

第一章 概述

量子计算的基本概念

- 1.1 量子计算和量子信息简介
- 1.2 量子比特
- 1.3 量子计算
- 1.4 量子算法
- 1.5 量子信息处理（量子计算）实验
- 1.6 量子信息

1.1 量子计算和量子信息简介

- 量子计算：利用量子系统及其特性完成信息处理任务
 - 量子计算领域的远期目标是实现实用的通用量子计算，在某些问题的计算效率上超越任何经典计算机。将量子计算应用到基础物理、计算机、化学、生物、医学、金融等领域上。
 - 基础物理：运用量子信息论的见解和技术，可以在量子引力(控制大质量物体的退相干)、量子场论和基础物理其他方面的理解方面取得重大进展。
- 量子系统的重要特性
 - 态的线性叠加 (态和物理量的数学刻画)
 - 概率性 (测量公理)
 - 信息被储存在量子态中，但是以概率幅叠加的方式出现，不容易直接存取
 - 纠缠和非定域性 (Bell test: 量子理论违背了定域实在性)
 - 对环境敏感
- 小史
 - 1930 能带结构计算首次用于预测新材料的特性
 - 1956 约翰·巴丁 (John Bardeen)、莱昂·库珀 (Leon Cooper) 和约翰·施里弗 (John Schrieffer) 低温超导的BCS理论
 - 1980 Klaus von Klitzing、Dorda 和 Pepper发现量子霍尔效应
 - 1984 Charles Bennett 和 Gilles Brassard 将量子理论应用于密码协议，并证明量子密钥分发 (QKD)可以增强信息安全
 - 1980s Richard Feynman：在量子计算机上模拟量子系统
 - 1985 Deutsch 算法 (oracle问题)
 - 1994 Shor算法，多项式时间的因数分解算法。因数分解 \in BQP
 - 1996 Grover 算法对非结构化搜索问题进行了量子加速
 - 1996 Seth Lloyd：以小时间步长演化可以有效模拟任何包含少粒子相互作用的多体量子哈密顿量；模拟所需的总时间呈多项式增长。另一个链接

可计算性(computability)简介

- 数学、计算机科学、哲学(数理逻辑)
 - 如何对数学中的算法进行抽象和严格定义？(图灵机)
 - 计算机程序能够完成的任务的限度是什么？(computable function) 有什么东西是超出限度的？(不可计算函数，停机问题等)
 - 如何根据算法的复杂性进行分类？(复杂性类)
 - 什么是证明？真命题都能够被证明吗？基于语义考虑的逻辑推论和基于形式推理考虑的定理的关系？
- 图灵机和可计算函数

定义 18.1 (图灵机 (TM))[Def.1.1]Kitaev2002cqcomputation)**DEF** 图灵机 包括:

- 有限集合 S 称为 **DEF** 字母表 (alphabet), 字母表中存在空格。
- 子集 $A \subset S$ 称为 **DEF** 外部字母表 (external), A 不包含空格。
- 有限集合 Q , 其中的元素称为图灵机的 **DEF** 状态。
- 初始状态 $q_0 \in Q$ 。
- 偏函数 $\delta: Q \times S \rightarrow Q \times S \times \{-1, 0, 1\}$ 称为 **DEF** 转移函数。

- 概率图灵机 (PTM) 两个转移函数, 每步都等概率地使用其中一个。这个概念被用于定义 NP 问题和 BPP 问题。
- 记 A^* 为 A 能产生的所有字符串的集合

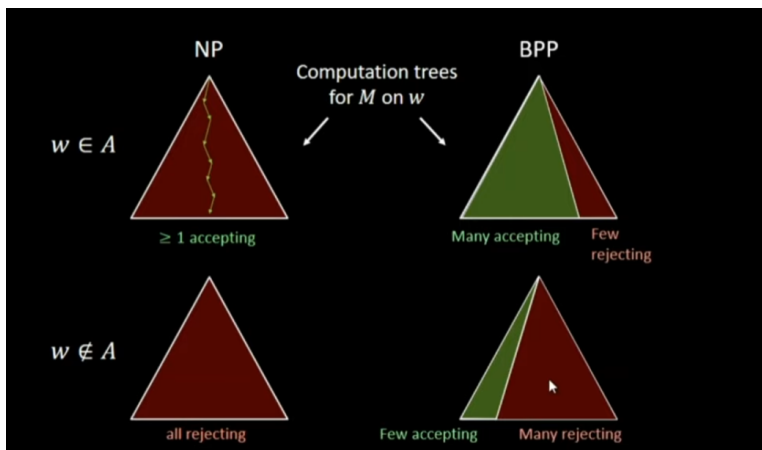
定义 18.2 (可计算函数)

- 从 A^* 到 A^* 的偏函数 f 称为 **DEF** 可计算偏函数, 如果任意取 $\alpha \in A^*$, 存在输出结果, 记为 $\varphi_M(\alpha)$, 满足 $\varphi_M(\alpha) = f(\alpha)$ 。
- 函数的数量不可数多, 图灵机可数多, 一定存在一些不可计算的函数。
- 谓词(predicative)P 从 A^* 到 {真, 假} 的函数。可计算的谓词是可判定的。
- Turing-Church Thesis
 - 数学上有严格定义的 computable partial function 正确的建立了 effectively calculable partial function 的概念。它本身并不是一个能够被证明的数学陈述。但人们可以寻找支持或反对丘奇论点的证据, 有利的证据越多, 这个论题就越合理。目前为止还没有反例。另外, 独立构建的计算模型, 比如 λ 演算和图灵机等最终都被证明是等价的。(令人惊讶的殊途同归)
 - 就像函数的连续性的 $\delta - \epsilon$ 定义一样, 尽管可能和函数连续的直观理解在某种程度上有所出入, 连续也已经被严格地数学化了, 而且该定义非常有用。

计算复杂性类的简介

计算复杂度关心一个问题的计算效率，计算效率分为时间和空间复杂度。

- P (polynomial time): 可以在确定性图灵机上在多项式时间内解决的决策问题的复杂性类。
- NP (nondeterministic polynomial time): 可以 (存在非零的概率使得正确答案是 Yes 时返回 Yes) 在图灵机 (PTM) 上在多项式时间内解决的决策问题的复杂性类, 或者等价地, 是可通过确定性图灵机在多项式时间内验证的问题的复杂性类。
- NP-hard: c 属于 NP 困难当且仅当 NP 内的所有问题 (我们能确定 NP 内都有哪些问题吗? 给定一个问题我们一定能判断它是否 NP 吗?) 都能在多项式时间内约化 (什么是约化?) 为 c 。
- NP-complete: 问题 c 属于 NP 完备类当且仅当 c 属于 NP 类且 NP 内的所有问题都能在多项式时间内约化为 c 。由于 NP 完备多了一个额外的条件, 所以显然有, NP 完备被 NP 困难包含。迄今为止, 既没有求解 NP 完全问题的多项式时间算法, 也没有人能够证明对这类问题不存在多项式时间算法。这一 $P \neq NP$ 问题自 1971 年被提出以后, 已经成为理论计算机科学领域中最深奥、最令人费解的开放问题。
- RP (randomized polynomial time): 决策问题的复杂性类别, 可以在概率图灵机上以多项式时间内用至多 (1-sided error) 来解决, 具体来说, No 总是正确返回 No, Yes 以二分之一的概率返回 Yes/No。
- ZPP (zero-error probabilistic polynomial time): 概率图灵机以多项式时间返回结果的问题的复杂性类的概率返回正确的 Yes 或 No, 但有 1/2 的概率返回不知道。
- BPP (bounded-error probabilistic polynomial time): 存在图灵机以使得 Yes 至少以三分之二的概率返回 Yes 或 No 以至多三分之一的概率返回 No, 类似 BQP。
- BQP (bounded-error quantum polynomial time)



- 计算模型的层级(?)
 - 在出现了概率性算法和量子计算以后, 丘奇-图灵论题出现了一些变种。Ethan Bernstein 和 Umesh Vazirani 提出了经典复杂性理论丘奇-图灵论题: 概率图灵机可以高效地模拟任何现实的计算模型。量子复杂性理论的丘奇-图灵论题: 量子图灵机可以高效地模拟任何现实的计算模型。如果最终人们证明了 BPP 严格小于 BQP , 也即量子计算在某些方面要比经典计算更加高效。那么经典复杂性理论丘奇-图灵论题将是错误的。当然, 由于最原始的丘奇-图灵论题不涉及复杂性理论, 只关注了函数的可计算性, 所以其有效性不会受到影响。
 - 计算模型之间有一个层级结构, 可以简单地想成高级的模型能够高效地模拟低级模型的计算过程, 反之则不能。我们想知道是否存在一个最高级的东西。基于信息是物理的这种想法, 利用物理规律进行的计算或许就是最高级的。可能基于相对论、量子场论、大一统理论的计算模型能够比量子计算更高级。是否存在一种最终极的计算模型, 它能够高效地模拟任何一种其他的计算模型?

