2021/5/11

5章　ベルの不等式

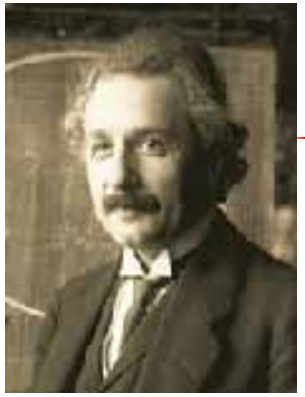
ベルの不等式(p.71)

ニールズ・ボーア

粒子のスピン、光子の偏光に関係する
量子力学のごく一部を数学的にモデル化

コペンハーゲン解釈

・確率の使用
・遠隔作用の概念



アルベルト・アインシュタイン

神はサイコロを振らない

・隠れた変数
・局所実在論

ランダム性を排除して謎
を説明する論理を考える

ベルの不等式(p.72)

コペンハーゲンの
モデル

アインシュタインの
モデル

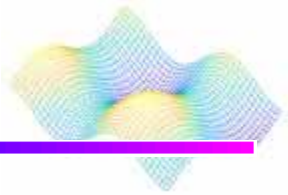
ジョン・スチュアート・ベル

2つのモデルを区別することが
できる独創的なテストを考案

結果は常にコペンハーゲンの解釈
に従った

ベルの不等式
暗号化されたメッセージの送信に使える

異なる基底のもつれ合った量子ビット(p.73)



もつれ合った2つの時計の状態

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

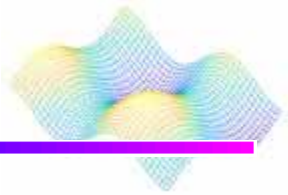
正規直交基底($|b_0\rangle, |b_1\rangle$)に書き換える

$$\frac{1}{\sqrt{2}}|b_0\rangle\otimes|b_0\rangle+\frac{1}{\sqrt{2}}|b_1\rangle\otimes|b_1\rangle$$

次式を証明する

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix}+\frac{1}{\sqrt{2}}\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}=\frac{1}{\sqrt{2}}|b_0\rangle\otimes|b_0\rangle+\frac{1}{\sqrt{2}}|b_1\rangle\otimes|b_1\rangle$$

異なる基底のもつれ合った量子ビット(p.74)



方程式 $\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix}=\begin{bmatrix} a \\ c \end{bmatrix}$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}=a\begin{bmatrix} a \\ b \end{bmatrix}+c\begin{bmatrix} c \\ d \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}\otimes\begin{bmatrix} 1 \\ 0 \end{bmatrix}=\left(a\begin{bmatrix} a \\ b \end{bmatrix}+c\begin{bmatrix} c \\ d \end{bmatrix}\right)\otimes\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$=a\begin{bmatrix} a \\ b \end{bmatrix}\otimes\begin{bmatrix} 1 \\ 0 \end{bmatrix}+c\begin{bmatrix} c \\ d \end{bmatrix}\otimes\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$=\begin{bmatrix} a \\ b \end{bmatrix}\otimes\begin{bmatrix} a \\ 0 \end{bmatrix}+\begin{bmatrix} c \\ d \end{bmatrix}\otimes\begin{bmatrix} c \\ 0 \end{bmatrix} \quad \cdots ①$$

$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix}=\begin{bmatrix} a \\ c \end{bmatrix}$から、同様にして

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}\otimes\begin{bmatrix} 0 \\ 1 \end{bmatrix}=\begin{bmatrix} a \\ b \end{bmatrix}\otimes\begin{bmatrix} 0 \\ b \end{bmatrix}+\begin{bmatrix} c \\ d \end{bmatrix}\otimes\begin{bmatrix} 0 \\ d \end{bmatrix} \quad \cdots ②$$

①+②より $\begin{bmatrix} 1 \\ 0 \end{bmatrix}\otimes\begin{bmatrix} 1 \\ 0 \end{bmatrix}+\begin{bmatrix} 0 \\ 1 \end{bmatrix}\otimes\begin{bmatrix} 0 \\ 1 \end{bmatrix}=\begin{bmatrix} a \\ b \end{bmatrix}\otimes\left(\begin{bmatrix} a \\ 0 \end{bmatrix}+\begin{bmatrix} 0 \\ b \end{bmatrix}\right)+\begin{bmatrix} c \\ d \end{bmatrix}\otimes\left(\begin{bmatrix} c \\ 0 \end{bmatrix}+\begin{bmatrix} 0 \\ d \end{bmatrix}\right)$

$$=\begin{bmatrix} a \\ b \end{bmatrix}\otimes\begin{bmatrix} a \\ b \end{bmatrix}+\begin{bmatrix} c \\ d \end{bmatrix}\otimes\begin{bmatrix} c \\ d \end{bmatrix}$$

$|b_0\rangle\otimes|b_0\rangle+|b_1\rangle\otimes|b_1\rangle$

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 0 \end{bmatrix}\otimes\begin{bmatrix} 1 \\ 0 \end{bmatrix}+\frac{1}{\sqrt{2}}\begin{bmatrix} 0 \\ 1 \end{bmatrix}\otimes\begin{bmatrix} 0 \\ 1 \end{bmatrix}=\frac{1}{\sqrt{2}}|b_0\rangle\otimes|b_0\rangle+\frac{1}{\sqrt{2}}|b_1\rangle\otimes|b_1\rangle$$

異なる基底のもつれ量子ビット(p.75)

この結果より、アリスとボブは次のもつれた量子ビットを持つ

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

アリスとボブが正規直交基底($|b_0\rangle, |b_1\rangle$)を基準に量子ビットを測定した場合、次のように書き換える

$$\frac{1}{\sqrt{2}}|b_0\rangle|b_0\rangle+\frac{1}{\sqrt{2}}|b_1\rangle|b_1\rangle$$

アリスとボブが量子ビットを測定した場合、両者とも0か1になるかで、どちらの結果も同じように起こりえる

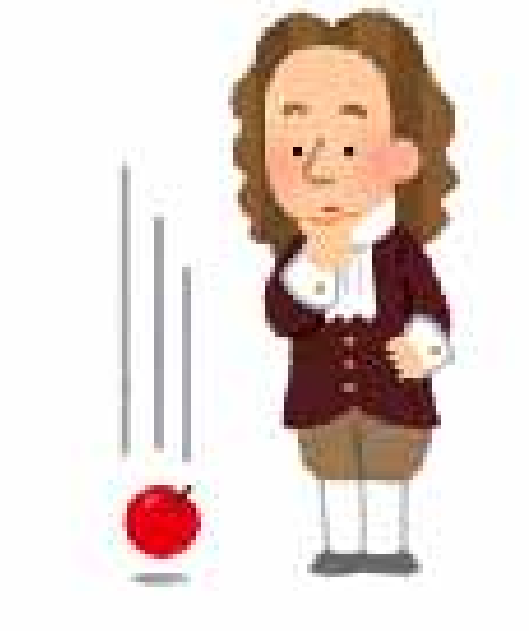
ベルの結果を得るためには、もつれ合った量子ビットを3つの異なる基底を用いて測定する　　0°、120°、240°

もつれ合った時計の場合では、針が12時、4時、8時のどれを指しているか3つの質問をする

$$\frac{1}{\sqrt{2}}|\uparrow\rangle|\uparrow\rangle+\frac{1}{\sqrt{2}}|\downarrow\rangle|\downarrow\rangle$$

アインシュタインと局所実在論(p.76)

重力は局所実在論を説明するよい例



ニュートンの法則

・質量の大きさ
・離れた距離
・重力定数

→ 引力の大きさ

重力の働きを説明するものではない

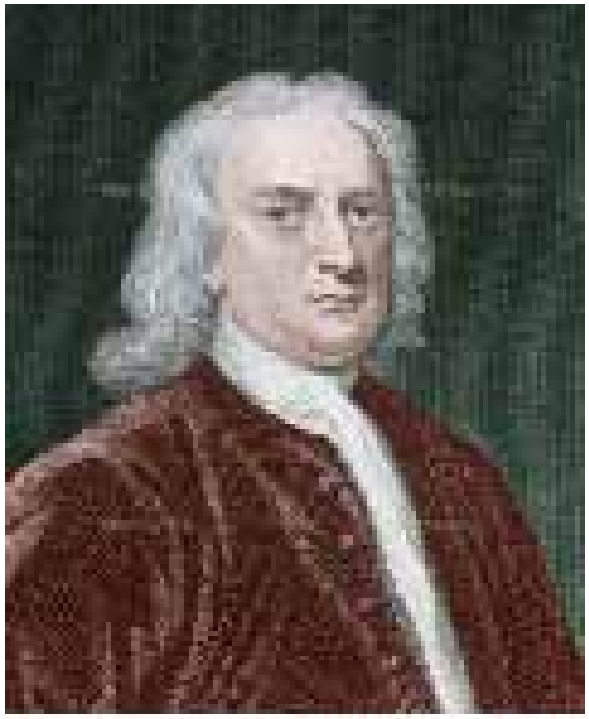
重力のメカニズムについては意見が一致しなかったが、何らかの説明がなされるだろうという意見は一致していた

局所実在論

2021/5/11

P76 ～ P80

ニュートン



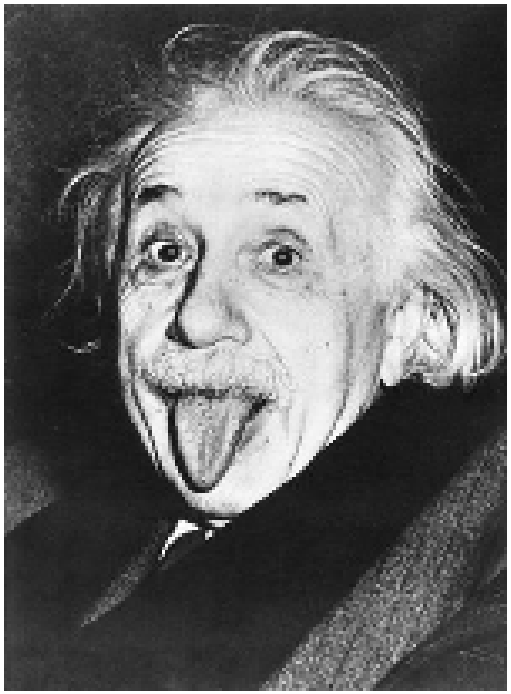
ニュートン

1687年　万有引力の法則

➡ 惑星の軌道を正確に計算

重力がどのように働くかがわからない

重力の働き方



アインシュタイン

重力

↓

時空がゆがむ

↓

惑星は時空に従って動く

アインシュタインの考え

量子力学(全ての物理学)は決定論(古典論)

無限の精度で初期条件を把握 → 正確に未来を予測！

例）ビリヤード

・位置　　・打つ強さ
・角度　　・摩擦係数

打った瞬間に落ちる球は決まっている

アインシュタインの考え

実際には…

完璧な初期状態の把握は不可能

↓

小さな誤差発生

↓

時間と共に増加し、正確に予想できない

例）一週間以上後の天気予報はあてにならない

本質は決定論！

確率論に見える ➡

- 理論が未完成
- 未知の変数

量子力学

量子論（確率論）

・測定するまで量子の状態は決まっておらず、
確率でしか量子の状態はわからない。
測定して初めて量子の状態がわかる。

・量子のペアが遠く離れていても、
片方を測定した瞬間にもう片方の状態もすぐに変化する

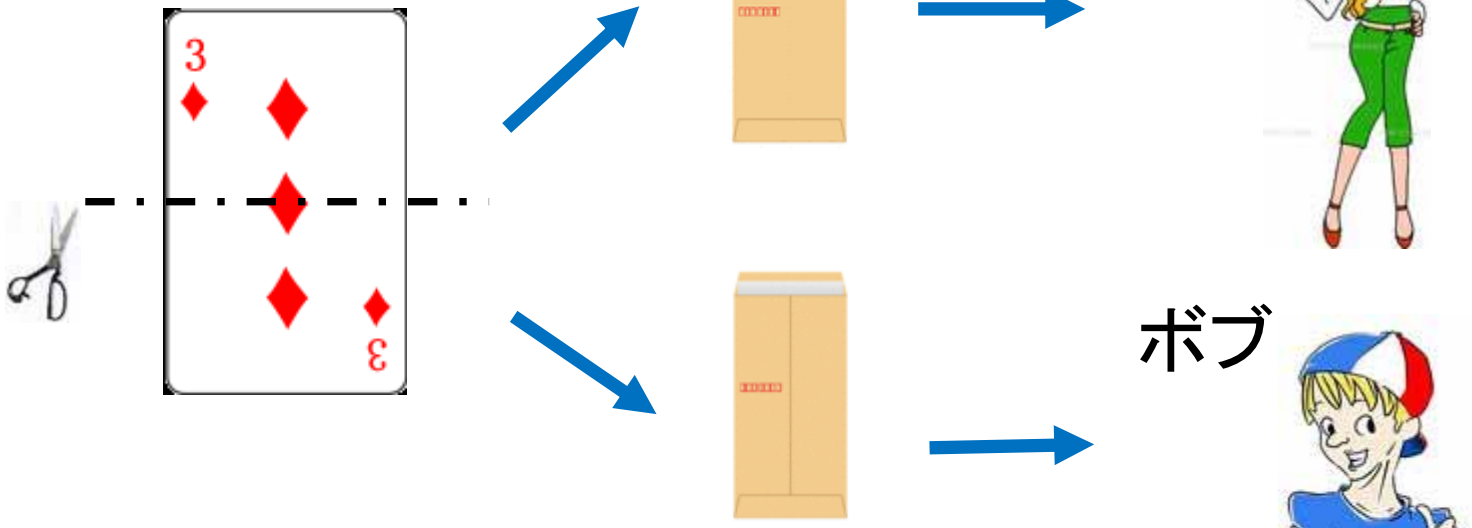
EPRパラドックス[Einstein Podolsky Rosen]

情報は光より速く移動できないが、
瞬間的な変化は情報の光速を越えた瞬間的移動を示す

➡ EPRパラドックスは矛盾ではないと証明

EPRパラドックスは矛盾ではない

イメージ



あたりまえ

アリスは封筒をあけダイヤの3と知る

アリスが封筒を開けた瞬間ボブのカードも決まる

ベルの不等式(量子論)

アリスとボブに量子対 $\frac{1}{\sqrt{2}}|\uparrow\rangle|\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\rangle|\downarrow\rangle$が連続で送られる

アリスが↑, ↘, ↙ をランダムに選択

直後にボムもランダムに選択

[1]ボブが同じ方向を選択する場合($^1/_3$)

例)↑↑, ↘↘, ↙↙

[2]ボブが違う方向を選択する場合($^2/_3$)

例)↑↘, ↘↙, ↙↑

ベルの不等式(量子論)[1]

[1]ボブが同じ方向を選択する場合($^1/_3$)

例)アリスとボブが↘を選択した場合

アリスの量子状態 $\frac{1}{\sqrt{2}}|\searrow\rangle|\searrow\rangle + \frac{1}{\sqrt{2}}|\nwarrow\rangle|\nwarrow\rangle$

$^1/_2$の確率で↘(0),$^1/_2$の確率で↖(1)

$$|\searrow\rangle = \begin{bmatrix} 1/2 \\ -\sqrt{3}/2 \end{bmatrix}, |\nwarrow\rangle = \begin{bmatrix} \sqrt{3}/2 \\ 1/2 \end{bmatrix}, \begin{bmatrix} 1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & 1/2 \end{bmatrix} \begin{bmatrix} 1/2 \\ -\sqrt{3}/2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

ボブの量子状態 $1|\searrow\rangle|\searrow\rangle + 0|\nwarrow\rangle|\nwarrow\rangle = |\searrow\rangle|\searrow\rangle$となり、必ず0

アリスとボブの数字が同じ・・・A(違ったらD)

ボブが同じ方向を選択→必ずA！

ベルの不等式（量子論）[2]

[2]ボブが違う方向を選択する場合($^2/_3$)

例）アリスが↘, ボブが ↙ を選択した場合

アリスの量子状態 $\frac{1}{\sqrt{2}}|\searrow\rangle|\searrow\rangle + \frac{1}{\sqrt{2}}|\nwarrow\rangle|\nwarrow\rangle$

$^1/_2$ の確率で ↘ (0), $^1/_2$ の確率で ↖ (1)

$$|\swarrow\rangle = \begin{bmatrix} -1/2 \\ -\sqrt{3}/2 \end{bmatrix}, \quad |\nearrow\rangle = \begin{bmatrix} \sqrt{3}/2 \\ -1/2 \end{bmatrix}, \quad |\searrow\rangle = \begin{bmatrix} 1/2 \\ -\sqrt{3}/2 \end{bmatrix},$$

$$\begin{bmatrix} -1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix} \begin{bmatrix} 1/2 \\ -\sqrt{3}/2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \sqrt{3}/2 \end{bmatrix}$$

ボブの量子状態 $\frac{1}{2}|\swarrow\rangle|\swarrow\rangle + \frac{\sqrt{3}}{2}|\nearrow\rangle|\nearrow\rangle$

$^1/_4$の確率で0を記録し、Aとなる

ベルの不等式（量子論）

[1]ボブが同じ方向を選択する場合（$^1/_3$）・・・必ずA

[2]ボブが違う方向を選択する場合（$^2/_3$）・・・$^1/_4$の確率でA

量子論でAとなる確率 $\frac{1}{3} \times 1 + \frac{2}{3} \times \frac{1}{4} = \frac{1}{2}$

2021.4

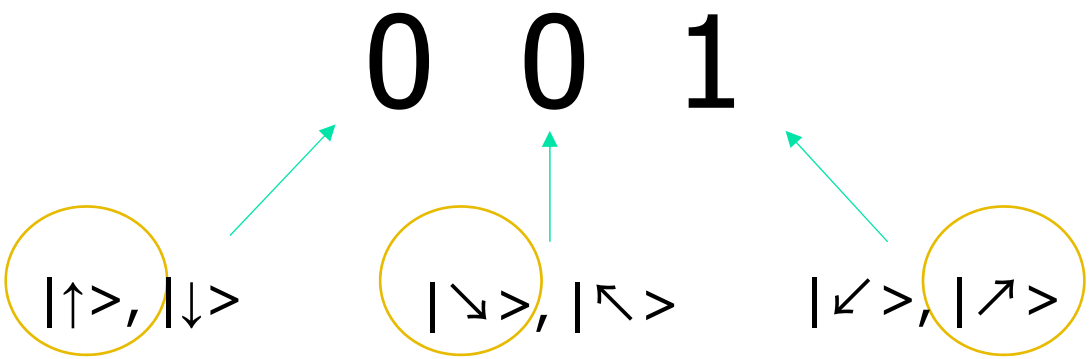
p.81–85

UNIVERSITY
OF TOYAMA

古典モデル　1

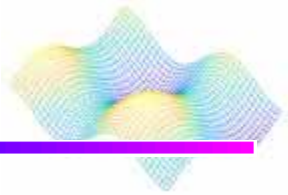
古典的な考え方は、すべての測定値は最初から決まっているというものである。その測定値は、3つの方向があり、それぞれの方向を測定すると0か1のどちらかが得られる。これにより000, 001, 010, 011,100, 101, 110, 111の8つの構成が得られる。

左の桁が0の場合
　|↑>, 1の場合|↓>
中央の桁が0の場合
　|↘>, 1の場合|↖>
右の桁が0の場合
　|↙>, 1の場合|↗>



もしアリスの量子ビットが001である場合ボブの量子ビットも同じである

古典モデル　２

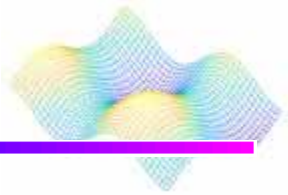


例えば、二人の電子が001にあり、アリスが1番目の基底(|↑>, |↓>)を使い、ボブが3番目の基底(|↙>, |↗ >)で測定した場合、アリスの測定値は0，ボブの測定値は1となり一致しない。

	Measurement directions								
Config.	(a,a)	(a,b)	(a,c)	(b,a)	(b,b)	(b,c)	(c,a)	(c,b)	(c,c)
000	A	A	A	A	A	A	A	A	A
001	A	A	D	A	A	D	D	D	A
010	A	D	A	D	A	D	A	D	A
011	A	D	D	D	A	A	D	A	A
100	A	D	D	D	A	A	D	A	A
101	A	D	A	D	A	D	A	D	A
110	A	A	D	A	A	D	D	D	A
111	A	A	A	A	A	A	A	A	A

左の図は測定結果が一致しているかどうかを表した図である

古典モデル　３



量子論モデルでは、アリスとボブの配列は、ちょうど半分の確率で一致する。それに対して古典モデルでは、少なくとも9分の5の確率で一致する（ベルの不等式）。

ジョンクラウザーとスチュワートフリードマンがどちらが正しいかの実験を行った結果、量子論モデルが正しいことが分かった。
しかし、量子論モデルの方に確認できない問題点があり、古典モデルが正しい可能性が残されていた。

古典モデル　４

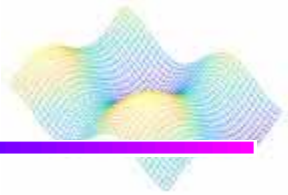
その問題点は、１．アリスとボブの距離が近すぎること
２．絡み合った粒子の数が多すぎること
３．測定方向の選択がランダムではなかったこと

1.2の問題点は、ダイヤモンドに閉じ込められた電子が光子と絡み合っていることを利用することで解決することができた。

3の問題点は、完全に証明することはできないが、量子力学的に生成された文字列が無相関であるという結論に至った。

これにより、古典モデルよりも量子論モデルの方が正しいという結論に至った

計測

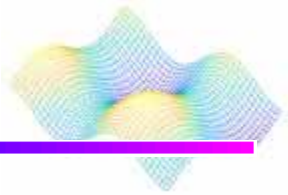


- 量子力学では、測定を行うと状態ベクトルが基底ベクトルにジャンプするとされている。
- 測定を行う前まではすべてが決定論的であり、測定を行うと基底ベクトルのいずれかに移動し、決定論的なものから確率論的なものに変わる。
- 量子力学の一般論では、測定を行うとシュレディンガーの波動方程式の解が崩壊するとされている。
- 標準的な説明では、測定にはミクロの装置との相互作用が伴うとされている。
- 測定装置は、古典力学で記述できる程度の大きさであり、量子論的分析に組み込む必要はない。測定を行う時は、測定対象物と物理的に相互作用しなければならず、この相互作用がジャンプの原因となる。

2021/05/11

量子鍵配布のためのEkertのプロトコル

量子鍵配布のためのEkertのプロトコル



【Ekertのプロトコル】

・1991年　qubitのもつれを利用した方法

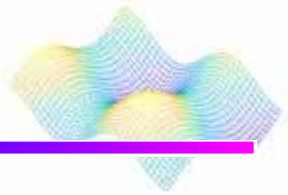
AliceとBob qubitをそれぞれ受け取る

$$\frac{1}{\sqrt{2}}|\uparrow\rangle\uparrow\rangle+\frac{1}{\sqrt{2}}|\downarrow\rangle\downarrow\rangle$$

・同じ正規基底で測定
　→ビット列は0か1のランダムな並びになる

しかし，EveがBobのqubitを傍受して，標準基底で測定し，
　Bobに送り返す．
→Alice，Bob，Eveのビット列が同じに！

量子鍵配布のためのEkertのプロトコル

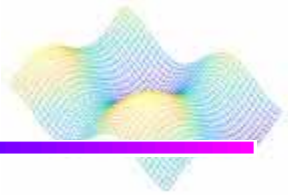


【Ekertのプロトコル】

・3つの基底をランダムに選択し，**qubit**を測定，記憶
測定のたびに結果と，選択した基底を書き留める

・**3n**回の測定後，**Alice**と**Bob**で比較
→二人が同じ基底を選んだら，同じ測定がされる
n個の**0**と**1**の文字列が得られ，一致すると鍵になる.

