2023-09-18 徐一茗 2100012911

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。

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

无论Evil(i)是否会终止，都会推出矛盾。因此不存在这样的判定算法Halt.

假设我们把符号分析的抽象域改成{自然数、负、粑}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。

$$\alpha(i)=\begin{cases}\text{自然数}, & i\text{是0或正整数}\\ \text{负数}, & i<0\\ \text{粑}, & \text{else}\end{cases}$$

，针对+，/操作定义对应的抽象域上的运算⊕，⊘

$$a\oplus b=\begin{cases}\text{自然数}, & a=\text{自然数},\ b=\text{自然数}\\ \text{负数}, & a=\text{负数},\ b=\text{负数}\\ \text{粑}, & \text{else}\end{cases}$$

$$a\oslash b=\begin{cases}\text{自然数}, & a=\text{负数},\ b=\text{负数}\\ \text{粑}, & \text{else}\end{cases}$$

式子：$1/1$，原始分析：$1/1=1$，结果为自然数；而抽象域上结果为粑