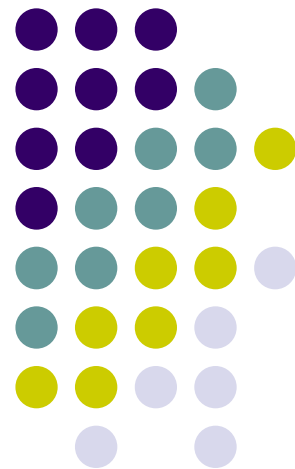


自然数与 数学归纳法

马晓星

南京大学·计算机科学与技术系





提要

- 自然数
 - 自然数的公理化定义
 - 自然数的集合论构造
 - 自然数基本运算定义
- 数论初步
 - 整数除法与同余算术
 - 素数与算术基本定理
 - 同余方程与中国剩余定理
 - 费马小定理与欧拉定理



自然数

小朋友的问题



$$1 + 2 = 3 \quad ?$$



自然数的直觉理解

- 自然数(Natural Numbers)的用途
 - 记序 - 序数 (Ordinal Numbers)
 - 记量 - 基数 (Cardinal Numbers)
 - 记名 - Nominal Numbers

皮亚诺公理

(Peano axioms for natural numbers)



1. 零(记作0)是个自然数(自然数集记作 \mathbf{N} , 或 \mathbb{N}).
2. 每个自然数(x)都有一个自然数后继($S(x)$).
3. 零不是任何自然数的后继.
4. 不同的自然数有不同的后继.
5. 设由自然数组成的某个集合含有零, 且每当该集合含有某个自然数时便也同时含有这个数的后继, 那么该集合定含有全部自然数.

【归纳公理】

注: 此处略去了关于自然数“相等”的四条公理。

皮亚诺公理 (Peano axioms for natural numbers)



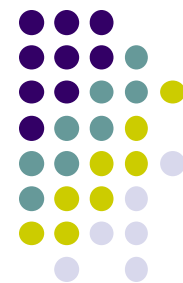
1. $0 \in \mathbf{N}$
2. $x \in \mathbf{N} \rightarrow S(x) \in \mathbf{N}$
3. $x \in \mathbf{N} \rightarrow S(x) \neq 0$
4. $x \in \mathbf{N} \wedge y \in \mathbf{N} \wedge S(x) = S(y) \rightarrow x = y$
5. $0 \in M \wedge \forall x(x \in M \rightarrow S(x) \in M) \rightarrow \mathbf{N} \subseteq M$
for any property M (axiom of induction).



自然数的集合论构造

- 自然数：从“理论”到“模型”
- 目标：
 - 从集合和集合运算出发，构造一个结构 $\langle 0, \mathbf{N}, S \rangle$ 满足皮亚诺公理。

冯·诺伊曼(von Neumann)的自然数（序数）构造



- 记号 0 表示: \emptyset
- 函数 S 定义: $S(x) = x \cup \{x\}$
- 自然数集合 \mathbf{N} :

$$0 = \emptyset$$

$$1 = S(0) = S(\emptyset) = \emptyset \cup \{\emptyset\} = \{\emptyset\} = \{0\}$$

$$2 = S(1) = S(\{0\}) = \{0\} \cup \{\{0\}\} = \{0, \{0\}\} = \{0, 1\}$$

$$3 = S(2) = S(\{0, 1\}) = \{0, 1\} \cup \{\{0, 1\}\} = \{0, 1, \{0, 1\}\} = \{0, 1, 2\}$$

...

