

组合数学

(课程试用版本)

清华大学《组合数学》教材编写组

2023 年 9 月 14 日

前言

本书是清华大学研究生课程《组合数学》的教学用书。往届课程中使用的教材已经较为老旧，与课程教学体系也不完全匹配。因此，教材编写组决定撰写本书，作为一本适配课程的讲义。

本书的目标是建立完整、自洽且流畅的组合数学理论体系。为此，教材编写组以课件幻灯片、学堂在线视频课程为基础梳理了课程内容逻辑，并按照这一逻辑在本书中安排了各项知识点的讲解。

在撰写本书的过程中，教材编写组大量参考了以下教材：

- 卢开澄，卢华明.《组合数学（第4版）》.北京：清华大学出版社，2008；
- 许胤龙，孙淑玲.《组合数学引论》.安徽：中国科学技术大学出版社，2010；
- [美] Brualdi, Richard A.《组合数学（原书第4版）》.冯舜玺，罗平，裴伟东译，卢开澄、冯舜玺校.北京：机械工业出版社，2010；
- [美] Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., Stein, Clifford.《算法导论（原书第3版）》.殷建平，徐云，王刚，刘晓光，苏明，邹恒明，王宏志译.北京：机械工业出版社，2013.

在此对这些教材的著者、译者和编者致以诚挚的感谢。

目录

绪论	1
第一章 排列组合	2
1.1 基本计数原理	2
1.2 集合的排列与组合	7
1.2.1 排列	7
1.2.2 组合	10
1.3 球盒模型与格路模型	13
1.3.1 球盒模型	13
1.3.2 格路模型	14
1.4 二项式定理与组合恒等式	17
1.5 多重排列与环形排列	19
1.5.1 多重排列	20
1.5.2 环状排列	22
1.6 可重组与不相邻组合	24
1.6.1 可重组	24
1.6.2 不相邻组合	29
1.7 生成全排列	29
1.7.1 Stirling 近似公式	30
1.7.2 字典序法	31
1.7.3 递增进位制数法	34
1.7.4 递减进位制数法	36
1.7.5 SJT 算法	37
1.8 生成组合	40
习题	40
第二章 鸽巢原理	42
2.1 鸽巢原理的基本形式	42
2.2 鸽巢原理的推广形式	45

2.3 整点问题	47
2.4 Ramsey 问题	49
2.4.1 完全图二染色的 Ramsey 定理	50
2.4.2 Ramsey 定理的推广形式	52
习题	54
第三章 母函数	55
3.1 引论	55
3.2 母函数的性质	58
3.3 整数拆分与 Ferrers 图像	62
3.3.1 有序拆分	63
3.3.2 无序拆分	63
3.3.3 Ferrers 图像	68
3.4 指数型母函数	70
习题	75
第四章 线性常系数递推关系	76
4.1 引论	76
4.2 Fibonacci 数列	78
4.3 母函数与递推关系	85
4.4 齐次线性常系数递推关系	96
4.4.1 特征多项式	97
4.4.2 通过特征多项式求解齐次线性常系数递推关系	98
4.5 非齐次线性常系数递推关系	112
4.5.1 差分法	112
4.5.2 特解法	118
习题	124
第五章 特殊计数序列	126
5.1 Catalan 数	126
5.2 错位排列	133
5.3 第二类 Stirling 数	135
5.4 第一类 Stirling 数	145
第六章 容斥原理	149
6.1 容斥原理及其证明	149
6.2 带约束的排列问题	156
6.3 带约束的组合问题	165
6.4 广义容斥原理	170

6.5 Möbius 反演	171
习题	180
第七章 Pólya 计数理论	181
7.1 群论基础	181
7.2 置换群	185
7.3 Burnside 引理	193
7.4 Pólya 计数定理	198
7.5 空间多面体的染色问题	201
习题	210
附录 A 离散数学基础	212
A.1 特殊数学运算与记号法则	212
A.2 集合与偏序关系	214
A.3 偏序卷积与 Möbius 反演	219
附录 B 代数结构	228
B.1 群	228
B.2 环与形式幂级数环	230

绪论

欢迎选修《组合数学》课程。这是清华大学计算机系主要面向本系研究生新生开设的一门数学基础课。本书是与本课程配套的讲义，目前它还并不十分完善（例如，你现在读到的这段文字本应是一段完整的绪论，但它还没有完成），如果你对本书有任何建设性的意见，欢迎联系教学团队。

期待你能享受这段组合数学的学习过程。

第一章 排列组合

计数问题是组合数学学科中最普遍的一类问题，读者在正式接触组合数学这门应当就已经有许多完成计数任务的经验。计数的核心原则是**不重不漏**，即每一个被计数的对象都应该恰好被计算一次，不能多也不能少。

从数学运算的角度来说，为了求解计数问题，我们总是需要对数字进行加减乘除等基本运算以求出最终结果。加、减、乘、除这四则基本运算分别对应于一条基本计数原理，各条原理遵循“不重不漏”的前提，指明了对应的运算适合用在何种问题场景中。在这四条计数原理的支持下，求解计数问题时进行的每一步运算都有明确的组合学意义。

排列与组合建立在四条基本计数原理之上，它们对组合数学中常见的问题形式进行了进一步的抽象。大部分计数问题都能转化为排布一个集合中的元素、并统计符合要求的排布方案数目的问题。这时，若我们关心集合中元素的排布顺序，则问题就称为**排列**，否则就称为**组合**。

本章中，我们将依次介绍四条基本计数原理、排列与组合，以及与其相关的计数公式。经典的排列组合问题有数个常见变种，同时我们也有两个通用的数学模型可用于刻画并解决这些变种问题，本章将会一并介绍。最后，我们讨论如何生成一个已知集合的所有全排列和组合。

1.1 基本计数原理

加法原理

加法原理指出全体等于部分的和。

定理 1.1 (加法原理). 设 S 是有限集， S 可被划分为 S_1, S_2, \dots, S_n 这 n 个集合，满足：

$$S = S_1 \cup S_2 \cup \dots \cup S_n \quad (\text{完备性})$$

$$S_i \cap S_j = \emptyset \quad (1 \leq i, j \leq n, i \neq j) \quad (\text{互斥性})$$

则 S 的大小等于划分出的各集合大小之和：

$$|S| = |S_1| + |S_2| + \dots + |S_n| \quad (1-1)$$

完备性 (completeness) 和 **互斥性** (exclusiveness) 是加法原理的两个关键条件, 体现了“不重不漏”的关键思想. 完备性意味着“部分”加起来必须能完整覆盖“全体”, 体现了“不漏”; 互斥性意味着 S_1, S_2, \dots, S_n 两两交集为空, 即“部分”之间不能有任何重叠, 体现了“不重”.

⚠ 注意

加法原理仅适用于 S 是**有限集**的情况, 其余三条基本计数原理也是如此. 本书后续使用“集合”一词时, 除非特别说明, 指的均是有限集.

有些问题中 S_1, S_2, \dots, S_n 均是已知的, 而且这些集合之间也显然没有公共元素, 这时利用加法原理将这些部分合并起来即可得到想要的整体. 在另一些问题中, 我们需要自行构造 S_1, S_2, \dots, S_n , 这时为了使它们两两交集为空, 我们必须为这些集合设置彼此互斥的“准入条件”; 同时, 所有集合的“准入条件”合在一起又应当是完备的, 使得 S 中的每一个元素都一定属于 S_1, S_2, \dots, S_n 中的某一个集合.

例 1.1.1. 某人希望在某日乘铁路从北京直达上海, 中间不能换乘. 经在 12306 官网上查询, 该日从北京到上海, 一共有 $S_G = 44$ 个高铁车次、 $S_D = 2$ 个动车车次、 $S_Z = 1$ 个直达特快车次、 $S_T = 1$ 个特快车次以及 $S_{\text{other}} = 1$ 个普快车次可供选择. 因此, 出行方案的总数为

$$S = S_G + S_D + S_Z + S_T + S_{\text{other}} = 49$$

这个例子中, 我们将“北京到上海的铁路车次”这一全体, 按照车次的类型, 划分为高铁、动车、直达特快、特快和普快五个部分. 这五种类型涵盖了北京到上海的所有客运列车车次, 所以五个部分加在一起是完备的; 又因为每个车次只可能属于其中一种类型, 所以五个部分之间是互斥的. 因此, 按车次类型进行的划分拥有完备性和互斥性, 从而使用加法原理可以得到正确的结果.

例 1.1.2. 一所大学想要对实验室资产 (例如工位屏风、工位电脑、服务器、空调, 等等) 做一次清查. 根据这所大学的管理制度, 每个实验室都恰好由一个院系管理, 并且实验室的每一样资产都恰好由一位受雇于该院系的教师来负责. 因此, 管理部门可以将清查任务分配到每一个院系, 院系又可以将任务分配给它雇佣的每一位教师, 教师统计属于他 (她) 自己的实验室资产并上报. 这样一来, 院系只需要将每位教师管理的资产相加, 就得到该院系下属的资产总额; 学校管理部门再将每个院系的资产相加, 就得到整所大学所有实验室的资产总额.

这个例子运用了两次加法原理. 第一次中, 我们将“大学拥有的实验室资产”这一全体, 划分为“每个院系拥有的实验室资产”这些部分. 由于每个实验室都隶属于恰好一个院系, 所以这一次划分是不重不漏的. 第二次中, 我们将“一个院系拥有的实验室资产”这一全体, 划分为“每个老师拥有的实验室资产”这些部分. 由于每项资产也恰好属于一位老师, 所以这一次划分也是不重不漏的.

乘法原理

整数乘法可以看作加法的推广, 同理, 乘法原理也可看作加法原理的推广.

定理 1.2 (乘法原理). 设 S 是满足下列条件的全体有序对 (a, b) 构成的集合: a 有 n 种可能的取值, 同时对于 a 的每一个取值, b 都有 m 种取值. 则 S 的大小等于 n, m 之积:

$$|S| = nm \quad (1-2)$$

可以用加法原理来证明乘法原理. 设 $a \in \{a_1, a_2, \dots, a_n\}$, 可以将 S 划分为 n 个部分 S_1, S_2, \dots, S_n , 其中 S_i ($1 \leq i \leq n$) 包含了 S 中所有形如 (a_i, b) 的有序对. 显而易见, 这一划分方式是完备且互斥的. 由于对任何给定的 a_i , b 都有 m 种取值方式, 因此

$$|S_1| = |S_2| = \dots = |S_n| = m$$

这样一来, 根据加法原理,

$$\begin{aligned} |S| &= |S_1| + |S_2| + \dots + |S_n| \\ &= \underbrace{m + m + \dots + m}_{n \text{ 个 } m} \\ &= nm \end{aligned}$$

乘法原理的关键在于, 对于 a 的每一种可能的取值, b 一视同仁地均有 m 种取值方式. 最常见的情况就是 a, b 的取值彼此独立, 互不影响; 有时虽然 b 的取值依赖 a 的取值, 但 b 的取值方式种类数始终为 m , 这种情况也适用乘法原理.

上面讨论的乘法原理只涉及两个元素构成的有序对, 但是结论可以自然地推广到任意有限多个元素构成的有限元组的情形:

推论 1.3. 设 S 是满足下列条件的全体有序元组 (a_1, a_2, \dots, a_k) 构成的集合: a_1 有 n_1 种可能的取值, 同时对于 a_1, a_2, \dots, a_t ($1 \leq t < k$) 的每一组取值, a_{t+1} 都有 n_{t+1} 种取值. 则 S 的大小等于 n_1, n_2, \dots, n_k 之积:

$$|S| = n_1 n_2 \dots n_k = \prod_{i=1}^k n_i \quad (1-3)$$

证明. 对 k 施加归纳. 当 $k = 1$ 时命题成立, 当 $k = 2$ 时由乘法原理命题成立. 当 $k > 2$ 时, 有序元组 (a_1, a_2, \dots, a_k) 可以视为有序对 $((a_1, a_2, \dots, a_{k-1}), a_k)$. 根据归纳假设, 该有序对的第一个元素 $(a_1, a_2, \dots, a_{k-1})$ 有 $n_1 n_2 \dots n_{k-1}$ 种取值方式; 并且由定义, 对 $(a_1, a_2, \dots, a_{k-1})$ 的每一种取值, a_k 都有 n_k 种取值方式. 因此, 由乘法原理, 我们有

$$|S| = (n_1 n_2 \dots n_{k-1}) \cdot n_k = \prod_{i=1}^k n_i$$

□

下面我们讨论是几个乘法原理的简单例子.

例 1.1.3. 某人希望在某日乘铁路从北京前往厦门, 具体出行方案是首先乘车直达上海, 到达上海的当天下车休息, 次日再乘车直达厦门. 经在 12306 官网上查询, 出行当日从北京直达上海, 一共有 49 个车次可供选择; 从上海到厦门, 每天均有 21 个车次可供选择. 因此, 出行方案总数为 $49 \times 21 = 1029$.

上面这个例子没有限定换乘时间, 同时巧合的是 (在本书撰写时) 上海到厦门每天的车次数都相同, 因此无论第一段旅程选择哪个车次, 第二段旅程可选择的车次数都是固定的, 满足使用乘法原理的条件. 如果上述假设发生了改变, 例如中途下车不再休息, 必须在 2 小时内乘上下一班列车, 则第二段旅程的车次选择将受到第一段旅程的影响, 这种独立性就消失了, 乘法原理也很可能就不再适用.

例 1.1.4. 一家餐厅的单人套餐允许顾客在 6 种主食、4 种小食和 10 种饮料中分别选择一种. 如此一来, 不同的饮食搭配的种类数量为 $6 \times 4 \times 10 = 240$.

例 1.1.5. 求 $151361784 = 2^3 \times 3^5 \times 7^3 \times 227$ 约数的个数.

151361784 的所有的约数都应该具有如下形式:

$$2^a \times 3^b \times 7^c \times 227^d \quad (0 \leq a \leq 3, 0 \leq b \leq 5, 0 \leq c \leq 3, 0 \leq d \leq 1)$$

其中 a, b, c, d 四个非负整数的取值彼此独立, 互不影响. 根据乘法原理, 上面这个大数一共有 $4 \times 6 \times 4 \times 2 = 192$ 个不同的约数.

减法原理

在四则运算中, 减法可看作加法的逆运算. 因此, 减法原理可看作加法原理逆转过来的情况.

定理 1.4 (减法原理). 设 S 是有限集, 并且是一个更大的有限集 U 的子集. 记 S 在 U 中的补集为 \bar{S} , 则 \bar{S} 的大小等于 U, S 的大小之差:

$$|\bar{S}| = |U| - |S| \quad (1-4)$$

根据补集的定义, \bar{S} 中的任何元素均不属于 S , 因此 S 和 \bar{S} 是两个不相交的集合, 二者满足互斥性. 另一方面, 如果 U 中的某个元素不属于 S , 则该元素一定属于 \bar{S} , 从而 U 中的元素必属 S, \bar{S} 之一, 二者满足完备性. 因此, S 和 \bar{S} 可看作对 U 的一个划分. 根据加法原理, 我们有 $|U| = |S| + |\bar{S}|$, 对该式进行移项就得到减法原理的式 (1-4).

例 1.1.6. 一名同学决定靠抛硬币来合理安排他的研究生生活. 每天起床后, 他会抛掷 3 枚公平的硬币, 只要出现了正面就去实验室开展科研, 否则就在宿舍休息. 求他每周平均有多少天在科研.

假设抛硬币的所有可能结果构成集合 U , 其中不出现正面的那些结果构成集合 S , 那么我们要求的就是 $|\overline{S}|$. 每抛一枚硬币, 会等可能性地出现正、反两种结果, 根据乘法原理, 抛掷 3 枚硬币一共有 $|U| = 2 \times 2 \times 2 = 8$ 种结果, 这些结果出现的可能性都相等. 在所有可能的情况中, 不出现正面的情况只有一种, 即三次都抛出反面, 因此 $|S| = 1$. 根据减法原理,

$$|\overline{S}| = |U| - |S| = 8 - 1 = 7$$

这名同学平均每 8 天中会科研 7 天、休息 1 天. 因此, 他每周平均花

$$\frac{7}{8} \times 7 = 6.125$$

天在科研上.

例 1.1.7. 一个班级中有 20 位男生和 10 位女生, 现在希望选举一位男生担任班长、一位女生担任班副班长. 不过, 班级中有一对双胞胎兄妹, 为了让选举结果看起来公平, 这对兄妹不应该同时当选. 求有多少种满足这一要求的选举结果.

这个例子有两个解决方案. 在第一个方案中, 我们按照班长人选分类讨论. 如果兄妹中的哥哥没有当选班长, 那么其他 19 位男生可以当选, 同时 10 位女生也都有权担任班副班长. 另一方面, 如果他当选班长, 那么只有 9 位女生能担任班副班长. 运用乘法原理和加法原理, 合法的选举结果种类数量为:

$$19 \times 10 + 9 = 199$$

在第二个方案中, 我们先不考虑追加的限制条件, 这时根据乘法原理, 一共有 $20 \times 10 = 200$ 种选举结果. 然而, 兄妹同时当选的结果是不合题意的, 需要从中去除, 这样的结果仅有唯一一种可能. 运用乘法原理和减法原理, 合法的选举结果种类数量为:

$$20 \times 10 - 1 = 199$$

可以看到, 两种计算方式得到的结果是相同的.

除法原理

与减法原理类似, 除法原理可看作乘法原理逆转过来的情况.

定理 1.5 (除法原理). 设 S 是有限集, S 可被划分为 S_1, S_2, \dots, S_n 这 n 个集合, 在满足完备性和互斥性的同时, 还有:

$$|S_1| = |S_2| = \dots = |S_n|$$

则 S_1, S_2, \dots, S_n 中每个集合的大小都等于 S 的大小除以 n :

$$|S_1| = |S_2| = \dots = |S_n| = \frac{|S|}{n} \quad (1-5)$$

如果我们反过来知道每一部分的大小,但部分的总数是未知的,则除法原理同样适用.唯一要做的是对式 (1-5) 进行移项:

$$n = \frac{|S|}{|S_1|} \quad (1-6)$$

例 1.1.8. 某课程有 300 位同学参加期末考试,任课教师打算把他们安排在能容纳 60 人的考场中.如此一来,所需要的考场数目就是:

$$\frac{300}{60} = 5$$

假如教务处发现考场紧张,只有 3 个考场可用,那么每个考场中的人数就变为:

$$\frac{300}{3} = 100$$

例 1.1.9. 食堂设有 20 个窗口,每个窗口都准备了 8 道菜式,并且不同窗口的菜式没有重复.一位同学每顿饭吃 2 道菜,假如他想吃遍该食堂的所有菜式,那么根据乘法原理和除法原理,完成这个任务需要的天数就是

$$\frac{20 \times 8}{2} = 80$$

1.2 集合的排列与组合

排列和组合是两类典型的计数模型,它们考察从一个有限集合中选出若干个元素的方案数.如果需要考虑选出元素的顺序,则称为对集合中元素的排列;反之,如果不考虑元素的顺序,则称为对集合中元素的组合.

▲ 注意

排列和组合考虑的均是从集合中选出元素的方案数,这一定是一个非负整数,而不会是负数或者整数以外的实数.因此,方便起见,我们约定本节中出现的变量(除集合外)均是非负整数,除非我们给出特别说明.

1.2.1 排列

设 S 是有限集, $|S| = n$. 从 S 中选出 k 个元素排成有序的一行,称为 S 的一个 k 元素排列,方案数记为 $P(n, k)$,称为排列数,其中 P 意为排列(permutation). 为了强调将元素“排成一行”,这种排列有时也被称为线排列. 特别地,当 $n = k$ 时,排列中包含 S 中全部元素,称为 S 的全排列.

例 1.2.1. 从 $S = \{1, 2, 3\}$ 中选出 $k = 2$ 个元素排成一行,有 $P(3, 2) = 6$ 种方案:

12, 13, 21, 23, 31, 32

如果元素的排列方式是“排成一行”的等价形式, 则问题也适用排列这一模型. 例如, 将元素排列成给定尺寸的矩阵, 本质上和排列成一行没有区别.

例 1.2.2. 在奥运会的游泳比赛决赛赛场上, 8 个赛道上各有一位选手, 最终会有 3 位选手分获金银铜牌 (假设不存在并列名次). 选手们角逐出奖牌获得者的过程可以看作 $n = 8, k = 3$ 的排列问题. 另一方面, 如果不只看前三名, 而是考虑每一位选手的名次, 那么问题就变为 $n = 8, k = 8$ 的排列问题, 也就是全排列. 选手的排名顺序直接关系到他们的名次和奖牌, 所以“顺序”在这个例子中是很重要的.

例 1.2.3. 许多读者可能对高考试卷中所谓的“七选五”题型并不陌生. 这类题目要求考生从 7 个选项中选择 5 个, 填写到试卷上的 5 个空中, 每个选项至多使用一次. 这可以看作 $n = 7, k = 5$ 的排列问题. 选项必须按照正确的顺序填入每个空中, 否则就算作错误回答, 所以“顺序”在这个例子中也是很重要的.

下面我们讨论排列数 $P(n, k)$ 的计算方式. 考虑按如下的方式从集合 S 中依次选择 k 个元素, 生成一个长为 k 的排列:

- 第一步, $|S| = n$, 从 S 中选取一个元素, 并将其放置在排列的第一个位置; 这个元素已经不能再次选用了, 因此我们将其从 S 中移除;
- 第二步, $|S| = n - 1$, 从 S 中再选取一个元素, 并将其放置在排列第二个位置; 这个元素同样要从 S 中移除.
-
- 第 k 步, $|S| = n - k + 1$, 从 S 中选取一个元素, 并将其放置在排列的第 k 个位置.

在每一步中, 我们都可以从 S 中任取一个元素, 因此取出元素的方案数就等于那一步中 S 的大小. 此外, 所有的 k 步显然也各自独立, 例如无论第一步取哪一个元素, 第二步中 S 的大小均为 $n - 1$. 根据乘法原理, 从 S 中取出 k 个元素排成有序的一行的方案数, 应当等于每一步中选取元素的方案数之积.