量子鍵配布のためのEkertのプロトコル



【傍受の確認方法】

・AliceとBob

　異なる基底を選択したときに生じる0と1の文字列を比較

長さが2nの0と1の文字列ができる

誰も傍受していない場合

→ベルの不等式より，文字列が1/4の割合で一致

Eveが傍受している場合

→AliceとBobの文字列の一致する割合が3/8になる

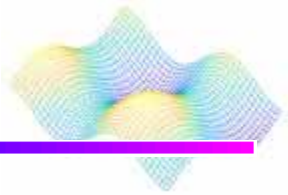
これにより，Eveの存在が確認できる！

文字列の一致が1/4だと鍵として使用できる

2021.4

6章

6，古典的な論理ゲートと回路



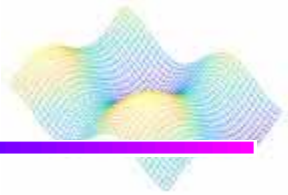
【論理ゲート】

・19世紀後半　**George Boole**
ブール関数

・1930s **Claude Shannon**
ブール関数から論理ゲートの発展

・**1980s Richard Feynman**
　カリフォルニア工科大学で，計算機工学の授業
興味を持った理由
→ **Edward Fredkin**との交流

6．古典的な論理ゲートと回路



【Edward Fredkinとの交流】

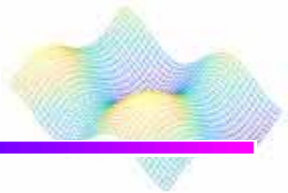
・宇宙はコンピューターであり，物理法則は可逆的である．
したがって，可逆的な計算や可逆的なゲートを計算するべきだ．

→あまり受け入られてないが，型破りなアイディアはすばらしいと
　評価

・ファインマンの著書
ビリヤードボールコンピューター
ボールを互いに跳ね合わせることであらゆる計算が可能になる

→ボールの代わりに粒子を使ってはどうか？

6．古典的な論理ゲートと回路



・Negation 否定

ある命題が真→否定は偽

Ex 2＋2＝4　→ 2＋2 ≠ 4

P	¬P
T	F
F	F

・And

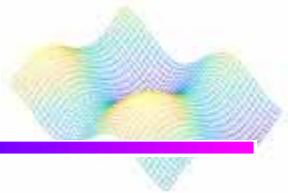
PとQ2つが真の場合真

P	Q	P∧Q
T	T	T
T	F	F
F	T	F
F	F	F

2021.5.11

論理演算について(p91~p95)

And（論理積）

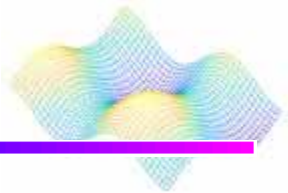


１列目と2列目に命題Pと命題Qがあり，真の場合T，偽の場合Fとする．
3列目はP∧Qであり，命題P,Qの双方ともTの時に限り，P∧QもTとなる．

→PかつQが真なら結果は真である

P	**Q**	**P∧Q**
T	T	T
T	F	F
F	T	F
F	F	F

Or（包括的論理和）

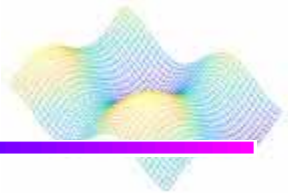


命題PとQがあり，orの記号はＶであらわす．
orはPとQのどちらか一方が真（T）であれば，ＰＶＱは真（T）である．

→PまたはQが真なら結果は真であるといえる

P	**Q**	**PVQ**
T	T	T
T	F	T
F	T	T
F	F	F

XOR（exclusive or,排他的論理和）

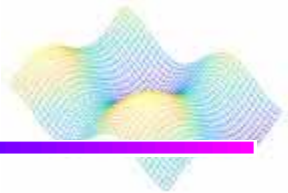


命題PとQがあり，exclusive orの記号は⊕であらわす．
exclusive orはPとQのどちらか一方が真（T），もう一方が偽(F)であれば
P ⊕ Qは真(T)である．

※orとの違いは，PとQの双方が真（T）の時はP ⊕ Qは偽(F)となる

P	Q	P ⊕ Q
T	T	F
T	F	T
F	T	T
F	F	F

ブール代数



命題PとQの真理値表において，具体例として¬(¬P∧¬Q)を1ステップずつ考える.

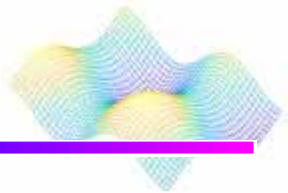
PとQの真理値表がある　→　次に，PとQの否定である，¬P，¬Qを求める

P	Q
T	T
T	F
F	T
F	F

→

P	Q	¬P	¬Q
T	T	F	F
T	F	F	T
F	T	T	F
F	F	F	T

ブール代数(Boolean Algebra)



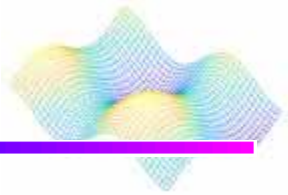
次は，求まった¬P，¬Qを用いて ¬P∧¬Qを求める．

P	Q	¬P	¬Q
T	T	F	F
T	F	F	T
F	T	T	F
F	F	T	T

→

P	Q	¬P	¬Q	¬P∧¬Q
T	T	F	F	F
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

ブール代数(Boolean Algebra)



¬P∧¬Qが求まったので，¬(¬P∧¬Q)を求める.

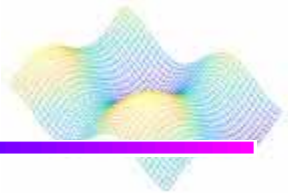
P	Q	¬P	¬Q	¬P∧¬Q	¬(¬P∧¬Q)
T	T	F	F	F	T
T	F	F	T	F	T
F	T	T	F	F	T
F	F	T	T	T	F

ここで中間のステップを省略すると，次の表が出来上がる.

→

P	Q	¬(¬P∧¬Q)
T	T	T
T	F	T
F	T	T
F	F	F

論理的等価性



ここで，P∨Qと¬(¬P∧¬Q)は同一の結果となることに気づく.

P∨Q	¬(¬P∧¬Q)
T	T
T	T
T	T
F	F

よって，次のように表記する.

→　P∨Q≡¬(¬P∧¬Q)

ということは，∨　⇒　¬　&　∧　で置き換え可能

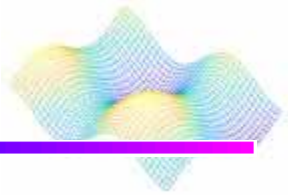
- exclusive orの⊕は¬　&　∧で置き換え可能か？

P	Q	P ⊕ Q
T	T	F
T	F	T
F	T	T
F	F	F

P ⊕ QのTを¬＆∧で表す

→ P∧¬Q ,¬ P∧Q

論理的等価性



- P ⊕ QのT → P∧¬Q , ¬ P∧Q で表現可

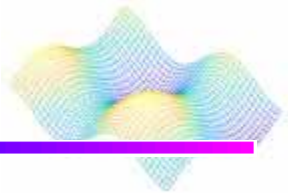
⇒ P ⊕ Q ≡ (P∧¬Q) ∨ (¬ P∧Q)

P∨Q≡¬(¬P∧¬Q)であると先ほど求めたので
∨ ⇒ ¬ & ∧ で置き換え

⇒ P ⊕ Q ≡ ¬ (¬ (P ∧ ¬Q) ∧ ¬ (¬ P∧Q))

exclusive orの⊕は¬ & ∧で**置き換え可能**！

ブール関数



ブール関数

→TまたはFの多くの入力を持ち，それに応じてT，Fのいずれかの値を出力する．

（例） P,Q,Rの三つの入力があり，f(P,Q,R)で出力 → 2^8 通り　の出力

f(P,Q,R) を ¬ & ∧ で表すことを考える！

P	Q	R	f(P,Q,R)
T	T	T	F
T	T	F	F
T	F	T	T
T	F	F	F
F	T	T	F
F	T	F	T
F	F	T	F
F	F	F	T

ブール関数

P	Q	R	f(P,Q,R)
T	F	T	T
F	T	F	T
F	F	F	T

→ f(P,Q,R)がTのものだけ表示
(P∧¬Q∧R),
(¬P∧Q∧¬R),
(¬P∧¬Q∧¬R)　と表す

f(P,Q,R) ≡ (P∧¬Q∧R)∨(¬P∧Q∧¬R) ∨(¬P∧¬Q∧¬R)

→ここで　∨　⇒　¬　&　∧ で置き換え．（P∨Q≡¬(¬P∧¬Q)を使って）
左から順に置き換え

f(P,Q,R) ≡ ¬(¬(P∧¬Q∧R) ∧ ¬(¬P∧Q∧¬R))∨(¬P∧¬Q∧¬R)
さらに

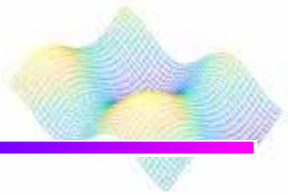
f(P,Q,R) ≡¬(¬ [¬(¬(P∧¬Q∧R) ∧ ¬(¬P∧Q∧¬R))] ∧¬ [¬P∧¬Q∧¬R])

¬　&　∧のみでの表現に成功！

2021.5.11

Nand,Gates,Circuitsについて

Nand :notとandの混成語(↑)

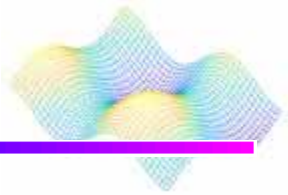


- $P \uparrow Q = \neg(P \wedge Q)$
- Nand自体が関数的に完全であること

どんなブール演算子もNandだけを使った等価な関数に書き換えられること

P	Q	P↑Q
T	T	F
T	F	T
F	T	T
F	F	T

NAND(↑)のみで表現する



Notの置き換え

$$\neg P \equiv \neg(P \wedge P)$$

$\neg(P \wedge P)$は$P \uparrow P$

$$\neg P \equiv P \uparrow P$$

---

andの置き換え

$$P \wedge P \equiv \neg\neg(P \wedge Q)$$

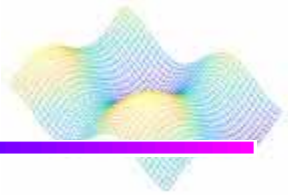
$\neg(P \wedge Q)$は$P \uparrow Q$

$$P \wedge Q \equiv \neg(P \uparrow Q)$$

$\neg P$は$P \uparrow P$

$$P \wedge Q \equiv (P \uparrow Q) \uparrow (P \uparrow Q)$$

Gates：二項演算子に対するスイッチの組合せ



すべてのブール代数を電気的なスイッチで行うことができると示した。

（Claude Shannon）

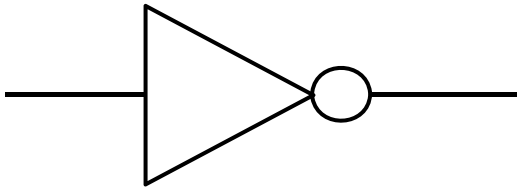
適切な時間間隔で電気のパルスを受信すれば、これは真理値T、つまりビット値1を表している。

受信しない場合、これは真理値F、つまりビット値0を表している。

Gatesの一般例

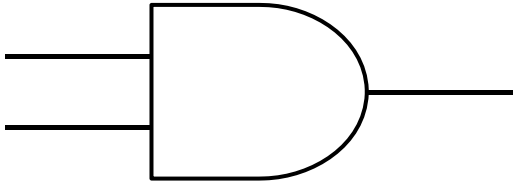
NOTゲート

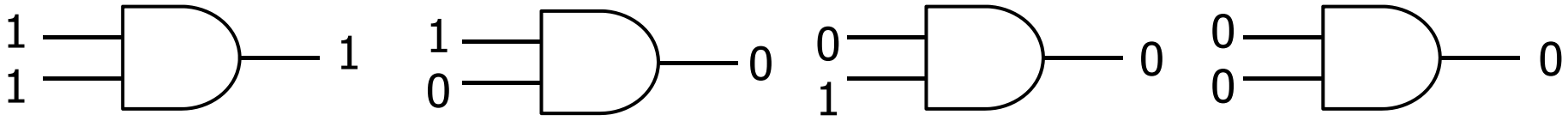
1を入力すれば0が出力され、0を入力すれば1が出力されます。



ANDゲート

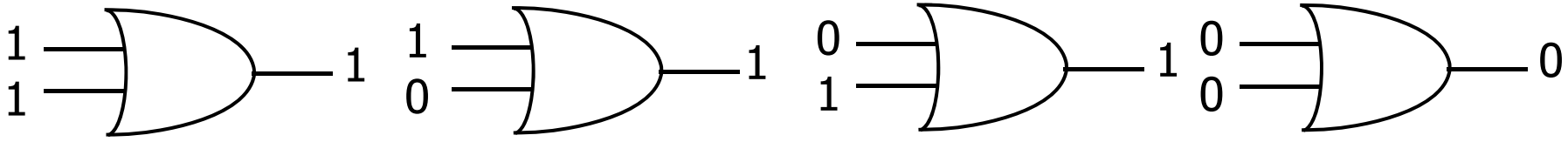
0か1かの2つの入力と1つの出力を持っています。



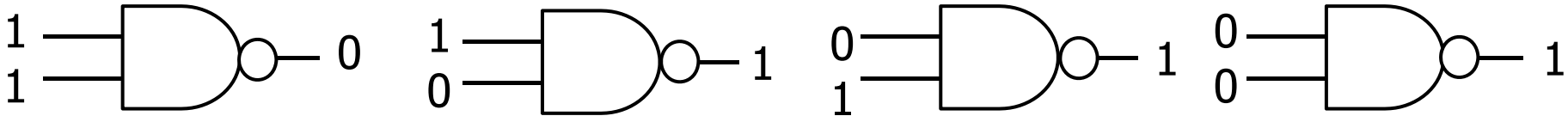


Gatesの一般例

ORゲート



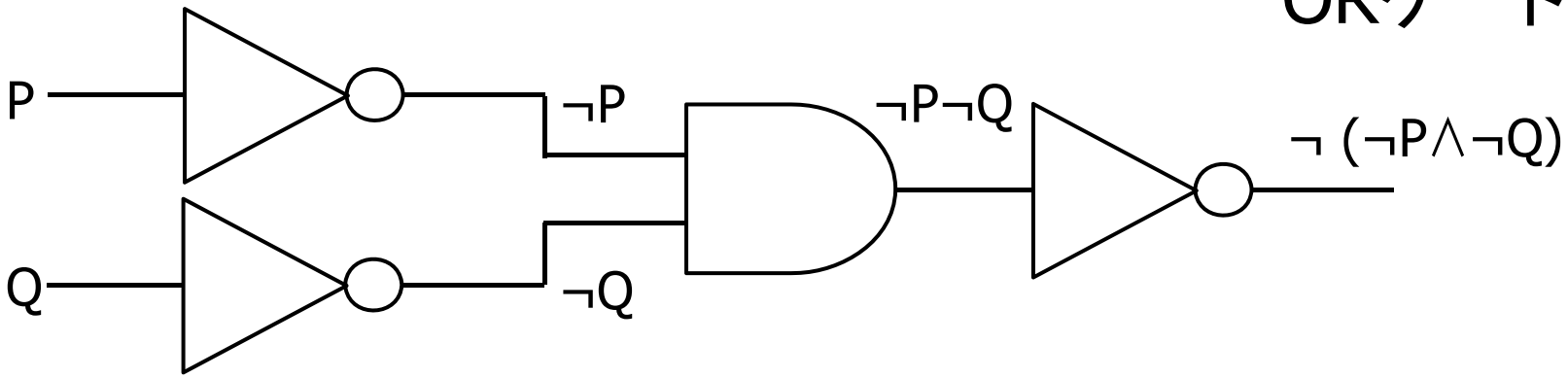
NANDゲート



Circuits：ゲートをつなぎ合わせて回路ができる

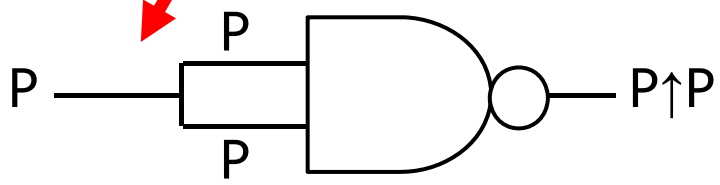
$\neg(\neg P \wedge \neg Q)$

ORゲート



$P \uparrow Q$

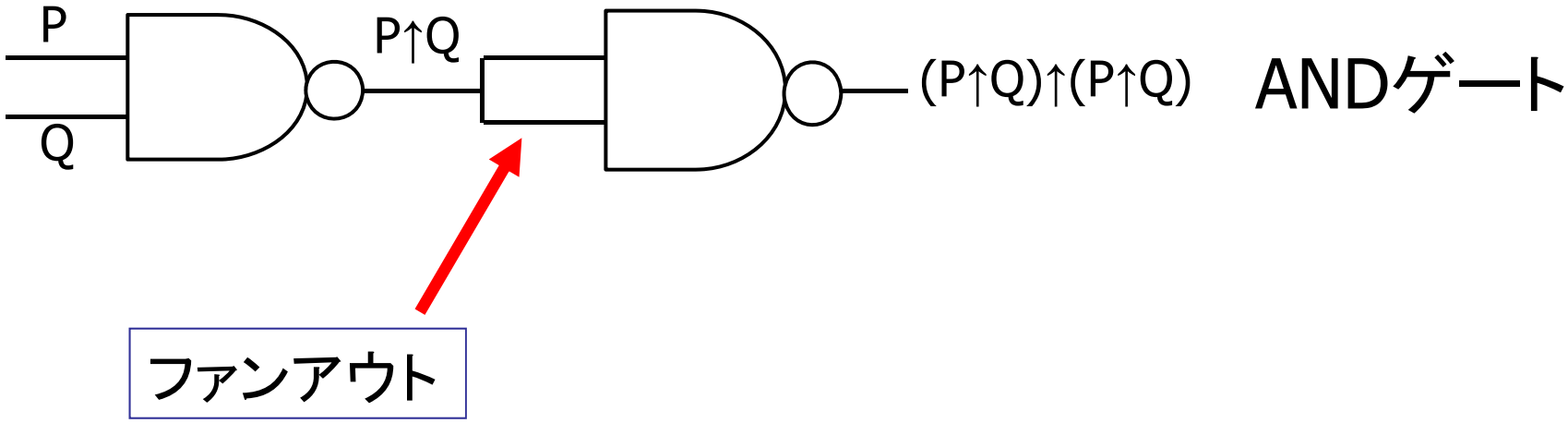
ファンアウト：信号を複数に分割すること



NOTゲート

Circuits：ゲートをつなぎ回路作製ができる

$$(P \uparrow Q) \uparrow (P \uparrow Q)$$



2021/5/18(火)

p.100下3行～p.105

NANDはユニバーサルゲート

PvQ =¬(¬P^¬Q)の発生をNANDゲートに
置き換えることができることを示す

→ NANDはユニバーサルゲート

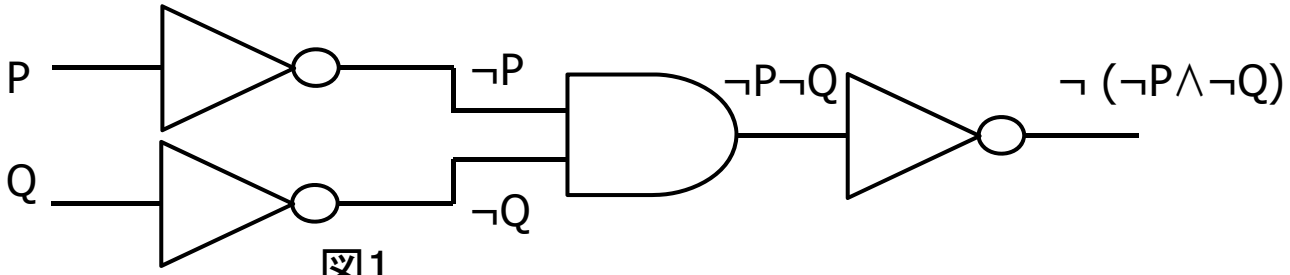


図1

図1の回路は、ORゲートの必要なし
→NOTゲートとANDゲートだけでブール関数を計算する回路を構築

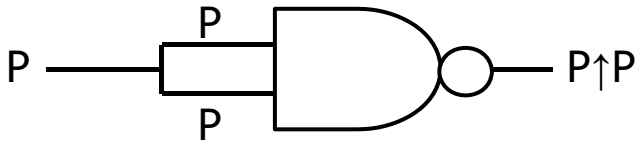


図2

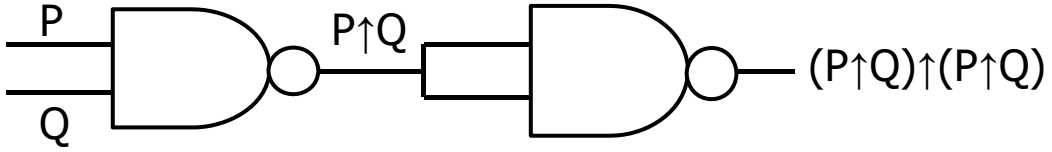
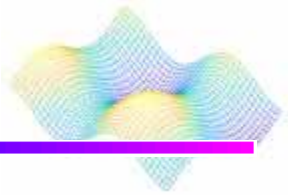