每个（序）数都定义为所有在其前面的数构成的集合

麻烦在这里！如何保证没有引入其它东西？

冯·诺伊曼的 自然数构造



- 设 a 为集合, 称 $a \cup \{a\}$ 为 a 的**后继**, 记为 $S(a)$, 或 a^+ 。
- 设 A 是集合, 若 A 满足下列条件, 称 A 为**归纳集**:
 - $\emptyset \in A$
 - $\forall a(a \in A \rightarrow S(a) \in A)$
- 定义自然数集合 \mathbf{N} 为 **所有归纳集的交集**。
 - 因此: $\mathbf{N} = \{ \emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}, \dots \}$

冯·诺伊曼的 自然数构造

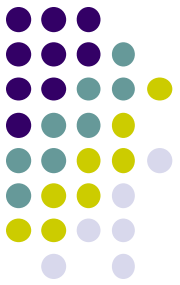


- 判断下列命题对不对

- $1 \in 3$ $2 \in 3$

- $1 \subseteq 3$ $2 \subseteq 3$

- $1 \cup 3 = 3$ $2 \cap 3 = 2$



自然数的算术运算

- 如何定义加法？

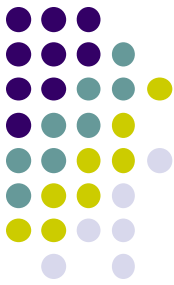
- 对于任意的自然数 m, n

$$m + 0 = m ; \quad \text{【A1】}$$

$$m + S(n) = S(m + n) . \quad \text{【A2】}$$

- 于是，回答“小朋友的问题”：

$$\begin{aligned} 1 + 2 &= 1 + S(1) = S(1+1) = S(1+S(0)) \\ &= S(S(1+0)) = S(S(1)) = S(2) = 3 \end{aligned}$$



自然数的算术运算

- 试证明： $0 + m = m$

证明： 对 m 作归纳.

基础步骤： $0 + 0 = 0$.

归纳步骤： 假设 $0 + k = k$, 则

$$0 + S(k) = S(0 + k) = S(k).$$

证毕。

- 【练习】 试证明： 自然数的加法满足结合律。
- 【练习】 试证明： 自然数的加法满足交换律。



自然数的算术运算

- 同样地，我们可以定义乘法：

- 对于任意的自然数 m, n

$$m \times 0 = 0 ; \quad \text{【M1】}$$

$$m \times S(n) = m + (m \times n) . \quad \text{【M2】}$$

- 【练习】

试证明：自然数的乘法对加法满足分配律。



Let $x, y \in \mathbb{N}$ be arbitrary.

For all $z \in \mathbb{N}$, let $P(z)$ be the **proposition**:

$$\forall x, y \in \mathbb{N} : (x + y) \times z = (x \times z) + (y \times z)$$

Basis for the Induction

$P(0)$ is the case:

$$(x + y) \times 0 = 0$$

Definition of **Natural Number Multiplication**

$$= 0 + 0$$

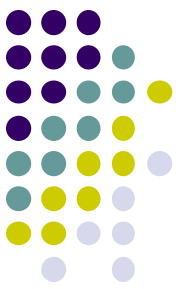
Definition of **Natural Number Addition**

$$= x \times 0 + y \times 0$$

Definition of **Natural Number Multiplication**

and so $P(0)$ holds.

This is our **basis for the induction**.



Induction Hypothesis

Now we need to show that, if $P(k)$ is true, where $k \geq 0$, then it logically follows that $P(k^+)$ is true.

So this is our **induction hypothesis**:

$$\forall x, y \in \mathbb{N} : (x + y) \times k = (x \times k) + (y \times k)$$

Then we need to show:

$$\forall x, y \in \mathbb{N} : (x + y) \times k^+ = (x \times k^+) + (y \times k^+)$$



Induction Step

This is our **induction step**:

$$\begin{aligned}(x + y) \times k^+ &= (x + y) \times k + (x + y) \\&= (x \times k) + (y \times k) + (x + y) \\&= ((x \times k) + x) + ((y \times k) + y) \\&= (x \times k^+) + (y \times k^+)\end{aligned}$$

Definition of **Natural Number Multiplication**

Induction Hypothesis

Natural Number Addition is Commutative and Associative

Definition of **Natural Number Multiplication**

So $P(k) \implies P(k^+)$ and the result follows by the **Principle of Mathematical Induction**:

$$\forall x, y, z \in \mathbb{N} : (x + y) \times n = (x \times z) + (y \times z)$$