定义 1.6. 设 $|S| = n$, S 的 k 元素排列的方案数 $P(n, k)$ 为:

$$P(n, k) = n(n-1)(n-2) \cdots (n-k+1) = \prod_{k=0}^{k-1} (n-k) \quad (1-7)$$

在这里, 我们给出了一个定义, 而非定理. 尽管式 (1-7) 是我们利用乘法原理推出的, 但它应当被视作 $P(n, k)$ 的原始定义. 这样做的优点是, 我们之前的讨论实际上隐含了集合 S 非空, 且 k 个元素都能成功选取的假设 (即 $n \geq k > 0$), 但定义式 (1-7) 则可将 n, k 的定义域都扩展到全体非负整数上. 实际上, 此式还可推广至 n 为实数的情况, 此时 $P(n, k)$ 称为 n 的 k 次下降幂, 我们将在 5.3 节中作简单的讨论.

推论 1.7. 若 $n \geq k$, 则

$$P(n, k) = \frac{n!}{(n-k)!} \quad (1-8)$$

推论 1.8. 若 $k > n$, 则 $P(n, k) = 0$.

以上两条推论都有实际意义. 推论 1.7 可以理解为除法原理: 从 n 个元素中选出 k 个元素排成一行, 可以看作先把 n 个元素全都排成一行, 然后再去掉尾部的 $n - k$ 个元素. 推论 1.8 更加直观: 要从一个集合中选出比元素总数还多的元素, 显然是不可能的, 因此方案数自然即为 0.

此外, 从推论 1.7 出发, 还能得到两个常用结论:

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

$$P(n, 0) = 1$$

第一个结论的意义是 n 个元素的全排列数目恰好为 $n!$, 这也同时是 $P(n, n)$ 的含义. 第二个结论的意义是若要从一个集合中选出 0 个元素, 则应该视为有且仅有一种方案, 即“什么都不做”. 可以通过乘法原理来理解这条结论: 设想从 $n + k$ 个元素中选出 k 个元素排成一行, 可以分两步完成: 第一步先从 $n + k$ 个元素中选出 k 个元素排成一行, 第二步再从剩下的 n 个元素中选出 0 个, 相当于什么都不做. 这两个步骤彼此独立的, 因此由乘法原理, 我们有

$$P(n + k, k) = P(n + k, k) \cdot P(n, 0)$$

从而必然有 $P(n, 0) = 1$.

下面我们给出 $P(n, k)$ 满足的两个递推关系.

推论 1.9. 设 $n > 0, k > 0$, 则:

$$P(n, k) = n \cdot P(n - 1, k - 1)$$

证明. 设想从集合中逐个选出元素, 这个过程可以分为“选出第一个元素”和“选出剩下的元素”两个部分. 选出第一个元素有 n 种方案. 在这之后, 需要在剩余的 $n - 1$ 个元素中再选出 $k - 1$ 个元素, 按顺序排在之前选出的第一个元素后面, 有 $P(n - 1, k - 1)$ 种方案. 由乘法原理, $P(n, k) = n \cdot P(n - 1, k - 1)$. \square

推论 1.10. 设 $n > 0, k > 0$, 则:

$$P(n, k) = P(n - 1, k) + k \cdot P(n - 1, k - 1)$$

证明. 从集合中随意选出一个元素, 不妨把这个特别的元素记为 x . 因为 $k > 0$, 集合的 k 元素排列一定能分为“不包含 x ”和“包含 x ”两类.

- 如果排列中不包含 x , 那么相当于从剩余的 $n - 1$ 个元素中选出 k 个排成一行, 方案数为 $P(n - 1, k)$;
- 如果排列中包含 x , 那么 x 可以放在 k 个位置中的任何一个, 其他的 $k - 1$ 个位置则可放置剩余的 $n - 1$ 个元素, 总方案数为 $k \cdot P(n - 1, k - 1)$.

由加法原理, $P(n, k) = P(n - 1, k) + k \cdot P(n - 1, k - 1)$. \square

1.2.2 组合

设 S 是有限集, $|S| = n$. 从 S 中选出 k 个元素, 不考虑这些元素之间的顺序, 称为 S 的一个 k 元素组合, 方案数记为 $C(n, k)$, 称为组合数, 其中 C 意为组合 (combination).

例 1.2.4. 从 $S = \{1, 2, 3\}$ 中选出 $k = 2$ 个元素, 不考虑元素之间的顺序, 有 $C(3, 2) = 3$ 种方案:

$$\{1, 2\}, \{1, 3\}, \{2, 3\}$$

S 的 k 元素组合实际上即等价于 S 的大小为 k 的子集.

例 1.2.5. 在人大代表选举中, 选民在选票上认可的候选人名字前画“○”. 如果候选人有 4 人, 要选出 3 人, 那么填写选票的过程可以看作 $n = 4, k = 3$ 的组合问题. 只要最终选择的候选人的集合相同, 先选哪位、后选哪位并不会对最终的选票产生影响, 所以“顺序”在这个例子中无关紧要.

例 1.2.6. 许多文字编辑软件都允许给字体设置加粗、斜体和下划线三种样式. 这些样式不存在先后关系: 例如, 无论是先设置加粗再设置斜体, 还是先设置斜体再设置加粗, 最后都得到同样的“加粗斜体”字体, 所以“顺序”在这个例子中无关紧要. 因此, 字体样式可看作 $n = 3, k = 0, 1, 2, 3$ 的组合问题.

下面我们讨论组合数 $C(n, k)$ 的计算方式. 我们可以借助之前讨论过的集合排列. 组合只关心使用了集合中的哪些元素, 不关心它们的顺序; 排列既关心使用的元素也关心它们的顺序, 因此我们考虑从排列的方案数中去掉元素顺序造成的影响, 以此求出组合的方案数.

集合 S 的 k 元素排列一共有 $P(n, k)$ 个, 但是其中有一些排列使用了相同的元素, 只是元素排列的顺序不同. 假设我们已经确定好了要选出的 k 个元素. 这时, 根据排列数的定义, 这 k 个元素一共有 $P(k, k)$ 种不同的排列顺序, 每种排列顺序都对应一个不同的排列, 但所有排列顺序都对应同一个组合. 因此, 我们可以将全部 $P(n, k)$ 个排列按照其中使用的元素分组, 每组中含有 $P(k, k)$ 个元素种类相同但顺序不同的排列, 对应于一个组合. 利用除法原理去掉元素排列顺序对结果造成的影响, 即可利用 $P(n, k)$ 求出 $C(n, k)$.

定义 1.11. 设 $|S| = n$, S 的 k 元素组合的方案数 $C(n, k)$ 为:

$$C(n, k) = \frac{P(n, k)}{P(k, k)} = \frac{1}{k!} \prod_{k=0}^{k-1} (n - k) \quad (1-9)$$

与排列数那时的情况类似, 我们将式 (1-9) 看作组合数 $C(n, k)$ 的定义式, 使之对所有非负整数 n, k 皆有定义. 实际上, 此式还可推广至 n 为实数的情况, 我们将在 1.4 节中作简单的讨论.

推论 1.12. 若 $n \geq k$, 则

$$C(n, k) = \frac{n!}{k!(n-k)!} \quad (1-10)$$

推论 1.13. 若 $k > n$, 则 $C(n, k) = 0$.

组合数可以构成很多恒等式, 统称为组合恒等式. 组合恒等式不仅在求解组合计数问题时发挥重要作用, 而且也是组合数学领域的一个重要研究范畴. 下面我们以定理的形式给出几个简单的组合恒等式, 并且解释它们的组合意义.

定理 1.14. $C(n, k) = C(n, n - k)$.

证明. 通过式 (1-10) 可以直接得出这个恒等式, 因为将 k 替换为 $n - k$ 后等号右侧分式的分母不变. 这个恒等式表明, 从 n 个元素中选出 k 个的方案, 等价于将该集合分为大小为 k 和 $n - k$ 的两部分的方案, 因此也就等价于选出 $n - k$ 个元素的方案. 换句话说, 选择集合 S 的一个子集, 和选择该子集在 S 中的补集是等价的. \square

定理 1.15. 设 $n > 0, k > 0$, 则

$$C(n, k) = C(n - 1, k) + C(n - 1, k - 1) \quad (1-11)$$

证明. 从集合中随意选出一个元素, 不妨把这个特别的元素记为 x . 因为 $k > 0$, 集合的 k 元素组合一定能分为“不包含 x ”和“包含 x ”两类.

- 如果组合中不包含 x , 那么相当于从剩余的 $n - 1$ 个元素中选出 k 个作组合, 方案数为 $C(n - 1, k)$;
- 如果组合中包含 x , 那么相当于从剩余的 $n - 1$ 个元素中选出 $k - 1$ 个作组合, 方案数为 $C(n - 1, k - 1)$.

由加法原理, $C(n, k) = C(n - 1, k) + C(n - 1, k - 1)$. \square

上述定理对应着著名的杨辉三角 (或称为 Pascal 三角), 如图 1.1 所示. 杨辉三角是一个无限向下延伸的等腰三角形 (有时也画成左对齐的直角三角形) 数阵, 由全部组合数构成, 其中 $C(n, k)$ 出现在第 $n + 1$ 行的第 $k + 1$ 个位置. 式 (1-11) 指出, 在杨辉三角中, 除了每行的第一个和最后一个数字 (即 $C(n, 0)$ 和 $C(n, n)$) 上方只有一个数字 1 以外, 其余每个组合数都等于它上方的两个组合数之和, 如图 1.1 所示.

从杨辉三角中还能发现一个有趣的事实: 如果用上方一行数字的和来表示某一行所有数字, 例如第五行的 5 个数字:

$$\begin{array}{rcl} 1 & = & 1 \\ 4 & = & 1 + 3 \\ 6 & = & 3 + 3 \\ 4 & = & 3 + 1 \\ 1 & = & 1 \end{array}$$

则上一行的每个数字都恰好被使用了 2 次; 表现在图 1.1 中, 就是每个数字下方都有两个指向下一行数字的箭头. 这表明, 杨辉三角中任何一行数字的和都等于其上一行数字之和

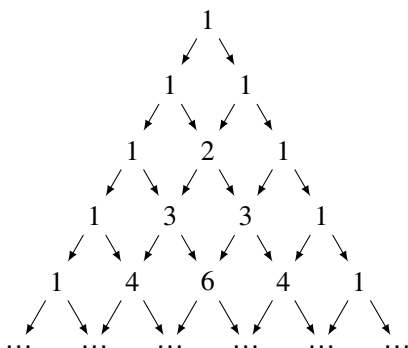


图 1.1: 杨辉三角

的 2 倍. 由于第一行只有一个数字 $C(0, 0) = 1$, 根据数学归纳法, 我们得到:

$$\sum_{k=0}^n C(n, k) = C(n, 0) + C(n, 1) + \cdots + C(n, n) = 2^n \quad (1-12)$$

该式有着直观的组合意义: 在一个大小为 n 的集合 S 中随意选择元素构成子集, 每个元素要么选、要么不选, 并且和是否选择其他元素独立, 因此根据乘法原理, 方案数为 2^n . 另一方面, 子集的大小只可能取 $0, 1, 2, \dots, n$ 这些值, 当子集大小为 k 的时候, 根据组合数的定义, 方案数即为 $C(n, k)$; 考虑子集大小的所有可能性, 总方案数为 $\sum_{k=0}^n C(n, k)$. 这两个值分别是通过不同方式计算同一个方案数得出的结果, 因此二者应该相等, 由此即得到等式 (1-12). 我们将在 1.4 节中给出该结论的一个形式化的证明.

定理 1.16. 设 $n \geq k, m \geq k$, 则

$$C(n, m) \cdot C(m, k) = C(n, k) \cdot C(n - k, m - k) \quad (1-13)$$

证明. 根据乘法原理, 等号左侧相当于从 n 个元素中选出 m 个, 再从这 m 个元素中重新选出 k 个的方法数. 可以换一种选择元素的步骤来达到相同的结果: 首先直接从 n 个元素中选出最终的那 k 个, 方法数为 $C(n, k)$; 然后再从剩余的 $n - k$ 个元素中, 选出之前那些第一次被选中、第二次未被选中的 $m - k$ 个元素, 方法数为 $C(n - k, m - k)$. 由乘法原理, $C(n, m) \cdot C(m, k) = C(n, k) \cdot C(n - k, m - k)$. \square

定理 1.17 (Vandermonde 恒等式). 设 $n \geq k, m \geq k$, 则

$$C(n + m, k) = \sum_{i=0}^k C(n, i) \cdot C(m, n - i) \quad (1-14)$$

证明. 等号左侧相当于从 $n + m$ 个元素中选出 k 个的方法数. 现在将所有 $n + m$ 个元素任意分为两组, 一组 n 个, 另一组 m 个. 假设选出的 k 个元素中有 i 个落入第一组, 其余

$k-i$ 个落入第二组, 则上述情形就等价于在第一组中先选取 i 个元素, 然后在第二组中再选取另外 $k-i$ 个, 最终凑齐 k 个. 根据乘法原理, 方案数为

$$C(n, i) \cdot C(m, k-i)$$

i 可能取 0 到 k 中的任意非负整数值. 由加法原理, 将上式对所有可能的 i 求和, 就得到在所有元素中选出 k 个的方法数. \square

例 1.2.7. 从 26 个英文字母中选出 4 个, 排成一个 4 位的字符串.

- (1) 如果允许字母重复, 求方案数;
- (2) 如果不允许字母重复, 求方案数;
- (3) 如果不允许字母重复的同时, 还要求字母 b 和 d 不能相邻, 求方案数.

这个例子结合使用了基本计数原理和排列组合知识. 第一问中, 因为允许字母重复, 所以字符串的每一位都可以独立地选用 26 个字母之一. 根据乘法原理, 方案数为

$$26^4 = 456976$$

第二问中字母不允许重复, 因此相当于从 26 个字母构成的集合中选出 4 个排成一行, 方案数为

$$P(26, 4) = 26 \times 25 \times 24 \times 23 = 358800$$

第三问直接计数比较困难, 我们可以转而计算字母 b 和 d 相邻的方案数, 然后利用减法原理求解. 若确定选取字母 b 和 d, 则从剩下的字母中再选两个的方案数是 $C(24, 2)$; b 和 d 必须相邻, 因此这两个字母可以当作排列中的一整个元素来对待, 从而排列中共有 3 个元素, 方案数为 $3!$; 最后, b 和 d 两个字母可以排成 bd 也可以排成 db, 有 2 种方案. 三者相乘即为 b 和 d 相邻的字符串的数目. 因此, 满足要求的字符串的数目为

$$P(26, 4) - C(24, 2) \times 3! \times 2 = 355488$$

1.3 球盒模型与格路模型

在解决实际问题时, 如果要从问题中直接提取出集合的排列或组合等抽象概念, 有时可能稍为难以入手. 这时, 我们可先将问题转化为清晰的数学模型, 然后再套用排列或组合解决模型中的问题. 本节中, 我们介绍两种排列组合问题中常用的数学模型.

1.3.1 球盒模型

球盒模型使用小球和盒子来描述组合问题. 经由该模型表述的问题一般是如下形式:

将若干个小球全部放入若干个盒子, 在满足某种条件的前提下, 有多少种不同的放置方案?

球盒模型的优点在于, 它与排列和组合问题存在较好的对应关系, 小球、盒子能够和集合中的元素、元素在排列中的位置等要素建立直观联系. 此外, 该模型相当灵活, 许多类型的组合数学问题都能转化为球盒模型.

例 1.3.1. 集合排列问题中, 需要从 n 个元素中选出 r 个, 然后将其排成一行. 此问题对应于如下的球盒模型: 将 r 个不同的小球放入 n 个不同的盒子, 每个盒子最多放一个小球.

这个例子中, r 个小球对应于一行中的 r 个位置, 盒子则对应于集合中的元素. 将某个小球放入某个盒子中, 相当于是将盒子对应的元素放在了小球对应的位置上.

例 1.3.2. 集合组合问题中, 需要从 n 个元素中选出 r 个, 选出的元素之间没有先后顺序. 此问题对应于如下的球盒模型: 将 r 个相同的小球放入 n 个不同的盒子, 每个盒子最多放一个小球.

这个例子中, r 个小球对应于选出的元素, 盒子则对应于集合中的元素. 将小球放入某个盒子中, 相当于是要从集合中选出盒子对应的元素.

值得一提的是, 以上两个球盒模型均能正确处理 $r > n$ 的情况. 在这种情况下, 我们有 $P(n, r) = C(n, r) = 0$. 另一方面, 在上述两个球盒模型中, 如果球数多于盒数, 那么由鸽巢原理 (在第二章中将详细讨论), 无论如何放球都会出现有盒子中多于一个小球的情况, 因此满足要求的放球方案数目为 0.

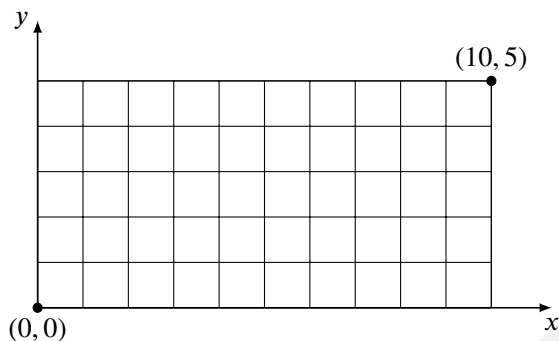
概括地说, 球盒模型适合解决含有“分配”或“对应”等要素的组合问题. 这是因为将小球放入盒子可以视为对小球的一种分配方案, 同时也可以看作构造一种从小球到盒子的对应关系.

1.3.2 格路模型

格路模型使用网格和路径来描述组合问题. 这类问题一般是下面这种形式:

xOy 平面上有一个由整点 (即坐标均为整数的点) 及其连线构成的横平竖直的网格. 要从 $(0, 0)$ 沿网格走到 (m, n) , 每一步只能向右或向上走一个单位长度; 在满足某种条件的前提下, 有多少种不同的路线?

格路模型中因为存在行走方向限制, 所以能走的部分仅限于起点为左下角、终点为右上角的一个矩形范围内. 画出网格时也通常只画这个范围. 图 1.2 即展示了一个从 $(0, 0)$ 到 $(10, 5)$ 的网格.

图 1.2: 一个从 $(0,0)$ 到 $(10,5)$ 的网格

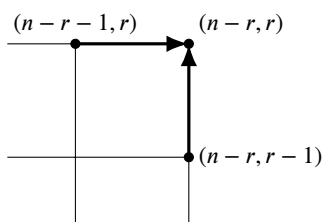
与球盒模型不同，格路模型一般仅限于描述组合问题。这是因为球盒模型中的小球和盒子都可以彼此不同，从而具备表达排列问题中元素顺序关系的能力；但格路模型中每一步则只有向右走和向上走两种情况，一般无法处理排列问题。

例 1.3.3. 集合组合问题中，需要从 n 个元素中选出 r 个，选出的元素之间没有先后顺序。它对应于从 $(0,0)$ 走到 $(n-r, r)$ 的格路模型。

在这个例子中，从 $(0,0)$ 走到 $(n-r, r)$ 一共需要走 n 步。既然每一步都只能向右或向上走一个单位，那么为了从 $y=0$ 走到 $y=r$ ，必须要有恰好 r 步向上走。因此，我们可以让在网格上走的每一步对应集合中的一个元素；如果某一步向上走，则相当于选出这一步对应的元素；如果向右走，则代表不选择这个元素。这样就构成了从 n 个元素中选出 r 的一个组合。

与球盒模型类似，格路模型在这个例子中也能正确处理 $r > n$ 时的边界情况。此时，终点 $(n-r, r)$ 在第四象限，但是从 $(0,0)$ 出发向右或向上走，将永远无法抵达第四象限，因此满足要求的路线数目为 0。

例 1.3.4. 设 $n > r > 0$ 。从 $(0,0)$ 沿网格走到 $(n-r, r)$ ，必定要经过 $(n-r-1, r)$ 或 $(n-r, r-1)$ 这两个点其中之一。从 $(0,0)$ 走到 $(n-r-1, r)$ 的方案数是 $C(n-1, r)$ ，从 $(0,0)$ 走到 $(n-r, r-1)$ 的方案数是 $C(n-1, r-1)$ 。如图 1.3 所示，根据加法原理， $C(n, r) = C(n-1, r) + C(n-1, r-1)$ 。

图 1.3: 格路模型对 $C(n, r) = C(n-1, r) + C(n-1, r-1)$ 的直观解释

例 1.3.5. 从 $(0,0)$ 走到 $(n-r, r)$ 的方案数为 $C(n, r)$. 现在将网格沿直线 $y = x$ 翻转, $(n-r, r)$ 被翻转到 $(r, n-r)$, 从 $(0,0)$ 走到 $(r, n-r)$ 的方案数为 $C(n, n-r)$. 如图 1.4 所示, 翻转前后的合法路线互为镜像、一一对应, 因此以上两个方案数相同, 即 $C(n, r) = C(n, n-r)$.

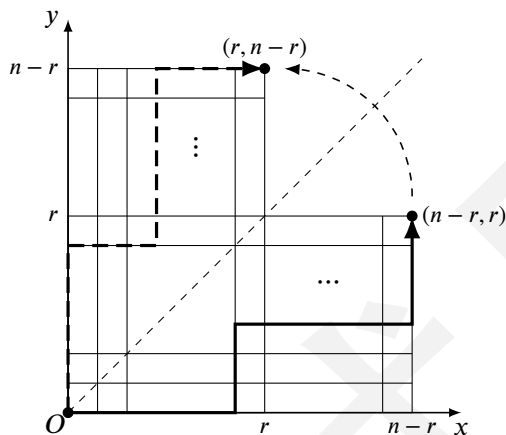


图 1.4: 格路模型对 $C(n, r) = C(n, n-r)$ 的直观解释

例 1.3.6. 设 $n \geq m$. 从 $(0,0)$ 走到 (n, m) 的方案数为 $C(n+m, m)$. 作直线 $x+y = m$, 它与网格交于 $(0, m), (1, m-1), \dots, (m, 0)$ 这 $m+1$ 个格点. 从 $(0,0)$ 走到 (n, m) , 必定要经过上述 $m+1$ 个格点中的一个, 而且也只能经过一个. 如此一来, 根据路径经过的是哪一个格点, 可以将路径分为 $m+1$ 类; 每一类路线中又因为已经确定了经过的一个格点, 可以以那个格点为界将路径分为两段. 利用乘法原理和加法原理, 可知

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

这是 Vandermonde 恒等式 (1-14) 的一个特例.

