이 세상을 시뮬레이션 할 수 없는 이유

김재우

컴퓨터 게임을 하다보면 일반적으로 게이머가 조종하는 게임의 주인공 외에도 다양한 환경들이 있음을 알 수 있다. 나는 가끔씩 우리가 사는 세상도 어쩌면 게임이 아닐까? 하는 생각이 든다. 게임에서 승리하는 기준이 무엇인지는 사람에 따라 다르겠지만, 본인을 주인공으로 삼고 주변 사람들과 환경을 게임처럼 생각한다면, 오픈 월드 장르의 게임과 비슷하게 생각할 수 있다.

그렇다면 우리가 사는 세상도 어딘가에 있는 거대한 컴퓨터에 의해 시뮬레이션 되는 것이 아닐까? 그리고 실제로 그런 컴퓨터와 프로그램을 개발할 수 있을 것인가? 이 의문은 지금까지 여러 사람들에 의해 던져진 바 있다. 미국의 사업가 일론 머스크(Elon Musk) 또한 우리가 사는 세상이 시뮬레이션일 확률이 99% 이상이라고 발언한 바 있다. 하지만 나는 세상이 우리가 아는 컴퓨터로는 시뮬레이션이 가능한지는 결정할 수 없으며, 아마도 불가능하다고 생각한다. 이 글에서는 이 질문에 대한 답을 조금 더 깊이있게 탐구해보고자 한다.

이 세상이 시뮬레이션 가능한지 알 수 없거나, 불가능함을 설명하기 위해 우리가 말하는 "컴퓨터"를 어떻게 정의할지에 대해 생각해보자. 현대 컴퓨터의 기본적인 구조를 제시한 사람은 영국의 수학자 앨런 튜링(Alan Turing) 이다. 앨런 튜링은 "튜링 기계" 라고 하는 현대 컴퓨터의 모태가 되는 기계를 제안했다. 튜링 기계는 무한히 긴 자기 테이프와, 그 자기 테이프를 읽고, 쓰고 움직일 수 있는 헤드로 이루어진다. 튜링 기계의 헤드는 유한 상태 기계(finite state machine)이며, 헤드가 다음에 무슨 동작을 할지는 헤드의 상태와 자기 테이프의 상태에 의해 결정된다. 다시 말해, 완전한 상태의 튜링 기계는 자기 테이프의 상태, 헤드의 상태에 의해 항상 하나의 다음 동작을 정할 수 있으며, 이를 "결정론적 튜링 기계" (Deterministic Turing machine) 이라고 한다. 여기서 한 가지 제한을 풀어서, 자기 테이프와 헤드의 상태가 주어졌을 때, 다음으로 할 수 있는 동작이 여러 개인 기계가 있다고 하자. 이러한 컴퓨터는 그 여러 가지 동작을 동시에 실행해서, 여러 경우의 수를 동시에 처리할 수 있다. 이를 우리는 비결정론적 튜링 기계(Non-deterministic Turing machine) 이라고 부른다. 비결정론적 튜링 기계로는 동시에 여러 경우의 수를 처리할 수 있어 결정론적 튜링 기계로 지수 시간이 걸리는 문제를 다항 시간에 계산할 수 있다. 양자컴퓨터는 대표적인 비결정론적 튜링 기계이다. 양자는 동시에 여러 가지 상태를 가질 수 있으므로 일반적인 컴퓨터로 엄청난 시간이 걸리는 일들을 훨씬 빠르게 처리할 수 있다.

튜링 기계로 풀 수 있는 문제라면, 이론상 무한한 크기의 메모리가 있을 때, 컴퓨터로도 풀 수 있다. 우리가 사는 세상이 시뮬레이션 가능한지 알고 싶다면, 무한한 크기의 자기 테이프에 이 우주의 모든 정보가 주어졌을 때, 다음 상태를 정확히 알 수 있으며, 결론적으로 세상 시뮬레이션 프로그램이 언젠가 끝나게 될지, 그렇지 않을지도 알 수 있어야 한다. 이것에 대해 생각해보기 위해 먼저 문제의 정의를 명확히 하겠다. 시뮬레이션을 하나의 프로그램이라 가정하고, 우리가 사는 세상을 시뮬레이션하여 세상이 종말에 이른다면, 시뮬레이션 프로그램은 그 종말점을 출력해야 한다. 만약 프로그램이 영원히 종료되지 않는다면, 프로그램은 "이 프로그램은 끝나지 않음" 이라는 메시지를 출력한다고 하자. 우리가 결정론적, 또는 비결정론적 튜링 기계인 컴퓨터를 가지고 있다면 우리의 의문은 "우리 세상의 현재 상태가 주어졌을 때, 이 세상이 종말하여 하나의 결정적인 상태가 되는지, 혹은 알 수 없는지를 알려주는 튜링 기계를 위한 프로그램을 만들 수 있는가?" 에 대한 답으로 귀결된다.

