

제 1 교시

국어 영역

출수형

[1~4] 다음 글을 읽고 물음에 답하시오.

현대에 일반적으로 사용되고 있는 고전 컴퓨터의 한계를 극복할 수 있는 대안으로 양자 컴퓨터가 제시된다. 고전 컴퓨터가 하나의 비트로 0 또는 1 중 단 하나의 값만을 처리하는 반면, 양자 컴퓨터는 큐비트라는 기본 단위를 통해 0과 1의 상태가 동시에 존재하는 중첩을 구현한다. 이러한 중첩 상태를 정밀하게 제어하고 원하는 결과를 얻어내기 위해, 양자 컴퓨터는 고유한 연산 방식을 사용한다. 양자 전산의 모든 연산은 가장 기본적인 두 가지 연산, 즉 회전 연산과 CNOT 연산의 조합으로 수행될 수 있다는 사실이 증명되어 있다.

회전 연산은 단일 큐비트의 상태를 변화시키는 가장 기본적인 조작이다. 큐비트의 상태는 전자의 스핀과 같은 물리적 속성을 통해 구현되는데, 이는 스핀 업과 스핀 다운이라는 두 고유 상태로 나타낼 수 있다. 이 두 상태를 각각 1과 0에 대응시킨다고 할 때, 스핀 업 상태의 큐비트에 180도 회전이라는 물리적 조작을 가하면 스핀 다운 상태로 바뀐다. 이는 입력값 1을 출력값 0으로 바꾸는 것이므로, 고전 컴퓨터의 NOT 연산과 동일한 결과를 낳는다. 그러나 회전 연산의 진정한 힘은 임의의 각도로 회전이 가능하다는 데 있다. 예를 들어, 스핀 업 상태의 큐비트를 90도만 회전시키면, 큐비트는 업도 다운도 아닌, 두 상태의 특성이 절반씩 섞인 중첩 상태가 된다. 이처럼 회전 각도를 조절함으로써, 0과 1의 중첩 비율을 원하는 대로 제어하는 것이 가능하다.

회전 연산이 단일 큐비트를 제어하는 방식이라면, CNOT 연산은 둘 이상의 큐비트들 사이의 상호작용을 제어하는 방식이다. 조건부 NOT 연산이라고도 불리는 이 연산은, 두 개의 큐비트를 입력받아 그중 하나의 상태를 조건으로 삼아 다른 하나의 상태를 변화시킨다. 이때 조건으로 사용되는 첫 번째 큐비트를 제어 비트라 하고, 상태가 변하는 두 번째 큐비트를 표적 비트라고 한다. CNOT 연산의 규칙은 명확하다. 제어 비트가 0이면 표적 비트는 아무런 변화 없이 그대로 통과하지만, 제어 비트가 1이면 표적 비트는 0에서 1로, 1에서 0으로 상태가 뒤바뀌는 NOT 연산이 수행된다. 이 CNOT 연산은 큐비트들을 서로 얽히게 만들어, 한 큐비트의 상태가 다른 큐비트의 상태에 즉각적으로 영향을 미치는 복잡한 양자 시스템을 구축하는데 핵심적인 역할을 한다.

결론적으로, 양자 컴퓨터는 회전 연산을 통해 각 큐비트의 개별적인 중첩 상태를 자유자재로 만들고, CNOT 연산을 통해 이 큐비트들을 논리적으로 연결하여 복잡한 연관 관계를 설정한다. 위 두 연산 방식을 사용하여 양자 컴퓨터는 고전 컴퓨터가 닿지 못하는 문제를 풀게 될 것이다.

1. 학생이 윗글을 읽은 방법으로 적절하지 않은 것은?
 - ① 1문단에서 '양자 컴퓨터'가 '고전 컴퓨터'의 대안임을 확인하고, 두 대상의 차이점을 중심으로 글이 전개될 것을 예측하며 읽는다.
 - ② 2문단과 3문단의 관계를 파악하기 위해, '단일 큐비트'와 '둘 이상의 큐비트'라는 서술에 주목하여 두 연산 방식의 작동 범위를 비교하며 읽는다.
 - ③ 3문단에서 '조건부 NOT 연산'이라는 명칭을 보고, '조건'과 'NOT 연산'이 각각 어떤 역할을 하는지 구체적인 작동 규칙에 집중하여 읽는다.
 - ④ 2문단에서 'NOT 연산'이 언급된 것을 바탕으로, 양자 컴퓨터의 연산 방식의 차이점을 이해한다.
 - ⑤ 1문단의 '중첩'이라는 핵심 개념이, 2문단의 '회전 연산'과 3문단의 'CNOT 연산'에서 어떻게 구체적으로 제어되는지를 연결하며 읽는다.

2. 윗글을 이해한 내용으로 가장 적절한 것은?
 - ① '회전 연산'은 입력 큐비트의 상태와 상관없이 '0'과 '1'의 중첩 상태를 출력한다.
 - ② 'CNOT 연산'에서 '제어 비트'가 '1'일 때, '표적 비트'의 최종 상태는 '제어 비트'의 상태와 동일해진다.
 - ③ 고전 컴퓨터와 달리 양자 컴퓨터는 정보를 처리할 때 이진법을 사용하지 않는다.
 - ④ 회전 연산은 각도 조절을 통해 중첩 비율을 다르게 하여 큐비트를 서로 얽히도록 한다.
 - ⑤ 큐비트가 '0'과 '1'의 중첩 상태를 구현하는 것은, 스핀의 두 고유 상태를 특정 비율로 섞는 물리적 조작을 통해 가능하다.

3. 중첩 상태에 대한 설명으로 적절하지 않은 것은?

- ① 스핀 업과 스핀 다운 어디에도 속하지 않는 상태이다.
- ② 회전 연산의 각도를 조절함으로써 그 상태의 특성이 변화될 수 있다.
- ③ 단일 큐비트뿐만 아니라, 둘 이상의 큐비트 시스템에서도 나타날 수 있다.
- ④ 고전 컴퓨터의 비트와 달리, 하나의 단위로 둘 이상의 정보를 동시에 표현하는 것을 가능하게 한다.
- ⑤ 중첩 상태는 CNOT 연산을 통해 큐비트들이 서로 얽힌 결과이다.

4. 밑글을 바탕으로 <보기>를 이해한 내용으로 적절하지 않은 것은? [3점]

< 보 기 >

어떤 양자 회로에 두 개의 큐비트,
A(제어 비트)와 B(표적 비트)가 있다.
A의 초기 상태는 '0과 1이 50:50으로 중첩된 상태'이고,
B의 초기 상태는 '0'이다.
이 두 큐비트가 CNOT 연산을 한 번 거쳤다고 하자.

- ① 회전 연산을 거치지 않은 초기 상태의 A는, NOT 연산을 거치지 못할 것이다.
- ② 만약 연산 결과를 측정했을 때 A가 '0'으로 확정된다면, B는 반드시 '0'으로 확정될 것이다.
- ③ 만약 연산 결과를 측정했을 때 A가 '1'으로 확정된다면, B는 반드시 '1'으로 확정될 것이다.
- ④ NOT 연산이 수행되기 위한 전제로서, A의 회전 각도는 180도일 것이다.
- ⑤ 이 회로를 통해, 우리는 최종적으로 '00' 또는 '11'이라는 두 가지 결과 조합 중 하나만을 얻게 될 것이다.