离散数学

（第 3 版）

智能科学与技术学院 2024 级

目录

目录

10.1 加法法则与乘法法则

加法法则 :设事件 A 有 m 种产生方式,事件 B 有 n 种产生方式,当 A 与 B 产生的方式不重叠时,"事件 A 或 B"有 $m+n$ 种产生方式.

10.1 加法法则与乘法法则

加法法则：设事件 A 有 m 种产生方式，事件 B 有 n 种产生方式，当 A 与 B 产生的方式不重叠时，“事件 A 或 B”有 $m+n$ 种产生方式.

加法法则适用的条件是事件 A 与 B 产生的方式不能重叠. 也就是说，每一种产生的方式不能同时属于两个事件.

10.1 加法法则与乘法法则

加法法则：设事件 A 有 m 种产生方式，事件 B 有 n 种产生方式，当 A 与 B 产生的方式不重叠时，“事件 A 或 B”有 $m+n$ 种产生方式.

加法法则适用的条件是事件 A 与 B 产生的方式不能重叠. 也就是说，每一种产生的方式不能同时属于两个事件.

加法法则可以推广到 n 个事件的情形. 设 $A_1, A_2, \cdots, A_n$ 是 n 个事件，它们的产生方式分别有 $p_1, p_2, \cdots, p_n$ 种，当其中任何两个事件产生的方式都不重叠时，事件“A_1 或 A_2 或 $\cdots$ 或 A_n”有 $p_1 + p_2 + \cdots + p_n$ 种产生的方式.

10.1 加法法则与乘法法则

加法法则：设事件 A 有 m 种产生方式，事件 B 有 n 种产生方式，当 A 与 B 产生的方式不重叠时，“事件 A 或 B”有 $m+n$ 种产生方式.

加法法则适用的条件是事件 A 与 B 产生的方式不能重叠. 也就是说，每一种产生的方式不能同时属于两个事件.

加法法则可以推广到 n 个事件的情形. 设 $A_1, A_2, \cdots, A_n$ 是 n 个事件，它们的产生方式分别有 $p_1, p_2, \cdots, p_n$ 种，当其中任何两个事件产生的方式都不重叠时，事件“A_1 或 A_2 或 $\cdots$ 或 A_n”有 $p_1 + p_2 + \cdots + p_n$ 种产生的方式.

乘法法则：设事件 A 有 m 种产生方式，事件 B 有 n 种产生方式，当 A 与 B 产生的方式彼此独立时，“事件 A 与 B”有 mn 种产生方式.

10.1 加法法则与乘法法则

加法法则：设事件 A 有 m 种产生方式，事件 B 有 n 种产生方式，当 A 与 B 产生的方式不重叠时，“事件 A 或 B”有 $m+n$ 种产生方式.

加法法则适用的条件是事件 A 与 B 产生的方式不能重叠. 也就是说，每一种产生的方式不能同时属于两个事件.

加法法则可以推广到 n 个事件的情形. 设 $A_1, A_2, \cdots, A_n$ 是 n 个事件，它们的产生方式分别有 $p_1, p_2, \cdots, p_n$ 种，当其中任何两个事件产生的方式都不重叠时，事件“A_1 或 A_2 或 $\cdots$ 或 A_n”有 $p_1 + p_2 + \cdots + p_n$ 种产生的方式.

乘法法则：设事件 A 有 m 种产生方式，事件 B 有 n 种产生方式，当 A 与 B 产生的方式彼此独立时，“事件 A 与 B”有 mn 种产生方式.

乘法法则适用的条件是事件 A 与 B 产生的方式彼此独立. 换句话说，事件 A 与事件 B 对产生方式的选择彼此没有影响.

10.1 加法法则与乘法法则

加法法则：设事件 A 有 m 种产生方式，事件 B 有 n 种产生方式，当 A 与 B 产生的方式不重叠时，“事件 A 或 B”有 $m+n$ 种产生方式.

加法法则适用的条件是事件 A 与 B 产生的方式不能重叠. 也就是说，每一种产生的方式不能同时属于两个事件.

加法法则可以推广到 n 个事件的情形. 设 $A_1, A_2, \cdots, A_n$ 是 n 个事件，它们的产生方式分别有 $p_1, p_2, \cdots, p_n$ 种，当其中任何两个事件产生的方式都不重叠时，事件“A_1 或 A_2 或 $\cdots$ 或 A_n”有 $p_1 + p_2 + \cdots + p_n$ 种产生的方式.

乘法法则：设事件 A 有 m 种产生方式，事件 B 有 n 种产生方式，当 A 与 B 产生的方式彼此独立时，“事件 A 与 B”有 mn 种产生方式.

乘法法则适用的条件是事件 A 与 B 产生的方式彼此独立. 换句话说，事件 A 与事件 B 对产生方式的选择彼此没有影响.

乘法法则也可以推广到 n 个事件的情况. 设 $A_1, A_2, \cdots, A_n$ 是 n 个事件，它们的产生方式分别有 $p_1, p_2, \cdots, p_n$ 种，当其中任何两个事件产生的方式都彼此独立时，事件“A_1 与 A_2 与 $\cdots$ 与 A_n”有 $p_1 p_2 \cdots p_n$ 种产生的方式.

例 1.1.1

设 A 为 n 元集, 问:

1. A 上的自反关系有多少个?
2. A 上的对称关系有多少个?
3. A 上的反对称关系有多少个?
4. A 上的函数有多少个? 其中双射函数有多少个?

例 1.1.1

设 A 为 n 元集,问:

1. A 上的自反关系有多少个?
2. A 上的对称关系有多少个?
3. A 上的反对称关系有多少个?
4. A 上的函数有多少个?其中双射函数有多少个?

解

(1) 在 A 上自反关系的关系矩阵中,主对角线元素都是 1,其他位置的元素可以是 1,也可以是 0,有 2 种选择. 这种位置有 n^2-n 个,根据乘法法则,自反关系的个数是 2^{n^2-n}.

例 1.1.1

设 A 为 n 元集,问:

1. A 上的自反关系有多少个?
2. A 上的对称关系有多少个?
3. A 上的反对称关系有多少个?
4. A 上的函数有多少个?其中双射函数有多少个?

解

(1) 在 A 上自反关系的关系矩阵中,主对角线元素都是 1,其他位置的元素可以是 1,也可以是 0,有 2 种选择. 这种位置有 $n^2 - n$ 个,根据乘法法则,自反关系的个数是 2^{n^2-n}.

(2) 考虑 A 上对称关系的矩阵. 先考虑主对角线上的元素. 对于主对角线的每个位置,元素可以选择 0 或 1,有 2 种选法. 再考虑不在主对角线上的元素,它们的值的选择并不是完全独立的. 因为矩阵是对称的,i 行 j 列的元素 r_{ij} 必须与 j 行 i 列的元素 r_{ji} 相等. 因此,当矩阵的上三角元素(或者下三角元素)的值确定以后,另一半对称位置的元素就完全确定了. 这种能够独立选择 0 或者 1 的位置有 $(n^2 - n)/2$ 个. 加上主对角线的 n 个位置,总计 $(n^2 + n)/2$ 个位置,根据乘法法则,构成矩阵的方法数是 $2^{(n^2+n)/2}$.

例1.1.1(续)

(3) 类似于 (2) 的分析，也分两步考虑，区别在于对非主对角线位置元素取值的约束条件不同. 将这些位置分成 $(n^2-n)/2$ 组，每组包含处在对称位置的两个元素 r_{ij} 和 r_{ji}. 根据反对称的性质，r_{ij} 与 r_{ji} 的取值有以下 3 种可能：

$$\begin{cases} r_{ij}=1, \\ r_{ji}=0, \end{cases} \qquad \begin{cases} r_{ij}=0, \\ r_{ji}=1, \end{cases} \qquad \begin{cases} r_{ij}=0, \\ r_{ji}=0. \end{cases}$$

因此所有这些位置元素的选择方法数为 $3^{(n^2-n)/2}$. 再考虑到主对角线元素的选取，由乘法法则总方法数为 $2^n\cdot 3^{(n^2-n)/2}$.

例1.1.1(续)

(3) 类似于 (2) 的分析，也分两步考虑，区别在于对非主对角线位置元素取值的约束条件不同. 将这些位置分成 $(n^2-n)/2$ 组，每组包含处在对称位置的两个元素 r_{ij} 和 r_{ji}. 根据反对称的性质，r_{ij} 与 r_{ji} 的取值有以下 3 种可能：

$$\begin{cases} r_{ij}=1, \\ r_{ji}=0, \end{cases} \quad \begin{cases} r_{ij}=0, \\ r_{ji}=1, \end{cases} \quad \begin{cases} r_{ij}=0, \\ r_{ji}=0. \end{cases}$$

因此所有这些位置元素的选择方法数为 $3^{(n^2-n)/2}$. 再考虑到主对角线元素的选取，由乘法法则总方法数为 $2^n\cdot 3^{(n^2-n)/2}$.

(4) 设 $A=\{x_1,x_2,\cdots,x_n\}$，任何 A 上的函数 $f:A\to A$ 都可以表示成下述形式：$f=\{\langle x_1,y_1\rangle,\langle x_2,y_2\rangle,\cdots,\langle x_n,y_n\rangle\}$，其中每个 $y_i, i=1,2,\cdots,n$，有 n 种可能的选择. 根据乘法法则，有 n^n 个不同的函数. 如果 f 是双射的，那么当 y_1 确定以后，y_2 只有 $n-1$ 种可能的取值. 通过类似的分析可以知道，y_3 只有 $n-2$ 种可能的取值，$\cdots$，y_n 只有 1 种取值. 根据乘法法则，构成双射函数的方法数是 $n(n-1)(n-2)\cdots 1=n!$.

例 1.1.2

根据 *IPv4* 网络协议，每台计算机的地址是由 *32* 位二进制数字构成的串．如图 *1.1.1* 所示，A 类地址第一位是 *0*，接着 *7* 位网络标识，再接着 *24* 位主机标识．B 类地址前两位是 *10*，接着 *14* 位网络标识，再接着 *16* 位主机标识．C 类地址前 *3* 位是 *110*，接着 *21* 位网络标识，再接着 *8* 位主机标识．此外，A 类地址中全 *1* 不能做网络标识，在 *3* 类地址中全 *0* 和全 *1* 都不能作为主机标识．问：按照 *IPv4* 协议，在 *Internet* 中有多少个有效的计算机地址？

位数　0　1　2　3　4　8　16　24　31

A类	0	网络标识	主机标识		
B类	1	0	网络标识	主机标识	
C类	1	1	0	网络标识	主机标识

图 1.1.1

例1.1.2(续)

解

令 N 是 Internet 上计算机的有效地址数, N_A, N_B 和 N_C 分别表示 A 类、B 类和 C 类的有效地址数. 由加法法则, $N = N_A + N_B + N_C$.

