第5章 组合计数

组合计数在许多学科中都会用到，特别是在计算机的算法设计与分析中用于估计算法的复杂度函数。本章介绍了加法原理、排列与组合，并详细介绍了计数的两个典型应用——鸽笼原理和递推关系。

递归关系是序列中第 n 个元素与它前若干个元素之间的关系，可用于解决一些特定的计数问题。由于递归关系和递归算法密切相关，因此递归关系可以很自然地用于递归算法分析。

5.1 基本原理

5.1.1 加法原理

加法原理是一个初等原理。它是全体等于它的各部分之和这一原理的公式化。

定理 5.1(加法原理) 假定 $S_1, S_2, \cdots, S_t$ 均为集合，第 i 个集合 S_i 有 n_i 个元素。若 $\{S_1, S_2, \cdots, S_t\}$ 为两两不交的集合(若 $i \neq j$，$S_i \cap S_j = \varnothing$)，则可以从 $S_1, S_2, \cdots, S_t$ 选择出的元素总数为 $n_1 + n_2 + \cdots + n_t$(即集合 $S_1 \cup S_2 \cup \cdots \cup S_t$ 含有 $n_1 + n_2 + \cdots + n_t$ 个元素)。

加法原理可总结为：当计数的元素可分解为若干个不相交的子集时，可将每个子集元素的个数相加得到元素的总数。

例 5.1 在 1，2，…，10 中，有 5 个偶数，4 个素数，求 1，2，…，10 中是偶数或是素数的个数。

解：由于 2 既是偶数，又是素数，所以，1，2，…，10 中，是偶数或是素数的个数不是简单地 $5+4=9$，而是 8。此例说明，不能不加分析地、简单地使用加法原理。

例 5.2 当执行完以下代码后，k 的值是多少？

```
k:=0
for i₁:=1 to n₁
```

```
        k:=k+1
for i₂:=1 to n₂
        k:=k+1
          ⋮
for iₘ:=1 to nₘ
        k:=k+1
```

解：k 的初始值是 0，在执行完

```
for i₁:=1 to n₁
        k:=k+1
```

后，$k=n_1$。实际上，这段程序是让 k 从 0 起，不断地加 1，共 n_1 次。同理，在执行完

```
for i₂:=1 to n₂
        k:=k+1
```

后，$k=n_1+n_2$。执行完最后的

```
for iₘ:=1 to nₘ
        k:=k+1
```

后，$k=n_1+n_2+\cdots+n_m$。所以，当执行完所有的代码后，k 的值是 $n_1+n_2+\cdots+n_m$。

5.1.2 乘法原理

乘法原理要稍微复杂一些，相当于加法原理的一个推论，主要反映乘法是重复加法的这一事实。

定理 5.2(乘法原理) 如果一项工作需要 t 步完成，第一步有 n_1 种不同的选择，第二步有 n_2 种不同的选择，……，第 t 步有 n_t 种不同的选择，那么，完成这项工作所有可能的不同的选择总数为 $n_1\times n_2\times\cdots\times n_t$。

乘法原理可总结为：当一项工作分为若干步时，将每一步的可选择数相乘便得到这项工作的所有可选择个数。

例 5.3 从 5 本不同的计算机书、3 本不同的数学书和 2 本不同的艺术书中选出不同类的两本，共有多少种选法？

解：根据乘法原理不难得出，选择一本计算机书和一本数学书，共有 $5\times3=15$ 种选法；同理，选择一本计算机书和一本艺术书共有 $5\times2=10$ 种选法；选择一本数学书和一本艺术书共有 $3\times2=6$ 种选法。由于这 3 个集合两两不相交，根据加法原理可得：从 5 本不同的计算机书、3 本不同的数学书和 2 本不同的艺术书中选出不同类的两本，共有