概括地说, 格路模型适合解决含有“对称”或“递推”等要素的组合问题. 首先, 网格拥有良好的对称性; 对于网格上的任何一条合法路径 (即只向右或向上走的路径), 作它关于直线 $y = x + C$ (其中 C 是任意整数) 的对称, 则仍然能得到一条合法路径, 但起点和终点可能发生改变. 其次, 如果把网格中的每个格点都视为问题的一个状态, 那么如果给定起点走到终点的一条路径, 就可以将其看作从初态到终态的一次转移; 鉴于每个状态都拥有确定的前驱 (左方和下方) 和后继 (右方和上方) 状态, 因此可以非常容易地写出状态之间应满足的递推关系. 最后, 尽管网格的结构相对固定, 但我们可以根据问题需要对其进行修饰, 例如添加对角线或规定不能经过某些格点等, 再灵活运用格路模型的结论解决问题.

1.4 二项式定理与组合恒等式

二项式 (binomial) 指的是仅有两项的多项式, 是仅次于单项式的最简单多项式. 二项式定理描述了二项式的幂展开成多项式的形式, 它由著名的英国数学家、物理学家牛顿 (Issac Newton) 于 1664 年到 1665 年间提出.

二项式定理研究 $(x+y)^n$ 的多项式展开, 其中 n 是任意非负整数. 根据乘幂的定义, 我们知道 $(x+y)^n$ 等于 n 个 $(x+y)$ 相乘:

$$(x+y)^n = \underbrace{(x+y)(x+y)\cdots(x+y)}_{n \text{ 个 } (x+y)}$$

$(x+y)$ 是一个一次的二项式, 并且不含常数项. 因此, 如果将 $(x+y)^n$ 展开为多项式的形式, 则由乘法分配律可知, 结果的每一项中 x 和 y 的幂指数之和一定等于 n . 这即是说, 展开式中系数非零的项只可能是

$$x^n, x^{n-1}y, x^{n-2}y^2, \dots, xy^{n-1}, y^n$$

这 $n+1$ 项, 它们均具有 $x^{n-k}y^k$ 这一共同的形式 ($0 \leq k \leq n$). 用 $\binom{n}{k}$ 表示展开式中 $x^{n-k}y^k$ 这一项的系数, 那么我们就可以写出二项式定理.

定理 1.18 (二项式定理). 设 n 是非负整数, 则

$$\begin{aligned} (x+y)^n &= \binom{n}{0}x^n + \binom{n}{1}x^{n-1}y + \cdots + \binom{n}{n}y^n \\ &= \sum_{k=0}^n \binom{n}{k}x^{n-k}y^k \end{aligned} \quad (1-15)$$

其中 $\binom{n}{k}$ 称为二项式系数 (binomial coefficient).

下面我们求 $\binom{n}{k}$, 也就是 $x^{n-k}y^k$ 的系数. 将 $(x+y)^n$ 展开为 n 个 $(x+y)$ 的连乘积, 根据乘法分配律, 每一个 $(x+y)$ 都会为 $x^{n-k}y^k$ 贡献 x 或 y 的一次幂. $x^{n-k}y^k$ 这一项是由 k 个 $(x+y)$ 中的 y , 以及其余 $n-k$ 个 $(x+y)$ 中的 x 相乘得到的. 因此, $x^{n-k}y^k$ 的系数就等于从 n 个 $(x+y)$ 中选出 k 个并“使用”其中的 y , 然后再“使用”其余 $n-k$ 个 $(x+y)$ 中的 x 的方案数. 这可以抽象为从 n 个元素中选出 k 个的问题, 实际上就是集合组合. 根据组合数的定义, 我们有

$$\binom{n}{k} = C(n, k)$$

换句话说, 二项式系数等价于组合数.

基于这种等价关系, 在组合数学中, 人们也经常将组合数称为二项式系数; $C(n, r)$ 与 $\binom{n}{r}$ 均是组合数的记号, 二者在数学意义上完全等价.

注意

从美观性的角度考虑, 二项式系数记号 $\binom{n}{r}$ 通常比 $C(n, r)$ 显得更漂亮、清晰和简洁.

本书后续将始终使用 $\binom{n}{r}$ 表示组合数, 我们也推荐读者尽可能使用这一记号.

二项式定理构建了一个关于 $n+1$ 个组合数的恒等式, 其中还包含 x 、 y 两个待定系数; 通过为 x 、 y 指定不同的值并代入二项式定理, 我们就能得到不同形式的等式. 因此, 二项式定理成为了推导组合恒等式的绝佳工具.

例 1.4.1. 在式 (1-15) 中令 $x = 1, y = 1$, 得到

$$2^n = \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = \sum_{k=0}^n \binom{n}{k} \quad (1-16)$$

式 (1-16) 就是式 (1-12); 在 1.2 节中我们通过解释组合意义的方法说明了这个等式成立, 而上面这个例子则通过二项式定理为该式给出了严格的证明.

例 1.4.2. 在式 (1-15) 中令 $x = 1, y = -1$, 得到

$$0 = \binom{n}{0} - \binom{n}{1} + \cdots + (-1)^n \binom{n}{n} = \sum_{k=0}^n (-1)^k \binom{n}{k} \quad (1-17)$$

交错和 (alternating sum) 指的是每一项的正负号都和前一项相反的和式. 式 (1-17) 指出组合数的交错和等于 0. 这可以通过组合数的递推性质来解释:

$$\binom{n}{r} = \binom{n-1}{r} + \binom{n-1}{r-1}$$

将其代入式 (1-17) 中, 会发现 $\binom{n-1}{0}$ 到 $\binom{n-1}{n-1}$ 之间的 n 个组合数每个都恰好被加减各一次, 从而相消为 0.

例 1.4.3. 求下列和式:

$$\sum_{k=1}^n k \binom{n}{k}$$

这个例子有两种不同的解决思路. 第一种思路是先对式子进行变形, 化为更容易求和的形式. 式 (1-13) 指出

$$\binom{n}{m} \binom{m}{r} = \binom{n}{r} \binom{n-r}{m-r}$$

因此我们有

$$k \binom{n}{k} = \binom{n}{k} \binom{k}{1} = \binom{n}{1} \binom{n-1}{k-1} = n \binom{n-1}{k-1}$$

这一步变形后，我们就可以直接套用式 (1-16) 进行求和：

$$\sum_{k=1}^n k \binom{n}{k} = \sum_{k=1}^n n \binom{n-1}{k-1} = n \sum_{k=0}^{n-1} \binom{n-1}{k} = n \cdot 2^{n-1}$$

第二种方式基于二项式定理. 二项式定理为求解组合数和式提供了一种明确、直观和机械化的思路. 具体地，我们的目标是对二项式定理进行变形，将等号右侧的展开式部分凑成所要求解的和式，这时等号左侧即是答案. 首先将 $x = 1$ 代入式 (1-15) 中：

$$(1+y)^n = \binom{n}{0} + \binom{n}{1}y + \binom{n}{2}y^2 + \cdots + \binom{n}{n}y^n$$

下一步，我们希望能在右侧每一个 y^k 项前面都乘上 k . 要做到这一点，只需在等号两侧同时对 y 求导：

$$n(1+y)^{n-1} = 0 + \binom{n}{1} + \binom{n}{2} \cdot 2y + \cdots + \binom{n}{n} ny^{n-1}$$

然后，再代入 $y = 1$ ：

$$n \cdot 2^{n-1} = \binom{n}{1} + 2\binom{n}{2} + \cdots + n\binom{n}{n}$$

这时等号右侧即为 $\sum_{k=1}^n k \binom{n}{k}$ ，它即等于等号左侧的 $n \cdot 2^{n-1}$.

本节的最后，我们简单讨论二项式系数 $\binom{n}{k}$ 在 n 是实数时的推广. 这时，它的组合意义可以描述为 $(1+x)^n$ 的展开式中 x^k 项的系数. 当 n 是任意实数时， $(1+x)^n$ 无法直接作多项式展开；为了求解 x^k 项的系数，我们可转而利用 Maclaurin 展开（即 $x = 0$ 处的 Taylor 展开）将其写成关于 x 的幂级数：

$$(1+x)^n = \sum_{k=0}^{\infty} \left[\prod_{j=0}^{k-1} (n-j) \right] \cdot \frac{x^k}{k!}$$

由此可得

$$\binom{n}{k} = \frac{1}{k!} \prod_{j=0}^{k-1} (n-j)$$

这与 n 是非负整数时的定义具有完全相同的形式. 因此，二项式系数的定义可以自然地推广至 n 是实数时的情况. 注意， k 仍然必须是非负整数.

1.5 多重排列与环形排列

本节中我们讨论两类变形的排列问题. 在前面几节中，我们对排列问题的基本假设是：

- 从集合中选出若干元素，集合中的元素彼此不同、没有重复；
- 将选出的元素排成有序的一行。

多重排列与**环形排列**分别针对这两个假设作出推广。具体地，多重排列讨论多重集中元素的排列问题，环形排列则将选出的元素排成一个圆环，允许进行旋转或翻转操作得到等价的环形排列。

1.5.1 多重排列

本小节讨论的多重排列指的是多重集的全排列。**多重集**（multiset）即允许包含多个相同元素的集合。我们首先简单介绍多重集与相关的概念；其中有些内容在本小节中虽然不会涉及，但在后面的章节中需要用到。

定义 1.19 (多重集). 设 $n \in \mathbb{Z}^* \cup \{\infty\}$ ，称

$$M = \{a_1 \cdot x_1, a_2 \cdot x_2, \dots, a_n \cdot x_n\} \quad (1-18)$$

为多重集，其中

$$x_1, x_2, \dots, x_n$$

为 M 中的元素种类，

$$a_1, a_2, \dots, a_n \in \mathbb{Z}^* \cup \{\infty\}$$

表示 M 中元素的数量。称上述多重集 M 包含 a_1 个元素 x_1 、 a_2 个元素 x_2 、……、 a_n 个元素 x_n ；若 $a_k = \infty$ ，则表示 M 中包含无限多个元素 x_k 。

多重集 M 的大小定义为

$$|M| = \sum_{i=1}^n a_i \quad (1-19)$$

若 $|M| < \infty$ ，称 M 是有限多重集，否则称 M 是无限多重集。

非负整数集 \mathbb{Z}^* 上的加减运算和大小关系可以自然地扩展到 $\mathbb{Z}^* \cup \{\infty\}$ 上，由此可以定义多重集的子集和补集：

定义 1.20 (多重集的子集). 设

$$M = \{a_1 \cdot x_1, a_2 \cdot x_2, \dots, a_n \cdot x_n\}$$

为多重集。若

$$b_1, b_2, \dots, b_n \in \mathbb{Z}^* \cup \{\infty\}$$

且

$$b_1 \leq a_1, b_2 \leq a_2, \dots, b_n \leq a_n$$

则称多重集

$$A = \{b_1 \cdot x_1, b_2 \cdot x_2, \dots, b_n \cdot x_n\}$$

是多重集 M 的子集。 M 的全体子集组成的集合称为 M 的幂集，记作 $P(M)$ 。

定义 1.21 (多重集的补集). 设

$$M = \{a_1 \cdot x_1, a_2 \cdot x_2, \dots, a_n \cdot x_n\}$$

为有限多重集,

$$A = \{b_1 \cdot x_1, b_2 \cdot x_2, \dots, b_n \cdot x_n\}$$

是多重集 M 的子集, 则称多重集

$$M - A = \{(a_1 - b_1) \cdot x_1, (a_2 - b_2) \cdot x_2, \dots, (a_n - b_n) \cdot x_n\}$$

为 A 在 M 中的补集.

在有了多重集的概念后, 下面我们讨论多重排列问题. 此类问题的基本形式如下:

给定一个有限多重集, 问其中的元素有多少种排列成一行的方式? 在进行排列时, 相同的元素之间没有顺序.

例 1.5.1. 考虑多重集

$$M = \{2 \cdot a, 1 \cdot b, 1 \cdot c\}$$

它有以下 12 种不同排列方式:

$aabc, aacb, abac, acab, abca, acba,$

$baac, caab, baca, caba, bcaa, cbaa$

下面我们讨论如何求解多重排列方案数. 考虑一个一般形式的有限多重集

$$M = \{a_1 \cdot x_1, a_2 \cdot x_2, \dots, a_n \cdot x_n\} \quad (a_1, a_2, \dots, a_n > 0)$$

其大小记为 $N = |M|$. 此多重集对应的多重排列数记为 $P(N; a_1, a_2, \dots, a_n)$.

M 的多重排列将其中的 N 个元素排成一行. 为此, 想象先留出一行 N 个空位, 然后依次将 M 中的每一类元素填入空位中:

- 第一步, 从 N 个位置中选出 a_1 个, 放置元素 x_1 , 有 $\binom{N}{a_1}$ 种方案;
- 第二步, 从余下的 $N - a_1$ 个位置中选出 a_2 个, 放置元素 x_2 , 有 $\binom{N - a_1}{a_2}$ 种方案;
-
- 第 $n - 1$ 步, 从余下的 $N - a_1 - a_2 - \dots - a_{n-2}$ 个位置中选出 a_{n-1} 个, 放置元素 x_{n-1} , 有 $\binom{N - a_1 - a_2 - \dots - a_{n-2}}{a_{n-1}}$ 种方案;
- 第 n 步, 恰好余下 $N - a_1 - a_2 - \dots - a_{n-1} = a_n$ 个位置, 全部用于放置元素 x_n , 有 $\binom{N - a_1 - a_2 - \dots - a_{n-1}}{a_n} = 1$ 种方案.

上述 n 个步骤彼此独立, 结果互不影响, 因此由乘法原理, 所有步骤的方案数之积就是总的多重排列方案数. 因此,

$$\begin{aligned}
 P(N; a_1, a_2, \dots, a_n) &= \binom{N}{a_1} \binom{N-a_1}{a_2} \dots \binom{N-a_1-a_2-\dots-a_{n-1}}{a_n} \\
 &= \frac{N!}{a_1! (N-a_1)!} \frac{(N-a_1)!}{a_2! (N-a_1-a_2)!} \dots \frac{(N-a_1-a_2-\dots-a_{n-1})!}{a_n! 0!} \\
 &= \frac{N!}{a_1! a_2! \dots a_n!} \quad (1-20)
 \end{aligned}$$

上述结果也可以用除法原理解释: N 个元素的全排列有 $P(N, N) = N!$ 种方案, 但是同类元素之间没有区别; 分别考虑每一类元素, 元素 x_k ($k = 1, 2, \dots, n$) 有 a_k 个, 保持其他元素不变、交换这些元素的顺序得到的 $a_k!$ 个排列是相同的, 因此最终结果应当除以 $a_k!$.

注意到当 $n = 2$ 时, 多重排列数即变为组合数. 组合数与二项式系数等价; 类似地, 多重排列数也与多项式系数等价:

定理 1.22 (多项式定理). 设 n 是正整数, N 是非负整数, 则

$$(x_1 + x_2 + \dots + x_n)^N = \sum_{a_1+a_2+\dots+a_n=N} \binom{N}{a_1 \ a_2 \ \dots \ a_n} x_1^{a_1} x_2^{a_2} \dots x_n^{a_n} \quad (1-21)$$

讨论乘出 $x_1^{a_1} x_2^{a_2} \dots x_n^{a_n}$ 这一项的方法数, 可知多项式系数即为多重排列数:

$$\binom{N}{a_1 \ a_2 \ \dots \ a_n} = P(N; a_1, a_2, \dots, a_n) = \frac{n!}{a_1! a_2! \dots a_n!}$$

例 1.5.2. 英文单词 COMBINATORICS 中包含 13 个字母, 其中 C、O、I 分别出现两次, 其余 7 个字母均只出现一次. 因此, 这些字母一共有

$$\frac{13!}{2! \times 2! \times 2!} = 778377600$$

种不同的排列方式.

1.5.2 环状排列

环状排列指的是从集合中选出若干元素, 并将其排列成一个圆环. 与排列成一行不同, 圆环没有首尾之分, 经由旋转 (或翻转) 可以得到等价的环状排列.

圆排列 (necklace) 指的是仅允许旋转操作的环状排列. 所谓旋转指的是圆环在其所在的平面上绕圆心旋转.

例 1.5.3. a, b, c 这三个元素只有两种不同的圆排列:



任何其他的圆排列方案都能通过上面两个圆排列旋转得到.

将圆排列从某个位置处切开, 可以将其展开成一个线排列. 一个由 r 个元素构成的全排列可以从 r 个位置切开, 得到 r 个不同的线排列. 另一方面, 从 n 个元素中选出 r 个作线排列, 有 $P(n, r)$ 种方案. 因为每一个圆排列都对应 r 个线排列, 所以圆排列数为

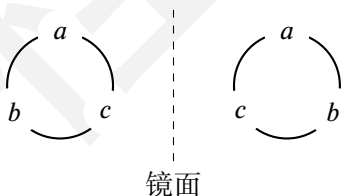
$$\frac{P(n, r)}{r} \quad (1-22)$$

例 1.5.4. 要将 12 道不同的菜式摆在圆饭桌中央的转盘上. 因为转盘可以旋转, 所以这可看作 $n = r = 12$ 的圆排列, 方案数为

$$\frac{P(12, 12)}{12} = 11!$$

手镯排列 (bracelet) 指的是允许旋转和翻转操作的环状排列. 翻转操作也可以理解为镜像, 这就如同一只左手上的手镯和右手上的手镯互为镜像.

例 1.5.5. a, b, c 这三个元素只有一种手镯排列. 这是因为它们的圆排列只有两种, 而且互为镜像, 因此在允许翻转的情况下, 两种圆排列实际上对应于同一种手镯排列.



手镯排列的数量可以使用圆排列来计算. 如同上例中展示的那样, 一对互为镜像的圆排列对应于同一个手镯排列. 因此, 手镯排列的数量恰好是相同参数下圆排列数的一半:

$$\frac{P(n, r)}{2r} \quad (r \geq 3) \quad (1-23)$$

上式要求选出的元素不能少于 3 个. 如果 2 个以下元素排列成圆环, 那么此时是否允许翻转操作没有任何区别, 这时手镯排列数就等于圆排列数.

需要注意, 与多重排列不同, 环状排列中不允许包含相同的元素. 否则, 一个长度为 r 的圆排列从不同位置切开后, 将不能保证得到 r 个不同的线排列, 从而式 (1-22) 和 (1-23) 也就不再适用; 我们将在第六、七章中讨论可重圆排列问题的解决方式.

1.6 可重组与不相邻组合

本节中, 我们讨论两类变形的组合问题. 在前面几节中, 我们对组合问题的基本假设是:

- 从集合中选出若干元素, 集合中的元素彼此不同、没有重复;
- 选出元素时不受额外约束.

可重组与不相邻组合分别针对这两个假设作出推广. 具体地, 可重组讨论多重集中元素的组合问题, 不相邻组合则要求选出的元素在原先的集合中不能相邻.

1.6.1 可重组

可重组问题的基本形式如下:

给定一多重集, 包含有限种元素, 每种元素的数量均为无限多, 问其有多少个大小为 r 的子集?

此类问题可以认为是从 n 个种类的元素中选出 r 个, 方案数记为 $\overline{C}(n, r)$.

例 1.6.1. 考虑多重集

$$M = \{\infty \cdot a, \infty \cdot b, \infty \cdot c\}$$

它有 $\overline{C}(3, 2) = 6$ 个不同的大小为 2 的子集:

$$\{aa\}, \{bb\}, \{cc\}, \{ab\}, \{ac\}, \{bc\}$$

有很多种方法可用来计算可重组数, 我们下面介绍其中三种.

方法一. 第一种方法的核心思路是将可重组转化为普通的组合. 设给定的多重集为

$$M = \{\infty \cdot x_1, \infty \cdot x_2, \dots, \infty \cdot x_n\}$$

考虑 M 的一个大小为 r 的子集, 我们按照下标升序依次列出其中的所有元素. 下面是一个 $n = 4, r = 5$ 时的例子:

$$\{x_1, x_1, x_3, x_3, x_4\}$$

关注这 r 个元素的下标构成的单调递增数列 $\{a_k\} (k = 1, 2, \dots, r)$, 并考虑数列 $\{k + a_k - 1\}$:

$$\begin{array}{cccccc} k = & 1 & 2 & 3 & 4 & 5 \\ a_k = & 1 & 1 & 3 & 3 & 4 \\ \hline k + a_k - 1 = & 1 & 2 & 5 & 6 & 8 \end{array}$$

注意到 $\{k + a_k - 1\}$ 中没有相同的元素. 下面我们证明 $\{k + a_k - 1\}$ 实际上可视为一个集合组合, 并且可重组 $\{a_k\}$ 与其有一一对应关系.

推论 1.23. 由上述方式构造的数列 $\{k + a_k - 1\}$ 可视为从一大小为 $n + r - 1$ 的集合中选择 r 个元素的组合.

证明. 首先, 我们有 $1 \leq k + a_k - 1 \leq n + r - 1$. 这是因为 $k + a_k - 1$ 在 $k = 1, a_0 = 1$ 时取到最小值 1, 在 $k = r, a_r = n$ 时取到最大值 $n + r - 1$.

其次, 数列 $\{k + a_k - 1\}$ 严格递增. 考虑此数列中任意的相邻两项 $k + a_k - 1$ 和 $(k + 1) + a_{k+1} - 1$: 由于 $\{a_k\}$ 单调递增, 因此 $a_k \leq a_{k+1}$; 又因为 $k < k + 1$, 相加可得

$$k + a_k - 1 < (k + 1) + a_{k+1} - 1$$

这表明数列 $\{k + a_k - 1\}$ 严格递增, 从而其中不含相同的元素.

