えるか?

- 素因数分解(Shor's algorithm):
  量子計算すると：O(L^3) for L-bit number
  古典的な計算方法だと：O(2^L)
- 検索(Grover's algorithm):
  O(sqrt(N)) to search N items (N=2^L)
- Quantum Key Distribution:
  物理学のせいで、絶対セキュア

量子計算の基本

- Superposition, phase, and the ket notation
- Entanglement
- 1 and 2-qubit gates
- Measurement and decoherence

Superposition（重ね合わせ）and ket Notation

- Qubit state is a vector
- |0> means the vector for 0;
  |1> means the vector for 1;
  |00> means two bits, both 0;
  |010> is three bits, middle one is 1;
  etc.
- A qubit may be partially both!
  (but stay tuned for measurement...)

1-qubitの状態とBloch球（Phase）

1-qubitの状態の標準基底

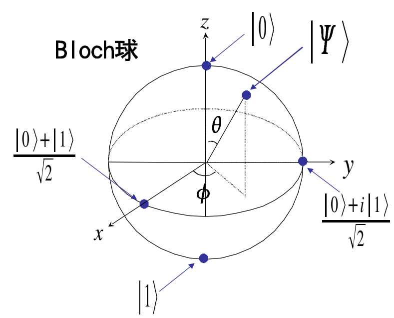
$$|0\rangle=\begin{bmatrix}1\\0\end{bmatrix} \quad |1\rangle=\begin{bmatrix}0\\1\end{bmatrix}$$

任意の重ね合わせ状態

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

$$|\alpha|^2+|\beta|^2=1 \quad (\alpha,\beta\in C)$$

複素2変数－1束縛条件＝実3変数



$$|\Psi\rangle=e^{i\gamma}\left[\cos\frac{\theta}{2}|0\rangle+e^{i\phi}\sin\frac{\theta}{2}|1\rangle\right]$$

測定結果に影響しない

実2変数（物理的要請）

相対位相

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

Entanglementとは?
絡み付き

- 二つのqubitのvalue (0,1)は相手次第である

$$\frac{|00\rangle+|11\rangle}{\sqrt{2}}=\frac{1}{\sqrt{2}}|00\rangle+0|01\rangle+0|10\rangle+\frac{1}{\sqrt{2}}|11\rangle$$

Bit0	Bit1	
0	0	50%
0	1	0%
1	0	0%
1	1	50%

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\\0\\1\end{bmatrix}$$

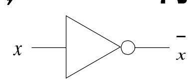
どちかを測定すると、
相手のvalueは決まる。
0でも1でもの確率は50%だが、
(0,1)と(1,0)の確率は0!

Measurement and Decoherence
(測定と位相緩和)

- Qubitを測定すると、重ね合わせがなくなります。必ず1か0かどちかの結果になります。
- その重ね合わせは計算に大事なので、計算がおわってから測定する。
- 偶然に測定されると、decoherence(位相緩和)と呼ぶ。この場合は、計算は失敗である。

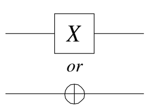
1-qubitの演算の例, Pauli行列

古典回路における1-bit演算 ⇒ NOTのみ



量子演算版NOT = Pauli-X ゲート

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



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

$$\begin{cases}X|0\rangle=|1\rangle\\X|1\rangle=|0\rangle\end{cases}\Leftrightarrow X\begin{bmatrix}1\\0\end{bmatrix}=\begin{bmatrix}0\\1\end{bmatrix},\ X\begin{bmatrix}0\\1\end{bmatrix}=\begin{bmatrix}1\\0\end{bmatrix}$$

$$X=\begin{bmatrix}0&1\\1&0\end{bmatrix}$$

$|1\rangle=\begin{bmatrix}0\\1\end{bmatrix}$ がどの状態になるかを表す

$|0\rangle=\begin{bmatrix}1\\0\end{bmatrix}$ がどの状態になるかを表す

Pauli-Y,Z
ゲート

$$Y=\begin{bmatrix}0&-i\\i&0\end{bmatrix}\qquad \alpha|0\rangle+\beta|1\rangle \;-\boxed{Y}-\; -i\beta|0\rangle+i\alpha|1\rangle$$

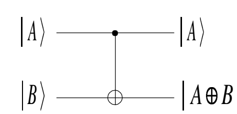
$$Z=\begin{bmatrix}1&0\\0&-1\end{bmatrix}\qquad \alpha|0\rangle+\beta|1\rangle \;-\boxed{Z}-\; \alpha|0\rangle-\beta|1\rangle$$

2-qubitの量子演算の例

例1. 制御NOTゲート

$$C_{AB}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$$

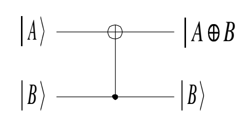
AB
$|00\rangle\rightarrow|00\rangle$
$|01\rangle\rightarrow|01\rangle$
$|10\rangle\rightarrow|11\rangle$
$|11\rangle\rightarrow|10\rangle$



mod 2 の足し算

$$C_{BA}=\begin{bmatrix}1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0\end{bmatrix}$$

AB
$|00\rangle\rightarrow|00\rangle$
$|01\rangle\rightarrow|11\rangle$
$|10\rangle\rightarrow|10\rangle$
$|11\rangle\rightarrow|01\rangle$



How Do Quantum Algos Work?

