

入力が分かれば、演算が可逆的になるよ！
 可逆的だよ！
 可逆的だよ！
 可逆的だよ！

引き算をすれば y が分かる！
 $y = 8 - 3 = 5$
 答えは $y = 5$

y が分かると x は？
 $SUM(3, y) = (3, 8)$
 演算を逆にしたら y を
 求められるかな？



2つの入力が分かるよ、以下のようにするよ
 $SUM(7, 4) = (7, 11)$
 掛け

$SUM(x, y) = (x, x+y)$
 掛け

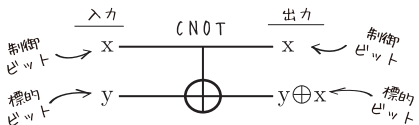
足し算の演算が入力の1つ (x) を
 出カと一緒に返したら、可逆的になるの？

可逆的な足し算？

量子演算は
 可逆でなければいけない！



だから、
 情報を失うことが
 許されないんだ。



入力		出力	
x	y	x	y
0>	0>	0>	0>
0>	1>	0>	1>
1>	0>	1>	1>
1>	1>	1>	0>



標的ビット (y)

制御ビット (x)
 変化なし！
 標的ビットが
 反転するかどうかは
 制御ビット (x) によって
 決まるよ。

CNOTを作用させて逆にする！

入力が分かれば、CNOTの
 真理値表から出力を決められるよ。

$CNOT(|0\rangle, |1\rangle) = (|0\rangle, |1\rangle)$
 入力 出力

もちろん、演算を逆にできるよね！

出力が分かれば、CNOTの真理値表から
 入力を決められるよ。

やってみよう！

まずは、入力から出力を求めよう：
 $CNOT(|1\rangle, |0\rangle) = (|1\rangle, |1\rangle)$
 入力 出力
 そして、出力から入力を求めよう：
 $CNOT(|1\rangle, |1\rangle) = (|1\rangle, |0\rangle)$
 入力 出力
 $\langle 1| \langle 0|$ の場合 $\langle 1| \langle 0|$ の場合
 $\langle 1| \langle 1|$ の場合 $\langle 0| \langle 1|$ の場合
 : 45

$? + ? = 8$

- 和が8の場合:
- 1 + 7 = 8
- 2 + 6 = 8
- 3 + 5 = 8
- 4 + 4 = 8



足し算は不可逆的

和があるだけでは、元の数を求められない。

元の値に戻った！

m = 5 のときを考えよう
 n に 2 + n + x を付けたら、m = 5 になるよ
 m から 2 + n + x を外すと、m = 5 になるよ

符号反転は可逆的

数字における演算
 一部は可逆

量子コンピューティングについて
 もっと知りたいならこちら

<https://www.epiqc.cs.uchicago.edu/resources/>

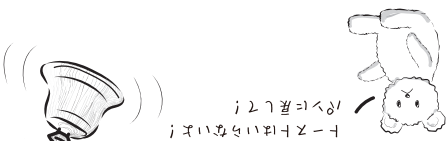
March 2019 (v2)

Translated by QCSG, Kyushu University, Japan

This work is funded in part by EPIQC,
 an NSF Expedition in Computing,
 under grant 1730449



可逆性 (元に戻せること) は
 身のまわりにはあふれているよ！



トーストはいいよ！
 トーストはいいよ！
 トーストはいいよ！

水が沸けるね



トーストはいいよ！
 トーストはいいよ！
 トーストはいいよ！



噛いたクッキーは
 元の生地に戻せない



チャックは
 開閉できるね



靴紐は結ぶと
 解けるね



水は沸けるね