

1. 可以。给定某邪恶程序  $Evil(i)$ ，不妨设  $j$  为一个  $Evil$  函数的可行输入。那么，~~void~~  $Evil(i)$  定义为：

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

那么，当  $i$  取  $j$  时，若  $Halt(Evil, j)$  为真，则  $Evil(j)$  不停机，矛盾；  
若  $Halt(Evil, j)$  为假，则  $Evil(j)$  停机，矛盾。

~~即  $Halt(Evil, i)$  为停机问题判定算法。~~

2. 加法计算规则：

自然数 + 自然数 = 自然数	负数 + 自然数 = 疑
自然数 + 负数 = 疑	负数 + 负 = 负
自然数 + 疑 = 疑	负 + 疑 = 疑
疑 + 自然数 = 疑	疑 + 负 = 疑
	疑 + 疑 = 疑

除法计算规则：

自然数   自然数 = 疑	负   自然数 = 疑
自然数   负 = 疑	负   负 = 自然数
自然数   疑 = 疑	负   疑 = 疑
疑   自然数 = 疑	疑   负 = 疑
	疑   疑 = 疑

对于 ~~这个~~ 式子，原始分析中，当  $b$  为负时， $a$  为正，则  $a/b$  为正， $a$  为零，则  $a/b$  为零，均可判定。

抽象域修议后， $a$  为自然数， $b$  为负时， $a/b$  只能为疑。