

软件分析技术作业1

杜思娴 2000017403

第一题

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。

如果存在停机问题的判定算法 `Halt(p, i)` 判定对于输入*i*程序*p*是否会终止，则我们可以构造这样一个函数：

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

如果 `Halt(Evil, i) == True`, 说明 `Evil(i)` 根据判定算法会停机，但实际上在 `Evil(i)` 函数中会陷入 `while(1)` 循环，不会停机，矛盾！

如果 `Halt(Evil, i) == False`, 说明 `Evil(i)` 根据判定算法不会停机，但实际上在 `Evil(i)` 函数中会直接 `return`，停机，矛盾！

因此停机问题的证明可以改成在带输入的函数上，证明出不存在一个算法能回答停机问题的所有实例。

第二题

假设我们把符号分析的抽象域改成{自然数、负、糅}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。

以下的运算符 (+, /) 均为定义在抽象域上的运算

$$\begin{aligned} a + b &= \begin{cases} \text{自然数} & a = \text{自然数}, b = \text{自然数} \\ \text{负} & a = \text{负}, b = \text{负} \\ \text{糅} & \text{其他情况} \end{cases} \\ a/b &= \begin{cases} \text{自然数} & a = \text{负}, b = \text{负} \\ \text{糅} & \text{其他情况} \end{cases} \end{aligned} \tag{1}$$

对于式子 $a/b + b$, ($a = \text{自然数}, b = \text{负}$), 通过抽象域上的结果: $a/b = \text{糅}$, 因此 $a/b + b = \text{糅}$

原始分析: 由于 $a/b \leq 0, b < 0$, 因此 $a/b + b$ 为负

该式在抽象域上得到的结果不如原始分析精确