

Subject: prove pnp problem

Author: Hyunho Song

Abstract

As you know, the pnp problem is related to the time complexity of the algorithm. I delved into this problem with logic. With mathematical logic I solved this problem. In this I want to dig deep into the logic of the pnp problem and prove that $p \neq np$. This proof method is simple. No algorithm can be used to reduce an NP-complete problem to a polynomial.

I will explain why the NP problem cannot be turned into a p problem with logic laws.

본문

pnP 문제는 알다시피 알고리즘의 시간 복잡도와 관련 된 문제다. 나는 이 문제를 논리로 파고들었다. 수학적 논리로 나는 이 문제를 해결했다. 나는 이것에서 pnp 문제의 논리에 깊게 파고들어 $p \neq np$ 라는 것을 증명하고자 한다. 이 증명 방법은 간단하다. NP 완전 문제에 어떤 알고리즘을 대입해도 다항식으로 환원 할 수 없다. 왜냐하면 다항식과 np문제는 상극이다. 왜냐하면 np 문제는 비결정인 문제 특성상 운에 의존하기 때문이다. 즉 운에 의존하는 문제는 어떠한 알고리즘을 가져와도 운이라는 성질이 변하지 않는다. 그런데 np문제의 비결정적인 것과 운은 매우 비슷하기에 운을 다항식으로 바꾸지 못하는 것처럼 np 문제도 못 바꾸는 것이다.

운이 p문제로 못 바뀌는 이유를 논리 법으로 설명해보겠다.

일단 자세한 증명 과정을 보여주겠다. 비결정적인 np문제는

운과 관련이 깊다. 왜냐하면 np 문제의 일련의 과정 중 하나가 운이기 때문이다. 여기서 운이라는 것은 확률 그 자체로 p 문제 같은 경우 그냥 아무렇게나 해도 np 문제다. 하지만 운이 아닌 것을 운인 것으로 바꿀 순 있어도 운인 것을 운이 아닌 것으로 바꿀 수 없다. 그를 위해 간단한 논리 게임을 하겠다.

지금 상자 안에 빨간 공 99개와 파란 공 한개가 들어있다. 그리고 컴퓨터는 그것을 모른다.(여기서 튜링 기계를 편의상 컴퓨터라고 부른다.) 컴퓨터는 빨간 공이 아닌 공이 들어 있는지 확인하는 시간은 다항시간이 걸린다. 즉 공 한 개의 색상을 확인하는데 1초가 걸린다면 컴퓨터를 망가트리지 않는 이상 100초가 걸린다. 이 문제에서는 모든 공을 확인해야 하는 문제기에 더도 덜도 아닌 100초가 걸린다. 그런데 문제가 파란 공을 뽑는 문제라면? 공을 뽑는데 1초가 걸린다면 파란 공을 뽑는데 걸리는 시간은 1~100초 일 것이다. 참고로 이것은 지뢰가 파랑 공이고 나머지가 공이라는 것 그리고 재미 없다는 것을

제외하면 모두 지뢰 찾기와 동일한 NP 완전 문제다. 문제는 다항식 문제하면 무조건 100초가 걸려야한다. 그런데 어떤 신이 와서 물리 법칙을 바꾸지 않는 이상 상자 속의 파란 공을 100초 또는 1초 등 정확한 100%의 확률로 걸리는 시간을 계산하고 그리고 계속 그 시간만 걸리게 하는 자는 존재하지 않는다. 즉 파란 공이 나올 확률1%로라는 확률이 무의미해진다. 그리고 P문제가 걸리는 시간을 예상 못해도 다시 그 알고리즘으로 계산 했을 때 동일 시간 걸린다. 그런데 파란 공을 첫 번째 10초 안에 뺐으면 그 다음 번도 10초 안에 뺄 수 없다. 즉 P문제의 계산 시간을 못 계산한다고 말을 나도 확신을 가지고 말할 수는 없지만 계산 과정 재계산 했을 때 같은 시간이 걸리는 것은 똑같다고 확신해서 말할 수 있다. 왜냐하면 다항시간 걸린 알고리즘의 계산 시간이 옆집 꼬마가 오렌지 주스 컴퓨터에 쏟지 않은 이상 걸리는 시간은 똑같은 것이다. 즉 NP 문제는 재계산을 했을 때 확률적인(비결정적인) 선택으로 인해 걸리는 시간이 달라진다.(물론 운이 좋아 똑같은 시간이 걸릴 수

도 있다. 만약 똑같은 시간이 걸렸다면? 당장 로또 사러 가라.)
즉 np가 p문제가 되려면 당장 신 찾아가서 “신이시여. 제발 확률을 없애어주소서. 복권 뽑기 참 어렵습니다.” 이래야한다.
즉 확률이라는 것 밑에 있는 NP 완전 문제는 절대로 그 확률이라는 것을 깨고 필연이 될 수 없다. 즉 다항시간이 걸린다는 것을 알아야 하는데 애초에 NP 완전 문제의 다항시간 얼마나 걸리는지 알려고 하는 것이 확률이라는 것 때문에 불가능하다.
물론 예외도 있다. 하지만 그 경우에는 반복적인 알고리즘으로 알고리즘 과정의 개수의 배수다. 즉 알고리즘 과정이 7개면 7의 배수로 된다는 것이다. 하지만 NP 완전 문제는 알고리즘의 개수의 배수가 될 수 없다. 즉 알고리즘 과정 한 번 거치는 것이 N초면 $N \times X$ = 알고리즘의 시간 과정이다. 그리고 X는 계산의 크기다. 그리고 여기서 중요한 것은 이 알고리즘은 다시 한번 그 알고리즘을 실행 시켰을 때다. 첫 번째 때 10초 걸렸으면 다시 했을 때 10초다. 하지만 NP 문제는 이렇지 않다. 예를 들어 여덟 쿼 문제에서는 필연적으로 p문제와 달리 비결정

적인 선택을 해야 한다. 즉 알고리즘으로 구할 때 생기는 비정적인 선택으로 인한 변수로 인해 알고리즘 개수가 무엇의 배수가 될 수도 없다. 그리고 알고리즘이 확률적인 (비정적인) 선택으로 인해 알고리즘이 꼬여 버린다. 즉 다항식으로 바꿀 수 없다. 그리고 100번 양보해서 안다고 해도 내가 위해서 말한 상자의 역설로 인해 다시 계산 했을 때 걸린 시간이 다를 것이다. 무엇보다 결정적인 증거로 경우의 수를 아무리 생각해도 확률적이어서 선택이 운이 될 수 밖에 없기에 확률적인 선택으로 인해 알고리즘이 꼬여보인다. 즉 다항시간을 알 수 있는 NP가 등장한다고 해도 어찌 되었건 알고리즘이 꼬여버린다.(물론 다항시간을 알 수 있는 NP는 등장 할 수가 없다. 사실 다항시간을 알 수 있다면 애초에 꼬일 일도 없다.)

만약 NP 완전 문제를 P 문제로 환원하고 싶으면 신 찾아가서 세상 설계도 찢으면서 이렇게 말해야한다. “복권 뽑기도 힘들고, RSA 암호도 풀기 어렵고. 고등학교 수학 한 단원이라도 없애

고 싶으니까 확률이랑 비정적인 알고리즘 없애고 세상 다시 만
들어주세요.” 이래야한다. 그렇기에 나의 상자의 역설과 다른
알고리즘 과정의 시간 과정 차이로 $P \neq NP$ 증명 되었다.