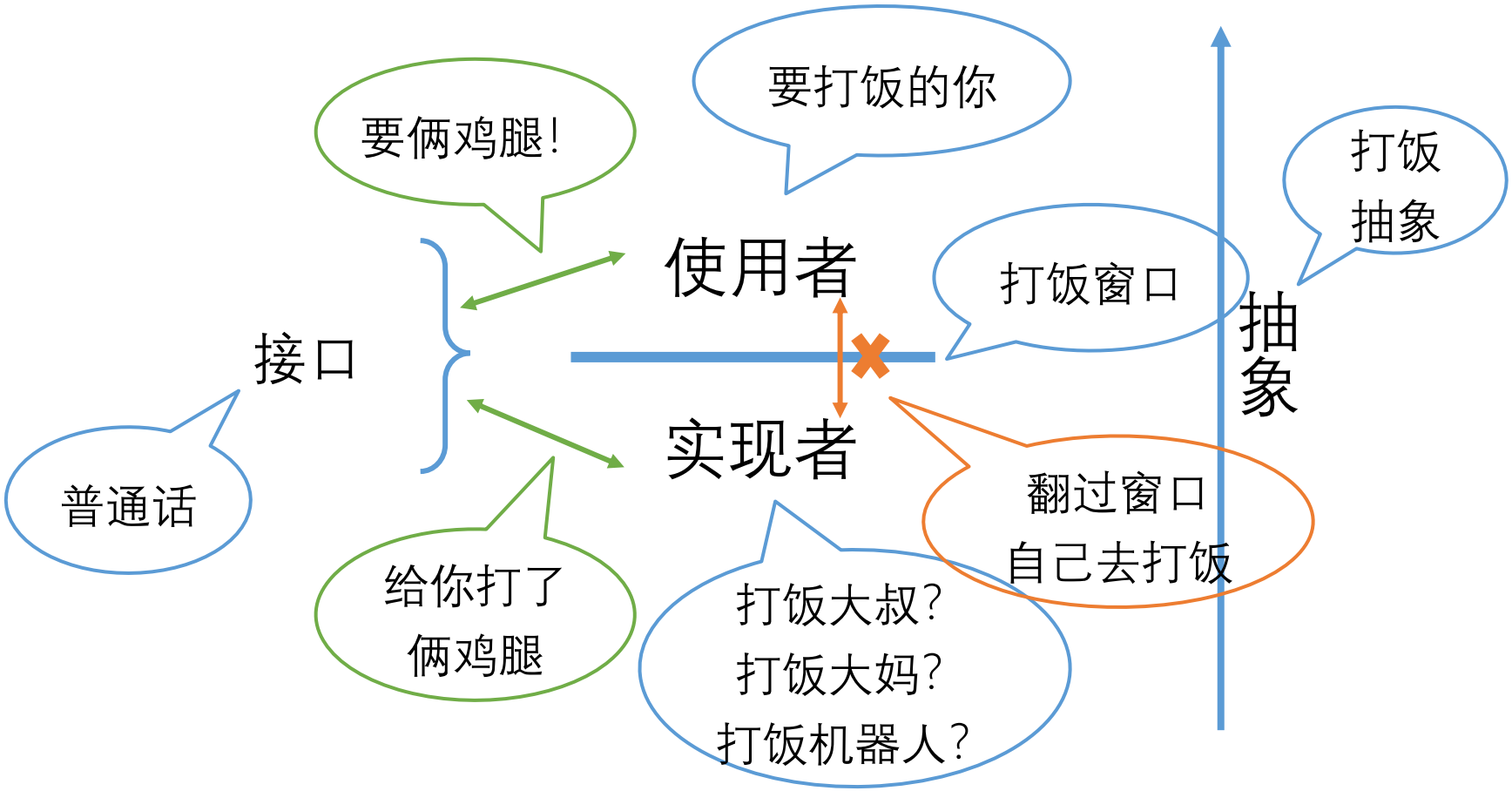
期中复习

陈钦霖 黄奕诚 蒋圣翊 陈智骐 林哲浩
2020.11.19

主要内容

- 抽象！抽象！抽象！
- List拾遗
- 习题选讲

计算机世界里的抽象

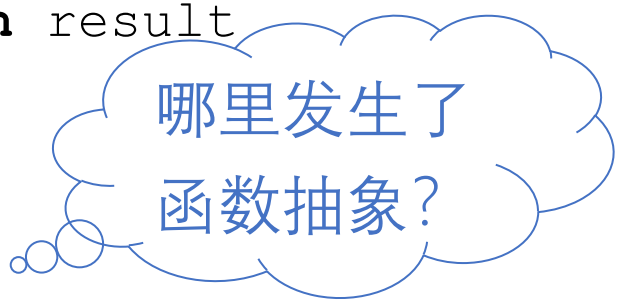


抽象在使用者和实现者之间建立了一堵墙，他们必须通过墙上的接口来沟通。

函数抽象

```
def sum(it):
  result = 0
  for x in it:
    result = result + x
  return result
```

