

◆ 14-6평 A형 19~21번

[19~21] 다음 글을 읽고 물음에 답하시오.

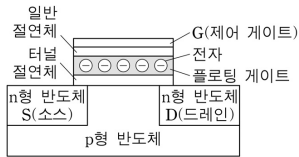
플래시 메모리는 수많은 스위치들로 이루어지는데, 각 스위치에 0 또는 1을 저장한다. 디지털 카메라에서 사진 한 장은 수백만 개 이상의 스위치를 켜고 끄는 방식으로 플래시 메모리에 저장된다. 메모리에서는 1비트의 정보를 기억하는 이 스위치를 셀이라고 한다. 플래시 메모리에서 셀은 그림과 같은 구조의 트랜지스터 1개로 이루어져 있다. 플로팅 게이트에 전자가 들어 있는 상태를 1, 들어 있지 않은 상태를 0이라고 정의한다.

플래시 메모리에서 데이터를 읽을 때는 그림의 반도체 D에 3V의 양(+)의 전압을 가한다. 그러면 다른 한 쪽의 반도체인 S로부터 전자들이 D 쪽으로 이끌리게 된다. 플로팅 게이트에 전자가 들어 있을 때는 S로부터 오는 전자와 플로팅 게이트에 있는 전자가 마치 자석의 같은 극처럼 서로 반발하기 때문에 전자가 흐르기 힘들다. 한편 플로팅 게이트에 전자가 없는 상태에서는 S와 D 사이에 전자가 흐르기 쉽다. 이렇게 전자의 흐름 여부, 즉 S와 D 사이에 전류가 흐르는가로 셀의 값이 1인지 0인지를 판단한다.

플래시 메모리에서는 두 가지 과정을 거쳐 데이터가 저장된다. 일단 데이터를 지우는 과정이 필요하다. 데이터 지우기는 여러 개의 셀이 연결된 블록 단위로 이루어진다. 블록에 포함된 모든 셀마다 G에 0V, p형 반도체에 약 20V의 양의 전압을 가하면, 플로팅 게이트에 전자가 있는 경우, 그 전자가 터널 절연체를 넘어 p형 반도체로 이동한다. 반면 전자가 없는 경우는 플로팅 게이트에 변화가 없다. 따라서 해당 블록의 모든 셀은 0의 상태가 된다. 터널 절연체는 전류 흐름을 항상 차단하는 일반 절연체와는 다르게 일정 이상의 전압이 가해졌을 때는 전자를 통과시킨다.

이와 같은 과정을 거친 후에야 데이터 쓰기가 가능하다. 데이터를 저장하려면 1을 쓰려는 셀의 G에 약 20V, p형 반도체에는 0V의 전압을 가한다. 그러면 p형 반도체에 있던 전자들이 터널 절연체를 넘어 플로팅 게이트로 들어가 저장된다. 이것이 1의 상태이다.

플래시 메모리는 EPROM과 EEPROM의 장점을 취하여 만든 메모리이다. EPROM은 한 개의 트랜지스터로 셀을 구성하여 셀 면적이 작은 반면, 데이터를 지울 때 칩을 떼어 내어 자외선으로 소거해야 한다는 단점이 있다. EEPROM은 전기를 이용하여 간편하게 데이터를 지울 수 있지만, 셀 하나당 두 개의 트랜지스터가 필요하다. 플래시 메모리는 한 개의 트랜지스터로 셀을 구성하며, 전기적으로 데이터를 쓰고 지울 수 있다. 한편 메모리는 전원 차단 시에 데이터의 보존 유무에 따라 휘발성과 비휘발성 메모리로 구분되는데, 플래시 메모리는 플로팅 게이트가 절연체로 둘러싸여 있기 때문에 전원을 꺼도 1이나 0의 상태가 유지되므로 비휘발성 메모리이다. 이런 장점 때문에 휴대용 디지털 장치는 주로 플래시 메모리를 이용하여 데이터를 저장한다.



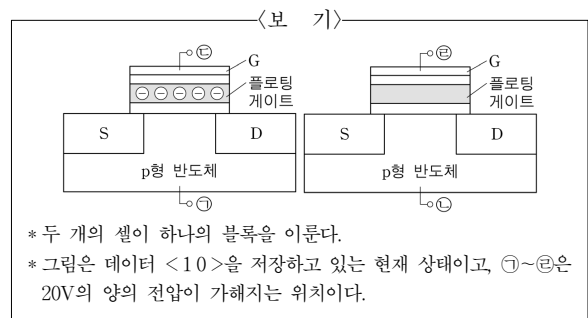
19. 윗글에 대한 설명으로 가장 적절한 것은?

- ① 대상의 구조를 바탕으로 작동 원리를 설명하고 있다.
- ② 대상의 장점을 설명한 뒤 사용 방법을 알려 주고 있다.
- ③ 대상의 크기를 기준으로 자세한 기능을 설명하고 있다.
- ④ 대상의 구성 요소를 설명한 뒤 제작 원리를 알려 주고 있다.
- ⑤ 대상의 단점을 나열하고 새로운 방식의 필요성을 제기하고 있다.

20. 윗글의 ‘플래시 메모리’에 대하여 추론한 내용으로 옳은 것은?

- ① D에 3V의 양의 전압을 가하면 플로팅 게이트의 전자가 사라진다.
- ② 터널 절연체 대신에 일반 절연체를 사용하면 데이터를 반복해서 지우고 쓸 수 없다.
- ③ 데이터 지우기 과정에서 자외선에 노출해야 데이터를 수정할 수 있다.
- ④ EEPROM과 비교되는 EPROM의 단점을 개선하여 셀 면적을 더 작게 만들었다.
- ⑤ 데이터를 유지하기 위해서는 전력을 계속 공급해 주어야 한다.

21. 윗글과 <보기>에 따라 플래시 메모리의 데이터 <10>을 <01>로 수정하려고 할 때, 단계별로 전압이 가해질 위치가 옳은 것은? [3점]



	1단계	2단계
①	㉠	㉣
②	㉢	㉡
③	㉠과 ㉡	㉣
④	㉡과 ㉢	㉣
⑤	㉢과 ㉣	㉡

◆ 16년 3월 고2 16~18번

[16~18] 다음을 읽고 물음에 답하시오.

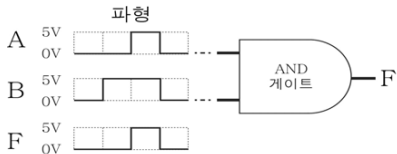
컴퓨터와 같은 디지털 장치는 1과 0밖에 구분하지 못한다. 더 정확하게 말하면 실제 숫자를 구분하는 것이 아니라 신호가 왔는지(1), 오지 않았는지(0)의 정보를 구분하는 것이다. 신호 여부는 두 개의 전압 레벨, 예를 들어 5V와 0V를 사용하여 구분한다. 즉 전압 레벨이 변화하는 것을 2진수로 처리하여 디지털 회로를 설계하는 것이다. 디지털 회로의 기본 요소는 논리 게이트인데, 논리 게이트는 하나 이상의 입력 값에 대한 논리 연산을 수행하여 출력 값을 얻는다.

