



链滴

图灵停机问题

作者: [plus7wist](#)

原文链接: <https://ld246.com/article/1618814617820>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



问题

假设有函数 `halt_on`，能判断函数在给定输入下是否会结束。例如下面的函数不会结束：

```
def sleep_forever():  
    while True:  
        time.sleep(1)
```

下面的函数在输入是任何整数的时候都会结束：

```
def add(a, b):  
    return a + b
```

我们期望 `halt_on(sleep_forever)` 返回 `False`，`halt_on(add, 1, 2)` 返回 `True`。

函数是编程语言的概念，是描述一般问题的图灵机器的概念上的实现，最初提出这个问题是在以图灵机为输入的图灵机上，函数的结束的用词也就改成了“停机”，所以这个问题称作是“图灵停机问题”。

构造

下面构造一个函数 `hack`：

```
def hack(program):  
    if halt_on(program, program):  
        sleep_forever() # this will never return  
    else:  
        return
```

如果 `halt_on(hack, hack)` 是真，那么程序认为 `hack(hack)` 会退出。但是根据 `hack` 的定义，`halt_on(hack, hack)` 是真时，它会进入无限沉睡。

如果 `halt_on(hack, hack)` 是假，问题刚好反过来，根据 `hack` 的定义又说明 `halt_on` 函数产生了错误。

所以 `hack` 的存在就说明，不可能存在 `halt_on` 函数。

类型

当一个值处在一种值的集合内，则说这个值属于此类型。对于求值永远无法结束的表达式（程序、图灵机），我们无法判断取得确切的值，但可以归类为一种特殊的类型，一般叫做 Never Type。Python 的 typing 模块用 `NoReturn` 表示它。

当一个值同时可能处在两种类型中，可以称这个值属于两个类型的并类型中，typing 中用 `Union[A, B]` 表示 A 和 B 的并类型。

`halt_on` 函数的输入函数的返回类型，应该是 `Union[NoReturn, T]`，后者是当函数会返回时返回的类型。由于 `halt_on` 不可能存在，所以永远也不能区分出这个函数的输出类型确切地是 T 还是 `NoReturn`。所以 `Union[NoReturn, T]` 永远都应该和 T 行为一致，不妨说 `Union[NoReturn, T]` 就是 T。