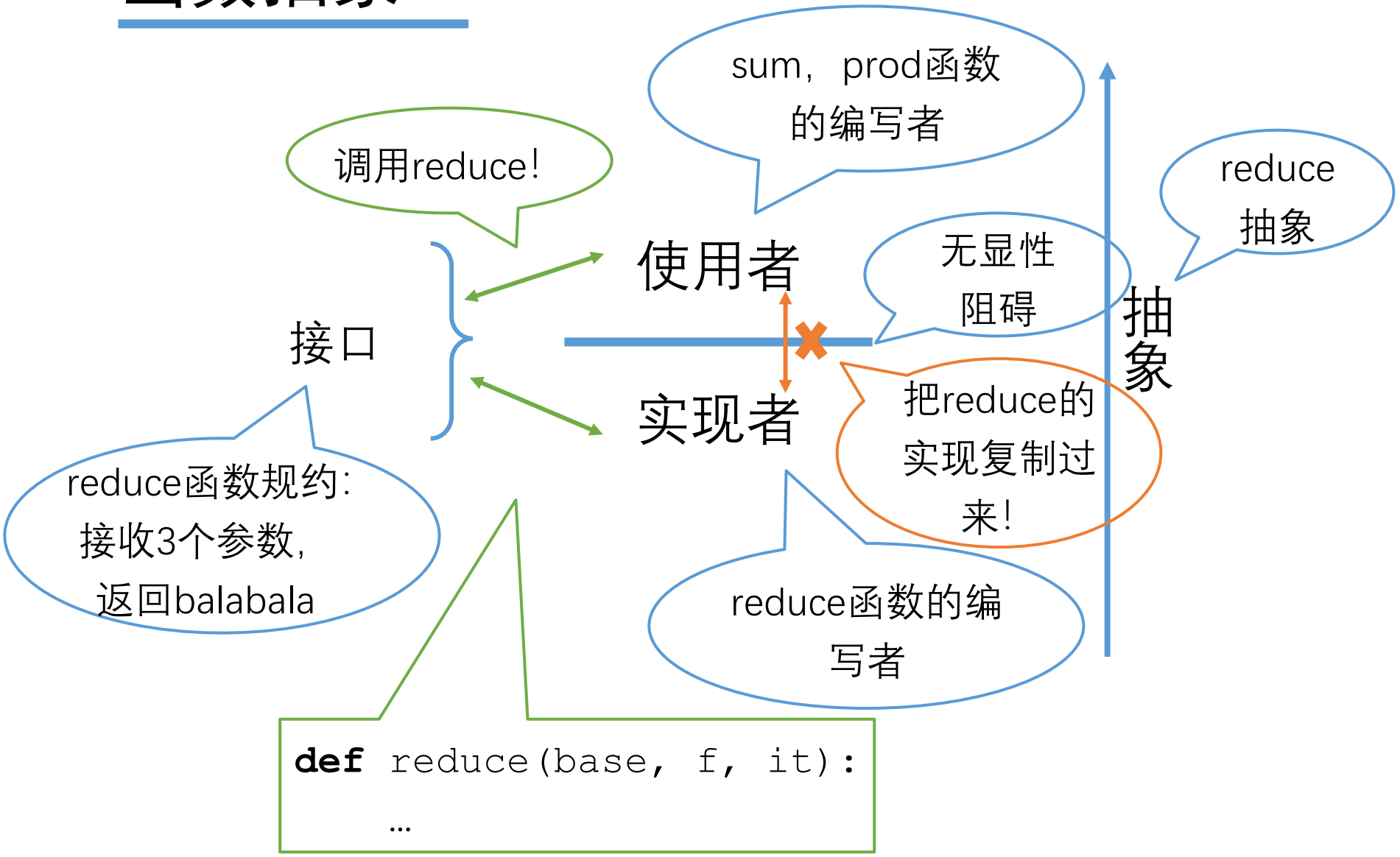
```
def prod(it):
  result = 1
  for x in it:
    result = result * x
  return result
```




```
def reduce(base, f, it):
  result = base
  for x in it:
    result = f(result, x)
  return result
```

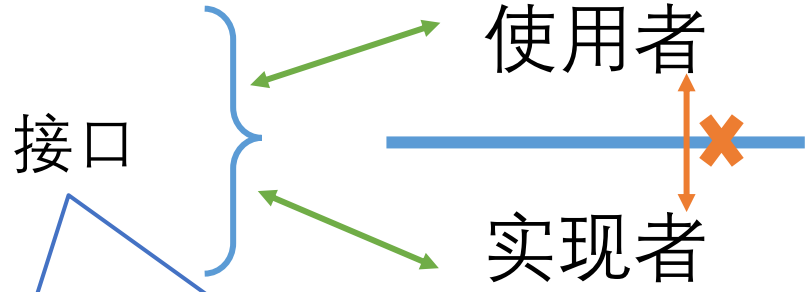
```
def sum(it):
  return reduce(0, add, it)

def prod(it):
  return reduce(1, mult, it)
```

函数抽象


sum，prod函数的编写者
调用reduce!
reduce抽象
使用者
无显性阻碍
抽象
接口
实现者
把reduce的实现复制过来!
reduce函数规约:接收3个参数,返回balabala
reduce函数的编写者
def reduce(base, f, it):
...