디지털 회로는 출력을 결정하는 방법에 따라 조합 논리 회로와 순차 논리 회로로 나눌 수 있다. 조합 논리 회로는 현재의 입력 값들만 이용하여 출력 값을 결정한다. 즉 회로를 구성하는 논리 게이트들이 입력 신호들을 받는 즉시 그것들을 조합하여 출력 신호를 발생시킨다. 반면 순차 논리 회로는 과거의 출력 값이 현재의 출력에 영향을 미친다. 출력 값이 그 시점의 입력 값뿐만 아니라 이전 상태의 출력 값에 의해서도 결정되는 것이다. 가령 디지털 장치에서 수를 셀 때, 이전 상태의 출력 값과 현재의 값을 논리 연산하여 출력하므로 다음 상태로 변화할 때까지 현 상태를 기억하는 기능이 필요하다. 이러한 특성 때문에 순차 논리 회로는 조합 논리 회로와 달리 기억 기능을 가지고 있다. 이전 상태의 출력 값은 다음 단계의 순차 논리 회로 동작을 위해 피드백 경로를 통해 다시 순차 논리 회로의 입력으로 들어가게 된다.

조합 논리 회로이든 순차 논리 회로이든 디지털 회로의 설계는 다양한 논리 게이트들을 얼마나 효율적으로 연결하느냐가 중요하다. 가장 기본적인 논리 게이트로는 NOT 게이트, AND 게이트, OR 게이트가 있다. NOT 게이트는 보통 인버터라 부르며 출력 값이 입력 값과 반대가 되도록 변환한다. 예를 들어 입력 값이 0이면 출력 값은 1이고 입력 값이 1이면 출력 값은 0이 된다. 따라서 입력 가능한 조합은 1과 0, 두 개뿐이다. AND 게이트는 입력 단자를 통해 들어오는 입력 값이 모두 1일 때만 출력 값이 1이고, 만일 한 개라도 0이면 출력 값은 0이 된다. OR 게이트는 입력 값이 어느 하나라도 1이면 출력 값이 1이 되고, 입력 값이 모두 0일 때만 출력 값이 0이 된다. 논리 게이트들의 입력 가능한 조합의 수는 2의 거듭제곱을 따른다. 즉 입력 단자가 2개면 입력 가능 조합은 4개, 입력 단자가 3개면 입력 가능 조합은 8개가 된다.

논리 게이트의 입력과 출력은 전기적 신호가 바뀌는 모습으로 나타낼 수 있다. 아래 그림은 AND 게이트에 입력 신호가 들어왔을 때, 어떤 출력 신호가 나오는지 나타낸 것이다. A와 B의 파형이 각각의 입력 단자

에 들어올 때, AND 게이트는 F와 같은 파형을 출력하게 된다. 여기서 파형이란 0과 1에 해당하는 전기적 신호(0V와 5V)가 시간에 따라 연속적으로 바뀌는 모습을 표현한 것을 말한다. 다른 게이트들도 이와 마찬가지로 전기적 신호가 바뀌는 모습을 표현하여 입력 신호와 출력 신호로 나타낼 수 있다.



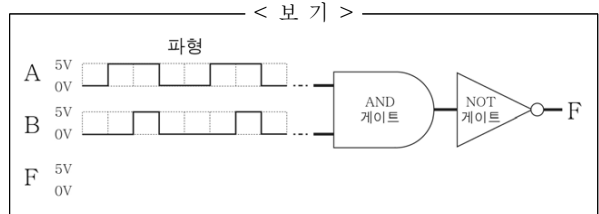
16. 밑글의 내용과 일치하지 않는 것은?

- ① 어떤 논리 게이트라도 출력 값은 0과 1, 둘 중 하나이다.
- ② 논리 게이트의 입력과 출력은 전기적 신호가 바뀌는 모습으로 나타낼 수 있다.
- ③ 어떤 논리 게이트에 n개의 입력 단자가 있으면 입력 가능한 조합의 수는 2^n 개이다.
- ④ NOT 게이트, AND 게이트, OR 게이트에서 입력 값이 모두 0이면 각각의 출력 값은 모두 0이다.
- ⑤ 조합 논리 회로와 순차 논리 회로는 둘 다 논리 게이트를 연결해 구현할 수 있다는 점에서는 같다.

17. [A]로 보아 출력을 결정하는 방법이 다른 하나는?

- ① 자동차의 문이 열리면 경고음이 울리는 경우
- ② 현관에 사람이 들어왔을 때 전등이 켜지는 경우
- ③ 사람이 다가가야만 움직이기 시작하는 에스컬레이터의 경우
- ④ 이용객이 안전띠를 착용하지 않아 운행되지 않는 놀이 기구의 경우
- ⑤ 은행에서 지폐를 세는 기계가 만 원권의 개수를 세어 총액을 나타내는 경우

18. 밑글을 참고할 때, <보기>에서 출력될 F의 파형으로 적절한 것은?



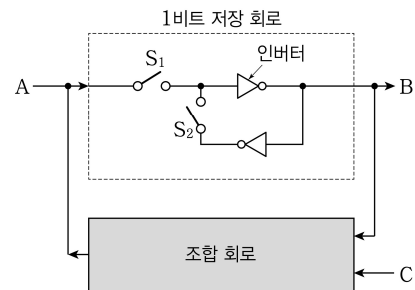
- ① F 5V 0V [Waveform: High for 1st interval, Low for others]
- ② F 5V 0V [Waveform: High for 2nd interval, Low for others]
- ③ F 5V 0V [Waveform: High for 3rd interval, Low for others]
- ④ F 5V 0V [Waveform: High for 4th interval, Low for others]
- ⑤ F 5V 0V [Waveform: High for 1st and 3rd intervals, Low for others]

◆ 15 LEET 언어이해 27~29번

[27~29] 다음 글을 읽고 물음에 답하시오.

컴퓨터의 CPU가 어떤 작업을 수행하는 것은 CPU의 '논리 상태'가 시간에 따라 바뀌는 것을 말한다. 가령, $Z = X + Y$ 의 연산을 수행하려면 CPU가 X와 Y에 어떤 값을 차례로 저장한 다음, 이것을 더하고 그 결과를 Z에 저장하는 각각의 기능을 순차적으로 진행해야 한다. CPU가 수행할 수 있는 기능은 특정한 CPU의 논리 상태와 일대일로 대응되어 있으며, 프로그램은 수행하고자 하는 작업의 진행에 맞도록 CPU의 논리 상태를 변경한다. 이를 위해 CPU는 현재 상태를 저장하고 이것에 따라 해당 기능을 수행할 수 있는 부가 회로도 갖추고 있다. 만약 CPU가 가지는 논리 상태의 개수가 많아지면 한 번에 처리할 수 있는 기능이 다양해진다. 따라서 처리할 데이터의 양이 같다면 이를 완료하는 데 걸리는 시간이 줄어든다.

논리 상태는 2진수로 표현되는데 논리 함수를 통해 다른 상태로 변환된다. 논리 소자가 연결된 조합 회로는 논리 함수의 기능을 가지는데, 조합 회로는 논리 연산은 가능하지만 논리 상태를 저장할 수는 없다. 어떤 논리 상태를 '저장'한다는 것은 2진수 정보의 시간적 유지를 의미하는데, 외부에서 입력이 유지되지 않더라도 입력된 정보를 논리 회로 속에 시간적으로 가둘 수 있어야 한다.



<그림> 순차 논리 회로

인버터는 입력이 0일 때 1을, 1일 때 0을 출력하는 논리 소자이다. <그림>의 점선 내부에 표시된 '1비트 저장 회로'를 생각해 보자. 이 회로에서 스위치 S_1 은 연결하고 스위치 S_2 은 끊은 채로 A에 정보를 입력한다. 그런 다음 S_2 을 연결하면 S_1 을 끊더라도 S_2 을 통하는 ㉠ 피드백 회로에 의해 A에 입력된 정보와 반대되는 값이 지속적으로 B에 출력된다. 따라서 이 회로는 0과 1 중 1개의 논리 상태, 즉 1비트의 정보를 저장할 수 있다. 이러한 회로가 2개가 있다면 00, 01, 10, 11의 4가지 논리 상태, n개가 있다면 2^n 가지의 논리 상태 중 1개를 저장할 수 있다.

