응용물리연구실심화실습3 (PHY3075)

2024년 11월 5주차

응용물리학과 2022006971 이민성

번역

- 텔포레이션

  텔레포테이션(양자 텔레포테이션 또는 단순히 텔레포테이션)은 송신자(Alice)가 수신자(Bob)에게 큐비트를 전송하는 프로토콜로, 공유된 얽힘 상태(구체적으로 한 개의 e-bit)와 두 비트의 고전적 통신을 활용합니다. "텔레포테이션"이라는 이름은 과학 소설에서 물질이 한 장소에서 다른 장소로 미래적인 과정을 통해 이동하는 개념을 암시하도록 의도되었지만, 양자 텔레포테이션에서 실제로 텔레포트되는 것은 물질이 아니라 양자 정보라는 점을 이해해야 합니다.

  텔레포테이션의 설정은 다음과 같습니다.

  우리는 Alice와 Bob이 e-bit를 공유하고 있다고 가정합니다. Alice는 큐비트 A를 가지고 있고, Bob은 큐비트 B를 가지고 있으며, 함께 쌍 (A,B)은 상태 $|\phi^+\rangle$에 있습니다. 예를 들어, Alice와 Bob이 과거에 같은 장소에 있었고, 그들이 A와 B 큐비트를 상태 $|\phi^+\rangle$로 준비한 후 각자의 큐비트를 가지고 떠났을 수 있습니다. 또는, 제3자나 복잡한 분산 프로세스를 포함한 다른 과정이 이 공유된 e-bit를 확립하는 데 사용되었을 수도 있습니다. 이러한 세부 사항은 텔레포테이션 프로토콜 자체의 일부는 아닙니다.

  그 후 Alice는 세 번째 큐비트 Q를 가지게 되며, 이를 Bob에게 전송하고자 합니다. 큐비트 Q의 상태는 Alice와 Bob 모두에게 알려지지 않은 것으로 간주되며, 이에 대해 어떤 가정도 하지 않습니다. 예를 들어, 큐비트 Q는 Alice와 Bob이 접근할 수 없는 하나 이상의 다른 시스템과 얽혀 있을 수 있습니다. Alice가 큐비트 Q를 Bob에게 전송하고자 한다는 것은 Alice가 Q를 물리적으로 Bob에게 건네준 것처럼, QQ가 프로토콜 시작 시 가졌던 상태와 동일한 상태로 Bob이 큐비트를 가지기를 원한다는 것을 의미합니다. 이는 Q가 다른 시스템과 가졌던 모든 상관관계를 포함합니다.

  물론 Alice가 큐비트 Q를 물리적으로 Bob에게 보내고, Q가 전달 중 변경되거나 방해받지 않았다고 가정하면 Alice와 Bob의 작업은 달성됩니다. 그러나 텔레포테이션의 맥락에서는 Alice가 큐비트 Q를 물리적으로 Bob에게 보낼 수 없다고 가정합니다. 대신, Alice는 Bob에게 고전적 정보를 보낼 수 있습니다.

  이러한 가정은 다양한 상황에서 합리적입니다. 예를 들어, Alice가 Bob의 정확한 위치를 알지 못하거나 두 사람 사이의 거리가 멀다면, 오늘날 또는 가까운 미래의 기술로 큐비트를 물리적으로 보내는 것은 어려운 일일 것입니다. 하지만 우리가 일상적으로 경험하듯이, 이러한 상황에서도 고전적 정보의 전송은 매우 간단합니다.

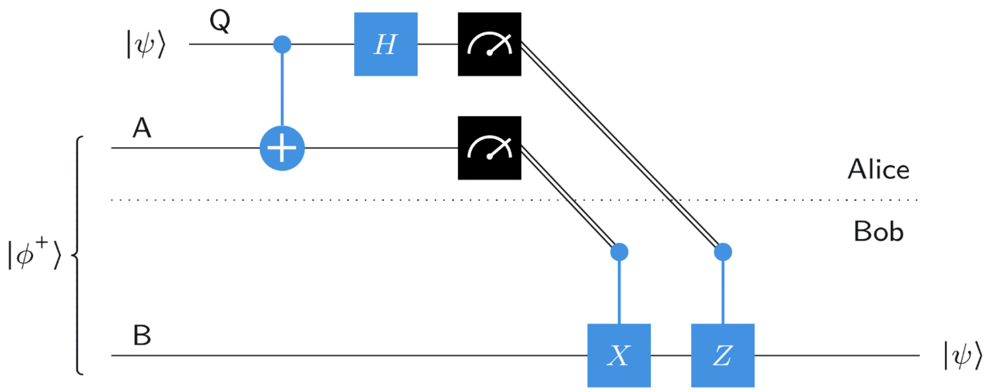
  이 시점에서 Alice와 Bob이 공유된 e-bit 없이 작업을 수행할 수 있는지, 즉 고전적 통신만을 사용하여 큐비트를 전송할 수 있는지 묻고자 할 수 있습니다. 이에 대한 대답은 "불가능하다"입니다. 고전적 통신만으로 양자 정보를 전송하는 것은 불가능합니다. 이는 이 시리즈의 세 번째 단원에서 다룰 기본적인 양자 정보 이론을 사용하여 증명하는 것이 그리 어렵지 않지만, 지금은 노-클로닝 정리를 통해 직관적으로 이 가능성을 배제할 수 있습니다.

