

1. 设有带输入的停机问题判定算法:

$\text{bool Halt}(p, i)$

又设有某邪恶程序 Evil, x 为被试程序

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

则当 $\text{Halt}(x, x)$ 返回真, 则 Evil 不停机, 矛盾.

当 $\text{Halt}(x, x)$ 返回假, 则 Evil 停机, 矛盾.

2. {自然数、负、数}

自然数 = {所有正数, 0}

负 = {所有的负数}

数 = {所有整数, NaN}

ADD ①	自然数	负	数	Div ②	自然数	负	数
自然数	自			自然数	自	数	数
负	数	负		负	数	自	数
数	数	数	数	数	数	数	数

设有映射将整数映射到抽象域:

$$\alpha(i) = \begin{cases} \text{自然数} & i \geq 0 \\ \text{负} & i < 0 \end{cases}$$

定义在抽象域上的运算 \oplus, \odot (如上表), 就可在抽象域上, 在不算出精确值的情况下带中符号. 如: $0 \div (-2)$ 可作为 $\alpha(0) \odot \alpha(-2)$

得 自 \odot 负 = 数

而原始分析为 零 \odot 负 = 零

由于新抽象域得到了结果数, 显然不如原始分析精确.