図3

図2と図3の回路は、ORゲートとNOTゲートとANDゲートの必要なし
→NANDゲートだけでブール関数を計算できる回路を構築

ゲートと計算



【ゲート】

現代のコンピューターの基本的な構成要素
論理演算の実行に加えて、ゲートを使用して計算可能

（例）
⊕で示される排他的論理和

0⊕0 = 0　　0⊕1 = 1　　1⊕0 = 1　　1⊕1 = 0

奇数と偶数の整数の加算と比較

偶数+偶数=偶数　偶数+奇数=奇数　奇数+偶数=奇数　奇数+奇数=偶数

この「奇数」と「偶数」の加算は、加算モジュロ2と呼ばれ
0を「偶数」、1を「奇数」とすると、加算モジュロ2は⊕で与えられる。

ゲートと計算

【XOR】

排他的論理和ゲートはXORと呼ばれ
図4の記号で示し
表1のような動きをする

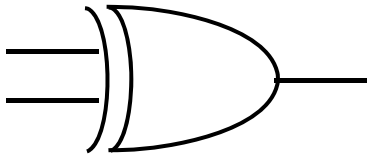


図4　XORゲート

表1

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

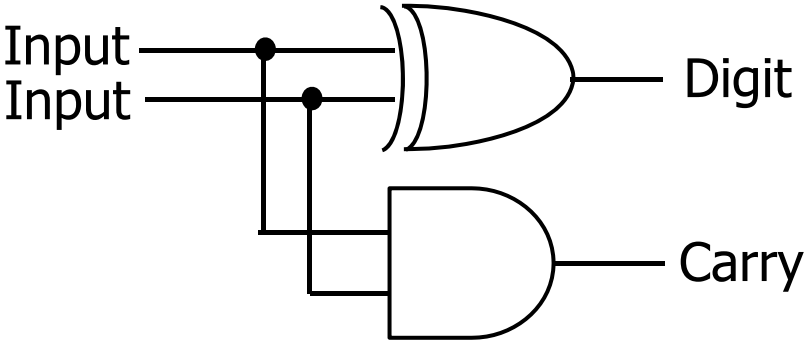


図5　バイナリ半加算器

バイナリ半加算器とはXORゲートとANDゲートで構築
XORゲートは桁、ANDゲートはキャリーを計算

0 + 0 = 0　　キャリー= 0
0 + 1 = 1　　キャリー= 0
1 + 0 = 1　　キャリー= 0
1 + 1 = 0　　キャリー= 1

すべてのゲートをNANDゲートに置き換えることができるため、
➡NANDゲートとファンアウトを使用するだけで加算器を構築
　➡これら2つの要素を使用するだけで、コンピューター全体を構築

記憶

ゲートの使用方法と、ゲートを使用して算術演算する方法を示したが
コンピューターを構築するには、データを格納できる必要あり

→ゲートを用いフリップフロップを作成することで可能

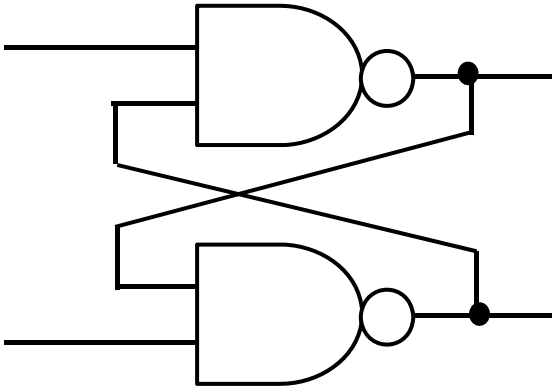
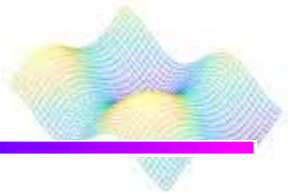


図6　フリップフロップを用いた2つのNANDゲート

2つのNANDゲートを使用した例を図6に示す。
入力と出力のタイミングを正確に取得することが重要で
一定の時間間隔で電気のパルスを送る

可逆計算



【可逆ゲート】出力が与えられた場合、入力が何であったか判断

AND

Input		Output
0	0	0
0	1	0
1	0	0
1	1	1

Half-adder

Input		Output	
		digit	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

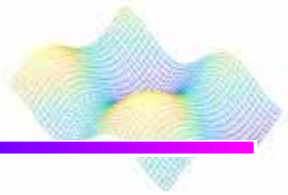
[AND]
出力が1の場合、入力値は両方とも1
出力値が0の場合、入力値のペアが3つあり判断ができない、、、
→ANDは可逆ゲートではない

[半加算器]
1の桁と0の桁上げを与える入力値のペアが2つ
どちらの場合も、2ビットの入力があるが、2ビットの出力は得られない、、、
→計算を行っている情報が損失

ジョン・フォン・ノイマン
情報が失われるとエネルギーが消費され、熱として放散

Rolf Landauer
その結果を証明し、1ビットの情報を消去するために最小限のエネルギー（ランダウアー限界）を与えた。
ただし、計算が可逆的である場合、情報が失われず、
理論的にはエネルギー損失なし

制御された　NOT GATE



制御されたnotゲート、CNOTゲートは、2つの入出力を持つ

この関数は、f(x、y)=(x、xy)で与えられる

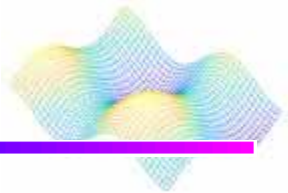
最初の出力制御ビットは最初の入力ビットであり、
xで示されます
　このビットは変更されず、最初の出力になる

2番目の出力は
制御ビットが0の場合　→　2番目のビットには影響なし
→2番目の入力と同じ

制御ビットが1の場合　→　2番目のビットを反転させる
→2番目の入力を反転

Billiard Ball computing

ビリヤードボール・コンピュータ

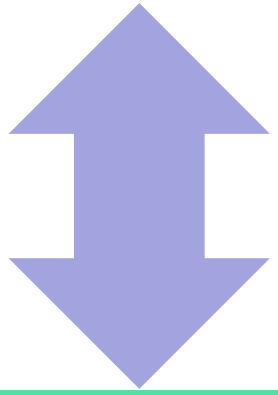


ボールの力学的な運動を基にした可逆計算モデル

エドワード・フレドキンにより提案

一般的なコンピュータ

電流電圧で情報伝達

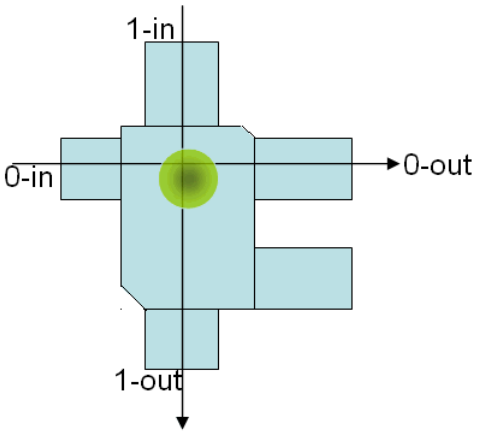


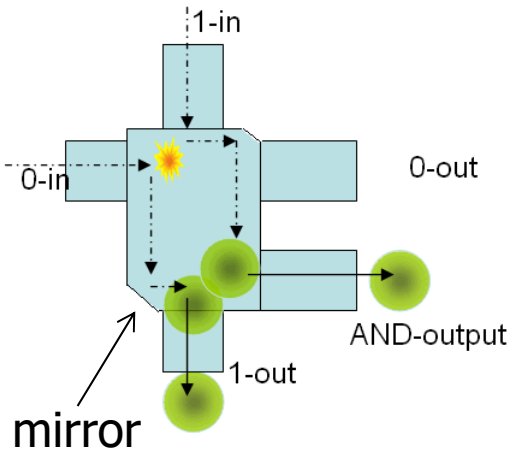
ビリヤードボール・コンピュータ

ボールの衝突・直線運動で情報伝達

仕組み

条件
摩擦ゼロ
完全弾性衝突
→エネルギー損ゼロ





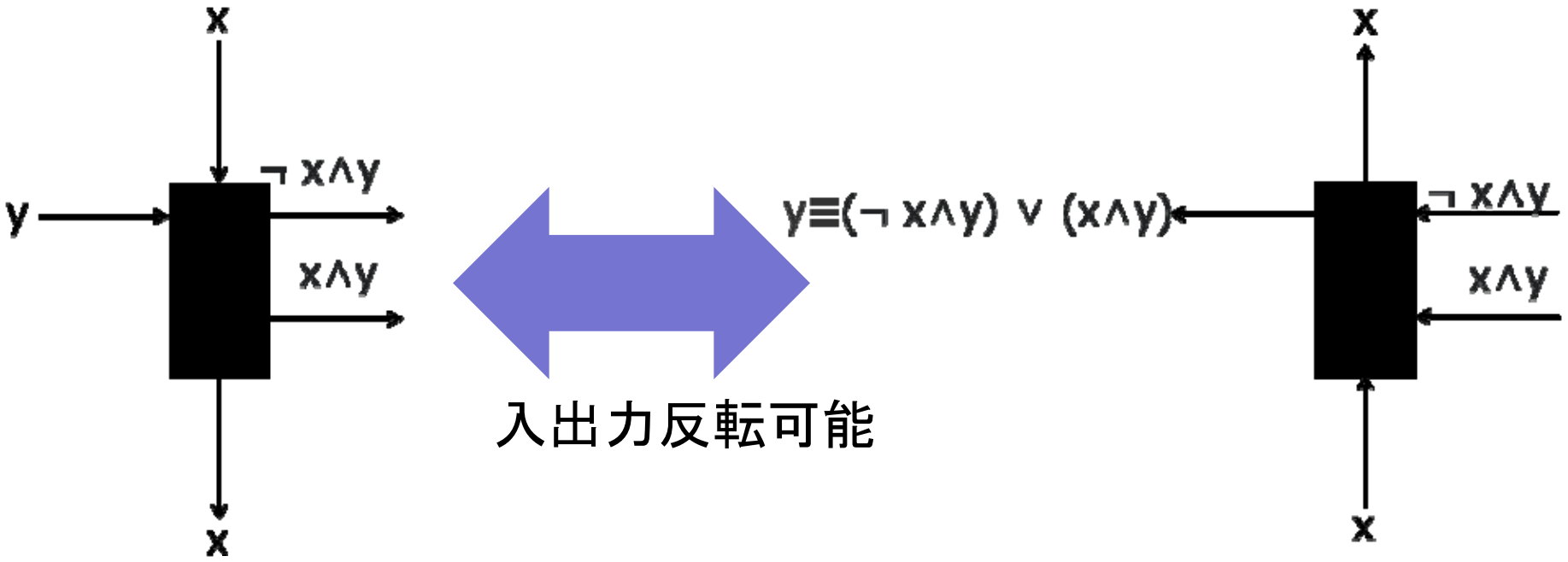
鏡（壁）
入射角＝反射角

2入力時
ボールが入れ替わる

可逆ANDゲート

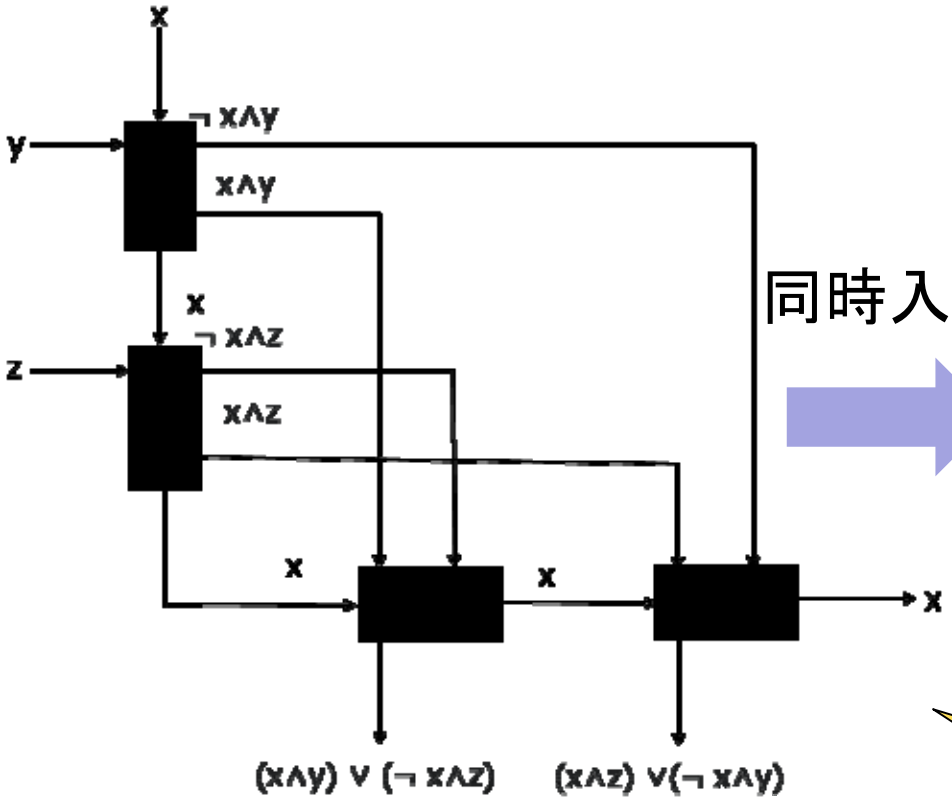
Input		Output		
1	0	1	0	AND
0	0	0	0	0
0	1	0	1	0
1	0	1	0	0
1	1	1	0	1

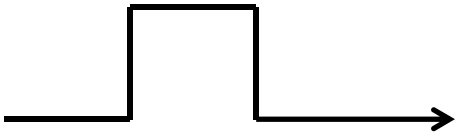
可逆性



入出力反転可能

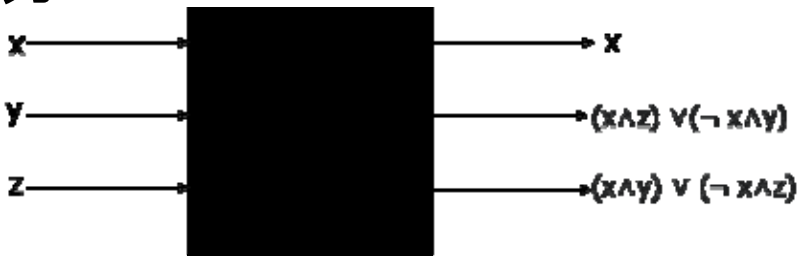
フレドキンゲート





鏡を使って遅延することで…

同時入出力

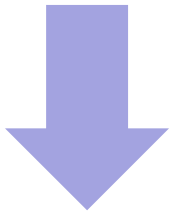


どの論理回路で
も作製可能

実現性

実際は完全弾性衝突ではない

摩擦有、熱損失有



理論上の機械

実際には作製出来ない

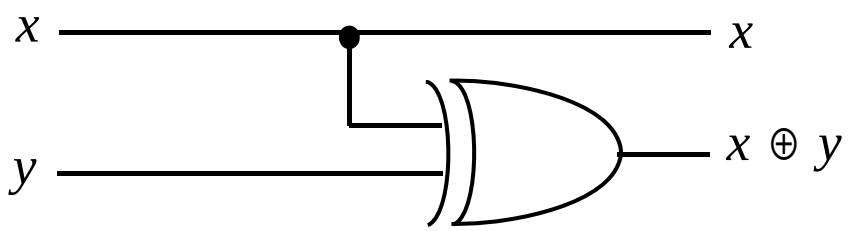
しかし、これが量子力学に基づくゲートのきっかけ

7章　論理ゲートから

CNOTゲート

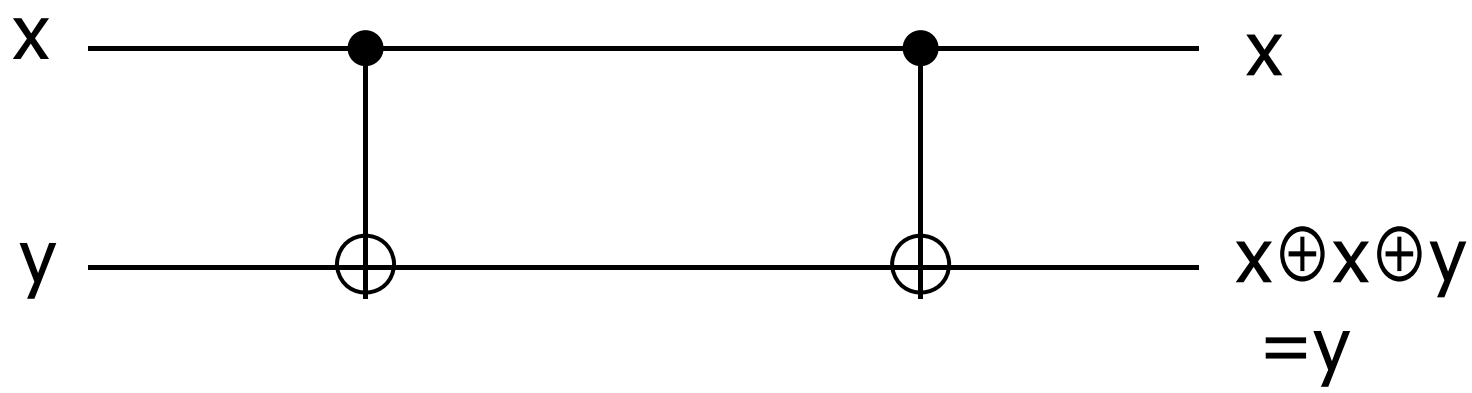
CNOTゲート

x	y	x	x⊕y
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0



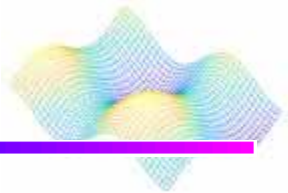
CNOTゲートは入力が二つあり、xは制御ビットである。
xが0ならyをそのまま出力し、xが1ならyを反転させる
2つの出力のうち、1つはxの入力をそのまま出力する
1つは入力のxと等価
もう一つはx⊕yの出力

CNOTゲート

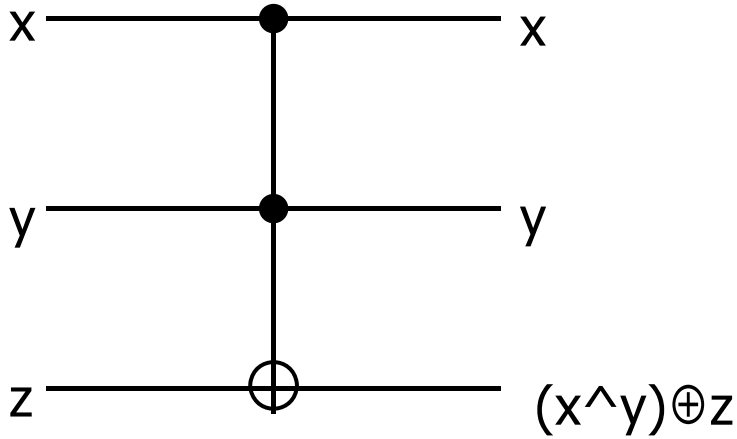


上記のようにCNOTゲートを直列につないだ場合、出力は
入力と等価である
$x \oplus x = 0$, $0 \oplus y = y$
ここからCNOTゲートは自身の逆数であることがわかる

Toffoliゲート



Toffoliゲートはx,yがともに1ならzを反転させ、それ以外ならzをそのまま出力させる。制御ビットが2つあるのでCCNOTゲートとも言われる

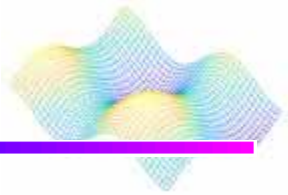


x	y	z	x	y	(x^y)⊕z
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

また、CNOTゲートと同様にしてToffoliゲートも自身の逆数であることがいえる

$$(x,y,(x\hat{}y)\oplus z) = (x,y,(x\hat{}y)\oplus(x\hat{}y)\oplus z) = (x,y,z)$$

