蘇州大學
SOOCHOW UNIVERSITY

Chap5 Induction and Recursion

Part I: Mathematical Induction

Jin-Hui Wu

2026-04-02

大纲

- 数学归纳法 (5.1)
- 强归纳法
- 递归定义和结构归纳
- 递归算法

数学归纳法

- 数学归纳法 (mathematical induction)
  - 一种证明方法

$$(P(1) \wedge \forall k\,(P(k) \rightarrow P(k+1))) \rightarrow \forall n\,P(n)$$

  - 不断使用全称实例和肯定前件的过程

数学归纳法

- 数学归纳法 (mathematical induction)
  - 一种证明方法

    $$(P(1) \land \forall k\,(P(k) \to P(k+1))) \to \forall n\,P(n)$$

  - 不断使用全称实例和肯定前件的过程

  - $P(1)$：基础步骤 (basic step)
  - $\forall k\,(P(k) \to P(k+1))$：归纳步骤 (induction step)

例

- 用数学归纳法证明

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

对$n \in \mathbb{N}^+$成立

- 需要题目给出求和结果，或先猜出求和结果

例

- 用数学归纳法证明

$$2^n < n!$$

对$n \geq 4$成立

例

- 用数学归纳法证明

$$3|(n^3 - n)$$

对$n \in \mathbb{Z}$成立

例

- 用数学归纳法证明

$$57|(7^{n+2}+8^{2n+1})$$

对$n \in \mathbb{N}$成立

例

- 用数学归纳法证明可以用右三联骨牌去覆盖任何一个去掉1个格的$2^n \times 2^n$格的棋盘

数学归纳法

- 演绎推理
  - 使用推理规则从前提导出结论
- 归纳推理
  - 通过证据来支持结论，而不是确定结论
- 数学归纳法是演绎推理，不是归纳推理

大纲

- 数学归纳法
- 强归纳法 (5.2)
- 递归定义和结构归纳
- 递归算法

强归纳法

- 强归纳法 (strong induction)
  - $(P(1) \land \forall k\,(P(1) \land \ldots \land P(k) \rightarrow P(k+1))) \rightarrow \forall n\, P(n)$
  - 也是不断使用全称实例和肯定前件的过程

强归纳法

- 强归纳法 (strong induction)
  - $(P(1) \land \forall k\,(P(1) \land \ldots \land P(k) \rightarrow P(k+1))) \rightarrow \forall n\, P(n)$
  - 也称为完整归纳法 (complete induction)、第二数学归纳法 (second principle of MI)
  - $P(1)$：基础步骤
  - $\forall k\,(P(1) \land \ldots \land P(k) \rightarrow P(k+1))$：归纳步骤

强归纳法

- 强归纳法 (strong induction)
  - $(P(1) \land \forall k\,(P(1) \land \ldots \land P(k) \rightarrow P(k+1))) \rightarrow \forall n\, P(n)$

- 和数学归纳法的关系
  - 能用 MI 证的都可以用强归纳证
  - 能用 MI 证明时，MI 的证明更简洁
  - 优先考虑 MI

例

- 用强归纳法证明算术基本定理
  - 若n是大于1的整数，则n可以写成素数之积

例

- 用强归纳法证明用4分和5分邮票可以组成大于或等于12分的每种邮资

大纲

- 数学归纳法
- 强归纳法
- 递归定义和结构归纳 (5.3)
- 递归算法

递归定义

- 递归定义的函数 (recursively defined function)
  - 基础步骤：规定这个函数在0处的值
  - 递归步骤：给出从较小的整数处的值来求出当前的值的规则

递归定义

- 递归定义的函数 (recursively defined function)
  - 基础步骤：规定这个函数在0处的值
  - 递归步骤：给出从较小的整数处的值来求出当前的值的规则