$$15+10+6=31$$

种选法。

用乘法原理可证明一个含有 n 个元素的集合共有 2^n 个子集。

例 5.4 用乘法原理证明含有 n 个元素的集合$\{x_1,x_2,\cdots,x_n\}$有 2^n 个子集。

证明：构造一个子集分为 n 个步骤：选取或不选取 x_1；选取或不选取 x_2；……；选取或不选取 x_n。每一步有两种选择，故所有可能的子集总数为

$$\underbrace{2 \times 2 \times \cdots \times 2}_{n个2} = 2^n$$

运用乘法原理可对需要若干步完成的对象计数，当计算不相交子集中对象的总数时，可运用加法原理。

5.2　排列与组合

5.2.1　排列

定义 5.1　n 个不同元素 $x_1, x_2, \cdots, x_n$ 的一种排列为 $x_1, x_2, \cdots, x_n$ 的一个排序。

定理 5.3　n 个元素的排列共有 $n!$ 种。

证明：运用乘法原理。确定 n 个元素的一个排列依次分为 n 个步骤：选择第一个元素；选择第二个元素；……；选择第 n 个元素。第一个元素有 n 种选法；当第一个元素选定后，第二个元素有 $n-1$ 种选法；当第二个元素选定后，第三个元素有 $n-2$ 种选法；以此类推。根据乘法原理，共有 $n(n-1)(n-2)\cdots\times2\times1=n!$ 种排列。

例 5.5　10 个元素的排列共有

$$10! = 10\times9\times8\times7\times6\times5\times4\times3\times2\times1 = 3\ 628\ 800 \text{ 种}$$

定义 5.2　n 个(不同)元素 $x_1, x_2, \cdots, x_n$ 的 r 排列是 $\{x_1, x_2, \cdots, x_n\}$ 的 r 元素子集上的排列。n 个不同元素上的 r 排列的个数记作 $P(n,r)$。

定理 5.4　n 个不同元素上的 r 排列数目为 $P(n,r)=n(n-1)(n-2)\cdots(n-r+1)$，$r\leqslant n$。

证明：对从 n 个不同元素中选取 r 个元素的排列方法进行计数。第一个元素有 n 种选法；当第一个元素选定以后，第二个元素有 $n-1$ 种选法；依次不断选取，直到当第 $r-1$ 个元素选定后，选取第 r 个元素。最后一个元素有 $n-r+1$ 种选法。根据乘法原理，n 个不同元素上的 r 排列数目为 $P(n,r)=n(n-1)(n-2)\cdots(n-r+1)$。

例 5.6　依定理 5.4，$X=\{a,b,c\}$ 上的 2 排列数为

$$P(3,2) = 3\times2 = 6$$

这 6 种排列依次为

$$ab, ac, ba, bc, ca, cb$$

5.2.2　组合

定义 5.3　给定集合 $X=\{x_1, x_2, \cdots, x_n\}$，其包含 n 个元素，从 X 中无序、不重复选取的 r 个元素称为 X 的一个 r 组合，X 的所有 r 组合的个数记作 $C(n,r)$。

接下来用两种方法导出 n 个元素上的 r 组合数 $C(n,r)$ 的公式：第一种方法是利用 $P(n,r)$ 公式导出；第二种方法是直接从 $C(n,r)$ 的性质入手。两种方法将得到相同的 $C(n,r)$ 公式。

将构造一个 n 个元素集 X 上的 r 排列分为两个步骤：选出一个 X 上的 r 组合；将这个 r 组合排序。例如，构造 $\{a,b,c,d\}$ 上的一个 2 排列，先选择一个 2 组合，再将 2 组合