—



小结

- 自然数的公理化和形式化
 - 皮亚诺公理及其形式逻辑表达
 - 冯·诺伊曼的归纳集构造
 - 基本算术运算的递归定义
基本运算性质的归纳证明

太初有道。

道生一，一生二，

二生三，三生万物。



数学归纳法与良序原理



提要

- 数学归纳法与良序原理
- 递归定义与结构归纳法
- 证明程序正确性与复杂度

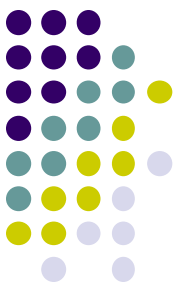


数学归纳法

- 回顾：皮亚诺公理

1. 零是个自然数.
2. 每个自然数都有一个自然数后继.
3. 零不是任何自然数的后继.
4. 不同的自然数有不同的后继.
5. 设由自然数组成的某个集合含有零，且每当该集合含有某个自然数时便也同时含有这个数的后继，那么该集合定含有全部自然数. 【归纳公理】

$\{x \in \mathbb{N} | P(x)\}$ 含有全部自然数 等同于 $P(n)$ 对所有自然数 n 成立.



数学归纳证明要点

- 证明目标

$P(n)$ 对所有的自然数 n 成立.

// 需明确定义 $P(n)$.

- 证明框架

说明将通过归纳来证明.

// 例如：我们对 n 做归纳.

- 基础步骤:

写出 $P(0)$ 并证明之.

// 通常比较简单.

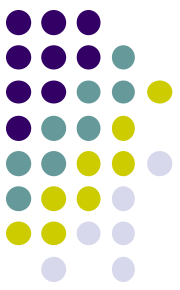
- 归纳步骤:

写出归纳假设 $P(k)$, 和待证明的 $P(k + 1)$, 而后证明之.

- 说明归纳证明完毕.

// 必要时重复原命题：因此， $P(n)$ 对所有的自然数 n 成立.

// 或者简单说证毕. 以及 ■



例：证明自然数加法满足结合律

加法结合律：对于任意自然数 a, b, c , 有 $(a + b) + c = a + (b + c)$

证明： 我们证明 $P(c)$ 对任意自然数 c 成立, 其中 $P(c)$ 为
对于任意自然数 a 和 b , 有 $(a + b) + c = a + (b + c)$
现对 c 做归纳。

基础步骤： $(a + b) + 0 = a + b = a + (b + 0)$.

归纳步骤： 假设 $P(k)$ 成立, 即 $(a + b) + k = a + (b + k)$, 现证明
 $(a + b) + (k + 1) = a + (b + (k + 1))$.

$$\begin{aligned}(a + b) + (k + 1) &= (a + b) + S(k) = S((a + b) + k) \\ &= S(a + (b + k)) = a + S(b + k) \\ &= a + (b + S(k)) = a + (b + (k + 1))\end{aligned}$$

于是归纳步骤完成。

归纳证明完毕。 ■

加法定义：对于任意的自然数 m, n

$$m + 0 = m ; \quad \text{【A1】}$$

$$m + S(n) = S(m + n) . \quad \text{【A2】}$$

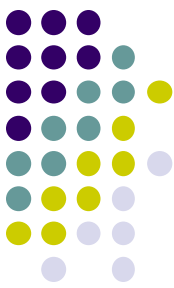


例：奇数个人的馅饼之战



- 平地上有奇数个人，人之间的距离各不相同。随着一声令下，每个人都朝距其最近的那个人扔馅饼。
- 试证明，至少有一个人没挨着馅饼。

如何定义 $P(n)$ ，并用数学归纳法证明之？



运用数学归纳法时犯的错误

- 平面上任何 $n \geq 2$ 条互不平行的直线必交于一点。

证明: 我们对 n 做归纳.