  고전적 통신만으로 양자 정보를 보낼 수 있는 방법이 있다고 가정해 보십시오. 고전적 정보는 쉽게 복사되고 방송될 수 있으므로, Alice가 Bob에게 보내는 모든 고전적 전송은 두 번째 수신자(예: Charlie)도 받을 가능성이 있습니다. 하지만 Charlie가 Bob과 동일한 고전적 통신을 받는다면, Charlie도 큐비트 Q의 사본을 얻을 수 있지 않을까요? 이는 Q가 복제되었다는 것을 암시하며, 이는 이미 노-클로닝 정리로 불가능하다고 알고 있습니다. 따라서 고전적 통신만으로는 양자 정보를 보낼 수 없다고 결론지을 수 있습니다.

  그러나 Alice와 Bob이 e-bit를 공유한다는 가정이 있으면, 그들의 작업을 수행하는 것이 가능합니다. 바로 이 점이 양자 텔레포테이션 프로토콜이 수행하는 것입니다.

  프로토콜

다음은 텔레포테이션 프로토콜을 설명하는 양자 회로도입니다:



텔레포테이션 회로 :

이 회로도는 Alice와 Bob 간의 분리를 나타내기 위해 약간 스타일화되었으며, Alice에서 Bob으로 보내지는 두 개의 대각선 선이 고전적 비트를 나타내지만, 그 외에는 일반적인 양자 회로도입니다.

단어로 설명하면, 텔레포테이션 프로토콜은 다음과 같습니다:

1. Alice는 (A,Q) 쌍에 대해 제어-NOT 연산(CNOT)을 수행합니다. 여기서 Q는 제어 비트이고 A는 타겟 비트입니다. 그런 다음 Q에 대해 하다마드(Hadamard) 연산을 수행합니다.
2. Alice는 A와 Q를 각각 표준 기저 측정(standard basis measurement)을 사용하여 측정하고, 그 측정 결과를 고전적 비트 형태로 Bob에게 전송합니다. A의 측정 결과를 a, Q의 측정 결과를 b라고 합니다.
3. Bob은 Alice로부터 a와 b를 받고, 이 비트들의 값에 따라 다음과 같은 연산을 수행합니다:
   - a=1인 경우, Bob은 자신의 큐비트 B에 대해 비트 플립(bit flip, 또는 X 게이트)을 수행합니다.
   - b=1인 경우, Bob은 자신의 큐비트 B에 대해 위상 플립(phase flip, 또는 Z 게이트)을 수행합니다. 즉, ab가 00, 01, 10, 11 중 하나일 때, Bob은 각각 I(아무것도 하지 않음), Z, X, ZX 연산 중 하나를 B에 수행합니다.

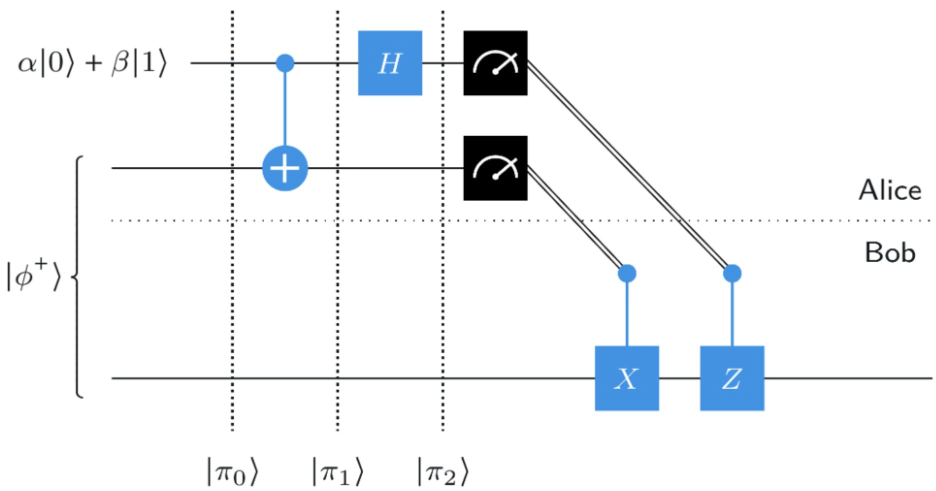
이것이 텔레포테이션 프로토콜에 대한 완전한 설명입니다. 아래에 나오는 분석에 따르면, 프로토콜이 실행되었을 때, 큐비트 B는 프로토콜 실행 전에 Q가 가졌던 상태, 그리고 Q가 다른 시스템과 가졌던 모든 상관관계까지 포함하여 동일한 상태에 있게 됩니다. 즉, 프로토콜은 Q의 상태를 B로 "텔레포트"하는 완벽한 큐비트 통신 채널을 효과적으로 구현한 것입니다.

분석으로 넘어가기 전에, 이 프로토콜이 Q의 상태를 복제하는 데 성공하지 않는다는 점에 주목하십시오. 이는 노-클로닝 정리에 의해 불가능하다는 것을 이미 알고 있습니다. 프로토콜이 끝났을 때 Q의 상태는 Alice가 수행한 측정 결과로 인해 원래 값에서 $|b\rangle$로 바뀌게 됩니다. 또한 e-bit가 프로세스에서 사실상 "소모되었다"는 점도 주목하십시오. A의 상태는 $|a\rangle$로 바뀌며, 더 이상 B 또는 다른 어떤 시스템과도 얽혀 있지 않습니다. 이것이 텔레포테이션의 비용입니다.

분석

텔레포테이션 프로토콜을 분석하기 위해, 위에서 설명한 회로의 동작을 단계별로 살펴보겠습니다. 시작점은 Q가 초기 상태 $\alpha|0\rangle+\beta|1\rangle$에 있을 때입니다. 이는 Q가 다른 시스템과 얽혀 있을 가능성을 다루지는 않으므로 가장 일반적인 상황은 아니지만, 이 간단한 경우부터 시작하면 분석이 더 명확해집니다. 보다 일반적인 경우는 이 단순한 사례에 대한 분석 후에 다룰 것입니다.