超(越极)限的量子计算

- 量子计算的限度在哪里? 我们如何超越这些极限?

1.2 量子比特

- 物理状态与物理量：经典vs量子
 - 经典力学：某一时刻相空间中的点
 - 经典电磁学：某一时刻空间中的电磁场电磁场分布
 - 统计物理：系综的概率密度函数
 - 经典信息(寄存器的状态): $B^n = \{0, 1\}^n$ 中的一个字符串
 - 量子力学/量子计算：态矢量、密度算符
 - 共性?

State	
of an ordinary computer	of a quantum computer
<ul style="list-style-type: none"> • $\square \square \dots \square$ bits $x_1 \ x_2 \dots x_n \ x_j \in \mathbb{B}$ 	$\square \square \dots \square$ qubits basis: $ x_1, x_2, \dots, x_n\rangle, \quad x_j \in \mathbb{B}$ arbitrary: $\sum_{x \in \mathbb{B}^n} c_x x\rangle$, where $\sum_{x \in \mathbb{B}^n} c_x ^2 = 1$

单比特

```
In[1]:= (*PacletInstall[
  "https://www.wolframcloud.com/obj/wolframquantumframework/QuantumFramework.paclet",
  ForceVersionInstall->True]*)(*第一次运行时需要下载Paclet*)
```

```
In[2]:= << Wolfram`QuantumFramework`(*每次重新启动Kernel都要导入Paclet*)
```

```
In[3]:= qubit = QuantumState[{α, β}];(*默认计算基, 常用的ONB*)
qubit // TraditionalForm(*叠加态*)
qubit["Matrix"] // MatrixForm(*态矢量*)
m = ((qubit["DensityMatrix"] // Normal) // FullSimplify) /. Conjugate[x_] -> x*;
m // MatrixForm(*密度矩阵*)
Tr[m] // FullSimplify[#, Abs[α]^2 + Abs[β]^2 == 1] &(*迹为1实际上是各个测量结果的概率归一*)
```

```
Out[4]//TraditionalForm=
```

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

```
Out[5]//MatrixForm=
```

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

```
Out[7]//MatrixForm=
```

$$\begin{pmatrix} \text{Abs}[\alpha]^2 & \alpha \beta^* \\ \beta \alpha^* & \text{Abs}[\beta]^2 \end{pmatrix}$$

```
Out[8]= 1
```

qb的参数化与Bloch矢量

```
In[10]:= qb = QuantumOperator[{"U3",  $\theta$ ,  $\phi$ ,  $-\phi$ ][QuantumState[{1, 0}]];
qb // TraditionalForm
```

```
Out[11]//TraditionalForm=
```

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

```
In[12]:= Flatten[qb["Matrix"]] // Normal // MatrixForm
 $\rho$  = qb["DensityMatrix"] // Normal // Simplify[#, ( $\phi$  |  $\theta$ )  $\in$  Reals] &;
% // MatrixForm
Table[Tr[ $\rho$ .PauliMatrix[i]], {i, 3}] // FullSimplify(* $\omega$ 和 $\phi$ 分别是纬度角和经度角*)
```

```
Out[12]//MatrixForm=
```

$$\begin{pmatrix} \cos\left[\frac{\theta}{2}\right] \\ e^{i\phi} \sin\left[\frac{\theta}{2}\right] \end{pmatrix}$$

```
Out[14]//MatrixForm=
```

$$\begin{pmatrix} \cos\left[\frac{\theta}{2}\right]^2 & \frac{1}{2} e^{-i\phi} \sin[\theta] \\ \frac{1}{2} e^{i\phi} \sin[\theta] & \sin\left[\frac{\theta}{2}\right]^2 \end{pmatrix}$$

```
Out[15]=
```

```
{Cos[ $\phi$ ] Sin[ $\theta$ ], Sin[ $\theta$ ] Sin[ $\phi$ ], Cos[ $\theta$ ]}
```

Bloch矢量的应用

- 思考：任意给定一个qubit的量子态 ψ ，如何找到 $U \in U(2)$ 满足 $\psi = U|0\rangle$ ？
 $|0\rangle$ 绕着躺在 $x-y$ 平面上的粉色轴 $(\theta = \frac{\pi}{2}, \phi)$ 旋转蓝色角度 ω 可以到达 ψ
 ψ 的Bloch矢量是 $\{\cos[\phi] \sin[\omega], \sin[\omega] \sin[\phi], \cos[\omega]\}$

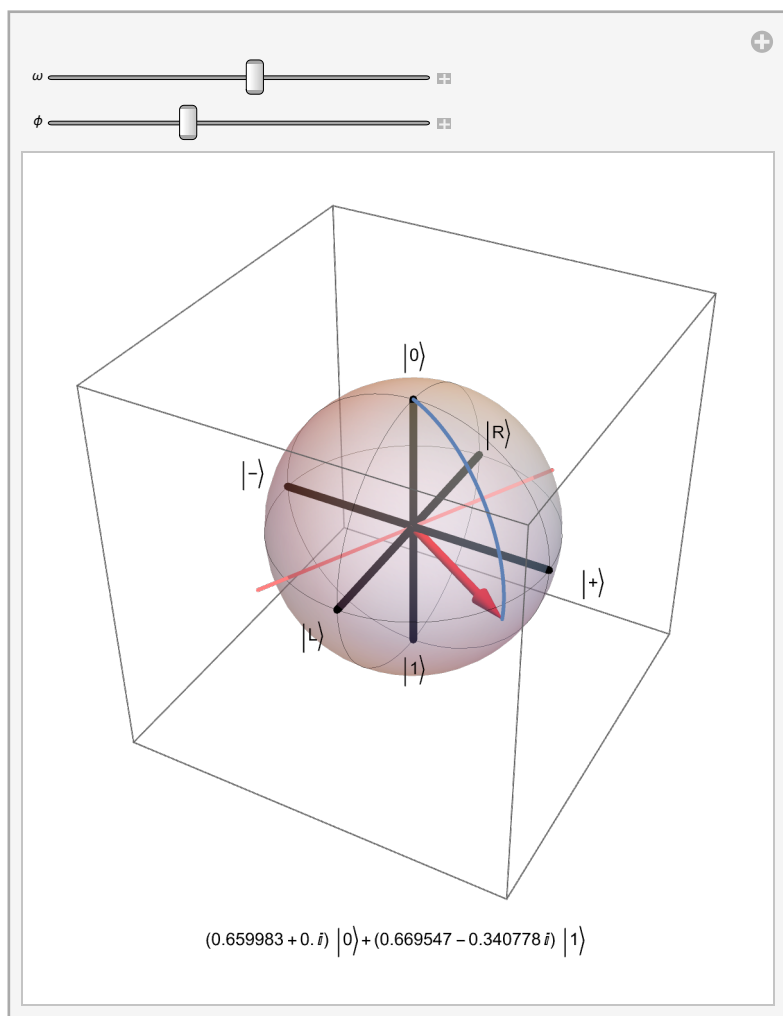
```

In[16]:= Manipulate[mat = Chop[(MatrixExp[-I/2 \omega Total[Table[PauliMatrix[i], {i, 3}] \times
CoordinateTransformData["Spherical" -> "Cartesian", "Mapping", {1, Pi/2, \phi}]]]);

GraphicsColumn[{Show[{ParametricPlot3D[Evaluate[m1 =
Chop[(MatrixExp[-I/2 \omega1 Total[Table[PauliMatrix[i], {i, 3}] \times CoordinateTransformData[
"Spherical" -> "Cartesian", "Mapping", {1, Pi/2, \phi}]]]);
\rho = KroneckerProduct[m1.{1, 0}, Conjugate[m1.{1, 0}]];
Table[Tr[\rho.PauliMatrix[i]], {i, 3}], {\omega1,
0.001, \omega}], QuantumState[mat.{1, 0}][\"BlochPlot\"],
Graphics3D[{Pink, Thick, Line[{CoordinateTransformData[\"Spherical\" -> \"Cartesian\",
\"Mapping\", {1.2, Pi/2, \phi}], -CoordinateTransformData[
\"Spherical\" -> \"Cartesian\", \"Mapping\", {1.2, Pi/2, \phi}]]]}],
Axes -> False, PlotRange -> All, AxesOrigin -> {0, 0, 0}], TraditionalForm[
QuantumState[QuantumState[mat.{1, 0}]]],
{{\omega, 1.7}, 0, Pi}, {{\phi, 1.1}, 0, Pi}]