例1.1.2(续)

解

令 N 是 Internet 上计算机的有效地址数,N_A,N_B 和 N_C 分别表示 A 类、B 类和 C 类的有效地址数. 由加法法则,$N = N_A + N_B + N_C$.

为了找到 N_A,由于 1111111 是无效的,故存在 $2^7 - 1 = 127$ 个 A 类的网络标识,对于每个网络标识,存在 $2^{24} - 2 = 16\ 777\ 214$ 个主机标识,这是由于全 0 和全 1 所组成的主机标识是无效的. 因此 $N_A = 127 \times 16\ 777\ 214 = 2\ 130\ 706\ 178$.

例1.1.2(续)

解

令 N 是 Internet 上计算机的有效地址数，N_A，N_B 和 N_C 分别表示 A 类、B 类和 C 类的有效地址数. 由加法法则，$N = N_A + N_B + N_C$.

为了找到 N_A，由于 1111111 是无效的，故存在 $2^7 - 1 = 127$ 个 A 类的网络标识，对于每个网络标识，存在 $2^{24} - 2 = 16\ 777\ 214$ 个主机标识，这是由于全 0 和全 1 所组成的主机标识是无效的. 因此 $N_A = 127 \times 16\ 777\ 214 = 2\ 130\ 706\ 178$.

为了找到 N_B 和 N_C，首先注意到存在 $2^{14} = 16\ 384$ 个 B 类网络标识和 $2^{21} = 2\ 097\ 152$ 个 C 类网络标识. 对每个 B 类网络标识存在着 $2^{16} - 2 = 65\ 534$ 个主机标识，而对每个 C 类网络标识存在着 $2^8 - 2 = 254$ 个主机标识，这也是考虑到全 0 和全 1 组成的主机标识是无效的. 因而，$N_B = 1\ 073\ 709\ 056$, $N_C = 532\ 676\ 608$. 可以断言 IPv4 中计算机的有效地址总数是

$$
\begin{aligned}
N = N_A + N_B + N_C &= 2\ 130\ 706\ 178 + 1\ 073\ 709\ 056 + 532\ 676\ 608 \\
&= 3\ 737\ 091\ 842.
\end{aligned}
$$

例1.1.2(续)

解

令 N 是 Internet 上计算机的有效地址数，N_A，N_B 和 N_C 分别表示 A 类、B 类和 C 类的有效地址数. 由加法法则，$N = N_A + N_B + N_C$.

为了找到 N_A，由于 1111111 是无效的，故存在 $2^7 - 1 = 127$ 个 A 类的网络标识，对于每个网络标识，存在 $2^{24} - 2 = 16\ 777\ 214$ 个主机标识，这是由于全 0 和全 1 所组成的主机标识是无效的. 因此 $N_A = 127 \times 16\ 777\ 214 = 2\ 130\ 706\ 178$.

为了找到 N_B 和 N_C，首先注意到存在 $2^{14} = 16\ 384$ 个 B 类网络标识和 $2^{21} = 2\ 097\ 152$ 个 C 类网络标识. 对每个 B 类网络标识存在着 $2^{16} - 2 = 65\ 534$ 个主机标识，而对每个 C 类网络标识存在着 $2^8 - 2 = 254$ 个主机标识，这也是考虑到全 0 和全 1 组成的主机标识是无效的. 因而，$N_B = 1\ 073\ 709\ 056$, $N_C = 532\ 676\ 608$. 可以断言 IPv4 中计算机的有效地址总数是

$$\begin{aligned} N &= N_A + N_B + N_C = 2\ 130\ 706\ 178 + 1\ 073\ 709\ 056 + 532\ 676\ 608 \\ &= 3\ 737\ 091\ 842. \end{aligned}$$

面向计算机的广泛使用，这些地址总数已经显得不够用了，扩展后的 IPv6 采用 128 位地址格式，能够提供更多的有效地址. □

10.2 排列与组合

排列和组合的计数是基本的计数问题. 根据从集合中选择元素的有序与无序、是否允许重复等限制条件，可以将这个问题划分成 4 个子类型——集合的排列、集合的组合、多重集的排列、多重集的组合.

先考虑不允许重复的选取——集合的排列与组合的计数.

10.2 排列与组合

排列和组合的计数是基本的计数问题. 根据从集合中选择元素的有序与无序、是否允许重复等限制条件, 可以将这个问题划分成 4 个子类型——集合的排列、集合的组合、多重集的排列、多重集的组合.

先考虑不允许重复的选取——集合的排列与组合的计数.

定义 1.2.1

设 S 为 n 元集,

1. 从 S 中有序选取的 r 个元素称作 S 的一个 r 排列. S 的不同 r 排列总数记作 $P(n,r)$. $r=n$ 时的排列称作 S 的 全排列.
2. 从 S 中无序选取的 r 个元素称作 S 的一个 r 组合. S 的不同 r 组合总数记作 $C(n,r)$.

定理 1.2.1

设 n, r 为自然数，规定 $0! = 1$，则

1. $P(n,r) = \begin{cases} \frac{n!}{(n-r)!}, & r \leq n, \\ 0, & r > n. \end{cases}$
2. $C(n,r) = \begin{cases} \frac{P(n,r)}{r!} = \frac{n!}{r!(n-r)!}, & r \leq n, \\ 0, & r > n. \end{cases}$

定理 1.2.1

设 n, r 为自然数，规定 $0! = 1$，则

1. $P(n,r) = \begin{cases} \frac{n!}{(n-r)!}, & r \le n, \\ 0, & r > n. \end{cases}$
2. $C(n,r) = \begin{cases} \frac{P(n,r)}{r!} = \frac{n!}{r!(n-r)!}, & r \le n, \\ 0, & r > n. \end{cases}$

证明.

$r > n$ 时不存在满足条件的排列和组合，下面考虑 $r \le n$ 的情况.

(1) 首先确定排列中的第一个元素，有 n 种选择的方式.

定理 1.2.1

设 n, r 为自然数，规定 $0! = 1$，则

1. $$P(n,r) = \begin{cases} \frac{n!}{(n-r)!}, & r \le n, \\ 0, & r > n. \end{cases}$$
2. $$C(n,r) = \begin{cases} \frac{P(n,r)}{r!} = \frac{n!}{r!(n-r)!}, & r \le n, \\ 0, & r > n. \end{cases}$$

证明.

$r > n$ 时不存在满足条件的排列和组合，下面考虑 $r \le n$ 的情况.

(1) 首先确定排列中的第一个元素，有 n 种选择的方式.

然后确定排列的第 2 个元素，它只能取自剩下的 $n-1$ 个元素，有 $n-1$ 种选法.

类似地，选择第 3 个元素，第 4 个元素，$\cdots$，第 r 个元素的方式数依次为 $n-2, n-3, \cdots, n-r+1$. 根据乘法法则，总的选法数为

$$n(n-1)(n-2)\cdots(n-r+1) = \frac{n!}{(n-r)!}.$$

定理 1.2.1

设 n, r 为自然数，规定 $0! = 1$，则

1. $P(n,r) = \begin{cases} \frac{n!}{(n-r)!}, & r \leq n, \\ 0, & r > n. \end{cases}$
2. $C(n,r) = \begin{cases} \frac{P(n,r)}{r!} = \frac{n!}{r!(n-r)!}, & r \leq n, \\ 0, & r > n. \end{cases}$

证明.

$r > n$ 时不存在满足条件的排列和组合，下面考虑 $r \leq n$ 的情况.

(1) 首先确定排列中的第一个元素，有 n 种选择的方式.

然后确定排列的第 2 个元素，它只能取自剩下的 $n-1$ 个元素，有 $n-1$ 种选法.

类似地，选择第 3 个元素，第 4 个元素，$\cdots$，第 r 个元素的方式数依次为 $n-2, n-3, \cdots, n-r+1$. 根据乘法法则，总的选法数为

$$n(n-1)(n-2)\cdots(n-r+1) = \frac{n!}{(n-r)!}.$$

(2) 分两步构成 r 排列. 首先无序地选出 r 个元素，然后再构造这 r 个元素的全排列. 无序选择 r 个元素的方法数是 $C(n,r)$；针对每种选法，能构造 $r!$ 个不同的全排列. 根据乘法法则，不同的 r 排列数满足 $P(n,r) = C(n,r)r!$.

定理 1.2.1

设 n, r 为自然数，规定 $0! = 1$，则

1. $P(n,r) = \begin{cases} \frac{n!}{(n-r)!}, & r \leq n, \\ 0, & r > n. \end{cases}$
2. $C(n,r) =$
$\begin{cases} \frac{P(n,r)}{r!} = \frac{n!}{r!(n-r)!}, & r \leq n, \\ 0, & r > n. \end{cases}$

证明.

$r > n$ 时不存在满足条件的排列和组合，下面考虑 $r \leq n$ 的情况.

(1) 首先确定排列中的第一个元素，有 n 种选择的方式.

然后确定排列的第 2 个元素，它只能取自剩下的 $n-1$ 个元素，有 $n-1$ 种选法. 类似地，选择第 3 个元素，第 4 个元素，$\cdots$，第 r 个元素的方式数依次为 $n-2, n-3, \cdots, n-r+1$. 根据乘法法则，总的选法数为

$$n(n-1)(n-2)\cdots(n-r+1) = \frac{n!}{(n-r)!}.$$

(2) 分两步构成 r 排列. 首先无序地选出 r 个元素，然后再构造这 r 个元素的全排列. 无序选择 r 个元素的方法数是 $C(n,r)$；针对每种选法，能构造 $r!$ 个不同的全排列. 根据乘法法则，不同的 r 排列数满足 $P(n,r) = C(n,r)r!$.

组合数 $C(n,r)$ 恰好是二项式 $(x+y)^n$ 的展开式中 $x^r y^{n-r}$ 项的系数，通常称作*二项式系数*，有时也记作 $\binom{n}{r}$.