- 函数$f(n)$和序列a_n等价
- 基础步骤 = 初始条件
- 递归步骤 = 递推关系

例

- 给出阶乘函数$f(n) = n!$的递归定义

例

- 斐波那契数列可写为递归定义的函数

$$f(0) = 0, f(1) = 1$$
$$f(n+2) = f(n+1) + f(n)$$

- 证明当$n \geq 3$时，$f(n) \geq \alpha^{n-2}$，其中$\alpha = (1+\sqrt{5})/2$

例

- 斐波那契数列可写为递归定义的函数

$$f(0) = 0, f(1) = 1$$
$$f(n+2) = f(n+1) + f(n)$$

- 证明当$n \geq 3$时，$f(n) \geq \alpha^{n-2}$，其中$\alpha = (1+\sqrt{5})/2$

- 证明Lamé定理：用欧几里得算法计算$\gcd(a, b)$所做的除法次数不超过b的位数的5倍
  - 计算复杂度的首个结论

Gabriel Lamé
(1795-1870)

递归定义

- 递归定义的集合和结构 (recursively defined sets and structures)
  - 基础步骤：定义初始元素
  - 递归步骤：基于已有元素定义新元素

例

- 用递归的方式定义：
  - 自然数集

例

- 用递归的方式定义：
  - 自然数集
  - 字母表Σ上的字符串集Σ^*

例

- 用递归的方式定义：
  - 自然数集
  - 字母表Σ上的字符串集Σ^*
  - 字符串的长度函数$l(w)$

例

- 用递归的方式定义：
  - 自然数集
  - 字母表Σ上的字符串集Σ^*
  - 字符串的长度函数$l(w)$
  - 由5种常见逻辑联结词组成的合法复合命题公式

例

- 用递归的方式定义：
  - 自然数集
  - 字母表Σ上的字符串集Σ^*
  - 字符串的长度函数$l(w)$
  - 由5种常见逻辑联结词组成的合法复合命题公式
  - 平衡括号集

递归定义

- 满二叉树 (full binary tree)
  - 基础步骤：单个顶点是满二叉树
  - 递归步骤：若T_1和T_2是满二叉树，则存在一个表示为$T_1 \cdot T_2$的满二叉树，其树根的左子树为T_1、右子树为T_2

递归定义

- 满二叉树 (full binary tree)
  - 基础步骤：单个顶点是满二叉树
  - 递归步骤：若T_1和T_2是满二叉树，则存在一个表示为$T_1 \cdot T_2$的满二叉树，其树根的左子树为T_1、右子树为T_2

- 高度 (height)
  - 基础步骤：仅含一个树根的树满足$h(T) = 0$
  - 递归步骤：$h(T_1 \cdot T_2) = \max\big(h(T_1), h(T_2)\big) + 1$

递归定义

- 满二叉树 (full binary tree)
  - 基础步骤：单个顶点是满二叉树
  - 递归步骤：若T_1和T_2是满二叉树，则存在一个表示为$T_1 \cdot T_2$的满二叉树，其树根的左子树为T_1、右子树为T_2
- 高度 (height)
  - 基础步骤：仅含一个树根的树满足$h(T) = 0$
  - 递归步骤：$h(T_1 \cdot T_2) = \max\big(h(T_1), h(T_2)\big) + 1$
- 顶点数 (number of vertices)
  - 基础步骤：仅含一个树根的树满足$n(T) = 1$
  - 递归步骤：$n(T_1 \cdot T_2) = n(T_1) + n(T_2) + 1$

例：结构归纳法

- T是满二叉树，证明$n(T) \leq 2^{h(T)+1} - 1$

例：广义归纳法

- 令$a_{0,0}=0$，定义

$$a_{m,n}=\begin{cases} a_{m-1,n}+1 & \text{if } n=0 \text{ and } m>0 \\ a_{m,n-1}+n & \text{if } n>0 \end{cases}.$$

证明$a_{m,n}=m+\frac{n(n+1)}{2}$对$(m,n)\in\mathbb{N}^2$成立

总结

- 数学归纳法

$$(P(1) \wedge \forall k\,(P(k) \rightarrow P(k+1))) \rightarrow \forall n\,P(n)$$

- 强归纳法

$$(P(1) \wedge \forall k\,(P(1) \wedge \ldots \wedge P(k) \rightarrow P(k+1))) \rightarrow \forall n\,P(n)$$

- 递归定义
  - 函数（阶乘）、集合（自然数集）、结构（满二叉树）
- 结构归纳法、广义归纳法