综上, $\{k + a_k - 1\}$ 中的元素均为 1 到 $n + r - 1$ 之间的正整数, 并且互不相同. 由于此数列有 r 项, 因此它可视为从 $n + r - 1$ 个元素中选择 r 个的集合组合. \square

推论 1.24. 任何一个从大小为 $n + r - 1$ 的集合中选择 r 个元素的组合, 均对应于多重集 M 的一个大小为 r 的可重组.

证明. 不妨设集合中的元素为

$$\{1, 2, 3, \dots, (n + r - 1)\}$$

则它的一个 r 元素组合可写成一个严格递增数列 $\{b_k\}$:

$$\{b_1, b_2, \dots, b_r\} \quad (1 \leq b_1 < b_2 < \dots < b_r \leq n + r - 1)$$

构造数列 $\{b_k - k + 1\}$. 首先, 由于 $\{b_k\}$ 严格递增, 并且其中的元素均为 1 到 $n + r - 1$ 间的正整数, 因此

$$k \leq b_k \leq n + k - 1 \quad (k = 1, 2, \dots, r)$$

其次, 比较数列 $\{b_k - k + 1\}$ 中任意的相邻两项 $b_k - k + 1$ 和 $b_{k+1} - k$:

$$\begin{aligned} (b_{k+1} - k) - (b_k - k + 1) &= b_{k+1} - b_k - [k - (k - 1)] \\ &= b_{k+1} - (b_k + 1) \\ &\geq 0 \end{aligned}$$

这表明数列 $\{b_k - k + 1\}$ 是单调递增的. 例如, 当 $n = 4, r = 5$ 时, 一个大小为 8 的集合的 5 元素组合

$$\{1, 2, 5, 6, 8\}$$

对应于可重组 $\{x_1, x_1, x_3, x_3, x_4\}$:

$$\begin{array}{rcccccc} b_k & = & 1 & 2 & 5 & 6 & 8 \\ k & = & 1 & 2 & 3 & 4 & 5 \\ \hline b_k - k + 1 & = & 1 & 1 & 3 & 3 & 4 \end{array}$$

综上, $\{b_k - k + 1\}$ 中的元素取值范围为 1 到 n 之间的正整数, 并且数列单调递增. 这个数列有 r 项, 因此它可视为包含 n 类元素的多重集 M 的一个大小为 r 个的可重组. \square

以上两个推论表明, M 的任何一个 r 元素可重组均对应一个大小为 $n + r - 1$ 的集合的 r 元素组合, 并且这种对应关系反过来也成立. 由此可知, 多重集 M 的 r 元素可重组与大小为 $n + r - 1$ 的集合的 r 元素组合之间存在一一对应关系; 这表明二者的数量应该相等, 从而我们能够利用组合数计算可重组数.

定理 1.25. 多重集

$$M = \{\infty \cdot x_1, \infty \cdot x_2, \dots, \infty \cdot x_n\}$$

的 r 元素可重组数为

$$\overline{C}(n, r) = \binom{n+r-1}{r} \quad (1-24)$$

方法二. 第二种方法基于球盒模型. 将 r 个要选出的元素视为 r 个小球, 将多重集 M 中的 n 类元素视为 n 个不同的盒子; 将一个小球放入某个盒子, 对应于在可重组中选出一个此类元素. 由此, 可重组问题抽象为如下球盒模型:

将 r 个相同的小球放入 n 个不同的盒子, 有多少种不同的放置方案?

想象把所有的盒子紧贴着按顺序排成一行. 这时, 相邻盒子间的盒壁可以视为将小球分隔在两侧的隔板. 我们仍以 $n = 4, r = 5$ 的这个可重组为例:

$$\{x_1, x_1, x_3, x_3, x_4\}$$

这个可重组对应于如图 1.5 所示的球盒模型. 保持小球的位置不变, 将两个盒子之间的盒壁改为隔板, 我们就能得到一个由小球和隔板构成的序列, 这个序列同样展示在图 1.5 中.

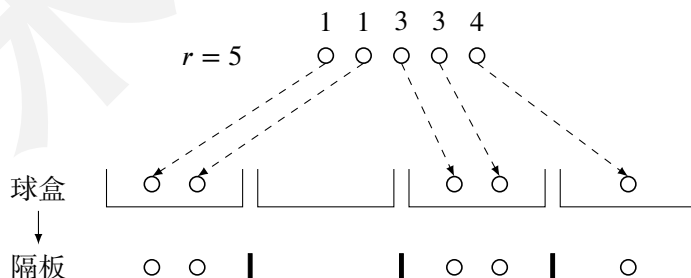


图 1.5: 可重组的球盒模型

上述序列中一共有 r 个小球和 $n - 1$ 个隔板, 总长度为 $n + r - 1$. 容易验证, 这样的小球-隔板序列与合法的放球方案之间存在一一对应关系. 因此, 多重集 M 的 r 元素可重组

合数目, 应等于由 r 个小球和 $n-1$ 个隔板构成的序列数目; 后者可以视为在 $n+r-1$ 个位置中选择 r 个放入小球, 其余的位置则放上隔板, 因此可以直接由集合组合计算出方案数:

$$\overline{C}(n, r) = \binom{n+r-1}{r}$$

这个结果与式 (1-24) 相同.

上面这种方法将“隔板”也作为一个元素, 让它和小球一起参与多重排列, 从而计算出可重组合数. 这种方法一般被称为隔板法.

方法三. 第三种方法基于格路模型, 也可以看作隔板法的一个变种.

考虑一个起点为 $(0, 0)$ 、终点为 $(r, n-1)$ 的格路模型. 在 $(0, 0)$ 为左下角、 $(r, n-1)$ 为右上角的矩形中, 共有 n 条横向的网格线. 我们令位于

$$y = y_0 \quad (y_0 = 0, 1, 2, \dots, n-1)$$

的网格线对应元素 x_{y_0+1} . 构造一条从起点 $(0, 0)$ 到终点 $(r, n-1)$ 的合法路径, 此路径在 $y = y_0$ 上向右走了多少个单位长度, 就在可重组合中选出多少个元素 x_{y_0+1} .

仍以 $n = 4, r = 5$ 时的这个可重组合为例:

$$\{x_1, x_1, x_3, x_3, x_4\}$$

这个可重组合对应于如图 1.6 所示的格路模型路径. 该路径在 $y = 0$ 和 $y = 2$ 上分别前进了 2 个单位长度, 在 $y = 3$ 上前进了 1 个单位长度, 对应于可重组合中的 2 个 x_1 、2 个 x_3 和 1 个 x_4 .

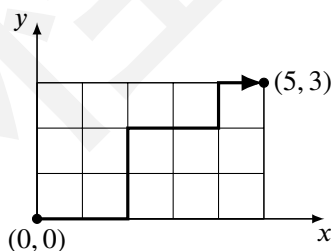


图 1.6: 可重组合的格路模型

容易验证, 可重组合方案与上述网格上的合法路径存在一一对应关系. 因此, 二者的方案数相同, 而我们已经知道如何计算格路模型上的路径数目:

$$\overline{C}(n, r) = \binom{n+r-1}{n-1} = \binom{n+r-1}{r}$$

格路模型的解法本质上和隔板法没有区别. 一条合法路径包含 $n+r-1$ 步, 其中有 $n-1$ 步向上走, 其余 r 步向右走; “向右走”对应于隔板法中的小球, 而“向上走”则对应于隔板.

例 1.6.2. 求方程 $x_1 + x_2 + x_3 + x_4 = 9$ 有多少组不同的非负整数解.

这个例子展示了可重组合的一个经典应用, 即线性方程的非负整数解问题. 线性方程的非负整数解和可重组合之间有着直观的一一对应关系. 例如, 上述方程的一组非负整数解

$$x_1 = 2, x_2 = 4, x_3 = 0, x_4 = 3$$

对应于如下 $n = 4, r = 9$ 的可重组合:

$$\{2 \cdot \hat{x}_1, 4 \cdot \hat{x}_2, 3 \cdot \hat{x}_4\}$$

即, 非负整数解中 x_k 的取值对应于可重组合中元素 \hat{x}_k 的数量.

利用这种对应关系, 我们可以通过可重组合数计算非负整数解的组数. 在这个例子中, 方程有

$$\binom{4+9-1}{9} = \binom{12}{3} = 220$$

组非负整数解.

例 1.6.3. 公园大门处有 6 个可供游客通过的闸机. 9 位游客在开门前来到公园前, 各自选择了一个闸机并在它前面排队. 求不同的排队方案数.

这个例子结合了可重组合和排列. 先不考虑 9 位游客之间的区别, 将他们分配到 6 个闸机前, 这是一个可重组合问题, 其方案数是

$$\binom{6+9-1}{9} = \binom{14}{9}$$

再将游客之间的区别纳入考虑. 想象将每个闸机与它前面排队的游客组合起来, 从左到右依次排成一行; 保持每个闸机前的人数不变, 游客之间可以自由交换位置, 产生 $9!$ 种不同的排列. 因此, 方案数也应乘以 $9!$:

$$\binom{14}{9} \times 9! = P(14, 9) = 726485760$$

也可以使用隔板法理解这个例子. 将闸机理解为隔板, 将游客理解为隔板间互不相同的小球, 则问题转化为一个多重排列问题, 方案数为

$$\frac{(5+9)!}{5! \cdot \prod_{k=1}^9 1!} = \frac{14!}{5!} = P(14, 9)$$

上面讨论的可重组合要求多重集 M 中的每种元素都有无限多个, 这使我们能在构造的多重组合中使用任意合法数量的任意种类元素. 如果 M 中某种元素的数量是有限个, 则上述条件就不再满足, 可重组合数 $\overline{C}(n, r)$ 也不再适用; 我们将在第六章中讨论如何使用容斥原理处理这种情况.

1.6.2 不相邻组合

不相邻组合指的是从一个线性有序的有限集合中选出若干元素，要求在此顺序下相邻的两个元素不能同时选出。例如，从一个由连续整数构成的集合中选出元素，要求选出的整数两两之间至少相差 2。

例 1.6.4. 从 $S = \{1, 2, 3, 4\}$ 中选出两个不相邻的元素，有 3 种方案：

$$\{1, 3\}, \{1, 4\}, \{2, 4\}$$

如果要选出三个不相邻的元素，那么方案数为 0，这是因为对于 $\{1, 2\}$ 和 $\{3, 4\}$ 这两组元素而言，总有一组中的两个元素要被同时选出，导致违反元素不相邻的要求。

与可重组合同样，不相邻组合也可以转化为普通的集合组合进行计数。为了方便起见，我们用 $1, 2, \dots, n$ 这 n 个正整数为每个元素编号。一个 r 元素不相邻组合可以看作一组 1 到 n 之间的正整数，将它们按升序排列后，相邻两项之间的差不小于 2。下面是一个 $n = 9, r = 5$ 的可重组的例子：

$$1, 3, 5, 7, 9$$

将其视为一个严格递增数列 $\{a_k\}$ 。与可重组合同样，我们计算数列 $\{a_k - k + 1\}$ ：

$$\begin{array}{rcccccc} a_k = & 1 & 3 & 5 & 7 & 9 \\ k = & 1 & 2 & 3 & 4 & 5 \\ \hline a_k - k + 1 = & 1 & 2 & 3 & 4 & 5 \end{array}$$

容易证明，数列 $\{a_k - k + 1\}$ 可视为一大小为 $n - r + 1$ 的集合中的 r 元素组合，并且这一对应关系反过来也成立；因此，二者间存在一一对应关系，方案数相同。由此可得不相邻组合的方案数：

$$\binom{n - r + 1}{r} \quad (1-25)$$

1.7 生成全排列

前面我们已经讨论了如何对排列和组合进行计数。从本节开始，我们讨论如何具体列出每一个满足要求的排列或组合。我们希望能够给出排列与组合的生成算法；通过这些算法，计算机可以自动化地生成排列与组合。

本节中，我们讨论几种常用的生成全排列的算法：给定一个大小为 n 的集合，算法按某种顺序输出集合中元素的全部 $n!$ 个全排列。在下面的讨论中，为了方便起见，我们假设此集合为

$$\{1, 2, 3, \dots, n\}$$

1.7.1 Stirling 近似公式

在介绍全排列生成算法前，我们先简单讨论这些算法的最低计算复杂度。一个 n 元素集共有 $n!$ 种全排列方式，每个排列均包含 n 个元素，因此生成这些排列的计算复杂度至少是 $\Omega(n \cdot n!)$ 。当 n 很大时，可使用 Stirling 近似公式对 $n!$ 的值进行估计。

定理 1.26 (Stirling 近似公式).

$$n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \quad (1-26)$$

其中，记号“ \sim ”的意义如下： $a_n \sim b_n$ 等价于 $\lim_{n \rightarrow \infty} \frac{a_n}{b_n} = 1$ 。

Stirling 近似公式指出，任何全排列生成算法的计算复杂度都是 $\Omega\left(\frac{n^{n+\frac{1}{2}}}{e^n}\right)$ 级别的。

需要指出的是，Stirling 近似公式给出的是比值意义上的近似。随着 $n \rightarrow \infty$ ，近似结果与 $n!$ 真实值之间的绝对差距实际上是发散的，即

$$\lim_{n \rightarrow \infty} n! - \sqrt{2\pi n} \left(\frac{n}{e}\right)^n = \infty$$

Stirling 近似公式的完整证明对本书而言过于复杂，我们在此仅作一些简化的讨论。注意到，当 n 是正整数时，有

$$\ln n! = \sum_{k=1}^n \ln k$$

利用积分对求和进行近似，我们有

$$\int_1^n \ln x dx \leq \sum_{k=1}^n \ln k \leq \int_2^{n+1} \ln x dx$$

即

$$n \ln n - n + 1 \leq \ln n! \leq (n+1) \ln(n+1) - (n+1) - 2 \ln 2 + 2 \quad (1-27)$$

先考虑式 (1-27) 左侧的不等号，它给出了 $\ln n!$ 的下界。对此不等号左右两侧分别求指数，得到

$$n! \geq e^{n \ln n - n + 1} = n^n e^{-n+1} = e \cdot \left(\frac{n}{e}\right)^n$$

再考虑式 (1-27) 右侧的不等号，它给出了 $\ln n!$ 的上界。对此不等号左右两侧分别求指数，

得到

$$\begin{aligned}
 n! &\leq e^{(n+1)\ln(n+1)-(n+1)-2\ln 2+2} \\
 &= \frac{e}{4} \cdot e^{(n+1)\ln(n+1)-n} \\
 &\leq \frac{e}{4} \cdot e^{(n+1)\left(\ln n + \frac{1}{n}\right)-n} \\
 &= \frac{e}{4} \cdot n^{n+1} e^{1+\frac{1}{n}-n} \\
 &= \frac{ne^{2+\frac{1}{n}}}{4} \cdot \left(\frac{n}{e}\right)^n
 \end{aligned}$$

综上, 我们有

$$e \cdot \left(\frac{n}{e}\right)^n \leq n! \leq \frac{ne^{2+\frac{1}{n}}}{4} \cdot \left(\frac{n}{e}\right)^n$$

利用渐进记号, 我们可得到 $n! = \Omega\left(\frac{n^n}{e^n}\right)$ 和 $n! = O\left(\frac{n^{n+1}}{e^n}\right)$.

1.7.2 字典序法

字典序是一种常见的对字符串进行排序的方式. 所谓字典序, 指的就是“像字典中一样的排序”; 在字典序下, 两个相同长度的排列可以按以下方式比较大小: 从第一个元素开始, 依次比较排列每一个位置上的元素, 在遇到第一处不同时, 按那个位置上元素的大小关系决定两个全排列的大小关系. n 个元素构成的全排列可以视为 n 位字符串, 因此也可按字典序排序. 例如, 在下面两个 1 到 9 的排列中, 第一处不同发生在第 5 个位置, 比较这个位置上的元素可知, 前一个排列在字典序下小于后一个排列:

$$8\ 3\ 9\ 6\ \boxed{4}\ 7\ 5\ 2\ 1 < 8\ 3\ 9\ 6\ \boxed{5}\ 1\ 2\ 7\ 4$$

容易看出, 在字典序中, 连续的排列的前缀会尽可能保持不变; 直到排列的后缀已经遍历了所有可能的排列方式时, 前缀才会发生变化. 例如, 可以观察 $n = 4$ 时按字典序排序的 24 个全排列 (第一列为第 1 至 6 个、第二列为第 7 至 12 个, 以此类推):

$$\begin{array}{cccc}
 1\ 2\ 3\ 4 & 2\ 1\ 3\ 4 & 3\ 1\ 2\ 4 & 4\ 1\ 2\ 3 \\
 1\ 2\ 4\ 3 & 2\ 1\ 4\ 3 & 3\ 1\ 4\ 2 & 4\ 1\ 3\ 2 \\
 1\ 3\ 2\ 4 & 2\ 3\ 1\ 4 & 3\ 2\ 1\ 4 & 4\ 2\ 1\ 3 \\
 1\ 3\ 4\ 2 & 2\ 3\ 4\ 1 & 3\ 2\ 4\ 1 & 4\ 2\ 3\ 1 \\
 1\ 4\ 2\ 3 & 2\ 4\ 1\ 3 & 3\ 4\ 1\ 2 & 4\ 3\ 1\ 2 \\
 1\ 4\ 3\ 2 & 2\ 4\ 3\ 1 & 3\ 4\ 2\ 1 & 4\ 3\ 2\ 1
 \end{array}$$

可以看到, 24 个排列可以分为 4 组, 每组 6 个排列, 这些排列中的第一个元素都相同; 直到它后面的 3 个元素已经用完了所有可能的 6 种排列方式, 第一个元素才会发生改变.

这种性质启示我们通过寻找可继续变化的最短后缀, 从而构造一种生成 n 个元素的字典序全排列 $p_1 p_2 \cdots p_n$ 的算法. 算法的具体描述如下:

- (1) 算法从 $p_1 p_2 \cdots p_n = 12 \cdots n$ 开始;
- (2) 寻找最短的可改变的后缀 $p_j p_{j+1} \cdots p_n$, 其中

$$j = \max \{i \mid p_i < p_{i+1}\}$$

如果找不到这样的 j , 说明已经找到了所有的排列, 算法结束;

- (3) 寻找后缀中比 p_j 大的最小的元素. 在上一步中找到的后缀中, 在 j 后面的元素都已经是降序排列, 因此这一步实际上就是寻找位置最靠后的比 p_j 大的元素;

$$k = \arg \min_i \{p_i \mid i > j \text{ 且 } p_i > p_j\}$$

- (4) 交换 p_j 和 p_k , 然后反转 $p_{j+1} p_{j+2} \cdots p_n$;
- (5) 回到第 (2) 步.

上面的算法是一个循环, 其中的 (2) 到 (4) 步用于从当前排列生成下一个排列. 我们通过下面这个例子展示字典序算法的具体运作流程.

例 1.7.1. 求 839647521 在字典序下的下一个排列.

首先, 寻找最短的可改变的后缀, 这个后缀出现在 $j = 5$:

$$8\ 3\ 9\ 6\ 4 < 7 > 5 > 2 > 1$$

下一步, 寻找后缀中比 $p_5 = 4$ 大的最小元素, 这个元素出现在 $k = 7$, $p_7 = 5$:

$$8\ 3\ 9\ 6\ \boxed{4}\ 7\ \boxed{5}\ 2\ 1$$

下一步, 交换 p_5 与 p_7 , 然后反转 $p_6 p_7 p_8 p_9$:

$$8\ 3\ 9\ 6\ 5\ 1\ 2\ 4\ 7$$

这就是 839647521 在字典序下的下一个排列.

C++ 语言的标准模板库 (STL) 提供 `next_permutation` 函数, 能够求出给定排列在字典序下的下一个排列. 它还同时提供 `prev_permutation` 函数, 能够求出给定排列在字典序下的上一个排列.

下面我们来讨论如何将排列与它在字典序中的序号对应起来. 所谓序号就是指给定的排列在全排列中的次序, 或者说是有多少个排列小于给定的排列; 序号均为非负整数, 从 0 开始, 直到 $n! - 1$ 结束. 例如, 当 $n = 4$ 时, 排列 3241 的序号是 15, 因为它在字典序中排第 16 位.

在字典序下, 我们可按照前缀对小于一个给定排列的所有排列进行分类, 分别计算出每一类中的排列数目并相加, 即可得到给定排列的序号.

