

软分第1次作业

姓名：陈格非

学号：2100012986

# 2023/09/12 课程介绍 [2023/09/19 截止]

完成课程介绍讲义中的练习并提交pdf文件

## 1.

停机问题的证明定义在没有输入的函数上，能否改成在带输入的函数上？注意这时 `Halt(p, i)` 函数接受两个参数，其中 `i` 是输入。

回忆原证明：

停机问题证明

- 假设存在停机问题判定算法： `bool Halt(p)`
- `p` 为特定程序
- 给定某邪恶程序

```
void Evil() {  
    if (!Halt(Evil)) return; else while(1);  
}
```

- `Halt(Evil)` 的返回值是什么？
- 如果为真，则 `Evil` 不停机，矛盾
- 如果为假，则 `Evil` 停机，矛盾

注：关于这个证明，群里有同学指出：

我不明白的地方是，如果 `Evil` 没有输入，它就必须自己输出自己的编码。[晕]

而自己输出自己的编码似乎需要一个非平凡的构造，证明中似乎并没有解释。

好在，这个证明的正确性不影响这个习题的证明，所以我们将不讨论原证明的正确性，直接用类似的方法构造。

对于原问题，可以参考这个 StackExchange 回答：[computability - Halting problem without input? - Computer Science Stack Exchange](#)

或者这个视频：[ECS 120 9a:3 no-input halting problem is undecidable - University of California, Davis](#)（思路：将带输入停机问题的归约成无输入的停机问题）

反证法：假设存在 `Halt(p, i)` 满足条件。

则构造一个函数 `Evil(p)`：

```
void Evil(p) {
    if (!Halt(p, p)) return; else while(1);
}
```

考虑 `Halt(Evil, Evil)` 的取值，即可导出矛盾：

- 假若 `Halt(Evil, Evil)` 为真，则 `Evil(Evil)` 反而会不停机，与 `Halt(p, i)` 的定义矛盾。
- 假若 `Halt(Evil, Evil)` 为假，则 `Evil(Evil)` 反而会停机，与 `Halt(p, i)` 的定义矛盾。

因此，`Halt(p, i)` 不存在。证毕。■

## 2.

假设我们把符号分析的抽象域改成 {自然数、负、罣} 三个值，其中自然数表示所有正数和零，请写出加法和除法的计算规则，并给出一个式子，在该抽象域上得到的结果不如原始分析精确。■

### 加法

↓+→	自然数	负	罣
自然数	自然数	罣	罣
负	罣	负	罣
罣	罣	罣	罣

### 除法

↓/→	自然数	负	罣
自然数	罣	罣	罣
负	罣	自然数	罣
罣	罣	罣	罣

### 例式

考虑式子 `2/1 = 2`，它在原始分析上表示为 `正/正=正`，在本题的抽象域中为 `自然数/自然数=罣`。

这个式子在该抽象域上得到的结果不如原始分析精确。■