- 基础步骤: 两条不平行的直线必交于一点.
- 归纳步骤: 假设任何 k 条互不平行的直线交于一点.

对于任意 $k + 1$ 条互不平行的直线, 其中:

前 k 条必交于一点, 记为 p_1 ;

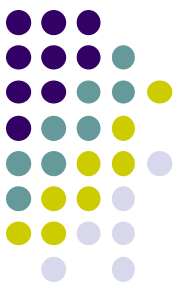
后 k 条必交于一点, 记为 p_2 ;

考虑到同时属于前 k 条与后 k 条的直线, 必有

$$p_1 = p_2$$

于是这 $k + 1$ 条互不平行的直线交于一点。

原命题归纳证明完毕. ■



强数学归纳法

- 证明目标
 - $P(n)$ 对所有的自然数 n 成立.
- 证明框架

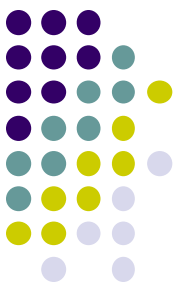
说明将通过归纳来证明.

 - 基础步骤:

写出 $P(0)$ 并证明之.
 - 归纳步骤:

假设 $P(0), P(1), \dots, P(k)$ 均成立, 证明 $P(k + 1)$ 成立。
 - 说明归纳证明完毕.

// 因此, $P(n)$ 对所有的自然数 n 成立. ■



强数学归纳法（一般形式）

- 设 $P(n)$ 是与整数 n 有关的陈述， a 和 b 是两个给定的整数，且 $a \leq b$.
- 如果能够证明下列陈述
 - $P(a), P(a+1), \dots, P(b)$.
 - 对任意 $k \geq b, P(a) \wedge \dots \wedge P(k) \rightarrow P(k+1)$
- 则下列陈述成立
 - 对任意 $n \geq a, P(n)$.



强数学归纳法（举例）

- 任意整数 $n(n \geq 2)$ 可分解为（若干个）素数的乘积
 - $n = 2$.
 - 考察 $n+1$.
- 用4分和5分就可以组成12分及以上的每种邮资.
 - $P(12), P(13), P(14), P(15)$.
 - 对任意 $k \geq 15$, $P(12) \wedge \dots \wedge P(k) \rightarrow P(k+1)$



良序原理

- **良序原理**: 自然数 \mathbf{N} 的任何非空子集 S 均有最小元素.
所谓“ S 有最小元素”即 $\exists a \in S (\forall b \in S (a \neq b \rightarrow a < b))$
- **良序原理与数学归纳法 (归纳公理)的关系**【严格说有差异】
 - \Rightarrow 【概要】 【注: 需在皮亚诺公理1-4基础上额外假设每个非0自然数都有一个直接前驱】
假设 $\forall n \in \mathbf{N} P(n)$ 不成立, 则 $\exists n (\neg P(n))$ 成立. 令 $S = \{ n \in \mathbf{N} \mid \neg P(n) \}$, S 非空.
根据良序原理, S 有最小元素, 记为 m , 由奠基步骤, $m \neq 0$
由 m 的最小性, $(m-1) \notin S$, 即 $P(m-1)$ 成立.
根据归纳步骤, $P(m)$ 成立, 即 $m \notin S$, 矛盾.
因此, $\forall n \in \mathbf{N} P(n)$ 成立.
 - \Leftarrow 【概要】 令 A 为 \mathbf{N} 的无最小元的非空子集, $B = \mathbf{N} - A$.
基础步骤: $0 \in B$. 这是因为 $0 \notin A$, 否则 0 即其最小元.
归纳步骤: 若 $0, \dots, n \in B$, 则 $(n+1) \in B$. 否则 $n+1$ 是 A 的最小元.
根据归纳原理 $B = \mathbf{N}$. 这与 A 非空矛盾.