구체적으로, 아래 그림에서 제시된 시간에 따라 큐비트 (B,A,Q)의 상태를 고려할 것입니다:



큐비트 Q가 프로토콜의 시작 시 상태 α ∣ 0⟩+β ∣ 1⟩에 있다고 가정하면, 세 큐비트 (B,A,Q)의 상태는 다음과 같습니다:

$$|\pi_0\rangle = |\phi^+\rangle \otimes \big(\alpha|0\rangle + \beta|1\rangle\big) = \frac{\alpha|000\rangle + \alpha|110\rangle + \beta|001\rangle + \beta|111\rangle}{\sqrt{2}}.$$

첫 번째로 수행되는 게이트는 Controlled-NOT 게이트이며, 이는 상태 ∣π0⟩를 다음과 같이 변환합니다:

$$|\pi_1\rangle = \frac{\alpha|000\rangle + \alpha|110\rangle + \beta|011\rangle + \beta|101\rangle}{\sqrt{2}}.$$

그 다음으로 하다마드 게이트를 적용하면, 상태 ∣π1⟩는 다음과 같이 변환됩니다:

$$\begin{aligned}|\pi_2\rangle &= \frac{\alpha|00\rangle|+\rangle + \alpha|11\rangle|+\rangle + \beta|01\rangle|-\rangle + \beta|10\rangle|-\rangle}{\sqrt{2}} \\ &= \frac{\alpha|000\rangle + \alpha|001\rangle + \alpha|110\rangle + \alpha|111\rangle + \beta|010\rangle - \beta|011\rangle + \beta|100\rangle - \beta|1}{2}\end{aligned}$$

텐서곱의 다중선형성을 사용하면 이 상태를 다른 형태로 작성할 수 있습니다.

$$
\begin{aligned}
|\pi_2\rangle = \quad & \frac{1}{2}\big(\alpha|0\rangle + \beta|1\rangle\big)|00\rangle \\
& + \frac{1}{2}\big(\alpha|0\rangle - \beta|1\rangle\big)|01\rangle \\
& + \frac{1}{2}\big(\alpha|1\rangle + \beta|0\rangle\big)|10\rangle \\
& + \frac{1}{2}\big(\alpha|1\rangle - \beta|0\rangle\big)|11\rangle.
\end{aligned}
$$

처음 보기에는 마치 마법 같은 일이 일어난 것처럼 보일 수 있습니다. 왜냐하면, 가장 왼쪽의 큐비트 BB가 아직 앨리스로부터 밥에게 어떠한 정보도 전달되지 않았음에도 불구하고, α\alpha와 β\beta에 의존하는 것처럼 보이기 때문입니다. 그러나 이것은 사실 착각에 불과합니다. 스칼라 값은 텐서곱 안에서 자유롭게 이동할 수 있으므로, α\alpha와 β\beta는 가장 왼쪽 큐비트와 다른 큐비트들 중 어느 것과도 특별히 더 많이, 혹은 더 적게 관련되어 있지 않습니다. 우리가 한 일은 단지 상태를 측정 분석을 용이하게 하기 위해 대수적으로 표현한 것뿐입니다.

이제 앨리스의 표준 기저 측정의 네 가지 가능한 결과와, 그 결과로 밥이 수행하는 동작을 고려해보겠습니다.

가능한 결과 :

앨리스의 측정 결과가 ab=00인 경우, 측정 결과의 확률은 다음과 같습니다:

$$
\left\| \frac{1}{2}\big(\alpha|0\rangle + \beta|1\rangle\big) \right\|^2 = \frac{|\alpha|^2 + |\beta|^2}{4} = \frac{1}{4},
$$

이 경우, (B,A,Q)의 상태는 다음과 같이 됩니다:

$$
\big(\alpha|0\rangle + \beta|1\rangle\big)|00\rangle.
$$

이때, 밥은 아무 동작도 하지 않으며, 이 상태가 세 큐비트의 최종 상태가 됩니다.

앨리스의 측정 결과가 ab=01인 경우, 측정 결과의 확률은 다음과 같습니다:

$$
\left\| \frac{1}{2}\big(\alpha|0\rangle - \beta|1\rangle\big) \right\|^2 = \frac{|\alpha|^2 + |-\beta|^2}{4} = \frac{1}{4},
$$

이 경우, (B,A,Q)의 상태는 다음과 같이 됩니다:

$$\big(\alpha|0\rangle - \beta|1\rangle\big)|01\rangle.$$

이때, 밥은 Z 게이트를 B 큐비트에 적용하며, (B,A,Q)의 상태는 다음과 같이 변경됩니다:

$$\big(\alpha|0\rangle + \beta|1\rangle\big)|01\rangle.$$

앨리스의 측정 결과가 ab=10인 경우, 측정 결과의 확률은 다음과 같습니다:

$$\left\|\frac{1}{2}\big(\alpha|1\rangle + \beta|0\rangle\big)\right\|^2 = \frac{|\alpha|^2 + |\beta|^2}{4} = \frac{1}{4},$$

이 경우, (B,A,Q)의 상태는 다음과 같이 됩니다:

$$\big(\alpha|1\rangle + \beta|0\rangle\big)|10\rangle.$$

이때, 밥은 X 게이트를 B 큐비트에 적용하며, (B,A,Q)의 상태는 다음과 같이 변경됩니다:

$$\big(\alpha|0\rangle + \beta|1\rangle\big)|10\rangle.$$

앨리스의 측정 결과가 ab=11인 경우, 측정 결과의 확률은 다음과 같습니다:

$$\left\|\frac{1}{2}\big(\alpha|1\rangle - \beta|0\rangle\big)\right\|^2 = \frac{|\alpha|^2 + |-\beta|^2}{4} = \frac{1}{4},$$

이 경우, (B,A,Q)의 상태는 다음과 같이 됩니다:

$$\big(\alpha|1\rangle - \beta|0\rangle\big)|11\rangle.$$

이때, 밥은 ZX 연산을 B 큐비트에 수행하며, (B,A,Q)의 상태는 다음과 같이 변경됩니다:

$$\big(\alpha|0\rangle + \beta|1\rangle\big)|11\rangle.$$

이제 우리는 네 가지 모든 경우에서 밥의 큐비트 B가 프로토콜 종료 시 α∣0⟩+β∣1⟩ 상태에 놓이게 된다는 것을 확인했습니다. 이는 초기 큐비트 Q의 상태와 동일합니다.