环排列

推论 1.2.1

元素依次排成一个圆圈的排列称作环排列. S 的 r 环排列数等于 $P(n,r)/r$.

环排列

推论 1.2.1

元素依次排成一个圆圈的排列称作环排列. S 的 r 环排列数等于 $P(n,r)/r$.

证明.

设线排列的 r 个元素依次为 $a_1, a_2, \cdots, a_r$, 将 a_1 接在 a_r 的后边就组成一个环排列. 只要相邻关系不变, 这 r 个元素中的任何一个作为线排列的首元素, 首尾相连所构成的环排列都相同. 因此环排列数是线排列数的 $1/r$. $\square$

环排列

推论 1.2.1

元素依次排成一个圆圈的排列称作环排列. S 的 r 环排列数等于 $P(n,r)/r$.

证明.

设线排列的 r 个元素依次为 $a_1, a_2, \cdots, a_r$,将 a_1 接在 a_r 的后边就组成一个环排列. 只要相邻关系不变,这 r 个元素中的任何一个作为线排列的首元素,首尾相连所构成的环排列都相同. 因此环排列数是线排列数的 $1/r$. □

下面推论1.2.2中的公式均可通过把定理 1.2.1的公式代入右边化简得到. 在此,我们将介绍一种组合分析的证明方法. 所谓组合分析,就是根据等式类型设计一个组合计数问题,使得公式两边都对应于这个问题的计数结果. 我们将以推论中(2) 和 (3) 为例来说明这种证明方法.

推论 1.2.2

设 n, r 为正整数，则

1. $C(n,r) = \frac{n}{r}C(n-1,r-1)$.
2. $C(n,r) = C(n,n-r)$.
3. $C(n,r) = C(n-1,r-1) + C(n-1,r)$.

推论 1.2.2

设 n, r 为正整数，则

1. $C(n,r) = \frac{n}{r}C(n-1,r-1)$.
2. $C(n,r) = C(n,n-r)$.
3. $C(n,r) = C(n-1,r-1) + C(n-1,r)$.

证明.

(2) 设 $S=\{1,2,\cdots,n\}$ 是 n 元集合，对于 S 的任意 r 组合 $A=\{a_1,a_2,\cdots,a_r\}$，都存在一个 S 的 $n-r$ 组合 $S-A$ 与之对应. 显然不同的 r 组合对应了不同的 $n-r$ 组合，反之也对，因此 S 的 r 组合数恰好与 S 的 $n-r$ 组合数相等.

推论 1.2.2

设 n, r 为正整数，则

1. $C(n,r) = \frac{n}{r}C(n-1,r-1)$.
2. $C(n,r) = C(n,n-r)$.
3. $C(n,r) = C(n-1,r-1) + C(n-1,r)$.

证明.

(2) 设 $S = \{1,2,\cdots,n\}$ 是 n 元集合，对于 S 的任意 r 组合 $A = \{a_1,a_2,\cdots,a_r\}$，都存在一个 S 的 $n-r$ 组合 $S-A$ 与之对应. 显然不同的 r 组合对应了不同的 $n-r$ 组合，反之也对，因此 S 的 r 组合数恰好与 S 的 $n-r$ 组合数相等.

(3) 考虑 (2) 中的 S 集合，将 S 的所有 r 组合划分成两类：包含 1 的 r 组合，不含 1 的 r 组合. 如果一个 r 组合包含 1，那么它的其余 $r-1$ 个元素取自 $\{2,3,\cdots,n\}$，有 $C(n-1,r-1)$ 种取法；如果一个 r 组合不含 1，那么它的其余 r 个元素都取自 $\{2,3,\cdots,n\}$，有 $C(n-1,r)$ 种取法. 根据加法法则，S 的 r 组合的总数等于 $C(n-1,r-1)+C(n-1,r)$. □

杨辉三角

上述公式有着广泛的应用. 例如, 利用 (2) 将 $C(100, 98)$ 写作 $C(100, 2)$. 而 (3) 中的公式就是我国古代著名的杨辉三角形(如图 1.2.1 所示), 也称作 Pascal 公式. 这些公式广泛用于组合公式的化简和恒等式的证明.

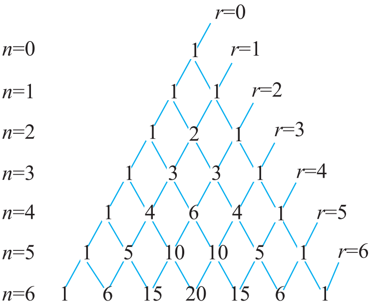


图 1.2.1

多重集的排列与组合

为了讨论允许重复的选取问题需要引入多重集的概念.

多重集 $S = \{n_1 \cdot a_1, n_2 \cdot a_2, \cdots, n_k \cdot a_k\}$,这里 $a_1, a_2, \cdots, a_k$ 是 k 种不同的元素, n_i 表示 a_i 在 S 中出现的次数,称作 a_i 的 重复度 ,其中 $0 < n_i \leqslant +\infty$, $i = 1, 2, \cdots, k$. 当 $n_i = +\infty$ 时表示有足够多的 a_i 以备选取.

多重集的排列与组合

为了讨论允许重复的选取问题需要引入多重集的概念.

多重集 $S = \{n_1 \cdot a_1, n_2 \cdot a_2, \cdots, n_k \cdot a_k\}$, 这里 $a_1, a_2, \cdots, a_k$ 是 k 种不同的元素, n_i 表示 a_i 在 S 中出现的次数, 称作 a_i 的重复度, 其中 $0 < n_i \leqslant +\infty$, $i = 1, 2, \cdots, k$. 当 $n_i = +\infty$ 时表示有足够多的 a_i 以备选取.

例如, $S_1 = \{2 \cdot a, 3 \cdot b, 5 \cdot c\}$, $S_2 = \{+\infty \cdot 1, +\infty \cdot 2, \cdots, +\infty \cdot k\}$ 都是多重集.

多重集的排列与组合

为了讨论允许重复的选取问题需要引入多重集的概念.

多重集 $S = \{n_1 \cdot a_1, n_2 \cdot a_2, \cdots, n_k \cdot a_k\}$,这里 $a_1, a_2, \cdots, a_k$ 是 k 种不同的元素,n_i 表示 a_i 在 S 中出现的次数,称作 a_i 的重复度,其中 $0 < n_i \leqslant +\infty$, $i = 1, 2, \cdots, k$. 当 $n_i = +\infty$ 时表示有足够多的 a_i 以备选取.

例如,$S_1 = \{2 \cdot a, 3 \cdot b, 5 \cdot c\}$,$S_2 = \{+\infty \cdot 1, +\infty \cdot 2, \cdots, +\infty \cdot k\}$ 都是多重集.

多重集 S 的子集也是多重集,可以记作 $A = \{x_1 \cdot a_1, x_2 \cdot a_2, \cdots, x_k \cdot a_k\}$,其中 $0 \leqslant x_i \leqslant n_i, i = 1, 2, \cdots, k$. 如果元素 a_i 不出现在子集 A 中,那么 $x_i = 0$.

多重集的排列与组合

为了讨论允许重复的选取问题需要引入多重集的概念.

多重集 $S=\{n_1\cdot a_1, n_2\cdot a_2,\cdots,n_k\cdot a_k\}$,这里 $a_1,a_2,\cdots,a_k$ 是 k 种不同的元素,n_i 表示 a_i 在 S 中出现的次数,称作 a_i 的重复度,其中 $0<n_i\leqslant+\infty$, $i=1,2,\cdots,k$. 当 $n_i=+\infty$ 时表示有足够多的 a_i 以备选取.

例如,$S_1=\{2\cdot a,3\cdot b,5\cdot c\}$,$S_2=\{+\infty\cdot 1,+\infty\cdot 2,\cdots,+\infty\cdot k\}$ 都是多重集.

多重集 S 的子集也是多重集,可以记作 $A=\{x_1\cdot a_1,x_2\cdot a_2,\cdots,x_k\cdot a_k\}$,其中 $0\leqslant x_i\leqslant n_i, i=1,2,\cdots,k$. 如果元素 a_i 不出现在子集 A 中,那么 $x_i=0$.

定义 1.2.2

设 $S=\{n_1\cdot a_1,n_2\cdot a_2,\cdots,n_k\cdot a_k\}$ 为多重集,$n=n_1+n_2+\cdots+n_k$ 表示 S 中元素的总数.

1. 从 S 中有序选取的 r 个元素称作多重集 S 的一个 r 排列. $r=n$ 的排列称作 S 的全排列.
2. 从 S 中无序选取的 r 个元素称作多重集 S 的一个 r 组合.

定理 1.2.2

设 $S = \{n_1 \cdot a_1, n_2 \cdot a_2, \cdots, n_k \cdot a_k\}$ 为多重集,

1. S 的全排列数是 $\frac{n!}{n_1!n_2!\cdots n_k!}$.
2. 若 $r \leqslant n_i, i = 1, 2, \cdots, k$,那么 S 的 r 排列数是 k^r.

定理 1.2.2

设 $S=\{n_1\cdot a_1, n_2\cdot a_2,\cdots,n_k\cdot a_k\}$ 为多重集，

1. S 的全排列数是 $\frac{n!}{n_1!n_2!\cdots n_k!}$.
2. 若 $r\leqslant n_i, i=1,2,\cdots,k$，那么 S 的 r 排列数是 k^r.

证明.

(1) 在 n 个位置中先选择 n_1 个放 a_1，有 $C(n,n_1)$ 种方法；再从剩下的 $n-n_1$ 个位置中选择 n_2 个放 a_2，有 $C(n-n_1,n_2)$ 种方法；⋯；最后在 $n-n_1-n_2-\cdots-n_{k-1}$ 个位置中选择 n_k 个放 a_k，有 $C(n-n_1-n_2-\cdots-n_{k-1},n_k)$ 种方法. 根据乘法法则，

$$
\begin{aligned}
&C(n,n_1)C(n-n_1,n_2)\cdots C(n-n_1-n_2-\cdots-n_{k-1},n_k)\\
=&\frac{n!}{n_1!(n-n_1)!}\frac{(n-n_1)!}{n_2!(n-n_1-n_2)!}\cdots\frac{(n-n_1-\cdots-n_{k-1})!}{n_k!0!}\\
=&\frac{n!}{n_1!n_2!\cdots n_k!}.
\end{aligned}
$$

定理 1.2.2

设 $S=\{n_1\cdot a_1, n_2\cdot a_2, \cdots, n_k\cdot a_k\}$ 为多重集，

1. S 的全排列数是 $\frac{n!}{n_1!n_2!\cdots n_k!}$.
2. 若 $r\leqslant n_i, i=1,2,\cdots,k$，那么 S 的 r 排列数是 k^r.

证明.