例 1.7.2. 设 $n = 4$, 给定的排列为 3241. 先于此排列的排列可按前缀分为 3 类:

- 第一位小于 3 的排列，数量为： $2 \times 3!$ ；
- 第一位等于 3、第二位小于 1 的排列，数量为： $1 \times 2!$ ；
- 前两位等于 32、第三位小于 4 的排列，数量为： $1 \times 1!$ 。

因此，上述排列的序号为

$$2 \times 3! + 1 \times 2! + 1 \times 1! = 12 + 2 + 1 = 15$$

例 1.7.3. 设 $n = 9$ ，给定的排列为 839647521。先于此排列的排列可按前缀分为 8 类：

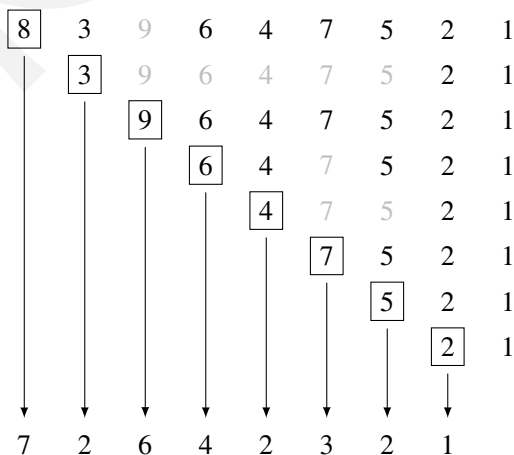
- 第一位小于 8 的排列，数量为： $7 \times 8!$ ；
- 第一位等于 8、第二位小于 3 的排列，数量为： $2 \times 7!$ ；
- 前两位等于 83、第三位小于 9 的排列，数量为： $6 \times 6!$ ；
- 前三位等于 839、第四位小于 6 的排列，数量为： $4 \times 5!$ ；
- 前四位等于 8396、第五位小于 4 的排列，数量为： $2 \times 4!$ ；
- 前五位等于 83964、第六位小于 7 的排列，数量为： $3 \times 3!$ ；
- 前六位等于 839647、第七位小于 5 的排列，数量为： $2 \times 2!$ ；
- 前七位等于 8396475、第八位小于 2 的排列，数量为： $1 \times 1!$ ；

因此，上述排列的序号为

$$7 \times 8! + 2 \times 7! + 6 \times 6! + 4 \times 5! + 2 \times 4! + 3 \times 3! + 2 \times 2! + 1 \times 1! = 297191$$

在上个例子中，我们可将计算过程中 $8!, 7!, \dots, 1!$ 前的系数抽出并连在一起，得到 72642321。此数字串称为排列的**中介数**，它是排列及其序号之间的桥梁。

下面考察中介数每一位的实际含义。例如，72642321 这个中介数从左向右的第三位是 6；由例 1.7.3 可知，这个 6 的意义是在排列前两位固定为 83 前提下，排列中位于第三个位置上的元素（即 9）右侧且比它更小的元素个数。



例 1.7.4. 由字典序中介数 72642321 计算出对应的排列. 中介数有 8 位, 因此是 9 个元素的排列 $p_1 p_2 \cdots p_9$.

- $7 + 1 = 8 \Rightarrow p_1 = 8$;
- $2 + 1 = 3 \Rightarrow p_2 = 3$;
- $6 + 1 = 7$, 但小于 7 的 3 已在左侧出现; $7 + 1 = 8$, 但 8 已被使用; $8 + 1 = 9 \Rightarrow p_3 = 9$;
- $4 + 1 = 5$, 但小于 5 的 3 已在左侧出现; $5 + 1 = 6 \Rightarrow p_4 = 6$;
- $2 + 1 = 3$, 但 3 已被使用; $3 + 1 = 4 \Rightarrow p_5 = 4$;
- $3 + 1 = 4$, 但 4 已被使用, 小于 4 的 3 也已在左侧出现; $4 + 2 = 6$, 但 6 已被使用; $6 + 1 = 7 \Rightarrow p_6 = 7$;
- $2 + 1 = 3$, 但 3 已被使用; $3 + 1 = 4$, 但 4 已被使用; $4 + 1 = 5 \Rightarrow p_7 = 5$;
- $1 + 1 = 2 \Rightarrow p_8 = 2$;
- 最后找到唯一未被使用的元素, $p_9 = 1$.

综上, $p_1 p_2 \cdots p_9 = 839647521$.

序号、中介数和排列之间有着一一对应关系. 一方面, 从排列可以构造中介数, 从中介数也可以恢复对应的排列; 另一方面, 将中介数的每一位乘上对应的权重就能得到排列的序号, 将序号分解为阶乘的倍数之和就能重新得到中介数.

1.7.3 递增进位制数法

我们继续考虑字典序的中介数. 中介数的每一位对应不同的权重, 最低位对应 $1!$, 次低位对应 $2!$, 以此类推, 从右向左数第 n 位的权重为 $n!$. 作为对比, 我们来观察 10 进制数:

$$297191 = 2 \times 10^5 + 9 \times 10^4 + 7 \times 10^3 + 1 \times 10^2 + 9 \times 10^1 + 1 \times 10^0$$

由于每一位都是逢十进一, 因此各个位置上的数字的权重也分别是 10 的不同幂次. 再来对照中介数:

$$(72642321) \uparrow = 7 \times 8! + 2 \times 7! + 6 \times 6! + 4 \times 5! + 2 \times 4! + 3 \times 3! + 2 \times 2! + 1 \times 1!$$

我们发现, 中介数也可以看成是一个数字, 它每一位的进制从由低位向高位依次递增, 最低位是 2 进制, 次低位是 3 进制, 以此类推. 像这样数字的进制由低位至高位依次递增的数称为递增进位制数, 其表示方式即如上所示, 用小括号将数字括起来, 在后面画一个上箭头.

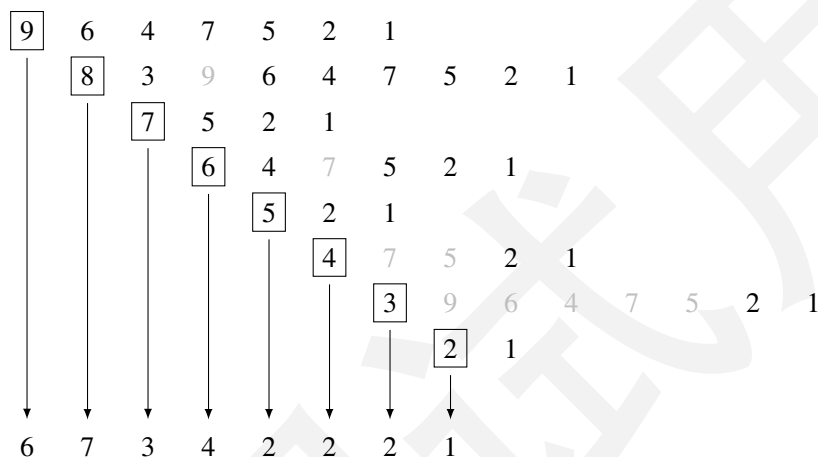
除了每一位进位的方式不同以外, 递增进位制数和普通的数字满足相同的计算规律. 例如:

$$(72642321) \uparrow + (4020) \uparrow = (72652011) \uparrow$$

字典序法中的中介数就是一种递增进位制数. 字典序法的中介数到排列的映射规则是: 从左至右依次考虑排列的每个位置, 结合当前位置左侧已经确定的元素计算出该位置上的

元素. 这种映射方式实际上并不唯一, 我们还可定义另一种由递增进位制数到全排列的一一映射关系.

在字典序法中, 中介数从最低位开始向左数第 i 位 a_i ($i = 1, 2, \dots, n-1$) 的意义是: 排列中位居最右侧的 i 个位置上的元素中, 比从右向左数第 $i+1$ 个位置上的元素更小的元素个数. 我们修改 a_i 的定义, 令其直接表示元素 $(i+1)$ 右侧比 $i+1$ 小的元素个数. 在上述定义下, 得到的递增进位制数实际是即相当于对字典序中介数各位上的数字作了重新排列; 例如, 排列 839647521 按照这一定义, 对应的递增进位制数即是 (67342221) \uparrow .



例 1.7.5. 由递增进位制中介数 (67342221) \uparrow 计算出对应的排列. 中介数有 8 位, 因此是 9 个元素的排列 $p_1 p_2 \cdots p_9$.

- 9 的右边有 6 个数比 9 小, 因此放在右数第 7 个空位:

— — 9 — — — — —

- 8 的右边有 7 个数比 8 小, 因此放在右数第 8 个空位:

8 — 9 — — — — —

- 7 的右边有 3 个数比 7 小, 因此放在右数第 4 个空位:

8 — 9 — — 7 — — —

- 6 的右边有 4 个数比 6 小, 因此放在右数第 5 个空位:

8 — 9 6 — 7 — — —

- 5 的右边有 2 个数比 5 小, 因此放在右数第 3 个空位:

8 — 9 6 — 7 5 — —

- 4 的右边有 2 个数比 4 小, 因此放在右数第 3 个空位:

8 _ 9 6 4 7 5 _ _

- 3 的右边有 2 个数比 3 小, 因此放在右数第 3 个空位:

8 3 9 6 4 7 5 _ _

- 2 的右边有 1 个数比 2 小, 因此放在右数第 2 个空位:

8 3 9 6 4 7 5 2 _

- 将最后的 1 填入唯一一个空位:

8 3 9 6 4 7 5 2 1

综上, $p_1 p_2 \cdots p_9 = 839647521$.

利用递增进位制数和上述计算全排列的方式, 我们得到了一个新的全排列生成顺序. 从中介数 $(00 \cdots 0) \uparrow$ 中可生成第一个排列, 之后不断将中介数加 1 并生成新的排列, 最终一共生成 $n!$ 个排列. 例如, 当 $n = 4$ 时, 从 $(000) \uparrow$ 开始直到 $(321) \uparrow$ 结束, 之间每一个递增进位制数都对应一个排列.

	$(0**) \uparrow$	$(1**) \uparrow$	$(2**) \uparrow$	$(3**) \uparrow$
$(*00) \uparrow$	1 2 3 4	1 2 4 3	1 4 2 3	4 1 2 3
$(*01) \uparrow$	2 1 3 4	2 1 4 3	2 4 1 3	4 2 1 3
$(*10) \uparrow$	1 3 2 4	1 3 4 2	1 4 3 2	4 1 3 2
$(*11) \uparrow$	2 3 1 4	2 3 4 1	2 4 3 1	4 2 3 1
$(*20) \uparrow$	3 1 2 4	3 1 4 2	3 4 1 2	4 3 1 2
$(*21) \uparrow$	3 2 1 4	3 2 4 1	3 4 2 1	4 3 2 1

1.7.4 递减进位制数法

使用递增进位制数生成全排列时, 由于低位的进制更小, 因此进位非常频繁; 并且, 观察上一小节末尾的表格可知, 进位前后生成的排列有明显的变化. 为了减少进位, 我们考虑将递增进位制数反转, 得到递减进位制数. 与递增进位制数相反, 在表示递减进位制数时, 将数字用小括号括起来, 并在后面画一个下箭头:

$$(67342221) \uparrow \longrightarrow (12224376) \downarrow$$

长度为 n 的排列对应 $n-1$ 位的递减进位制数, 其最低位的进制为 n , 次低位进制为 $n-1$, 以此类推, 最高位的进制为 2.

通过递减进位制数也可定义一个全排列生成顺序，如下表所示。

	(00*) ↓	(01*) ↓	(02*) ↓	(10*) ↓	(11*) ↓	(12*) ↓
(**0) ↓	1 2 3 4	1 3 2 4	3 1 2 4	2 1 3 4	2 3 1 4	3 2 1 4
(**1) ↓	1 2 4 3	1 3 4 2	3 1 4 2	2 1 4 3	2 3 4 1	3 2 4 1
(**2) ↓	1 4 2 3	1 4 3 2	3 4 1 2	2 4 1 3	2 4 3 1	3 4 2 1
(**3) ↓	4 1 2 3	4 1 3 2	4 3 1 2	4 2 1 3	4 2 3 1	4 3 2 1

递减进位制数进位不频繁，其优点是容易求出给定排列的下一个或上一个排列。例如，

$$(12224376) \downarrow \longrightarrow 839647521$$

$$(12224377) \downarrow \longrightarrow 893647521$$

$$(12224378) \downarrow \longrightarrow 983647521$$

结合从中介数恢复出排列的算法可知，如果在加（减）递减进位制数时不发生任何进位或借位，则只需将排列中的最大元素向左（向右）移动即可。

1.7.5 SJT 算法

SJT 算法的全称是 Steinhaus-Johnson-Trotter 算法，该算法生成的全排列中，每一个排列都是前一个排列交换一对相邻元素得到的。

在 SJT 算法中，排列中的每个元素都有一个向左或向右的方向，表示为该元素上方一个向左或向右的箭头。如果某一个元素的方向指向比它小的元素，那么就说这个元素是可移动的。算法描述如下：

- (1) 算法从 $p_1 p_2 \cdots p_n = \bar{1} \bar{2} \cdots \bar{n}$ 开始；
- (2) 寻找最大的可移动元素，使它和它指向的元素交换位置，然后反转所有比它更大的元素的方向；
- (3) 不断重复第 (2) 步直到没有元素可以移动。

下面演示如何使用 SJT 算法来生成 $n = 4$ 的全排列。

$\bar{1} \bar{2} \bar{3} \bar{4}$	$\bar{1} \bar{3} \bar{4} \bar{2}$	$\bar{4} \bar{3} \bar{2} \bar{1}$	$\bar{2} \bar{4} \bar{3} \bar{1}$
$\bar{1} \bar{2} \bar{4} \bar{3}$	$\bar{1} \bar{3} \bar{2} \bar{4}$	$\bar{3} \bar{4} \bar{2} \bar{1}$	$\bar{4} \bar{2} \bar{3} \bar{1}$
$\bar{1} \bar{4} \bar{2} \bar{3}$	$\bar{3} \bar{1} \bar{2} \bar{4}$	$\bar{3} \bar{2} \bar{4} \bar{1}$	$\bar{4} \bar{2} \bar{1} \bar{3}$
$\bar{4} \bar{1} \bar{2} \bar{3}$	$\bar{3} \bar{1} \bar{4} \bar{2}$	$\bar{3} \bar{2} \bar{1} \bar{4}$	$\bar{2} \bar{4} \bar{1} \bar{3}$
$\bar{4} \bar{1} \bar{3} \bar{2}$	$\bar{3} \bar{4} \bar{1} \bar{2}$	$\bar{2} \bar{3} \bar{1} \bar{4}$	$\bar{2} \bar{1} \bar{4} \bar{3}$
$\bar{1} \bar{4} \bar{3} \bar{2}$	$\bar{4} \bar{3} \bar{1} \bar{2}$	$\bar{2} \bar{3} \bar{4} \bar{1}$	$\bar{2} \bar{1} \bar{3} \bar{4}$

从 n 个元素的排列中移除 n 就可以得到 $n-1$ 个元素的排列，因此反过来也可以在 $n-1$ 个元素的全排列中插入 n 来得到 n 个元素的全排列。利用这种思路，可以构建一个递归算

法, 而 SJT 算法则可以理解为该递归算法展开后的形式. 例如, $n = 3$ 时, SJT 给出如下全排列:

$$\bar{1}\bar{2}\bar{3}, \bar{1}\bar{3}\bar{2}, \bar{3}\bar{1}\bar{2}, \bar{3}\bar{2}\bar{1}, \bar{2}\bar{3}\bar{1}, \bar{2}\bar{1}\bar{3}$$

如果从上述 $n = 4$ 时的全排列中移除 4, 然后合并相邻的相同排列, 就能得到 $n = 3$ 时的全排列. 反过来说, 将 $n = 3$ 时的全排列中每个排列复制 4 份, 让 4 从排列的最右侧开始一步步移动到最左侧, 然后再一步步移动回最右侧, 如此往复, 就能得到 $n = 4$ 时的全排列.

这种递归性质使我们能为 SJT 算法生成的全排列也设计一个中介数. 由于 SJT 算法每一步都移动最大的可移动元素, 因此这个中介数应当是一个递减进制数. 我们令中介数为 $b_2 b_3 \cdots b_n$, 其中 b_i 表示 i 的反方向所有比 i 小的数字的个数. 例如, 排列

$$\bar{4}\bar{2}\bar{3}\bar{1}$$

对应的中介数为 $(113)\downarrow$, 因为 2 的右侧、3 的左侧、4 的右侧分别有 1、1、3 个数字比它们更小. 和之前讨论过的所有中介数相同, SJT 算法的中介数也只有 $n - 1$ 位, 可以理解为 1 永远不是可移动的, 无需记录在中介数代表的“状态”之中.

若事先不知道排列中每个元素的方向, 同样也能计算出中介数. 这是因为 1 永远不可移动, 导致 2 的方向永远不会改变, 因此 2 的方向一定向左, 从而我们可以求出 b_2 . 之后, 我们可以利用 b_i 的奇偶性一步步推出每一个元素的方向, 规则如下:

- 若 i 为奇数, 则 i 的方向由 b_{i-1} 的奇偶性决定, 奇数则向右、偶数则向左;
- 若 i 为偶数, 则 i 的方向由 $b_{i-1} + b_{i-2}$ 的奇偶性决定, 奇数则向右、偶数则向左.

仍以 4231 这个排列为例. 首先, 1 和 2 都固定向左, 所以 $b_2 = 1$. 3 的方向由 b_2 的奇偶性决定, b_2 是奇数, 所以 3 向右, $b_3 = 1$. 4 的方向由 $b_2 + b_3$ 的奇偶性决定, $b_2 + b_3$ 是偶数, 所以 4 向左, $b_4 = 3$. 由此, 我们同样求出了中介数 $(113)\downarrow$.

例 1.7.6. 设 $n = 9$, 给定的排列为 839647521.

- 2 的方向向左, 2 右侧比它小的数字有 1 个, $b_2 = 1$;
- b_2 为奇数, 因此 3 的方向向右, 3 左侧比它小的数字有 0 个, $b_3 = 0$;
- $b_2 + b_3$ 为奇数, 因此 4 的方向向右, 4 左侧比它小的数字有 1 个, $b_4 = 1$;
- b_4 为奇数, 因此 5 的方向向右, 5 左侧比它小的数字有 2 个, $b_5 = 2$;
- $b_4 + b_5$ 为奇数, 因此 6 的方向向右, 6 左侧比它小的数字有 1 个, $b_6 = 1$;
- b_6 为奇数, 因此 7 的方向向右, 7 左侧比它小的数字有 3 个, $b_7 = 3$;
- $b_6 + b_7$ 为偶数, 因此 8 的方向向左, 8 右侧比它小的数字有 7 个, $b_8 = 7$;
- b_8 为奇数, 因此 9 的方向向右, 9 左侧比它小的数字有 2 个, $b_9 = 2$.

综上, 排列的中介数为:

$$(b_2 b_3 \cdots b_9)\downarrow = (10121372)\downarrow$$

同时, 我们也算出了每个元素的方向:

$$\bar{8}\bar{3}\bar{9}\bar{6}\bar{4}\bar{7}\bar{5}\bar{2}\bar{1}$$

从中介数恢复排列的过程与构造中介数的过程相反, 要从最大的数开始逐个填入空位. 填数时, 首先用与之前类似的方式计算出元素的方向, 然后从对应的方向数出对应数量的空位, 填入元素.

例 1.7.7. 由 SJT 算法中介数 $(10121372) \downarrow$ 计算出对应的排列. 中介数有 8 位, 因此是 9 个元素的排列 $p_1 p_2 \cdots p_9$.

- b_8 为奇数, 因此 9 的方向向右; $b_8 = 2$, 因此放在左数第 3 个空位:

— — 9 — — — — —

- $b_6 + b_7$ 为偶数, 因此 8 的方向向左; $b_8 = 7$, 因此放在右数第 8 个空位:

8 — 9 — — — — —

- b_6 为奇数, 因此 7 的方向向右; $b_7 = 3$, 因此放在左数第 4 个空位:

8 — 9 — — 7 — — —

- $b_4 + b_5$ 为奇数, 因此 6 的方向向右; $b_6 = 1$, 因此放在左数第 2 个空位:

8 — 9 6 — 7 — — —

- b_4 为奇数, 因此 5 的方向向右; $b_5 = 2$, 因此放在左数第 3 个空位:

8 — 9 6 — 7 5 — —

- $b_2 + b_3$ 为奇数, 因此 4 的方向向右; $b_4 = 1$, 因此放在左数第 2 个空位:

8 — 9 6 4 7 5 — —

- b_2 为奇数, 因此 3 的方向向右; $b_3 = 0$, 因此放在左数第 1 个空位:

8 3 9 6 4 7 5 — —

- 2 的方向向左, $b_2 = 1$, 因此放在右数第 2 个空位:

8 3 9 6 4 7 5 2 —

- 将最后的 1 填入唯一一个空位:

8 3 9 6 4 7 5 2 1

综上, $p_1 p_2 \cdots p_9 = 839647521$.

1.8 生成组合

本节中, 我们讨论生成 n 个元素的所有 r 元素组合的算法. 我们只讨论字典序这种生成组合的顺序.

与上一节类似, 我们规定这 n 个元素为 $1, 2, \dots, n$. 这些元素的一个 r 元素组合可以看作一个长度为 r 的严格递增序列 $\{c_i\}$, 其中每个元素都是 1 到 n 之间的正整数. 显然,

$$i \leq c_i \leq n - r + i \quad (i = 1, 2, \dots, r)$$

在字典序下, 如果组合中的所有元素满足 $c_i = i$, 则是第一个组合; 如果所有元素满足 $c_i = n - r + i$, 则是最后一个组合. 因此, 如果 $c_i < n - r + i$, 那么可以认为 c_i 还有“上升空间”. 由此我们可以构建一个算法, 它通过不断替换序列 $\{c_i\}$ 中有上升空间的后缀来以字典序遍历每一个组合. 算法描述如下:

- (1) 算法从 $c_1 c_2 \cdots c_r = 1 2 \cdots r$ 开始;
- (2) 寻找最大的有上升空间的 c_k :

$$k = \max \{i \mid c_i + 1 \leq n, \text{ 且 } c_i + 1 \notin \{c_1, c_2, \dots, c_n\}\}$$

如果找不到这样的 c_k , 那么说明已经遍历完每一个组合, 算法结束;

- (3) 将 c_k 替换为 $c_k + 1$, 并且将由它开始的后缀替换为最小的后缀:

$$c_1 c_2 \cdots c_{i-1} c_k c_{i+1} \cdots c_n \longrightarrow c_1 c_2 \cdots c_{i-1} (c_k + 1) (c_k + 2) \cdots (c_k + r - i + 1)$$

- (4) 回到第 (2) 步.

例 1.8.1. 求当 $n = 6, r = 4$ 时 $\{1, 2, 5, 6\}$ 的下一个组合.

首先, 寻找最大的有上升空间的 c_k , 它出现在 c_2 处:

$$c_2 = 2 < 6 - 4 + 2$$

下一步, 将 $c_2 = 2$ 替换为 $c_2 = 3$, 然后将它后面的元素 c_3, c_4 分别替换为 $c_2 + 1 = 4$ 和 $c_2 = 5$:

$$\{1, 3, 4, 5\}$$

这就是 $\{1, 2, 5, 6\}$ 的下一个组合.