- Runs are begun by creating a superposition of all possible input values.
- Executing a function gives a superposition of answers of all possible inputs! The hard part is extracting the answer we want.
- Every part of the superposition works *independently* on the algorithm.
- They all work by using *interference*. The *phase* of parts of the superposition are arranged to cancel out and leave only the interesting answer.

量子並列性

n-qubitに対するHadamardゲート

$$|0\rangle^{\otimes n} — \boxed{H^{\oplus n}} —$$

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

$$H^{\oplus n}|0\rangle|0\rangle\cdots|0\rangle = \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\cdots\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)$$

$$= \frac{1}{2^{n/2}}(|0\cdots 00\rangle + |0\cdots 01\rangle + \cdots + |1\cdots 11\rangle)$$

$$= \frac{1}{2^{n/2}}(|0\rangle + |1\rangle + |2\rangle + \cdots + |N-1\rangle) = \underbrace{\frac{1}{2^{n/2}}\sum_{x}^{N-1}|x\rangle}_{2^n = N \text{ 個の状態の等しい重みの重ね合わせ}}$$

例えば，x=0,1,…,N-1 に対して0か1の値をとる2値関数 f(x) が与えられたとする

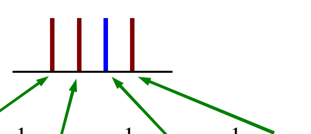
さらに，量子並列性によって f(x) に関する全ての情報の重ね合わせをつくれたとする

$$\frac{1}{2^{n/2}}(|f(0)\rangle + |f(1)\rangle + |f(2)\rangle + \cdots + |f(N-1)\rangle)$$

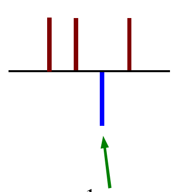
f(x) を決定できるか?

NO! 測定したら f(x) の値のどれか1つを得るだけ

Graphic Representation



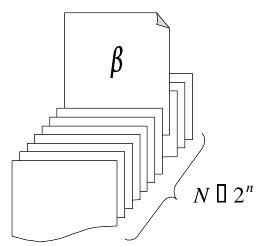
$$\frac{1}{\sqrt{4}}|00\rangle + \frac{1}{\sqrt{4}}|01\rangle + \frac{1}{\sqrt{4}}|10\rangle + \frac{1}{\sqrt{4}}|11\rangle$$



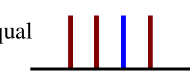
$$\frac{1}{\sqrt{4}}|00\rangle + \frac{1}{\sqrt{4}}|01\rangle - \frac{1}{\sqrt{4}}|10\rangle + \frac{1}{\sqrt{4}}|11\rangle$$

Each bar is the amplitude of the wave function, that is, the square root of the probability, of finding the system in a particular state.

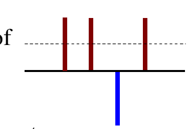
Example: Search



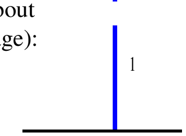
Start with all equal probabilities:



Flip the phase of the answer:



Flip all states about the mean (average):

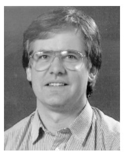


The analog nature of phase figures in strongly!

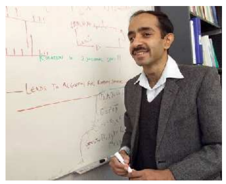
量子アルゴリズム

- Deutsch-Jozsa(D-J)のアルゴリズム
  - Proc. R. Soc. London A, 439, 553 (1992)
- Groverの検索アルゴリズム
  - Phys. Rev. Lett., 79, 325 (1997)
- Shorの素因数分解アルゴリズム
  - SIAM J. Comp., 26, 1484 (1997)

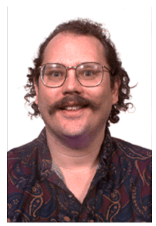
D. Deutsch



R. Jozsa



L. K. Grover



P. W. Shor

Deutschの問題

x=0,1,…,2^n−1 に対して定義された2値関数 f(x)
が "constant" であるか "balanced" であるか判定せよ

constant: 全ての x に対して f(x) の値が同じ (全て0 or 全て1)

n=2の例 $f(x)=(0,0,0,0)$ or $f(x)=(1,1,1,1)$

balanced: f(x) の値の半分は0, 半分は1

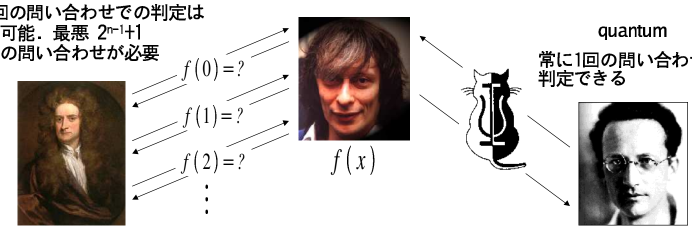
n=2の例 $f(x)=(0,0,1,1)$ とその並べ替え

classical

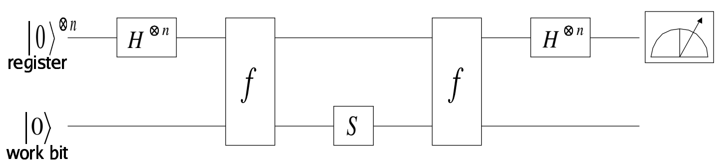
1回の問い合わせでの判定は
不可能. 最悪 $2^{n-1}+1$
回の問い合わせが必要