이로써 우리가 입증하려고 했던 바를 보여줍니다: 텔레포테이션 프로토콜은 이 상황에서 올바르게 작동했습니다.

또한, 큐비트 A와 Q는 앨리스의 측정 결과에 따라 ∣00⟩, ∣01⟩, ∣10⟩, ∣11⟩ 중 하나의 상태에 확률 1/4로 놓이게 됩니다.

이는 앞서 언급했던 대로, 프로토콜 종료 시 앨리스는 더 이상 α∣0⟩+β∣1⟩ 상태를 가지고 있지 않음을 보여줍니다. 이는 복제 불가능성 정리(no-cloning theorem)에 의해 예상된 결과입니다.

또한 앨리스의 측정은 α∣0⟩+β∣1⟩ 상태에 대한 어떤 정보도 제공하지 않는다는 점에 주목하십시오.

네 가지 가능한 측정 결과 각각의 확률은 1/4로 동일하며, 이는 α와 β의 값에 관계없이 동일합니다. 이는 텔레포테이션이 올바르게 작동하기 위해 필수적입니다: 알 수 없는 양자 상태로부터 정보를 추출하면 일반적으로 상태가 방해를 받게 되지만, 이 경우 밥은 상태를 방해받지 않고 얻습니다.

이제, 초기 큐비트 Q가 또 다른 시스템 R과 얽혀(entangled) 있는 더 일반적인 상황을 고려해 봅시다.

위와 유사한 분석을 통해, 텔레포테이션 프로토콜은 이러한 더 일반적인 상황에서도 올바르게 작동한다는 것을 알 수 있습니다:

프로토콜 종료 시 밥이 가진 큐비트 B는 초기 상태에서 Q가 R과 얽혀 있던 방식과 동일하게 R과 얽히게 됩니다. 마치 앨리스가 Q를 밥에게 직접 전달한 것과 같습니다.

이를 증명하기 위해, 다음과 같이 가정합시다:

쌍 (Q,R)의 상태는 초기적으로 다음 형태의 양자 상태 벡터로 주어져 있습니다:

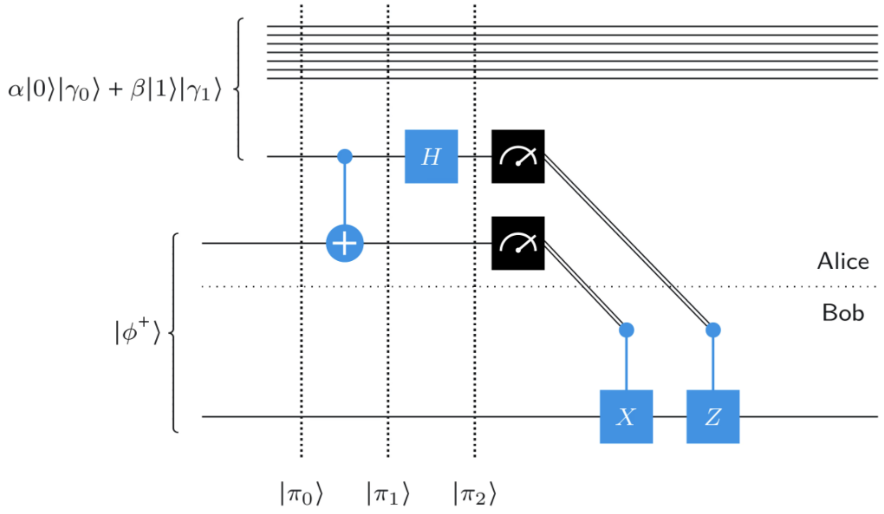
$$\alpha|0\rangle_{\mathsf{Q}}|\gamma_0\rangle_{\mathsf{R}} + \beta|1\rangle_{\mathsf{Q}}|\gamma_1\rangle_{\mathsf{R}},$$

여기서 ∣γ0⟩와 ∣γ1⟩는 단위 벡터이며,

α와 β는 복소수로 ∣α∣^2+∣β∣^2=1을 만족합니다.

쌍 (Q,R)의 모든 양자 상태 벡터는 위와 같은 방식으로 표현될 수 있습니다.

다음 그림은 이전과 동일한 회로를 나타내며, 시스템 R이 추가된 것을 보여줍니다 (위쪽에 위치한 여러 큐비트로 표시되며, 이는 아무런 영향을 받지 않습니다).



텔레포테이션 프로토콜을 실행할 때 일어나는 일을 분석하기 위해, 시스템의 순서를 재배치하는 것이 유용합니다. 이는 Lesson 2에서 설명된 방식과 유사합니다. 구체적으로, 시스템의 순서를 (B,R,A,Q)로 고려할 것입니다(원래 순서는 (B,A,Q,R)였습니다). 이후 식에서는 명확성을 위해 시스템 이름을 첨자로 표기합니다.