进行排序。图 5.1 说明了如何通过这种方法构造一个$\{a,b,c,d\}$上的 2 排列。由乘法原理可知,r 排列数等于 r 组合数与 r 个元素排列数的乘积,即

$$P(n,r)=C(n,r)r!$$

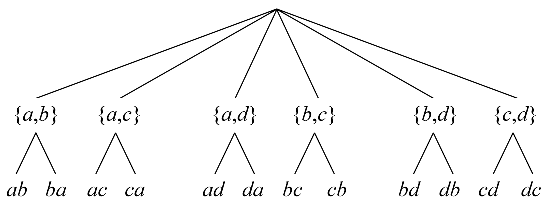

图 5.1 $\{a,b,c,d\}$的 2 排列

于是

$$C(n,r)=\frac{P(n,r)}{r!}$$

下面的定理将给出 $C(n,r)$的另外一种表示法。

定理 5.5 n 个不同元素上的 r 组合数为

$$C(n,r)=\frac{P(n,r)}{r!}=\frac{n(n-1)\cdots(n-r+1)}{r!}=\frac{n!}{(n-r)!\ r!},\quad r\leqslant n$$

例 5.7 (1) 从 10 个人中选出一个 3 个人的委员会,共有多少种不同的选法?

(2) 从 5 个女人和 6 个男人中选出由 2 个女人和 3 个男人组成的委员会,共有多少种选法?

解:(1) 由于委员会中的成员不计次序,故共有

$$C(10,3)=\frac{10\times 9\times 8}{3!}=120\text{ 种}$$

(2) 选出 2 名女性委员,共有 $C(5,2)=10$ 种选法,选出 3 名男性委员,共有 $C(6,3)=20$ 种选法。选出委员会可分为两步:选出女性委员;选出男性委员。根据乘法原理,共有 $10\times 20=200$ 种选法。

5.3 排列组合生成算法

5.3.1 排列生成算法

如果将整数 n 从$\{1,2,\cdots,n\}$的一个排列中删除,则结果是$\{1,2,\cdots,n-1\}$的一个排列。给定一个整数 k,通过在其上画一个向左或向右的箭头表示方向:$\overleftarrow{k}$ 或 $\overrightarrow{k}$。考虑$\{1,2,\cdots,n\}$的一个排列,其中的每一个整数都给定一个方向。如果一个整数 k 的箭头指向一个与其相邻但比它小的整数,那么这个整数 k 就是活动的。例如,

$$\overrightarrow{2}\ \overrightarrow{6}\ \overrightarrow{3}\ \overleftarrow{1}\ \overrightarrow{5}\ \overrightarrow{4}$$

只有 3,5 和 6 是活动的。由此可知,1 绝不可能是活动的,因为$\{1,2,\cdots,n\}$中不存在比 1 还小的整数。除下面两种情况外,整数 n 总是活动的。

(1) n 是第一个整数,而它的箭头指向左边:$\overleftarrow{n}\cdots$。

(2) n 是最后一个整数，而它的箭头指向右边：$\cdots\overrightarrow{n}$。

这是因为只要 n 的箭头指向一个整数，它就是活动的，因为 n 是集合 $\{1,2,\cdots,n\}$ 中最大的整数。下面给出生成 $\{1,2,\cdots,n\}$ 的所有排列的算法。

生成 $\{1,2,\cdots,n\}$ 的排列的算法

从 $\overleftarrow{1}\overleftarrow{2}\cdots\overleftarrow{n}$ 开始。

当存在一个活动的整数时，

(1) 求出最大的活动整数 m；

(2) 交换 m 和其箭头指向的与其相邻的整数；

(3) 交换所有满足 $p>m$ 的整数 p 的方向。

这里就 $n=4$ 叙述该算法。结果用两列显示，第一列给出前 12 个排列。

$$\overleftarrow{1}\,\overleftarrow{2}\,\overleftarrow{3}\,\overleftarrow{4}\qquad\overrightarrow{4}\,\overrightarrow{3}\,\overleftarrow{2}\,\overleftarrow{1}$$

$$\overleftarrow{1}\,\overleftarrow{2}\,\overleftarrow{4}\,\overleftarrow{3}\qquad\overrightarrow{3}\,\overrightarrow{4}\,\overleftarrow{2}\,\overleftarrow{1}$$