C语言视角

```
int main() {
    printf("%d\n", factorial(10));
}
```



接口

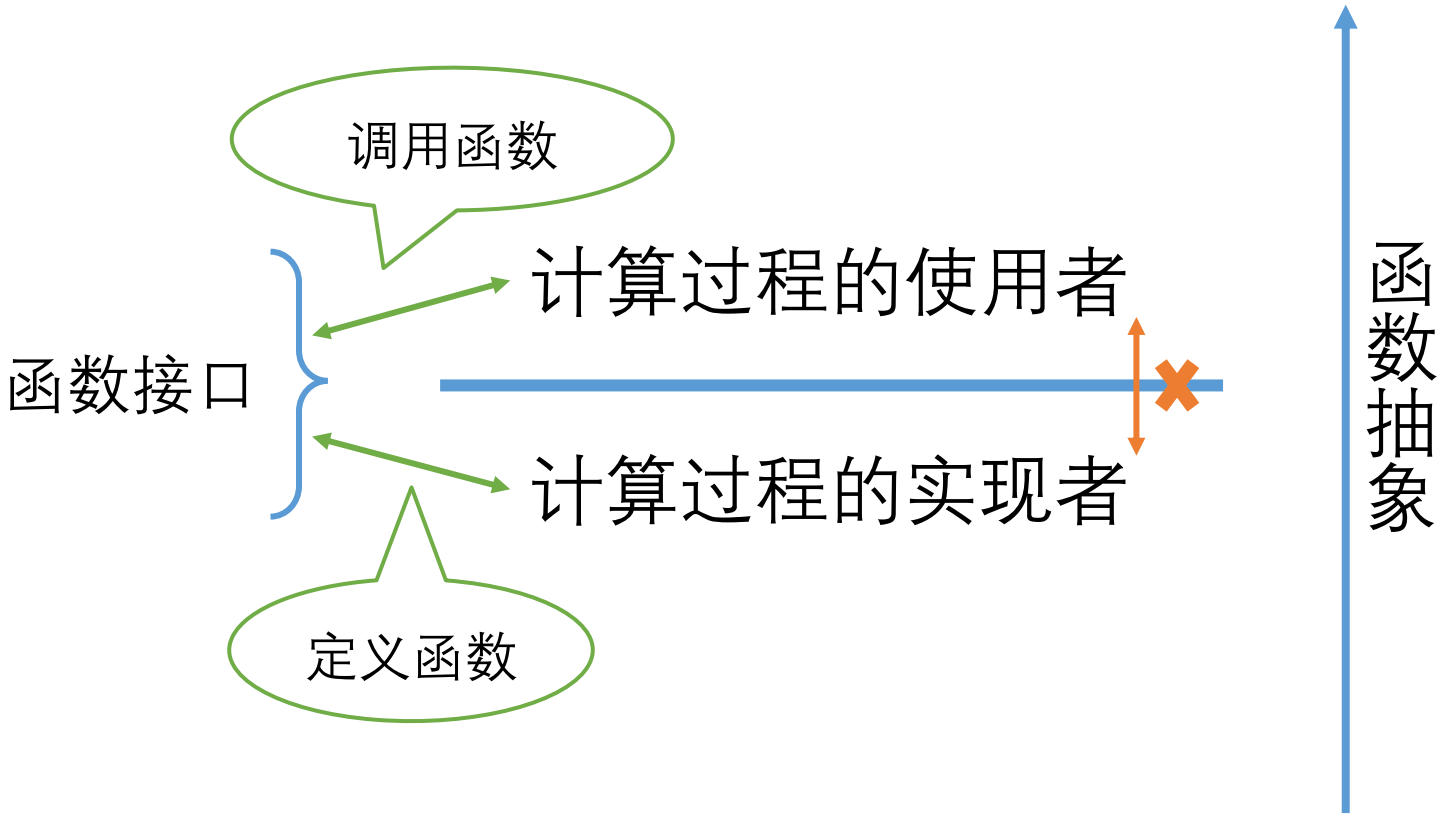
```
// 函数声明
int factorial(int n);
```

抽象

factorial
抽象

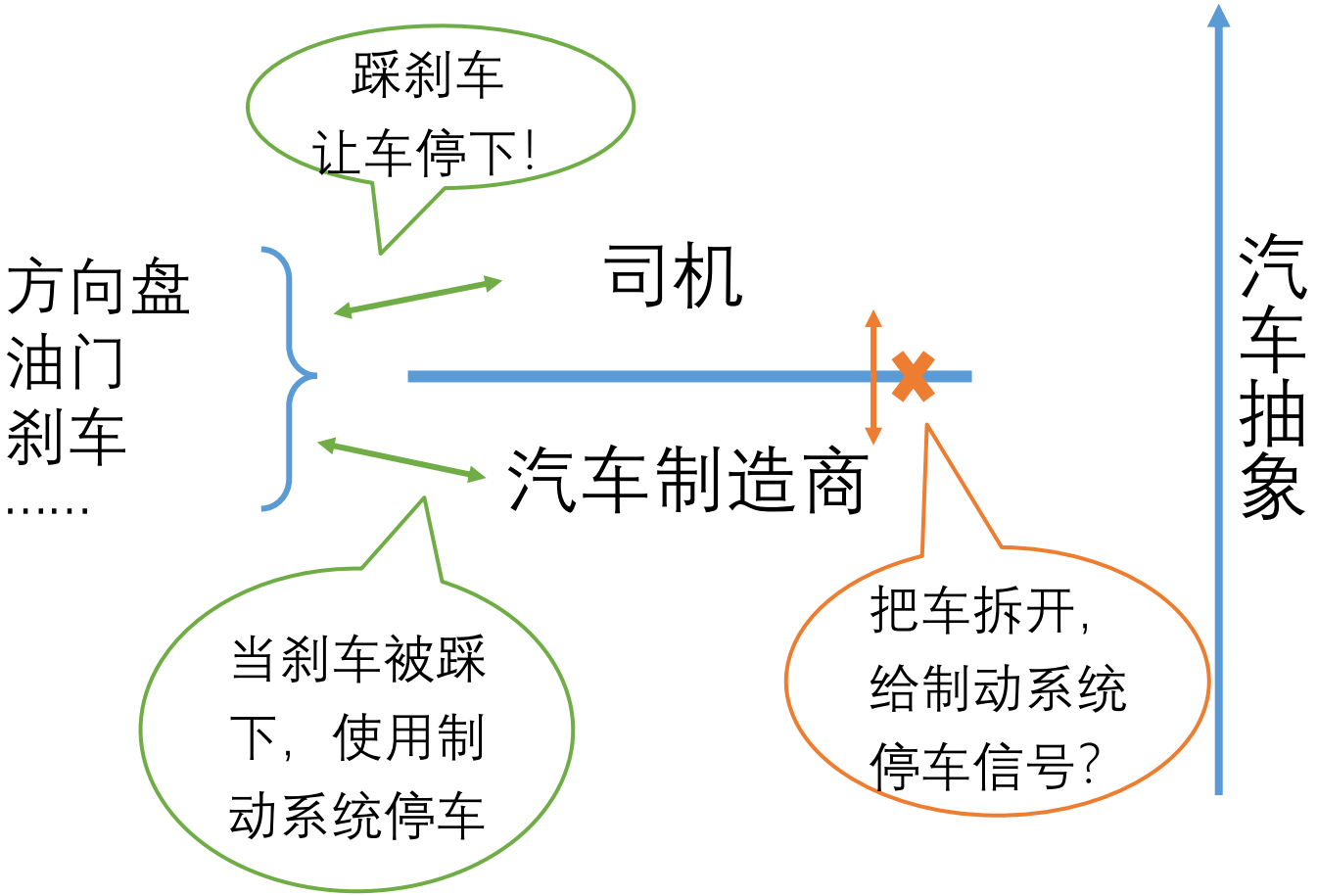
```
int factorial(int n) {
  if (n == 0 || n == 1)
    return 1;
  return n * factorial(n - 1);
}
```