그렇다면 논리 상태의 변화는 어떻게 일어날까? 이제 <그림>과 같이 1비트 저장 회로와 조합 회로로 구성되는 '순차 논리 회로'를 생각해 보자. 이 회로에서 조합 회로는 외부 입력 C와 저장 회로의 출력 B를 다시 입력으로 되받아, 내장된 논리 함수를 통해 논리 상태를 변환하고, 이를 다시 저장 회로의 입력과 연결하는 ㉡ 피드백 회로를 구성한다. 예를 들어 조합 회로가 두 입력이 같을 때는 1을, 그렇지 않을 경우 0을 출력한다고 하자. 만일 B에서 1이 출력되고 있을 때 C에 1이 입력된다면 조합 회로는 1을 출력하게

된다. 이때 외부에서 어떤 신호를 주어 S_2 가 열리자마자 S_1 이 닫힌 다음 다시 S_2 가 닫히고 S_1 이 열리는 일련의 스위치 동작이 일어나도록 하면, 조합 회로의 출력은 저장 회로의 입력과 연결되어 있으므로 B에서 출력되는 값은 0으로 바뀐다. 그런 다음 C의 값을 0으로 바꾸어주면, 일련의 스위치 동작이 다시 일어나더라도 B의 값은 바뀌지 않는다. 하지만 C에 다시 1을 입력하고 일련의 스위치 동작이 일어나도록 하면 B의 출력은 1로 바뀐다. 따라서 C에 주는 입력에 의해 저장 회로가 출력하는 논리 상태를 임의로 바꿀 수 있다.

만일 이 회로에 2개의 1비트 저장 회로를 병렬로 두어 출력을 2비트로 확장하면 00~11의 4가지 논리 상태 중 1개를 출력할 수 있다. 조합 회로의 외부 입력도 2비트로 확장하면 조합 회로는 저장 회로의 현재 출력과 합친 4비트를 입력받게 된다. 이를 내장된 논리 함수에 의해 다시 2비트 출력을 만들어 저장 회로의 입력과 연결한다. 이와 같이 2비트로 확장된 순차 논리 회로에서 외부 입력을 주고 스위치 동작이 일어나도록 하면, 저장 회로의 출력은 2배로 늘어난 논리 상태 중 하나로 바뀐다.

이 회로에 일정한 시간 간격으로 외부 입력을 바꾸고 스위치 동작 신호를 주면, 주어지는 외부 입력에 따라 특정한 논리 상태가 순차적으로 출력에 나타나게 된다. ④ 이런 회로가 N비트로 확장된 대표적인 사례가 CPU이며 스위치를 동작시키는 신호가 CPU 클록이다. 회로 외부에서 입력되는 정보는 컴퓨터 프로그램의 '명령 코드'가 된다. 명령 코드를 CPU의 외부 입력으로 주고 클록 신호를 주면 CPU의 현재 논리 상태는 특정 논리 상태로 바뀐다. 이때 출력에 연결된 회로가 바뀐 상태에 해당하는 기능을 수행하게 된다. CPU 클록은 CPU의 상태 변경 속도, 즉 CPU의 처리 속도를 결정한다.

27. 밑줄의 내용과 일치하지 않는 것은?

- ① CPU가 수행할 수 있는 기능과 그에 해당하는 논리 상태는 정해져 있다.
- ② 인버터는 입력되는 2진수 논리 값과 반대되는 값을 출력하는 논리 소자이다.
- ③ 순차 논리 회로에서 저장 회로의 출력은 조합 회로의 출력 상태와 동일하다.
- ④ CPU는 프로그램 명령 코드에 의한 논리 상태 변경을 통해 작업을 수행한다.
- ⑤ 조합 회로는 2진수 입력에 대해 내부에 구현된 논리 함수의 결과를 출력한다.

28. ㉠과 ㉡에 대한 이해로 적절한 것은?

- ① ㉠은 조합 회로를 통해서, ㉡은 인버터를 통해서 피드백 기능이 구현된다.
- ② ㉠과 ㉡의 각 회로에서 피드백 기능을 위해 입력하는 정보의 개수는 같다.
- ③ ㉠과 ㉡은 모두 외부에서 입력되는 논리 상태를 그대로 저장하는 기능이 있다.
- ④ ㉠은 정보를 저장하기 위한 구조이며, ㉡은 논리 상태를 변경하기 위한 구조이다.
- ⑤ ㉠은 스위치 S_1 이 연결될 때, ㉡은 스위치 S_2 가 연결될 때 피드백 기능이 동작한다.

29. ㉠에서 N을 증가시켰을 때의 변화를 이해한 것으로 적절하지 않은 것은?

- ① 프로그램에서 사용 가능한 명령 코드의 종류가 증가한다.
- ② 조합 회로가 출력하는 논리 상태의 가짓수가 증가한다.
- ③ CPU가 가질 수 있는 논리 상태의 가짓수가 증가한다.
- ④ CPU에서 진행되는 상태 변경의 속도가 증가한다.
- ⑤ 동일한 양의 데이터를 처리하는 속도가 증가한다.

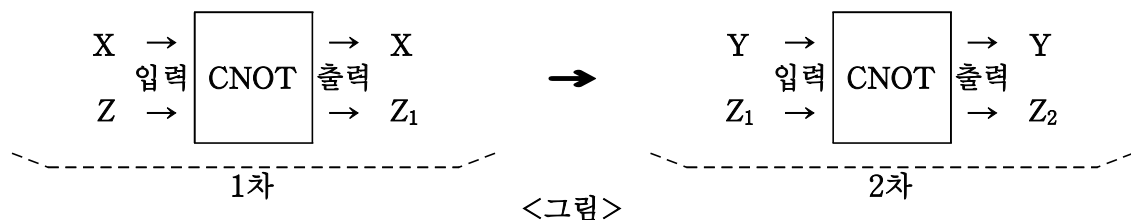
[15 ~ 18] 다음 글을 읽고 물음에 답하시오.

미시 세계에서 양자의 중첩은 여러 상태들이 겹쳐 있는 것이다. 이 중 특정 상태가 측정될 확률은 알 수 있지만 관측하기 전까지는 어떤 상태인지 정확히 알 수 없다. 어떤 상태인지 알기 위해서는 관측을 해야 하는데, 거시 세계의 관측 행위로 인해 양자의 중첩이 붕괴되어 중첩 상태 중 어느 한 상태로 확정된다. 예를 들어, 미시 세계의 철수는 앉아 있거나 서 있는 두 가지 상태가 중첩되어 있다면, 관측 이전에는 서 있는 철수가 관측될 확률과 앉아 있는 철수가 관측될 확률만 알 수 있다. 이처럼 미시 세계의 철수는 서 있는 상태와 앉아 있는 상태가 중첩되어 있는데, 이 중첩 상태는 거시 세계의 선생님이 관측했을 때 붕괴되어 비로소 서 있는 상태나 앉아 있는 상태 중 어느 한 상태로 확정된다.

