> 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?

可以，假设 `halt` 可实现，考虑如下函数

```
void evil (func f) {
    if (halt(f,f)) while(1);
    return;
}
```

那么 `evil(evil)` 停机等价于 `halt(evil,evil) = false`（`evil` 的定义），等价于 `evil(evil)` 不停机（`halt` 的定义），矛盾。故不存在满足要求的 `halt`。

> 假设我们把符号分析的抽象域改成{自然数、负、槑}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。

加法：

	自然数	**负**	**槑**
自然数	自然数	槑	槑
负	槑	负	槑
槑	槑	槑	槑

除法（每格所在行为被除数，列为除数）：

	自然数	**负**	**槑**
自然数	槑	负	槑
负	槑	自然数	槑
槑	槑	槑	槑

考虑式子 `a/(b/c)`，所有变量均为负数。则抽象域分析结果为“槑”；但实际结果应为负数。