函数抽象

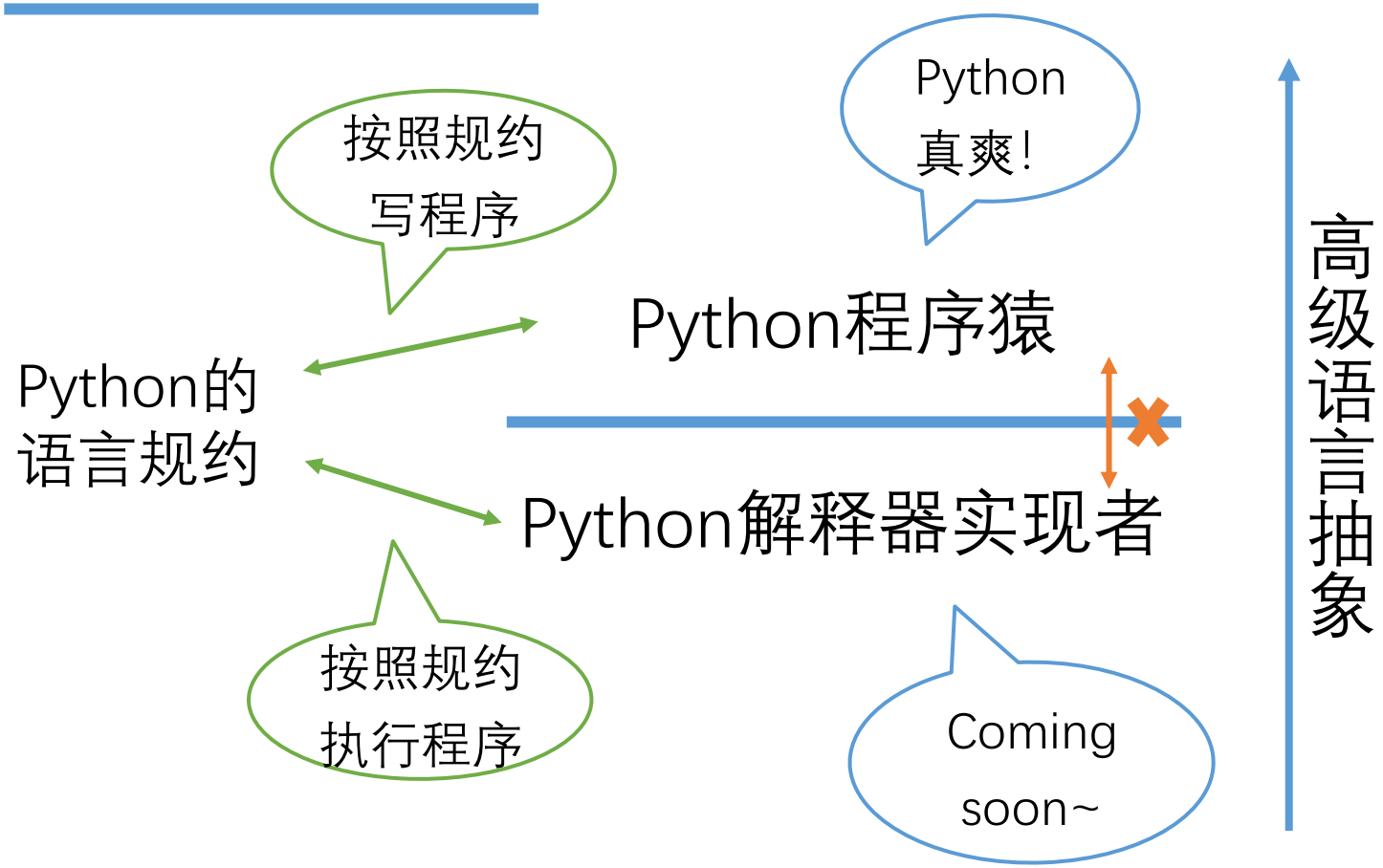


把计算过程提炼为函数的过程为函数抽象

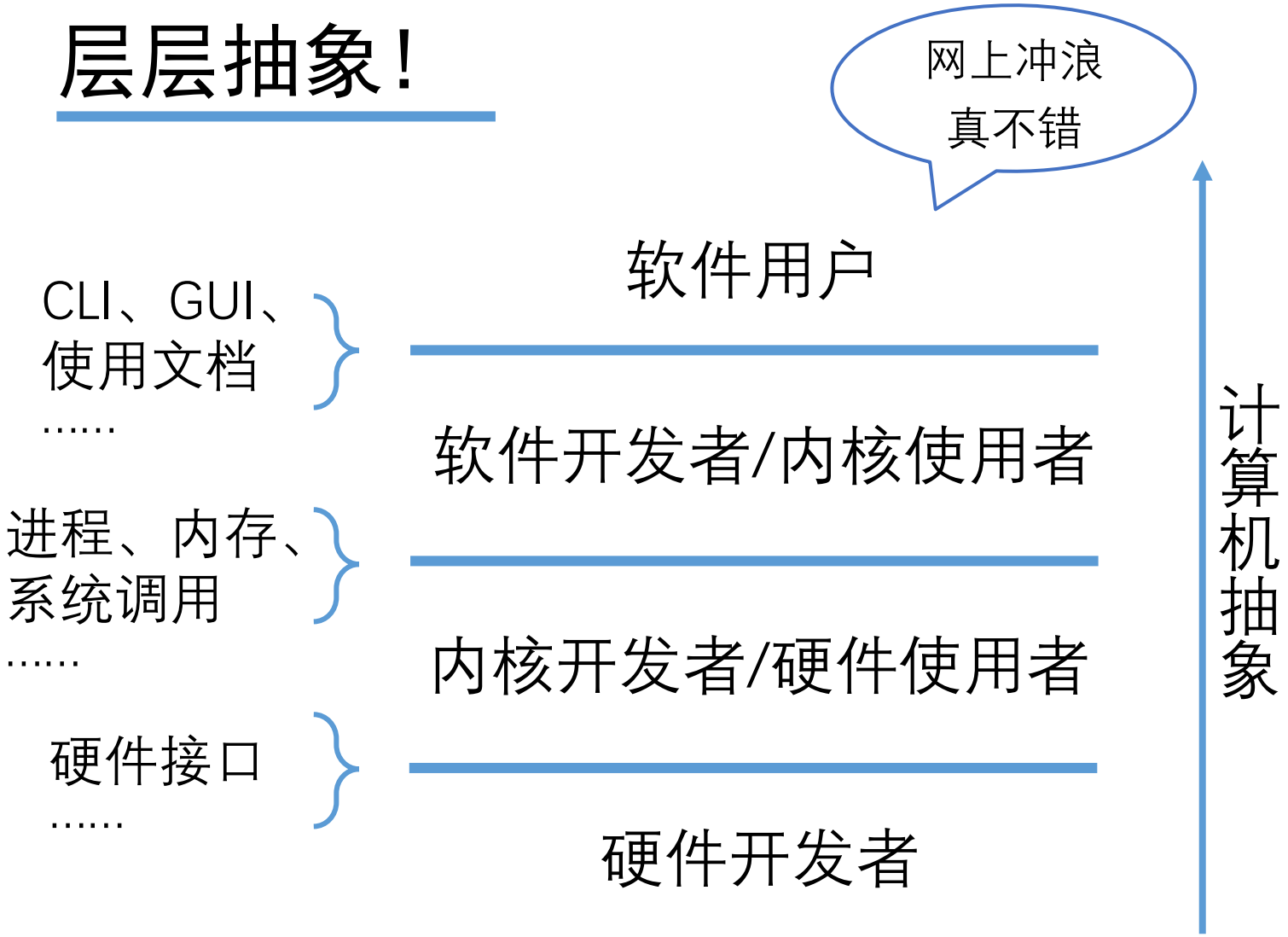
这是抽象


踩刹车
让车停下!
方向盘
油门
刹车
.....
司机
汽车制造商
当刹车被踩下，使用制动系统停车
