

- 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 $\text{Halt}(p, i)$ 函数接受两个参数，其中 i 是输入。

可以。假设存在停机问题判定算法 $\text{bool Halt}(p, i)$ ，其中 p 为特定程序， i 为输入。

则给定一个邪恶程序

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

则如果 $\text{Halt}(\text{Evil}, i)$ 为真，则 Evil 不停机，矛盾；如果为 $\text{Halt}(\text{Evil}, i)$ 假，则 Evil 停机，矛盾。所以不存在一个停机问题判定算法。

- 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。

+	自然数	负	糅
自然数	自然数	糅	糅
负	糅	负	糅
糅	糅	糅	糅

/	自然数	负	糅
自然数	糅	糅	糅
负	糅	自然数	糅
糅	糅	糅	糅

式子： A 是自然数， B 是负数，则 $A - B + B$ 这个式子原始分析的结果一定是自然数，但是在这个抽象域上分析的结果则是糅。