quantum

常に1回の問い合わせで
判定できる



D-Jを実行する量子回路



Hadamardゲート

$$H|x\rangle=\frac{1}{\sqrt{2}}\sum_z(-1)^{xz}|z\rangle$$

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

$$H^{\otimes n}|x\rangle=\frac{1}{2^{n/2}}\sum_z(-1)^{x\cdot z}|z\rangle$$

where, $x=x_1x_2\cdots x_n \quad z=z_1z_2\cdots z_n$

$x\cdot z=x_1z_1+x_2z_2+\cdots+x_nz_n$

f ゲート

$$f|x\rangle|y\rangle=|x\rangle|y\oplus f(x)\rangle$$

実行例

$$f|x\rangle|0\rangle=|x\rangle|0\oplus f(x)\rangle=|x\rangle|f(x)\rangle$$

$$f|x\rangle|f(x)\rangle=|x\rangle|f(x)\oplus f(x)\rangle=|x\rangle|0\rangle$$

S ゲート

$$S|x\rangle|y\rangle=(-1)^y|x\rangle|y\rangle$$

D-Jの実行過程

$$|0\rangle^{\otimes n}|0\rangle \xrightarrow{H^{\otimes n}} \boxed{\frac{1}{2^{n/2}}\sum_x|x\rangle}|0\rangle \xrightarrow{f} \frac{1}{2^{n/2}}\sum_x|x\rangle\boxed{|f(x)\rangle}$$

重ね合わせ状態をつくる

f(x) の情報をwork bitに乗せる (entanglement)

$$\xrightarrow{S} \frac{1}{2^{n/2}}\sum_x\boxed{(-1)^{f(x)}}|x\rangle|f(x)\rangle \xrightarrow{f} \frac{1}{2^{n/2}}\sum_x(-1)^{f(x)}|x\rangle\boxed{|0\rangle}$$

f(x) の情報を乗せた非局所的な位相シフト

work bitから f(x) の情報を消去 (quantum erasure)

$$\xrightarrow{H^{\otimes n}} \frac{1}{2^n}\sum_{x,z}(-1)^{f(x)+x\cdot z}|z\rangle|0\rangle$$

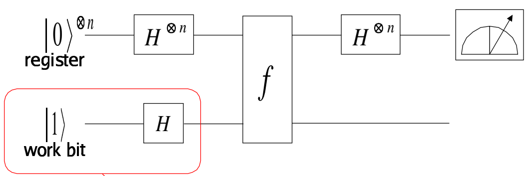
量子並列性と量子干渉を利用したアルゴリズム

registerを観測

$|z\rangle=|00\cdots0\rangle$ の確率振幅 $=\frac{1}{2^n}\sum_x(-1)^{f(x)}=\begin{cases}\pm1 & constant\\ 0 & balanced\end{cases}$

干渉効果

D-J（改良版）



$$|0\oplus f(x)\rangle-|1\oplus f(x)\rangle=(-1)^{f(x)}(|0\rangle-|1\rangle)$$

$$|0\rangle^{\otimes n}|1\rangle \xrightarrow{H^{\otimes n+1}} \frac{1}{2^{n/2}}\sum_x |x\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \xrightarrow{f} \frac{1}{2^{n/2}}\sum_x (-1)^{f(x)}|x\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$$

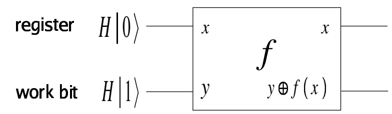
位相シフトとquantum erasureの両方の役割を果たしている

$$\xrightarrow{H^{\otimes n}} \frac{1}{2^n}\sum_{x,z}(-1)^{f(x)+x\cdot z}|z\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$$

S ゲートと2回目の f ゲート不要!!

f ゲートの例，2 bit

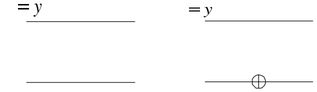
x	constant		balanced	
	f_1	f_2	f_3	f_4
0	0	1	0	1
1	0	1	1	0



constant

$y\oplus f_1(x)=y\oplus 0$
$=y$

$y\oplus f_2(x)=y\oplus 1$
$=\bar{y}$



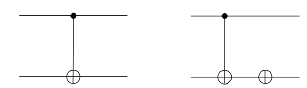
registerの変化 $H|0\rangle \xrightarrow{f_1} H|0\rangle \xrightarrow{H} |0\rangle$

何もしないので元通り．work bitにNOTが入っても全体の位相が変わるだけ

balanced

$y\oplus f_3(x)=y\oplus x$

$y\oplus f_4(x)=y\oplus \bar{x}$



registerの変化

$$\frac{|0\rangle+|1\rangle}{\sqrt{2}} \xrightarrow{f_3} \frac{|0\rangle-|1\rangle}{\sqrt{2}} \xrightarrow{H} |1\rangle$$

$$C_{rw}|1\rangle_r\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]_w=-|1\rangle_r\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]_w$$

制御NOTの掛かったregisterは1になる

D-J and Grover

Deutsch-Jozsa takes advantage of *interference* in the phase to cancel out unwanted terms in the superposition.