把车拆开，给制动系统停车信号?
汽车抽象

这也是抽象

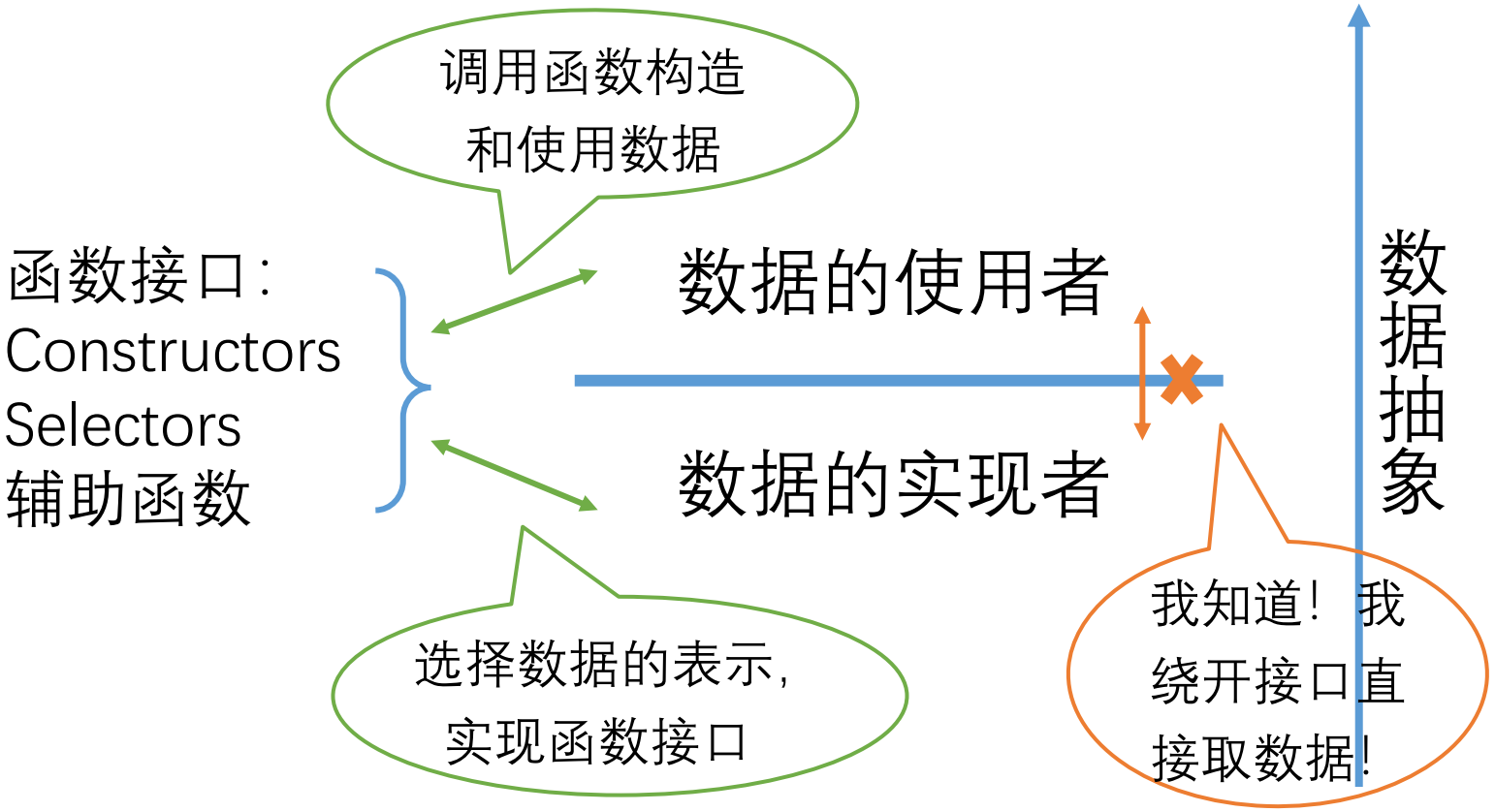

Python
真爽!
按照规约
写程序
Python程序猿
Python的
语言规约
Python解释器实现者
按照规约
执行程序
Coming
soon~
高级语言抽象