习题

1.1. m 个男生和 n 个女生排成一行 (m, n 均为正整数), 若

- (1) 任何两个男生不相邻 ($m \leq n + 1$);
- (2) n 个女生形成一个整体 (即任何两个女生之间没有男生);

(3) 男生 A 和女生 B 相邻.

分别讨论有多少种方案.

1.2. 6 个男生和 5 个女生围在一圆桌旁, 若

- (1) 任何两个女生不相邻;
- (2) 所有女生形成一个整体;
- (3) 女生 A 两侧均是男生.

分别讨论有多少种方案.

1.3. 计算:

$$\sum_{k=1}^n k \cdot k! = 1 \cdot 1! + 2 \cdot 2! + \cdots + n \cdot n!$$

1.4. 求 10^{40} 与 20^{30} 的公因数的数目.

1.5. 求从 1 到 1000000 的整数的十进制表示中, 数字 0 出现的总次数.

1.6. 将 n 个相同的小球放入 r 个不同的盒子中 ($n \geq r$), 禁止出现空盒, 求方案数.

1.7. 将 n 个相同的小球放入 r 个不同的盒子中, 每盒中至少 k 个球 ($n \geq rk$), 求方案数.

1.8. 8 个盒子排成一行, 将 5 个不同的小球放入这些盒子, 要求空盒不相邻, 求方案数.

1.9. 设 $A = \{(a, b) \mid a, b \in \mathbb{Z}, 0 \leq a \leq 9, 0 \leq b \leq 5\}$.

- (1) 求 xOy 平面上以 A 中的点为四个顶点、四边与坐标轴平行的长方形数目;
- (2) 求 xOy 平面上以 A 中的点为四个顶点、四边与坐标轴平行的正方形数目.

1.10. 分别求从如下多重集中选取 n 个元素的方案数:

- (1) 大小为 $2n$ 的多重集 $\{n \cdot 0, 1, 2, \cdots, n\}$;
- (2) 大小为 $3n + 1$ 的多重集 $\{n \cdot 0, 1, 2, \cdots, 2n + 1\}$.

1.11. 5 台教学机器编号为 1, 2, 3, 4, 5, 分配给 m 名学生使用, 使用第 1 台和第 2 台的人数相等, 求分配方案数.

1.12. 由 n 个 0 和 n 个 1 构成的 $2n$ 位二进制串, 要求任意前 k 位中 0 的数目不少于 1 的数目 ($1 \leq k \leq 2n$), 求满足要求的二进制串的数目.

第二章 鸽巢原理

鸽巢原理 (Pigeonhole Principle) 是组合数学中最简单也最基本的原理. 直观地说, 它指出若将一些物体放进数量不够多的容器中, 那么总要有多个物体共用同一个容器.

与基本计数原理和排列组合不同, 鸽巢原理不解决计数问题, 而是解决存在性问题.

2.1 鸽巢原理的基本形式

定理 2.1 (鸽巢原理). 将 $n+1$ 个物体任意放入 n 个容器中, 那么至少存在一个容器中有至少 2 个物体.

证明. 用反证法. 假设没有容器中有 2 个或更多物体, 则每个容器中最多只有 1 个物体, 从而 n 个容器中最多容纳 n 个物体; 这与有 $n+1$ 个物体这一事实矛盾. \square

下面我们展示如何使用鸽巢原理解决实际问题.

例 2.1.1. 任意找 367 人, 其中总有两人生日相同 (不考虑年份). 这是因为一年最多有 366 天.

例 2.1.2. 北京市内总有两个人头发数量一样多. 我们可以假设一个正常人最多有 100 万根头发, 而近年来北京人口都超过 2000 万.

例 2.1.3. 从 10 双手套中任意取 11 只, 总能凑出一双手套. 最坏情况是拿出了 10 只一边的手套, 不妨设为左手手套, 这时再拿出一只右手的手套就一定能和其中某一只配对.

例 2.1.4. 设 a_1, a_2, \dots, a_n 是由正整数构成的数列, 则必定存在 k, l ($1 \leq k \leq l \leq n$), 使得 $a_k + a_{k+1} + \dots + a_l$ 是 n 的倍数.

证明. 考虑数列 $\{a_n\}$ 的前缀和 $\{S_n\}$, 即

$$S_n = \sum_{i=1}^n a_i \quad (i = 0, 1, 2, \dots, n)$$

各 S_i ($i = 0, 1, 2, \dots, n$) 模 n 的余数只可能是 0 到 $n-1$ 的正整数, 一共有 n 种可能性, 但这些 S_i 共有 $n+1$ 个. 由鸽巢原理, 必定存在互异的 h, l 使得

$$S_h \equiv S_l \pmod{n}$$

不妨设 $l > h$, 则

$$S_l - S_h = a_{h+1} + a_{h+2} + \dots + a_l \equiv 0 \pmod{n}$$

从而它也就是 n 的倍数. □

例 2.1.5. $\left\lfloor \frac{n+3}{2} \right\rfloor$ 个整数中必有两个数的和或差为 n 的倍数.

证明. 将这些整数按被 n 除的余数划分为 n 类:

$$[k] = \{m \mid m \text{ 被 } n \text{ 除余数是 } k\} \quad (k = 0, 1, 2, \dots, n-1)$$

若两个数同属 $[k]$, 或者一个属于 $[k]$ 而另一个属于 $[n-k]$, 那么它们的和或差就是 n 的倍数. 因此, 可以把这些集合两两配对, 形成若干组集合. 当 n 为偶数时, 可以组成 $\frac{n}{2}+1$ 组:

$$\{[0]\}, \left\{ \left[\frac{n}{2} \right] \right\}, \{[k], [n-k]\} \quad (1 \leq k \leq \frac{n}{2}-1)$$

当 n 为奇数时, 则可以组成 $\frac{n+1}{2}$ 组:

$$\{[0]\}, \{[k], [n-k]\} \quad (1 \leq k \leq \frac{n-1}{2})$$

两种情况下均有 $\left\lfloor \frac{n+1}{2} \right\rfloor$ 组集合, 而整数的数量比集合的组数多 1. 由鸽巢原理, 必有两个整数处于同一组的集合中, 这两个整数的和或差即为 n 的倍数. □

例 2.1.6. 从 1 到 $2n$ 中选取 $n+1$ 个互异的正整数, 则其中必定存在两数互质.

证明. 将这 $2n$ 个正整数分为 n 组:

$$S_k = \{2k-1, 2k\} \quad (k = 1, 2, \dots, n)$$

由于选出了 $n+1$ 个互异正整数, 因此由鸽巢原理, 必然会从某一 S_k 中选出两个正整数; 这两个正整数相邻, 因此互质. □

例 2.1.7. 从 1 到 $2n$ 中选取 $n+1$ 个互异的正整数, 则其中必定存在两数, 其中之一为另外一个的倍数.

证明. 考虑选取的 $n+1$ 个正整数 a_1, a_2, \dots, a_{n+1} , 把每个偶数都反复除以 2, 直到得到一个奇数为止. 设经过上述操作后得到的数为 r_1, r_2, \dots, r_{n+1} , 其中所有数均为奇数, 且仍然是 1 到 $2n$ 之间的正整数. 由于 1 到 $2n$ 间只有 n 个奇数, 由鸽巢原理, 必定存在 $r_i = r_j$. 于是我们有

$$a_i = 2^\alpha r_i, a_j = 2^\beta r_j = 2^\beta r_i$$

若 $\alpha > \beta$, 则 a_i 就是 a_j 的倍数; 否则, a_j 就是 a_i 的倍数. □

例 2.1.8. 设 a_1, a_2, \dots, a_{100} 是由 1 和 2 组成的数列, 其中任何连续 10 个数之和都不超过 16:

$$a_i + a_{i+1} + \dots + a_{i+9} \leq 16 \quad (i = 1, 2, \dots, 91)$$

则必定存在 k, l ($1 \leq k \leq l \leq n$), 使得 $a_k + a_{k+1} + \dots + a_l = 39$.

证明. 考虑数列 $\{a_n\}$ 的前缀和 $\{S_n\}$, 即

$$S_i = \sum_{j=1}^i a_j$$

由于 a_i 均为正整数, 因此 $\{S_n\}$ 是严格递增的数列. 又由于 $\{a_n\}$ 中任何连续 10 个数之和都不超过 16, 因此

$$\begin{aligned} S_{100} &= (a_1 + \dots + a_{10}) + (a_{11} + \dots + a_{20}) + \dots + (a_{91} + \dots + a_{100}) \\ &\leq 16 \times 10 = 160 \end{aligned}$$

现在考虑数列

$$S_1, S_2, \dots, S_{100}, S_1 + 39, S_2 + 39, \dots, S_{100} + 39$$

这个数列中每一项均为正整数, 并且最大的一项为 $S_{100} \leq 160 + 39 = 199$. 由鸽巢原理, 数列中必有两项相等; 又由于 $\{S_n\}$ 严格递增, 因此上述数列一定是前 100 项中的一项与后 100 项中的一项相等, 不妨设它们分别为 S_l 和 $S_h + 39$. 于是必然有 $1 \leq h < l \leq 100$, 并且

$$S_l - S_h = a_{h+1} + a_{h+2} + \dots + a_l = 39 \quad \square$$

例 2.1.9. 将集合 $\{i \in \mathbb{Z} \mid 1 \leq i \leq 65\}$ 划分为四个子集 P_1, P_2, P_3, P_4 , 则必定有一个子集, 其中存在一个数能表示为这个子集中两个数的差.

证明. 用反证法. 假设没有任何一个子集中的数能表示为同一集合中两数之差. 由鸽巢原理, 四个子集中至少有一个至少包含 $\left\lceil \frac{65}{4} \right\rceil = 17$ 个数. 不妨设这个子集为 P_1 , 其中的 17 个元素从小到大为 a_1, a_2, \dots, a_{17} .

现在, 令 $b_i = a_{i+1} - a_1$ ($1 \leq i \leq 16$), $B = \{b_1, b_2, \dots, b_{16}\}$. 由反证法假设, 有 $B \cap P_1 = \emptyset$, 因为若二者的交集非空, 则交集的数就能表示为 P_1 中两数之差了. 因此, $B \subseteq (P_2 \cap P_3 \cap P_4)$. 再由鸽巢原理, 这三个子集中至少有一个包含 B 中的至少 $\left\lceil \frac{16}{3} \right\rceil = 6$ 个数.

之后的推导和上面完全类似. 不妨设 $\{b_1, b_2, \dots, b_6\} \subseteq P_2$. 令 $c_i = b_{i+1} - b_1$ ($1 \leq i \leq 5$), $C = \{c_1, c_2, \dots, c_5\}$. 由反证法假设, $C \cap (P_1 \cup P_2) = \emptyset$, 因此 $C \subseteq (P_3 \cap P_4)$. 再由鸽巢原理, 这两个子集中至少有一个包含 C 中的至少 $\left\lceil \frac{5}{2} \right\rceil = 3$ 个数.

最后, 不妨设 $\{c_1, c_2, c_3\} \subseteq P_3$. 令 $d_i = c_{i+1} - c_1$ ($i = 1, 2$), $D = \{d_1, d_2\}$. 由反证法假设, $D \cap (P_1 \cup P_2 \cup P_3) = \emptyset$, 因此 $D \subseteq P_4$; 同时还能从假设中推出 $d_2 - d_1 \notin (P_1 \cup P_2 \cup P_3 \cup P_4)$, 即 $d_2 - d_1 \notin \{i \in \mathbb{Z} \mid 1 \leq i \leq 65\}$. 但是显然 $1 \leq d_2 - d_1 \leq 65$, 矛盾. 因此最初的假设不成立, 从而必有一个子集, 其中存在一个数能表示为这个子集中两个数的差. \square

例 2.1.10. 设 m, n 是互质的正整数, 则对于任意非负整数 a 和 b ($a < m, b < n$), 都必定存在正整数 x 是联立同余方程组的解:

$$\begin{cases} x \equiv a \pmod{m} \\ x \equiv b \pmod{n} \end{cases}$$

这个例子即为著名的中国剩余定理 (Chinese Remainder Theorem).

证明. 用反证法. 假设联立同余方程组不存在解. 考虑 n 个被 m 除余 a 的正整数:

$$a, a+m, a+2m, \dots, a+(n-1)m$$

根据反证法假设, 这些正整数被 n 除的余数只有 $n-1$ 种可能性, 分别是 $0, 1, \dots, b-1, b+1, \dots, n$. 由鸽巢原理, 这些数中必有两项被 n 除余数相等. 设这两项分别为 $a+hm$ 和 $a+lm$ ($0 \leq h < l \leq n-1$), 则

$$(a+lm) - (a+hm) = (l-h)m \equiv 0 \pmod{n}$$

由于 m, n 互质, 因此只能是 $l-h$ 被 n 除余数为 0; 但 $l-h$ 是 1 到 $n-1$ 之间的正整数, 矛盾. 因此原方程组必定有解. \square

2.2 鸽巢原理的推广形式

定理 2.2 (鸽巢原理的推广形式). 设 m_1, m_2, \dots, m_n 是正整数. 将 $m_1 + m_2 + \dots + m_n - n + 1$ 个物体任意放入 n 个容器中, 那么至少存在一个 i ($1 \leq i \leq n$), 使得第 i 个容器中至少有 m_i 个物体.

证明. 用反证法. 假如对于所有的 i , 第 i 个容器中都只有至多 $m_i - 1$ 个物体, 那么 n 个容器中最多容纳的物体数量为

$$\sum_{i=1}^n (m_i - 1) = m_1 + m_2 + \cdots + m_n - n$$

这与有 $m_1 + m_2 + \cdots + m_n - n + 1$ 个物体这一事实矛盾. \square

推论 2.3. 将 m 个物体放入 n 个容器中, 则至少有一个容器内有至少 $\left\lceil \frac{m}{n} \right\rceil$ 个物体.

推论 2.4. 将 m 个物体放入 n 个容器中, 不允许容器为空, 则至少有一个容器内有至多 $\left\lfloor \frac{m}{n} \right\rfloor$ 个物体.

推论 2.5. 设 m_1, m_2, \dots, m_n 是正整数, 并且

$$\frac{m_1 + m_2 + \cdots + m_n}{n} > r - 1$$

则 m_1, m_2, \dots, m_n 中至少有一个不小于 r .

下面同样研究一些实际问题.

例 2.2.1. 设 $A = a_1 a_2 \cdots a_{20}$ 是 10 个 0 和 10 个 1 组成的二进制串. $B = b_1 b_2 \cdots b_{20}$ 是任意二进制串. C 是将 b 重复两次所得的二进制串:

$$C = c_1 c_2 \cdots c_{40} = b_1 b_2 \cdots b_{20} b_1 b_2 \cdots b_{20}$$

那么必然存在 $1 \leq i \leq 20$, 使得 $c_i c_{i+1} \cdots c_{i+19}$ 与 $a_1 a_2 \cdots a_{20}$ 至少有 10 位对应相等.

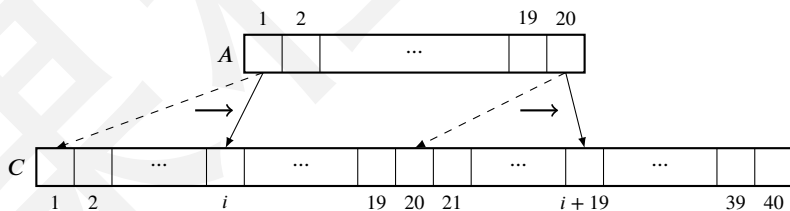


图 2.1: 例 2.2.1 的示意图

证明. 令 i 取遍 $1, 2, \dots, 20$, 这时每个 b_j 都和 a_1, a_2, \dots, a_{20} 分别恰好对上一次. 因为 A 中有 10 个 0 和 10 个 1, 因此每一个 b_j 都有 10 位次对应相等. 从而, 当 i 取遍 $1, 2, \dots, 20$ 时, 总共有 $10 \times 20 = 200$ 位次对应相等. 因此, 至少对某个 i , $c_i c_{i+1} \cdots c_{i+19}$ 与 $a_1 a_2 \cdots a_{20}$ 有至少 $\left\lfloor \frac{200}{20} \right\rfloor = 10$ 位对应相等. \square

例 2.2.2. 设 m, n 是正整数, 序列 $S = \{a_1, a_2, \dots, a_{mn+1}\}$ 中的各项均不相等, 则 S 中有一长度为 $m+1$ 的严格递增子序列, 或者有一长度为 $n+1$ 的严格递减子序列.

证明. 对于“或者……或者……”形式的命题, 可以采用“只要前者不满足, 则后者一定满足”的证明方式.

我们由 S 中的每一项 a_i 开始, 向后选取最长的严格递增子序列, 并设其长度为 l_i . 若存在 $l_i \geq m+1$, 那么命题自然成立. 否则, 对所有的 $1 \leq i \leq mn+1$, 有 $1 \leq l_i \leq m$, 因此由鸽巢原理, 其中必定有 $n+1$ 个 l_i 相等. 不妨设这些相等 l_i 的是

$$l_{i_1} = l_{i_2} = \cdots = l_{i_{n+1}} \quad (1 \leq i_1 < i_2 < \cdots < i_{n+1} \leq mn+1)$$

这时必然有

$$a_{i_1} > a_{i_2} > \cdots > a_{i_{n+1}}$$

否则, 不妨设 $a_{i_1} < a_{i_2}$, 则从 a_{i_2} 开始向后选取的最长的严格递增子序列的头部添加 a_{i_1} , 就构成了一个从 a_{i_1} 开始向后选取的更长的严格递增子序列, 与假设矛盾. 因此, $a_{i_1}, a_{i_2}, \cdots, a_{i_{n+1}}$ 构成了长度为 $n+1$ 的严格递减子序列. \square

从上面两节所展示的鸽巢原理及其推广形式的例子中可以看出, 鸽巢原理本身的运用非常简单, 但前提是必须先对问题作合适的抽象和转化, 才能有效地套用鸽巢原理解决.

2.3 整点问题

本节中, 我们基于鸽巢原理, 具体地讨论一类应用问题. 这些问题针对的是坐标系中一些坐标均为整数的点, 这样的点称为**整点** (lattice point), 或也称格点. 例如 $(1, 2)$ 、 $(0, -1, 0)$ 分别是二维和三维空间中的整点, 但 (π, e) 、 $(-0.1, 1, 1)$ 则不是.

例 2.3.1. 三维坐标系中有 9 个整点, 则在两两相连的线段中, 至少有一个坐标为整数的内点. 内点指的是在线段上、但不是线段端点的点.

证明. 考虑这些线段的中点. 令 9 个点的坐标分别为 (x_i, y_i, z_i) ($1 \leq i \leq 9$), 则

- x_1, x_2, \cdots, x_9 中必有 $\left\lfloor \frac{9}{2} \right\rfloor = 5$ 个奇偶性相同, 不妨设它们为 x_1, x_2, \cdots, x_5 ;
- y_1, y_2, \cdots, y_9 中必有 $\left\lfloor \frac{9}{2} \right\rfloor = 5$ 个奇偶性相同, 不妨设它们为 y_1, y_2, y_3 ;
- z_1, z_2, z_3 中必有 $\left\lfloor \frac{3}{2} \right\rfloor = 2$ 个奇偶性相同, 不妨设它们为 z_1, z_2 .

因此, $(x_1, y_1, z_1), (x_2, y_2, z_2)$ 这两点的三个维度坐标的奇偶性均相同, 因此它们连成的线段的中点是一个整点. \square

例 2.3.2. 平面直角坐标系中有 5 个整点, 则总能从中选出 3 个点, 以其为顶点的三角形的面积为整数 (允许三点共线, 此时认为面积为 0).

证明. 平面直角坐标系中 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ 三个点作为顶点, 构成的三角形面积为

$$S_{\Delta} = \frac{1}{2} \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}$$

由鸽巢原理可知, 5 个点中必有两个点, 其横、纵坐标奇偶性均相同. 不妨设它们为 (x_1, y_1) 和 (x_2, y_2) , 则可设

$$x_1 - x_2 = 2k, \quad y_1 - y_2 = 2l \quad (k, l \in \mathbb{Z})$$

利用行列式性质, 有

$$S_{\Delta} = \frac{1}{2} \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix} = \frac{1}{2} \begin{vmatrix} 0 & 1 & 1 \\ x_1 - x_2 & x_2 & x_3 \\ y_1 - y_2 & y_2 & y_3 \end{vmatrix} = \begin{vmatrix} 0 & 1 & 1 \\ k & x_2 & x_3 \\ l & y_2 & y_3 \end{vmatrix}$$

结果是一个整数. □

例 2.3.3. 平面直角坐标系中有 9 个整点, 则总能从中选出 3 个点使其重心也为整点.

证明. 三个点 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ 的重心坐标为

$$C = \left(\frac{x_1 + x_2 + x_3}{3}, \frac{y_1 + y_2 + y_3}{3} \right)$$

一个整点的两个坐标分量分别对 3 取模, 有 $3 \times 3 = 9$ 种情况; 我们可用一个 3×3 表格表示这 9 种情况, 如图 2.2 所示, 点 (x, y) 占据表格中坐标为 $(x \bmod 3, y \bmod 3)$ 的格子. 若 3 个整点的重心是整点, 则它们在这张表中的位置分布可能是以下四种情况之一:

- (1) 占据同一个格子;
- (2) 占据同一行的 3 个格子;
- (3) 占据同一列的 3 个格子;
- (4) 占据三个既不在同一行, 也不在同一列的格子.