```


Out[16]=



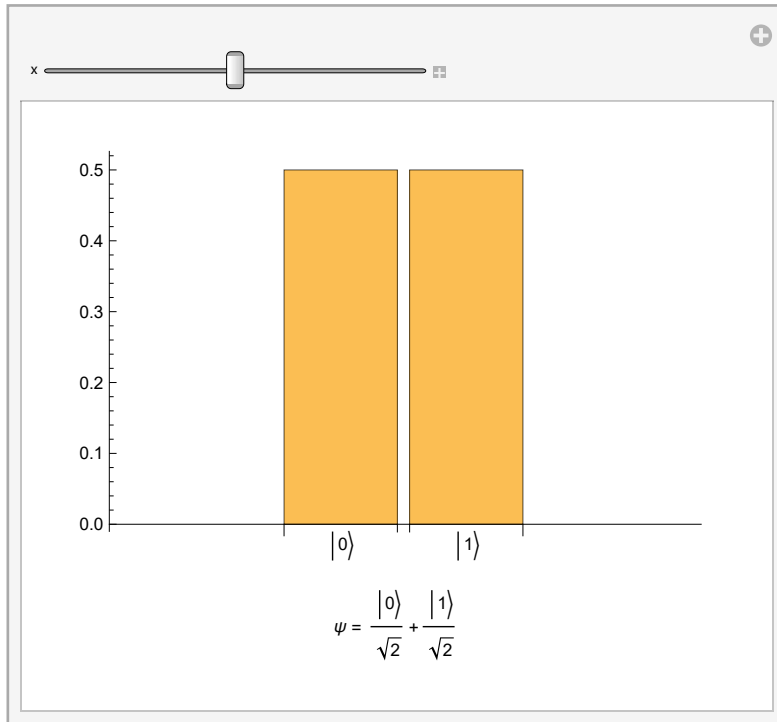
思考：还能找到其他转轴和转角实现目标吗？

单比特测量

计算基下的测量

```
In[17]:= Manipulate[qb = QuantumState[{Cos[x], Sin[x]}];
GraphicsColumn[{QuantumMeasurementOperator["Computational"][qb]["ProbabilityPlot"],
"ψ = " TraditionalForm[qb]], {{x,  $\frac{\text{Pi}}{4}$ }, 0,  $\frac{\text{Pi}}{2}$ }]
```

Out[17]=



除了计算基以外的测量：测量 σ_x

```
In[18]:= QuantumState[{α, β}] // TraditionalForm
```

```
q1 = QuantumState[%, QuantumBasis[⟨ | "+" →  $\frac{1}{\sqrt{2}}$  {1, 1}, "-" →  $\frac{1}{\sqrt{2}}$  {1, -1} ⟩ ]];
```

```
q1 // TraditionalForm
```

Out[18]//TraditionalForm=

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

Out[20]//TraditionalForm=

$$\left(\frac{\alpha}{\sqrt{2}} - \frac{\beta}{\sqrt{2}} \right) |-\rangle + \left(\frac{\alpha}{\sqrt{2}} + \frac{\beta}{\sqrt{2}} \right) |+\rangle$$

```
In[21]:= N[QuantumMeasurementOperator["X"][q1]["Probabilities"] /. α → 1 /. β → 0] // FullSimplify
```

Out[21]=

$$\langle | x_+ \rangle \rightarrow 0.5, | x_- \rangle \rightarrow 0.5 \rangle$$

多量子比特

多比特的计算基

```
In[22]:= MatrixForm /@ Flatten /@ QuantumBasis[{2, 2}][["Association"]]
```

```
Out[22]=
```

$$\langle | \begin{matrix} |00\rangle \rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, |01\rangle \rightarrow \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, |10\rangle \rightarrow \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, |11\rangle \rightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \end{matrix} \rangle$$

```
In[23]:= Table[Symbol["a" <-> ToString[i] <-> ToString[j]], {i, 0, 1}, {j, 0, 1}]
```

```
Out[23]=
```

```
{{a00, a01}, {a10, a11}}
```

```
In[24]:= qs2 = QuantumState[{a"00", a"01", a"10", a"11"}, QuantumBasis[{2, 2}]];  
% // TraditionalForm
```

```
Out[25]//TraditionalForm=
```

$$|00\rangle a_{00} + |01\rangle a_{01} + |10\rangle a_{10} + |11\rangle a_{11}$$

```
In[26]:= QuantumMeasurementOperator["Computational", {1}][qs2];(*在计算基下测量第一个比特*)  
#["Normalized"] & /@ %["StateAssociation"] // TraditionalForm
```

```
Out[27]//TraditionalForm=
```

$$\langle | \begin{matrix} |0\rangle \rightarrow \frac{|00\rangle a_{00}}{\sqrt{\text{Abs}[a_{00}]^2 + \text{Abs}[a_{01}]^2}} + \frac{|01\rangle a_{01}}{\sqrt{\text{Abs}[a_{00}]^2 + \text{Abs}[a_{01}]^2}}, \\ |1\rangle \rightarrow \frac{|10\rangle a_{10}}{\sqrt{\text{Abs}[a_{10}]^2 + \text{Abs}[a_{11}]^2}} + \frac{|11\rangle a_{11}}{\sqrt{\text{Abs}[a_{10}]^2 + \text{Abs}[a_{11}]^2}} \end{matrix} \rangle$$

Bell 基

```
In[28]:= QuantumState /@ QuantumBasis["Bell"][["Association"] // TraditionalForm
```

```
Out[28]//TraditionalForm=
```

$$\langle | \begin{matrix} |\Phi^+\rangle \rightarrow \frac{|00\rangle}{\sqrt{2}} + \frac{|11\rangle}{\sqrt{2}}, |\Phi^-\rangle \rightarrow \frac{|00\rangle}{\sqrt{2}} - \frac{|11\rangle}{\sqrt{2}}, |\Psi^+\rangle \rightarrow \frac{|01\rangle}{\sqrt{2}} + \frac{|10\rangle}{\sqrt{2}}, |\Psi^-\rangle \rightarrow \frac{|01\rangle}{\sqrt{2}} - \frac{|10\rangle}{\sqrt{2}} \end{matrix} \rangle$$

```
In[29]:= MatrixForm /@ QuantumBasis["Bell"][["Association"]]
```

```
Out[29]=
```

$$\langle | \begin{matrix} |\Phi^+\rangle \rightarrow \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}, |\Phi^-\rangle \rightarrow \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ -\frac{1}{\sqrt{2}} \end{pmatrix}, |\Psi^+\rangle \rightarrow \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}, |\Psi^-\rangle \rightarrow \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} \end{matrix} \rangle$$