良序原理在证明中的应用（举例）

- 其实我们经常不经意地用到良序原理, 例如
 - 我们在证明 $\sqrt{2}$ 不是有理数时, 说:
假设 $\sqrt{2}$ 是有理数, 那么总可以找到互素的正整数 p, q
使得 $\sqrt{2} = p/q$, …… (为什么?)
其实这里我们是在说:
令 p 为满足 $\sqrt{2} = p/q$ 的最小正整数, 则 p, q 一定互素,
否则假设 $\gcd(p, q) = c > 1$, 则 $\sqrt{2} = (p/c)/(q/c)$,
但是 $p/c < p$, 矛盾!



良序原理在证明中的应用（举例）

- 设 a 是整数, d 是正整数, 则存在唯一的整数 q 和 r 满足 $a = dq + r$ 其中 $0 \leq r < d$.
- 证明概要
 - 令 $S = \{a - dq \mid 0 \leq a - dq, q \in \mathbb{Z}\}$, S 非空.
 - 非负整数集合具有良序性
 - S 有最小元, 记为 $r_0 = a - dq_0$.
 - 可证 $0 \leq r_0 < d$
 - 唯一性证明, $0 \leq r_1 - r_0 = d(q_0 - q_1) < d$, 因此, $q_1 = q_0$

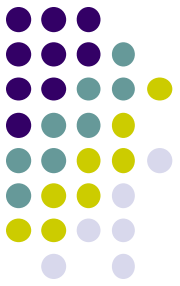


良序原理在证明中的应用（举例）

- 在循环赛胜果图中，若存在长度为 m （ $m \geq 3$ ）的回路，则必定存在长度为3的回路。

备注： $a_i \rightarrow a_j$ 表示 a_i 赢了 a_j

- 证明概要
 - 设最短回路的长度为 k // 良序原理的保证
 - 假设 $k > 3$
 - $a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow \dots \rightarrow a_k \rightarrow a_1$
 - 若 $a_3 \rightarrow a_1$, 存在长度为3的回路，矛盾。
 - 若 $a_1 \rightarrow a_3$, 存在长度为 $(k-1)$ 的回路，矛盾。



递归定义与结构归纳法



递归定义函数

- 递归地定义自然数集合 \mathbf{N} 上的函数。
 - 基础步骤：指定这个函数在0处的值；
 - 递归步骤：给出从较小处的值来求出当前值的规则。
- 例如：阶乘函数 $F(n) = n!$ 的递归定义
 - $F(0) = 1$
 - $F(n) = n \cdot F(n - 1)$ for $n > 0$

显然, 用数学归纳法可以证明这样的函数是well-defined.



Fibonacci 序列

- Fibonacci 序列 $\{f_n\}$ 定义如下

- $f_0 = 0,$
- $f_1 = 1,$
- $f_n = f_{n-1} + f_{n-2}$, 对任意 $n \geq 2$.

- 其前几个数

- $0, 1, 1, 2, 3, 5, 8, \dots$

- 证明：对任意 $n \geq 0$, $f_n = \frac{\alpha^n - \beta^n}{\alpha - \beta}$ (*)

$$\text{其中, } \alpha = \frac{1 + \sqrt{5}}{2}, \beta = \frac{1 - \sqrt{5}}{2}.$$



归纳证明: Fibonacci 序列

证明: 对 n 做归纳.

基础步骤: 当 $n = 0, 1$ 时, 易验证(*)式成立。

归纳步骤: 假设(*)式对于 $n \leq k$ 时成立 ($1 \leq k$),
现证明其对于 $n = k + 1$ 时成立:

$$\begin{aligned} f_{k+1} &= f_k + f_{k-1} \\ &= \frac{\alpha^k - \beta^k}{\alpha - \beta} + \frac{\alpha^{k-1} - \beta^{k-1}}{\alpha - \beta} \\ &= \frac{(\alpha^k + \alpha^{k-1}) - (\beta^k + \beta^{k-1})}{\alpha - \beta} \\ &= \frac{\alpha^{k+1} - \beta^{k+1}}{\alpha - \beta}. \end{aligned}$$

注意: $\alpha^2 = \alpha + 1$,
且对任意 $n \geq 1$
 $\alpha^{n+1} = \alpha^n + \alpha^{n-1}$.