But, D-J uses only +1 and -1 in the phase and essentially calculates parity.

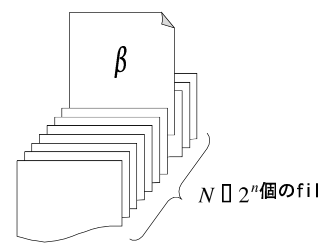
Grover's algorithm takes advantage of the full continuous nature of phase to create interference...

(Note: Remember, phase applies to the whole term in the superposition, not just a single qubit! Shift the phase on any qubit and you shift it on the whole term in the superposition.)

Groverの検索アルゴリズム

$N=2^n$ 個のfileの中から，所望のfile "β" を検索する

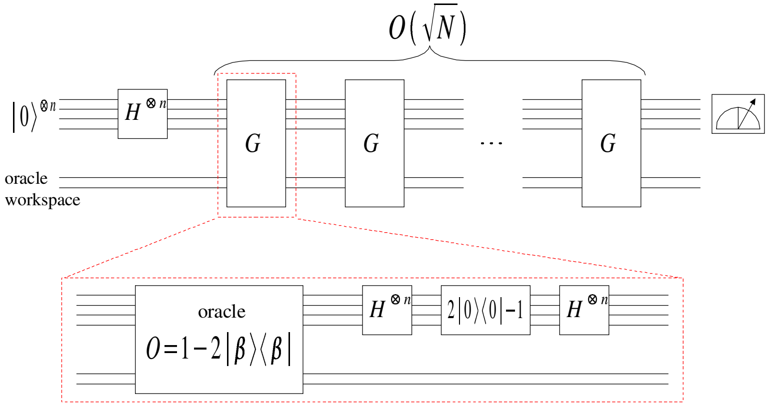
古典的には，順番にfileを調べて，平均N/2回程度の操作が必要



Groverのアルゴリズムでは，N 個のfile(状態)の重ね合わせから，出発して $\sqrt{N}$ 回程度のunitary演算G を実行することで，ほぼ所望のfileに到達

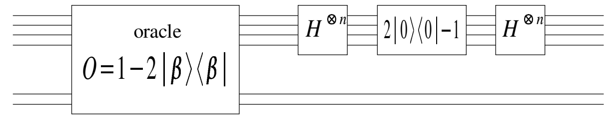
$$|\Psi\rangle=\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle \longrightarrow \approx|\beta\rangle$$

Groverを実行する量子回路



oracle: 神託．論理ゲートを実行するblack box

Gゲートの解析(1)



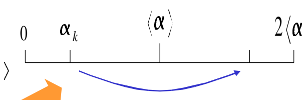
$$\begin{cases} O|\beta\rangle=|\beta\rangle-2|\beta\rangle\langle\beta|\beta\rangle=-|\beta\rangle \\ O|x\rangle=|x\rangle-2|\beta\rangle\langle\beta|x\rangle=|x\rangle\,(x\neq\beta)\end{cases}$$

所望のfileに対してのみ，符号反転

$$H^{\otimes n}(2|0\rangle\langle 0|-1)H^{\otimes n}=2H^{\otimes n}|0\rangle\langle 0|H^{\otimes n}-1$$
$$=2|\psi\rangle\langle\psi|-1$$

$$(2|\psi\rangle\langle\psi|-1)\sum_k \alpha_k|k\rangle=\frac{2}{N}\sum_{k,k',k''}\alpha_k|k''\rangle\langle k'|k\rangle-\sum_k\alpha_k|k\rangle$$
$$=\sum_k[2\langle\alpha\rangle-\alpha_k]|k\rangle$$

“inversion about average”



$$\langle\ \rangle\equiv\sum_k\frac{k}{N}$$

Gゲートの解析(2)

所望のfile(β)以外のN-1個のfileの重ね合わせ

$$|\alpha\rangle=\frac{1}{\sqrt{N-1}}\sum_x|x\rangle=\begin{bmatrix}1\\0\end{bmatrix}$$
$$¿\ |\beta\rangle=\begin{bmatrix}0\\1\end{bmatrix}¿¿$$