$$\overleftarrow{1}\,\overleftarrow{4}\,\overleftarrow{2}\,\overleftarrow{3}\qquad\overrightarrow{3}\,\overleftarrow{2}\,\overrightarrow{4}\,\overleftarrow{1}$$

$$\overleftarrow{4}\,\overleftarrow{1}\,\overleftarrow{2}\,\overleftarrow{3}\qquad\overrightarrow{3}\,\overleftarrow{2}\,\overleftarrow{1}\,\overrightarrow{4}$$

$$\overrightarrow{4}\,\overleftarrow{1}\,\overleftarrow{3}\,\overleftarrow{2}\qquad\overleftarrow{2}\,\overrightarrow{3}\,\overleftarrow{1}\,\overleftarrow{4}$$

$$\overleftarrow{1}\,\overrightarrow{4}\,\overleftarrow{3}\,\overleftarrow{2}\qquad\overleftarrow{2}\,\overrightarrow{3}\,\overleftarrow{4}\,\overleftarrow{1}$$

$$\overleftarrow{1}\,\overleftarrow{3}\,\overrightarrow{4}\,\overleftarrow{2}\qquad\overleftarrow{2}\,\overleftarrow{4}\,\overrightarrow{3}\,\overleftarrow{1}$$

$$\overleftarrow{1}\,\overleftarrow{3}\,\overleftarrow{2}\,\overrightarrow{4}\qquad\overleftarrow{4}\,\overleftarrow{2}\,\overrightarrow{3}\,\overleftarrow{1}$$

$$\overleftarrow{3}\,\overleftarrow{1}\,\overleftarrow{2}\,\overleftarrow{4}\qquad\overrightarrow{4}\,\overleftarrow{2}\,\overleftarrow{1}\,\overrightarrow{3}$$

$$\overleftarrow{3}\,\overleftarrow{1}\,\overleftarrow{4}\,\overleftarrow{2}\qquad\overleftarrow{2}\,\overrightarrow{4}\,\overleftarrow{1}\,\overrightarrow{3}$$

$$\overleftarrow{3}\,\overleftarrow{4}\,\overleftarrow{1}\,\overleftarrow{2}\qquad\overleftarrow{2}\,\overleftarrow{1}\,\overrightarrow{4}\,\overrightarrow{3}$$

$$\overleftarrow{4}\,\overleftarrow{3}\,\overleftarrow{1}\,\overleftarrow{2}\qquad\overleftarrow{2}\,\overleftarrow{1}\,\overrightarrow{3}\,\overrightarrow{4}$$

由于在 $\overleftarrow{2}\overleftarrow{1}\overrightarrow{3}\overrightarrow{4}$ 中没有活动的整数，所以算法终止。

通过对 n 的归纳法可以得知，这个算法生成 $\{1,2,\cdots,n\}$ 的所有的排列，并且具有与前面的方法相同的顺序。我们叙述从 $n=3$ 到 $n=4$ 的归纳法中的一步。从 $\overleftarrow{1}\overleftarrow{2}\overleftarrow{3}\overleftarrow{4}$ 开始，其中 4 是最大的活动整数。4 始终是活动的，直到达到最左边位置为止。此时 4 已经以各种可能的方式插入 $\{1,2,3\}$ 的排列 123 中。现在 4 又不再是活动的。最大的活动整数是 3，它和 $\overleftarrow{1}\overleftarrow{2}\overleftarrow{3}$ 中的最大的活动整数相同。然后 3 和 2 交换位置且 4 改变方向。这个交换与 $\overleftarrow{1}\overleftarrow{2}\overleftarrow{3}$ 中出现的交换是相同的。现在的结果变成 $\overrightarrow{4}\overleftarrow{1}\overleftarrow{3}\overleftarrow{2}$；此时 4 又变成活动的，并将活动状态保持到 4 到达最右边位置为止。然后再进行交换，该交换与发生在 $\overleftarrow{1}\overleftarrow{3}\overleftarrow{2}$ 中的交换相同。算法如此继续进行，4 以各种可能的方式交错地插入 $\{1,2,3\}$ 的每一个排列中。