(1) 在 n 个位置中先选择 n_1 个放 a_1，有 $C(n,n_1)$ 种方法；再从剩下的 $n-n_1$ 个位置中选择 n_2 个放 a_2，有 $C(n-n_1,n_2)$ 种方法；$\cdots$；最后在 $n-n_1-n_2-\cdots-n_{k-1}$ 个位置中选择 n_k 个放 a_k，有 $C(n-n_1-n_2-\cdots-n_{k-1},n_k)$ 种方法. 根据乘法法则，

$$
\begin{aligned}
& C(n,n_1)C(n-n_1,n_2)\cdots C(n-n_1-n_2-\cdots-n_{k-1},n_k) \\
= & \frac{n!}{n_1!(n-n_1)!}\frac{(n-n_1)!}{n_2!(n-n_1-n_2)!}\cdots\frac{(n-n_1-\cdots-n_{k-1})!}{n_k!0!} \\
= & \frac{n!}{n_1!n_2!\cdots n_k!}.
\end{aligned}
$$

(2) r 个位置中的每个位置都有 k 种选法，由乘法法则得 S 的 r 排列数是 k^r.

多重集 $S=\{n_1\cdot a_1, n_2\cdot a_2, \cdots, n_k\cdot a_k\}$ 的全排列数也记作 $\binom{n}{n_1\, n_2 \cdots n_k}$，这个数恰好是多项式 $(x_1+x_2+\cdots+x_k)^n$ 的展开式中 $x_1^{n_1}x_2^{n_2}\cdots x_k^{n_k}$ 项的系数，也称作多项式系数. 关于它的性质在后面还会进一步加以讨论.

多重集 $S=\{n_1\cdot a_1,n_2\cdot a_2,\cdots,n_k\cdot a_k\}$ 的全排列数也记作 $\binom{n}{n_1\,n_2\cdots n_k}$，这个数恰好是多项式 $(x_1+x_2+\cdots+x_k)^n$ 的展开式中 $x_1^{n_1}x_2^{n_2}\cdots x_k^{n_k}$ 项的系数，也称作多项式系数. 关于它的性质在后面还会进一步加以讨论.

定理 1.2.3

当 $r\leqslant n_i, i=1,2,\cdots,k,$ 时，多重集 S 的 r 组合数是 $C(k+r-1,r)$.

多重集 $S=\{n_1\cdot a_1,n_2\cdot a_2,\cdots,n_k\cdot a_k\}$ 的全排列数也记作 $\binom{n}{n_1\,n_2\cdots n_k}$，这个数恰好是多项式 $(x_1+x_2+\cdots+x_k)^n$ 的展开式中 $x_1^{n_1}x_2^{n_2}\cdots x_k^{n_k}$ 项的系数，也称作*多项式系数*．关于它的性质在后面还会进一步加以讨论．

定理 1.2.3

当 $r\leqslant n_i, i=1,2,\cdots,k,$ 时，多重集 S 的 r 组合数是 $C(k+r-1,r)$.

证明.

可使用一一对应的思想来证明这个定理. S 的一个 r 组合为 S 的一个子多重集 $\{x_1\cdot a_1,x_2\cdot a_2,\cdots,x_k\cdot a_k\}$，其中 $x_1+x_2+\cdots+x_k=r$, x_i 为非负整数, $i=1,2,\cdots,k$. 这个方程称作*不定方程*，可在它的非负整数解 $x_1,x_2,\cdots,x_k$ 和 r 个 1、$k-1$ 个 0 的排列之间建立一一对应. 对于解 $x_1,x_2,\cdots,x_k$，排列具有下述形式：

$$\underbrace{1\quad 1\quad\cdots\quad 1}_{x_1\text{ 个 }1}\quad 0\quad\underbrace{1\quad 1\quad\cdots\quad 1}_{x_2\text{ 个 }1}\quad 0\quad\cdots\quad 0\quad\underbrace{1\quad 1\quad\cdots\quad 1}_{x_k\text{ 个 }1}$$

其中 $k-1$ 个 0 将 r 个 1 分成 k 段，每段含有 1 的个数分别为 $x_1,x_2,\cdots,x_k$. 不难看出，这个排列是多重集 $\{r\cdot 1,(k-1)\cdot 0\}$ 的全排列，根据定理 1.2.2，这样的排列有 $\frac{(r+k-1)!}{r!(k-1)!}=C(k+r-1,r)$ 个，因此 S 的 r 组合数是 $C(k+r-1,r)$.

例 1.2.1

排列 26 个字母，使得 a 与 b 之间恰有 7 个字母，求排列的方法数.

例 1.2.1

排列 *26* 个字母，使得 a 与 b 之间恰有 *7* 个字母，求排列的方法数.

解

采用分步处理的方法. 先固定 a 和 b，中间插入 7 个字母，构成一个结构，有 $2P(24,7)$ 种方法. 将这个结构看作一个大字母再与其余 17 个字母进行全排列，有 $18!$ 种排列的方法. 根据乘法法则，$N = 2P(24,7)\cdot 18!$. □

例 1.2.1

排列 *26* 个字母,使得 a 与 b 之间恰有 *7* 个字母,求排列的方法数.

解

采用分步处理的方法. 先固定 a 和 b,中间插入 7 个字母,构成一个结构,有 $2P(24,7)$ 种方法. 将这个结构看作一个大字母再与其余 17 个字母进行全排列,有 $18!$ 种排列的方法. 根据乘法法则,$N = 2P(24,7) \cdot 18!$. □

例 1.2.2

把 $2n$ 个人分成 n 组,每组 *2* 人,求不同的分法数 N.

例 1.2.1

排列 26 个字母,使得 a 与 b 之间恰有 7 个字母,求排列的方法数.

解

采用分步处理的方法. 先固定 a 和 b,中间插入 7 个字母,构成一个结构,有 $2P(24,7)$ 种方法. 将这个结构看作一个大字母再与其余 17 个字母进行全排列,有 $18!$ 种排列的方法. 根据乘法法则,$N = 2P(24,7)\cdot 18!$. □

例 1.2.2

把 $2n$ 个人分成 n 组,每组 2 人,求不同的分法数 N.

解

因为这个分组是无序的,而且每组人数还有限制,没有直接的计数模型可以使用. 我们的思路是:先计数有序分组的方法,然后除以 $n!$,就得到无序分组的方法数.

$$
\begin{aligned}
N &= \frac{1}{n!} C(2n,2)C(2n-2,2)\cdots C(2,2) \\
&= \frac{1}{n!}\frac{(2n)!}{(2n-2)!2}\frac{(2n-2)!}{(2n-4)!2}\cdots\frac{2!}{0!2} = \frac{(2n)!}{2^n n!}.
\end{aligned}
$$

例 1.2.3

下面给出一段简单的程序，问：它的输出 x 是什么？

Algorithm

1. $x \leftarrow 0$

2. *for* $i_1 \leftarrow 1$ *to* n

3. *for* $i_2 \leftarrow 1$ *to* i_1

$\cdots$

$k+1.$ *for* $i_k \leftarrow 1$ *to* i_{k-1}

$k+2.$ $x \leftarrow x+1$

$k+3.$ *return* x

例 1.2.3

下面给出一段简单的程序，问：它的输出 x 是什么？

Algorithm

1. $x \leftarrow 0$

2. *for* $i_1 \leftarrow 1$ *to* n

3. *for* $i_2 \leftarrow 1$ *to* i_1

$\cdots$

$k+1.$ *for* $i_k \leftarrow 1$ *to* i_{k-1}

$k+2.$ $x \leftarrow x+1$

$k+3.$ *return* x

解

程序中包含一个 k 重的嵌套循环，对于给定的 $i_1, i_2, \cdots, i_k$，其中 $1 \leqslant i_k \leqslant i_{k-1} \leqslant \cdots \leqslant i_1 \leqslant n$，循环体运行 1 次，就对 x 加 1. x 的初值是 0，于是 x 的输出就是循环体的执行次数. 这恰好对应了整数序列 $i_1, i_2, \cdots, i_k$ 可能的取值个数，即多重集 $S = \{+\infty \cdot 1, +\infty \cdot 2, \cdots, +\infty \cdot n\}$ 的 k 组合数. 一旦从 S 中选定 k 个整数，按照从大到小的顺序排列(这里允许两个数相等)，就唯一确定了一组 $i_1, i_2, \cdots, i_k$ 的值. 根据定理 1.2.3，$x = C(n+k-1, k)$.

例 1.2.4

从 $S = \{1, 2, \cdots, n\}$ 中选择 k 个不相邻的数，有多少种方法？

例 1.2.4

从 $S=\{1,2,\cdots,n\}$ 中选择 k 个不相邻的数，有多少种方法？

解

设 $a_1,a_2,\cdots,a_k$ 是选出的 k 个不相邻的数，由这 k 个数对应生成另外的 k 个数 $b_1,b_2,\cdots,b_k$. 产生规则是 $b_i=a_i-(i-1)$. 对于两组不同的数 $a_1,a_2,\cdots,a_k$ 与 $a'_1,a'_2,\cdots,a'_k$，生成的两组数 $b_1,b_2,\cdots,b_k$ 与 $b'_1,b'_2,\cdots,b'_k$ 也不相同. 反之，如果生成的两组数不相同，那么原来的两组数也不相同. 它们之间存在一一对应关系. 只需计数生成的序列 $b_1,b_2,\cdots,b_k$ 有多少个，就可以得到原来问题的解. 由于所有的 b_i 允许相邻，且 b_k 至多是 $n-(k-1)$，因此这些序列的个数就是从 $\{1,2,\cdots,n-(k-1)\}$ 中无序选取 k 个元素的方法数，从而得到问题的解是 $C(n-(k-1),k)=C(n-k+1,k)$. □

例 1.2.4

从 $S=\{1,2,\cdots,n\}$ 中选择 k 个不相邻的数，有多少种方法？

解

设 $a_1,a_2,\cdots,a_k$ 是选出的 k 个不相邻的数，由这 k 个数对应生成另外的 k 个数 $b_1,b_2,\cdots,b_k$. 产生规则是 $b_i=a_i-(i-1)$. 对于两组不同的数 $a_1,a_2,\cdots,a_k$ 与 $a_1',a_2',\cdots,a_k'$，生成的两组数 $b_1,b_2,\cdots,b_k$ 与 $b_1',b_2',\cdots,b_k'$ 也不相同. 反之，如果生成的两组数不相同，那么原来的两组数也不相同. 它们之间存在一一对应关系. 只需计数生成的序列 $b_1,b_2,\cdots,b_k$ 有多少个，就可以得到原来问题的解. 由于所有的 b_i 允许相邻，且 b_k 至多是 $n-(k-1)$，因此这些序列的个数就是从 $\{1,2,\cdots,n-(k-1)\}$ 中无序选取 k 个元素的方法数，从而得到问题的解是 $C(n-(k-1),k)=C(n-k+1,k)$. □

例 1.2.3 与 1.2.4 使用的是一一对应的求解技巧. 有许多典型的组合计数问题，已经得到相应的公式或者求解的方法. 换句话说，已经建立了相应的组合计数模型. 当遇到其他组合计数问题，如果可以与这些典型的计数模型一一对应，那么就可以直接应用有关的结果来求解. 这是一种非常有用的方法.