프로토콜이 시작될 때, 시스템의 상태는 다음과 같습니다:

$$
\begin{aligned}
|\pi_0\rangle &= |\phi^+\rangle_{\mathsf{BA}} \otimes \big(\alpha|0\rangle_{\mathsf{Q}}|\gamma_0\rangle_{\mathsf{R}} + \beta|1\rangle_{\mathsf{Q}}|\gamma_1\rangle_{\mathsf{R}}\big) \\
&= \frac{\alpha|0\rangle_{\mathsf{B}}|\gamma_0\rangle_{\mathsf{R}}|00\rangle_{\mathsf{AQ}} + \alpha|1\rangle_{\mathsf{B}}|\gamma_0\rangle_{\mathsf{R}}|10\rangle_{\mathsf{AQ}} + \beta|0\rangle_{\mathsf{B}}|\gamma_1\rangle_{\mathsf{R}}|01\rangle_{\mathsf{AQ}} + \beta|1\rangle_{\mathsf{B}}|\gamma_1\rangle_{\mathsf{R}}|}{\sqrt{2}}
\end{aligned}
$$

먼저 CNOT 게이트가 적용됩니다. 이 작업은 상태를 다음과 같이 변환합니다:

$$
|\pi_1\rangle = \frac{\alpha|0\rangle_{\mathsf{B}}|\gamma_0\rangle_{\mathsf{R}}|00\rangle_{\mathsf{AQ}} + \alpha|1\rangle_{\mathsf{B}}|\gamma_0\rangle_{\mathsf{R}}|10\rangle_{\mathsf{AQ}} + \beta|0\rangle_{\mathsf{B}}|\gamma_1\rangle_{\mathsf{R}}|11\rangle_{\mathsf{AQ}} + \beta|1\rangle_{\mathsf{B}}|\gamma_1\rangle_{\mathsf{R}}|}{\sqrt{2}}
$$

그다음 Hadamard 게이트가 적용됩니다. 결과 상태를 확장하고 단순화하면 다음과 같은 상태를 얻습니다:

$$
\begin{aligned}
|\pi_2\rangle = \quad & \frac{1}{2}\big(\alpha|0\rangle_{\mathsf{B}}|\gamma_0\rangle_{\mathsf{R}} + \beta|1\rangle_{\mathsf{B}}|\gamma_1\rangle_{\mathsf{R}}\big)|00\rangle_{\mathsf{AQ}} \\
& + \frac{1}{2}\big(\alpha|0\rangle_{\mathsf{B}}|\gamma_0\rangle_{\mathsf{R}} - \beta|1\rangle_{\mathsf{B}}|\gamma_1\rangle_{\mathsf{R}}\big)|01\rangle_{\mathsf{AQ}} \\
& + \frac{1}{2}\big(\alpha|1\rangle_{\mathsf{B}}|\gamma_0\rangle_{\mathsf{R}} + \beta|0\rangle_{\mathsf{B}}|\gamma_1\rangle_{\mathsf{R}}\big)|10\rangle_{\mathsf{AQ}} \\
& + \frac{1}{2}\big(\alpha|1\rangle_{\mathsf{B}}|\gamma_0\rangle_{\mathsf{R}} - \beta|0\rangle_{\mathsf{B}}|\gamma_1\rangle_{\mathsf{R}}\big)|11\rangle_{\mathsf{AQ}}.
\end{aligned}
$$

앞서 단순한 사례를 분석한 것과 동일한 방식으로, 알리스의 측정 결과 4가지 경우에 따른 밥의 행동을 고려하면, 프로토콜이 끝날 때 (B,R) 시스템의 상태는 항상 다음과 같음을 알 수 있습니다:

$$\alpha|0\rangle|\gamma_0\rangle + \beta|1\rangle|\gamma_1\rangle.$$

간단한 사례와 비교했을 때, 분석은 크게 달라지지 않습니다. $|\gamma0\rangle$와 $|\gamma1\rangle$는 단순히 "함께 이동"할 뿐입니다. 따라서, 텔레포테이션은 완벽한 양자 통신 채널을 생성하여, 큐비트 Q의 내용을 B로 전송하고 다른 시스템과의 모든 상관관계를 유지합니다.

간단한 경우의 분석에서 드러났듯이, 이 과정은 큐비트에 대해 항등 연산(identity operation)처럼 작동하며, 이는 프로토콜이 노이즈 없는 완벽한 양자 채널을 구현한다는 것을 의미합니다. 큐비트가 다른 시스템과 얽혀 있는 경우에도 이 프로토콜이 올바르게 작동할 수 있음을 보장합니다.

추가 논의

다음은 텔레포테이션에 대한 몇 가지 간략한 결론적 의견입니다.

첫째, 텔레포테이션은 양자 정보의 응용이 아니라 양자 통신을 수행하기 위한 프로토콜임을 이해해야 합니다. 이것은 양자 통신이 유용한 만큼만 유용합니다.

둘째, 텔레포테이션이 언젠가 양자 정보를 전달하는 표준적인 방법이 될 수 있다는 추측은 타당합니다. 예를 들어, 이는 얽힘 증류(entanglement distillation)로 알려진 과정을 통해 이루어질 수 있습니다. 얽힘 증류는 다수의 잡음이 섞인(또는 불완전한) e-비트를 소수의 고품질 e-비트로 변환하는 과정으로, 이를 통해 잡음이 없거나 거의 없는 텔레포테이션을 수행할 수 있습니다. 이 아이디어의 핵심은, 얽힘 증류 과정이 직접적인 양자 통신만큼 섬세하지 않다는 점입니다. 예를 들어, 손실을 허용할 수 있고, 과정이 실패하면 단순히 다시 시도하면 됩니다. 반면, 우리가 실제로 전달하려는 큐비트는 훨씬 더 소중할 수 있습니다.

