结构归纳法：
字符串构造与拼接证明

2026 年 4 月 8 日星期三

黄世鑫

1. 什么是结构归纳法？

结构归纳法是数学归纳法在递归定义的数据结构上的自然推广。

数学归纳法（针对自然数）：

- 基础情况：证明 P(0) 成立
- 归纳步骤：假设 P(n) 成立，推导 P(n+1) 成立
- 结论：对所有自然数 n，P(n) 成立

结构归纳法（针对递归数据结构）：

- 基础情况：证明基础元素满足性质 P
- 构造步骤：假设构造前的元素满足 P，推导构造后的元素仍满足 P
- 结论：递归数据类型中所有元素都满足 P

关键差异：归纳的对象不再是“数字的大小”，而是数据结构的构造方式。

2. 递归定义数据类型（以 Python 字符串为例）

字符串 str 的形式化递归定义：

- 基础情况：空字符串属于 str '' ∈ str
- 构造情况：若 u 是 Unicode 字符，且 s ∈ str，则 'u' + s ∈ str

这两条规则可以生成所有可能的字符串。

示例：“abc” 的构造过程：

text

"abc" = 'a' + "bc" "bc" = 'b' + "c" "c" = 'c' + "" "" ← 基础情况

3. 为证明做准备：递归定义相关操作

拼接操作 + 的递归定义：

- 基础：'' + s = s（空串是左恒等元）
- 构造：('u' + s) + t = 'u' + (s + t)

长度函数 len 的递归定义：

- 基础：len('') = 0
- 构造：len('u' + s) = len(s) + 1

Python 验证（与内置行为一致）：

s = "bc"

print(len("")) # 0

print$len("a" + s)$ # 3

print"" + s # 'bc'

print"a" + s # 'abc'

4. 直观类比：多米诺骨牌

- 条件 1（基础情况）：第一张骨牌会倒 ≡ 证明空字符串 '' 满足性质 P
- 条件 2（构造步骤）：任何一张倒下都会带倒下一张 ≡ 假设 P(r) 成立，

则 P('u' + r) 成立

- 结论：所有骨牌都会倒 ≡ 递归数据类型中所有元素满足 P

5. 结构归纳法证明示例

定理：对任意字符串 s, t ∈ str，有 len(s + t) = len(s) + len(t)
证明（使用结构归纳法）：
定义归纳谓词 P(s)： 对任意 t ∈ str，len(s + t) = len(s) + len(t) 成立。
基础情况（s = ''）：
text
len('' + t) = len(t) ← 拼接基础定义（左恒等）
= 0 + len(t)
= len('') + len(t) ✓
构造情况（s = 'u' + r，其中 r 为较短字符串）：
text
len(('u' + r) + t) = len('u' + (r + t)) ← 拼接构造定义
= 1 + len(r + t) ← len 构造定义
= 1 + len(r) + len(t) ← 归纳假设 P(r) 成立
= len('u' + r) + len(t)
= len(s) + len(t) ✓
由结构归纳原理，对所有字符串 s，P(s) 成立。因此定理得证。 □

构造(Construct)与拼接(concatenation)

构造：字符＋字符串

Character × String ⟶ String

时间复杂度：$O(1)$

在理论计算机科学默认的递归结构（如单向链表）中，把一个字符放到一个字符串前面，只需要在内存里申请一个新节点，把数据填入，然后把指针指向原来的字符串即可。它是瞬间完成的，只需一步

拼接：字符串+字符串

String× String ⟶ String

时间复杂度：O(N)

要完成 String A + String B，计算机必须启动一个循环或递归过程：

1. 它必须把字符串 A 拆解开，走到 A 的最尾端。

然后把 B 挂载到 A 的尾端。

2. 如果 A 的长度是 N，这个操作需要走 N 步。所以它的时间复杂度是与左边字符串长度成正比的线性时间 O(N)。

拼接操作(splicing operation)与结合律(associative law)

拼接操作的递归定义是系统的公理和计算语义。

结合律是基于上述定义推导出的定理和代数性质。

拼接定义的构造情况：('u'+s)+t = 'u'+(s+t)

变量类型：u 是单字符，s 和 t 是字符串

这个等式是微观层面的“拆解与重组”，当左边的操作数是由一个字符 u 和一个字符串 s 构成时，你应该如何把这个字符 u 提取出来，放到最外层。

结合律：(r+s)+t = r+(s+t)

变量类型：r、s、t 全部都是字符串

这是一个宏观层面的等式。它讨论的是三个黑盒字符串在任意拼接组合下，最终结果的等价性。

总结对照表

维度	**拼接操作的递归定义 (**`'u'+s+t = ...`**)**	**结合律定理 (**`(r+s)+t = ...`**)**
理论地位	公理 (Axiom) / 定义 (Definition)	定理 (Theorem) / 性质 (Property)
参与元素	**字符 (char)** 与 字符串 (string)	全部是 **字符串 (string)**
逻辑方向	人为规定，无需证明，用来执行	必须被证明，需要被验证
证明中的作用	证明其他定理的**工具 / 武器**	需要被证明的**目标 / 靶子**
本质	描述了函数的**计算语义**（怎么算）	描述了函数的**代数性质**（有什么规律）