10.3 二项式定理与组合恒等式

定理 1.3.1

设 n 是正整数，对一切实数 x 和 y，有 $(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}$.

10.3 二项式定理与组合恒等式

定理 1.3.1

设 n 是正整数,对一切实数 x 和 y,有 $(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}$.

二项式 $(x+y)^n$ 的展开式中每项的系数都是组合数,因此组合数也称作二项式系数. 可以使用数学归纳法证明二项式定理,这里使用组合分析的方法加以证明.

10.3 二项式定理与组合恒等式

定理 1.3.1

设 n 是正整数,对一切实数 x 和 y,有 $(x+y)^n=\sum_{k=0}^{n}\binom{n}{k}x^k y^{n-k}$.

二项式 $(x+y)^n$ 的展开式中每项的系数都是组合数,因此组合数也称作二项式系数. 可以使用数学归纳法证明二项式定理,这里使用组合分析的方法加以证明.

证明.

当乘积被展开时其中的项都是下述形式:$x^i y^{n-i}, i=0,1,2,\cdots,n$. 而构成形如 $x^i y^{n-i}$ 的项,必须从 n 个二项式 $(x+y)$ 中选 i 个提供 x,其他的 $n-i$ 个提供 y. 因此,$x^i y^{n-i}$ 的系数是 $\binom{n}{i}$,定理得证. □

10.3 二项式定理与组合恒等式

定理 1.3.1

设 n 是正整数，对一切实数 x 和 y，有 $(x+y)^n=\sum_{k=0}^{n}\binom{n}{k}x^k y^{n-k}$.

二项式 $(x+y)^n$ 的展开式中每项的系数都是组合数，因此组合数也称作二项式系数. 可以使用数学归纳法证明二项式定理，这里使用组合分析的方法加以证明.

证明.

当乘积被展开时其中的项都是下述形式：$x^i y^{n-i}, i=0,1,2,\cdots,n$. 而构成形如 $x^i y^{n-i}$ 的项，必须从 n 个二项式 $(x+y)$ 中选 i 个提供 x，其他的 $n-i$ 个提供 y. 因此，$x^i y^{n-i}$ 的系数是 $\binom{n}{i}$，定理得证. □

在二项式定理中令 $y=1$ 可以得到以下推论.

推论 1.3.1

设 n 是正整数，则 $(1+x)^n=\sum_{k=0}^{n}\binom{n}{k}x^k$.

组合恒等式

下面给出有关二项式系数的一些主要的恒等式，这些恒等式也称作组合恒等式.

命题 1.3.1

$$(1)\quad \binom{n}{k} = \binom{n}{n-k},\quad n, k \in \mathbb{N}, k \le n. \tag{1.3.1}$$

$$(2)\quad \binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1},\quad n, k \in \mathbb{Z}^+, k \le n. \tag{1.3.2}$$

$$(3)\quad \binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1},\quad n, k \in \mathbb{Z}^+, k \le n. \tag{1.3.3}$$

组合恒等式

下面给出有关二项式系数的一些主要的恒等式，这些恒等式也称作组合恒等式.

命题 1.3.1

(1) $$\binom{n}{k} = \binom{n}{n-k}, \quad n, k \in \mathbb{N}, k \leq n. \tag{1.3.1}$$

(2) $$\binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}, \quad n, k \in \mathbb{Z}^+, k \leq n. \tag{1.3.2}$$

(3) $$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}, \quad n, k \in \mathbb{Z}^+, k \leq n. \tag{1.3.3}$$

以上公式的证明已经在前一节给出过.

这些递推式在计算组合数的序列和或恒等式证明中经常用到，主要用于组合数的化简或者变形.

命题 1.3.2

(1) $$\sum_{k=0}^{n}\binom{n}{k}=2^n,\ n\in\mathbb{N}. \tag{1.3.4}$$

(2) $$\sum_{k=0}^{n}(-1)^k\binom{n}{k}=0,\ n\in\mathbb{N}. \tag{1.3.5}$$

命题 1.3.2

(1) $$\sum_{k=0}^{n}\binom{n}{k}=2^n,\ n\in\mathbb{N}. \tag{1.3.4}$$

(2) $$\sum_{k=0}^{n}(-1)^k\binom{n}{k}=0,\ n\in\mathbb{N}. \tag{1.3.5}$$

上述公式的组合数 $\binom{n}{k}$ 中的 n 不变，而 k 是随项的标号改变的，简称为变下项的求和公式. 这些公式的证明主要使用二项式定理或者组合分析方法. 这里只证明式 (1.3.4).

命题 1.3.2

(1) $$\sum_{k=0}^{n}\binom{n}{k}=2^n,\ n\in\mathbb{N}. \tag{1.3.4}$$

(2) $$\sum_{k=0}^{n}(-1)^k\binom{n}{k}=0,\ n\in\mathbb{N}. \tag{1.3.5}$$

上述公式的组合数 $\binom{n}{k}$ 中的 n 不变，而 k 是随项的标号改变的，简称为变下项的求和公式. 这些公式的证明主要使用二项式定理或者组合分析方法. 这里只证明式 (1.3.4).

证明.

方法一. 在二项式定理中令 $x=y=1$ 即可.

方法二. 组合分析法. 设 $S=\{1,2,\cdots,n\}$，下面计数 S 的所有子集. 一种方法就是分类处理，将所有的子集按照含有元素的多少进行分类. n 元集合的 k 子集个数是 $\binom{n}{k}$，根据加法法则，子集总数是 $\sum_{k=0}^{n}\binom{n}{k}$. 另一种方法是分步处理，为构成 S 的子集 A，依次考虑元素 $1,2,\cdots,n$ 是否加入 A. 每个元素有 2 种选择，根据乘法法则，子集总数是 2^n. □

命题 1.3.3

$$(1)\quad \sum_{l=0}^{n}\binom{l}{k}=\binom{n+1}{k+1},\ n,k\in\mathbb{N}. \tag{1.3.6}$$

$$(2)\quad \binom{n}{r}\binom{r}{k}=\binom{n}{k}\binom{n-k}{r-k},\ k\le r\le n,k,r,n\in\mathbb{N}. \tag{1.3.7}$$

命题 1.3.3

$$(1)\quad \sum_{l=0}^{n}\binom{l}{k}=\binom{n+1}{k+1},\ n,k\in\mathbb{N}. \tag{1.3.6}$$

$$(2)\quad \binom{n}{r}\binom{r}{k}=\binom{n}{k}\binom{n-k}{r-k},\ k\le r\le n,k,r,n\in\mathbb{N}. \tag{1.3.7}$$

证明.

先证明式 (1.3.6), 使用组合分析的方法. 令 $S=\{a_1,a_2,\cdots,a_{n+1}\}$ 为 $n+1$ 元集合. 等式右边是 S 的 $k+1$ 子集数. 考虑另一种分类计数的方法. 将所有的 $k+1$ 子集分成如下 $n+1$ 类.

命题 1.3.3

$$(1)\quad \sum_{l=0}^{n}\binom{l}{k}=\binom{n+1}{k+1},\ n,k\in\mathbb{N}. \tag{1.3.6}$$

$$(2)\quad \binom{n}{r}\binom{r}{k}=\binom{n}{k}\binom{n-k}{r-k},\ k\le r\le n,k,r,n\in\mathbb{N}. \tag{1.3.7}$$

证明.

先证明式 (1.3.6),使用组合分析的方法. 令 $S=\{a_1,a_2,\cdots,a_{n+1}\}$ 为 $n+1$ 元集合. 等式右边是 S 的 $k+1$ 子集数. 考虑另一种分类计数的方法. 将所有的 $k+1$ 子集分成如下 $n+1$ 类.

第 1 类:含 a_1,剩下的 k 个元素取自 $\{a_2,\cdots,a_{k+1}\}$,有 $\binom{n}{k}$ 种取法;

命题 1.3.3

$$(1)\quad \sum_{l=0}^{n}\binom{l}{k}=\binom{n+1}{k+1},\ n,k\in\mathbb{N}. \tag{1.3.6}$$

$$(2)\quad \binom{n}{r}\binom{r}{k}=\binom{n}{k}\binom{n-k}{r-k},\ k\le r\le n, k,r,n\in\mathbb{N}. \tag{1.3.7}$$

证明.

先证明式 (1.3.6),使用组合分析的方法. 令 $S=\{a_1,a_2,\cdots,a_{n+1}\}$ 为 $n+1$ 元集合. 等式右边是 S 的 $k+1$ 子集数. 考虑另一种分类计数的方法. 将所有的 $k+1$ 子集分成如下 $n+1$ 类.

第 1 类:含 a_1,剩下的 k 个元素取自 $\{a_2,\cdots,a_{k+1}\}$,有 $\binom{n}{k}$ 种取法;

第 2 类:不含 a_1,含 a_2,剩下的 k 个元素取自 $\{a_3,\cdots,a_{n+1}\}$,有 $\binom{n-1}{k}$ 种方法;

命题 1.3.3

$$(1)\quad \sum_{l=0}^{n}\binom{l}{k}=\binom{n+1}{k+1},\ n,k\in\mathbb{N}. \tag{1.3.6}$$

$$(2)\quad \binom{n}{r}\binom{r}{k}=\binom{n}{k}\binom{n-k}{r-k},\ k\le r\le n, k,r,n\in\mathbb{N}. \tag{1.3.7}$$

证明.