이 명제에 답하기 전에 먼저 이것이 증명 가능한지에 대해 생각해보자. 수학이 완전한 학문이라면 이 또한 반드시 증명이 가능할 것이다. 역사적인 수학자 쿠르드 괴델 (Kurt Godel)은 수학의 완전성에 대해 탐구하였는데, 안타깝게도 그는 수학에는 증명 불가능한 명제가 존재한다고 결론내렸다. 괴델은 이를 보이기 위해 "괴델 수(Godel number)" 라는 개념을 제안하였다. 괴델은 수학적 증명에 사용되는 모든 기호에 고유의 숫자(괴델 수)를 부여하였다. 이를 통해 어떠한 증명이나 복잡한 수식도 하나의 고유한 괴델 수로 표현할 수 있는데, 어떠한 수식의 순서에 각각 소수를 부여하고, 그 소수에 각 기호를 지수승을 취하여 복잡한 수식을 괴델 수 하나로 표현하는 방법을 사용한다. 예컨데, 1 + 1 = 2 라는 식을 쓰고, 1에 대한 괴델 수를 10, ’+; 에 대한 괴델 수를

11, '=' 에 대한 괴델 수를 5, 2에 대한 괴델 수를 13 이라고 하자. 그러면 이 식에 대한 괴델 수는, $2^{10} * 3^{11} * 5^{10} * 7^{12} * 11^{13}$ 이 된다. 보다시피 무지막지하게 큰 수일 것이다. 괴델은 수학의 불완전성을 이 괴델 수를 이용해 증명하였다. y의 괴델 수가 17 이라고 가정했을 때, $\neg\exists Dem(x, sub(a, 17, a))$ 라는 식을 정의하자 (여기서 Dem(a, b)는 a가 b에 대한 증명이라는 의미이며 sub(x, 17, x)는 괴델 수 x를 가지는 식에서 괴델수가 17인 부분을 x로 치환한다는 의미이다(substitution)). 이 식의 괴델 수를 g 라고 하자. 그리고, $\neg\exists Dem(x, sub(y, 17, y))$의 괴델 수는 a라고 하자. 그렇다면, $sub(a, 17, a)$의 괴델 수는 g가 된다. 왜냐하면 $sub(a, 17, a)$의 정의대로 $\neg\exists Dem(x, sub(y, 17, y))$ 에서 괴델 수가 17인 y를 a로 대체한다면, 원래 우리가 괴델 수를 g라고 정의했던 식과 같아지기 때문이다. 따라서 이 문제를 증명한다면, g를 증명하는 꼴이 되어 버리는데, 이 명제 g의 정의 자체가 "g는 증명이 불가능하다" 이다. 명제 g는 의미상 참이지만, 증명할 수 없다. 따라서 수학에는 항상 참인 명제이지만, 증명할 수 없는 명제가 존재한다.