양자 컴퓨터는 이러한 양자의 중첩 현상을 활용한 것이다. 기존의 컴퓨터는 정보의 기본 단위인 비트를 사용하며, 비트는 0 또는 1이라는 확정적인 값을 갖는다. 이와 달리 양자 컴퓨터는 큐비트를 사용하며, 큐비트는 0일 확률과 1일 확률을 가진 중첩된 상태를 갖는다. 따라서 기존 컴퓨터는 확정된 값을 입력해서 한 번에 하나씩 연산하여 확정된 값을 출력하지만, 양자 컴퓨터는 중첩된 큐비트를 한 번 입력함으로써, 중첩 상태가 가질 수 있는 모든 가능한 경우에 대한 연산을 한꺼번에 수행한다. 이런 원리상 양자 컴퓨터는 기존 컴퓨터보다 처리 속도가 빠르다.

기존 컴퓨터와 양자 컴퓨터는 오류를 검증하는 방법에도 차이가 있다. 기존 컴퓨터는 데이터를 저장할 때 동일 비트를 세 번 이상 저장한 후 다수결의 원리에 따라 오류를 보정할 수 있다. 일반적인 컴퓨터는 정상 비트일 가능성이 오류 비트일 가능성보다 높기 때문에, 0을 000으로, 1을 111로 저장해두면 그것이 설령 010, 001, 011, 101로 읽힌다고 하더라도 ㉔ 각각의 원래 값을 추정하기는 어렵지 않은 것이다. 그러나 양자 컴퓨터에서는 이러한 방식을 쓸 수 없다. 왜냐하면 중첩된 큐비트를 복사하거나 비교하려고 그 값을 관측하는 순간 중첩이 붕괴되기 때문이다.

이러한 문제를 해결하기 위해 양자 컴퓨터는 논리 게이트 CNOT(Controlled-NOT)을 통해 오류를 검증한다. 중첩된 큐비트가 오류일 가능성보다 정상일 가능성이 높은 양자 컴퓨터에서, 같은 내용의 연산을 통해 도출된 중첩된 큐비트 X와 중첩된 큐비트 Y가 있다고 하자. 연산에 오류가 없다면 X와 Y의 관측값은 같아야 한다. 관측은 중첩된 두 큐비트를 읽지 않고도 서로 같은지 다른지를 어떻게 판정할 수 있는가이다. <그림>의 CNOT에는 위아래로 두 개의 입력과 두 개의 출력이 있는데, 위의 입력은 입력된 그대로 출력된다. 위의 입력에 0이 들어오면 아래의 입력은 그대로 출력되며, 위의 입력에 1이 들어오면 아래의 입력이 반전되어 출력된다. 반전이 되는 경우, 0은 1로 1은 0으로 바뀐다.



먼저 큐비트 Z는 0으로 고정한다. 그리고 X와 Z를 CNOT에 입력한다. X가 1인 경우 Z를 반전하여 출력한다. 출력된 값을 각각 X와 Z₁이라고 하자. 다음으로, Y와 Z₁을 CNOT에 입력한다. Y가 1인 경우 Z₁을 반전하여 출력한다. 출력된 값을 각각 Y와 Z₂라고 하자. 만약 Z₂가 0이라면 X와 Y는 같고, 이는 연산에 오류가 있을 가능성이 낮다는 것을 의미한다. 반면 Z₂가 1이라면 X와 Y는 다르며, 이는 어디엔가 오류가 있음을 의미한다. 이러한 오류 검증의 과정에서 X와 Y의 값을 관측하지 않는다.

15. 윗글에 대한 설명으로 가장 적절한 것은? [3점]

- ① 상황을 가정하여 특정 대상의 향후 전망을 제시하고 있다.
- ② 특정 대상이 발전되어 온 과정을 통시적으로 설명하고 있다.
- ③ 다른 대상과의 차이점을 바탕으로 특정 대상을 설명하고 있다.
- ④ 서로 다른 이론을 적용하여 특정 대상의 장단점을 분석하고 있다.
- ⑤ 유사한 두 대상의 공통점을 제시한 후 각각의 의의를 서술하고 있다.

16. 윗글을 읽은 학생의 반응으로 적절하지 않은 것은? [3점]

- ① 거시 세계에서는 관측을 통해 양자의 중첩된 상태를 확인할 수 없겠군.
- ② 오류 비트일 가능성이 50% 이상이라면 다수결을 이용한 오류 검증은 실효성이 없겠군.
- ③ 미시 세계에서 중첩된 양자는 어떤 상태가 어느 정도의 확률로 있는지를 알 수 없겠군.
- ④ 기존 컴퓨터가 여러 번 수행해야 하는 연산을 양자 컴퓨터는 한 번에 수행할 수 있겠군.
- ⑤ 기존 컴퓨터의 오류 검증에서 동일 비트의 저장 횟수를 늘리면 검증의 정확도가 올라가겠군.

17. 문맥을 고려할 때 ㉠을 추정한 내용으로 적절한 것은? [3점]

- ① 0, 0, 0, 1 ② 0, 0, 1, 1 ③ 0, 1, 1, 1 ④ 1, 0, 1, 0 ⑤ 1, 1, 0, 0

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

— < 보 기 > —

다음은 오류 검증을 위해 사용한 CNOT 게이트의 진리표*를 간략하게 제시한 것이다.

	X	Y	Z	Z ₁	Z ₂
㉠	(㉠)	0	0	0	0
㉡	0	1	0	0	(㉢)
㉢	1	0	0	1	(㉣)
㉣	(㉤)	1	0	1	0

* 진리표 : 논리연산의 가능한 모든 입력값의 조합을 열거하고, 각각의 조합에 대한 출력값을 나타내는 표.

- ① ㉠과 ㉣행의 Z₂가 각각 0이므로 ㉠은 0, ㉤는 1이다.
- ② ㉡과 ㉢행에서 X와 Y가 서로 다르므로, ㉢와 ㉣는 서로 다르다.
- ③ Z₁에 영향을 미치는 것은 X이다.
- ④ 오류 검증을 위해 관측하는 것은 Z₂이다.
- ⑤ X와 Y는 서로 영향을 미치지 않는다.

- 정답: 15.③ 16.③ 17.② 18.②

◆ 17년 3월 고2 29~34번

[29~34] 다음 글을 읽고 물음에 답하십시오.

논리학은 논증에서 전제들로부터 결론이 도출될 수 있는지를 판단하는 학문이다. 논리학을 학문으로 체계화한 사람은 기원전 3세기의 철학자 아리스토텔레스이다. 그는 논증의 일반적인 원리를 연구함으로써 논증의 타당성을 검토하려고 했다.

아리스토텔레스는 정언 문장으로 이루어진 연역 논증을 중심으로 논리학을 연구하였는데, 이러한 논리학을 ㉠ 전통 논리학이라 부른다. 연역 논증은 결론이 이미 전제에 포함되어 있기 때문에 전제가 참이면 결론이 반드시 참이 되는 형식의 논증을 말한다. 그리고 정언 문장이란 참과 거짓을 판별할 수 있는 문장 중에서 ‘주어-술어’로 이루어진 다음 네 가지 형식의 문장을 말한다.

- 모든 A는 B이다. • 모든 A는 B가 아니다.
- 어떤 A는 B이다. • 어떤 A는 B가 아니다.

(1)은 연역 논증의 하나로 세 개의 정언 문장으로 구성된 정언 삼단 논증의 예이다.