于是我们证明了(*)式对于所有自然数 n 成立. ■



递归定义集合

- 字母表 Σ 上的**字符串**集合 Σ^*

- 基础步骤: $\lambda \in \Sigma^*$ (λ 表示空串);
- 递归步骤: 若 $\omega \in \Sigma^*$ 且 $x \in \Sigma$, 则 $\omega x \in \Sigma^*$.

隐含规则: Σ^* 中所有元素均系通过有限次应用这两条规则得到.

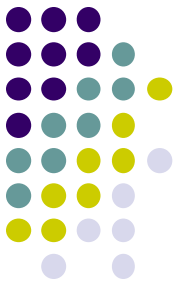
或曰, Σ^* 是满足这两个条件的所有集合的交集.

- 字符串的**长度** (Σ^* 上的函数 l)

- 基础步骤: $l(\lambda)=0$;
- 递归步骤: $l(\omega x) = l(\omega) + 1$, 若 $\omega \in \Sigma^*$ 且 $x \in \Sigma$.

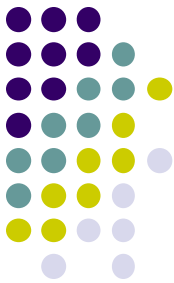
- Σ^* 上的字符串**连接运算** (Concatenation)

- 基础步骤: 若 $\omega \in \Sigma^*$, 则 $\omega \cdot \lambda = \omega$;
- 递归步骤: 若 $\omega_1 \in \Sigma^*$ 且 $\omega_2 \in \Sigma^*$ 以及 $x \in \Sigma$,
则 $\omega_1 \cdot (\omega_2 x) = (\omega_1 \cdot \omega_2) x$. // $\omega_1 \cdot \omega_2$ 也常写成 $\omega_1 \omega_2$



递归定义集合

- 复合命题的合式公式
 - 基础步骤： \mathbf{T} , \mathbf{F} , s 都是合式公式, 其中 s 是命题变元;
 - 递归步骤： 若 E 和 F 是合式公式, 则 $(\neg E)$ 、 $(E \wedge F)$ 、 $(E \vee F)$ 、 $(E \rightarrow F)$ 和 $(E \leftrightarrow F)$ 都是合式公式.
 - 排斥规则： 合适公式仅限于此.



结构归纳法

- 关于递归定义的集合的命题，进行**结构归纳**证明。
 - 基础步骤：证明对于初始元素来说，命题成立；
 - 递归步骤：针对生产新元素的规则，若相关元素满足命题，则新元素也满足命题
- 结构归纳法的有效性源于自然数上的数学归纳法
 - 即, 对应用规则(上述递归步骤)的次数做归纳.



结构归纳法（举例）

- $l(xy) = l(x) + l(y)$, x 和 y 属于 Σ^* 。

证明:

设 $P(y)$ 表示: 对于任意的 x 属于 Σ^* , 有 $l(xy) = l(x) + l(y)$. 我们对 y 做归纳.

基础步骤: $P(\lambda)$ 成立: 对于任意的 x 属于 Σ^* , 有 $l(x\lambda) = l(x) + l(\lambda)$.

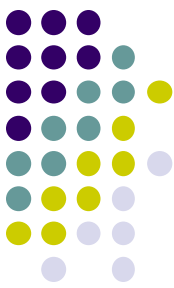
递归步骤: 假设 $P(y)$ 为真, a 属于 Σ , 要证 $P(ya)$ 为真. 即要证:

对于任意的 x 属于 Σ^* , 有 $l(xya) = l(x) + l(ya)$.

$P(y)$ 为真, $l(xy) = l(x) + l(y)$; 于是

$$l(xya) = l(xy) + 1 = l(x) + l(y) + 1 = l(x) + l(ya)$$

于是 $P(y)$ 对于所有的 y 属于 Σ^* 成立, 即原题成立. ■



广义归纳

- 集合 X 上的良基关系($<$): X 的每个非空子集都有极小元.