Toffoliゲート



ToffoliゲートにはNANDゲートと同じ論理の完全性の性質がある

NANDゲートは以下のように示せる

$$f(x,y) = \neg(x \wedge y)$$

ここで

$$\neg(x \wedge y) = (x \wedge y) \oplus 1$$

であることを確認する

よってToffoliゲートでNAND出力を得るには

$T(x,y,1) = (x,y,(x \wedge y) \oplus 1) = (x,y,\neg(x \wedge y))$

とすればよい

また、1つの入力を2つに増やすには

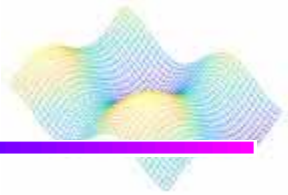
$$T(x,1,0) = (x,1,x)$$

とする

XOR

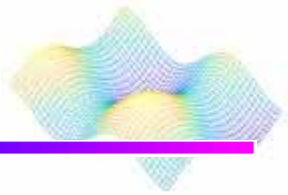
x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

アンシラビットとガベレージビット



前ページのように接続したい入出力と実際のゲートの入出力の数が合わないなどで入力される0または1のことをアンシラビット、その数自体に意味を持たない出力ビットをガベレージビットという。

Fredkinゲート



Fredkinゲートはx,y,zの3つの入出力を持つ。このうちxは制御ビットである。0ならy,zはそのまま出力される。1ならy,zは入れ替えられて出力される

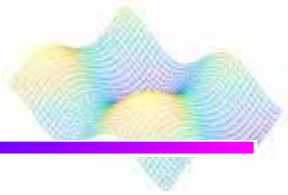
$$F(0,y,z) = (0,y,z)\ F(1,y,z) = (1,z,y)$$

また、CNOTゲートやToffoliゲートと同じようにこのゲートは自身の逆数となっている。

また、入力と出力の1の数が一致しており、この性質は次の章のビリヤードボールゲートに生かされる

x	y	z	x		
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Fredkinゲート



y=0,z=1とすると

F(0,0,1) = (0,0,1)

F(1,0,1) = (1,1,0)

F(x,0,1) = (x,x,¬x)

NOTゲート、出力を2つに増やすことの両方が可能

またz=0とすると

F(0,0,0) = (0,0,0)

F(0,1,0) = (0,1,0)

F(1,0,0) = (1,0,0)

F(x,y,0) = (x,¬x^y,x^y)

F(1,1,0) = (1,0,1)

ANDの出力を得ることができる

NOT,ANDの出力を得ることができるため、Fredkinゲートですべての論理ゲートが実現できる

Fredkinゲート

Fredkinゲートを1つの式で表す

1つ目の出力はx

2つ目の出力は x=0かつy=1 または x=1かつz=1のとき1

$$(\neg x \wedge y) \vee (x \wedge z)$$

3つ目の出力は x=0かつz=1 または x=1かつy=1のとき1

$$(\neg x \wedge z) \vee (x \wedge y)$$

よって

$$F(x,y,z) = (x, (\neg x \wedge y) \vee (x \wedge z), (\neg x \wedge z) \vee (x \wedge y)$$

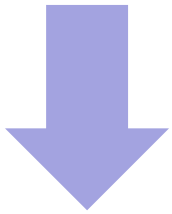
で示される

この式は複雑であるが次のビリヤードボールゲートで用いる

実現性

実際は完全弾性衝突ではない

摩擦有、熱損失有



理論上の機械

実際には作製出来ない

しかし、これが量子力学に基づくゲートのきっかけ

P. 118～119

UNIVERSITY
OF TOYAMA

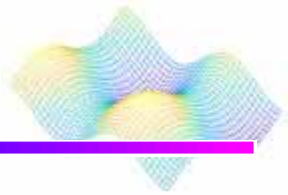
量子ビット

$$\left(\begin{bmatrix}1\\0\end{bmatrix},\begin{bmatrix}0\\1\end{bmatrix}\right) \longrightarrow (|\uparrow\rangle,|\downarrow\rangle)$$

$$(|\uparrow\rangle,|\downarrow\rangle) \longrightarrow |0\rangle \ , \ |1\rangle$$

$$|0\rangle \longrightarrow \begin{bmatrix}1\\0\end{bmatrix} \qquad |1\rangle \longrightarrow \begin{bmatrix}0\\1\end{bmatrix}$$

量子ビット



1量子ビットは$\alpha_0|0\rangle+\alpha_1|1\rangle$と表現される

ここでα_0，α_1は $\alpha_0^2+\alpha_1^2=1$を満たす

$|0\rangle$を得る確率はα_0^2 ,$|1\rangle$を得る確率はα_1^2

量子ビット

$$\left(\begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix},\begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix},\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix},\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}\right)$$

$$\downarrow$$

$$(|0\rangle\otimes|0\rangle,|0\rangle\otimes|1\rangle,|1\rangle\otimes|0\rangle,|1\rangle\otimes|1\rangle)$$

$$\downarrow$$

$$(|0\rangle|0\rangle,|0\rangle|1\rangle,|1\rangle|0\rangle,|1\rangle|1\rangle)$$

$$\downarrow$$

$$(|00\rangle,|01\rangle,|10\rangle,|11\rangle)$$

7　量子ゲートと回路

p.119~p.121 5行目

UNIVERSITY
OF TOYAMA

① CNOTゲート

CNOT

Input		Output	
x	y	x	$x \oplus y$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

表1

CNOT

Input		Output	
x	Y	x	$x \oplus y$
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

表2

CNOT

Input	Output
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|01\rangle$
$\|10\rangle$	$\|11\rangle$
$\|11\rangle$	$\|10\rangle$

表3

基底ベクトルの線形結合

CNOT(r|00⟩+s|01⟩+t|10⟩+u|11⟩)
=r|00⟩+s|01⟩+u|10⟩+t|11⟩

① CNOTゲート

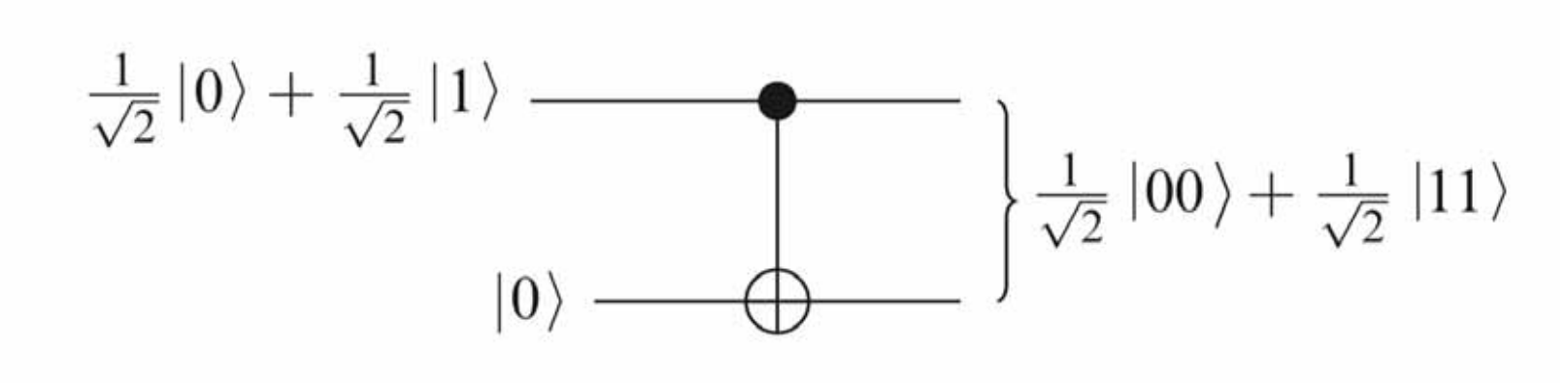


図1

もつれた状態

片方を測定するともう片方の状態に影響を与える

② 量子ゲート

基底ベクトルを並べ替えると別の正規直交基底が得られ
基底のいずれにも直交行列がある

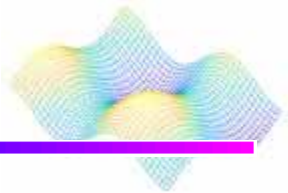
CNOTゲートに対応する行列は直交する

量子ゲートとは、直交行列で記述できる演算

P.121～125

UNIVERSITY
OF TOYAMA

1つの量子ビットに対応する量子ゲート



従来の可逆計算

・ビットを変更しない恒等式
・0と1の値を反転する*NOT*

量子ビット

・無限に存在

パウリ変換

・恒等式に対応する2つの量子ゲート
・量子ビットの0と1を反転する2つの量子ビット

Iゲート　Zゲート

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

任意の量子ビット $a_0|0\rangle + a_1|1\rangle$に反映

・$I(a_0|0\rangle + a_1|1\rangle) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = a_0|0\rangle + a_1|1\rangle$

・$Z(a_0|0\rangle + a_1|1\rangle) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} a_0 \\ -a_1 \end{bmatrix} = a_0|0\rangle - a_1|1\rangle$

Iは恒等式として機能するため量子ビットは変化しないが、
Zは$|1\rangle$の確率振幅の符号を変える

Zゲートが基底ベクトルにどのように作用するのか？

$$\begin{cases} Z(|0\rangle) = |0\rangle \\ Z(|1\rangle) = -|1\rangle = |1\rangle \end{cases}$$

＊状態ベクトルの特性

両方の基底ベクトルを保存しているにも関わらず、1つおきの量子ビットを変更している

量子ビットの
相対的な位相変化

*NOT*ゲート　アダマールゲート

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

NOT ゲート

Yは相対的な位相を変化

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

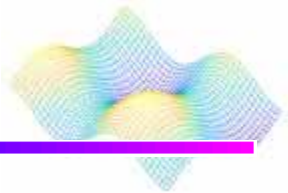
アダマールゲート

標準基底ベクトルを
重ねるときに使用

1つの量子ビットに作用する5つの量子ゲートを挙げたが、他にも無限に存在している

1つの量子ビットに作用するゲートは右図のように示す

普遍的な量子ゲートは存在するのか？



これまでの計算

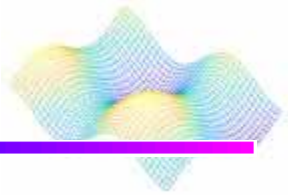
・全てのブール関数はブレドキンゲートのみを使った
回路で与えられ、フレドキンゲートが万能である
・NANDもファンアウトも普遍的である

量子ゲートにも万能なものがあるのか？

有限個のゲートを使用して有限個の方法で接続すると、限られた数の回路になる

有限個のゲートで無限個の回路を作るのは不可能

量子複製不可能定理



ファンアウト操作…入力信号が二つの同じコピーに分割される

可逆論理ゲートに着目。このゲートは2つの出力があれば2つの入力が必要であるが、アンシラビットを使用し2つ目の入力を常に0にすることでファンアウト操作を実現

*CNOT*ゲートを使用

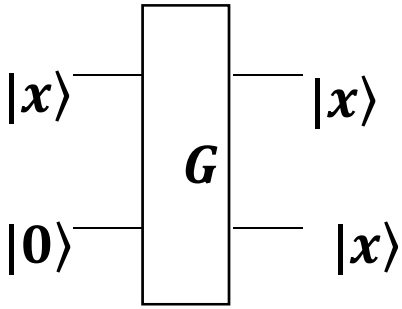
$$CNOT(|x\rangle|0\rangle) = |x\rangle|x\rangle$$
$$(|x\rangle = |0\rangle\ or|1\rangle)$$

しかし…

$|x\rangle = |0\rangle\ or|1\rangle$でない場合、コピーするのは不可能

量子コンピューティングに活用するために、この二つ以外（ビット）のコピーも可能にしたい

量子ビットのクローンを作成するのは不可能であることの証明



右図のような量子のクローンを作成出来るゲートGが存在すると仮定し、矛盾することを導く

・Gが存在すると仮定すると、Gのクローン特性は、

1. $G(|0\rangle|0\rangle) = |0\rangle|0\rangle$　2. $G(|1\rangle|0\rangle) = |1\rangle|1\rangle$

3. $$G\left(\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)|0\rangle\right) = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)$$

・これらの式は、以下のように置き換えられる

1. $G(|00\rangle) = |00\rangle$　2. $G(|00\rangle) = |11\rangle$

3. $$G\left(\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle\right) = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

・Gは他の行列演算子同様に線形でなければならないため、

$$G\left(\frac{1}{\sqrt{2}}|00\rangle+\frac{1}{\sqrt{2}}|10\rangle\right)=\frac{1}{\sqrt{2}}G(|00\rangle)+\frac{1}{\sqrt{2}}G(|10\rangle)$$

・前式より、右辺を以下のように置き換える

$$G\left(\frac{1}{\sqrt{2}}|00\rangle+\frac{1}{\sqrt{2}}|10\rangle\right)=\frac{1}{\sqrt{2}}|00\rangle+\frac{1}{\sqrt{2}}|11\rangle$$

・しかし、前式で $G\left(\frac{1}{\sqrt{2}}|00\rangle+\frac{1}{\sqrt{2}}|10\rangle\right)=\frac{1}{2}(|00\rangle+|01\rangle+|10\rangle+|11\rangle)$
と述べられているため、
$\frac{1}{\sqrt{2}}G(|00\rangle)+\frac{1}{\sqrt{2}}G(|10\rangle)=\frac{1}{2}(|00\rangle+|01\rangle+|10\rangle+|11\rangle)$という等式
が成り立つということになり、矛盾していることが分かる

Gは存在しない

p.126-130

UNIVERSITY
OF TOYAMA

非クローン理論

キュービットを複製することが出来ないことは、多くの重要な結果を持っています。私たちは、常にファイルをバックアップし、他の人にコピーを送ります。コピーは、ユビキタスです。我々の各々のコンピューターは、フォンノイマンの基本概念設計に基づきコピー能力に非常に基礎を置いています。計算が始まるとビットをある場所から他へ常にコピーします。量子コンピューティングの一般的キュービットでは、これは不可能です。もしプログラマブルなコンピュータが設計されたとすると、それは我々の基本概念設計に基づくものでは有りません。

第一に、キュービットがクローン化出来ないことは、重大な欠点の様なきがしますが、しかし多くの重要なコメントを行う必要が有ります。

多くの場合、我々はコピーを防ぎたいと思います。データーを保護したいのですが、通信を盗まれたくないのです。ここでは、イブと一緒に見たように、キュービットのクローンを作成できないと言う事実は、不要なコピーを防ぐために利用できます。

第二のコメントは非常に重要なため、独自のセクションで議論します。

量子コンピューテーション対古典コンピューテーション

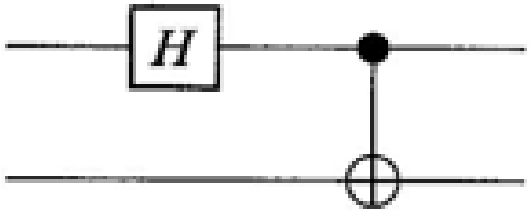
キュービット|0⟩と|1⟩は0と1に対応します。もし、キュービット|0⟩と|1⟩による量子CNOTゲートを実行し、重ね合わせを使用しない場合、計算は以下の様に0と1で古典的CNOTゲートを走らせた場合と全く同じになります。同じことが、古典的Fredkinゲートの場合でも、量子的Fredkinゲートで|0⟩と|1⟩を用いる場合でも同じことが起こり、量子回路では、古典的回路で計算できるものは何でも計算できることが分かります。量子複製不可能定理は気になる様に思われるかもしれませんが、それは私たちが古典的な計算を行うことを制限するものでは有りません。

これは深い結果です。これは、古典計算と量子計算を比較する場合、これらを異なるタイプの計算と見なすべきではないことを意味しています。量子計算には、全ての古典的計算が含まれています。これは、より一般的な計算です。キュービットは、計算の基本単位であり、ビットではありません。

いくつかの基本的ゲートを見たので、回路を構成するために、これらを一緒に接続してみます。

ベルの回路

下記の量子回路を、ベル回路と呼ぶ。

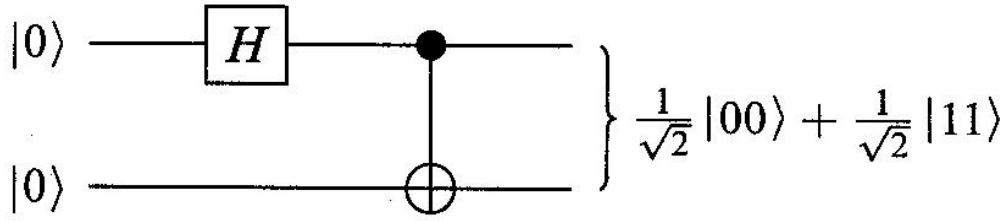


$|00\rangle$=$|0\rangle|0\rangle$とし、アダマールゲートは、それを$\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle$と変換して

$$\left(\frac{1}{\sqrt{2}}\left|0\right\rangle+\frac{1}{\sqrt{2}}\left|1\right\rangle\right)|0\rangle=\frac{1}{\sqrt{2}}|00\rangle+\frac{1}{\sqrt{2}}|10\rangle$$

CNOTゲートでは、$|10\rangle$が$|11\rangle$となり、最終状態は$\frac{1}{\sqrt{2}}|00\rangle+\frac{1}{\sqrt{2}}|11\rangle$となる

次に下記の条件に付いて考えると



ベルの回路(2)

まとめると、

$$\mathrm{B}(|00\rangle) = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle$$

同様に解くと、

$$\mathrm{B}(|01\rangle) = \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle$$

$$\mathrm{B}(|10\rangle) = \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle$$

$$\mathrm{B}(|11\rangle) = \frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle$$

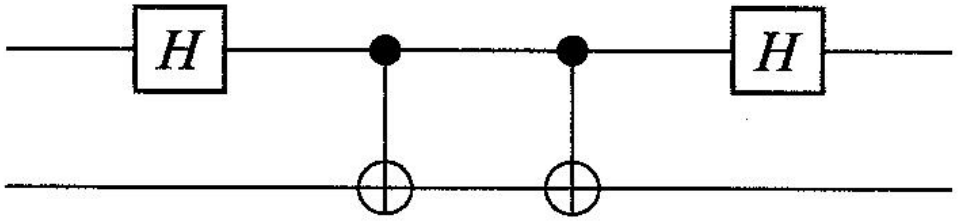
これは$\mathbb{R}^4$の正規直交基底を与える4つのエンタングルしたケットであり、
ベルの基底と呼ばれる。

ベルの回路(3)

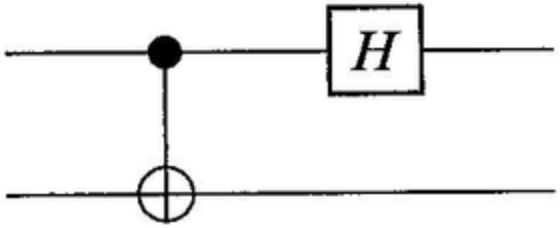
ここで正方行列Aと転置行列A^Tを考えると、全く同じ行列となり、全てのゲートでAA＝I(単位行列)となる。すなわち、2回のゲート操作で元に戻る。

以下、ベルの回路の使用法を幾つか述べる。

最初に、アマダールゲートとCNOTが、それぞれ逆である事実を用いる。



このアダマールゲート、CNOT回路に続くCNOT、アダマールゲートで、元に戻される（逆ベル回路）。



ベルの回路(4)

入力　$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ として、出力　$|00\rangle$

入力　$\frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle$ として、出力　$|01\rangle$

入力　$\frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle$ として、出力　$|10\rangle$

入力　$\frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle$ として、出力　$|11\rangle$

以下、超高密度暗号化と量子テレポーテーションについて述べる。

超高密度暗号化

絡み合ったスピンを持つ2つの電子 $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$

一方をアリス、他方をボブに与える。彼らは離れてその絡み合った状態を見ないこととする。

アリスはボブへ、2つの古典的情報を送信する。 00, 01, 10, 11

彼女は、ボブへ1キュービットを送りこれを行う（彼女の電子）。

最初、アリスはボブへ1キュービット$a_0|0\rangle + a_1|1\rangle$を送信し、ボブは、$|0\rangle$か$|1\rangle$のスピンを測定する。

アリスが$a_0|0\rangle + a_1|1\rangle$を送信　ボブの確率$|0\rangle \Rightarrow a_0^2,\ |1\rangle \Rightarrow \mathrm{a}_1^2$

初期　アリスとボブ　1電子づつ持つ

最終　ボブ　両電子を持ちスピンを測定、2つのワイヤを持つ量子回路を持つ

アリス　00送信　ボブ　電子対の絡み合っていない$|00\rangle$

01送信　ボブ　測定以前の状態を$|01\rangle$にする必要が有る　....