层层抽象！

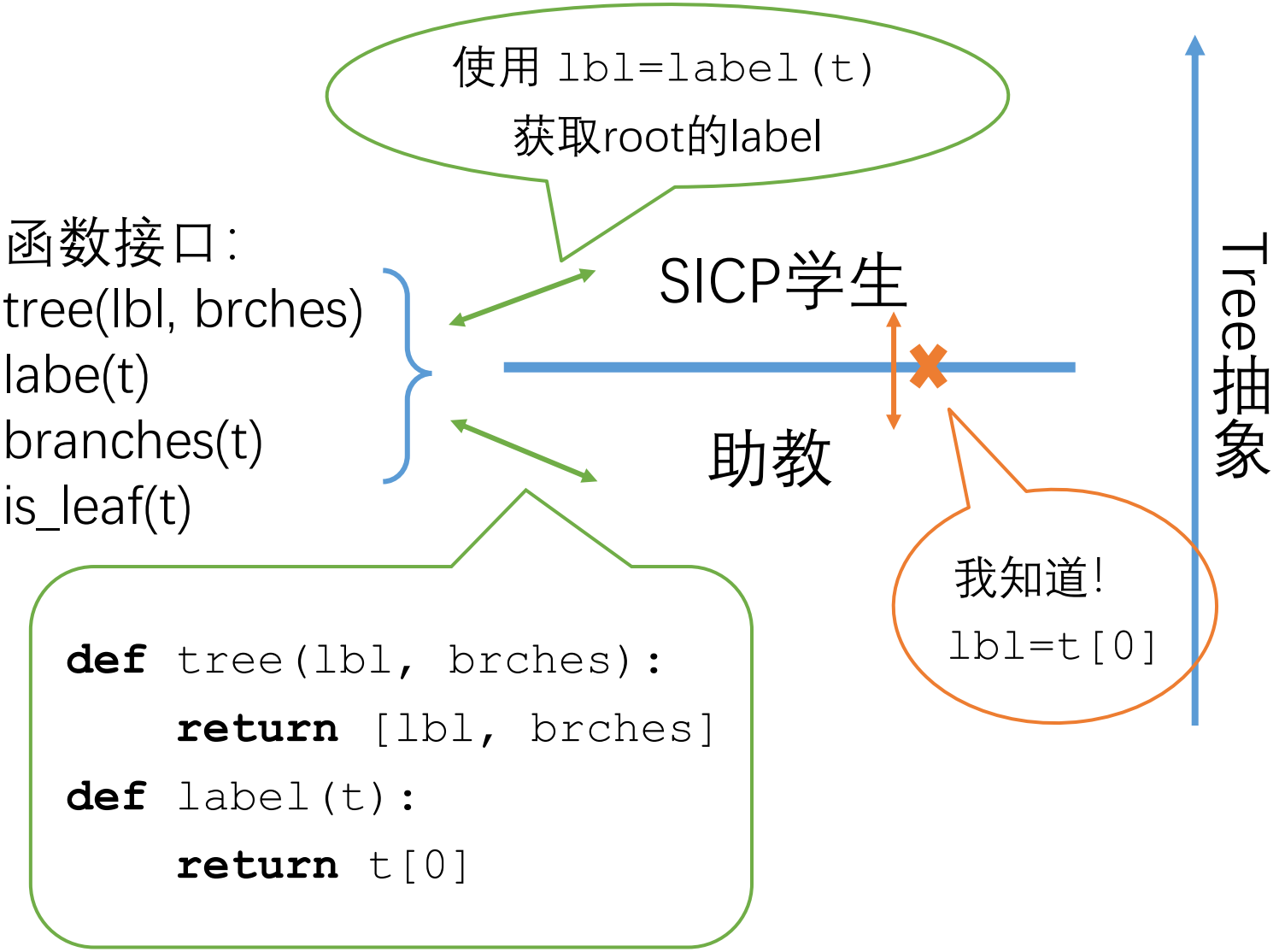


未来四年都会学到……

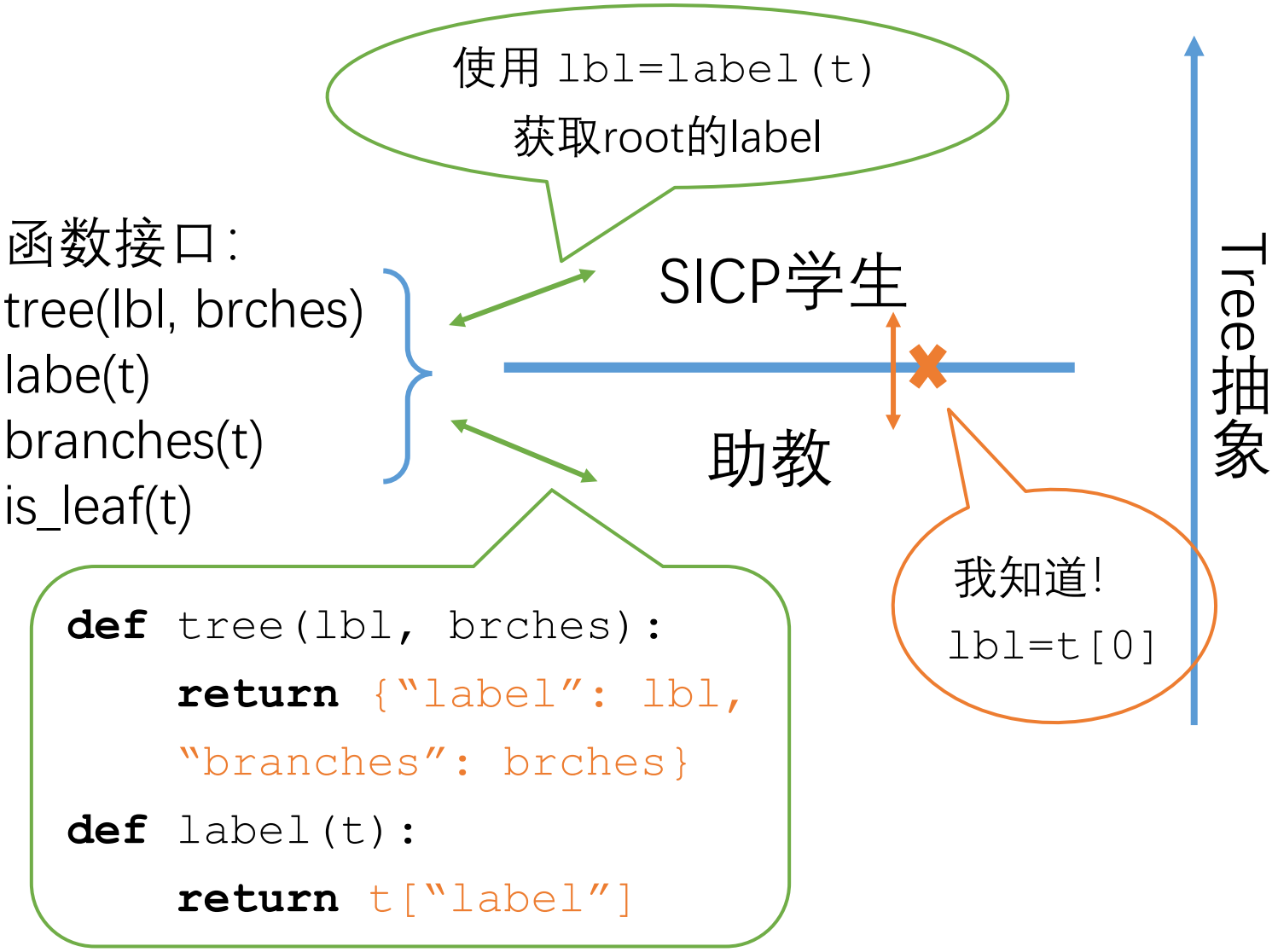
数据抽象


调用函数构造
和使用数据
函数接口:
Constructors
Selectors
辅助函数
数据的使用者
数据的实现者
数据抽象
选择数据的表示,
实现函数接口
我知道! 我
绕开接口直
接取数据!

数据抽象


使用 lbl=label(t)
获取root的label
函数接口:
tree(lbl, brches)
labe(t)
branches(t)
is_leaf(t)
SICP学生
助教
Tree抽象
我知道!
lbl=t[0]
def tree(lbl, brches):
return [lbl, brches]
def label(t):
return t[0]

数据抽象


使用 lbl=label(t)
获取root的label
函数接口:
tree(lbl, brches)
labe(t)
branches(t)
is_leaf(t)
SICP学生
助教
Tree抽象
我知道!
lbl=t[0]
def tree(lbl, brches):
return {"label": lbl,
"branches": brches}
def label(t):
return t["label"]