(1) 모든 [아버지]는 [남자]이다. <전제1>
 어떤 [사람]은 [아버지]이다. <전제2>
 그러므로 어떤 [사람]은 [남자]이다. <결론>

(1)에서 결론의 주어가 되는 개념인 ‘사람’을 소명사(S), 결론의 술어가 되는 개념인 ‘남자’를 대명사(P)라 하며, ‘아버지’와 같이 전제에만 있으면서 전제들을 엮을 수 있도록 하는 개념을 증명사(M)라 한다. 만약 술어가 ‘걷는다’와 같이 동사인 경우에는 ‘걷는 존재’와 같은 명사(名辭)로 나타낼 수 있다. 그리고 대명사가 포함된 전제를 대전제, 소명사가 포함된 전제를 소전제라 한다. 이를 사용하여 (1)을 형식화하면 (2)와 같다.

(2) 모든 [M]은 [P]이다. <대전제>
 어떤 [S]는 [M]이다. <소전제>
 그러므로 어떤 [S]는 [P]이다. <결론>

정언 삼단 논증에서 증명사(M)는 전제들 사이에서 소명사(S)와 대명사(P)를 연결시키는 역할을 맡는다. 만약 전제 증명사가 없으면 소명사와 대명사를 연결시킬 수 없으므로 논증을 구성할 수 없다. (2)에서 결론의 [S]-[P]는 배열이 고정되어 있지만, 전제의 ‘M, P, S’는 배열이 자유롭기 때문에 ‘M, P, S’를 조합해서 ㉡ 정언 삼단 논증의 네 가지 유형을 만들 수 있다. 이를 아리스토텔레스는 정언 삼단 논증의 제1격에서부터 제4격이라고 명명하였다. 이와 같이 정언 문장을 대명사, 증명사, 소명사로 분석한 전통 논리학을 명사 단위의 논리학이라 한다.

그런데 (3)은 정언 삼단 논증의 형태를 띠고 있는 것처럼 보이지만 정언 삼단 논증의 유형에서 벗어나 있다.

(3) 만약 비가 온다면, 소풍은 취소된다. <전제1>
 비가 온다. <전제2>
 그러므로 소풍은 취소된다. <결론>

<전제1>은 ‘비가 온다.’와 ‘소풍은 취소된다.’의 두 문장이

결합된 것이다. <전제2>는 <전제1>을 구성하고 있는 문장 중 하나이며, <결론>은 <전제1>을 구성하고 있는 나머지 문장이다. 따라서 정언 문장만을 대상으로 한 전통 논리학으로는 이 논증의 타당성을 분석할 수 없다.

20세기 독일의 논리학자 프레게는 소명사, 대명사, 증명사를 중심으로 논증의 타당성을 검토하는 정언 삼단 논증의 한계를 지적하면서, 명제를 단위로 논증을 분석하는 ㉢ 명제 논리학을 제안하였다. 명제란 참과 거짓을 판단할 수 있는 문장이다. 전통 논리학에서는 정언 문장을 명사 단위로 나누어서 분석하였지만, 명제 논리학에서는 명제 자체를 논증의 기본 단위로 삼았다. 그리고 더 이상 분해할 수 없는 명제를 단순 명제라 하여 ‘p, q, r’ 등의 기호로 표시하고, 단순 명제에 논리적 연결사인 ‘∨(또는)’, ‘∧(그리고)’, ‘→(만약 ...이면 ...이다)’, ‘~(...가 아니다)’ 등을 사용하여 복합 명제를 만들었다.

가령 (3)의 <전제1>은 ‘비가 온다.’와 ‘소풍은 취소된다.’의 두 개의 단순 명제가 연결된 복합 명제로, 각각의 단순 명제를 ‘p’와 ‘q’로 나타낼 수 있다. 그리고 단순 명제 ‘p’와 ‘q’는 ‘만약 ...이면 ...이다.’에 해당하는 논리적 연결사 ‘→’를 사용하여 ‘p→q’와 같은 복합 명제로 나타낼 수 있다. 따라서 (3)을 기호화하여 나타내면 다음과 같다.

(4) 만약 p이면 q이다. (4') p → q
 p이다. ⇒ p
 그러므로 q이다. q

아리스토텔레스는 정언 문장에서 명사들 간의 관계에 의존하여 논증의 타당성을 설명하였지만, 명제 논리학에서는 명제들의 진릿값과 논리적 연결사에 의존하여 논증의 타당성을 평가했다. 가령, ‘p∨q’는 ‘p’와 ‘q’ 중 하나라도 참이면 참이 되지만, ‘p∧q’는 ‘p’와 ‘q’ 모두 참일 때에만 참이 된다. 또한 ‘p→q’는 ‘p’와 ‘q’가 모두 참인 경우에는 참이지만, ‘p’가 참이고 ‘q’가 거짓인 경우에는 거짓이 된다. 따라서 복합 명제의 진릿값은 단순 명제의 진릿값과 논리적 연결사에 의존한다. (4')는 <전제2>가 <전제1>의 선행 조건인 p를 긍정함으로써 <결론>인 q가 성립된다고 주장하는 논증인데, 이러한 형식을 ㉣ 전건 긍정이라 한다.

명제 논리학은 정언 문장만을 분석의 대상으로 삼는 전통 논리학에서 다루지 못하는 문장들까지 논증의 대상으로 포함시켰다는 점에서 의미가 있다. 또한 논증의 모든 요소를 기호화하여 ㉤ 명제 논리학은 자연 언어를 컴퓨터로 프로그래밍할 수 있는 길을 열어 주었다. 이후 명제 논리학은 술어 논리학으로 발전되었는데, 술어 논리학은 술어 기호를 사용하여 명제 논리학에서 다루지 못한 명제 내의 논리 구조를 분석함으로써 논리학의 범위를 한층 더 확대시켰다.

* 명사(名辭): 하나의 개념을 언어로 나타내며 명제를 구성하는 데에 요소가 되는 말.

29. 밑줄에 대한 설명으로 가장 적절한 것은?

- ① 논리학의 발전 과정을 개괄적으로 소개하고 있다.
- ② 논리학의 의미를 다양한 관점에서 고찰하고 있다.
- ③ 논리학의 특징을 인접 학문과 비교하여 분석하고 있다.
- ④ 논리학의 논증 방식이 단순화된 배경을 설명하고 있다.
- ⑤ 논리학의 변화에 영향을 준 여러 학문을 고찰하고 있다.

30. 밑줄의 내용과 일치하지 않는 것은?

- ① 연역 논증에서 전제가 참이면 결론이 참이 된다.
- ② 전통 논리학은 정언 문장을 명사 단위로 분석한다.
- ③ 주어와 술어로 구성된 모든 문장은 정언 문장이다.
- ④ 명제 논리학은 명제 자체를 논증의 기본 단위로 삼는다.
- ⑤ 술어 논리학은 명제 내의 논리 구조를 분석하여 논증한다.

31. ㉠과 ㉡의 입장에서 <보기>를 분석한 것으로 적절하지 않은 것은?

— < 보 기 > —	
┐	└
모든 생명체는 죽는다.	<전제1> 민수는 일하거나 논다.
어떤 사람은 생명체이다.	<전제2> 민수는 일하지 않는다.
어떤 사람은 죽는다.	<결론> 민수는 논다.

- ① ㉠: ┐에서 '모든 생명체는 죽는다.'는 '모든 [생명체]는 [죽는 존재]이다.'와 같이 나타낼 수 있다.
- ② ㉠: ┐에서 '생명체'는 전제에만 나타나므로 증명사이고, '사람'은 결론의 주어가 되는 개념이므로 소명사이다.
- ③ ㉡: ┐에서 '모든 생명체는 죽는다.'를 '만약 생명체라면 죽는 존재이다.'로 재구성한다면, 이는 'p → q'의 구조에 해당한다.
- ④ ㉡: └의 <전제1>은 복합 명제에, <전제2>는 단순 명제에 해당한다.
- ⑤ ㉡: └의 '민수는 일하거나 논다.'를 기호로 나타내기 위해서는 논리적 연결사가 필요하다.

