

陆初

脑子不太好用的普通人。

对于SICP中1.41习题的思考

DBQ，我学着学着ObjC又去摸鱼看起了SICP（逃）

今天提起的这个练习1.41看上去很简单，但实际上（至少在我看来）还挺难的，第一眼很容易做错。

思考了一晚上，碰到了另一个问题才刚好想起来这个问题似乎有突破口，于是我就来水一篇博客了。

（阅读以下内容或许需要懂得Lisp的基本语法）

Lisp的语法其实很简单，就是由很多括号组成的。

例如：(foo a b c)

foo 就是函数名，a b c 是参数名。整个括号会变成该函数的返回值。

今天看到了 1.3 用高阶函数做抽象。其中练习 1.41 是这样子的：

1.41 请定义一个过程 double，它以一个有一个参数的过程作为参数，double 返回一个过程。这一过程将原来那个参数过程应用两次。例如，若 inc 是个给参数加一的过程，(double inc) 将给参数加2.下面表达式返回什么值：

```
((double (double double)) inc) 5)
```

这还用问？3 个 double，也就是 2 的 3 次方，无非就是加 1 执行 8 次嘛！

答案是 13。

是才有鬼了。

很遗憾，答案并不是 13，是 21。

为什么会这样？

首先来看 double 的实现。这个很简单，无非就是把一个函数执行两次就好了。

```
(define (double f)
  (lambda (x)
    (f (f x))))
```

如果只想知道这个表达式的结果的话，特别简单。把定义复制到 IDE 里面，然后再运行就能得到结果了。

```
Welcome to DrRacket, version 7.6 [3m].
Language: racket, with debugging; memory limit: 128 MB.
> (((double (double double)) inc) 5)
21
>
|
```

可是，为什么呢？？？

我仔细想了一晚上也没想出来为什么，直到刚才我遇到了另一个函数递归的问题，才找到一个思路。那就是函数嵌套。

如果用高中数学的情况去想的话，无非就是 $f(f(f(f(x))))$ 这样，对一个值嵌套求多个函数。

而要消去嵌套有两个方向，从内往外或者从外往内。

例如 $g(x) = x + 1$ ， $f(x) = 2x$ ，那么 $g(f(x)) = ?$

如果从内往外，就是 $f(x) = 2x$ ， $g(2x) = 2x + 1$ 。

从外往内的话，就是 $g(f(x)) = f(x) + 1 = 2x + 1$ 。

（其实函数嵌套操作本身就是遵守结合律的，如果有更多层的话还能先求其中一小部分再结合起来一起求，不够在这里说就是题外话了）

那我自己的思路和正确答案的思路差别究竟在哪？

无非就是搞错了函数嵌套的顺序。

如果是

```
((double (double (double inc))) 5)
```

的话，那么答案的确是 13，因为

最里面的 (double inc) 相当于执行两次 inc，所以也就是 +2 操作。

既然已知 (double inc) 就是 +2，那么 (double (double inc)) 就是 (double +2)，+2 执行两次就是 +4。

同理，最外层的 double 就是执行两次 +4，所以答案就是 $5 + 8 = 13$ 。

写成数学（或者说，中缀表达式？）形式的话就是 $d(d(d(inc)))(5)$

那么题目中的

```
((double (double double)) inc) 5)
```

写成中缀表达式就是这样：

$[(d(d(d)))inc](5)$

inc 过程并不是最内部的过程，不能够像刚才那样分析。

那么该怎么分析 $(d(d(d)))inc$ 呢？这是一个什么样的过程？

可以从里往外分析也可以从外往里分析，先尝试从里往外吧。

首先(double double) 的意思很简单，就是把 double 自己执行两次，所以相当于把作为参数的过程执行4次。就记作 d4 吧。(double double) 等价于 d4

那么 (double d4) 就是把 d4 执行两次，((double d4) inc)就是执行 8 次inc，对吗？

不对。

可以回到最开始 double 的定义进行带入，如果参数是 d4 的话……

`((double f) x)` 等价于 `(f (f x))`

那么

`((double d4) x)` 等价于 `(d4 (d4 x))`

问题就在这里。**嵌套的顺序**。

写成这种形式就清晰很多了。`(d4 inc)` 就是把 +1 执行了4次，也就是 +4。

那么 `(d4 (d4 inc))` 就是 `(d4 +4)` 了，意思是把 +4 执行了 4次，也就是 +16。

所以 `(+16 5)` 就是正确答案 21。

做错的核心原因就是，误把 `((double f) x)` 和 `(* 2 (f x))` 等价了。

但其实并不是。前者是 `(f (f x))`，是一个函数的嵌套调用，而右边是乘2。

……当我写到这里的时候，我才突然知道引起这个错误的根本原因是什么。

当它把这个过程命名为 double 的时候，我下意识就会觉得这是一个把目标乘以2的操作……

如果命名为 repeat 的话，或许我就会立刻察觉到这是一个函数嵌套调用，而不是乘以2……

害，算了，可能也只是我笨。总而言之，先睡觉了。



作者：陆初

脑子不太好用的普通人。顺带一提性格也有点古怪。在老妈子和厌世肥宅中来回切换。

[查看陆初的所有文章](#)



《对于SICP中1.41习题的思考》有4个想法

 5:59pm

2020-06-17 09:55

如果只是误把 $((\text{double } f) x)$ 和 $(* 2 (f x))$ 等价了话，好像也得不到 13 的结果🤔？
关键就是在于 $((d (d d)) f)$ 不等于 $(d (d (d f)))$ 而是等于 $(d (d (d (d f))))$ 吧。BTW 王垠之前发了一道练习题就是这道题🤔

 陆初 

2020-06-17 12:55

不愧是jp大佬，一眼就看出问题真正的本质！
从 $((d (d d)) f)$ 展开成 $(d (d (d (d f))))$ 的过程能不能写一下，实在想不出来orz

 Kawaii ... !

2020-06-17 23:25

$(d d)$ 就两个 d 了嘛，前面再加个 d 就四个 d 了嘛（
用 S 表达式比较难描述，你改成用一般的表达式去理解可能会好一点。

 6: 00PM

2020-06-17 13:15

老师太会了！

此站点使用Akismet来减少垃圾评论。 [了解我们如何处理您的评论数据。](#)