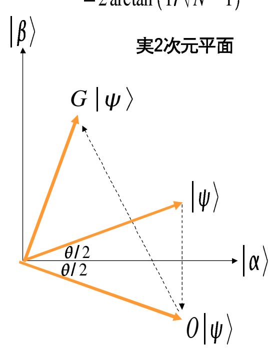
$$|\psi\rangle=\sqrt{\frac{N-1}{N}}|\alpha\rangle+\frac{1}{\sqrt{N}}|\beta\rangle=\begin{bmatrix}\cos(\theta/2)\\ \sin(\theta/2)\end{bmatrix}$$
$$=2\arctan(1/\sqrt{N-1})$$

$$O=1-2|\beta\rangle\langle\beta|$$
$$=\begin{bmatrix}1&0\\0&1\end{bmatrix}-2\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}0&1\end{bmatrix}=\boxed{\begin{bmatrix}1&0\\0&-1\end{bmatrix}|\alpha\rangle\text{に関する折り返し}}$$

$$2|\psi\rangle\langle\psi|-1=2\begin{bmatrix}\cos(\theta/2)\\ \sin(\theta/2)\end{bmatrix}\begin{bmatrix}\cos(\theta/2)&\sin(\theta/2)\end{bmatrix}-\begin{bmatrix}1&0\\0&1\end{bmatrix}$$
$$\boxed{=\begin{bmatrix}\cos\theta&\sin\theta\\ \sin\theta&-\cos\theta\end{bmatrix}|\psi\rangle\text{に関する折り返し}}$$

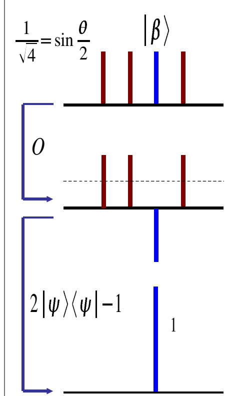
$$G=(2|\psi\rangle\langle\psi|-1)O=\boxed{\begin{bmatrix}\cos\theta&-\sin\theta\\ \sin\theta&\cos\theta\end{bmatrix}\theta\text{ 回転}}$$

実2次元平面



実行例，N=4

各fileの確率振幅の変化



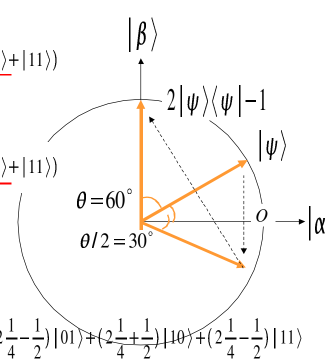
$$|\psi\rangle=\frac{1}{2}(|00\rangle+|01\rangle+|10\rangle+|11\rangle)$$

$$|\psi\rangle=\frac{1}{2}(|00\rangle+|01\rangle-|10\rangle+|11\rangle)$$

$$\boxed{2\langle\alpha\rangle-\alpha_k}$$

$$|\psi\rangle=(2\frac{1}{4}-\frac{1}{2})|00\rangle+(2\frac{1}{4}-\frac{1}{2})|01\rangle+(2\frac{1}{4}+\frac{1}{2})|10\rangle+(2\frac{1}{4}-\frac{1}{2})|11\rangle$$
$$=|10\rangle$$

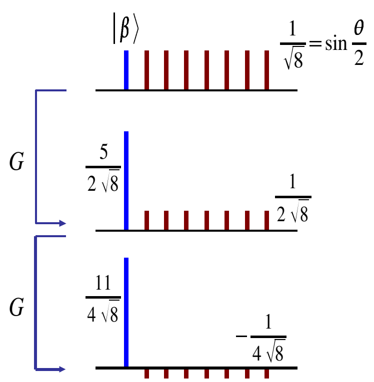
$|\alpha\rangle|\beta\rangle$ 平面での $|\psi\rangle$の変化



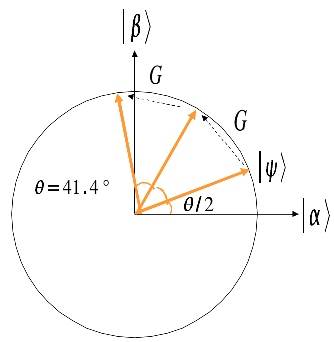
1回のG で100%の確率で所望のfileを得る（rare case）

実行例，N=8

各fileの確率振幅の変化



$|\alpha\rangle|\beta\rangle$ 平面での $|\psi\rangle$ の変化



2回のG でほぼ所望のfileに到達．これ以上やると遠ざかる

Groverのアルゴリズムの効率

所望のfileに到達するまで，何回のG ゲートが必要か?

始状態が $|\psi\rangle = \begin{bmatrix}\cos(\theta/2) \\ \sin(\theta/2)\end{bmatrix}$ で，1回 G を実行するごとに θ 回転するので，

k 回実行した後の状態は

$$G^k|\psi\rangle = \begin{bmatrix}\cos\frac{2k+1}{2}\theta \\ \sin\frac{2k+1}{2}\theta\end{bmatrix}$$

アルゴリズムを終了するのは $\frac{2n+1}{2}\theta \approx \frac{\pi}{2}$ となるとき，

$\sin\frac{\theta}{2} = \frac{1}{\sqrt{N}} \approx \frac{\theta}{2}$ とすると，

$n \approx \frac{\pi}{4}\sqrt{N}$ 回程度繰り返せばよい．

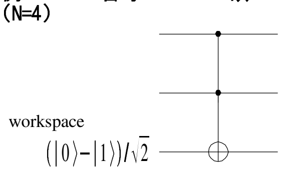
Oracle

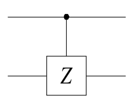
所望のfileの中身を“知らない”のに，oracleを構成できるのか?