在计算基下测量Bell态，总是得到相同的结果

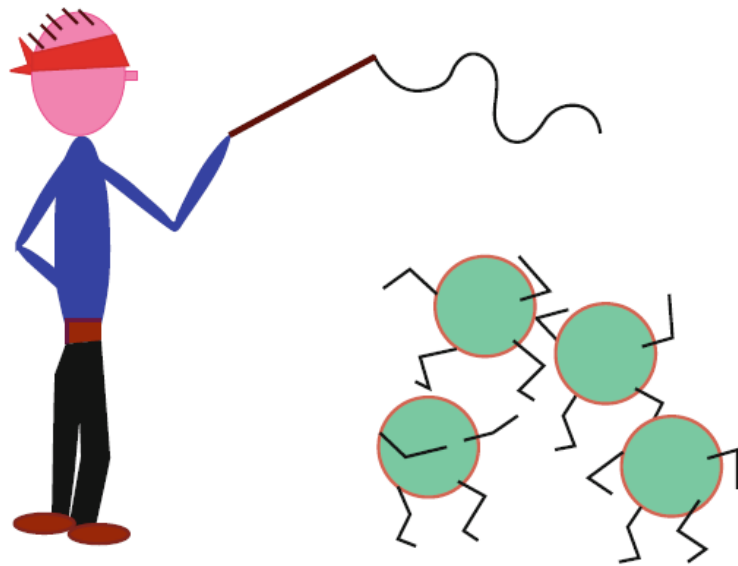
```
In[30]:= QuantumMeasurementOperator["Computational", {1}][QuantumState[{1, 0, 0, 1}]]["StateAssociation"] // TraditionalForm
```

```
Out[30]//TraditionalForm=
```

$$\langle |0\rangle \rightarrow |00\rangle, |1\rangle \rightarrow |11\rangle \rangle$$

1.3 量子计算

Quantum Computing,
Fig. 1 Unitary evolution:
controlling qubits without
looking at them



In[]:=

单比特门 U(2)

X 门和 $R_X(\theta)$ 门

```
In[31]:= opx = QuantumOperator["X"];
          opx = QuantumOperator[PauliMatrix[1]];
          opx["Matrix"] // Normal // MatrixForm
```

Out[33]//MatrixForm=

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

```
In[34]:= opx // TraditionalForm
```

Out[34]//TraditionalForm=

$$|0\rangle\langle 1| + |1\rangle\langle 0|$$

```
In[35]:= (QuantumState /@ (#/Norm[#] & /@ Eigenvectors[PauliMatrix[1]])) // TraditionalForm
```

Out[35]//TraditionalForm=

$$\left\{ -\frac{|0\rangle}{\sqrt{2}} + \frac{|1\rangle}{\sqrt{2}}, \frac{|0\rangle}{\sqrt{2}} + \frac{|1\rangle}{\sqrt{2}} \right\}$$

```
In[36]:= opx[QuantumState[{α, β}]] // TraditionalForm
```

Out[36]//TraditionalForm=

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

```
In[37]:= QuantumOperator[{"RX",  $\theta$ ]["Matrix"] // Normal // FullSimplify // MatrixForm
```

```
MatrixExp[-I  $\frac{\theta}{2}$  PauliMatrix[1]] // MatrixForm(*绕着x轴旋转角度 $\theta$ *)
```

```
I % /.  $\theta \rightarrow \text{Pi}$  // MatrixForm
```

```
Out[37]//MatrixForm=
```

$$\begin{pmatrix} \cos\left[\frac{\theta}{2}\right] & -i \sin\left[\frac{\theta}{2}\right] \\ -i \sin\left[\frac{\theta}{2}\right] & \cos\left[\frac{\theta}{2}\right] \end{pmatrix}$$

```
Out[38]//MatrixForm=
```

$$\begin{pmatrix} \cos\left[\frac{\theta}{2}\right] & -i \sin\left[\frac{\theta}{2}\right] \\ -i \sin\left[\frac{\theta}{2}\right] & \cos\left[\frac{\theta}{2}\right] \end{pmatrix}$$

```
Out[39]//MatrixForm=
```

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Y 门和 $R_Y(\theta)$ 门

```
In[40]:= opy = QuantumOperator[PauliMatrix[2]];
          opy["Matrix"] // Normal // MatrixForm
```

```
Out[41]//MatrixForm=
```

$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

```
In[42]:= QuantumOperator[{"RY", \theta}]["Matrix"] // Normal // FullSimplify // MatrixForm
```

$$\text{MatrixExp}\left[-i \frac{\theta}{2} \text{PauliMatrix}[2]\right] // \text{MatrixForm}$$

```
I % /. \theta \to Pi // MatrixForm
```

```
Out[42]//MatrixForm=
```

$$\begin{pmatrix} \cos\left[\frac{\theta}{2}\right] & -\sin\left[\frac{\theta}{2}\right] \\ \sin\left[\frac{\theta}{2}\right] & \cos\left[\frac{\theta}{2}\right] \end{pmatrix}$$

```
Out[43]//MatrixForm=
```

$$\begin{pmatrix} \cos\left[\frac{\theta}{2}\right] & -\sin\left[\frac{\theta}{2}\right] \\ \sin\left[\frac{\theta}{2}\right] & \cos\left[\frac{\theta}{2}\right] \end{pmatrix}$$

```
Out[44]//MatrixForm=
```

$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

Z 门和 $R_Z(\theta)$ 门

```
In[45]:= QuantumCircuitOperator["Z"];
          %["Matrix"] // Normal // MatrixForm
```

```
Out[46]//MatrixForm=
```

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

```
In[47]:= QuantumOperator[{"RZ", \theta}]["Matrix"] // Normal // FullSimplify // MatrixForm
```

$$\text{MatrixExp}\left[-i \frac{\theta}{2} \text{PauliMatrix}[3]\right] // \text{MatrixForm}$$

```
I % /. \theta \to Pi // MatrixForm
```

```
Out[47]//MatrixForm=
```

$$\begin{pmatrix} e^{-\frac{i\theta}{2}} & 0 \\ 0 & e^{\frac{i\theta}{2}} \end{pmatrix}$$

```
Out[48]//MatrixForm=
```

$$\begin{pmatrix} e^{-\frac{i\theta}{2}} & 0 \\ 0 & e^{\frac{i\theta}{2}} \end{pmatrix}$$

```
Out[49]//MatrixForm=
```

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

绕着轴 $\mathbf{n}=(x,y,z)$ 旋转角度 θ

```
In[50]:= n[i_] := {x, y, z}[[i]];
```

```
In[51]:= MatrixExp[-I  $\frac{\theta}{2}$  Sum[n[i] * PauliMatrix[i], {i, 3}]] // Simplify[#, {x, y, z} ∈ Sphere[]] & // FullSimplify //
```

MatrixForm(*如何计算? 习题4.4*)

```
Out[51]//MatrixForm=
```

$$\begin{pmatrix} \cos\left[\frac{\theta}{2}\right] - i z \sin\left[\frac{\theta}{2}\right] & (-i x - y) \sin\left[\frac{\theta}{2}\right] \\ (-i x + y) \sin\left[\frac{\theta}{2}\right] & \cos\left[\frac{\theta}{2}\right] + i z \sin\left[\frac{\theta}{2}\right] \end{pmatrix}$$

Hadamard 门

```
In[52]:= H = QuantumCircuitOperator["H"];
%["Matrix"] // Normal // MatrixForm
```

```
Out[53]//MatrixForm=
```

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

```
In[54]:= (H/*H)["Matrix"] // Normal // MatrixForm
```

```
Out[54]//MatrixForm=
```

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

```
In[55]:= H[QuantumState[{1, 0}]] // TraditionalForm
H[QuantumState[{0, 1}]] // TraditionalForm
```

```
Out[55]//TraditionalForm=
```

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

```
Out[56]//TraditionalForm=
```