先证明式 (1.3.6),使用组合分析的方法. 令 $S=\{a_1,a_2,\cdots,a_{n+1}\}$ 为 $n+1$ 元集合. 等式右边是 S 的 $k+1$ 子集数. 考虑另一种分类计数的方法. 将所有的 $k+1$ 子集分成如下 $n+1$ 类.

第 1 类:含 a_1,剩下的 k 个元素取自 $\{a_2,\cdots,a_{k+1}\}$,有 $\binom{n}{k}$ 种取法;

第 2 类:不含 a_1,含 a_2,剩下的 k 个元素取自 $\{a_3,\cdots,a_{n+1}\}$,有 $\binom{n-1}{k}$ 种方法;

$\cdots$

第 $n+1$ 类:不含 $a_1,a_2,\cdots,a_n$,含 a_{n+1},剩下的 k 个元素取自空集,有 $\binom{0}{k}$ 种方法.

命题 1.3.3

$$(1)\quad \sum_{l=0}^{n}\binom{l}{k}=\binom{n+1}{k+1},\ n,k\in\mathbb{N}. \tag{1.3.6}$$

$$(2)\quad \binom{n}{r}\binom{r}{k}=\binom{n}{k}\binom{n-k}{r-k},\ k\le r\le n,k,r,n\in\mathbb{N}. \tag{1.3.7}$$

证明.

先证明式 (1.3.6),使用组合分析的方法. 令 $S=\{a_1,a_2,\cdots,a_{n+1}\}$ 为 $n+1$ 元集合. 等式右边是 S 的 $k+1$ 子集数. 考虑另一种分类计数的方法. 将所有的 $k+1$ 子集分成如下 $n+1$ 类.

第 1 类:含 a_1,剩下的 k 个元素取自 $\{a_2,\cdots,a_{k+1}\}$,有 $\binom{n}{k}$ 种取法;

第 2 类:不含 a_1,含 a_2,剩下的 k 个元素取自 $\{a_3,\cdots,a_{n+1}\}$,有 $\binom{n-1}{k}$ 种方法;

$\cdots$

第 $n+1$ 类:不含 $a_1,a_2,\cdots,a_n$,含 a_{n+1},剩下的 k 个元素取自空集,有 $\binom{0}{k}$ 种方法.

根据加法法则,等式左边也是 S 的 $k+1$ 子集个数.

命题1.3.3证明(续)

下证式 (1.3.7). 公式左边计数了先从 n 元集 S 中选取 r 个元素，然后在这 r 个元素中再选 k 个元素的方法. 公式右边的 $\binom{n}{k}$ 是从 S 中直接选取 k 子集的方法数. 显然前一种方法选择的同一个 k 子集会重复出现.

命题1.3.3证明(续)

下证式 (1.3.7). 公式左边计数了先从 n 元集 S 中选取 r 个元素，然后在这 r 个元素中再选 k 个元素的方法. 公式右边的 $\binom{n}{k}$ 是从 S 中直接选取 k 子集的方法数. 显然前一种方法选择的同一个 k 子集会重复出现.

例如，从集合 $\{a, b, c, d, e\}$ 中先选 4 子集，然后从这些 4 子集再选 3 子集. 那么 3 子集 $\{b, c, d\}$ 可能被选出 2 次，一次是从 4 子集 $\{a, b, c, d\}$ 中选出的，另一次是从 4 子集 $\{b, c, d, e\}$ 中选出的.

命题1.3.3证明(续)

下证式 (1.3.7). 公式左边计数了先从 n 元集 S 中选取 r 个元素, 然后在这 r 个元素中再选 k 个元素的方法. 公式右边的 $\binom{n}{k}$ 是从 S 中直接选取 k 子集的方法数. 显然前一种方法选择的同一个 k 子集会重复出现.

例如, 从集合 $\{a, b, c, d, e\}$ 中先选 4 子集, 然后从这些 4 子集再选 3 子集. 那么 3 子集 $\{b, c, d\}$ 可能被选出 2 次, 一次是从 4 子集 $\{a, b, c, d\}$ 中选出的, 另一次是从 4 子集 $\{b, c, d, e\}$ 中选出的.

下面计算采用第一种方法时同一个 k 子集重复出现的次数. 换句话说, 就是计算有多少个 r 子集能够选出相同的 k 子集. 设 k 子集为 A, 一个 r 子集中除了 A 的元素外, 剩下的 $r-k$ 个元素取自 $S-A$. 因此, 有 $\binom{n-k}{r-k}$ 个 r 子集能生成相同的 k 子集. 这就证明了等式左边的值恰好是 $\binom{n}{k}$ 的 $\binom{n-k}{r-k}$ 倍.

命题1.3.3证明(续)

下证式 (1.3.7). 公式左边计数了先从 n 元集 S 中选取 r 个元素,然后在这 r 个元素中再选 k 个元素的方法. 公式右边的 $\binom{n}{k}$ 是从 S 中直接选取 k 子集的方法数. 显然前一种方法选择的同一个 k 子集会重复出现.

例如,从集合 $\{a, b, c, d, e\}$ 中先选 4 子集,然后从这些 4 子集再选 3 子集. 那么 3 子集 $\{b, c, d\}$ 可能被选出 2 次,一次是从 4 子集 $\{a, b, c, d\}$ 中选出的,另一次是从 4 子集 $\{b, c, d, e\}$ 中选出的.

下面计算采用第一种方法时同一个 k 子集重复出现的次数. 换句话说,就是计算有多少个 r 子集能够选出相同的 k 子集. 设 k 子集为 A,一个 r 子集中除了 A 的元素外,剩下的 $r-k$ 个元素取自 $S-A$. 因此,有 $\binom{n-k}{r-k}$ 个 r 子集能生成相同的 k 子集. 这就证明了等式左边的值恰好是 $\binom{n}{k}$ 的 $\binom{n-k}{r-k}$ 倍.

注意在式 (1.3.6) 中,当 $l < k$ 时等式左边的项都等于 0. 这个公式的组合数 $\binom{l}{k}$ 中的下项 k 不变,而上项 l 随项的序号改变,是变上项的求和式. 这种求和式主要用于有关组合数序列的求和或者证明组合恒等式.

命题 1.3.4

(1) $$\sum_{k=0}^{r}\binom{m}{k}\binom{n}{r-k}=\binom{m+n}{r},\ m,n,r\in\mathbb{N}, r\leqslant\min(m,n). \tag{1.3.8}$$

(2) $$\sum_{k=0}^{n}\binom{m}{k}\binom{n}{k}=\binom{m+n}{m},\ m,n\in\mathbb{N}. \tag{1.3.9}$$

命题 1.3.4

(1) $$\sum_{k=0}^{r}\binom{m}{k}\binom{n}{r-k}=\binom{m+n}{r},\quad m,n,r\in\mathbb{N}, r\leqslant\min(m,n). \tag{1.3.8}$$

(2) $$\sum_{k=0}^{n}\binom{m}{k}\binom{n}{k}=\binom{m+n}{m},\quad m,n\in\mathbb{N}. \tag{1.3.9}$$

证明.

如果在式 (1.3.8) 中用 n 替换 r 就可得到式 (1.3.9). 因此只需要证明式 (1.3.8). 考虑集合 $A=\{a_1,a_2,\cdots,a_m\}, B=\{b_1,b_2,\cdots,b_n\}$. 等式右边计数了从这两个集合中选出 r 个元素的方法. 将这些选法按照含有 A 中元素的个数 k 进行分类, $k=0,1,\cdots,r$. 考虑含有 A 中 k 个元素的选法数. 先确定 A 中的 k 个元素, 有 $\binom{m}{k}$ 种方式, 接着确定 B 中的 $r-k$ 个元素, 有 $\binom{n}{r-k}$ 种方法. 由乘法法则, 恰含 k 个 A 中元素的方法有 $\binom{m}{k}\binom{n}{r-k}$ 种, 根据加法法则对 k 求和, 定理得证. □

例 1.3.1

证明以下组合恒等式.

1. $\sum_{k=0}^{n} k\binom{n}{k} = n2^{n-1}, \ n \in \mathbb{Z}^+$.
2. $\sum_{k=0}^{n} k^2\binom{n}{k} = n(n+1)2^{n-2}, \ n \in \mathbb{Z}^+$.

例 1.3.1

证明以下组合恒等式.

1. $\sum_{k=0}^{n} k\binom{n}{k} = n2^{n-1}$, $n \in \mathbb{Z}^+$.
2. $\sum_{k=0}^{n} k^2\binom{n}{k} = n(n+1)2^{n-2}$, $n \in \mathbb{Z}^+$.

证明.

(1) 由二项式定理有 $(1+x)^n = \sum_{k=0}^{n} \binom{n}{k} x^k$. 两边求导得 $n(1+x)^{n-1} = \sum_{k=1}^{n} \binom{n}{k} kx^{k-1}$. 在上面的公式中令 $x = 1$ 即可.

例 1.3.1

证明以下组合恒等式.

1. $\sum_{k=0}^{n} k\binom{n}{k} = n2^{n-1},\ n \in \mathbb{Z}^+$.
2. $\sum_{k=0}^{n} k^2\binom{n}{k} = n(n+1)2^{n-2},\ n \in \mathbb{Z}^+$.

证明.

(1) 由二项式定理有 $(1+x)^n = \sum_{k=0}^{n}\binom{n}{k}x^k$. 两边求导得 $n(1+x)^{n-1} = \sum_{k=1}^{n}\binom{n}{k}kx^{k-1}$. 在上面的公式中令 $x=1$ 即可.