32. ㉠에 해당하지 않는 것은?

- | | | |
|---------------------|---------------------|---------------------|
| ① M-P
S-M
S-P | ② P-M
S-M
S-P | ③ P-M
M-S
S-P |
| ④ M-P
P-S
S-P | ⑤ M-P
M-S
S-P | |

33. ㉠의 사례로 가장 적절한 것은?

- ① 차가 달리지 않으면 멈춘다. 차가 달린다. 그러므로 차가 멈추지 않는다.
- ② 만약 그것이 생명체라면 죽는다. 그것이 죽는다. 그러므로 그것은 생명체이다.
- ③ 비가 오면 가뭄이 끝난다. 아직 가뭄이 끝나지 않았다. 그러므로 비가 오지 않았다.
- ④ 교실 청소가 끝나면 집에 갈 수 있다. 교실 청소가 끝났다. 그러므로 집에 갈 수 있다.
- ⑤ 공부를 하면 성적이 오른다. 철수는 공부를 하지 않았다. 그러므로 철수는 성적이 오르지 않았다.

34. <보기>는 ㉡을 심화 학습하는 과정에서 얻은 자료이다. 이를 이해한 내용으로 적절하지 않은 것은? [3점]

— < 보 기 > —

논리 게이트는 '1'과 '0'의 이진법 정보로 운용되는 전자 회로로 명제 논리학에 착안하여 만들어졌다. 입력값이 '1'인 것은 명제의 진릿값이 참인 경우에, 입력값이 '0'인 것은 명제의 진릿값이 거짓인 경우에 대응될 수 있다. 논리 게이트는 두 개의 입력 단자 'A', 'B'와 하나의 출력 단자 'Y'로 구성된다. <그림>은 논리 게이트 중 'OR 게이트'이다.

Y = A + B

이는 'A'와 'B' 중 하나 이상의 입력값이 '1'이면 출력값이 '1'이 되고, 입력값이 모두 '0'이면 출력값이 '0'이 되는 경우이다. 이 외에 'A', 'B'의 입력값이 모두 '1'일 때만 출력값이 '1'이 되는 'AND 게이트'도 있다.

- ① 논리 게이트에서 입력 단자 'A', 'B'는 명제 논리학의 단순 명제, 출력 단자 'Y'는 복합 명제에 대응된다고 할 수 있겠군.
- ② 논리 게이트에서 입력값에 의해 출력값이 결정되는 것은 명제 논리학에서 단순 명제의 진릿값과 논리적 연결사에 의해 복합 명제의 진릿값이 결정되는 것과 같은 원리이겠군.
- ③ 'OR 게이트'의 'A + B'를 명제 논리학의 논리적 연결사로 기호화하여 나타내면 'A ∨ B'에 해당하겠군.
- ④ 'OR 게이트'는 명제 논리학에서 두 명제 중 적어도 하나의 진릿값이 참일 때 결론의 진릿값이 참인 경우에 해당하겠군.
- ⑤ 'AND 게이트'에서 'Y'가 1인 것은 명제 논리학에서 두 명제의 진릿값 중 하나라도 참인 경우에 해당하겠군.

[27~32] 다음을 읽고 물음에 답하시오.

고전 역학에 ㉔ 따르면, 물체의 크기에 관계없이 초기 운동 상태를 정확히 알 수 있다면 일정한 시간 후의 물체의 상태는 정확히 측정될 수 있으며, 배타적인 두 개의 상태가 공존할 수 없다. 하지만 20세기에 등장한 양자 역학에 의해 미시 세계에서는 상호 배타적인 상태들이 공존할 수 있음이 알려졌다.

미시 세계에서의 상호 배타적인 상태의 공존을 이해하기 위해, 거시 세계에서 회전하고 있는 반지름 5cm의 팽이를 생각해 보자. 그 팽이는 시계 방향 또는 반시계 방향 중 한쪽으로 회전하고 있을 것이다. 팽이의 회전 방향은 관찰하기 이전에 이미 정해져 있으며, 다만 관찰을 통해 ㉕ 알게 되는 것뿐이다. 이와 달리 미시 세계에서 전자만큼 작은 팽이 하나가 회전하고 있다고 상상해 보자. 이 팽이의 회전 방향은 시계 방향과 반시계 방향의 두 상태가 공존하고 있다. 하나의 팽이에 공존하고 있는 두 상태는 관찰을 통해서 한 가지 회전 방향으로 결정된다. 두 개의 방향 중 어떤 쪽이 결정될지는 관찰하기 이전에는 알 수 없다. 거시 세계와 달리 양자 역학이 지배하는 미시 세계에서는, 우리가 관찰하기 이전에는 상호 배타적인 상태가 공존하는 것이다. 배타적인 상태의 공존과 관찰 자체가 물체의 상태를 결정한다는 개념을 받아들이기 힘들었기 때문에, 아인슈타인은 ㉖ “당신이 달을 보기 전에는 달이 존재하지 않는 것인가?”라는 말로 양자 역학의 해석에 회의적인 태도를 취하였다.

최근에는 상호 배타적인 상태의 공존을 적용함으로써 초고속 연산을 수행하는 양자 컴퓨터에 대한 연구가 진행되고 있다. 이는 양자 역학에서 말하는 상호 배타적인 상태의 공존이 현실에서 실제로 구현될 수 있음을 잘 보여 주는 예라 할 수 있다. 미시 세계에 대한 이러한 연구 성과는 거시 세계에 대해 우리가 자연스럽게 ㉗ 지니게 된 상식적인 생각들에 근본적인 의문을 ㉘ 던진다. 이와 비슷한 의문은 논리학에서도 볼 수 있다.

고전 논리는 ‘참’과 ‘거짓’이라는 두 개의 진리치만 있는 이치 논리이다. 그리고 고전 논리에서는 어떠한 진술이든 ‘참’ 또는 ‘거짓’이다. 이는 우리의 상식적인 생각과 잘 ㉙ 들어맞는다. 그러나 프리스트에 따르면, ‘참’인 진술과 ‘거짓’인 진술 이외에 ‘참인 동시에 거짓’인 진술이 있다. 이를 설명하기 위해 그는 ‘거짓말쟁이 문장’을 제시한다. 거짓말쟁이 문장을 이해하기 위해 자기 지시적 문장과 자기 지시적이지 않은 문장을 구분해 보자.

자기 지시적 문장은 말 그대로 자기 자신을 가리키는 문장을 말한다. 예를 들어 “이 문장은 모두 열여덟 음절로 이루어져 있다.”라는 ‘참’인 문장은 자기 자신을 가리키며 그것이 몇 음절로 이루어져 있는지 말하고 있다. 반면 “페루의 수도는 리마이다.”라는 ‘참’인 문장은 페루의 수도가 어디인지 말할 뿐 자기 자신을 가리키는 문장은 아니다.