例えば，「37で割り切れる番号のfileが欲しい」ときには，「file番号を37で割る回路」をつくって，「割り切れたときのみ符号反転」させればよい．つまり， oracle は「検索条件」だけで構成できる

応用範囲が広い!!（e.g. quantum simulation, quantum counting）

例　「file番号3のfileが欲しい」ときのoracle（N=4）





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

Grover and Shor

Grover uses interference in phase to cancel unwanted terms.

Shor goes a step further, broadening the range of conditions in which useful interference occurs, by doing a Fourier transform...

Shorの素因数分解アルゴリズム

$$66554087 =?\ 6703 \times 9929$$

古典的な方法では，指数オーダーの時間を要する素因
数分解アルゴリズムしか知られていない

古典的には、O(2^L)

量子Fourier変換を使って、O(L^3)

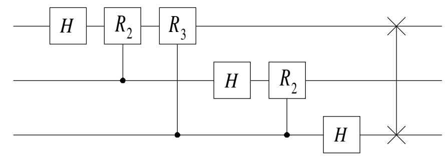
一番有名な量子計算のアルゴリズム

量子Fourier変換

FFTの量子計算版 $|j\rangle \xrightarrow{QFT_N} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp(2\pi ijk/N)|k\rangle$

例　QFT_8を実行する量子回路

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp(2\pi i/2^k) \end{bmatrix}$$



QFT_8の行列表示

$$QFT_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega^1 & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^1 & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\ 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\ 1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega^1 & \omega^6 & \omega^3 \\ 1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega^1 \end{bmatrix}$$

$$\omega = \exp(2\pi i/8) = \sqrt{i}$$
$$\omega^i + \omega^{i+4} = 0$$

QFTの実行例，N=8

$$\sum_{j=0}^{7} \alpha_j |j\rangle \xrightarrow{QFT_8} \sum_{k=0}^{7} \beta_k |k\rangle$$

r	input string {$_j$}		output string {$_k$}	N/r
8	1 0 0 0 0 0 0 0		1 1 1 1 1 1 1 1	1
4	1 0 0 0 1 0 0 0		1 0 1 0 1 0 1 0	2
2	1 0 1 0 1 0 1 0		1 0 0 0 1 0 0 0	4
1	1 1 1 1 1 1 1 1		1 0 0 0 0 0 0 0	8

$$\frac{1}{\sqrt{2}}(|0\rangle + |4\rangle) \xrightarrow{QFT_8} \frac{1}{2}(|0\rangle + |2\rangle + |4\rangle + |6\rangle)$$

input string {$_j$}		output string {$_k$}
1 0 0 0 1 0 0 0		1 0 1 0 1 0 1 0
0 1 0 0 0 1 0 0		1 0 i 0 −1 0 $-i$ 0
0 0 1 0 0 0 1 0		1 0 −1 0 1 0 −1 0
0 0 0 1 0 0 0 1		1 0 $-i$ 0 −1 0 i 0

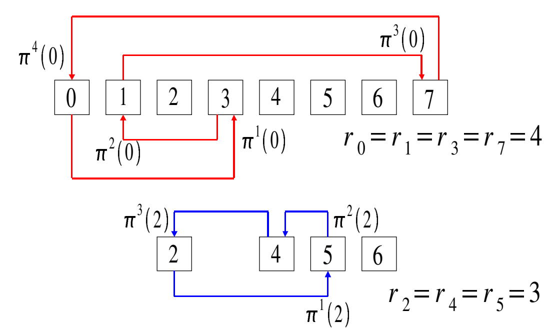
$$\frac{1}{\sqrt{2}}(|3\rangle + |7\rangle) \xrightarrow{QFT_8} \frac{1}{2}(|0\rangle - i|2\rangle - |4\rangle + i|6\rangle)$$

置換の位数(order)

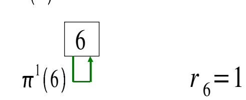
y から置換π を繰り返して，元のy に戻る最小の回数を置換π (y)の位数 r_y と呼ぶ

置換π (y)の例

y	$\pi(y)$
0	3
1	7
2	5
3	1
4	2
5	4
6	6
7	0

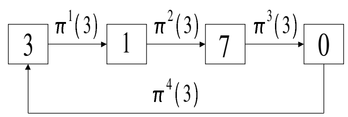


一般に，置換の位数の決定(order-finding)には，指数オーダーの時間を要する

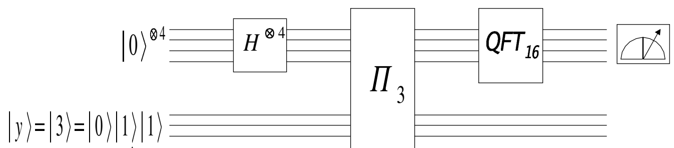


Order-finding

例として，y = 3の場合を考える



$$\pi^0(3)=\pi^4(3)=\pi^8(3)=\pi^{12}(3)=\cdots=3$$
$$\pi^1(3)=\pi^5(3)=\pi^9(3)=\pi^{13}(3)=\cdots=1$$
$$\pi^2(3)=\pi^6(3)=\pi^{10}(3)=\pi^{14}(3)=\cdots=7$$
$$\pi^3(3)=\pi^7(3)=\pi^{11}(3)=\pi^{15}(3)=\cdots=0$$