(2) $\sum_{k=0}^{n} k^2\binom{n}{k} = \sum_{k=1}^{n} k^2\frac{n}{k}\binom{n-1}{k-1} = \sum_{k=1}^{n} kn\binom{n-1}{k-1}$

$= \mathrm{n}\sum_{k=1}^{n}[(k-1)+1]\binom{n-1}{k-1}$

$= \mathrm{n}\sum_{k=1}^{n}(k-1)\binom{n-1}{k-1} + n\sum_{k=1}^{n}\binom{n-1}{k-1}$

$= \mathrm{n}\sum_{k=0}^{n-1} k\binom{n-1}{k} + n2^{n-1} = n(n+1)2^{n-2}$. (利用 (1) 的结果)

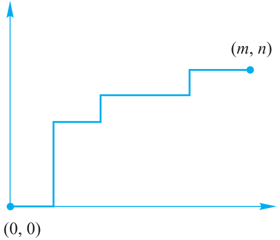
组合恒等式的证明方法

除了上面命题中给出的这些经常使用的组合恒等式，还有许多组合恒等式没有列出来.
总结一下有关组合恒等式的证明方法，大致有以下几种.

1. 已知恒等式代入并化简；
2. 使用二项式定理比较相同项的系数，或者进行级数的求导或者积分；
3. 数学归纳法；
4. 组合分析方法.

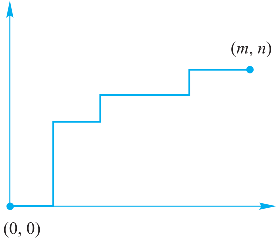
非降路径问题

考察下图. 设 m, n 是正整数, 从 $(0,0)$ 点到 (m,n) 点的非降路径是一条折线, 这条折线由 $m+n$ 次移动构成, 每次允许向上或者向右移动一步. 问: 不同的非降路径有多少条?



非降路径问题

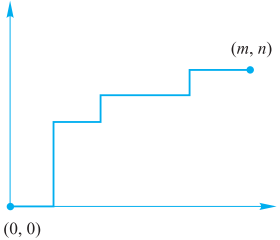
考察下图. 设 m, n 是正整数, 从 $(0,0)$ 点到 (m,n) 点的非降路径是一条折线, 这条折线由 $m+n$ 次移动构成, 每次允许向上或者向右移动一步. 问: 不同的非降路径有多少条?



不同的路径取决于 $m+n$ 步的选择, 其中包含 m 步向右, n 步向上. 这种路径条数等于从 $m+n$ 个位置中选 m 个位置的方法数, 即 $\binom{m+n}{m}$ 或 $\binom{m+n}{n}$.

非降路径问题

考察下图. 设 m, n 是正整数，从 $(0,0)$ 点到 (m,n) 点的非降路径是一条折线，这条折线由 $m+n$ 次移动构成，每次允许向上或者向右移动一步. 问：不同的非降路径有多少条?



不同的路径取决于 $m+n$ 步的选择，其中包含 m 步向右，n 步向上. 这种路径条数等于从 $m+n$ 个位置中选 m 个位置的方法数，即 $\binom{m+n}{m}$ 或 $\binom{m+n}{n}$.

下面考虑这个问题的其他情况.

给定非负整数 a, b, m, n，$a \leqslant m, b \leqslant n$. 从 (a,b) 点到 (m,n) 点的非降路径数等于从 $(0,0)$ 点到 $(m-a, n-b)$ 点的非降路径数，这相当于坐标平移，或者说两类路径之间一一对应. 因此，这种路径条数等于 $\binom{m-a+n-b}{m-a}$.

设 a, b, c, d, m, n 是非负整数，其中 $a \leqslant c \leqslant m, b \leqslant d \leqslant n$. 从 (a,b) 点经过 (c,d) 点到 (m,n) 点的非降路径数等于从 (a,b) 点到 (c,d) 点的非降路径数与从 (c,d) 点到 (m,n) 点的非降路径数之积. 这里采用了分步处理的思想.

例 1.3.2

求集合 $\{1, 2, \cdots, n\}$ 上的单调递增函数个数.

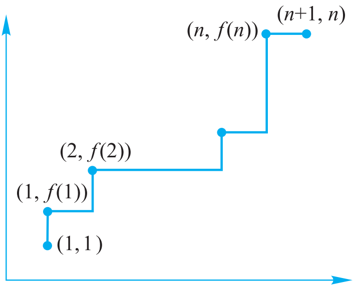


图 1.3.1

例 1.3.2

求集合 $\{1,2,\cdots,n\}$ 上的单调递增函数个数.

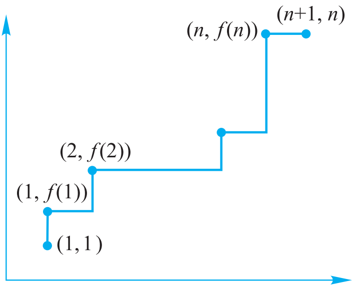


图 1.3.1

解

考虑单调递增函数 $f:\{1,2,\cdots,n\}\to\{1,2,\cdots,n\}$. 如左图所示,可以将自变量看作横坐标,对应的函数值看作纵坐标,得到 n 个点. 在图上增加 $(1,1)$ 和 $(n+1,n)$ 两个点,并按照下面的方法连接这 $n+2$ 个点:如果 $f(1)$ 不等于 1,那么从 $(1,1)$ 点开始向上连接到 $(1,f(1))$ 点. 从 $(1,f(1))$ 点先向右再向上连接到 $(2,f(2))$ 点,依照“先向右,后向上”的规则顺次连接 $(3,f(3))$,$\cdots$,直到 $(n+1,n)$ 点. 而这条连线恰好构成从 $(1,1)$ 点到 $(n+1,n)$ 点的一条非降路径. 显然这种非降路径与单调函数是一一对应的,只需计数非降路径条数就可以得到所求的单调函数个数. 根据公式,非降路径数是 $\binom{2n-1}{n}$. 因此,集合 $\{1,2,\cdots,n\}$ 上的单调递增函数个数也是 $\binom{2n-1}{n}$. □

可以将例1.3.2的结论推广. 设 $A=\{1,2,\cdots,m\}, B=\{1,2,\cdots,n\}$, 那么从 A 到 B 的单调函数(包括单调递增与单调递减函数)个数等于从 $(1,1)$ 点到 $(m+1,n)$ 点的非降路径数的两倍减去 n 个常值函数, 即 $2\binom{m+n-1}{m}-n$.

可以将例1.3.2的结论推广. 设 $A=\{1,2,\cdots,m\}, B=\{1,2,\cdots,n\}$,那么从 A 到 B 的单调函数(包括单调递增与单调递减函数)个数等于从 $(1,1)$ 点到 $(m+1,n)$ 点的非降路径数的两倍减去 n 个常值函数,即 $2\binom{m+n-1}{m}-n$.

例 1.3.3

下面考虑一个涉及栈输出的计数问题. 设有正整数 $1,2,\cdots,n$,从小到大排成一个队列. 将这些整数按照排列的次序依次压入一个栈(即后进先出栈). 当后面的整数进栈时,已经在栈中的整数可以在任何时刻输出. 问可能有多少种不同的输出序列. 例如,整数 $1,2,3$ 可能的输出序列有 $1,2,3$;对应的操作是:*1* 进栈,*1* 出栈,*2* 进栈,*2* 出栈,*3* 进栈,*3* 出栈. 也可能输出 $1,3,2$;对应的操作是:*1* 进栈,*1* 出栈,*2* 进栈,*3* 进栈,*3* 出栈,*2* 出栈.

可以将例1.3.2的结论推广. 设 $A=\{1,2,\cdots,m\}, B=\{1,2,\cdots,n\}$,那么从 A 到 B 的单调函数(包括单调递增与单调递减函数)个数等于从 $(1,1)$ 点到 $(m+1,n)$ 点的非降路径数的两倍减去 n 个常值函数,即 $2\binom{m+n-1}{m}-n$.

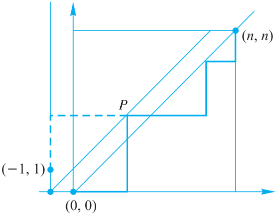
例 1.3.3

下面考虑一个涉及栈输出的计数问题. 设有正整数 $1,2,\cdots,n$,从小到大排成一个队列. 将这些整数按照排列的次序依次压入一个栈(即后进先出栈). 当后面的整数进栈时,已经在栈中的整数可以在任何时刻输出. 问可能有多少种不同的输出序列. 例如,整数 $1,2,3$ 可能的输出序列有 $1,2,3$;对应的操作是:*1* 进栈,*1* 出栈,*2* 进栈,*2* 出栈,*3* 进栈,*3* 出栈. 也可能输出 $1,3,2$;对应的操作是:*1* 进栈,*1* 出栈,*2* 进栈,*3* 进栈,*3* 出栈,*2* 出栈.

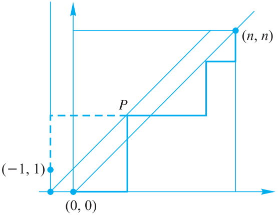
解

将进栈、出栈分别记作 x,y,一个输出对应了 n 个 x,n 个 y 的排列,且排列的任何前缀中的 x 个数不少于 y 的个数. 考虑非降路径的模型,从 $(0,0)$ 点出发,将排列中的 x 看作向右走一步,y 看作向上走一步,就可以得到一条从 $(0,0)$ 点到 (n,n) 点不穿过对角线的非降路径.

将所有从 $(0,0)$ 到 (n,n) 点的非降路径分成两类：穿过对角线的与不穿过对角线的. 只要求出了穿过对角线的条数 N_1，从总数中减去它即得所求路径条数.



将所有从 $(0,0)$ 到 (n,n) 点的非降路径分成两类：穿过对角线的与不穿过对角线的. 只要求出了穿过对角线的条数 N_1，从总数中减去它即得所求路径条数.



如上图所示，任何一条从 $(0,0)$ 点到 (n,n) 点穿过对角线的路径一定要接触直线 $y=x+1$，有可能接触多次，但最后会离开这条直线上的一点 P，沿直线 $y=x+1$ 下方的一条非降路径到达 (n,n) 点. 把这条路径的前半段，即 $(0,0)$ 点到 P 点的部分，以直线 $y=x+1$ 为轴进行翻转，生成一段新的从 $(-1,1)$ 点到 P 点的部分非降路径(左图中虚线表示的路径). 用这段新路径替换原来路径的前半段，就得到一条从 $(-1,1)$ 点到 (n,n) 点的非降路径. 容易看出，这种路径与从 $(0,0)$ 点到 (n,n) 点中间穿过对角线的非降路径之间存在一一对应. 因此，从 $(0,0)$ 点到 (n,n) 点穿过对角线的非降路径数 $N_1=\binom{2n}{n-1}$. 从 $(0,0)$ 点到 (n,n) 点的非降路径总数为 $\binom{2n}{n}$ 条，从而得到不同的输出序列个数是

$$\begin{aligned} N &= \binom{2n}{n} - \binom{2n}{n-1} \\ &= \frac{(2n)!}{n!n!} - \frac{(2n)!}{(n-1)!(n+1)!} = \frac{1}{n+1}\binom{2n}{n}. \end{aligned}$$

10.4 多项式定理

定理 1.4.1

设 n 为正整数, x_i 为实数, $i = 1, 2, \cdots, t$, 那么有

$$(x_1 + x_2 + \cdots + x_t)^n = \sum_{\substack{\text{满足 } n_1+\cdots+n_t=n \\ \text{的非负整数解}}} \binom{n}{n_1\ n_2\ \cdots\ n_t} x_1^{n_1} x_2^{n_2} \cdots x_t^{n_t},$$

这里 $\binom{n}{n_1\ n_2\ \cdots\ n_t} = \frac{n!}{n_1!n_2!\cdots n_t!}$, 称作 多项式系数.