超高密度暗号化(2)

最終観察

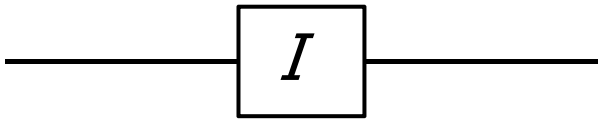
ボブ　受信する全ての電子対に対して同じことを行う。

アリスが、何を送ろうとしているのか分からないので、異なることが出来ない。

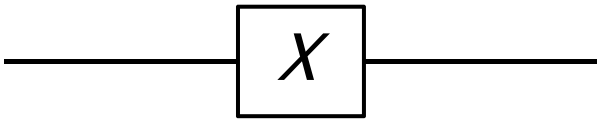
背景では、アリスが4方法の1つで、自身の電子に作用する。

いずれの方法でも、キュービットがベル基底の基底ベクトルの一つになる。ボブは、逆ベル回路で正しく絡み合っていない状態を取得する。

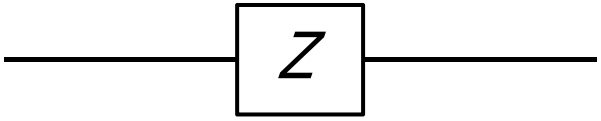
アリスは4つの量子回路を持つ。2ビットの選択肢ごとに一つ。各回路は、以下のパウリゲートを持つ。



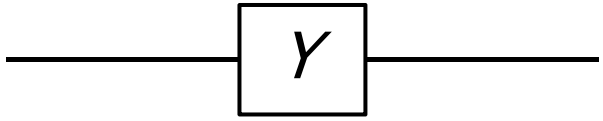
00用回路



01用回路



10用回路



11用回路

量子テレポーテーション p131,p132

量子テレポーテーション

ボブ

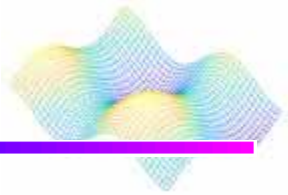
アリス

超高密度符号化のように離れている

電子のもつれた状態

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

量子テレポーテーション



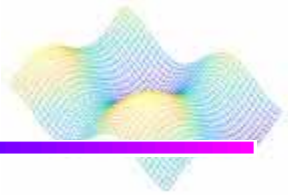
・アリスのもう一つの電子

$$a|0\rangle+b|1\rangle$$

・アリスは、確率振幅aとbが何であるかを知らない。

・しかし、アリスとボブは、ボブの電子を状態$a|0+b|1\rangle$に変えたい

量子テレポーテーション



アリスの電子の状態をボブの電子にテレポートしたい

そのためには、アリスはボブに二つの古典ビットを送る必要がある。

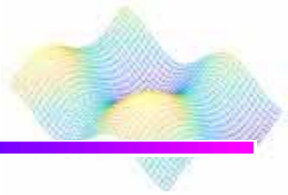
しかし、二人ともそれが何であるかを正確に知ることはできない。

そのために測定を行わなければならない。

ボブの測定

まず、ボブが測定を行うと電子の状態が0か1のどちらかになってしまい、$a|0\rangle+b|1\rangle$の状態にならない。

アリスの測定



最初のステップ

・彼女が制御する二つの量子ビットをCNOTゲートに通す。

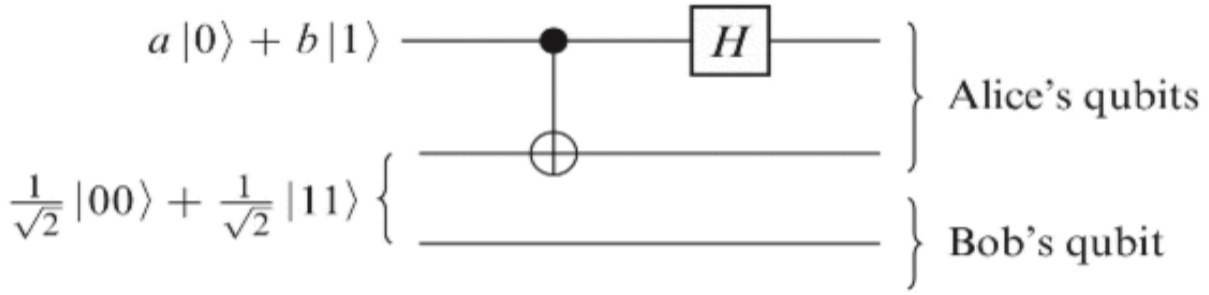
2番目のステップ

・一番上の量子ビットにハダマードゲートを適用すること

それによって、実際にアリスは、自分がコントロールする
二つの量子ビットを逆ベル回路に通すことになる。

量子テレポーテーション

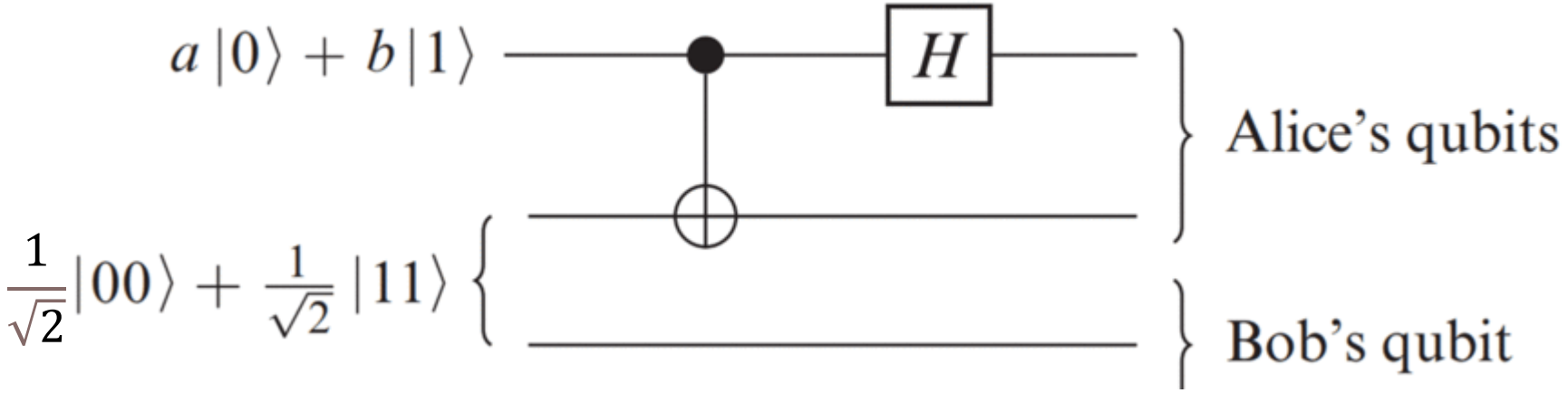
その結果下の図のように
アリスの量子ビットがボブの量子ビットの上に表示されています。



p.133~134 3行目

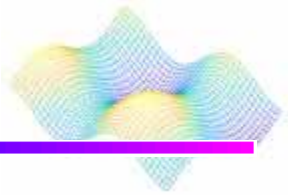
量子ビットの初期状態

アリスとボブの量子ビットの状況は次のようになっている



ボブの量子ビットの上にアリスの量子ビットがあり2行目、
3行目は絡み合った量子ビットを示している

量子ビットの初期状態



アリスとボブの3つの量子ビットの初期状態を下記の式に示す

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

この式は以下のように書き換えられる

$$\frac{a}{\sqrt{2}}|000\rangle+ \frac{a}{\sqrt{2}}|011\rangle + \frac{b}{\sqrt{2}}|100\rangle + \frac{b}{\sqrt{2}}|111\rangle$$

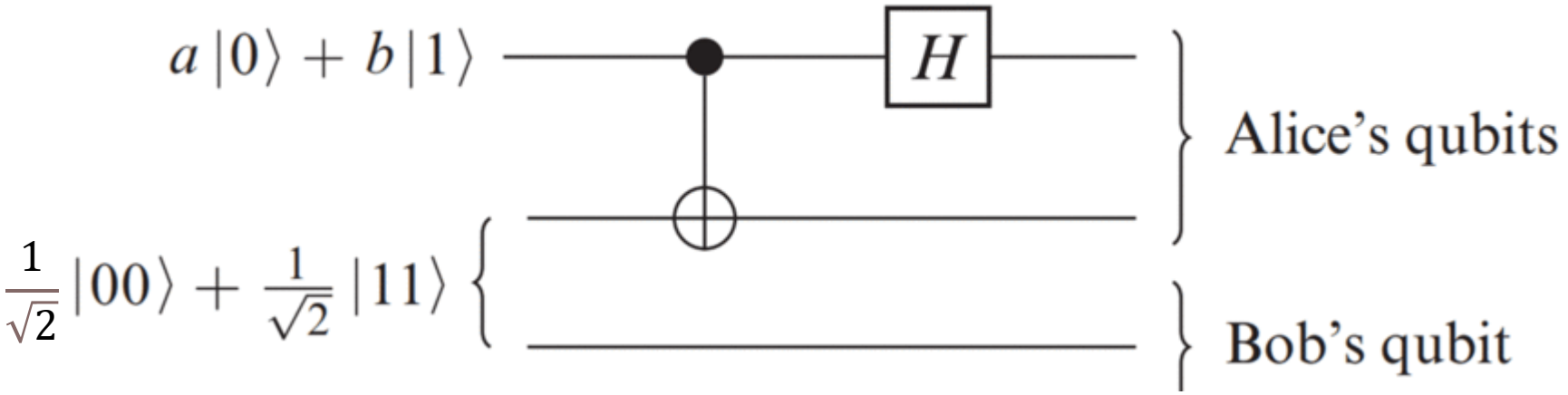
逆ベル回路の適用

次にアリスは自身の量子ビットに作用するので
これを強調して式を変形する

$$\frac{a}{\sqrt{2}}|00\rangle \otimes |0\rangle + \frac{a}{\sqrt{2}}|01\rangle \otimes |1\rangle$$
$$+ \frac{b}{\sqrt{2}}|10\rangle \otimes |0\rangle + \frac{\mathrm{b}}{\sqrt{2}}|11\rangle \otimes |1\rangle$$

この式にアリスは逆ベル回路を適用しようとしている

逆ベル回路



第１段階：CNOTゲートの適用
第２段階：アダマールゲートの適用

第１段階：CNOTゲートの適用

２つの量子にCNOTゲートを適用する

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

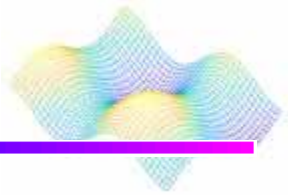
CNOT

Input	Output
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|01\rangle$
$\|10\rangle$	$\|11\rangle$
$\|11\rangle$	$\|10\rangle$

↓CNOTゲート適用

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

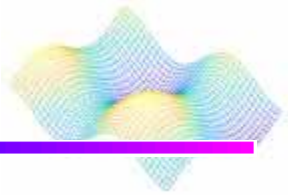
第2段階：アダマールゲートの適用



アダマールゲートを適用させる為に第1量子ビットを強調する

$$\frac{a}{\sqrt{2}}|0\rangle \otimes |0\rangle \otimes |0\rangle + \frac{a}{\sqrt{2}}|0\rangle \otimes |1\rangle \otimes |1\rangle$$
$$+ \frac{b}{\sqrt{2}}|1\rangle \otimes |1\rangle \otimes |0\rangle + \frac{\mathrm{b}}{\sqrt{2}}|1\rangle \otimes |0\rangle \otimes |1\rangle$$

アダマールゲートの適用



アダマールゲートを適用すると以下のような変換が行える

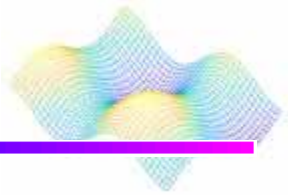
$$|0\rangle \quad \rightarrow \quad \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

$$|1\rangle \quad \rightarrow \quad \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

アダマールゲートの適用

$$\frac{a}{2}|0\rangle \otimes |0\rangle \otimes |0\rangle + \frac{a}{2}|1\rangle \otimes |0\rangle \otimes |0\rangle + \frac{a}{2}\ |0\rangle \otimes |1\rangle \otimes |1\rangle + \frac{a}{2}\ |1\rangle \otimes |1\rangle \otimes |1\rangle + \frac{b}{2}|0\rangle \otimes |1\rangle \otimes |0\rangle - \frac{b}{2}|1\rangle \otimes |1\rangle \otimes |0\rangle + \frac{\mathrm{b}}{2}|0\rangle \otimes |0\rangle \otimes |1\rangle - \frac{\mathrm{b}}{2}|1\rangle \otimes |0\rangle \otimes |1\rangle$$

簡略化



アダマールゲートを適用した式を簡略化する

$$\frac{1}{2}|00\rangle \otimes (\mathrm{a}|0\rangle + b|1\rangle) + \frac{1}{2}|01\rangle \otimes (\mathrm{a}|1\rangle + b|0\rangle)$$
$$+ \frac{1}{2}|10\rangle \otimes (\mathrm{a}|0\rangle - b|1\rangle) + \frac{1}{2}|11\rangle \otimes (\mathrm{a}|1\rangle - b|0\rangle)$$

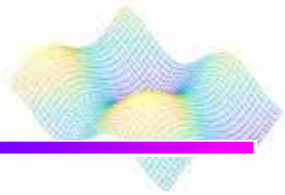
量子ゲートと回路

P134 4行～P135 下13行

2つの電子を測定

- アリスが|00〉を測定した場合、量子ビットa|0〉+b|1〉の状態になる
- |01〉 → 量子ビットa|1〉+b|0〉
- |10〉 → 量子ビットa|0〉ーb|1〉
- |11〉 → 量子ビットa|1〉ーb|0〉

ボブに知らせるために

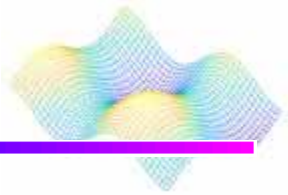


アリスとボブは$a|0\rangle+b|1\rangle$の状態になることを望んでいる。アリスはボブに4つの可能な状況のうちどれにいるかを知らせなければなりません。

そのために

測定結果に対応する00、01、10、11という2つの古典的なビット情報をボブに送り、それを知らせる。

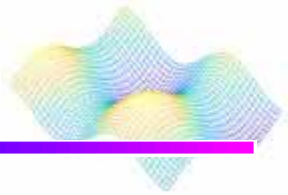
古典的なビット情報を受け取った後



- ボブが00を受け取った場合、彼は何もしません
- 01を受け取った場合、
  自分の量子ビットが$a|1\rangle+b|0\rangle$ ➡ ゲートXを適用
- 10を受け取った場合、
  自分の量子ビットが$a|0\rangle-b|1\rangle$ ➡ ゲートZを適用
- 11を受け取った場合、
  自分の量子ビットが$a|1\rangle-b|0\rangle$ ➡ ゲートYを適用

ボブの量子ビットは、アリスがテレポートしようとした量子ビットの元の状態である$a|0\rangle+b|1\rangle$の状態で終わります。

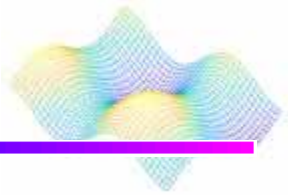
超高密度符号化と量子テレポーテーション



- 超高密度符号化では，アリスがボブに1量子ビットを送り、ビットの情報を送信します。

- 量子テレポーテーションの場合、アリスはボブに2つの古典ビットの情報を送り、1つの量子ビットをテレポーテーションします。

量子テレポーテーションとは



量子ビットを表す粒子を実際に転送することなく、ある場所から別の場所に量子ビットを転送する方法

エラーを修正するために様々な方法で使用される

電子デバイス工学特論

p.135-140

P135～

【レコードとCD】

・ レコード

表面に指紋が付かないようにレコードの端を持って扱う

帯電防止スプレーやクリーニングブラシでほこりを取り除く

・ CD

表面を引っかいても音楽が完璧に再生できる

P136～

【エラー訂正】

・ レコードはエラー訂正が組み込まれていない

→レコードを損傷すると元のサウンドを復元できない

・ CDはエラー訂正コードが組み込まれている

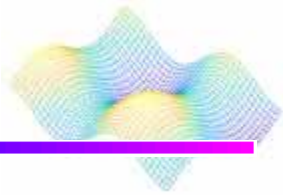
→わずかにデータが損傷しても復元できる

例　：　0とデータを送るときには、000と送る

もし、000とデータを受け取った場合、そのデータは0

もし、010と受け取った場合、3つの内一つがエラーをしたと判断して、0のデータと判断する

P137～



今のエラー訂正方法

・3つのビットの認識

⇒　3つのビットb_0、b_1、b_2の内、2つづつに注目して

①　2つが同じかどうか見る　（例：b_0＝b_1、b_1＝b_2）

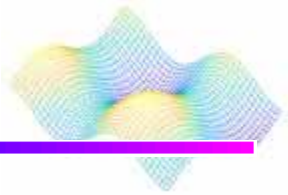
②　2つの合計を見て、他の2つ合計と同じかどうか見る

（例：b_0＋b_1＝□、b_0＋b_2＝□）

もし、全てのビットが0または1のとき　→　間違えは0である

全てが同じでないとき　→　2つが同じで、1つが違うと判断して、3つ目を0→1、または1→0に変える

P137～



そのあと、もし

$b_0 = b_1 \neq b_2$のとき → $b_0 + b_1 = 0$、$b_0 + b_2 = 1$

$b_0 = b_2 \neq b_1$のとき → $b_0 + b_1 = 1$、$b_0 + b_2 = 0$

$b_0 \neq b_1 = b_2$のとき → $b_0 + b_1 = 1$、$b_0 + b_2 = 1$

もし、$b_0 + b_1 = 0$、$b_0 + b_2 = 0$と受け取ったとき

→ そのようなことはないはずなのでエラーと判断する

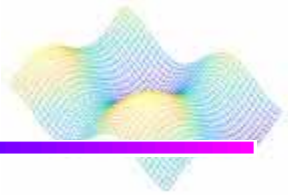
このようにして得られた2つのビットより

（0、1） → b_2をエラーとみなす

（1、0） → b_1をエラーとみなす

（1、1） → b_0をエラーとみなす

P137



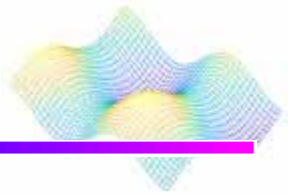
その後、エラーとみなしたビットを正しいものに置き換える事で、エラーを訂正している

例：3ビットが011、または100の時

→ 先ほどの法則により（1、1）という答えを得る

→ b_0がエラーと判断してb_0を正しいものに入れ替える

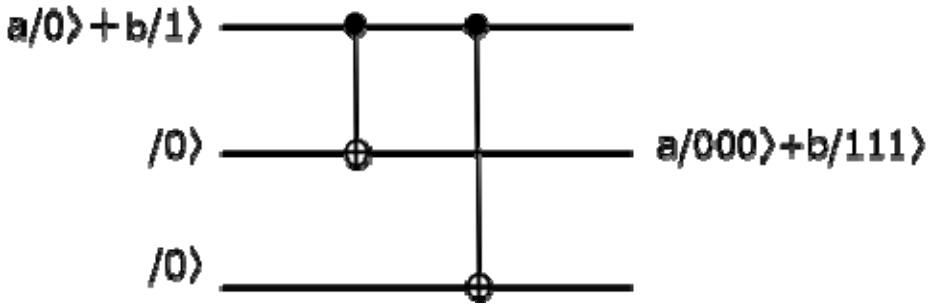
→ 011は111に、100は000に変換される



- 量子ビットフリップ補正

CNOTゲート：制御ビットが0の時、標的ビットをそのまま出力し制御ビットが1の時、標的ビットを反転させて出力させる

入力をa/0)、b/1)とするとa/000)、b/111)、と変換する



ビットを送信するときにノイズが入り、そのビットが反転して送られてしまうことがある

a/000）+b/111)→　a/000）+b/111)

a/100）+b/011)

a/010）+b/101)

a/001）+b/110)

（100）、（010）、（001）→（000）に変換可能と考えられていたが、エラーの検出と修正を両方しようとしてもつれてしまい、情報を見失う結果となりました

2021/5/25

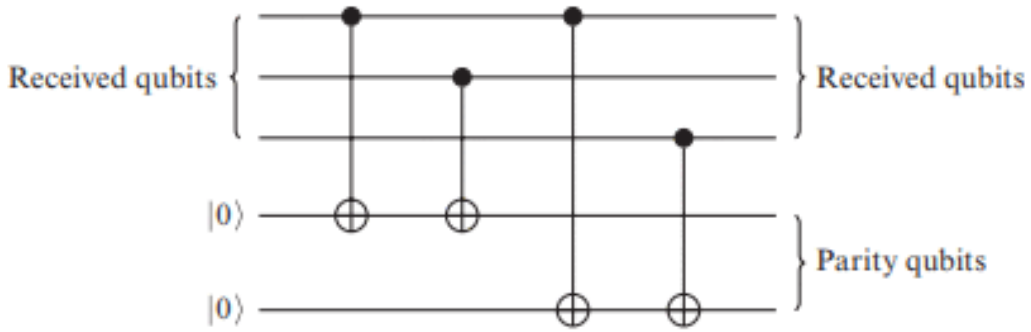
QUANTUM COMPUTING FOR EVERYYONE

P.139～142 6行目

量子ゲートと回路

ボブが $a|c_0c_1c_2\rangle + b|d_0d_1d_2\rangle$ を受け取ったとする

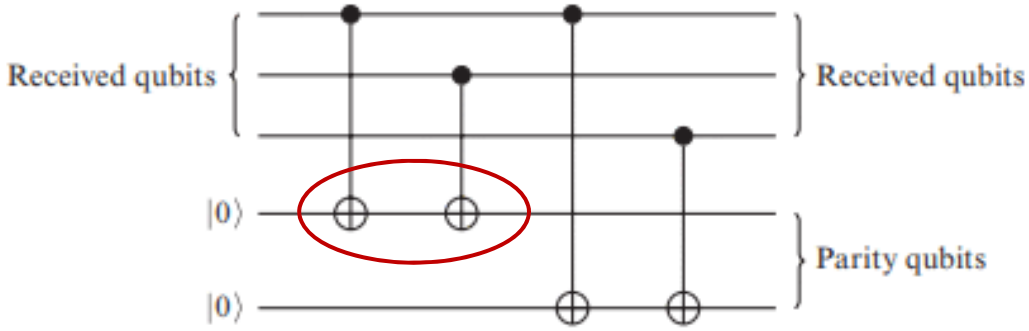
エラーが発生した場合、$c_0c_1c_2$と$d_0d_1d_2$の両方で同じ場所に発生する。
パリティチェックを行うと両方の文字列で同じ結果が得られる。



5番目を無視して考える。最初の4キュービットの入力は以下の式になる

$$(a|c_0c_1c_2\rangle + b|d_0d_1d_2\rangle)|0\rangle = a|c_0c_1c_2\rangle|0\rangle + b|d_0d_1d_2\rangle|0\rangle .$$

量子ゲートと回路



4番目のワイヤに接続されている2つのCNOTゲートは最初の2桁でパリティチェックを実行する。ただし、$c_0 \oplus c_1 = d_0 \oplus d_1$