5.3.2 组合生成算法

令 S 是 n 个元素的集合。为了分析清楚起见，取 S 为集合

$$S=\{x_{n-1},\cdots,x_1,x_0\}$$

现在我们寻找一种生成 S 所有 2^n 个组合(子集)的算法。也就是说，要找一个将 S 的所有组合列出的系统过程。算法的结果应该包含 S 的所有的组合(并且只是 S 的组合)，而且没有重复。

给定 S 的一个组合 A，每一个元素 x_i 或者属于 A 或者不属于 A。如果用 1 表示属于，用 0 表示不属于，就能够用 2^n 个 0 和 1 的 n 元组

$$(a_{n-1},\cdots,a_1,a_0)=a_{n-1}\cdots a_1a_0$$

区分 S 的 2^n 个组合。对于每个 $i=0,1,\cdots,n-1$，令 n 元组的第 i 项 a_i 对应元素 x_i。

例如，当 $n=3$ 时，$2^3=8$ 个组合以及它们对应的 3 元组如下：

	a_2	a_1	a_0
$\varnothing$	0	0	0
$\{x_0\}$	0	0	1
$\{x_1\}$	0	1	0
$\{x_1, x_0\}$	0	1	1
$\{x_2\}$	1	0	0
$\{x_2, x_0\}$	1	0	1
$\{x_2, x_1\}$	1	1	0
$\{x_2,x_1, x_0\}$	1	1	1

例 5.8 令 $S=\{x_6,x_5,x_4,x_3, x_2,x_1, x_0\}$，对应组合 $\{x_5,x_4, x_2, x_0\}$ 的 7 元组 0110101。对应 7 元组 1010001 的组合是 $\{x_6,x_4, x_0\}$。

1. 生成 $\{x_{n-1},\cdots,x_1,x_0\}$ 组合的算法

从 $a_{n-1}\cdots a_1a_0=0\cdots 00$ 开始。

当 $a_{n-1}\cdots a_1a_0\neq 1\cdots 11$ 时，

(1) 求出使得 $a_j=0$ 的最小的整数 j(在 $n-1$ 和 0 之间)；

(2) 用 1 代替 a_j 并用 0 代替 $a_{j-1},\cdots,a_1,a_0$ 中的每一个值(由对 j 的选择可知，在用 0 代替以前，它们都等于 1)。

当 $a_{n-1}\cdots a_1a_0=1\cdots 11$ 时算法结束，它是在结果列表中最后的二进制 n 元组。

定理 5.6 令 $a_1a_2\cdots a_r$ 是 $\{1,2,\cdots,n\}$ 的一个 r-组合。在字典排序中，第一个 r-组合是 $12\cdots r$。最后一个 r-组合是 $(n-r+1)(n-r+2)\cdots n$。设 $a_1a_2\cdots a_r\neq(n-r+1)(n-r+2)\cdots n$。令 k 是满足 $a_k<n$ 且使得 a_k+1 不同于 $a_1,a_2,\cdots,a_r$ 中的任何一个数的最大整数。那么，在字典排序中，$a_1a_2\cdots a_r$ 的直接后继 r-组合是