10.4 多项式定理

定理 1.4.1

设 n 为正整数，x_i 为实数，$i = 1, 2, \cdots, t$，那么有

$$(x_1 + x_2 + \cdots + x_t)^n = \sum_{\substack{\text{满足 } n_1+\cdots+n_t=n \\ \text{的非负整数解}}} \binom{n}{n_1\ n_2\ \cdots\ n_t} x_1^{n_1} x_2^{n_2} \cdots x_t^{n_t},$$

这里 $\binom{n}{n_1\ n_2\ \cdots\ n_t} = \frac{n!}{n_1!n_2!\cdots n_t!}$，称作 多项式系数.

证明.

展开式中的项 $x_1^{n_1} x_2^{n_2} \cdots x_t^{n_t}$ 是如下构成的：在 n 个因式中选 n_1 个因式贡献 x_1，从剩下 $n - n_1$ 个因式中选 n_2 个因式贡献 x_2，$\cdots$，从剩下的 $n - n_1 - n_2 \cdots - n_{t-1}$ 个因式中选 n_t 个因式贡献 x_t. 根据乘法法则，这种项的个数是

$$\binom{n}{n_1}\binom{n-n_1}{n_2}\cdots\binom{n-n_1-\cdots-n_{t-1}}{n_t} = \frac{n!}{n_1!n_2!\cdots n_t!} = \binom{n}{n_1\,n_2\cdots n_t}.$$

10.4 多项式定理

定理 1.4.1

设 n 为正整数，x_i 为实数，$i = 1, 2, \cdots, t$，那么有

$$(x_1 + x_2 + \cdots + x_t)^n = \sum_{\substack{\text{满足 } n_1 + \cdots + n_t = n \\ \text{的非负整数解}}} \binom{n}{n_1 \ n_2 \ \cdots \ n_t} x_1^{n_1} x_2^{n_2} \cdots x_t^{n_t},$$

这里 $\binom{n}{n_1 \ n_2 \ \cdots \ n_t} = \frac{n!}{n_1!n_2!\cdots n_t!}$，称作 多项式系数.

证明.

展开式中的项 $x_1^{n_1} x_2^{n_2} \cdots x_t^{n_t}$ 是如下构成的：在 n 个因式中选 n_1 个因式贡献 x_1，从剩下 $n - n_1$ 个因式中选 n_2 个因式贡献 x_2，$\cdots$，从剩下的 $n - n_1 - n_2 \cdots - n_{t-1}$ 个因式中选 n_t 个因式贡献 x_t. 根据乘法法则，这种项的个数是

$$\binom{n}{n_1}\binom{n-n_1}{n_2}\cdots\binom{n-n_1-\cdots-n_{t-1}}{n_t} = \frac{n!}{n_1!n_2!\cdots n_t!} = \binom{n}{n_1 \, n_2 \cdots n_t}.$$

二项式定理是多项式定理的特例. 当 $t = 2$ 时，有 $\binom{n}{n_1 \ n_2} = \frac{(n_1+n_2)!}{n_1!n_2!} = C(n, n_1)$，因此，多项式定理就变成了二项式定理.

多项式定理的推论

推论 1.4.1

在多项式定理的展开式中,右边不同的项数为不定方程 $n_1 + n_2 + \cdots + n_t = n$ 的非负整数解的个数 $\binom{n+t-1}{n}$.

多项式定理的推论

推论 1.4.1

在多项式定理的展开式中，右边不同的项数为不定方程 $n_1+n_2+\cdots+n_t=n$ 的非负整数解的个数 $\binom{n+t-1}{n}$.

证明.

根据定理1.4.1，项 $x_1^{n_1}x_2^{n_2}\cdots x_t^{n_t}$ 中的指数和方程 $n_1+n_2+\cdots+n_t=n$ 的非负整数解之间存在一一对应. $\square$

多项式定理的推论

推论 1.4.1

在多项式定理的展开式中，右边不同的项数为不定方程 $n_1 + n_2 + \cdots + n_t = n$ 的非负整数解的个数 $\binom{n+t-1}{n}$.

证明.

根据定理1.4.1，项 $x_1^{n_1} x_2^{n_2} \cdots x_t^{n_t}$ 中的指数和方程 $n_1 + n_2 + \cdots + n_t = n$ 的非负整数解之间存在一一对应. $\square$

推论 1.4.2

$\sum \binom{n}{n_1\ n_2\ \cdots\ n_t} = t^n$，其中求和是对方程 $n_1 + n_2 + \cdots + n_t = n$ 的所有的非负整数解求和.

多项式定理的推论

推论 1.4.1

在多项式定理的展开式中，右边不同的项数为不定方程 $n_1 + n_2 + \cdots + n_t = n$ 的非负整数解的个数 $\binom{n+t-1}{n}$.

证明.

根据定理1.4.1，项 $x_1^{n_1} x_2^{n_2} \cdots x_t^{n_t}$ 中的指数和方程 $n_1 + n_2 + \cdots + n_t = n$ 的非负整数解之间存在一一对应. □

推论 1.4.2

$\sum \binom{n}{n_1\ n_2 \cdots n_t} = t^n$，其中求和是对方程 $n_1 + n_2 + \cdots + n_t = n$ 的所有的非负整数解求和.

证明.

在多项式公式中令 $x_i = 1, i = 1, 2, \cdots, t$. □

多项式系数

多项式系数 $\binom{n}{n_1\,n_2\cdots n_t}$ 经常在一些组合问题中出现，回顾第 10.2 节，它恰好是多重集 $S=\{n_1\cdot a_1, n_2\cdot a_2, \cdots, n_t\cdot a_t\}$ 的全排列数，同时它也对应了 n 个不同的球放到 t 个不同的盒子里，使得第 1 个盒子含有 n_1 个球，第 2 个盒子含有 n_2 个球，$\cdots$，第 t 个盒子含有 n_t 个球的方法数.

先从 n 个球中选出 n_1 个球放入第 1 个盒子，然后从剩下的 $n-n_1$ 个球中选出 n_2 个球放入第 2 个盒子，$\cdots$，最后从 $n-n_1-n_2-\cdots-n_{t-1}$ 个球中选出 n_t 个球放入第 t 个盒子. 根据乘法法则，放球的方法数恰为

$$\binom{n}{n_1}\binom{n-n_1}{n_2}\cdots\binom{n-n_1-\cdots-n_{t-1}}{n_t}=\frac{n!}{n_1!n_2!\cdots n_t!}=\binom{n}{n_1\ n_2\ \cdots\ n_t}.$$

多项式系数

下面使用多项式系数的性质给出费马小定理的另一个证明.

例 1.4.1

证明:设 p 是素数, n 与 p 互素, 则 $n^{p-1} \equiv 1 \pmod p$.

多项式系数

下面使用多项式系数的性质给出费马小定理的另一个证明.

例 1.4.1

证明:设 p 是素数,n 与 p 互素,则 $n^{p-1} \equiv 1 \pmod p$.

引理 1.4.1

若 $\binom{p}{k_1\, k_2 \cdots k_n} \neq 1$,则 $p \mid \binom{p}{k_1\, k_2 \cdots k_n}$.

多项式系数

下面使用多项式系数的性质给出费马小定理的另一个证明.

例 1.4.1

证明:设 p 是素数,n 与 p 互素,则 $n^{p-1} \equiv 1 \pmod p$.

引理 1.4.1

若 $\binom{p}{k_1\,k_2\cdots k_n} \neq 1$,则 $p \mid \binom{p}{k_1\,k_2\cdots k_n}$.

证明.

由于 p 是素数,根据全排列数的定义显然有

$$\binom{p}{k_1\,k_2\cdots k_n} = 1 \Leftrightarrow \text{存在某个 } j \text{ 使得 } k_j = p\text{,其他 } k_i = 0, i \neq j.$$

于是由 $\binom{p}{k_1\,k_2\cdots k_n} \neq 1$ 可知 $k_1!k_2!\cdots k_n!$ 中不含 p. 由于 $\binom{p}{k_1\,k_2\cdots k_n} = \frac{p!}{k_1!k_2!\cdots k_n!}$ 是整数,且 $k_1!k_2!\cdots k_n!$ 中不含 p,因此 $k_1!k_2!\cdots k_n!$ 整除 $(p-1)!$,从而证明了 $p \mid \binom{p}{k_1\,k_2\cdots k_n}$. □

费马小定理的证明

设 p 是素数，n 与 p 互素，则 $n^{p-1} \equiv 1 \pmod{p}$.

费马小定理的证明

设 p 是素数，n 与 p 互素，则 $n^{p-1} \equiv 1 \pmod p$.

证明.

根据多项式定理有

$$(x_1 + x_2 + \cdots + x_n)^p = \sum_{\sum k_i = p} \binom{p}{k_1\, k_2 \cdots k_n} x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}.$$

令所有的 $x_i = 1$ 得到

$$n^p = \sum_{\sum k_i = p} \binom{p}{k_1\, k_2 \cdots k_n}.$$

根据引理，当 $\binom{p}{k_1\, k_2 \cdots k_n} \neq 1$ 时，有 $p \mid \binom{p}{k_1\, k_2 \cdots k_n}$. 右边恰有 n 项的值等于 1，其余各项之和为 $n^p - n$. 因为 p 整除其余的每一项，故 $p \mid (n^p - n)$. 又因为 p 与 n 互素，所以有 $n^{p-1} \equiv 1 \pmod p$. $\square$