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

用 H 门制作均匀(等系数)叠加态

```
In[57]:= QuantumCircuitOperator["HHH"][QuantumState[{1, 0, 0, 0, 0, 0, 0, 0}]] // TraditionalForm
```

```
Out[57]//TraditionalForm=
```

$$\frac{|000\rangle}{2\sqrt{2}} + \frac{|001\rangle}{2\sqrt{2}} + \frac{|010\rangle}{2\sqrt{2}} + \frac{|011\rangle}{2\sqrt{2}} + \frac{|100\rangle}{2\sqrt{2}} + \frac{|101\rangle}{2\sqrt{2}} + \frac{|110\rangle}{2\sqrt{2}} + \frac{|111\rangle}{2\sqrt{2}}$$

相位门 S

```
In[58]:= QuantumOperator["S"]["Matrix"] // Normal // MatrixForm
Out[58]//MatrixForm=
```

$$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

T 门

```
In[59]:= QuantumOperator["T"]["Matrix"] // Normal // MatrixForm
Out[59]//MatrixForm=
```

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{pmatrix}$$

单比特门的 z-y-z 分解 旋量空间的欧拉角

```
In[60]:= zyz =
  E^I \alpha QuantumOperator[{"RZ", \beta}] @ QuantumOperator[{"RY", \gamma}] @ QuantumOperator[{"RZ", \delta}];
zyz["Matrix"] // Normal // MatrixForm // FullSimplify(*定理4.1*)
Out[61]//MatrixForm=
```

$$\begin{pmatrix} \frac{1}{2} e^{\frac{1}{2} i (2 \alpha - \beta - \gamma - \delta)} (1 + e^{i \gamma}) & \frac{1}{2} i e^{\frac{1}{2} i (2 \alpha - \beta - \gamma + \delta)} (-1 + e^{i \gamma}) \\ -\frac{1}{2} i e^{\frac{1}{2} i (2 \alpha + \beta - \gamma - \delta)} (-1 + e^{i \gamma}) & \frac{1}{2} e^{\frac{1}{2} i (2 \alpha + \beta - \gamma + \delta)} (1 + e^{i \gamma}) \end{pmatrix}$$

```
In[62]:= Det[zyz["Matrix"] // Normal](*么正与特殊么正之间只相差相位角，半直积*)
Out[62]=
```

$$e^{2i\alpha}$$

多比特量子门 $U(2^n)$

CNOT (controlled-NOT) 门

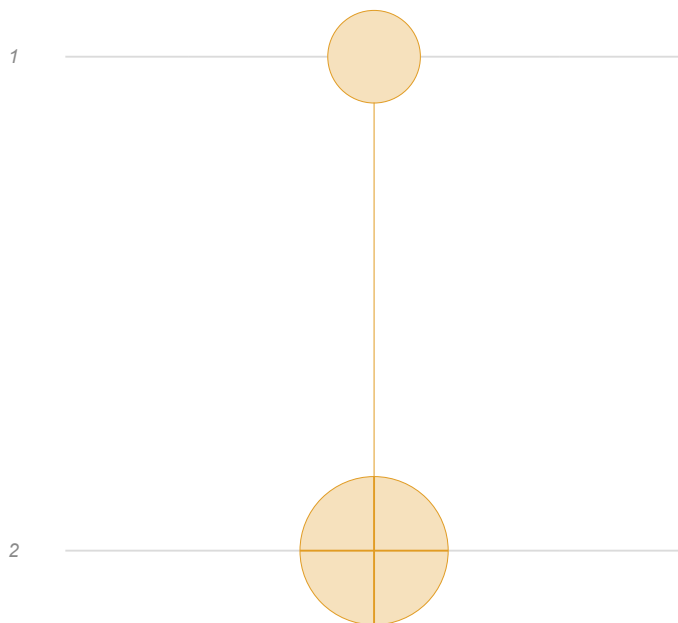
```
In[63]:= QuantumOperator["CNOT"] // TraditionalForm(*模2加法*)
```

```
QuantumCircuitOperator["CNOT"] // TraditionalForm
```

```
Out[63]//TraditionalForm=
```

$$|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 11| + |11\rangle\langle 10|$$

```
Out[64]//TraditionalForm=
```



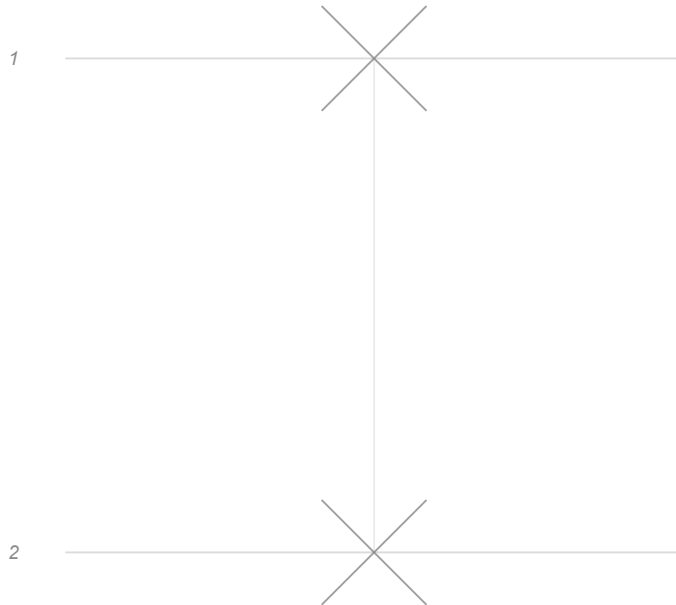
交换门

```
In[65]:= QuantumOperator["SWAP"] // TraditionalForm
QuantumCircuitOperator["SWAP"] // TraditionalForm
```

```
Out[65]//TraditionalForm=
```

$$|00\rangle\langle 00| + |01\rangle\langle 10| + |10\rangle\langle 01| + |11\rangle\langle 11|$$

```
Out[66]//TraditionalForm=
```



SWAP门拆成三个CNOT门

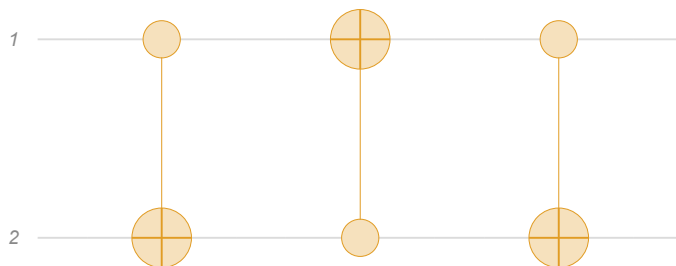
```
In[67]:= QuantumCircuitOperator[{QuantumOperator["CNOT", {1, 2}], QuantumOperator["CNOT", {2, 1}],
  QuantumOperator["CNOT", {1, 2}]}] == QuantumCircuitOperator["SWAP"]
```

```
Out[67]=
```

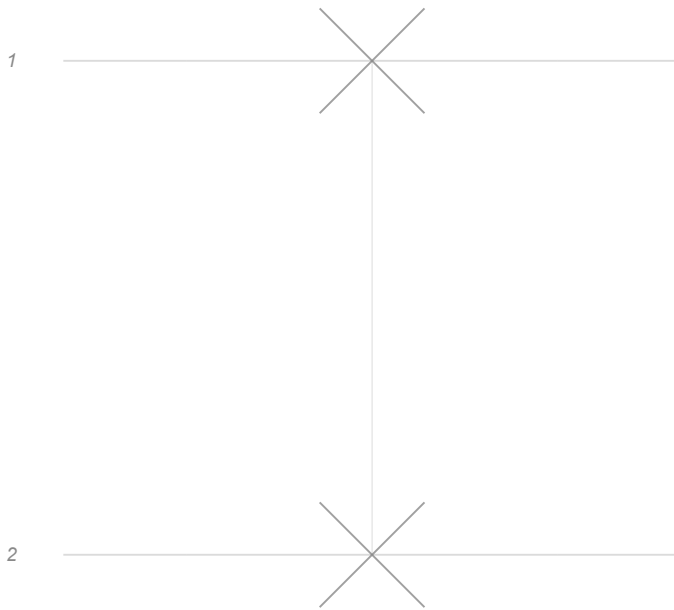
True

```
In[68]:= QuantumCircuitOperator[{QuantumOperator["CNOT", {1, 2}],
  QuantumOperator["CNOT", {2, 1}], QuantumOperator["CNOT", {1, 2}]]["Diagram"]
```


```
Out[68]=
```