定理 5.6 证明

$$a_1\cdots a_{k-1}(a_k+1)(a_k+2)\cdots(a_k+r-k+1)$$

从定理 5.6 断言，下列算法生成 $\{1,2,\cdots,n\}$ 的字典序的所有 r-组合。

2. 生成 $\{1,2,\cdots,n\}$ 的字典序 r-组合的算法

从 r-组合 $a_1a_2\cdots a_r=12\cdots r$ 开始。

当 $a_1a_2\cdots a_r\neq(n-r+1)(n-r+2)\cdots n$ 时，

(1) 确定最大的整数 k，使 $a_k+1\leqslant n$ 且 a_k+1 不是 $a_1,a_2,\cdots,a_r$。

(2) 用 r-组合

$$a_1\cdots a_{k-1}(a_k+1)(a_k+2)\cdots(a_k+r-k+1)$$

替换 $a_1a_2\cdots a_r$。

例 5.9　应用生成 $S=\{1,2,3,4,5,6\}$ 的 4-组合算法，得到下列结果。

1234	1256	2345
1235	1345	2346
1236	1346	2356
1245	1356	2456
1246	1456	3456

如果将生成一个集合的排列的算法与生成一个 n-元素集合的 r-组合的算法结合起来，就得到生成 n-元素集合的 r-排列的算法。

例 5.10　生成$\{1,2,3,4\}$的 3-排列。首先生成字典序的 3-组合：123，124，134，234。对于每一个 3-组合，再生成其所有的排列：

123	124	134	234
132	142	143	243
312	412	413	423
321	421	431	432
231	241	341	342
213	214	314	324

通过确定在$\{1,2,\cdots,n\}$的 r-组合的字典序中的每一个 r-组合的位置结束本节。

例 5.11　若按字典序列出$\{1,2,3,4,5,6,7\}$上所有的组合，第一个为 12345，接下来的两个为 12346 和 12347，然后是 12356 和 12357，最后一个为 34567。

例 5.12　若按字典序列出 $X=\{1,2,3,4,5,6,7\}$上所有的组合，13467 的下一个是什么？

以 134 开头的最大的字符串为 13467，故 13467 的下一个以 135 开头。因为 13567 是以 135 开头的最小的字符串，故 13467 的下一个是 13567。不难发现上例中的模式。给定一个与$\{s_1,\cdots,s_r\}$上的 r 组合对应的字符串 $\alpha=s_1\cdots s_r$，求 α 的下一个字符串 $\beta=t_1\cdots t_r$。从右向左找到第一个非最大值的元素 s_m(s_r 的最大值为 n，s_{r-1} 的最大值为 $n-1$，以此类推)，则

$$\begin{aligned}&t_i=s_i,\quad i=1,\cdots,m-1\\&t_m=s_m+1\\&t_{m+1}\cdots t_r=(s_m+2)(s_m+3)\cdots\end{aligned}$$

由此可得组合生成算法。

例 5.13　说明组合生成算法如何生成$\{1,2,3,4,5,6,7\}$上 23467 的下一个 5-组合。设

$$s_1=2,\quad s_2=3,\quad s_3=4,\quad s_4=6,\quad s_5=7$$

s_3 是从右向左第一个非最大值的元素。将 s_3 赋值为 5，s_4 和 s_5 分别被赋值为 6 和 7，

此时,

$$s_1=2,\quad s_2=3,\quad s_3=5,\quad s_4=6,\quad s_5=7$$

于是就生成了23467的下一个5-组合23567。

例5.14 找到{1,2,3,4,5,6}上排列163542的后继,应使左边尽可能多的数字保持不变。

例5.14解答

5.4 广义的排列和组合

前述的排列组合主要考虑不允许重复的情况下,如何对选择和排序计数。本节介绍允许重复的情况下,如何对选择和排序计数,即广义的排列和组合。

定理5.7 设序列S包含n个对象,其中第1类对象有n_1个,第2类对象有n_2个,……,第t类对象有n_t个,则S的不同排序个数为

$$\frac{n!}{n_1!n_2!\cdots n_t!}$$

证明:指定n个对象的位置可得S的一个排序。共有$C(n,n_1)$种不同的方法为n_1个第1类的对象指定位置;指定第1类对象的位置后,共有$C(n-n_1,n_2)$种不同的方法为n_2个第2类的对象指定位置;以此类推。根据乘法原理,不同的排序个数为

$$\begin{aligned}&C(n,n_1)C(n-n_1,n_2)C(n-n_1-n_2,n_3)\cdots C(n-n_1-\cdots-n_{t-1},n_t)\\&=\frac{n!}{n_1!\,(n-n_1)!}\cdot\frac{(n-n_1)!}{n_2!\,(n-n_1-n_2)!}\cdot\cdots\cdot\frac{(n-n_1-\cdots-n_{t-1})!}{n_t!\,0!}\\&=\frac{n!}{n_1!\,n_2!\,\cdots n_t!}\end{aligned}$$

例5.15 将8本不同的书分给3个学生,学生甲分4本,乙分2本,丙分2本,共有多少种不同的分法?