마지막으로, 텔레포테이션의 아이디어와 작동 방식은 양자 정보와 계산에서 매우 근본적인 개념입니다. 이는 양자 정보 이론의 초석이며, 그 아이디어의 변형도 자주 등장합니다. 한 가지 예로, 텔레포테이션을 사용하여 양자 게이트를 구현할 수 있습니다. 이 경우, 특정 초기 상태와 측정이 선택되어 결과적으로 특정 연산을 적용하는 데 사용됩니다. 이는 단순히 정보를 전달하는 것과는 다른 방식입니다.

- Qiskit 구현

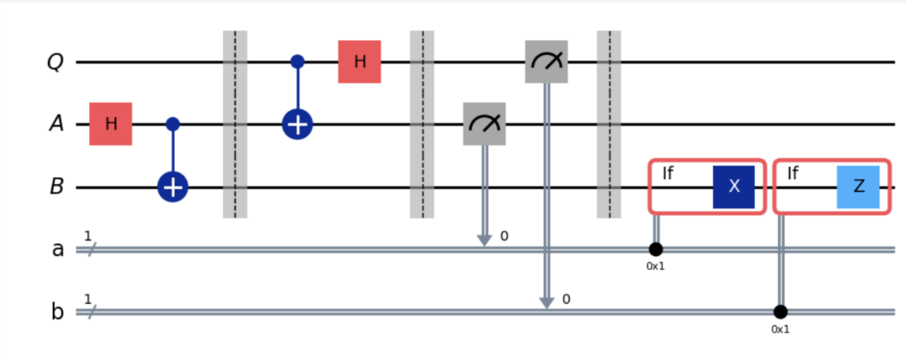
```
# Required imports

from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister
from qiskit_aer import AerSimulator
from qiskit.visualization import plot_histogram
from qiskit.result import marginal_distribution
from qiskit.circuit.library import UGate
from numpy import pi, random
```

여기 텔레포테이션 프로토콜의 양자 회로 구현이 있습니다.

```
qubit = QuantumRegister(1, "Q")
ebit0 = QuantumRegister(1, "A")
ebit1 = QuantumRegister(1, "B")
a = ClassicalRegister(1, "a")
b = ClassicalRegister(1, "b")

protocol = QuantumCircuit(qubit, ebit0, ebit1, a, b)

# Prepare ebit used for teleportation
```

```
protocol.h(ebit0)
protocol.cx(ebit0, ebit1)
protocol.barrier()

# Alice's operations
protocol.cx(qubit, ebit0)
protocol.h(qubit)
protocol.barrier()

# Alice measures and sends classical bits to Bob
protocol.measure(ebit0, a)
protocol.measure(qubit, b)
protocol.barrier()

# Bob uses the classical bits to conditionally apply gates
with protocol.if_test((a, 1)):
    protocol.x(ebit1)
with protocol.if_test((b, 1)):
    protocol.z(ebit1)

display(protocol.draw())
```

Output:



회로는 Qiskit에서 이전 수업에서 본 적이 없는 몇 가지 기능을 사용합니다. 그 중에는 `barrier`와 `if_test` 함수가 포함됩니다. `barrier` 함수는 회로 다이어그램을 더 읽기 쉽게 만들기 위해 시각적인 구분을 생성하며, 회로가 실제 하드웨어에서 실행될 때 컴파일 중에 다양한 단순화 및 최적화를 방지합니다. `if_test` 함수는 고전 비트나 레지스터에 따라 조건적으로 연산을 적용합니다.

회로는 먼저 (A,B)를 $|\phi^+\rangle$ 상태로 초기화한 후 (이는 프로토콜 자체의 일부가 아님), 이어서 Alice의 연산, 측정, 그리고 마지막으로 Bob의 연산이 진행됩니다.

프로토콜이 제대로 작동하는지 테스트하기 위해, 우리는 무작위로 생성된 단일 큐비트 게이트를 초기화된 $|0\rangle$ 상태에 적용하여 텔레포트될 무작위 양자 상태 벡터를 얻습니다. 그런 다음 프로토콜이 실행된 후, 그 게이트의 역(즉, 켤레 전치)을 B에 적용하여 상태가 $|0\rangle$ 상태로 돌아왔는지 측정하여 텔레포트가 제대로 이루어졌는지 확인할 수 있습니다.

먼저 무작위 유니타리 큐비트 게이트를 선택합니다.

```
random_gate = UGate(
    theta=random.random() * 2 * pi,
    phi=random.random() * 2 * pi,
    lam=random.random() * 2 * pi,
)

display(random_gate.to_matrix())
```

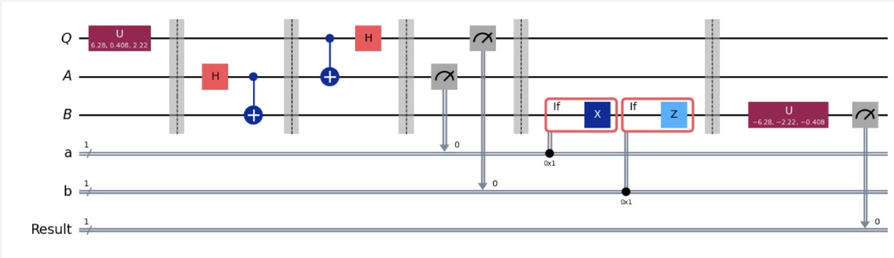
Output:

```
array([[-9.99999078e-01+0.j        ,  8.19563447e-04-0.00108289j],
       [ 1.24668618e-03+0.00053861j,  8.70230844e-01-0.4926423j ]])
```