$c_0 \oplus c_1 = d_0 \oplus d_1 = 0$. ならば、

$$a|c_0c_1c_2\rangle|1\rangle + b|d_0d_1d_2\rangle|1\rangle = (a|c_0c_1c_2\rangle + b|d_0d_1d_2\rangle)|1\rangle$$

$c_0 \oplus c_1 = d_0 \oplus d_1 = 1$. ならば、

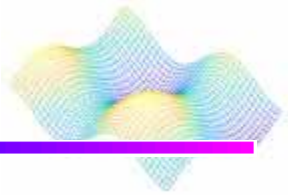
$$a|c_0c_1c_2\rangle|1\rangle + b|d_0d_1d_2\rangle|1\rangle = (a|c_0c_1c_2\rangle + b|d_0d_1d_2\rangle)|1\rangle$$

5番目のワイヤについても同様に、

$c_0 \oplus c_2 = d_0 \oplus d_2 = 0,$ ならば $|0\rangle$

$c_0 \oplus c_2 = d_0 \oplus d_2 = 0,$ ならば $|1\rangle$

量子ゲートと回路



ボブが00を取得した場合

修正するものは何もないので、彼は何もしない

ボブが01を取得した場合

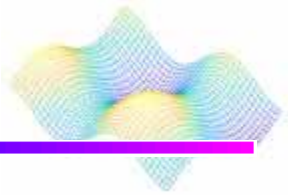
彼は3番目のワイヤーにXゲートを取り付けることにより3番目のキュービットを反転する。

ボブが10を取得した場合

彼は2番目のワイヤーにXゲートを取り付けることにより、を使用して2番目のキュービットを反転する。

ボブが11を取得した場合

彼は１番目のワイヤにXゲートをインストールすることにより、１番目のキュービットを反転する。



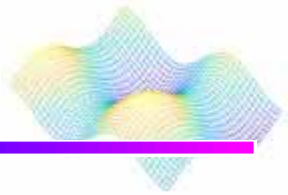
まとめ

本章では、量子ゲートと回路の概念を紹介した。ほんの数個の量子ゲートでできる驚くべきことがいくつか見られた。また、量子計算には古典的な計算がすべて含まれていることもわかった。これは、古典的な計算を実行するために量子コンピューターを使用することを意味するものではないが、量子計算がより基本的な計算形式であることを示している。

次章では、量子回路を使用して、従来の回路よりも高速に計算を実行できるかどうかに関するものである。計算の速度をどのように測定するか？量子コンピューターは常に古典的なコンピューターよりも高速か？これらは、次の章で検討する質問の一部である。

第8章　量子アルゴリズム

量子アルゴリズム



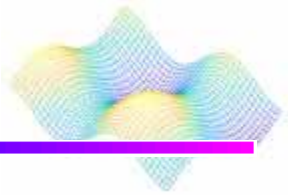
- 一般的には、量子アルゴリズムは従来のアルゴリズムと違い、全ての可能な入力を用いて量子並列処理を使用してアルゴリズムを同時に実行できるため通常のアルゴリズムよりもはるかに高速であると説明される

- 結果の測定を行うと、これらの答えの1つがランダムに得られるのではないか。その得られる答えは間違った答えになる可能性が高いのはないか。という問題が残る。

- 従って、量子アルゴリズムはこれらの問題を解決する必要があり、本章ではそれに対する3つの量子アルゴリズムを取り扱う

- 量子アルゴリズムは高速化された古典的なアルゴリズムではなく、代わりに問題を新しい観点から観るためのアイデアがある

量子アルゴリズム



- 量子アルゴリズムはブルートフォースを使用するのではなく量子の観点のみから観ることが出来る基本的なパターンを活用し独創的な方法で機能している

- 3つのアルゴリズムは基礎となる数学的パターンの活用でできている
- 数学の本であるように星で難易度を示すなら、

  Deutsch-Jozsaのアルゴリズム：星

  Simonのアルゴリズム:　　二重星　である

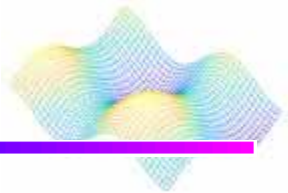
- 本章の最後で、量子アルゴリズムが古典的なアルゴリズムよりも速く解決するために問題が持たなければならない特性と、なぜそれらが難しく見えるのかについて述べる
- ただし、先ずはアルゴリズムの速度を測定する方法を説明する必要がある

2021/05/25

p.142 7行 ～ p.145 下14行

UNIVERSITY
OF TOYAMA

複雑なクラスのPとNP



下記の問題を計算機などを用いず紙とペンのみで
計算しなければいけない

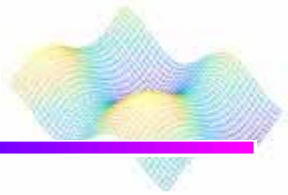
- 積が35になる、1より大きな2つの整数を求めよ。
- 1より大きな2つの整数で、その積が187に等しいものを求めよ。
- 積が2,407に等しい1より大きな2つの整数を求めよ。
- 積が88,631に等しい1より大きな2つの整数を求める。

結果を得るために時間を費やす

- 7に5をかけて、35になることを確認せよ。
- 11に17をかけて、187になることを確認せよ。
- 29に83を掛けて、2407になることを確認せよ。
- 337に263を掛けて、88,631になることを確認せよ。

格段に簡単

複雑なクラスのPとNP



入力された数字の桁数をnとする

T(n) はn桁の問題を解くための時間、または同等のステップ数を表す

nの増加によってT(n)がどのように成長するかを見る

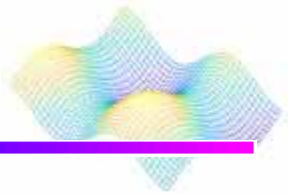
nのすべての値に対して

$T(n) \leq kn^p$となる正の数kとp

また、$T(n) > kc^n$となる正の数kと数c>1

→指数時間を必要するという

複雑なクラスのPとNP



多項式的成長<指数関数的成長

多項式時間で解ける問題は簡単
指数関数時間を必要とする問題は難しい

実際にnが増加すると

多項式時間の問題の多くは次数の小さい多項式
→現在の計算力ではできなくても
数年後には得られるようになるはず
指数関数時間→問題が非常に難しくなり、近い将来
に解決できる可能性は低くなる

複雑なクラスのPとNP

- 積が35になる、1より大きな2つの整数を求めよ。
- 1より大きな2つの整数で、その積が187に等しいものを求めよ。
- 積が2,407に等しい1より大きな2つの整数を求めよ。
- 積が88,631に等しい1より大きな2つの整数を求める。

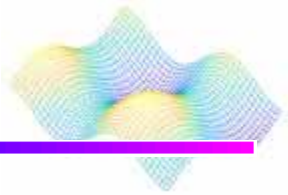
nが指数関数的なのでnの多項式ではない

- 7に5をかけて、35になることを確認せよ。
- 11に17をかけて、187になることを確認せよ。
- 29に83を掛けて、2407になることを確認せよ。
- 337に263を掛けて、88,631になることを確認せよ。

多項式時間の問題

証明を見つけていない

複雑なクラスのPとNP

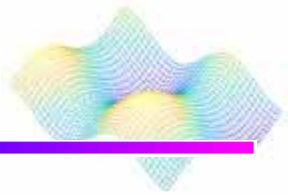


1991年 RSA研究所が2つの素数の積を因数分解する課題を出した

100桁の数字→比較的早くできた
300桁以上→未だにできない

多項式時間で解ける、つまり二つの数字の掛け合わせる問題は複雑さのクラスPに属する
反対に、二つの素数に因数分解する問題はクラスNPに属する

複雑なクラスのPとNP



Pに属する問題はNPに属するが、反対のことは言えるか？

きわめて可能性は低いが、ほとんどの人はPがNPに等しくないことを証明した人はない

NPがPに等しいかどうかという問題は計算機科学において最も重要な問題の一つ

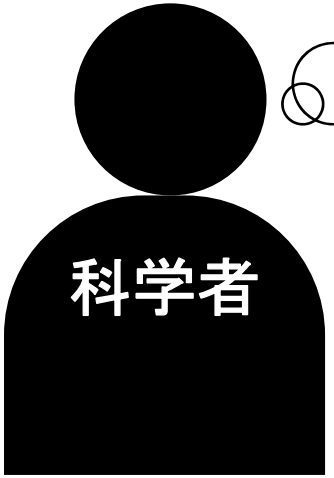
量子アルゴリズムは古典アルゴリズムより早いか？

PはNPにはならない

NPではあるがPではない

→量子コンピュータは多項式時間で解ける

古典コンピュータは説くことができない



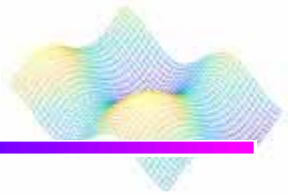
量子アルゴリズムと古典アルゴリズムの速度を比較する方法は二つある

量子アルゴリズムは古典アルゴリズムより早いか？

論理的方法
→複雑さを測定する新しい方法を発明して証明を簡単に構築すること

実用的方法
→現実世界の重要な問題を多項式時間で解くための量子アルゴリズムを構築すること
　Ex.ショアーのアルゴリズム
　　→二つの素数の積を因数分解する多項式時間で動作する量子アルゴリズム

クエリの複雑性



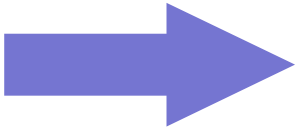
以下のアルゴリズムはすべて関数の評価に関するもの

Deutsch-Jozsaのアルゴリズム

二つのクラスに属する関数を対象
ランダムに与えられる関数がどちらに属しているかの判断が必要

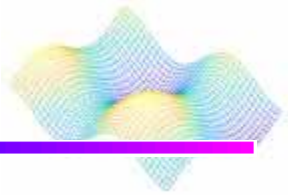
Simonのアルゴリズム

特殊なタイプの周期関数を対象
ランダムに与えられた関数の周期の決定が必要



実行には関数の評価が必要

クエリの複雑性2



クエリの複雑さ＝答えを得るために関数を評価しなければならない回数

この時の関数→ブラックボックス、オラクル

→ブラックボックスやオラクルに
問い合わせをしている（クエリしている）
ともいえる

関数をエミュレートするアルゴリズムの
書き方の心配をする必要がない！
質問の数を記録するだけ

2021/06/01

David Deutsch のアルゴリズム

David Deutsch（デイビット・ドイチュ）

・量子コンピュータの創始者の１人

・オックスフォード大学の名誉フェローの
物理学客員教授

引用元

http://www.daviddeutsch.org.uk/

ドイチュの疑問

前提条件

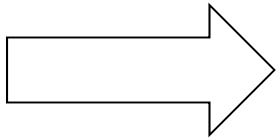
関数F_0,F_1,F_2,F_3の4つの関数がある

$F_0(0)=0$, $F_0(1)=0$

$F_1(0)=0$, $F_1(1)=1$

$F_2(0)=1$, $F_2(1)=0$

$F_3(0)=1$, $F_3(1)=1$



F_0,F_3 ・・・一定型

F_1,F_2 ・・・バランス型

Question

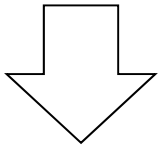
4つの関数のうち一つをランダムで与えられたとき、その関数が一定か均衡か判断する場合に何回評価しないといけないのか？

一般的な考え

実際に与えられた関数F_0～F_3に0 or 1の値を入れ評価する

例　関数$F_i(0)=0$の場合（0を代入）

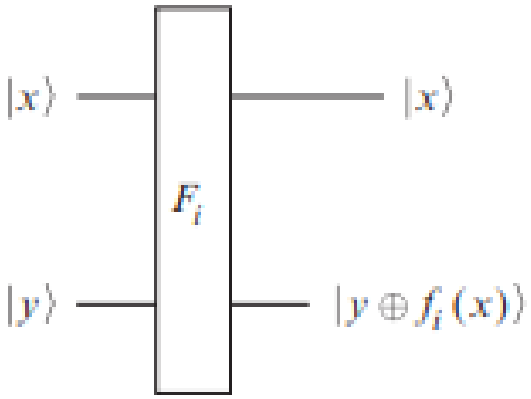
前提条件より、F_0,F_1のどちらかであることがわかる



判断するためには1を代入して再度評価する

2回評価が必要な上に
時間がかかる

量子的に考える①



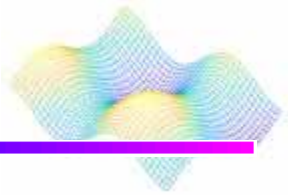
左図は4つの関数に対応したゲートである

入力	出力
$\|0\rangle \otimes \|0\rangle$	$\|0\rangle \otimes \|F_i(0)\rangle$
$\|0\rangle \otimes \|1\rangle$	$\|0\rangle \otimes \|F_i(0) \oplus 1\rangle$
$\|1\rangle \otimes \|0\rangle$	$\|1\rangle \otimes \|F_i(1)\rangle$
$\|1\rangle \otimes \|1\rangle$	$\|1\rangle \otimes \|F_i(1) \oplus 1\rangle$

⇨ 一定型

⇨ 一定型

量子的に考える②



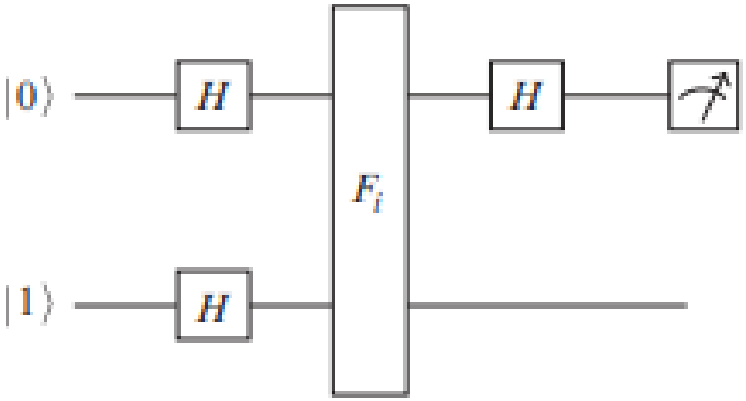
$|0\rangle \otimes |0\rangle$を例に考えたとき、

出力が$|0\rangle \otimes |F_i(0)\rangle$

これだと結局、数値を代入して
評価するのと変わらない

0と1を重ね合わせた量子ビットを
入れてみれば変化するのではないか？

重ね合わせを含む量子ビット①



左図はドイチェが考案した図

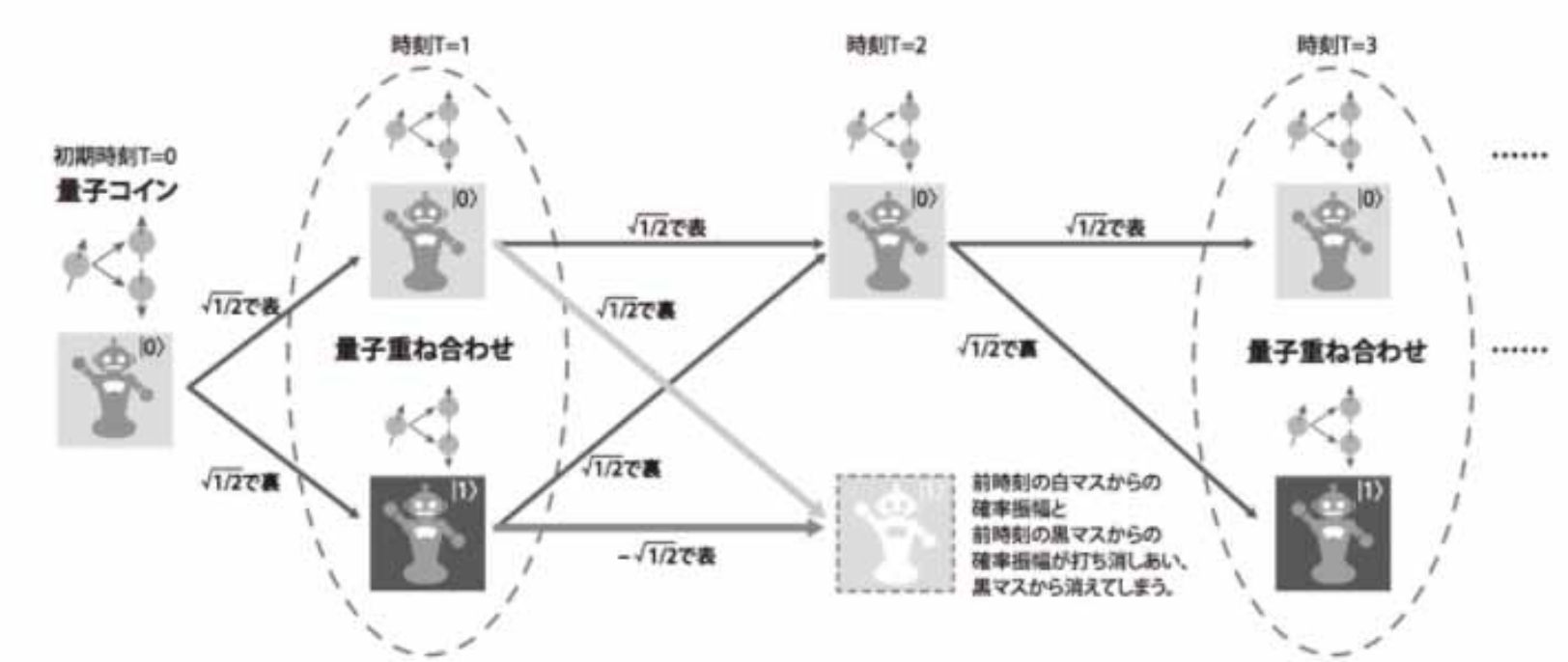
メーターのマークは計測を指している。
つまり、2ビット目の情報は考えないものとしている。

H・・・アダマールゲート
　　入力ビットである|0⟩と|1⟩が50%ずつ重なり合った状態にするゲート
　　2回ゲートを通ると元の値に戻る

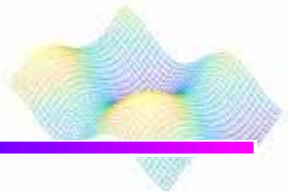
重ね合わせを含む量子ビット②

なぜHを通った後の状態が以下の式で表すことができるのか

$$|0\rangle \otimes |1\rangle \Rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$



重ね合わせを含む量子ビット③



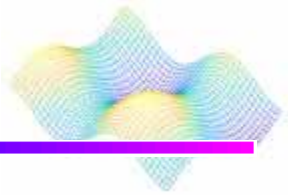
$F_0(0)=F_0(1)=0$	$\frac{1}{\sqrt{2}}(\|0\rangle+\|1\rangle)$
$F_1(0)=0,F_1(1)=1$	$\frac{1}{\sqrt{2}}(\|0\rangle-\|1\rangle)$
$F_2(0)=1,F_2(1)=0$	$-\frac{1}{\sqrt{2}}(\|0\rangle-\|1\rangle)$
$F_3(0)=F_3(1)=1$	$-\frac{1}{\sqrt{2}}(\|0\rangle+\|1\rangle)$

1ビット目の入力に対する結果

最後に2回目のHを通ると

$$\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\rightarrow|0\rangle \qquad \frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)\rightarrow|1\rangle$$

重ね合わせを含む量子ビット④



アダマールゲート(H)を使用することで、1回の評価だけで関数の判断を可能にすることができた

⇩

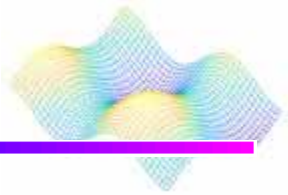
一般的なアルゴリズムよりも高速である量子アルゴリズムが確立された最初の例である

2021/06/01

The Kronecker Product of Hadamard Matrices

p.149 下4行 ～ p.152 10行

アダマール行列



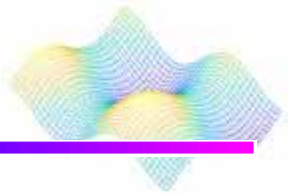
【アダマール行列とは】

・要素が1または-1のいずれかであり、
各行が互いに直交であるような正方行列

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

クロネッカー積



【クロネッカー積とは】

・任意サイズの行列の間に定義される二項演算で、
その結果は区分行列として与えられる。

（例） $A=\begin{bmatrix}1 & 3\\ 2 & 4\end{bmatrix}$ $B=\begin{bmatrix}5 & 7\\ 6 & 8\end{bmatrix}$

$$A\otimes B=\left[\begin{array}{cc|cc}1\cdot5 & 3\cdot5 & 1\cdot7 & 3\cdot7\\ 2\cdot5 & 4\cdot5 & 2\cdot7 & 4\cdot7\\ \hline 1\cdot6 & 3\cdot6 & 1\cdot8 & 3\cdot8\\ 2\cdot6 & 4\cdot6 & 2\cdot8 & 4\cdot8\end{array}\right]=\begin{bmatrix}5 & 15 & 7 & 21\\ 10 & 20 & 14 & 28\\ 6 & 18 & 8 & 24\\ 12 & 24 & 16 & 32\end{bmatrix}$$

アダマール行列のクロネッカー積（n=1）

アダマール行列は次の様に表せる

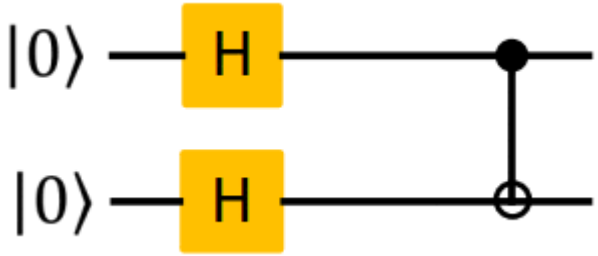
- $H=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$

アダマールゲートに状態0と1を入れると

$|0\rangle$ — H — $H(|0\rangle)=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 0\end{bmatrix}+\frac{1}{\sqrt{2}}\begin{bmatrix}0\\ 1\end{bmatrix}=\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle,$