(x 是极小元是指不存在 $y \in X$ 使得 $y < x$)

可理解为“ X 中不存在无限下降序列 $x_0 > x_1 > x_2 > \dots$ ”.

例如: 自然数集合上的 $<$ 关系是良基的; \leq 不是. 实数集合上的 $<$ 也不是.

(关于“关系”和“序”的概念将在后续课程讨论.)

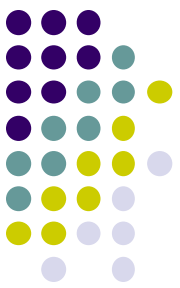
- 广义归纳:

对于一个性质 P , 一个集合 X , 及其上的良基关系 $<$,

基础步骤: $P(x)$ 对所有 X 上的极小元 x 成立.

归纳步骤: 如果 $P(x)$ 对所有 $y < x'$ 成立, 那么 $P(x')$ 成立.

于是 $P(x)$ 对所有 $x \in X$ 成立.



广义归纳(举例)

- 考虑如下程序:

```
s(x, y):  
    if (x == 0 && y == 0) return 0;  
    else if (y == 0) return s(x-1, y) + 1;  
    else return s(x, y-1) + y;
```

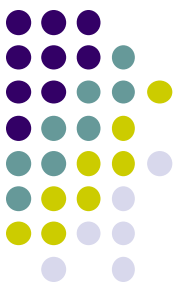
试说明该程序对所有的自然数 x 和 y , 返回 $y(y+1)/2 + x$.

- 解: 对于 $(x, y), (x', y') \in \mathbf{N}^2$, 定义 $(x, y) < (x', y')$ iff. $x + y < x' + y'$.

易见 $<$ 是良基的 (不存在无限下降序列 $(x_0, y_0) > (x_0, y_0) > (x_0, y_0) > \dots$)

令 $P(x, y)$ 表示 “ $s(x, y)$ 返回 $y(y+1)/2 + x$ ” .

现以广义归纳法证明对于任意的自然数 x 和 y , $P(x, y)$ 成立.



续上页

基础步骤: $P(0,0)$ 成立, 这是因为由于 $x == 0$ 且 $y == 0$,
 $s(0, 0) = 0 \cdot (0+1)/2 + 0 = 0$.

归纳步骤: 证明 $\forall x', y' \in \mathbf{N} (x' + y' < x + y \Rightarrow P(x', y')) \Rightarrow P(x, y)$

情形1: $y = 0$

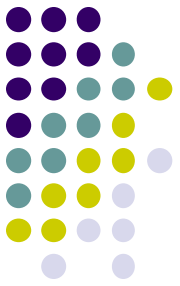
$$\begin{aligned} s(x, y) &= 1 + s(x-1, y) && (\text{since } x \neq 0 \text{ and } y == 0) \\ &= 1 + y(y+1)/2 + x - 1 && (\text{by the inductive hypothesis}) \\ &= y(y+1)/2 + x. && () \end{aligned}$$

情形2:

$$\begin{aligned} s(x, y) &= s(x, y-1) + y && (\text{since } x \neq 0 \text{ and } y \neq 0) \\ &= y(y-1)/2 + x + y && (\text{by the inductive hypothesis}) \\ &= (y^2 - y + 2y)/2 + x && () \\ &= y(y+1)/2 + x. && () \end{aligned}$$

无论哪种情形 $s(x, y)$ 都返回 $y(y+1)/2 + x$.

于是我们归纳证明了对于任意的自然数 x 和 y , $P(x, y)$ 成立.