List拾遗

- Index中的负数
- Slice中的负数

Demo

没能消化的自己课后查资料自学。

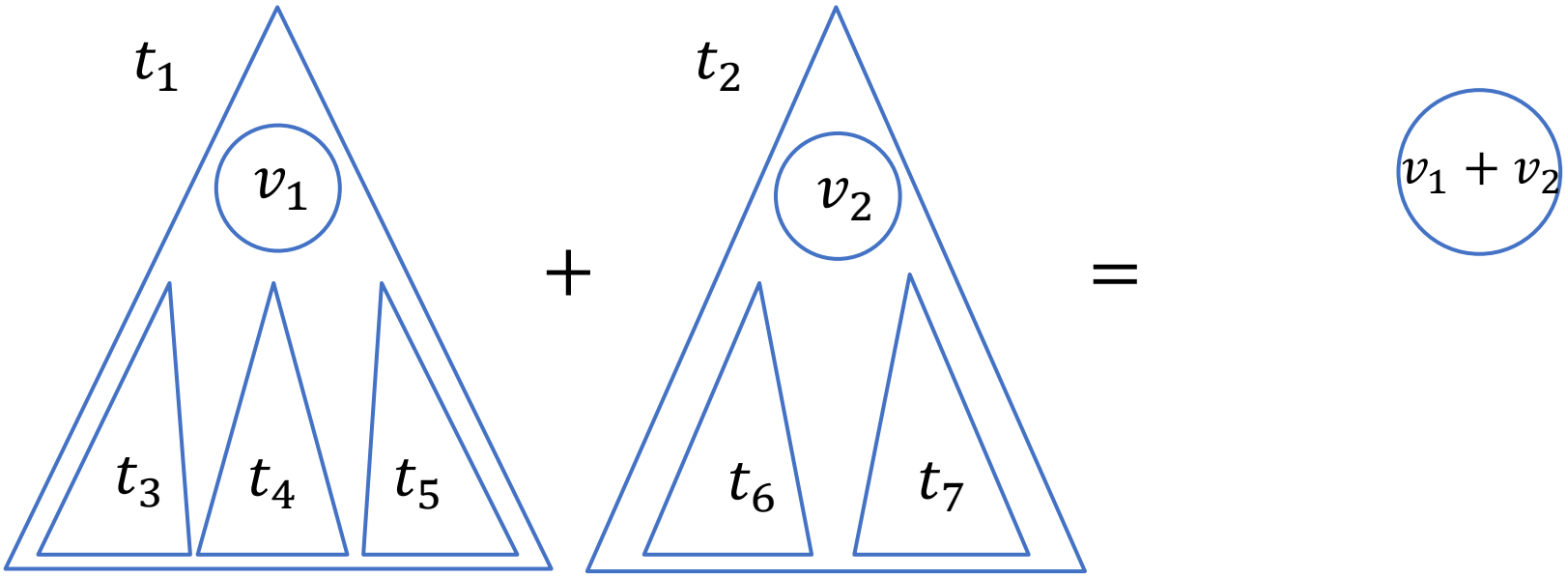
习题选讲

- Tree
  - hw04-03-3: Add Trees
  - lab04-02-01: Preorder
- Cats
  - cats-07, 10, 06

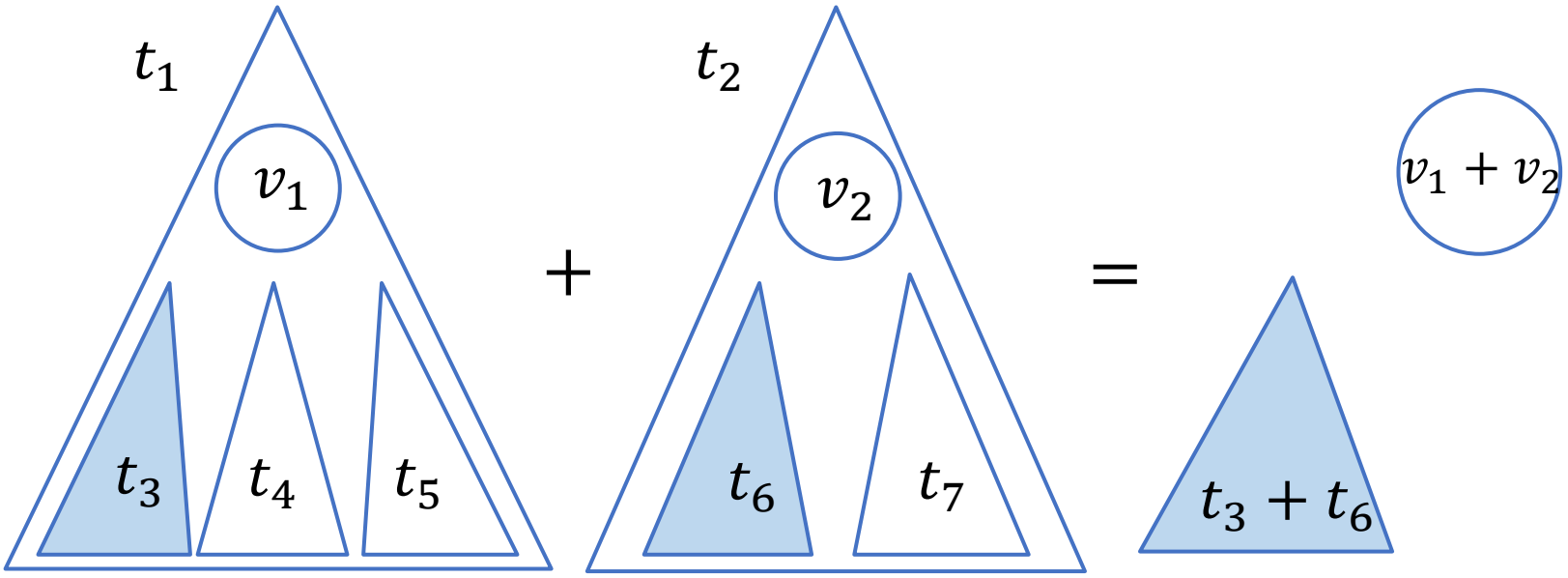
Tree的技巧

- Tree是递归定义的:
  - Base case: leaf（没有branches）是Tree
  - Inductive: branches都是Tree的也是Tree
- 做题:
  - Base case: is_leaf(t) == True
  - Otherwise: 对branches做递归