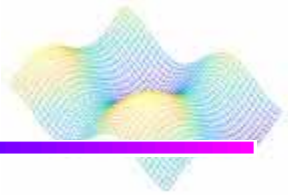
$|1\rangle$ — H — $H(|1\rangle)=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\ -1\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 0\end{bmatrix}-\frac{1}{\sqrt{2}}\begin{bmatrix}0\\ 1\end{bmatrix}=\frac{1}{\sqrt{2}}|0\rangle-\frac{1}{\sqrt{2}}|1\rangle.$

2つの量子ビットをそれぞれ入れたときの基底ベクトル



- $|0\rangle \otimes |0\rangle$の時 $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle).$
- $|0\rangle \otimes |1\rangle$の時 $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle).$
- $|1\rangle \otimes |0\rangle$の時 $\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle).$
- $|1\rangle \otimes |1\rangle$の時 $\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle).$

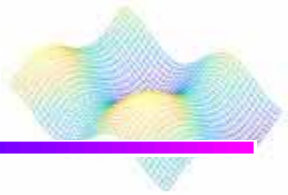
アダマール行列のクロネッカー積(n=2)



4つの結果を4次元の行列に置き換えると

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

アダマール行列のクロネッカー積(n=3)



同様にして、$H^{\otimes 3}$を$H^{\otimes 2}$を用いて求めると

$$H^{\otimes 3} = \frac{1}{\sqrt{2}}\begin{bmatrix} H^{\otimes 2} & H^{\otimes 2} \\ H^{\otimes 2} & -H^{\otimes 2} \end{bmatrix} = \frac{1}{2\sqrt{2}}\begin{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} & -\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \end{bmatrix}$$

これらの結果からnを用いて*H*を表すと

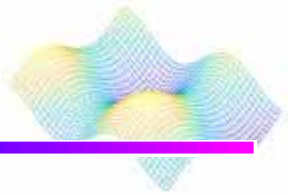
$$H^{\otimes n} = \frac{1}{\sqrt{2}}\begin{bmatrix} H^{\otimes(n-1)} & H^{\otimes(n-1)} \\ H^{\otimes(n-1)} & -H^{\otimes(n-1)} \end{bmatrix}.$$

2021/06/01

Deutsch-Jozsa Algorithm

p152 L11 ~ p155 L5

入力と関数評価



関数の種類

・定数型:　入力の値にかかわらず、
出力がすべて0またはすべて1を返す関数

・バランス型: n個の入力を入れると、
$\frac{n}{2}$個の0と$\frac{n}{2}$個の1を出力で返す関数

例）入力数 n＝3

$2^3 = 8$パターン

:(0,0,0),(0,0,1),(0,1,0),(0,1,1),(1,0,0),(1,0,1),(1,1,0),(1,1,1)

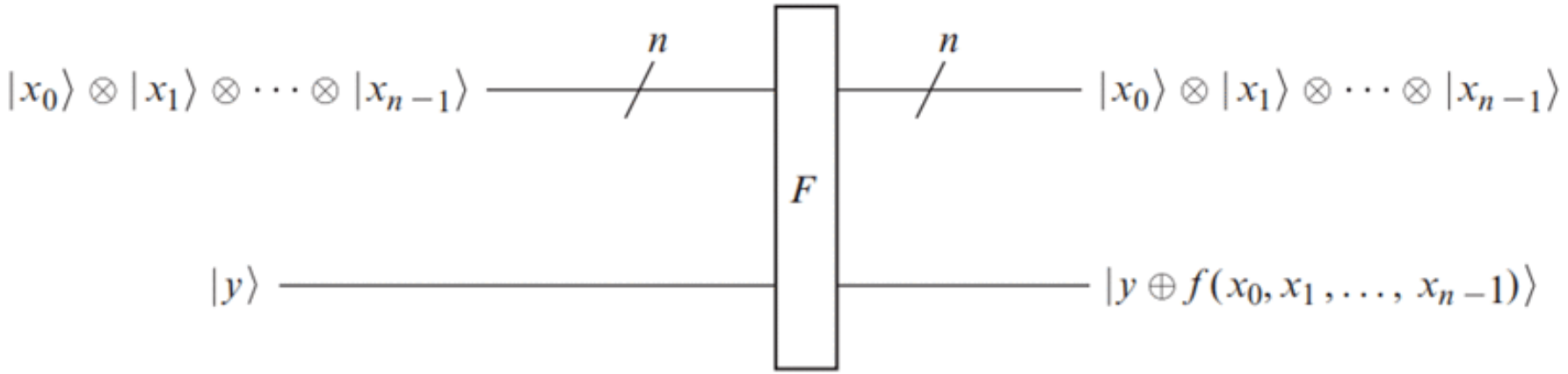
出力	1	1	1	1	?		

最大で5回の評価が必要になる

Deutsch-Jozsa algorithm

n個のブール型入力を持つ回路

・下図の入力は $|x_0\rangle \otimes |x_1\rangle \otimes \ldots \otimes |x_{n-1}\rangle$ の n個と $|y\rangle$ の
n + 1個のケットがある
・ n/ は n個の入力が並列に存在していることを表す

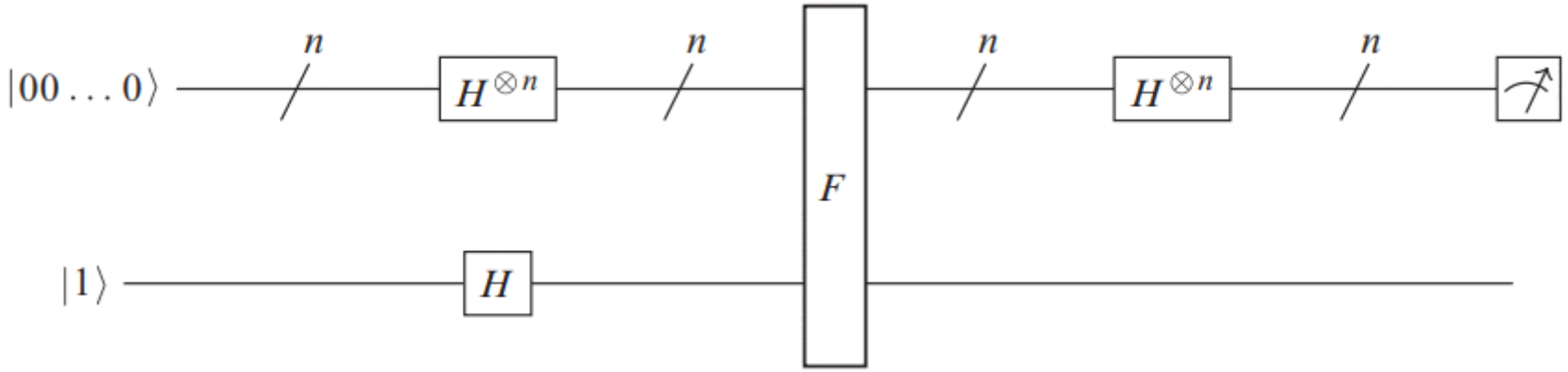


N個の入力と出力の関係

Deutsch-Jozsa algorithm

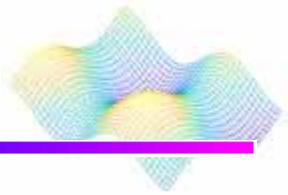
Deutsch-Jozsa algorithm

- Deutsch algorithmの回路が複数の入力に対応したもの
- 入力のケット数と出力のケット数は同数
- 入力数nがいくつでもすべてのステップは同様に動作する



n個入力の Deutsch-Jozsa algorithm

step1.



【n=2の時】

入力：|00〉

$$H^{\otimes 2}(|00\rangle) = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

Hadamard gate:入力の可能性を重ね合わせる

➡ 可能性のある状態すべてが重ね合わさっている

step1.

y=1の時 $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$

3つの入力量子ビットは

$$\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)$$

書き換えると

$$\frac{1}{2\sqrt{2}}|00\rangle \otimes (|0\rangle - |1\rangle)$$
$$+\frac{1}{2\sqrt{2}}|01\rangle \otimes (|0\rangle - |1\rangle)$$
$$+\frac{1}{2\sqrt{2}}|10\rangle \otimes (|0\rangle - |1\rangle)$$
$$+\frac{1}{2\sqrt{2}}|11\rangle \otimes (|0\rangle - |1\rangle)$$

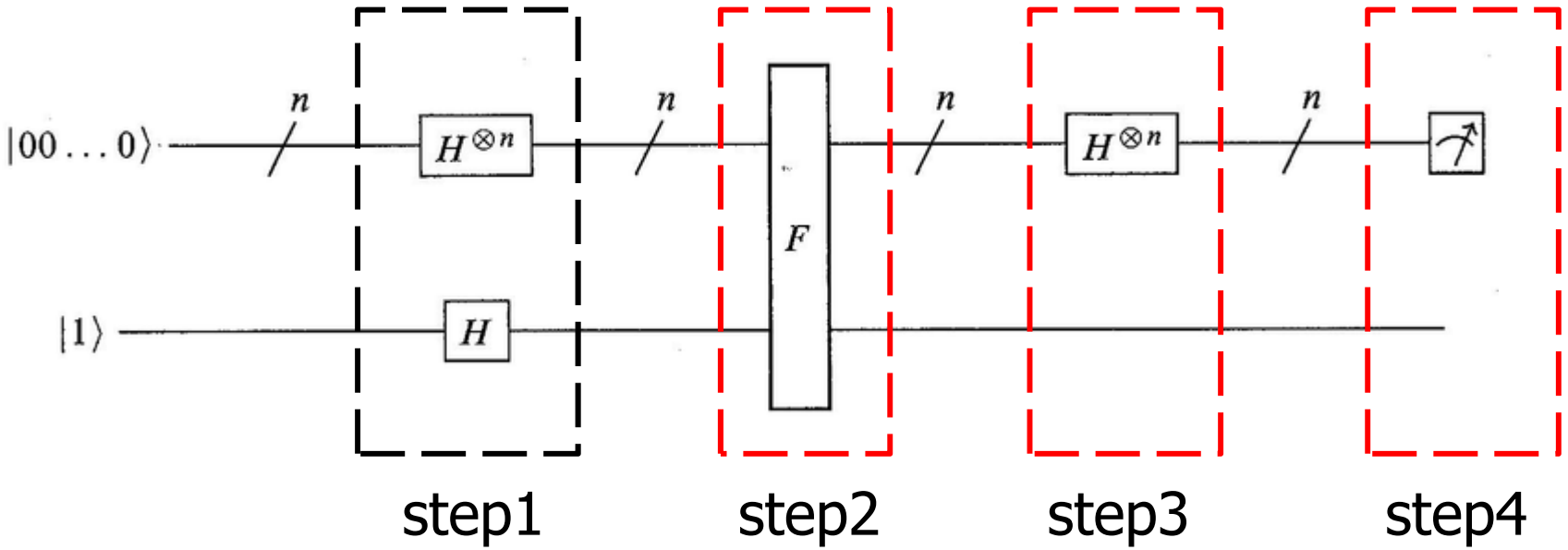
2021.6.1

ドイッチュ・ジョサのアルゴリズム（ステップ2−）

サイモンのアルゴリズム

(p155.L6-p157.L16)

ドイッチュ・ジョサのアルゴリズム



ドイッチュ・ジョサのアルゴリズム（ステップ2） p155

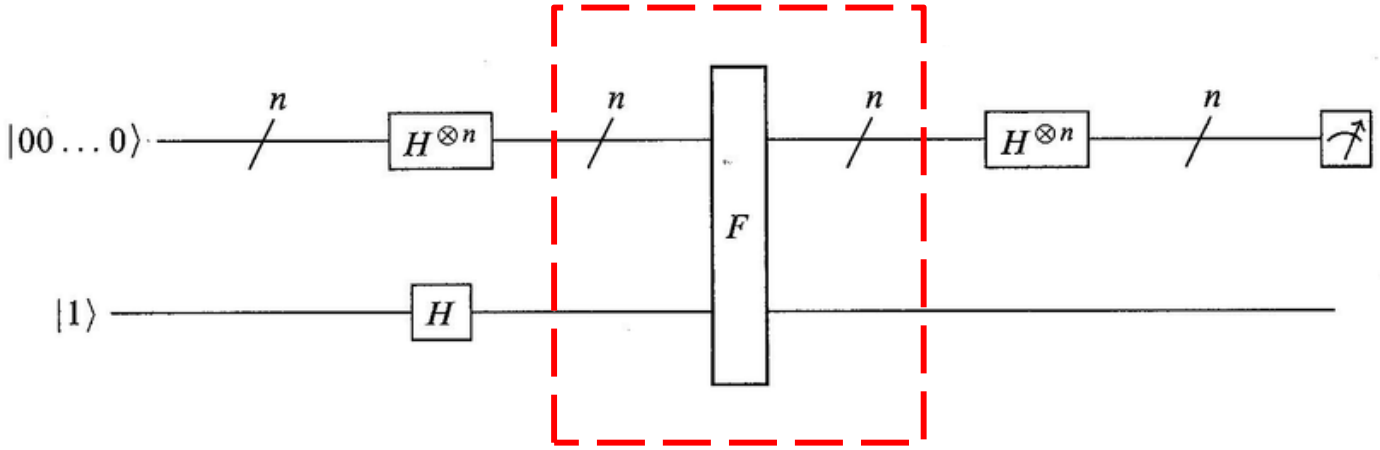
ステップ2：量子ビットがFゲートを通過する

通過前

$$\frac{1}{2\sqrt{2}}|00\rangle\otimes(|0\rangle-|1\rangle)$$
$$+\frac{1}{2\sqrt{2}}|01\rangle\otimes(|0\rangle-|1\rangle)$$
$$+\frac{1}{2\sqrt{2}}|10\rangle\otimes(|0\rangle-|1\rangle)$$
$$+\frac{1}{2\sqrt{2}}|11\rangle\otimes(|0\rangle-|1\rangle)$$

通過後

$$\frac{1}{2\sqrt{2}}|00\rangle\otimes(|f(0,0)\rangle-|f(0,0)\oplus 1\rangle)$$
$$+\frac{1}{2\sqrt{2}}|01\rangle\otimes(|f(0,1)\rangle-|f(0,1)\oplus 1\rangle)$$
$$+\frac{1}{2\sqrt{2}}|10\rangle\otimes(|f(1,0)\rangle-|f(1,0)\oplus 1\rangle)$$
$$+\frac{1}{2\sqrt{2}}|11\rangle\otimes(|f(1,1)\rangle-|f(1,1)\oplus 1\rangle)$$



ドイッチュ・ジョサのアルゴリズム（ステップ2） p155

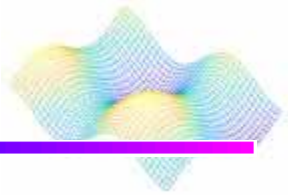
$$\frac{1}{2\sqrt{2}}|00\rangle\otimes(|f(0,0)\rangle-|f(0,0)\oplus 1\rangle)$$
$$+\frac{1}{2\sqrt{2}}|01\rangle\otimes(|f(0,1)\rangle-|f(0,1)\oplus 1\rangle)$$
$$+\frac{1}{2\sqrt{2}}|10\rangle\otimes(|f(1,0)\rangle-|f(1,0)\oplus 1\rangle)$$
$$+\frac{1}{2\sqrt{2}}|11\rangle\otimes(|f(1,1)\rangle-|f(1,1)\oplus 1\rangle)$$

$|a\rangle-|a\oplus 1\rangle=(-1)^a(|0\rangle-|1\rangle)$ を用いて変形

$$(-1)^{f(0,0)}\frac{1}{2}|00\rangle\otimes\frac{1}{2}(|0\rangle-|1\rangle)$$
$$+(-1)^{f(0,1)}\frac{1}{2}|01\rangle\otimes\frac{1}{2}(|0\rangle-|1\rangle)$$
$$+(-1)^{f(1,0)}\frac{1}{2}|10\rangle\otimes\frac{1}{2}(|0\rangle-|1\rangle)$$
$$+(-1)^{f(1,1)}\frac{1}{2}|11\rangle\otimes\frac{1}{2}(|0\rangle-|1\rangle)$$

ステップ1同様、
上下の量子ビットが絡まっていない

ドイッチュ・ジョサのアルゴリズム（ステップ2） p155



上の量子ビットに注目

$$\frac{1}{2}\left((-1)^{f(0,0)}|00\rangle + (-1)^{f(0,1)}|01\rangle + (-1)^{f(1,0)}|10\rangle + (-1)^{f(1,1)}|11\rangle\right)$$

一般的には各ケットに$\left(\frac{1}{\sqrt{2}}\right)^{n}(-1)^{f(x_0,x_1,\ldots,x_{n-1})}$を乗算すると得られる

ドイッチュ・ジョサのアルゴリズム（ステップ3） p156

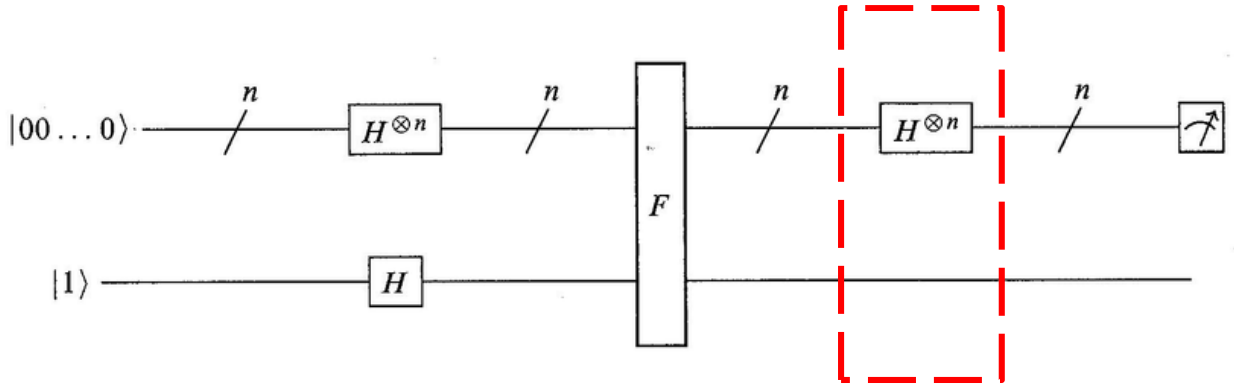
ステップ3：上の量子ビットがアダマールゲートを通過する

1.列ベクトルに変換する

$$\frac{1}{2}\left((-1)^{f(0,0)}|00\rangle + (-1)^{f(0,1)}|01\rangle + (-1)^{f(1,0)}|10\rangle + (-1)^{f(1,1)}|11\rangle\right) \Rightarrow \frac{1}{2}\begin{bmatrix}(-1)^{f(0,0)}\\(-1)^{f(0,1)}\\(-1)^{f(1,0)}\\(-1)^{f(1,1)}\end{bmatrix}$$

2.アダマール関数を乗算

$$\frac{1}{4}\begin{bmatrix}1 & 1 & 1 & 1\\1 & -1 & 1 & -1\\1 & 1 & -1 & -1\\1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}(-1)^{f(0,0)}\\(-1)^{f(0,1)}\\(-1)^{f(1,0)}\\(-1)^{f(1,1)}\end{bmatrix}$$



ドイッチュ・ジョサのアルゴリズム（ステップ3） p156

ただし、結果の行ベクトルの一番上だけを計算

$$\frac{1}{4}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}(-1)^{f(0,0)}\\ (-1)^{f(0,1)}\\ (-1)^{f(1,0)}\\ (-1)^{f(1,1)}\end{bmatrix}$$

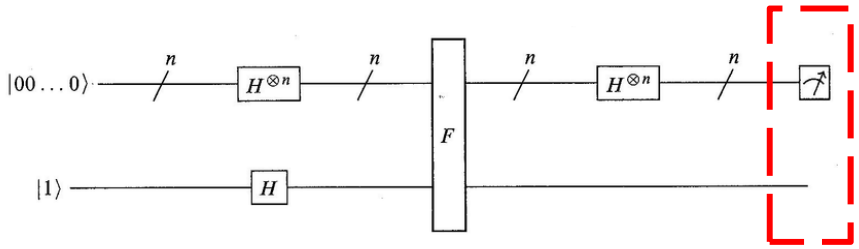
$$\Rightarrow \quad \frac{1}{4}\left((-1)^{f(0,0)}+(-1)^{f(0,1)}+(-1)^{f(1,0)}+(-1)^{f(1,1)}\right)$$

$|00\rangle$の確率振幅

- もし関数 fが一定型で、すべてを0に出力する場合、確率振幅は1
- もし関数 fが一定型で、すべてを1に出力する場合、確率振幅は-1
- もし関数 fがバランス型で、半分に0、もう半分に1を出力する場合、確率振幅は0

ドイッチュ・ジョサのアルゴリズム（ステップ4） p156

ステップ4：上の量子ビットを測定



00,01,10,11のいずれかが得られる

問題は「00を得るかどうか」

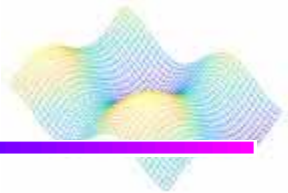
測定結果が00であった場合、関数は定数型
測定結果が00でない場合、関数はバランス型

一般的には*n*を用いて確率振幅を次のように示す （いままでの例は*n*=2の時）

$$\frac{1}{2^n}\left((-1)^{f(0,0,\ldots,0)} + (-1)^{f(0,0,\ldots,1)} + \cdots + (-1)^{f(1,1,\ldots,1)}\right)$$

n=2と同様、
すべての測定で0が得られた場合、関数は定数
少なくとも1つが1の場合、関数は均等

ドイッチュ・ジョサのアルゴリズム（ステップ4） p157



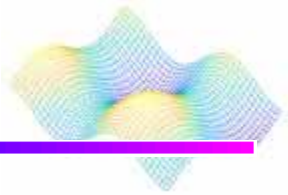
オラクルに1回質問するだけで任意の値nのドイッチュ・ジョサの問題（関数が一定か否か）を解くことができる