现在用反证法. 假设不存在某 3 个点的重心是整点, 则 9 个点横、纵坐标模 3 余数的分布必须同时不满足以上四种情况. 情况 (1) 不被满足, 意味着每个格子中最多有 2 个点, 因此 9 个点至少占据 $\left\lceil \frac{9}{2} \right\rceil = 5$ 个格子; 情况 (2) 不被满足, 意味着表格的每行中最多有 2 个格子被占据, 因此被占据的格子分布在至少 $\left\lceil \frac{5}{2} \right\rceil = 3$ 行中, 即每行均有被占据的格子. 不妨设第一、二行各有 2 个格子被占据, 第三行则有 1 个格子被占据. 容易验证, 无论这些格子在什么位置, 都存在占满同一行或同一列, 或者能选出 3 个格子使每行、每列恰有一个的情况, 如图 2.2 所示, 这与假设矛盾. 因此, 从 9 个点中总能选出 3 个点, 其重心也为整点. □

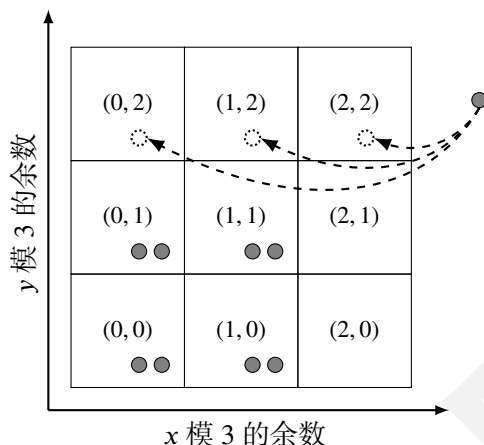


图 2.2: 例 2.3.3 解法中整点坐标分量模 3 后的情形

2.4 Ramsey 问题

假设外星人入侵地球，并且威胁人类必须在一年内算出 Ramsey 数 $R(5, 5)$ ，不然就要摧毁地球；那么我们应该调集世界上最智慧的人和最顶尖的计算机，这样就还有可能在一年内算出这个数。但是，假如外星人要的是 $R(6, 6)$ ，我们将别无选择，只能抢先对外星人发起攻击了。

埃尔德什 (P. Erdős)，匈牙利数学家

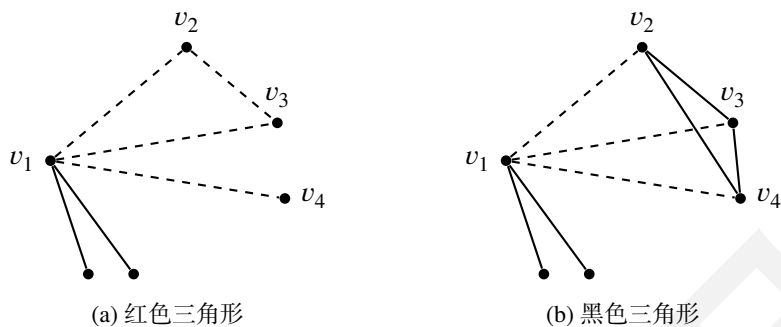
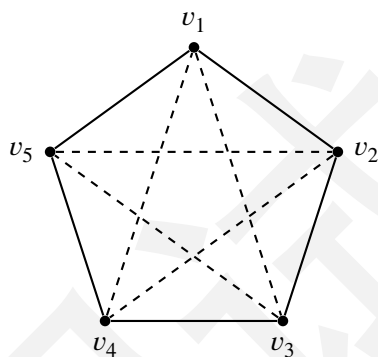
本章最后，我们讨论鸽巢原理的一个经典应用：Ramsey 问题，它以英国数学家拉姆齐 (F. P. Ramsey) 命名。拉姆齐在 1930 年的一篇论文中首次讨论了这个问题，并且给出了一个如今在组合数学界人人耳熟能详的结果：

例 2.4.1. 假设任何两个人或者相互认识，或者相互不认识。则 6 个人中或者有 3 人相互认识，或者有 3 人相互不认识。

证明. 把人看作节点，把认识或不认识的关系看作不同颜色的无向边，问题转化为对完全图的边进行着色的形式，即：对 6 阶完全图 K_6 的边进行红黑二染色，则或者存在红色三角形 (K_3)，或者存在黑色三角形 (K_3)。

从完全图 K_6 中挑选任意一个节点，记为 v_1 。共有 5 条边与 v_1 相连，由鸽巢原理，其中必有 3 条边同色；不妨设这三条边为 v_1v_2, v_1v_3, v_1v_4 ，且它们均是红色。现在考虑 v_2v_3, v_3v_4, v_2v_4 这三条边：如图 2.3 所示，若其中有某条边也是红色，则这条边的两个顶点与 v_1 共同构成一个红色三角形；反之，三条边均为黑色，则它们就构成一个黑色三角形。□

容易验证，若要保证完全图 K_n 中或者有红色 K_3 ，或者有黑色 K_3 ，则必须有 $n \geq 6$ 。这是因为可以构造一个 5 阶完全图，其中没有三边同色的三角形，如图 2.4 所示。

图 2.3: K_6 中总存在红色 (虚线) 或黑色 (实线) 三角形图 2.4: K_5 中可能没有任何同色的三角形

2.4.1 完全图二染色的 Ramsey 定理

Ramsey 问题将以上讨论内容推广到任意阶完全图 K_p 和 K_q 上, 即: 是否存在某个完全图, 对其所有边进行红黑二染色后, 其中或者存在红色的完全子图 K_p , 或者存在黑色的完全子图 K_q ?

Ramsey 定理表明问题的答案是肯定的. 我们给出这一定理的严格表述:

定理 2.6 (Ramsey 定理). 设 $p, q \geq 2$ 为正整数, 则存在最小的正整数 $R(p, q)$ (称为 **Ramsey 数**), 使得若 $n \geq R(p, q)$, 将 K_n 的边进行红黑二染色, 其中或者存在一个红 K_p , 或者存在一个黑 K_q .

之前的讨论表明 $R(3, 3) = 6$. 下面讨论 Ramsey 数的一些简单性质, 以及一些已知的 Ramsey 数的值.

推论 2.7. 设 $p, q \geq 2$ 为正整数, 则

- (1) $R(p, q) = R(q, p)$;
- (2) $R(p, 2) = p$;
- (3) 若 $p, q > 2$, 则 $R(p, q) \leq R(p-1, q) + R(p, q-1)$.

证明. 前两条性质都是平凡的. 对于第一条性质, 只需对换红色和黑色即可得到结论; 对于第二条性质, 因为 K_2 中仅有一条边, 所以如果 K_p 中所有边均为红边, 则构成红 K_p , 反之只要有一条黑边就构成黑 K_2 .

第三条性质的证明如下. 从 $R(p-1, q) + R(p, q-1)$ 阶完全图中随意挑选一个节点, 记为 v_1 ; 有 $R(p-1, q) + R(p, q-1) - 1$ 条边与 v_1 相连. 由鸽巢原理, 其中或者存在 $R(p-1, q)$ 条红边, 或者存在 $R(p, q-1)$ 条黑边.

- 在第一种情况下, 与 v_1 连红边的节点构成的完全子图 $K_{R(p-1, q)}$ 中或者存在红 K_{p-1} , 或者存在黑 K_q . 若是后者, 则命题直接得证; 若是前者, 则这个红 K_{p-1} 与 v_1 共同构成一个红 K_p ;
- 在第二种情况下, 与 v_1 连黑边的节点构成的完全子图 $K_{R(p, q-1)}$ 中或者存在红 K_p , 或者存在黑 K_{q-1} . 若是前者, 则命题直接得证; 若是后者, 则这个黑 K_{q-1} 与 v_1 共同构成一个黑 K_q .

综上, $R(p-1, q) + R(p, q-1)$ 阶完全图中必然存在红 K_p 或黑 K_q , 因此我们有

$$R(p, q) \leq R(p-1, q) + R(p, q-1) \quad \square$$

例 2.4.2. $R(3, 4) = 9$.

证明. 首先证明 $R(3, 4) \leq 9$, 即 K_9 中或者存在一个红 K_3 , 或者存在一个黑 K_4 . 将 9 个顶点标号为 v_1, v_2, \dots, v_9 . 注意到其中至少存在一个点, 它的红边的度 (即连到该点上红边的数量) d_r 不等于 3. 否则, 整张图中的红边数量为

$$\frac{1}{2} \sum_{i=1}^9 d_r(v_i) = \frac{27}{2}$$

这不是一个整数, 因此是不可能的. 从而, 必然有一个点的红边的度不是 3, 不妨设其为 v_1 . 下面按 $d_r(v_1)$ 的取值范围分成两类讨论:

- $d_r(v_1) \geq 4$: 设 $v_1v_2, v_1v_3, v_1v_4, v_1v_5$ 是四条红边. 若 v_2, v_3, v_4, v_5 构成的 K_4 中有一条红边, 则它就和 v_1 共同构成一个红 K_3 ; 否则, 这就是一个黑 K_4 ;
- $d_r(v_1) \leq 2$: v_1 的黑边的度 $d_b(v_1)$ 不小于 6, 因此可设 v_1v_2, \dots, v_1v_7 是六条黑边, 则 v_2, v_3, \dots, v_7 构成的 K_6 中必然存在红 K_3 或黑 K_3 . 若是前者, 则命题直接得证; 若是后者, 则这个黑 K_3 就与 v_1 共同构成一个黑 K_4 .

综上所述, K_9 中必定存在红 K_3 或黑 K_4 .

其次证明 $R(3, 4) > 8$, 即 K_8 中可以既没有红 K_3 也没有黑 K_4 . 这只能通过构造一个这样的 K_8 来证明, 如图 2.5 所示. 综上, 我们得到 $8 < R(3, 4) \leq 9$, 从而可以断定 $R(3, 4) = 9$. □

例 2.4.3. $R(4, 4) = 18$.

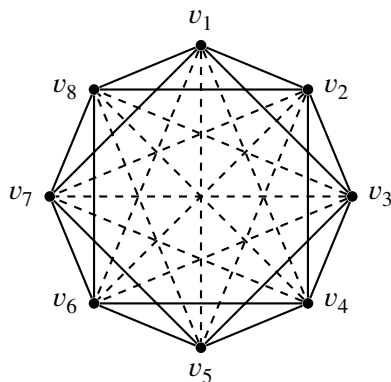


图 2.5: K_8 中可能没有任何红色 (虚线) K_3 或黑色 (实线) K_4

证明. 利用推论 2.7 中的第三条结论, 我们有

$$R(4, 4) \leq R(3, 4) + R(4, 3) = 9 + 9 = 18$$

即, 红黑二染色的 K_{18} 中必定存在红 K_4 或黑 K_4 , 从而 $R(4, 4) \leq 18$. 证明 $R(4, 4) > 17$ 需要构造一个既没有红 K_4 又没有黑 K_4 的 K_{17} , 这非常困难, 具体的构造方式省略. \square

从上面的例子中可以看出, Ramsey 定理虽然说明了 Ramsey 数的存在性, 但它并没有给求解具体的某个 $R(p, q)$ 提供任何帮助. 此外, 尽管求出 Ramsey 数的上界是非常简单的, 但求出下界通常并不容易; 得到的上下界很可能也不是紧界, 这使求解 $R(p, q)$ 的准确值的过程困难重重.

截至目前, 除了 $R(p, 2)$ 这种平凡的 Ramsey 数以外, 人类已知的 Ramsey 数的准确值不超过 10 个. 如果进一步要求 $p = q$, 那么人类已知的 Ramsey 数仅剩 2 个, 分别是我们上面提到的 $R(3, 3) = 6$ 和 $R(4, 4) = 18$. 对于 $R(5, 5)$ 和 $R(6, 6)$, 目前最好的结果分别是 $43 \leq R(5, 5) \leq 48$ 和 $102 \leq R(6, 6) \leq 165$.

2.4.2 Ramsey 定理的推广形式

Ramsey 定理可从两方面作推广:

- (1) 颜色数目可以从 2 种推广为使用 k 种;
- (2) 染色的对象可以从边 (等价于点集的 2 元子集) 推广为点集的 r 元子集.

下面给出推广的 Ramsey 定理:

定理 2.8 (Ramsey 定理的推广形式). 设 $r, k \geq 1$, $q_i \geq r$ ($i = 1, 2, \dots, k$) 是给定的正整数. 则存在一个最小的正整数 $R(q_1, q_2, \dots, q_k; r)$, 使得当 $n > R(q_1, q_2, \dots, q_k; r)$ 时, 如果将 n 元素集 S 的所有 r 元子集划分为 k 个子集族 T_1, T_2, \dots, T_k , 那么或者存在 S 的 q_1 元子集, 它

的所有 r 元子集都属于 T_1 ; 或者存在 S 的 q_2 元子集, 它的所有 r 元子集都属于 T_2 ; ……; 或者存在 S 的 q_k 元子集, 它的所有 r 元子集都属于 T_k .

$R(q_1, q_2, \dots, q_k; r)$ 同样称为 Ramsey 数; 当 $r = 2$ 时, 可将 r 省略, 并简记为 $R(q_1, q_2, \dots, q_k)$.

当 $r = 1$ 时, 一元子集即是所谓的单元素集, 也就是元素本身; 容易发现, 在这种情况下, 上述定理就退化为鸽巢原理的推广形式. 因此,

$$R(q_1, q_2, \dots, q_k; 1) = q_1 + q_2 + \dots + q_k - k + 1$$

然而, 当 $r \geq 2$ 时, 求解 $R(q_1, q_2, \dots, q_k; r)$ 准确值的工作极为困难, 人类对这些 Ramsey 数的了解也非常有限.

例 2.4.4. 用红、蓝、绿三种颜色对 K_{17} 的边进行染色, 一定存在一个同色的 K_3 . 即, $R(3, 3, 3) \leq 17$.

证明. 从 K_{17} 中任选一个节点 v_1 . 由鸽巢原理, 有至少 $\left\lceil \frac{16}{3} \right\rceil = 6$ 条同色边与它相连, 不妨设这些边为红色. 考虑这些边另一端的 6 个节点构成的完全子图 K_6 :

- 若此 K_6 中有红边, 则这些边的顶点就与 v_1 共同构成一个红 K_3 ;
- 若此 K_6 中只有绿色或蓝色边, 则由 $R(3, 3) = 6$ 可知, 其中必有蓝 K_3 或绿 K_3 .

因此, 三染色的 K_{17} 中必有同色 K_3 , 从而有 $R(3, 3, 3) \leq 17$. □

例 2.4.5. 设 $n \geq 4$, 则存在一个最小的正整数 $N(n)$, 使得平面上任意三点不共线的 $N(n)$ 个点中, 总能选出 n 个, 将其顺次相连后构成一个凸 n 边形.

证明. 首先考虑 $n = 4$ 的情形, 这时有 $N(4) = 5$, 可以通过对平面上 5 个点的凸包作分类讨论证明:

- 若凸包是五边形或四边形, 则命题直接得证;
- 若凸包是三角形, 则此三角形的内部有两个点; 由鸽巢原理, 三角形必有两个顶点在其内部两个点所确定的直线的同一侧, 这四个点就共同构成一个凸四边形.

下面考虑 $n > 4$ 的情形. 考察平面上一点构成的点集的所有 4 元子集, 若这 4 个点构成的四边形是凸的, 则把它分到子集族 T_1 中, 否则分到 T_2 中. 由 Ramsey 定理的推广形式可知, 若平面上有 $R(n, 5; 4)$ 个点 (其中任意三点不共线), 则或者存在 n 个点, 其中任意 4 个点构成的四边形都在 T_1 中 (即都是凸的), 或者存在 5 个点, 其中任意 4 个点构成的四边形都在 T_2 中 (即都是凹的). 但是, 上面的论述表明 5 个点中必然能选出 4 个点构成凸四边形, 即后一种情况不可能出现; 因此, 只能是前一种情况成立, 即存在 n 个点, 其中任意 4 个点均能构成凸四边形, 从而这 n 个点也就构成一个凸 n 边形. 综上所述, 我们有 $N(n) = R(n, 5; 4)$. □

习题

2.1. 将集合 $\{i \in \mathbb{Z} \mid 1 \leq i \leq 326\}$ 划分为 5 个子集, 证明必有一个子集, 其中存在一个数能表示为这个子集中两个数的差.

2.2. 设集合 A 包含 13 个互不相等的实数, 证明必定存在 $x, y \in A$, 使得

$$0 < \frac{x-y}{1+xy} \leq 2 - \sqrt{3}$$

2.3. $(m+1)$ 行、 $\left[m\binom{m}{2} + 1\right]$ 列的方格, 用 m 种颜色给每个方格染色, 证明必能找出一个由方格组成的矩形, 其四角的方格染相同颜色.

2.4. 有 7 个互不相同的正整数, 证明其中至少存在 2 个正整数 a, b 使得 $a+b$ 或 $a-b$ 能被 10 除尽.

第三章 母函数

母函数这种手段有点类似于袋子. 我们不需要单独拿着很多小物件——那样可能会让人很尴尬; 我们把它们都装进袋子里, 然后就只需要拿着袋子这一样东西了.

波利亚 (G. Pólya), 匈牙利数学家

母函数是一种强大的解决组合计数问题的工具. 它的中心思想是将数列

$$\{a_n\} : a_0, a_1, a_2, \dots$$

用如下幂级数来表示:

$$G_A(x) = \sum_{n=0}^{\infty} a_n x^n = a_0 + a_1 x + a_2 x^2 + \dots \quad (3-1)$$

此幂级数将数列中的项组合为了一个整体; 通过研究 $G_A(x)$, 我们可以推得数列 $\{a_n\}$ 的性质 (例如它的通项表示).

3.1 引论

对于数列 $\{a_n\}$, 母函数 $G_A(x)$ 将其表示为一个关于 x 的幂级数, 其中 x^k 项的系数是 a_k . 我们下面通过几个例子揭示母函数采用这种表示方式的原因.

例 3.1.1. 考虑我们在第一章中介绍的二项式定理:

$$(x + y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k$$

在上式中令 $y = 1$, 得到

$$(1 + x)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} = \sum_{k=0}^n \binom{n}{k} x^k$$

在上述等式右侧, 我们已得到了一个关于 x 的 (有限) 幂级数. 为此, 我们构造数列 $\{a_k\}$, 其通项为

$$a_k = \binom{n}{k} \quad (0 \leq k \leq n)$$

则可知此数列的母函数即为

$$G_A(x) = (1+x)^n \quad (3-2)$$

$$= \sum_{k=0}^n \binom{n}{k} x^k = \binom{n}{0} + \binom{n}{1}x + \binom{n}{2}x^2 + \cdots + \binom{n}{n}x^n \quad (3-3)$$

$G_A(x)$ 有两种表示方式, 它除了可以写为幂级数形式 (3-3) 以外, 也可写为多项式 $(1+x)$ 的 n 次幂的封闭形式 (3-2). 我们在 1.4 节中已经解释过二项式定理的组合意义, 这里我们从母函数的角度出发重新讨论一次.

$a_k = \binom{n}{k}$ 的组合意义是从一个大小为 n 的集合中选出 k 个元素的方法数. 在对此集合合作组合时, 集合中的每个元素都可以独立地选或不选, 因此不同元素的选择情况适用乘法原理. 我们用 $x^0 = 1$ 表示不选某一元素, 用 $x^1 = x$ 表示选择这一元素, 二者相加得到多项式 $(1+x)$. 考虑 $(1+x)^n$ ——即 n 个 $(1+x)$ 的乘积——的展开式, 根据乘法分配律, 展开式中的 x^k 项一定是使用了 k 个 $(1+x)$ 中的 x 这一项, 以及其余 $n-k$ 个 $(1+x)$ 中的 1 这一项, 它们分别对应了 k 个被选出的元素以及 $n-k$ 个未被选出的元素. 因此, 展开式中 x^k 项的系数即为从 n 个元素中选出 k 个的方案数.

进一步地, 我们还注意到

$$1+x = 1 + 1 \cdot x$$

实际上就是集合中只有一个元素 (即单元素集) 时的母函数形式. 常数项 1 和一次项 x 表示从单元素集中不选任何元素或选出一个元素均有 1 种方案. 当集合中的元素从一个变为 n 个时, 母函数的形式随之从上式变为上式的 n 次幂.

例 3.1.2. 有一枚均匀的正方体骰子, 抛掷两次, 出现的点数之和为 9, 求此事件发生的概率.

这是一个离散概率问题, 每一次投掷骰子, 出现的点数是 1, 2, 3, 4, 5, 6 其中之一, 每种点数的概率均为 $\frac{1}{6}$. 若抛掷两次骰子, 则基本事件空间的大小即为 $6^2 = 36$, 且各基本事件是等可能的.

下面求点数之和为 9 对应的基本事件数量. 我们用 x^k ($k = 1, 2, 3, 4, 5, 6$) 表示抛掷一次骰子时掷出了点数 k 的这一事件, 则抛掷一枚骰子的结果可表示为如下多项式:

$$G_1(x) = x + x^2 + x^3 + x^4 + x^5 + x^6$$

若抛掷两次, 则根据乘法原理, 应当将两个上述多项式相乘:

$$G_2(x) = (x + x^2 + x^3 + x^4 + x^5 + x^6) (x + x^2 + x^3 + x^4 + x^5 + x^6)$$

当我们展开上式右侧的多项式乘积时, 左侧括号中多项式的某一 x^j 和右侧括号中多项式的某一 x^k 相乘, 将得到 x^{j+k} 项; 其意义即是第一次骰子掷出 j 点、第二次掷出 k 点, 点数总和为 $j+k$ 的一个基本事件. 由此可知, 上式展开后, x^9 项的系数即为点数之和为 9 的基本事件数量. 实际进行展开 (具体过程略), 可得 x^9 项的系数为 4, 因此所求的概率为

$$P(\text{点数之和为 } 9) = \frac{4}{36} = \frac{1}{9}$$

进一步地, 假设抛掷 n 次骰子, 点数总和为 m , 则此事件对应的基本事件数量应为

$$G_n(x) = (x + x^2 + x^3 + x^4 + x^5 + x^6)^n$$

中 x^m 项的系数. 构造数列 $\{b_m\}$, 其中 b_m ($n \leq m \leq 6n$) 表示投掷 n 次骰子使点数总和为 m 的基本事件数量, 则 $G_n(x)$ 即为数列 $\{b_m\}$ 的母函数. 上面构造的 $G_1(x)$ 和 $G_2(x)$ 分别是 $n=1, 2$ 时的母函数形式.

例 3.1.3. 有一架托盘天平 (没有游码) 和四枚砝码, 砝码的重量分别为 1 克、2 克、3 克、4 克¹. 若砝码和物体必须放在天平上不同的两侧, 求这些砝码能称出多少种不同重量的物体、每种重量有多少种配置砝码的方案.