y = 0,1,…,7 なので3-qubitをwork bitに用意

$$\Pi_y|x\rangle|y\rangle=|x\rangle|\pi^x(y)\rangle$$

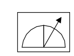
Order-findingの実行過程

$$|0\rangle^{\otimes 4}|3\rangle \xrightarrow{H^{\otimes 4}} \frac{1}{4}\sum_{x=0}^{15}|x\rangle|3\rangle$$

$$QFT_{16}=\frac{1}{\sqrt{16}}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^1 & \omega^2 & \cdots & \omega^{15} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{14} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{15} & \omega^{14} & \cdots & \omega^1 \end{bmatrix} \quad \omega=\exp(2\pi i/16) \quad \omega^i+\omega^{i+8}=0$$

$$\xrightarrow{\Pi_3} \frac{1}{4}\sum_{x=0}^{15}|x\rangle|\pi^x(3)\rangle$$
$$=\frac{1}{4}(|0\rangle+|4\rangle+|8\rangle+|12\rangle)|3\rangle$$
$$+\frac{1}{4}(|1\rangle+|5\rangle+|9\rangle+|13\rangle)|1\rangle$$
$$+\frac{1}{4}(|2\rangle+|6\rangle+|10\rangle+|14\rangle)|7\rangle$$
$$+\frac{1}{4}(|3\rangle+|7\rangle+|11\rangle+|15\rangle)|0\rangle$$

$$\xrightarrow{QFT_{16}} \frac{1}{4}(|0\rangle+|4\rangle+|8\rangle+|12\rangle)|3\rangle+$$
$$\frac{1}{4}(|0\rangle+i|4\rangle-|8\rangle-i|12\rangle)|1\rangle+$$
$$\frac{1}{4}(|0\rangle-|4\rangle+|8\rangle-|12\rangle)|7\rangle+$$
$$\frac{1}{4}(|0\rangle-i|4\rangle-|8\rangle+i|12\rangle)|0\rangle$$



0,4,8,12
(N /r =16/r の整数倍)

古典コンピュータを用いた連分数展開により r を決定

Shorのアルゴリズムの流れ

素因数分解したい数 L と互いに素な数 $a\,(1<x<L)$ をランダムに抽出

$a^r=1\,(mod\ L)$ を満たす最小のr を見つける

量子計算機の担当
order-finding

$r=even$ — NO / YES

$p=gcd(a^{r/2}-1,L)$
$q=gcd(a^{r/2}+1,L)$ を計算

$p,q\neq L$ — NO / YES

$L=pq$

アルゴリズムがうまく働かないケース

1. 偶数
2. 素数
3. 素数のべき乗

15

21

乗法群の位数

$a^r \equiv 1 \pmod L$ を満たす最小のr を「乗法群の位数」と呼ぶ

「置換の位数」との関係は?

$(y)\equiv ay\,(mod\ L)$ とすると，π (y) は「置換」になっている

L =15以下のL と互いに素な数 $a=\{2,4,7,8,11,13,14\}$

$a=7$ のとき

y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
(y)	0	7	14	6	13	5	12	4	11	3	10	2	9	1	8

$a=11$ のとき

y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
(y)	0	11	7	3	14	10	6	2	13	9	5	1	12	8	4

$a^x\,(mod\ L)$ ⇔ $\pi^x(1)$ だから，「乗法群の位数」は「置換π (y) の位数」と同じ

素因数分解，L=15の例

L =15以下のL と互いに素な数　$a=\{2,4,7,8,11,13,14\}$

$a=7$

$$\begin{tikzcd} 1 \arrow[r, "\pi^1(1)"] & 7 \arrow[r, "\pi^2(1)"] & 4 \arrow[r, "\pi^3(1)"] & 13 \arrow[lll, bend left, "\pi^4(1)"] \end{tikzcd}$$

$r_2=r_7=r_8=r_{13}=4$

$a^{r/2}-1=48 \rightarrow gcd(48,15)=3$

$a^{r/2}+1=50 \rightarrow gcd(50,15)=5$

$a=11$

$$\begin{tikzcd} 1 \arrow[r, "\pi^1(1)"] & 11 \arrow[l, bend left, "\pi^2(1)"] \end{tikzcd}$$

$r_4=r_{11}=r_{14}=2$

$a^{r/2}-1=10 \rightarrow gcd(10,15)=5$

$a^{r/2}+1=12 \rightarrow gcd(12,15)=3$

$a=14$

$$\begin{tikzcd} 1 \arrow[r, "\pi^1(1)"] & 14 \arrow[l, bend left, "\pi^2(1)"] \end{tikzcd}$$

失敗!!