例5.15解答

利用等价关系同样可以证明定理5.7。设序列S包含n个对象,其中第i类有n_i个相同的对象,$i=1,2,\cdots,t$。将S中的同类对象用下标加以区分,得到集合X。例如,序列S为

$$MISSISSIPPI$$

则集合X为

$$\{M,I_1,S_1,S_2,I_2,S_3,S_4,I_3,P_1,P_2,I_4\}$$

在X的所有排列上定义关系R:p_1Rp_2,当且仅当p_1可以通过交换同类对象(但不改变它们的位置)的位置而得到p_2。例如,容易验证如下关系R是X的所有排列集合上的等价关系。

$$(I_1S_1S_2I_2S_3S_4I_3P_1P_2I_4M)R(I_2S_3S_2I_1S_4S_1I_3P_1P_2I_4M)$$

若将同类对象看作相同的,则排列P的等价类中X的所有元素可视为相同,故每个等价类包含$n_1!\,n_2!\,\cdots n_t!$ $(i=1,2,\cdots,t)$个元素。由于等价类与S上的排序一一对应,故等价类的数目等于S上排序的数目。X上的排列共有$n!$个,故由定理5.7,S上排序的个数为

$$\frac{n!}{n_1!n_2!\cdots n_t!}$$

下面讨论允许重复的情况下，如何对不计顺序的选择计数。

定理 5.8 X 为包含 t 个元素的集合，在 X 中允许重复、不计顺序地选取 k 个元素，共有

$$C(k+t-1,t-1)=C(k+t-1,k)$$

种选法。

证明：令 $X=\{a_1,a_2,\cdots,a_t\}$。考虑将 k 个"×"和 $t-1$ "|"填入下面的 $k+t-1$ 个空格中，

$$\_\ \_\ \_\ \cdots\ \_\ \_$$

每种排列方法决定 X 上的一个选择：第一个"×"左边"|"的个数 n_1 表示选择 n_1a_1 的个数；第一个"|"和第二个"|"之间"×"的个数 n_2 表示选择 n_2a_2 的个数；以此类推。由于为 $t-1$ 个"|"选定位置共有 $C(k+t-1,t-1)$种选法，故共有 $C(k+t-1,t-1)$个 X 上的选择。若考虑为 k 个"×"选定位置，则共有 $C(k+t-1,k)$种选法，所以共有

$$C(k+t-1,t-1)=C(k+t-1,k)$$

种选法，在 X 上允许重复、不计顺序地选取 k 个元素。

例 5.16 把 12 本相同的数学书分给甲、乙、丙、丁 4 个学生，共有多少种分法？

若将问题看作在 12 本书上分别写上 4 个学生的名字，则可利用定理 5.8 计算分法数。这相当于在集合{甲，乙，丙，丁}上允许重复、不计顺序地选择 12 个元素。根据定理 5.8，共有

$$C(12+4-1,4-1)=C(15,3)=455$$

种分法。

例 5.17 (a)方程 $x_1+x_2+x_3+x_4=29$ 有多少个非负整数解？(b)上述方程有多少满足 $x_1>0,x_2>1,x_3>2,x_4\geqslant 0$ 的整数解？

例 5.17 解答

5.5 二项式系数和组合恒等式

本节主要讨论组合数的性质、组合数序列的求和以及组合恒等式的证明等内容。本节引入一个新的符号$\binom{n}{r}$，当 n 和 r 都是自然数时，它就等于组合数 $C(n,r)$。