이제 우리는 새로운 테스트 회로를 만들겠습니다. 이 회로는 먼저 무작위 게이트를 Q에 적용한 후, 텔레포테이션 회로를 실행하고, 마지막으로 무작위 게이트의 역을 큐비트 B에 적용한 뒤 측정합니다. 결과는 확실히 0이어야 합니다.

```
# Create a new circuit including the same bits and qubits used in the
# teleportation protocol.

test = QuantumCircuit(qubit, ebit0, ebit1, a, b)

# Start with the randomly selected gate on Q

test.append(random_gate, qubit)
test.barrier()

# Append the entire teleportation protocol from above.

test = test.compose(protocol)
test.barrier()

# Finally, apply the inverse of the random unitary to B and measure.

test.append(random_gate.inverse(), ebit1)

result = ClassicalRegister(1, "Result")
test.add_register(result)
test.measure(ebit1, result)

display(test.draw())
```

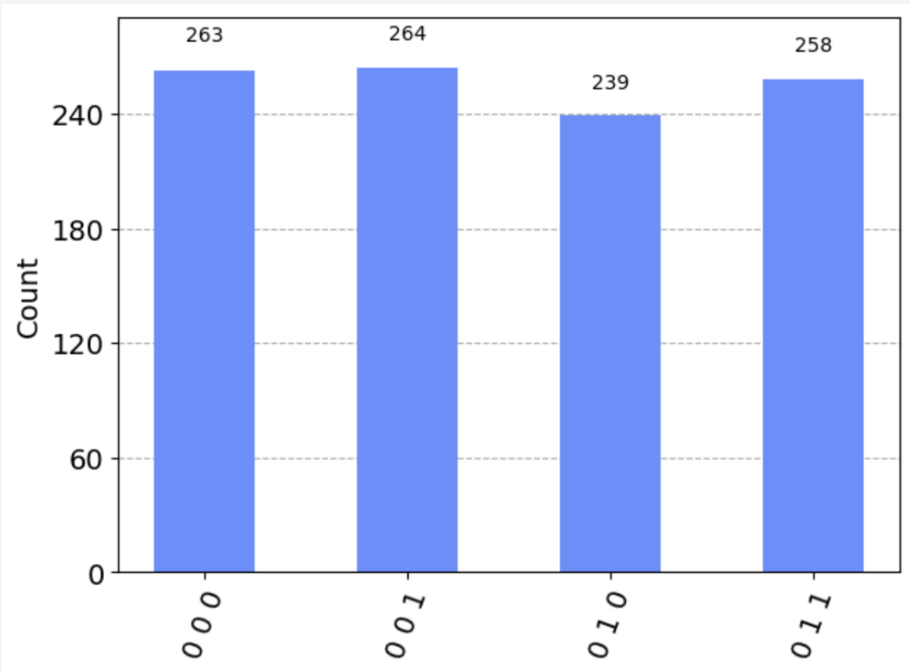
Output:



마지막으로 이 회로에서 Aer 시뮬레이터를 실행하고 출력의 히스토그램을 그려봅시다. 세 개의 고전 비트에 대한 통계를 확인할 수 있습니다. 맨 아래/왼쪽 비트는 항상 0이어야 하며, 이는 큐비트 Q가 성공적으로 B로 텔레포트되었음을 나타냅니다. 나머지 두 비트는 대체로 균등 분포일 것입니다.

```
result = AerSimulator().run(test).result()
statistics = result.get_counts()
display(plot_histogram(statistics))
```

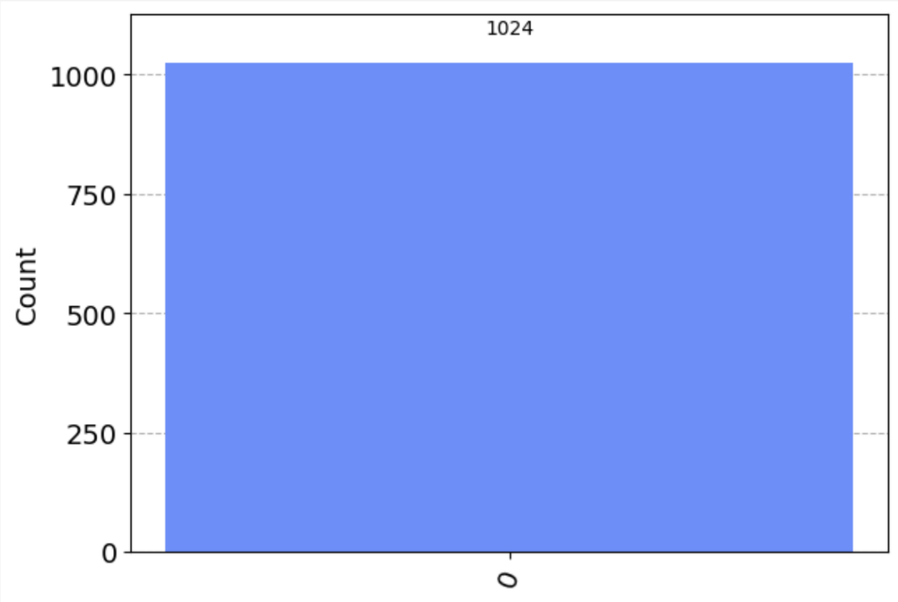
Output:



원하는 경우, 다음과 같이 통계를 필터링하여 테스트 결과 큐비트에만 집중할 수 있습니다.

```
filtered_statistics = marginal_distribution(statistics, [2])
display(plot_histogram(filtered_statistics))
```

Output:



정리

양자 텔레포테이션은 Alice가 Bob에게 큐비트를 전송하는 프로토콜로, 얽힌 상태와 고전적 통신을 사용합니다. 이 프로세스에서 중요한 점은 큐비트를 물리적으로 전송하지 않고도 큐비트의 양자 상태를 전송할 수 있다는 것입니다.