古典では最悪 $2^{n-1}+1$ 回の質問が必要

(n=2なら3回,n=3なら5回,n=8なら129回)

➡ 劇的に改善

サイモンのアルゴリズム p157



ドイッチュアルゴリズム
ドイッチュ・ジョサアルゴリズム

➡ **1回の質問で確実に最終的な答えが得られる珍しいケース**

ほとんどの量子アルゴリズムは
量子アルゴリズムと古典アルゴリズムの組み合わせ

・量子回路を複数使用
・確率を含む

サイモンのアルゴリズム

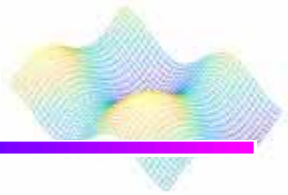
これらすべての要素を含む

ただし、その前に二進数を追加する新しい方法の導入が必要がある
（p157.L16~）

2021年　6月8日　火曜日

P157L17～P160L5

文字列のビットごとの加算法



⊕: XOR

0⊕0=0　0⊕1=1 1⊕0=1 1⊕1=0　であることから
この定義を同じ長さの2進数文字列の加算に拡張すると

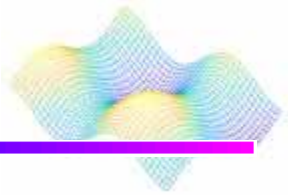
$a_0a_1...a_n + b_0b_1...b_n = c_0c_1...c_n$

$c_0 = a_0+b_0$, $c_1 = a_1 + b_1$, ... , $c_n = a_n+b_n$

これは一見すると2進数の足し算だが、キャリーを無視している
ビット単位の加算の具体的な例を示す

```
    1101
⊕   0111
-------
    1010
```

Simonの問題の記述



長さnの2進文字列を同じ長さの2進文字列に送る関数f

y = x, y = x+s のときのみ f(x) = f(y)となる2進文字列sが存在

ただし、sは0のみで構成される文字列ではない

→異なる入力文字列のペアが同じ出力文字列になるようにするため

問題は文字列sを決定すること

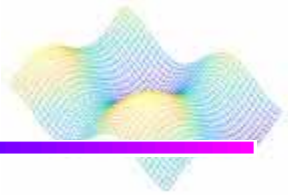
n = 3とすると、fは長さ3の2進文字列を受け取り、長さ3の別の2進文字列を与えることとし、s = 110とする

このとき

000⊕110=110 001⊕110=011 010⊕110=100 011⊕110=101

100⊕110=010 101⊕110=011 110⊕110=000 111⊕110=001

Simonの問題の記述

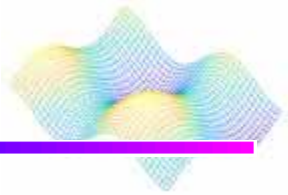


fやsは不明で、sを決めるためにいくつ関数を評価すれば
よいのかも不明
→繰り返し答えが得られるまで関数fを文字列で評価し続ける
同じ出力を与える2つの入力文字列が見つかればsを計算可能

例： f(011) = f(101)のとき
011 + $s_0s_1s_2$ = 101
また、
011 + 011 = 000
このことから、両辺の左に011をビット単位で加えると
$s_0s_1s_2$ = 011 + 101 = 110

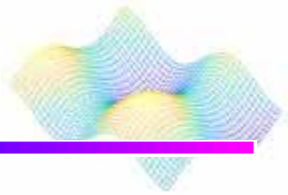
古典的アルゴリズムを用いる場合
8つのバイナリ文字列→4つを評価して4つの異なる答えを
得られるが、5つ目の評価では必ずペアを得られる

Simonの問題の記述



一般に、長さnの文字列の場合これらは2^n個の2値文字列であり、最悪の場合、繰り返しを得るには$2^{n-1}+1$回の関数評価が必要

ドット積とアダマール行列



同じ長さの2つの2進文字列、$a = a_0a_1...a_{n-1}$と$b = b_0b_1...b_{n-1}$が与えられたとき、ドット積を次のように定義する

$a \cdot b = a_0 \times b_0 + a_1 \times b_1 + ... + a_{n-1} \times b_{n-1}$

例： $a = 101, b = 111$ のとき、$a \cdot b = 1 \oplus 0 \oplus 1 = 0$

この操作は数列の対応する項を掛け合わせ、足し算をして、最後にその和が奇数か偶数かを判別している

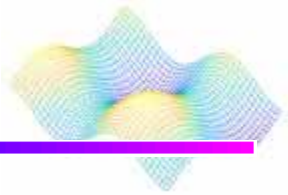
コンピュータサイエンスでは0から数え始める事が多いため、

1~4ではなく0~3を数える

2進法もよく使われる

0, 1, 2, 3を2進法で表すと、00, 01, 10, 11 となる

ドット積とアダマール行列



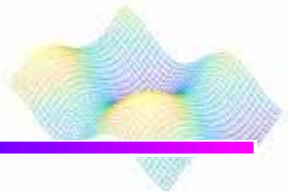
4×4の行列に対して、00~11を行と列に両方にラベル付けする。

また、この行列における項目の位置は、その項目が現れる行と列の両方を記載することで与えられ、i行j列目の項目をi･jとすると、次のような行列が得られる

$$\begin{array}{c} \\ 00 \\ 01 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

この行列を$H^{\otimes 2}$と比較してみる

ドット積とアダマール行列



点積行列の1の部分は、$H^{\otimes 2}$の負の部分と全く同じ位置にあることに注意する

$(-1)^0 = 1$、$(-1)^1 = -1$という事実を使って、次のように書ける

$$H^{\otimes 2} = \frac{1}{2}\begin{bmatrix} (-1)^{00\,00} & (-1)^{00\,01} & (-1)^{00\,10} & (-1)^{00\,11} \\ (-1)^{01\,00} & (-1)^{01\,01} & (-1)^{01\,10} & (-1)^{01\,10} \\ (-1)^{10\,00} & (-1)^{10\,01} & (-1)^{10\,10} & (-1)^{10\,11} \\ (-1)^{11\,00} & (-1)^{11\,01} & (-1)^{11\,10} & (-1)^{11\,11} \end{bmatrix}$$

この方法は、一般的には、正負の値がどこにあるかを見つける方法で、例えば、$H^{\otimes 3}$の101番の行と111番の列にある項目が欲しい場合は、ドット積を計算して0を求める
これは、その項目が正であることを示している

2021 / 06 / 08　　P160 - P167

1. アダマール行列とサイモンの問題

(Hadamard Matrices and Simon's Problem)

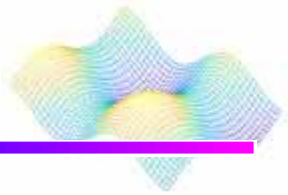
2. サイモンの問題による量子回路

(The Quantum Circuit for Simon's Problem)

3. サイモンの問題の古典的解法

(The Classical Part of Simon's Algorithm)

サイモンの問題



1対1対応関数　･･･　全ての出力値に対して1つの入力値が対応する

EX.　$f(1) \to 1$　$f(2) \to 2$　$f(3) \to 3$　$f(4) \to 4$

2対1対応関数　･･･　全ての出力値に対して必ず2つの入力値が対応する

EX.　$f(1) \to 1$　$f(2) \to 2$　$f(3) \to 1$　$f(4) \to 2$

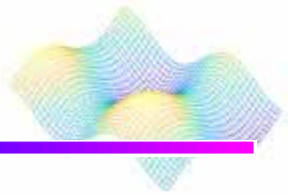
この2対1対応は秘密ビット s を用いる

入力　：x_1, x_2　に対し

$f(x_1) = f(x_2)$ が成り立つとき

$x_1 \oplus x_2 = s$ を必ず満たす

アダマール行列とサイモンの問題



アダマール行列のクロネッカー積を求める方法を用いて
サイモンの問題で与えられた秘密の文字列とペアになる
列を足すとどうなるか？

ここで例として$H^{\otimes 2}$を用い

秘密の文字列　s　を 10 とする

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

ラベル

	00	01	10	11
00	*	*	*	*
01	*	*	*	*
10	*	*	*	*
11	*	*	*	*

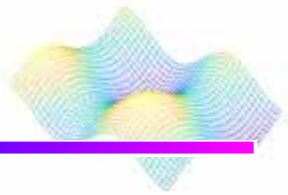
この時のペアは

$00 \oplus 10 = 10$　$01 \oplus 10 = 11$　$10 \oplus 10 = 00$　$11 \oplus 10 = 01$

であるから

$f(00) = f(10)$　　$f(01) = f(11)$

アダマール行列とサイモンの問題



ラベルの付け方

$$\underset{b}{00} \oplus \underset{s}{10} = \underset{b \oplus s}{10} \qquad \underset{b}{f(00)} = \underset{b \oplus s}{f(10)}$$

秘密の文字列 s とペアになる列同士を足すと…

列11と列01のペア

$$\underset{b}{\frac{1}{2}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix}} + \underset{b \oplus s}{\frac{1}{2}\begin{bmatrix}1\\-1\\-1\\1\end{bmatrix}} = \frac{1}{2}\begin{bmatrix}2\\-2\\0\\0\end{bmatrix}$$

列10と列00のペア

$$\underset{b}{\frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}} + \underset{b \oplus s}{\frac{1}{2}\begin{bmatrix}1\\1\\-1\\-1\end{bmatrix}} = \frac{1}{2}\begin{bmatrix}2\\2\\0\\0\end{bmatrix}$$

アダマール行列とサイモンの問題

積とビットごとの足し算は通常の指数法則に従う

$$(-1)^{a\cdot(b\oplus s)} = (-1)^{a\cdot b}\,(-1)^{a\cdot s}$$

秘密の文字列 s とペアになる列同士の足し算は
a行のエントリに対して 以下の性質を持つ

$$(-1)^{a\cdot(b\oplus s)} + (-1)^{a\cdot b} = \pm 2 \quad \text{if} \quad a\cdot s = 0$$

$$(-1)^{a\cdot(b\oplus s)} + (-1)^{a\cdot b} = 0 \quad \text{if} \quad a\cdot s = 1$$

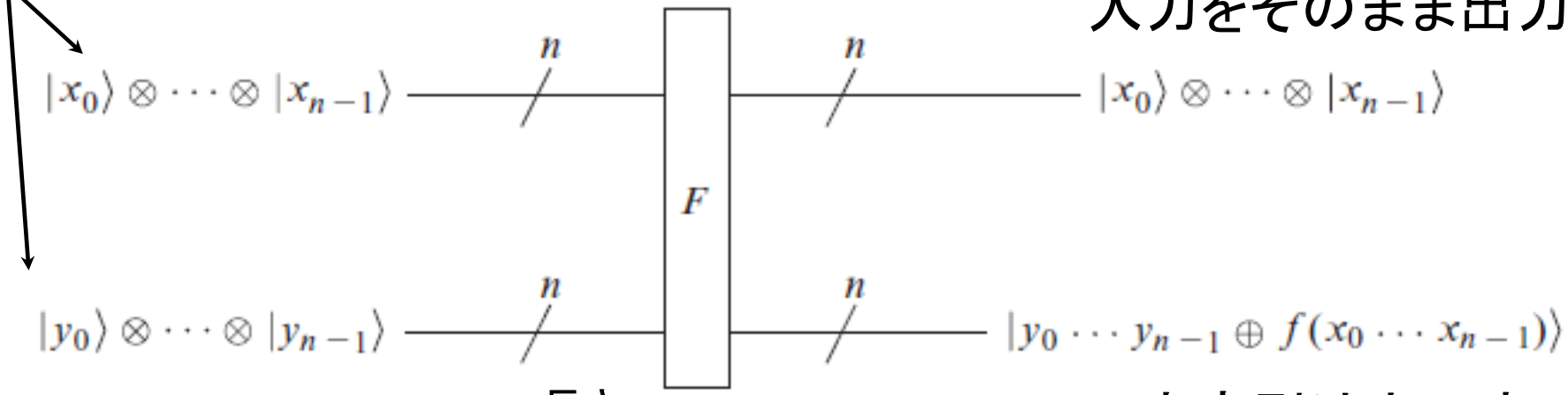
この性質を用いたものがサイモン
の問題による量子回路である

サイモンの問題による量子回路

ここで用いる F ゲート とは...

同じ長さの2つ
の文字列を入力

上の文字列は
入力をそのまま出力



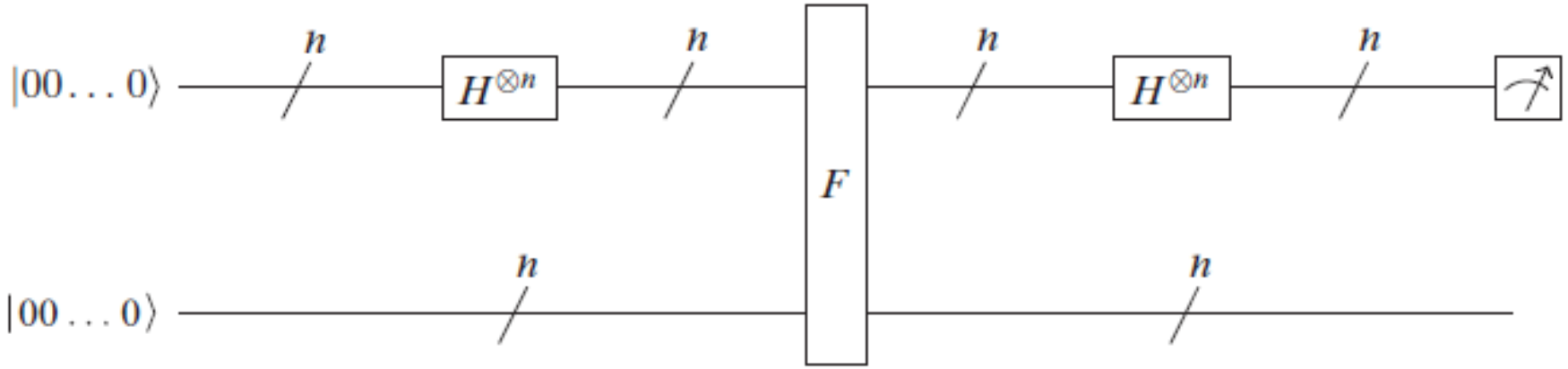
n：長さ

下の文字列は上の文
字列で評価された関数
を下の入力した文字列
と足したものを出力

Fゲート
(一連の流れを出力するゲートとする)

サイモンの問題による量子回路

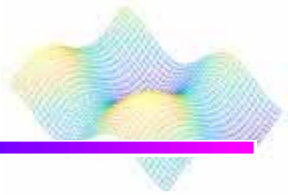
サイモンのアルゴリズムを表す量子回路



上の量子ビット：アダマールゲート ⇒ Fゲート
⇒ アダマールゲート ⇒ 出力

下の量子ビット：Fゲート⇒出力

サイモンの問題による量子回路



n = 2 のとき

上2つの量子ビットは最初 $|00\rangle$ の状態で
アダマールゲートを通過すると以下となり

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

下2つの量子ビットは $|00\rangle$ のままなので
4つの量子ビットは以下である

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

この量子ビットが、Fゲートを通過すると以下になる

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

サイモンの問題による量子回路

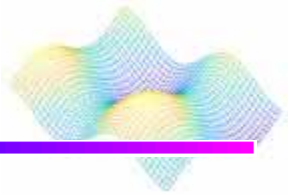
Fゲートを通過した後、上の量子ビットがアダマールゲートを通過すると右のようになる

$$\frac{1}{4}(|00\rangle+|01\rangle+|10\rangle+|11\rangle)\otimes|f(00)\rangle$$
$$+\frac{1}{4}(|00\rangle-|01\rangle+|10\rangle-|11\rangle)\otimes|f(01)\rangle$$
$$+\frac{1}{4}(|00\rangle+|01\rangle-|10\rangle-|11\rangle)\otimes|f(10)\rangle$$
$$+\frac{1}{4}(|00\rangle-|01\rangle-|10\rangle+|11\rangle)\otimes|f(11)\rangle$$

$$\frac{1}{4}|00\rangle\otimes(|f(00)\rangle+|f(01)\rangle+|f(10)\rangle+|f(11)\rangle)$$
$$+\frac{1}{4}|01\rangle\otimes(|f(00)\rangle-|f(01)\rangle+|f(10)\rangle-|f(11)\rangle)$$
$$+\frac{1}{4}|10\rangle\otimes(|f(00)\rangle+|f(01)\rangle-|f(10)\rangle-|f(11)\rangle)$$
$$+\frac{1}{4}|11\rangle\otimes(|f(00)\rangle-|f(01)\rangle-|f(10)\rangle+|f(11)\rangle)$$

項を並び変え、最初の2つの量子ビットについて解く

サイモンの問題による量子回路



ここで、s = 10 とすると

$$f(00) = f(10)$$
$$f(01) = f(11)$$

より

$$\frac{1}{4}|00\rangle \otimes (|f(00)\rangle + |f(01)\rangle + |f(00)\rangle + |f(01)\rangle)$$
$$+\frac{1}{4}|01\rangle \otimes (|f(00)\rangle - |f(01)\rangle + |f(00)\rangle - |f(01)\rangle)$$
$$+\frac{1}{4}|10\rangle \otimes (|f(00)\rangle + |f(01)\rangle - |f(00)\rangle - |f(01)\rangle)$$
$$+\frac{1}{4}|11\rangle \otimes (|f(00)\rangle - |f(01)\rangle - |f(00)\rangle + |f(01)\rangle)$$

$$\frac{1}{4}|00\rangle \otimes (2|f(00)\rangle + 2|f(01)\rangle)$$
$$+\frac{1}{4}|01\rangle \otimes (2|f(00)\rangle - 2|f(01)\rangle)$$
$$+\frac{1}{4}|10\rangle \otimes (0)$$
$$+\frac{1}{4}|11\rangle \otimes (0)$$

簡略化すると...

サイモンの問題による量子回路

結局…

$$\underbrace{\frac{1}{\sqrt{2}}|00\rangle \otimes \frac{1}{\sqrt{2}}(|f(00)\rangle + |f(01)\rangle)} + \underbrace{\frac{1}{\sqrt{2}}|01\rangle \otimes \frac{1}{\sqrt{2}}(|f(00)\rangle - |f(01)\rangle)}$$

上の2つの量子ビットを測定すると
00または01のいずれかが、それぞれ1／2の確率で得られる
（n>2でも同様に計算可能）

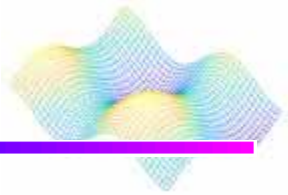
↕

秘密の文字列とのドット積が0になる文字列のいずれかが得られる
(回路を回す度に得られる文字列は変わるが条件は同じ)

ここまで来ても s がわからない…

➡ サイモンの問題の古典的な解法が登場

サイモンの問題の古典的解法

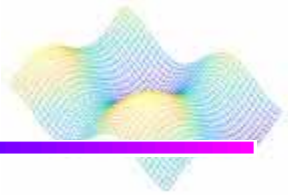


・どうやって s を見つける？ ⇒ 総当たり

・回路を何回回せばよい？ ⇒ 最大で $2^{n-1}+1$

同じ出力値に対して2つの入力値を
見つけるまで、可能な全ての入力値
の半分を確認しなければならない

サイモンの問題の古典的解法



s の見つけ方

ここで n = 5 , 秘密の文字列 s = $s_0s_1s_2s_3s_4$ とすると
s の文字列は31通り考えられる(※00000は選べない)

・一回目で回路から出力された値が 10100 とすると

$1 \times s_0 \oplus 0 \times s_1 \oplus 1 \times s_2 \oplus 0 \times s_3 \oplus 0 \times s_4 = 0$ であるから

$s_0 \oplus s_2 = 0$ より $s_0 = s_2$ とわかる

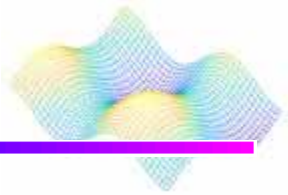
・二回目で回路から出力された値が 00100 とすると

$0 \times s_0 \oplus 0 \times s_1 \oplus 1 \times s_2 \oplus 0 \times s_3 \oplus 0 \times s_4 = 0$ より

$s_2 = 0$ とわかる

つまり、 $s_2 = s_0 = 0$

サイモンの問題の古典的解法



・三回目で回路から出力された値が 11110 とすると

$1\times 0 \oplus 1\times s_1 \oplus 1\times 0 \oplus 1\times s_3 \oplus 0\times s_4 = 0$　より

$s_1 = s_3$ とわかる

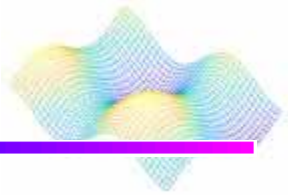
・四回目で回路から出力された値が 00111 とすると

$0\times 0 \oplus 0\times s_1 \oplus 1\times 0 \oplus 1\times s_3 \oplus 1\times s_4 = 0$　より

$s_3 = s_4$　$s_1 = s_3$　$s_1 = s_3 = s_4$　$s_1 = s_3 = s_4 = 1$　とわかる

よって s は　01011 である

サイモンの問題の古典的解法



・サイモンの問題の古典的解法では秘密文字列 s を得るために必要な回数は最大$2^{n-1}+1$である

出力はn個の未知数からなる線形独立な一次方程式となる為、これをn-1個解く

・実際の量子アルゴリズムでは、同じ出力を何度も得られる可能性がある為、最大$2^{n-1}+1$回以上実行する必要がある

次のセクションでこの回数を決めるアイディアを検討する