5.5.1 二项式定理

表达式$(a+b)^n$ 看似与组合数无关，但通过 n 个对象的 r 组合数可以得出表达式$(a+b)^n$的展开式。代数表达式在很多情况下都与计数问题相关，利用代数方法常常可以得到一些高级的计数技巧。二项式定理给出了$(a+b)^n$ 展开的各项系数。由于

$$\underbrace{(a+b)(a+b)\cdots(a+b)}_{n\text{个因子}}$$

从每个因子中选择 a 或 b，将 n 个因子中的选择相乘，再将所有选择的乘积相加，即得展开式。例如，为展开$(a+b)^3$，需从第一个因子中选择 a 或 b，从第二个因子中选择 a 或 b，从第三个因子中选择 a 或 b，将选出的 3 项相乘得展开式中的一项，将所有选择的乘

积相加得展开式。若从3个因子中都选择 a,则得项 aaa;若从第一个因子中选择 a,从第二个因子中选择 b,从第三个因子中选择 a,则得项 aba。表5.1列出了所有可能的项。将所有选择的乘积相加,有

$$\begin{aligned}(a+b)^3 &= (a+b)(a+b)(a+b) \\ &= aaa+aab+aba+abb+baa+bab+bba+bbb \\ &= a^3+a^2b+a^2b+ab^2+a^2b+ab^2+ab^2+b^3 \\ &= a^3+3a^2b+3ab^2+b^3\end{aligned}$$

式中,在 n 个因子中选择 k 个 b 和 $n-k$ 个 a,可得项 $a^{n-k}b^k$。因为从 n 个对象中选择 k 个共有 $C(n,k)$ 种选法,所以项 $a^{n-k}b^k$ 共有 $C(n,k)$ 个,则

$$\begin{aligned}(a+b)^n = {}& C(n,0)a^nb^0+C(n,1)a^{n-1}b^1+C(n,2)a^{n-2}b^2+\cdots+ \\ & C(n,n-1)a^1b^{n-1}+C(n,n)a^0b^n\end{aligned}$$

这就是所谓的二项式定理(见表5.1)。

表5.1 计算 $(a+b)^3$

从第一个因子 $(a+b)$ 中选择	从第二个因子 $(a+b)$ 中选择	从第三个因子 $(a+b)$ 中选择	选择结果的乘积
a	a	a	$aaa=a^3$
a	a	b	$aab=a^2b$
a	b	a	$aba=a^2b$
a	b	b	$abb=ab^2$
b	a	a	$baa=a^2b$
b	a	b	$bab=ab^2$
b	b	a	$bba=ab^2$
b	b	b	$bbb=b^3$

定理5.9(二项式定理) 设 a 和 b 为实数,n 为正整数,则

$$(a+b)^n=\sum_{k=0}^{n}C(n,k)a^{n-k}b^k$$

利用数学归纳法归纳于 n,同样可证二项式定理。

$C(n,r)$ 为 $a+b$ 的幂的展开式中的系数,故称为**二项式系数**。

例5.18 在定理5.9中令 $n=3$,可得

$$\begin{aligned}(a+b)^3 &= C(3,0)a^3b^0+C(3,1)a^2b^1+C(3,2)a^1b^2+C(3,3)a^0b^3 \\ &= a^3+3a^2b+3ab^2+b^3\end{aligned}$$

例5.19 利用二项式定理展开 $(3x-2y)^4$。

解:在定理5.9中,令 $a=3x$,$b=-2y$,$n=4$,可得

$$\begin{aligned}(3x-2y)^4 &= (a+b)^4 \\ &= C(4,0)a^4b^0+C(4,1)a^3b^1+C(4,2)a^2b^2+C(4,3)a^1b^3+C(4,4)a^0b^4 \\ &= C(4,0)(3x)^4(-2y)^0+C(4,1)(3x)^3(-2y)^1+C(4,2)(3x)^2(-2y)^2+ \\ &\quad C(4,3)(3x)^1(-2y)^3+C(4,4)(3x)^0(-2y)^4\end{aligned}$$