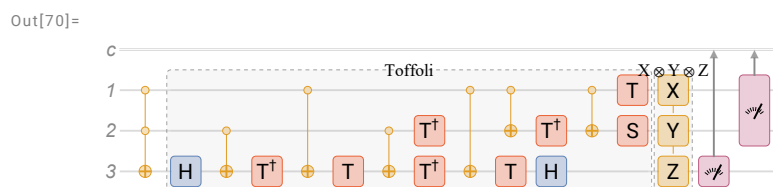


```
In[69]:= QuantumCircuitOperator["SWAP"]["Diagram"]
Out[69]=
```



Toffoli 门和测量

```
In[70]:= QuantumCircuitOperator[{QuantumOperator[ Picture: Schrödinger  
Dimension: 8→8 Arity: 3  
Qudits: 3→3],  
QuantumCircuitOperator["Toffoli"], QuantumCircuitOperator[{"XYZ"}],  
QuantumMeasurementOperator["X", {3}],  
QuantumMeasurementOperator["X", {1, 2}]]["Diagram"]
```



```
In[71]:= tfl[a_, b_, c_] := {a, b, Mod[c + a b, 2]}
```

量子线路

- 不出现回路，无环
- 不允许扇入扇出，扇入不可逆，扇出是量子克隆

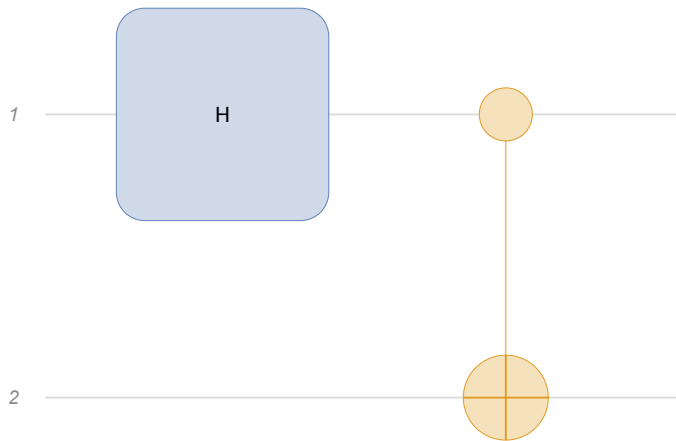
通用门

- 一组能够以任意的精度模拟任何量子么正变换的门。
- 精度越高，需要的门越多。

从计算基中的态获得Bell 态

```
In[72]:= qco = QuantumCircuitOperator[{"H", "CNOT"}];
qco["Diagram"]
```

Out[73]=



```
In[74]:= TableForm[
  {QuantumBasis[{2, 2}]["Names"], TraditionalForm /@ qco /@ (QuantumBasis[{2, 2}]["PureStates"]),
    #["Formula"] & /@ (QuantumState[#, QuantumBasis["Bell"]] & /@
      (qco /@ (QuantumBasis[{2, 2}]["PureStates"]))) //
    Transpose, TableHeadings -> {{1, 2, 3, 4}, {"输入", "输出"}}]
```

Out[74]//TableForm=

	输入	输出
1	$ 00\rangle$	$\frac{ 00\rangle}{\sqrt{2}} + \frac{ 11\rangle}{\sqrt{2}}$ $ \Phi^+\rangle$
2	$ 01\rangle$	$\frac{ 01\rangle}{\sqrt{2}} + \frac{ 10\rangle}{\sqrt{2}}$ $ \Psi^+\rangle$
3	$ 10\rangle$	$\frac{ 00\rangle}{\sqrt{2}} - \frac{ 11\rangle}{\sqrt{2}}$ $ \Phi^-\rangle$
4	$ 11\rangle$	$\frac{ 01\rangle}{\sqrt{2}} - \frac{ 10\rangle}{\sqrt{2}}$ $ \Psi^-\rangle$

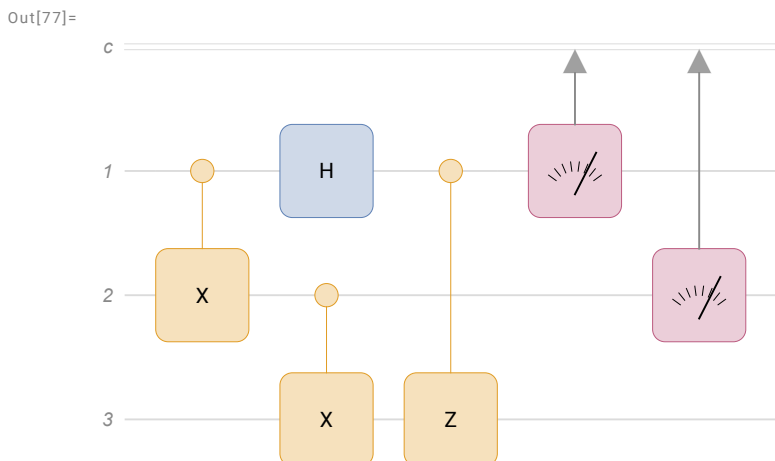
利用Bell态进行隐形传态(teleportation)

方法1

设第1,2个比特是Alice的，而第3个比特是Bob的，双横线表示经典比特通信

```
In[75]:= ops = {QuantumOperator["CX"], QuantumOperator["H"],
  QuantumOperator["CX", {2, 3}], QuantumOperator["CZ", {1, 3}],
  QuantumMeasurementOperator[], QuantumMeasurementOperator[{2}];
```

```
In[76]:= circuit = QuantumCircuitOperator[ops];
  %["Diagram"]
```



```
In[78]:= QuantumCircuitOperator[{"CX", "H", "CX" -> {2, 3}, "CZ" -> {1, 3}, {1}, {2}}] ==
  circuit(*也可以用简写来构建线路*)
```

```
Out[78]=
  True
```

Alice的第一个比特 ψ_0 想要传给Bob，这个比特对Alice和Bob来说都是未知的，Alice的第二个比特和Bob的比特被制备在Bell态上

```
In[79]:=  $\psi_0$  = QuantumTensorProduct[QuantumState[{ $\alpha$ ,  $\beta$ }], QuantumState["PhiPlus"]];
  %["Formula"]
```

```
Out[80]=
```

$$\frac{\alpha |000\rangle}{\sqrt{2}} + \frac{\alpha |011\rangle}{\sqrt{2}} + \frac{\beta |100\rangle}{\sqrt{2}} + \frac{\beta |111\rangle}{\sqrt{2}}$$

```
In[81]:= qss = circuit[ $\psi_0$ ]["StateAssociation"];
  % // TraditionalForm
```

```
Out[82]//TraditionalForm=
```

$$\left\langle \begin{aligned} &|00\rangle \rightarrow \frac{1}{2} \alpha |000\rangle + \frac{1}{2} \beta |001\rangle, |01\rangle \rightarrow \frac{1}{2} \alpha |010\rangle + \frac{1}{2} \beta |011\rangle, \\ &|10\rangle \rightarrow \frac{1}{2} \alpha |100\rangle + \frac{1}{2} \beta |101\rangle, |11\rangle \rightarrow \frac{1}{2} \alpha |110\rangle + \frac{1}{2} \beta |111\rangle \end{aligned} \right\rangle$$

无论Alice的测量结果是什么，Bob手中的态都是 ψ_0

```
In[83]:= (QuantumPartialTrace[#, {1, 2}] & /@ qss) // TraditionalForm
```

```
Out[83]//TraditionalForm=
```