$a^{r/2}-1=13 \rightarrow \cancel{gcd(13,15)=1}$

$a^{r/2}+1=15 \rightarrow \cancel{gcd(15,15)=15}$

Euclidの互除法

最大公約数を求めるアルゴリズム

例　$gcd(494,133)$

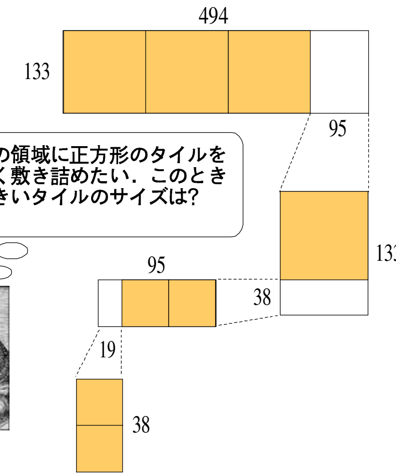
$494=133\times3+95$

$133=95\times1+38$

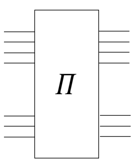
$95=38\times2+19$

$38=19\times2$

長方形の領域に正方形のタイルを隙間無く敷き詰めたい．このとき最も大きいタイルのサイズは?



Πゲート

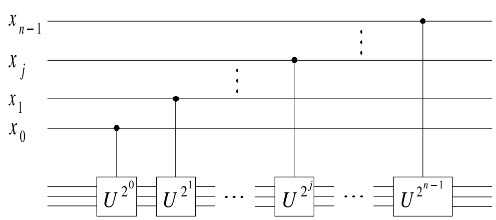


$x=2^{n-1}x_{n-1}+2^{n-2}x_{n-2}+\cdots+2x_1+x_0$ だから，

$a^x(\mathrm{mod}\ L)=a^{2^{n-1}x_{n-1}}a^{2^{n-2}x_{n-2}}\cdots a^{2x_1}a^{x_0}(\mathrm{mod}\ L)$

¿$(a^{2^{n-1}x_{n-1}}(\mathrm{mod}\ L))(a^{2^{n-2}x_{n-2}}(\mathrm{mod}\ L))\cdots(a^{x_0}(\mathrm{mod}\ L))$

$\Pi|x\rangle|y\rangle=|x\rangle|a^x y(mod\ L)\rangle$



$U^{2^j}|y\rangle=|a^{2^j}y(mod\ L)\rangle$

$a^{2^j}(mod\ L)$ の値は，古典計算機ですぐに計算できるので，ゲートを構成できる

「x_j が1のときにゲートを実行」を繰り返す

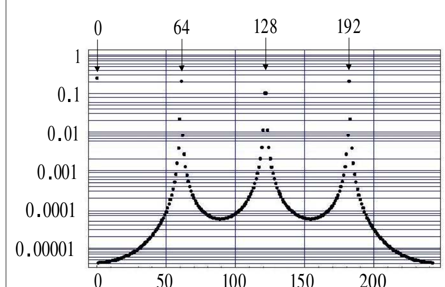
15の素因数分解

Step.1 ランダムに a を選ぶ

$a=7$

Step.2 Order-finding

QFTの結果の例（N =2^8=256）



© Colin P. Williams (1997)

Step.3 観測し，連分数展開でr を決定

$r=4$

Step.4　$p=gcd(a^{r/2}-1,L)$, $q=gcd(a^{r/2}+1,L)$ を計算

$a^{r/2}-1=48 \rightarrow gcd(48,15)=3$

$a^{r/2}+1=50 \rightarrow gcd(50,15)=5$

$15=3\times5$

アルゴリズム終了

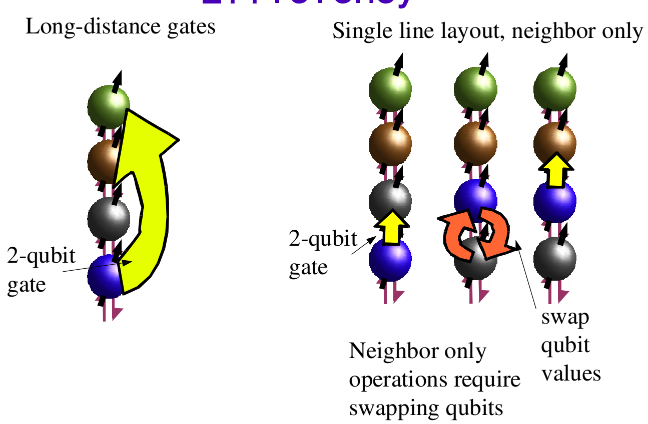
Other Order-Finding Algos.

- Abelian subgroup, discrete logarithm
- QFT based, but very different in classical portion of algorithm
- Hidden subgroup problems in general

Main Classes of Algorithms

- 1: Use the QFT to find periodicity
- 2: Grover's algorithm and friends
- 3: Simulating quantum physics
- (D-J seems to fall outside these)

Architecture Affects Algorithm Efficiency



Wrap-Up on Algorithms

- "Quantum" algorithms actually have both quantum and classical parts
- Use of quantum interference based on complex, analog phase is critical
- Period-finding algorithms work well (exponential speedup over best known algos, but not yet proven better)