“이 문장은 거짓이다.”는 거짓말쟁이 문장이다. 이는 ‘이 문장’이라는 표현이 문장 자체를 가리키며 그것이 ‘거짓’이라고 말하는 자기 지시적 문장이다. 그렇다면 프리스트는 왜 거짓말쟁이 문장에 ‘참인 동시에 거짓’을 부여해야 한다고 생각할까? 이에 답하기 위해 우선 거짓말쟁이 문장이 ‘참’이라고 가정해 보자. 그렇다면 거짓말쟁이 문장은 ‘거짓’이다. 왜냐하면 거짓말쟁이

문장은 자기 자신을 가리키며 그것이 ‘거짓’이라고 말하는 문장이기 때문이다. 반면 거짓말쟁이 문장이 ‘거짓’이라고 가정해 보자. 그렇다면 거짓말쟁이 문장은 ‘참’이다. 왜냐하면 그것이 바로 그 문장이 말하는 바이기 때문이다. 프리스트에 따르면 어떤 경우에도 거짓말쟁이 문장은 ‘참인 동시에 거짓’인 문장이다. 따라서 그는 거짓말쟁이 문장에 ‘참인 동시에 거짓’을 부여해야 한다고 본다. 그는 거짓말쟁이 문장 이외에 ‘참인 동시에 거짓’인 진리치가 존재함을 뒷받침하는 다양한 사례를 제시한다. 특히 그는 양자 역학에서 상호 배타적인 상태의 공존은 이 점을 시사하고 있다고 본다.

고전 논리에서는 ‘참인 동시에 거짓’인 진리치를 지닌 문장을 다룰 수 없기 때문에 프리스트는 그것도 다룰 수 있는 비고전 논리 중 하나인 LP*를 제시하였다. 그런데 LP에서는 직관적으로 호소력 있는 몇몇 추론 규칙이 성립하지 않는다. 전진 긍정 규칙을 예로 들어 생각해 보자. 고전 논리에서는 전진 긍정 규칙이 성립한다. 이는 ㉚ “P이면 Q이다.”라는 조건문과 그것의 전진인 P가 ‘참’이라면 그것의 후진인 Q도 반드시 ‘참’이 된다는 것이다. 이와 비슷한 방식으로 LP에서 전진 긍정 규칙이 성립하려면, 조건문과 그것의 전진인 P가 모두 ‘참’ 또는 ‘참인 동시에 거짓’이라면 그것의 후진인 Q도 반드시 ‘참’ 또는 ‘참인 동시에 거짓’이어야 한다. 그러나 LP에서 조건문의 전진은 ‘참인 동시에 거짓’이고 후진은 ‘거짓’인 경우, 조건문과 전진은 모두 ‘참인 동시에 거짓’이지만 후진은 ‘거짓’이 된다. 비록 전진 긍정 규칙이 성립하지는 않지만, LP는 고전 논리에 대한 근본적인 의문들에 답하기 위한 하나의 시도로서 의의가 있다.

*LP: ‘역설의 논리(Logic of Paradox)’의 약자.

27. 문맥을 고려할 때 ㉙의 의미를 추론한 내용으로 가장 적절한 것은?

- ㉙ 많은 사람들이 항상 달을 관찰하고 있으므로 달이 존재한다.
- ㉙ 달은 질량이 매우 큰 거시 세계의 물체이므로 관찰 여부와 상관없이 존재한다.
- ㉙ 달은 관찰 여부와 상관없이 존재하므로 누군가 달을 관찰하기 이전에도 존재한다.
- ㉙ 달은 원래부터 있었지만 우리가 관찰하지 않으면 존재 여부에 대해 말할 수 없다.
- ㉙ 달이 있을 가능성과 없을 가능성이 반반이므로 관찰 이후에 달이 있을 가능성은 반이다.

28. 윗글을 바탕으로, <보기>의 ‘양자 컴퓨터’와 ‘일반 컴퓨터’에 대해 이해한 내용으로 적절한 것은?

<보 기>

양자 컴퓨터는 여러 개의 이진수들을 단 한 번에 처리함으로써 일반 컴퓨터보다 훨씬 빠른 속도로 연산을 수행한다. 연산 속도에 영향을 미치는 다른 요소들을 배제하면, 이진수를 처리하는 횟수가 적어질수록 연산 결과를 빨리 얻을 수 있기 때문이다.

n자리 이진수를 나타내기 위해서는 n비트가 필요하고 n자리 이진수는 모두 2^n 개 존재한다. 일반 컴퓨터는 한 개의 비트에 0과 1 중 하나만을 담을 수 있어, 두 자리 이진수인 00, 01, 10, 11을 2비트를 이용하여 연산할 때 네 번에 걸쳐 처리한다. 하지만 공존의 원리를 이용하는 양자 컴퓨터는 0과 1을 하나의 비트에 동시에 담아 정보를 처리할 수 있어 두 자리 이진수를 2비트를 이용하여 연산할 때 단 한 번에 처리가 가능하다. 양자 컴퓨터는 처리할 이진수의 자릿수가 커질수록 연산 속도에서 압도적인 위력을 발휘한다.

*비트(bit): 컴퓨터가 0과 1을 이용하는 이진법으로 연산을 수행하기 위해 사용하는 최소의 정보 저장 단위.

- ① 양자 컴퓨터는 상태의 공존을 이용함으로써 연산에 필요한 비트의 수를 늘릴 수 있다.
- ② 3비트를 사용하여 세 자리 이진수를 모두 처리하려고 할 때 양자 컴퓨터는 일반 컴퓨터보다 속도가 6배 빠르다.
- ③ 한 자리 이진수를 모두 처리하기 위해 1비트를 사용한다고 할 때, 일반 컴퓨터와 양자 컴퓨터의 정보 처리 횟수는 같다.
- ④ 양자 컴퓨터의 각각의 비트에는 0과 1이 공존하고 있어 4비트로 한 번에 처리할 수 있는 네 자리 이진수의 개수는 모두 16개이다.
- ⑤ 3비트의 양자 컴퓨터가 세 자리 이진수를 모두 처리하는 속도는 6비트의 양자 컴퓨터가 여섯 자리 이진수를 모두 처리하는 속도보다 2배 빠르다.

29. [자기 지시적 문장]에 대해 이해한 내용으로 적절한 것은?

- ① “붕어빵에는 붕어가 없다.”는 자기 지시적 문장이다.
- ② “이 문장은 자기 지시적이다.”라는 자기 지시적 문장은 ‘거짓’이 아니다.
- ③ “이 문장은 거짓이다.”는 이치 논리에서 자기 지시적인 문장이 될 수 없다.
- ④ 고전 논리에서는 어떠한 자기 지시적 문장에도 진리치를 부여하지 못한다.
- ⑤ 비고전 논리에서는 모든 자기 지시적 문장에 ‘참인 동시에 거짓’을 부여한다.

30. 윗글을 통해 ㉠에 대해 적절하게 추론한 것은?

- ① LP에서 P가 ‘참인 동시에 거짓’이고 Q가 ‘거짓’이면, ㉠은 ‘거짓’이다.
- ② LP에서 ㉠과 P가 ‘참인 동시에 거짓’이면, Q도 반드시 ‘참인 동시에 거짓’이다.
- ③ LP에서 ㉠과 P가 ‘참’ 또는 ‘참인 동시에 거짓’이면, Q도 반드시 ‘참’ 또는 ‘참인 동시에 거짓’이다.
- ④ 고전 논리에서 ㉠과 P가 각각 ‘거짓’이 아닐 때, Q는 ‘거짓’이다.
- ⑤ 고전 논리에서 ㉠과 P가 ‘참’이면서 Q가 ‘거짓’인 것은 불가능하다.

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

<보 기>

A는 고전 논리를 받아들이고, B는 LP를 받아들일 뿐 아니라 양자 역학에서 상호 배타적인 상태의 공존이 시사하는 바에 대한 프리스트의 입장도 받아들인다.