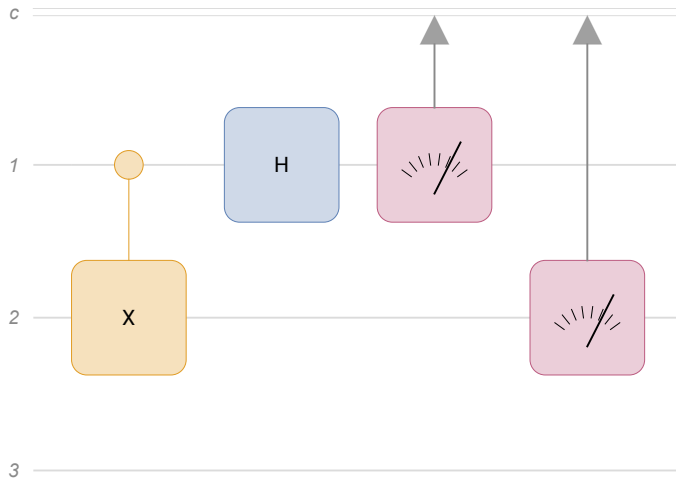
$$\begin{aligned}
 &\langle |00\rangle \rightarrow \frac{1}{4} \alpha \text{Conjugate}[\alpha] (|0\rangle\langle 0|) + \frac{1}{4} \alpha \text{Conjugate}[\beta] (|0\rangle\langle 1|) + \frac{1}{4} \beta \text{Conjugate}[\alpha] (|1\rangle\langle 0|) + \frac{1}{4} \beta \text{Conjugate}[\beta] (|1\rangle\langle 1|), \\
 &|01\rangle \rightarrow \frac{1}{4} \alpha \text{Conjugate}[\alpha] (|0\rangle\langle 0|) + \frac{1}{4} \alpha \text{Conjugate}[\beta] (|0\rangle\langle 1|) + \frac{1}{4} \beta \text{Conjugate}[\alpha] (|1\rangle\langle 0|) + \frac{1}{4} \beta \text{Conjugate}[\beta] (|1\rangle\langle 1|), \\
 &|10\rangle \rightarrow \frac{1}{4} \alpha \text{Conjugate}[\alpha] (|0\rangle\langle 0|) + \frac{1}{4} \alpha \text{Conjugate}[\beta] (|0\rangle\langle 1|) + \frac{1}{4} \beta \text{Conjugate}[\alpha] (|1\rangle\langle 0|) + \frac{1}{4} \beta \text{Conjugate}[\beta] (|1\rangle\langle 1|), \\
 &|11\rangle \rightarrow \\
 &\quad \frac{1}{4} \alpha \text{Conjugate}[\alpha] (|0\rangle\langle 0|) + \frac{1}{4} \alpha \text{Conjugate}[\beta] (|0\rangle\langle 1|) + \frac{1}{4} \beta \text{Conjugate}[\alpha] (|1\rangle\langle 0|) + \frac{1}{4} \beta \text{Conjugate}[\beta] (|1\rangle\langle 1|) \rangle
 \end{aligned}$$

方法2: LOCC (local operations and classical communication)

Alice在本地执行一系列的逻辑门和测量，将测量结果告诉Bob

```
In[84]:= qco2 = QuantumCircuitOperator[{"I" → 3, {"CX"}, "H", {1}, {2}}];
qco2["Diagram"]
```

Out[85]=



Bob根据Alice的测量结果对自己的态作用逻辑门以得到正确的 ψ_0

```
In[86]:= qss2 = qco2[ψ0]["StateAssociation"];(*没有正确的归一化*)
% // TraditionalForm
```

Out[87]//TraditionalForm=

$$\begin{aligned} \langle |00\rangle &\rightarrow \frac{1}{2} \alpha |000\rangle + \frac{1}{2} \beta |001\rangle, |01\rangle \rightarrow \frac{1}{2} \beta |010\rangle + \frac{1}{2} \alpha |011\rangle, \\ |10\rangle &\rightarrow \frac{1}{2} \alpha |100\rangle - \frac{1}{2} \beta |101\rangle, |11\rangle \rightarrow -\frac{1}{2} \beta |110\rangle + \frac{1}{2} \alpha |111\rangle \end{aligned}$$

```
In[88]:= QuantumOperator["X", 3]@qss2[[2]] // TraditionalForm
-QuantumOperator["Z", 3]@qss2[[3]] // TraditionalForm
-QuantumOperator["X", 3]@QuantumOperator["Z", 3]@qss2[[4]] // TraditionalForm
```

Out[88]//TraditionalForm=

$$\frac{1}{2} \alpha |100\rangle + \frac{1}{2} \beta |101\rangle$$

Out[89]//TraditionalForm=

$$\frac{1}{2} \alpha |100\rangle + \frac{1}{2} \beta |101\rangle$$

Out[90]//TraditionalForm=

$$\frac{1}{2} \alpha |000\rangle + \frac{1}{2} \beta |001\rangle$$

信息的传递就是物质的传递

1.4 量子算法

量子计算机上的经典计算

- Landauer原理

- Information is physical (Landauer, 1991).

能耗与计算中的不可逆性之间有什么联系？兰道尔原理（Landauer's principle）提供了二者之间的桥梁，它指出为了擦除信息必须要消耗能量。更确切地说，兰道尔原理可以陈述如下：

兰道尔原理（第一种形式）：假设一台计算机擦除了 1 比特的信息，那么耗散到整个环境中的能量至少是 $k_B T \ln 2$ ，其中 k_B 是玻尔兹曼常数， T 是电脑工作环境的温度。

根据热力学定律，可以给出兰道尔原理另一种基于熵耗散的形式，而不是能量耗散形式：

兰道尔原理（第二种形式）：假设一台计算机擦除了 1 比特的信息，那么整个环境的熵至少增加了 $k_B \ln 2$ ，其中 k_B 是玻尔兹曼常数。

- 不可逆的经典逻辑门都可以等效地用可逆的量子门替代。
 - 曲面 $z = f(x, y)$ 可能不是单射，但是 $\text{graph of a function } (x, y, z(x, y))$ 是单射
- 用量子测量的概率性模拟概率经典计算机

经典Toffoli门

```
In[91]:= tfl[a_, b_, c_] := {a, b, Mod[c + a b, 2]}

In[92]:= Flatten[Table[{a, b, c}, {a, 0, 1}, {b, 0, 1}, {c, 0, 1}], 2]
tfl @@@ %
tfl @@@ %(*连续作用两次回到原来的值*)

Out[92]=
{{0, 0, 0}, {0, 0, 1}, {0, 1, 0}, {0, 1, 1}, {1, 0, 0}, {1, 0, 1}, {1, 1, 0}, {1, 1, 1}}

Out[93]=
{{0, 0, 0}, {0, 0, 1}, {0, 1, 0}, {0, 1, 1}, {1, 0, 0}, {1, 0, 1}, {1, 1, 1}, {1, 1, 0}}

Out[94]=
{{0, 0, 0}, {0, 0, 1}, {0, 1, 0}, {0, 1, 1}, {1, 0, 0}, {1, 0, 1}, {1, 1, 0}, {1, 1, 1}}
```

量子并行

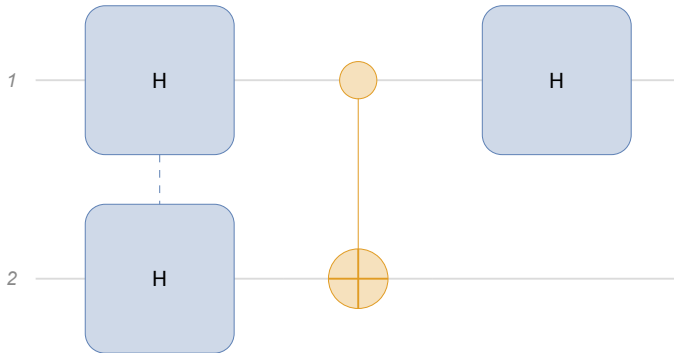
- 量子计算机能同时（在同一个幺正变换中）计算函数 $f(x)$ 在许多 x 处的值

Deutsch 算法 $f: \{0,1\} \rightarrow \{0,1\}$

平衡型函数 $f(x) = x$ 或者 $f(x) = x \oplus 1$

```
In[95]:= qc1 = QuantumCircuitOperator[{"H" -> {1, 2}, "CNOT", "H"}]; (*设 f(x) = x*)
qc1["Diagram"]
QuantumCircuitMultiwayGraph[qc1];
```