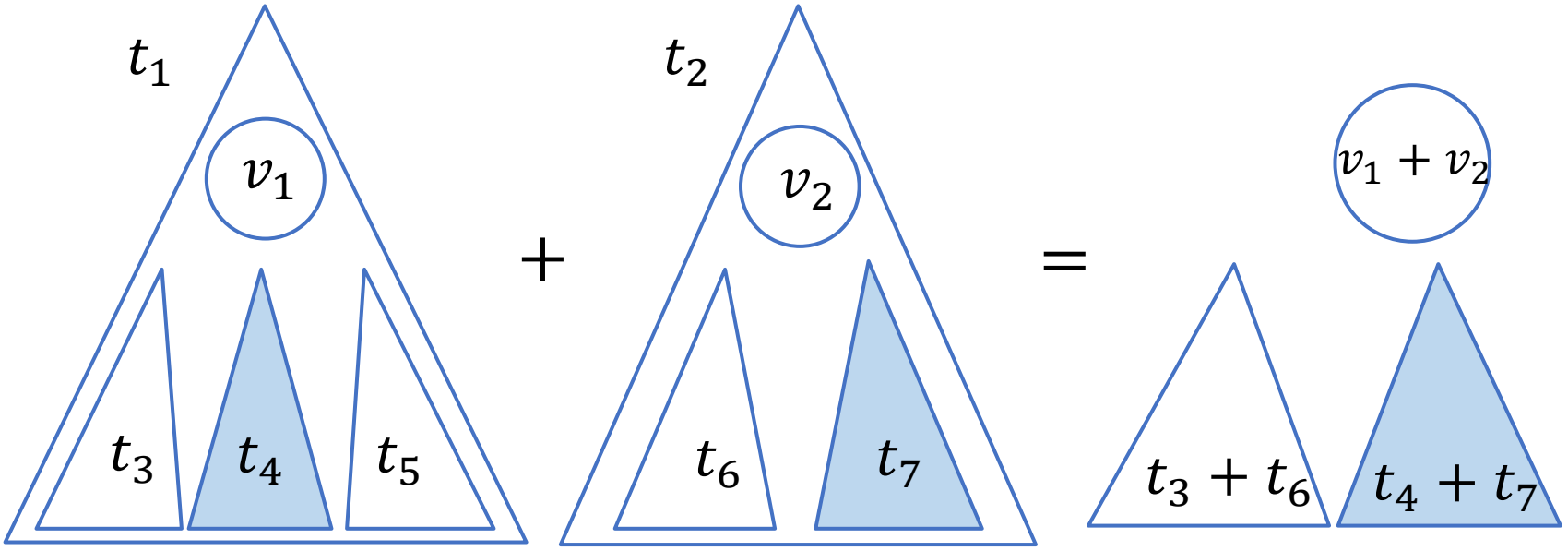
Add Tree



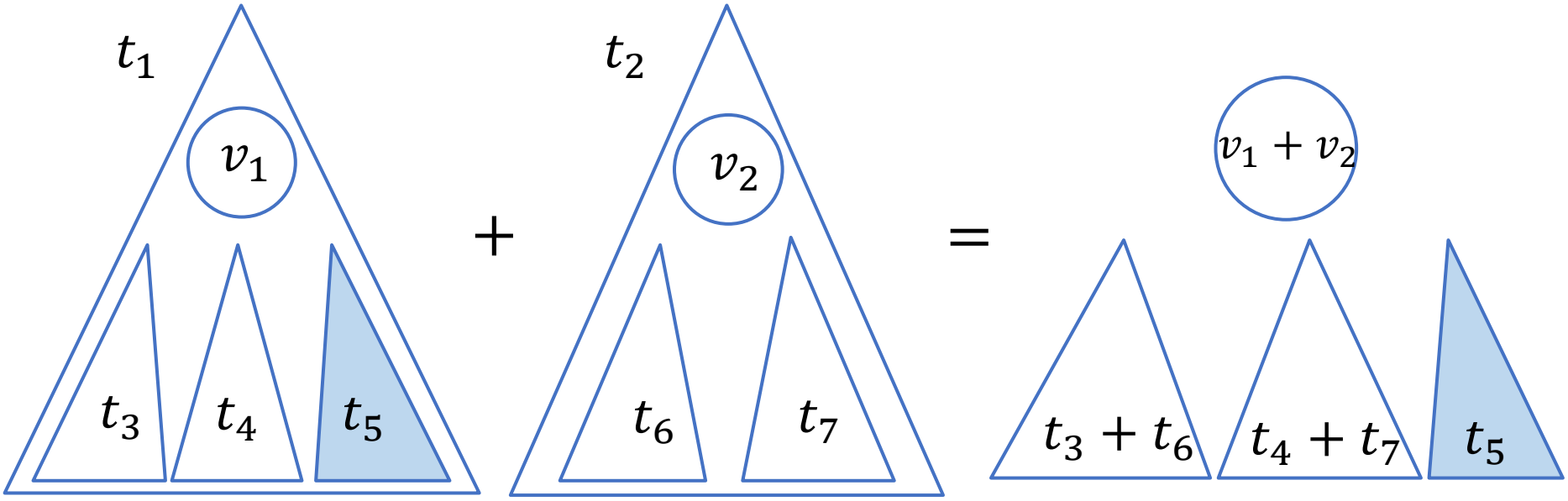
Add Tree



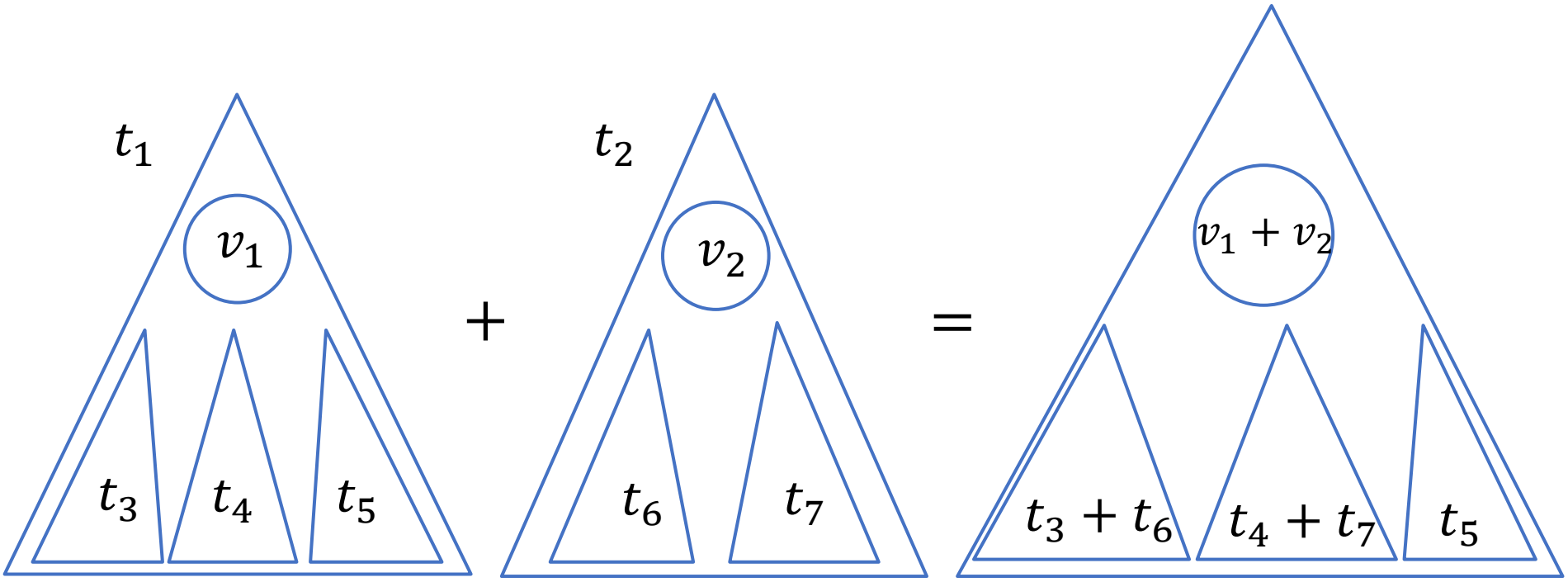
Add Tree



Add Tree



Add Tree



期中复习

- 抽象！抽象！抽象！
- List拾遗
- 习题选讲

Q&A