A와 B는 아래의 (ㄱ)~(ㄴ)에 대하여 토론을 하고 있다.

(ㄱ) 전자 e는 관찰하기 이전에 S라는 상태에 있다.

(ㄴ) 전자 e는 관찰하기 이전에 S와 배타적인 상태에 있다.

(ㄷ) 반지름 5cm의 팽이가 시계 방향으로 회전한다.

(ㄹ) 반지름 5cm의 팽이가 반시계 방향으로 회전한다.

(단, (ㄱ)과 (ㄴ)의 전자 e는 동일한 전자이고 (ㄷ)과 (ㄹ)의 팽이는 동일한 팽이이다.)

- ① A는 (ㄱ)이 ‘참’이 아니라면 ‘거짓’이고, ‘참’, ‘거짓’ 외에 다른 진리치를 가질 수 없다고 주장할 것이다.
- ② B는 (ㄱ)은 ‘참인 동시에 거짓’일 수 있다고 주장하지만, (ㄴ)은 ‘참’이 아니라면 ‘거짓’이라고 주장할 것이다.
- ③ A와 B는 모두 (ㄷ)이 ‘참’일 때 (ㄹ)도 ‘참’이 되는 것은 불가능하다고 주장할 것이다.
- ④ A는 B와 달리 (ㄴ)이 ‘참인 동시에 거짓’이 될 수 없다고 주장할 것이다.
- ⑤ B는 A와 달리 (ㄹ)이 ‘참’이 아니라면 ‘참인 동시에 거짓’이라고 주장할 것이다.

32. 문맥상 ㉠~㉤와 바꾸어 쓸 수 있는 말로 적절하지 않은 것은?

- ① ㉠: 의거(依據)하면 ② ㉡: 인지(認知)하게
- ③ ㉢: 소지(所持)하게 ④ ㉣: 제기(提起)한다
- ⑤ ㉤: 부합(符合)한다

※ 다음 글을 읽고 물음에 답하시오. [문 39 ~ 문 40]

양자역학은 이론과 인간 경험 사이의 간극을 잘 보여준다. 입자 하나가 가상의 선을 기준으로 오른쪽에 있거나 왼쪽에 있다고 하자. 오른쪽에 있는 입자를 관측하면 우리는 그 위치를 ‘오른쪽’이라고 하고, 왼쪽에 있는 입자를 관측하면 그 위치를 ‘왼쪽’이라고 할 것이다. 반면 양자역학에 따르면 입자는 오른쪽과 왼쪽의 ‘중첩’ 상태에 놓일 수 있다. 하지만 우리는 결코 이 중첩 상태를 경험하지 못하며, 언제나 ‘오른쪽’ 또는 ‘왼쪽’이라고 관측한다. 입자의 위치를 측정하고 나면, 우리는 ‘오른쪽’과 ‘왼쪽’ 가운데 오직 하나를 경험하며, 다른 경험은 결코 하지 못한다.

양자역학과 우리의 경험을 조화시키기 위해 양자역학에 대한 여러 해석이 제안되었다. 시간이 지남에 따라 우주가 여러 가지로 쪼개진다고 상상하고 여러 가지로 쪼개진 각각을 ‘가지’라고 하자. 이제 양자역학의 해석으로 다음 두 해석만 있다고 가정한다. 하나는 가지 치는 것을 허용하지 않는 ST 해석이고, 다른 하나는 이를 허용하는 MW 해석이다. 오직 두 해석만 있기 때문에 한 해석이 참이면 다른 해석은 거짓이다. 우리의 경험은 두 해석 중 무엇을 확증하는가?

알려졌듯이, 입자의 위치를 관측할 때 ‘오른쪽’이 관측될 확률과 ‘왼쪽’이 관측될 확률은 1/2로 동일하다. 이는 다음과 같이 표현될 수 있다.

	가지1	가지2
ST	‘오른쪽’ 또는 ‘왼쪽’이 관측되지만, 둘 다 동시에 관측될 수는 없다.	
MW	‘오른쪽’이 관측된다.	‘왼쪽’이 관측된다.

입자를 관측한 결과 ‘오른쪽’이 관측되었다고 가정하자. 이는 다음과 같은 증거 R이 주어졌음을 뜻한다.

R: 관측된 입자의 위치가 ‘오른쪽’인 가지가 존재한다.

이제 다음 정의를 받아들이자. ‘증거 E가 가설 H를 확증한다’는 것은 ‘가설 H가 참인 조건에서 증거 E가 참일 확률이 가설 H가 거짓인 조건에서 증거 E가 참일 확률보다 더 크다’는 것을 의미한다.

ST 해석과 MW 해석을 가설로 간주할 때 증거 R이 이들 가설을 각각 확증하는지 따져보자. ST가 참인 조건에서 R이 참일 확률은 1/2이다. 왜냐하면 ST가 참인 조건에서는 가지가 하나밖에 없고, 가지가 하나밖에 없는 우주에서 ‘오른쪽’이 관측될 확률은 1/2이기 때문이다. 반면 ST가 거짓인 조건, 즉 MW가 참인 조건에서 R이 참일 확률은 1이다. 왜냐하면 MW가 참이라는 조건에서는 두 개의 가지가 있고 이 중 하나에서는 반드시 ‘오른쪽’이 관측되기 때문이다. 비슷한 방식으로 우리는 MW가 거짓인 조건에서 R이 참일 확률이 얼마인지도 알아낼 수 있다. 따라서

이제 ‘왼쪽’이 관측되었다면 어떻게 될까? 이는 다음과 같은 증거 L이 주어졌음을 뜻한다.

L: 관측된 입자의 위치가 ‘왼쪽’인 가지가 존재한다.

ST가 참인 조건에서 증거 L이 참일 확률은 1/2이다. 왜냐하면 ST가 참인 조건에서는 가지가 하나밖에 없고, 가지가 하나밖에 없는 우주에서 ‘왼쪽’이 관측될 확률은 1/2이기 때문이다. 반면 ST가 거짓인 조건, 즉 MW가 참인 조건에서 L이 참일 확률은 1이다. 왜냐하면 MW가 참인 조건에서는 가지가 두 개가 있고, 두 가지 가운데 하나에서는 반드시 ‘왼쪽’이 관측되기 때문이다.

지금까지의 논의를 종합할 때 우리는 ㉠ 흥미로운 결론에 도달한다.

문 39. 위 글의 빈칸에 들어갈 진술로 가장 적절한 것은?

- ① R은 ST와 MW를 모두 확증한다.
- ② R은 ST와 MW 중 어느 것도 확증하지 못한다.
- ③ R은 ST를 확증하지 못하지만 MW는 확증한다.
- ④ R은 ST를 확증하지만 MW는 확증하지 못한다.
- ⑤ R이 ST와 MW 중 하나를 확증하지만 어느 것인지는 알 수 없다.

문 40. 위 글의 ㉠으로 가장 적절한 것은?

- ① 양자역학의 한 해석이 확증되면 다른 해석도 확증된다.
- ② 우리의 모든 경험이 확증하는 양자역학의 해석은 없다.
- ③ 우리의 경험이 다르면 그 경험이 확증하는 양자역학의 해석도 다르다.
- ④ 특정한 경험은 양자역학의 두 해석을 모두 확증하거나 모두 확증하지 못한다.
- ⑤ 어떤 경험을 하든지 우리의 경험은 양자역학의 특정한 해석 하나만을 확증한다.