Out[96]=



```
In[98]:= steps = ComposeList[qc1["Operators"], QuantumState[{0, 1, 0, 0}]];
```

```
In[99]:= Grid[Transpose[{Style[#, Bold] & /@ {"初态", "经过前两个H门", "经过U_f", "经过最后的H门"},
  #["Formula"] & /@ steps[[;;]]}], Frame -> All, Alignment -> Left]
```

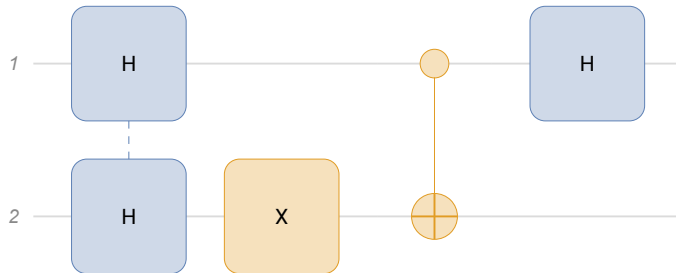
Out[99]=

初态	$ 01\rangle$
经过前两个H门	$\frac{ 00\rangle}{2} - \frac{ 01\rangle}{2} + \frac{ 10\rangle}{2} - \frac{ 11\rangle}{2}$
经过U _f	$\frac{ 00\rangle}{2} - \frac{ 01\rangle}{2} - \frac{ 10\rangle}{2} + \frac{ 11\rangle}{2}$
经过最后的H门	$\frac{ 10\rangle}{\sqrt{2}} - \frac{ 11\rangle}{\sqrt{2}}$

In[100]:=

```
qc2 = QuantumCircuitOperator[{"H" → {1, 2}, "X" → 2, "CNOT", "H"}; (*f(x)=x⊕1*)
qc2["Diagram"]
qc2[QuantumState[{0, 1, 0, 0}]]["Formula"]
```

Out[101]=



Out[102]=

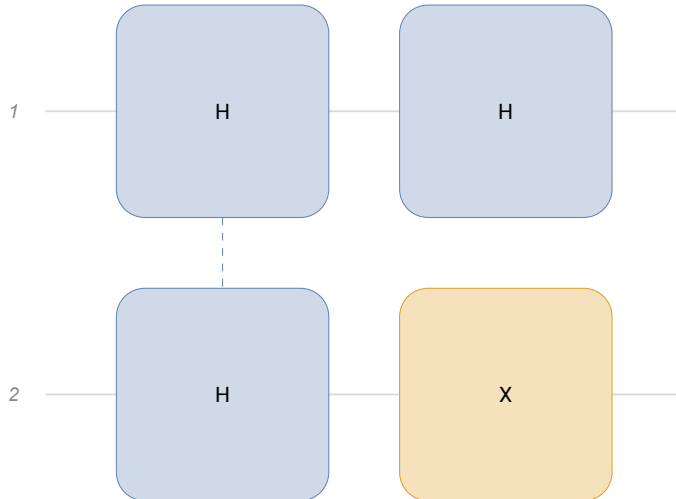
$$-\frac{|10\rangle}{\sqrt{2}} + \frac{|11\rangle}{\sqrt{2}}$$

常数型函数 $f(x)=1$ 或 $f(x)=0$

In[103]:=

```
qc3 = QuantumCircuitOperator[{"H" → {1, 2}, "X" → 2, "H"}; (*f(x)=1*)
qc3["Diagram"]
qc3[QuantumState[{0, 1, 0, 0}]]["Formula"]
```

Out[104]=



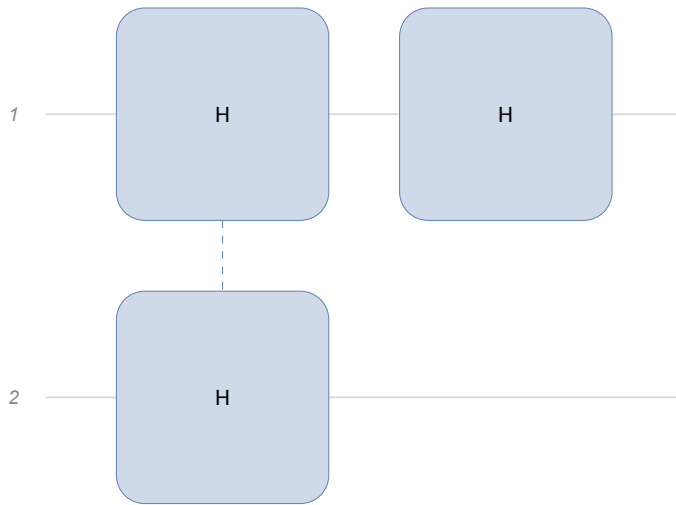
Out[105]=

$$-\frac{|00\rangle}{\sqrt{2}} + \frac{|01\rangle}{\sqrt{2}}$$

In[106]:=

```
qc4 = QuantumCircuitOperator[{"H" → {1, 2}, "H"}]; (*f(x)=1*)
qc4["Diagram"]
qc4[QuantumState[{0, 1, 0, 0}]]["Formula"]
```

Out[107]=



Out[108]=

$$\frac{|00\rangle}{\sqrt{2}} - \frac{|01\rangle}{\sqrt{2}}$$

- 根据第一个qubit的测量结果，我们就能知道函数是平衡型还是常数型，也即计算了 $f(0) \oplus f(1)$ 的值，对于常数型 $f(0) \oplus f(1)=0$ ，对于平衡型 $f(0) \oplus f(1)=1$
- 每个量子线路都只使用了一次 U_f ，
即只需要一次计算就完成了经典计算需要两次计算才能完成的任务
- 启发：对于那些经典计算获得的信息量超过问题的答案能够提供的信息量的问题（尝试计算信息量），也即经典计算存在计算冗余的问题，量子计算具有潜在的加速可能
- 判断平衡型和常数型的量子算法推广到 f 具有 n 个slot的情况，称为Deutsch-Jozsa算法，参见 Sec.1.4.4

量子算法的总结

- 基于量子Fourier变换的方法

- $y_s = \frac{1}{\sqrt{n}} \sum_{r=1}^n u_r e^{2\pi i (r-1)(s-1)/n}, s=1, \dots, n$

- $|j\rangle \rightarrow \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j k / 2^n} |k\rangle, j=0, \dots, 2^n-1$

- 量子傅里叶变换能提供指数加速，但是傅里叶变换的结果被保存在振幅中，需要巧妙地提取

- 应用：Shor算法、Deutsch-Jozsa算法、隐藏子群问题。

- 量子搜索算法 经典 $O\left(\frac{N}{2}\right)$ 量子 $O(\sqrt{N})$

- 量子模拟

- 量子计算的威力：BQP是严格大于BPP？但愿存在某些问题是量子计算机的概率算法能高效解决，但是概率图灵机不能高效解决的

1.5 量子信息处理（量子计算）实验

Stern-Gerlach

对于级联的ZXZ测量，量子力学正确地解释了实验结果

前景

- 阈值定理，如果量子噪声水平降低到某个阈值以下，就能在工程上真正实现任意规模的量子计算。
- 状态层析和量子层析
- 小规模：量子通信
- 中等规模（Noisy Intermediate Scale Quantum (NISQ)）量子计算
- 大规模
- 被用于量子计算的物理系统：1. 光学系统；2. 超导线路；3. 离子阱；4. 冷原子；5. NV 色心；6. 核磁共振；7.任意子；8.transmon

多少比特了？

- IBM Quantum
- 量子本源
- ...

1.6 量子信息

经典信息论

- 两个基本问题：
 - 数据压缩的极限是什么？答：熵 H
 - 数据传输速率的极限是什么？答：信道容量 C

量子信息论

- 狭义的量子信息：量子领域中和经典信息对应的理论。
- 量子信息的基本目标
 - 确定量子力学静态资源的基本类型（资源理论）
 - 确定量子力学动态过程的基本类型（量子信道）
 - 量化基本动态过程导致的资源的取舍。例如，回答问题：为了在噪声信道中可靠地传递消息，至少需要多少资源？

lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed
diam nonummy.

Lorem Ipsum