- **텔레포테이션 프로토콜 :**
  1. **공유된 얽힘 상태:** Alice와 Bob은 미리 얽힌 큐비트 쌍을 공유하고 있습니다.
  2. **큐비트 Q:** Alice는 전송하려는 큐비트 Q를 가지고 있습니다. 큐비트 Q는 초기 상태 $\alpha|0\rangle+\beta|1\rangle$에 있을 수 있습니다.
  3. **CNOT 연산:** Alice는 Q와 A에 대해 CNOT 연산을 수행합니다.
  4. **하다마드(Hadamard) 연산:** Q에 대해 하다마드 연산을 수행합니다.
  5. **측정:** Alice는 A와 Q를 각각 측정하고 그 결과(a, b)를 Bob에게 고전적 비트로 전송합니다.
  6. **Bob의 연산:** Bob은 a와 b의 값을 바탕으로 자신의 큐비트 B에 대해 적절한 연산을 수행합니다. 이 연산은 `x`, `z`, 또는 `xz` 연산이 될 수 있습니다.

  이 과정이 끝나면, 큐비트 B는 원래 Q의 상태를 그대로 "텔레포트"한 것처럼 상태를 가지게 됩니다.
- **분석 :**
  - 텔레포테이션은 큐비트의 상태를 복제하는 것이 아니라, 상태를 정확히 전송하는 것입니다.
  - 노-클로닝 정리 때문에, 큐비트 Q의 상태는 복제되지 않고, 대신 Alice와 Bob이 협력하여 상태를 전송합니다.
  - 텔레포테이션의 핵심은 얽힌 큐비트와 고전적 통신을 사용해 큐비트 상태를 전송하는 방식에 있습니다.
- **추가 논의 :**

  양자 텔레포테이션은 양자 컴퓨팅에서 중요한 개념으로, 다음과 같은 특성을 가집니다:

1. **양자 얽힘**: 양자 얽힘은 두 큐비트가 서로 강하게 연결되어 있어, 하나의 큐비트를 측정하면 다른 큐비트의 상태도 즉시 알 수 있는 현상입니다. 텔레포테이션은 얽힘 상태를 활용하여 큐비트의 상태를 전송합니다.
2. **고전적 통신의 필요성**: 양자 얽힘을 이용하더라도 큐비트의 상태를 정확하게 전달하기 위해서는 Alice와 Bob 간의 고전적 통신이 필요합니다. 즉, Alice는 큐비트를 측정하고, 그 결과를 Bob에게 전송하여 Bob이 적절한 연산을 할 수 있게 해야 합니다.
3. **비가역성**: 텔레포테이션 과정에서는 Alice의 큐비트 상태가 전송되면서 동시에 사라지게 됩니다. 이때 큐비트의 상태는 복제되지 않고, 그 상태를 Bob이 복제하는 형태로 바뀌므로, 큐비트의 상태가 소멸되는 비가역적인 과정이 포함됩니다.

- **텔레포테이션 구현 코드 예시 (Qiskit 사용) :**

```
from qiskit import QuantumCircuit, Aer, execute

# 1. 양자 회로 만들기 (2개의 큐비트, 2개의 고전 비트)
qc = QuantumCircuit(3, 2)

# 2. 얽힘 생성: 첫 번째 큐비트와 두 번째 큐비트 얽힘
qc.h(0)  # Hadamard 게이트 (큐비트 0에 적용)
qc.cx(0, 1)  # CNOT 게이트 (큐비트 0에서 큐비트 1로)

# 3. 큐비트 Q와 얽힌 큐비트 A에 대해 CNOT 및 Hadamard 연산 수행
qc.cx(0, 2)  # 큐비트 2에 대해 CNOT
qc.h(0)  # 큐비트 0에 대해 Hadamard 연산

# 4. 측정 단계: Alice는 큐비트 0과 큐비트 2를 측정
qc.measure([0, 2], [0, 1])

# 5. 고전적 통신 후 Bob은 큐비트 1에 대해 연산을 수행
qc.cx(1, 2)  # X 연산
qc.cz(0, 2)  # Z 연산

# 6. 결과 확인
qc.measure(2, 2)  # 큐비트 2를 측정하여 최종 상태 출력

# 7. 시뮬레이션 실행
simulator = Aer.get_backend('qasm_simulator')
result = execute(qc, simulator, shots=1024).result()
counts = result.get_counts(qc)

# 결과 출력
print(counts)
```

1. **초기 얽힘 상태 생성**: 첫 번째 큐비트와 두 번째 큐비트 사이에 얽힘을 생성합니다.
2. **큐비트 Q와 얽힌 큐비트 A에 대해 연산**: 큐비트 Q(여기서는 큐비트 0)와 큐비트 A(여기서는 큐비트 2)에 CNOT 및 Hadamard 연산을 적용하여 Alice가 측정할 준비를 합니다.
3. **측정과 고전적 통신**: Alice는 큐비트 0과 큐비트 2를 측정하고 그 결과를 Bob에게 전송합니다.

4. **Bob의 연산**: Bob은 고전적 통신을 통해 받은 정보를 바탕으로 큐비트 1에 대해 `cx` 및 `cz` 연산을 수행하여 원래 큐비트 Q의 상태를 복원합니다.
5. **결과 측정**: 마지막으로 큐비트 2의 상태를 측정하여 최종적인 상태를 확인합니다.

이 코드 예시는 Qiskit을 사용하여 양자 텔레포테이션의 핵심 프로세스를 시뮬레이션하는 것입니다. 이 과정에서 큐비트의 상태는 원본 큐비트에서 Bob의 큐비트로 정확히 텔레포트됩니다.