证明程序正确性与复杂度



Hoare三元组与程序正确性

- **Hoare三元组** $\{P\}S\{Q\}$:

- S是一段程序; P 和 Q 是关于程序中变量的断言(陈述), 分别称为**前置断言(precondition)**和**后置断言(postcondition)**.
- 这个三元组的意思是说, 如果在S执行之前程序变量使得 P 成立, 则S运行完后 Q 成立. (假设 P 成立是S的**权利**, 确保 Q 成立是S的**义务**.)
- 这称为“**部分正确性**”, 因为S是否能够执行完成需要另行说明. 程序的“**完全正确性**”还需要说明程序在有限步内终止.
- C. A. R. Hoare (Tony) 在Robert. W. Floyd 的工作基础上给出了一个形式逻辑系统, 用以形式地证明程序的正确性.
- Hoare当年写作 $P \{S\} Q$, 我们教材也这样写. 现在 $\{P\}S\{Q\}$ 更常见.
- 我们这里仅直观地简单介绍一些相关思想.



欧几里得算法的正确性

```
function gcd( $a, b$ ) // 不全为0的自然数  
 $\{P: a_0 \in \mathbf{N}, b_0 \in \mathbf{N}, \neg(a_0 = 0 \wedge b_0 = 0)\}$   
  while  $b \neq 0$   $\{INV\}$   
     $t := b$   
     $b := a \bmod b$   
     $a := t$   
   $\{Q: a_k = \text{gcd}(a_0, b_0)\}$   
  return  $a$ 
```

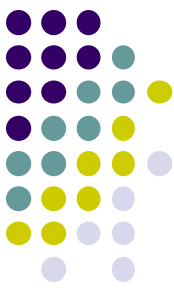
程序的终止性:
注意到 $b_{k+1} < b_k$ 且 $b_n \geq 0$ 即可.

- 如何证明? 我们不知道while循环会执行多少次.
- 我们可对执行次数做“归纳”. 问题是, “归纳假设”是什么?
 $INV: \text{gcd}(a_k, b_k) = \text{gcd}(a_0, b_0)$
- 这样循环开始时, $P \rightarrow INV$, 循环结束时, 由 $INV \wedge b_k = 0$, 即 Q 成立.
- 问题是如何证明“归纳步骤”.
即要证明:
 $\{\text{gcd}(a_k, b_k) = \text{gcd}(a_0, b_0) \wedge b_k \neq 0\}$
 $t := b$
 $b := a \bmod b$
 $a := t$
 $\{INV: \text{gcd}(a_{k+1}, b_{k+1}) = \text{gcd}(a_0, b_0)\}$



欧几里得算法的复杂度

- 拉梅定理: 设 a 和 b 是满足 $a \geq b$ 的正整数。则欧几里德算法为求出 $\gcd(a, b)$ 而使用除法的次数小于或等于 b 的十进制位数的5倍。 $5(\lfloor \log_{10} b \rfloor + 1)$
- 令 $r_0 = a, r_1 = b$.
- $r_0 = r_1 q_1 + r_2 \quad 0 \leq r_2 < r_1$
- $r_1 = r_2 q_2 + r_3 \quad 0 \leq r_3 < r_2$
- ...
- $r_{n-1} = r_n q_n + r_{n+1} \quad 0 = r_{n+1} < r_n$
- $\gcd(a, b) = r_n$ 使用了 n 次除法
- Let $r_0 = a, r_1 = b$.
- $q_i \geq 1$ for $1 \leq i < n$
- $q_n \geq 2$ because $q_n = r_{n-1}/r_n > 1$
- $r_n \geq 1 = f_2, r_{n-1} \geq 2r_n \geq 2 = f_3$
- $b = r_1 \geq r_2 + r_3 \geq f_n + f_{n-1} = f_{n+1} > \alpha^{n-1}$
- $\log_{10} b > (n-1)\log_{10} \alpha$ for $n \geq 2$
- $\log_{10} \alpha > 1/5$



小结

- 归纳与递归不仅是重要的数学工具, 也是计算机科学中的基本思维方式
- 各种形式的数学归纳法和良序原理的应用
- 递归定义集合, 归纳证明性质
- 归纳法可用于程序正确性和复杂度的证明