

정지 문제

2023.07.16

문제 개요

모든 프로그램은 다음 둘 중 하나에 해당 한다

1. 입력값을 넣고 실행하면, 유한한 절차를 거친 다음, 리턴값을 내놓는다.
2. 입력값을 넣고 실행하면, 무한 루프에 빠진다

소스코드를 보고 이를 구분할 수 있으면 유용하지 않을까?

정지 문제

무한 루프를 검토하는 프로그램을 H 라고 하자,
 H 는 모든 입력에 정확한 결과를 리턴하는 프로그램이다.

“ H 를 만드는 것이 가능한가?”



H를 만드는 것은 불가능하다

귀류법을 통해 증명 할 수 있다.

전제 : H는 모든 입력에 정확한 결과를 리턴하는 프로그램

1. H를 만드는 것이 가능하다고 가정한다.
2. H가 잘못된 결과를 리턴하는 경우를 찾아낸다.
3. 전제에 모순되므로, H를 만드는 것은 불가능하다.

H 구현 가정

```
1  def halt(f, x):  
2      # H를 구현하려고 시도하는 코드  
3      # ...  
4      # ...
```

- H를 구현하는 함수 halt
 - f 는 프로그램의 소스코드
 - x 는 f 에 입력할 입력값

H 구현 가정

```
6 def test(i):  
7     while i > 0:  
8         print("I'm in a loop!")  
9     return "I halted!"
```

- halt(test, 3) -> false
- halt(test, -1) -> true

H가 잘못된 결과를 리턴하는 경우

```
11 def foo(x):  
12     if halt(x, x):  
13         while true:  
14             print("I'm in an infinite loop!")  
15     else:  
16         return "I halted!"  
17  
18 foo(foo) ## ????
```

- foo(foo)는 halt(foo, foo) 실행
- halt(foo, foo)는 어떤 결과를 리턴할까?

halt(foo, foo) -> True

```
11 def foo(x):  
12     if halt(x, x):  
13         while true:  
14             print("I'm in an infinite loop!")  
15     else:  
16         return "I halted!"  
17  
18 foo(foo) ## ????
```

- halt(foo, foo) -> True, 즉 foo(foo)는 정지한다
- 하지만 foo의 내부를 보면 무한루프 한다
- halt(foo, foo) -> False 이어야 한다

halt(foo, foo) -> False

```
11 def foo(x):  
12     if halt(x, x):  
13         while true:  
14             print("I'm in an infinite loop!")  
15     else:  
16         return "I halted!"  
17  
18 foo(foo) ## ????
```

- halt(foo, foo) -> False, 즉 foo(foo)는 무한루프한다
- 하지만 foo의 내부를 보면 정지한다
- halt(foo, foo) -> True 이어야 한다

문제의 의의

세상엔 컴퓨터로 풀기 쉬운 문제가 있고, 풀기 어려운 문제가 있다.

그리고 세상엔 컴퓨터로 풀 수 없는 문제도 있다.

정지 문제는 지극히 추상적인 수학 문제가 소프트웨어의 한계를 규정하는 사례이다.

완벽한 백신 SW는 불가능하다, 완벽한 정적 분석은 불가능하다