괴델의 불완전성에 관한 증명은 우리의 질문이 어쩌면 대답이 불가능할지 모른다는 점을 알려준다. 수학이 완전하지 못하기 때문에 세상이 시뮬레이션 가능한 무언가인가? 에 대한 대답도 증명이 없을 수 있다는 점을 알려준다. 세상이 시뮬레이션 가능한지 알 수 없는 이유는 하나가 더 있다. 앞서 정의한 명제에서, 세상을 시뮬레이션 하기 위해서는 세상의 "마지막 상태" 가 어떤 상태인지 결정할 수 있어야 한다고 했다. 그렇다면, 모든 문제가 마지막 상태를 튜링 기계를 통해 결정할 수 있을까? 결론부터 말하자면 불가능하다. 만약 우리가 어떠한 프로그램이 주어졌을 때, 이것이 영원히 실행이 될지, 언젠가 종점에 도달할 지 알려주는 프로그램 'g'가 있다고 가정하자. 그렇다면 이 프로그램 'g'을 이용하여 프로그램의 출력이 "종점에 도달" 이면 영원히 실행되며, "종점에 도달하지 않음" 이 들어온다면 멈추는 프로그램도 정의할 수 있을 것이다. 이 프로그램을 'h' 라고 부르자. 그럼 만약 h를 h에 넣는다면, 프로그램은 어떻게 동작해야 할까? h가 영원히 실행된다면, h가 사용하는 프로그램 g는 "영원히 실행됨" 을 출력하므로, h는 멈춰야 한다. 반대로, h가 언젠가 종료된다면, g가 "종점에 도달"을 출력하므로, h는 영원히 실행되어야 한다. 이는 모순이므로, 우리는 모든 프로그램을 종점에 도달할지 결정하는 것은 불가능하다.

이제까지 우리는 수학의 모든 문제가 증명될 수도 없으며, 모든 프로그램이 종점에 도달할지, 그렇지 않을지 파악하는 것 역시 불가능하다는 것을 알게 되었다(더 정확히 표현하자면, 종점에 도달할지 결정이 불가능한 프로그램도 존재한다는 의미이다). 그렇다면 세상을 시뮬레이션 하는 프로그램은 만들어질 수 있는가? 지금까지 살펴본 대로라면 우리는 이 질문에 대한 답을 찾지 못할지도 모르지만 하나를 고른다면 나는 세상이 시뮬레이션 불가능하다는 쪽을 선택하겠다. 우선, 튜링 기계로 모델링할 수 없는 불확정성(uncertainty)은 자연계에서 흔히 관찰된다. 양자 역학이 대표적인 예이다. 어떠한 물질에서 정확히 그 물질을 구성하는 입자의 상태는 우리는 알 수 없다. 원자를 구성하는 전자를 보아도, 과학자들은 그 전자의 움직임과 방향을 정확히 예측하는 것이 아닌, 확률로써 표현한다. 즉, 세상을 시뮬레이션 할 수 있으려면, 이러한 불완전성을 모델링할 수 있어야 한다. 무한한 길이의 자기 테이프에 이 세상 모든 미립자(particle)들의 상태를 집어넣는다면, 튜링 기계는 이 자기 테이프의 상태와 헤드의 상태만을 가지고 다음 동작을 결정해야한다. 그러나, 이것이 결정론적이지 않다면, 이는 튜링 기계로 풀 수 없는 문제이다.

여기까지 읽었다면 한 가지 의문이 들 수 있다. 그렇다면 랜덤(random)한 수를 만들어내는 기능은 어떠한가? 랜덤한 수를 생성하면 확률적으로 무언가를 처리할 수도 있지 않은가? 하지만 이 랜덤 알고리즘 또한, 정해진 규칙대로 만들어져 랜덤함을 흉내내는 것이지 진정한 의미의 랜덤 알고리즘이 아니다. "랜덤"을 튜링 기계로 모델링하려면 정말로 랜덤한 무언가를 외부에서 주어야 한다. 이 외부에서 빌린 힘이 없다면 랜덤 알고리즘은 매번 같은 순서로 숫자를 출력한다. 실제로 랜덤 알고리즘은 무작위로 주어진 임의의 수를 요구한다. 프로그래머들은 이런 랜덤 함수를 구현하기 위해 '프로그램이 실행된 시간'과 같은 외부적인 변수를 넣어서 랜덤을 모델링하고는 한다. 하지만 이는 외부 환경의 힘을 빌린 것이지, 튜링 기계의 자체적인 기능은 아니다.

따라서 세상은 일반적인 컴퓨터로는 시뮬레이션 될 수는 없다. 만약 어떤 고지능 외계 생명체가 튜링 기계의 가정을 뛰어넘는 컴퓨터를 가지고 있다면 모르겠지만, 현대 컴퓨터의 기반인 튜링 기계로는 세상을 시뮬레이션 할 수 없거나, 정말로 세상이 시뮬레이션 가능한지 알 수 없다는 결론에 도달한다.