我们考虑用母函数来表示砝码能称出的物体重量, 其中 x^n 的系数就代表了有多少种方案称出 n 克的物体. 四枚砝码分别独立选用, 最终可称出的物体重量等于各枚砝码对天平一侧重量的贡献之和. 因此, 重量为 m 克的砝码对应的母函数应为

$$(1 + x^m)$$

将四枚砝码对应的母函数乘到一起, 得到

$$\begin{aligned} G(x) &= (1+x)(1+x^2)(1+x^3)(1+x^4) \\ &= 1+x+x^2+2x^3+2x^4+2x^5+2x^6+2x^7+x^8+x^9+x^{10} \end{aligned}$$

可见, 四枚砝码合在一起能称出 1 克、2 克、……、10 克这十种不同重量的物体, 其中 3 至 7 克这五种重量的物体有两种配置砝码的方案, 而其他重量的物体都仅有一种配置砝码的方案.

例 3.1.4. 考虑例 3.1.3 中的托盘天平, 以及六枚重量分别为 1 克、2 克、4 克、8 克、16 克、32 克的砝码, 砝码和物体仍然必须放在天平上不同的两侧. 求这些砝码能称出多少种不同重量的物体、每种重量有多少种配置砝码的方案.

这个例子中, 我们仍然先写出多项式乘积形式的母函数:

$$G(x) = (1+x)(1+x^2)(1+x^4)(1+x^8)(1+x^{16})(1+x^{32})$$

¹由于天平无法测量质量而只能比较重量, 尽管可能有些许不严谨, 但我们仍然刻意使用“重量”一词.

利用平方差公式展开等式右侧, 将会产生 $2^6 = 64$ 项:

$$\begin{aligned}
 G(x) &= \frac{(1-x)(1+x)(1+x^2)(1+x^4)(1+x^8)(1+x^{16})(1+x^{32})}{1-x} \\
 &= \frac{(1-x^2)(1+x^2)(1+x^4)(1+x^8)(1+x^{16})(1+x^{32})}{1-x} \\
 &= \frac{(1-x^4)(1+x^4)(1+x^8)(1+x^{16})(1+x^{32})}{1-x} \\
 &= \dots \\
 &= \frac{1-x^{64}}{1-x} = 1+x+x^2+x^3+\dots+x^{63}
 \end{aligned}$$

可见, 用这六枚砝码可以称出 1 克到 63 克中的每一个整数重量, 每种重量都恰好有一种称法. 实际上, 由于砝码的重量均是 2 的幂, 每种砝码的配置方案都对应一个六位的二进制数, 而六个二进制位恰好能唯一地表示 0 到 $2^6 - 1 = 63$ 这个范围中的每一个非负整数.

从以上几个例子中我们看到, 将计数数列 $\{a_n\}$ 的各项作为母函数 $G_A(x)$ 展开式中的各项系数, 并非是没有理由的直接规定, 其本质是利用多项式乘法解决计数问题. 具体地, 如果某一组合过程可分为若干独立的步骤 (如例 3.1.1 中的选择每一个元素、例 3.1.2 中的抛掷每一次骰子), 每一步骤有若干种可能的结果 (如元素选或不选、骰子掷出不同的点数), 最终结果可表示为每一步骤结果之和 (如选出元素的总数、骰子的点数之和), 则对此组合过程的计数问题特别适合用母函数解决:

- 每一步骤的不同结果适用加法原理, 可表示为一个多项式或幂级数, 其中的每一项均使用 x (或其他变量) 的幂次表示这一步骤产生的结果;
- 不同步骤间彼此独立, 适用乘法原理, 最终的母函数可表示为不同步骤对应的多项式或幂级数之积.

3.2 母函数的性质

使用母函数 $G(x)$ 研究数列时, 通常需要利用其性质对其作各种变形. 下面我们介绍两类常用的母函数性质.

第一类性质用于在级数形式的母函数

$$G(x) = \sum_{n=0}^{\infty} a_n x^n = a_0 + a_1 x + a_2 x^2 + \dots$$

和分式形式的母函数之间作相互转换. 分式形式的母函数的优点在于容易进行各类代数运算, 包括四则运算、求导和积分, 因此通常作为推导级数形式母函数手段使用. 一般情况下, 给定一个分式形式的母函数, 对其作 Maclaurin 展开即可得到级数形式. 我们最常用到

的是如下 Maclaurin 展开式:

$$\frac{1}{1-ax} = \sum_{n=0}^{\infty} a^n x^n = 1 + ax + a^2 x^2 + \dots \quad (3-4)$$

反过来, 将级数形式的母函数转化为封闭形式则较为困难, 通常需要先凑出上式右侧的形式, 再反向利用上式得到形如 $\frac{1}{1-ax}$ 的分式.

第二类性质描述两个数列和其对应的母函数之间的关联. 设有实数列 $\{a_n\}$ 和 $\{b_n\}$, 其母函数分别为

$$\begin{aligned} A(x) &= \sum_{n=0}^{\infty} a_n x^n = a_0 + a_1 x + a_2 x^2 + \dots \\ B(x) &= \sum_{n=0}^{\infty} b_n x^n = b_0 + b_1 x + b_2 x^2 + \dots \end{aligned}$$

若 $\{a_n\}$ 的项和 $\{b_n\}$ 的项之间存在某些数学关系, 则 $A(x)$ 和 $B(x)$ 间也应该存在相应的关系. 我们具体讨论 7 种不同类型的关系.

定理 3.1 (数列右移). 若

$$b_n = \begin{cases} 0 & (0 \leq n < m) \\ a_{n-m} & (n \geq m) \end{cases}$$

则

$$B(x) = x^m A(x) \quad (3-5)$$

证明.

$$\begin{aligned} B(x) &= 0 + 0 + \dots + 0 + b_m x^m + b_{m+1} x^{m+1} + \dots \\ &= a_0 x^m + a_1 x^{m+1} + \dots \\ &= x^m A(x) \end{aligned} \quad \square$$

定理 3.2 (数列左移). 若 $b_n = a_{n+m}$ ($m \geq 0$), 则

$$B(x) = \frac{1}{x^m} \left[A(x) - \sum_{k=0}^{m-1} a_k x^k \right] \quad (3-6)$$

证明.

$$\begin{aligned}
 B(x) &= a_m + a_{m+1}x + a_{m+2}x^2 + \cdots \\
 &= \frac{1}{x^m} (a_mx^m + a_{m+1}x^{m+1} + a_{m+2}x^{m+2} + \cdots) \\
 &= \frac{1}{x^m} [A(x) - a_0 - a_1x - a_2x^2 - \cdots - a_{m-1}x^{m-1}] \\
 &= \frac{1}{x^m} \left[A(x) - \sum_{k=0}^{m-1} a_k x^k \right]
 \end{aligned}$$

□

以上两条性质中, 数列 $\{b_n\}$ 由数列 $\{a_n\}$ 通过“左右平移”若干项得到, 相当于将母函数整体乘上或除以 x 的对应次幂. 需要注意的是当数列 $\{a_n\}$ “左移” m 项得到 $\{b_n\}$ 时, 其前 m 项 a_0, a_1, \dots, a_{m-1} 被丢弃了, 因此在由 $A(x)$ 计算 $B(x)$ 时也需要首先减去这些项对应的多项式, 再除以 x^m .

定理 3.3 (数列前缀和). 若 $b_n = \sum_{k=0}^n a_k$, 则

$$B(x) = \frac{A(x)}{1-x} \quad (3-7)$$

证明. 将 $B(x)$ 的表达式中的各项系数用 $\{a_n\}$ 中的项表示:

$$\begin{aligned}
 B(x) &= b_0 + b_1x + b_2x^2 + \cdots \\
 &= a_0 + (a_0 + a_1)x + (a_0 + a_1 + a_2)x^2 + \cdots
 \end{aligned}$$

考察上述和式中 a_0, a_1, a_2, \dots 这些数列元素的系数:

$$\begin{aligned}
 B(x) &= (1 + x + x^2 + \cdots) a_0 + (x + x^2 + \cdots) a_1 + (x^2 + \cdots) a_2 + \cdots \\
 &= \frac{1}{1-x} a_0 + \frac{x}{1-x} a_1 + \frac{x^2}{1-x} a_2 + \cdots \\
 &= \frac{1}{1-x} (a_0 + a_1x + a_2x^2 + \cdots) \\
 &= \frac{A(x)}{1-x}
 \end{aligned}$$

□

定理 3.4 (数列后缀和). 若级数 $\sum_{k=0}^{\infty} a_k$ 收敛, $b_n = \sum_{k=n}^{\infty} a_k$, 则

$$B(x) = \frac{A(1) - xA(x)}{1-x} \quad (3-8)$$

证明. 幂级数 $\sum_{k=0}^{\infty} a_k x^k$ 的收敛半径至少为 1, 因此我们有

$$A(1) = a_0 + a_1 + a_2 + \cdots = \sum_{k=0}^{\infty} a_k$$

从而就有

$$b_n = a_n + a_{n+1} + a_{n+2} + \cdots = A(1) - a_0 - a_1 - \cdots - a_{n-1}$$

将后缀和转化为 $A(1)$ 与前缀和的差后, 我们即可以与定理 3.3 中类似的方式进行后续推导:

$$\begin{aligned} B(x) &= b_0 + b_1x + b_2x^2 + \cdots \\ &= A(1) + [A(1) - a_0]x + [A(1) - a_0 - a_1]x^2 + \cdots \\ &= (1 + x + x^2 + \cdots)A(1) - (x + x^2 + \cdots)a_0 - (x^2 + \cdots)a_1 - \cdots \\ &= \frac{A(1)}{1-x} - \left(\frac{x}{1-x}a_0 + \frac{x^2}{1-x}a_1 + \cdots \right) \\ &= \frac{A(1)}{1-x} - \frac{x}{1-x}(a_0 + a_1x + a_2x^2 + \cdots) \\ &= \frac{A(1) - xA(x)}{1-x} \end{aligned} \quad \square$$

以上两条性质讨论的是 $\{b_n\}$ 是 $\{a_n\}$ 的前缀和或后缀和的场景. 当 $\{b_n\}$ 的母函数难以直接求出, 但其元素能表示为另一数列中元素之和时, 可以利用这两条性质, 首先构造数列 $\{a_n\}$ 并求出 $A(x)$, 进而求出 $B(x)$.

定理 3.5. 若 $b_n = na_n$, 则

$$B(x) = xA'(x) \quad (3-9)$$

证明.

$$\begin{aligned} B(x) &= 0 + a_1x + 2a_2x^2 + \cdots \\ &= x(0 + a_1 + 2a_2x + \cdots) \\ &= x(a_0 + a_1x + a_2x^2 + \cdots)' \\ &= xA'(x) \end{aligned} \quad \square$$

定理 3.6. 若 $b_n = \frac{a_n}{1+n}$, 则

$$B(x) = \frac{1}{x} \int_0^x A(t)dt \quad (3-10)$$

证明.

$$\begin{aligned} B(x) &= a_0 + \frac{a_1}{2}x + \frac{a_2}{3}x^2 + \frac{a_3}{4}x^3 + \cdots \\ &= \frac{1}{x} \left(a_0x + \frac{a_1}{2}x^2 + \frac{a_2}{3}x^3 + \frac{a_3}{4}x^4 + \cdots \right) \\ &= \frac{1}{x} \left(a_0t + \frac{a_1}{2}t^2 + \frac{a_2}{3}t^3 + \frac{a_3}{4}t^4 + \cdots \right) \Big|_0^x \\ &= \frac{1}{x} \int_0^x A(t)dt \end{aligned} \quad \square$$

以上两条性质通过对母函数求导和定积分来对数列进行变形. 求导和定积分这两个操作的独特性在于它们不是线性变换, 能在幂级数的不同项上引入不同的系数.

定理 3.7. 若数列 $\{c_n\}$ 满足 $c_n = \sum_{k=0}^n a_k b_{n-k}$, 则

$$C(x) = c_0 + c_1x + c_2x^2 + \cdots = A(x)B(x) \quad (3-11)$$

证明.

$$\begin{aligned} C(x) &= a_0b_0 + (a_1b_0 + a_0b_1)x + (a_2b_0 + a_1b_1 + a_0b_2)x^2 + \cdots \\ &= a_0(b_0 + b_1x + b_2x^2 + \cdots) + a_1x(b_0 + b_1x + \cdots) + a_2x^2(b_0 + \cdots) + \cdots \\ &= (a_0 + a_1x + a_2x^2 + \cdots)(b_0 + b_1x + b_2x^2 + \cdots) \\ &= A(x)B(x) \quad \square \end{aligned}$$

当 $b_n \equiv 1$ 时, 式 (3-11) 就退化为式 (3-7), 所以后者可看作前者的一个特殊情形. $C(x)$ 即为母函数 $A(x)$ 与 $B(x)$ 之积, 称为二者的 **Cauchy 乘积**.

例 3.2.1. 设数列 $\{a_n\}$ 的母函数为

$$G_A(x) = \frac{1}{(1-x)^2}$$

为了求数列的通项表示, 我们可借助如下结论:

$$G_B(x) = \frac{1}{1-x} = 1 + x + x^2 + \cdots$$

此母函数对应数列 $\{b_n\}$, 其中 $b_n \equiv 1$. 定理 3.3 指出 $\{a_n\}$ 中的项可表示为 $\{b_n\}$ 的前缀和:

$$G_A(x) = \frac{1}{1-x} \cdot \frac{1}{1-x} = \sum_{n=0}^{\infty} \left(\sum_{k=0}^n b_k \right) x^n = \sum_{n=0}^{\infty} (n+1)x^n$$

由此可得通项表达式 $a_n = n+1$.

在对母函数作任何变形时, 均无需考虑幂级数收敛性问题, 这是因为母函数 $G(x)$ 中的变量 x 可认为是一遵守实数运算规则的代数符号, 而不是某一具体实数. 附录 B.2 对此进行了详细的讨论.

3.3 整数拆分与 Ferrers 图像

本节中, 我们研究将一个正整数拆分为若干正整数之和的问题. 由于加法满足交换律, 因此在计算拆分方案数前需要先决定是否考虑求和顺序. 如果考虑顺序, 那么称为**有序拆分** (composition), 否则称为**无序拆分** (partition).

3.3.1 有序拆分

有序拆分的情形相对容易处理. 假如要将正整数 n 表示为 r 个正整数 x_1, x_2, \dots, x_r 之和, 即

$$n = x_1 + x_2 + \dots + x_r$$

这是线性方程的非负整数解问题, 可以利用隔板法求解, 我们在 1.6.1 小节已经对此作过讨论. 设想将 n 个相同的小球一字排开, 代表待拆分的正整数 n ; 将 $r-1$ 个隔板插入小球间的 $n-1$ 个空隙中, 方案数为

$$\binom{n-1}{r-1}$$

如果允许将 n 拆分为任意多个正整数, 那么只需要对所有可能的 r 求上式的和:

$$\sum_{r=1}^n \binom{n-1}{r-1} = 2^{n-1}$$

这个结果的组合学意义也可以用隔板法解释: n 个小球间的 $n-1$ 个空隙中, 每个空隙都可以独立地选择是否放置一个隔板, 共有 2 种选择, 利用乘法原理即得到上述结果.

例 3.3.1. $n=4$ 有 8 种有序拆分方案:

$$\begin{aligned} 4 &= 4 \\ &= 1+3 = 2+2 = 3+1 \\ &= 1+1+2 = 1+2+1 = 2+1+1 \\ &= 1+1+1+1 \end{aligned}$$

3.3.2 无序拆分

与有序拆分不同, 无序拆分只考虑拆分出的正整数的个数和取值, 而不关心对这些整数作加法的顺序. 在这种情况下, 隔板法就不再适用了, 这是因为“隔板”这一概念天然具有左右之分, 导致它无法处理不考虑顺序的情况.

无序拆分问题可以表述为如下等价形式:

从 $1, 2, \dots, n$ 中选取某些数, 其中每个数都允许重复选取, 对不同数的选取彼此独立, 问有多少种方案使选取的数总和为 n ?

我们可以使用母函数来解决无序拆分问题. 具体地, 我们考虑拆分得到的每种正整数对总和的贡献. 对任何正整数 k 而言, 拆分结果中可以出现 $0, 1, 2, \dots$ 个 k , 因此 k 对总和的贡献即是 $0, k, 2k, \dots$ 之一, 从而其对应的母函数即为

$$1 + x^k + x^{2k} + x^{3k} + \dots = \frac{1}{1 - x^k} \quad (3-12)$$

拆分结果中可能出现的正整数只有可能是 $1, 2, \dots, n$ 这 n 种情况, 将它们对应的母函数乘在一起, 得到

$$G_n(x) = \frac{1}{(1-x)(1-x^2)\cdots(1-x^n)} \quad (3-13)$$

上式展开成级数形式后, 其中 x^n 的系数即为 n 的无序拆分方案数, 称为 n 的拆分数, 记作 $p(n)$. 我们引入记号

$$[x^n]G(x) \quad (3-14)$$

表示幂级数 $G(x)$ 中 x^n 项的系数. 由此, 我们即有

$$p(n) = [x^n]G_n(x)$$

一般情况下, 若我们只关心 $G_n(x)$ 中 x^n 项的系数, 则可以将每个正整数 k 对应的母函数都写成多项式

$$1 + x^k + x^{2k} + x^{3k} + \cdots + x^{mk} = \frac{1 - x^{(m+1)k}}{1 - x^k} \quad (mk \leq n, (m+1)k > n)$$

而非如式 (3-12) 那样的无穷幂级数, 从而简化对展开式中 x^n 项系数的计算过程. 这是因为对任意给定的 n , 其拆分方案中每个正整数的出现次数都显然有上限, 否则拆分出的正整数之和将超过 n .

例 3.3.2. $n = 4$ 的无序拆分对应的母函数为

$$\begin{aligned} G_1(x) &= (1 + x + x^2 + x^3 + x^4)(1 + x^2 + x^4)(1 + x^3)(1 + x^4) \\ &= 1 + x + 2x^2 + 3x^3 + 5x^4 + \cdots \end{aligned}$$

因此正整数 4 有 5 种无序拆分方案.

这个例子中的母函数也可写成与式 (3-13) 类似的形式:

$$G_2(x) = \frac{1}{(1-x)(1-x^2)(1-x^3)(1-x^4)}$$

然后对其作 Maclaurin 展开到 4 阶项:

$$G_2(x) = 1 + x + 2x^2 + 3x^3 + 5x^4 + \cdots$$

由此能得到与上面相同的结论. 但是, 相较于求复杂分式的多阶导数, 化简多项式乘积显然是更加简便的做法.

上面考虑的母函数 $G_n(x)$ 是给定了 n 时的母函数, 仅在 $m \leq n$ 时有 $p(m) = [x^m]G_n(x)$, 对于更大的 m 则不成立. 为此, 我们考虑构造适用于所有 $p(n)$ 的母函数 (规定 $p(0) = 1$):

$$G(x) = \sum_{n=0}^{\infty} p(n)x^n$$

我们令 k 取遍全部正整数, 分别计算每个 k 对应的母函数 (同式 3-12), 将其全部相乘, 得到

$$G(x) = \prod_{n=1}^{\infty} \frac{1}{1-x^n} = \frac{1}{(1-x)(1-x^2)(1-x^3)\cdots} \quad (3-15)$$

我们始终没有使用母函数求 $p(n)$ 的具体值. 实际上, 用母函数求 $p(n)$ 并不方便, 需要作复杂的级数展开; 但在推导拆分数性质方面, 母函数则是最有力的工具之一. 下面我们讨论几个关于拆分数性质, 并通过母函数方法证明这些性质.

例 3.3.3. 将正整数 n 拆分为不大于 m 的正整数的和, 求拆分方案数.

这个例子中, 我们仅需考虑数 k ($k = 1, 2, \dots, m$) 对总和的贡献, 因此母函数为

$$G_m(x) = \frac{1}{(1-x)(1-x^2)\cdots(1-x^m)}$$

而所求的方案数即为 $[x^n]G_m(x)$.

例 3.3.4. 将正整数 n 拆分为不大于 m 的正整数的和, 要求拆分出至少一个 m , 求拆分方案数.

这个例子中, 我们仍然仅需考虑数 k ($k = 1, 2, \dots, m$) 对总和的贡献. 但是, 数 m 对应的母函数不再是

$$1 + x^m + x^{2m} + x^{3m} + \cdots = \frac{1}{1-x^m}$$

由于至少拆分出一个 m , 因此数 m 对总和的贡献只能是 $m, 2m, 3m, \dots$, 其母函数变为

$$x^m + x^{2m} + x^{3m} + x^{4m} + \cdots = \frac{x^m}{1-x^m}$$

因此, 母函数为

$$G(x) = \frac{x^m}{(1-x)(1-x^2)\cdots(1-x^m)}$$

这个例子还可以利用减法原理求解. 如果一个拆分方案不满足要求, 则其中就不能包含 m , 因此就相当于将正整数 n 拆分为不大于 $m-1$ 的正整数的和. 从上个例子的母函数中减去不满足要求的拆分方案对应的母函数, 就得到了待求的母函数:

$$G(x) = \frac{1}{(1-x)(1-x^2)\cdots(1-x^m)} - \frac{1}{(1-x)(1-x^2)\cdots(1-x^{m-1})}$$

容易验证, 上面两种 $G(x)$ 的表示方式是彼此等价的.

例 3.3.5. 将正整数 n 拆分为互不相同的正整数之和的方案数, 与将 n 拆分为奇数之和

的方案数相同. 例如, 当 $n = 8$ 时, 两者分别有 6 种拆分方案:

$$\begin{array}{ll}
 8 = 8 & 8 = 7 + 1 \\
 = 7 + 1 & = 5 + 3 \\
 = 6 + 2 & = 5 + 1 + 1 + 1 \\
 = 5 + 3 & = 3 + 3 + 1 + 1 \\
 = 5 + 2 + 1 & = 3 + 1 + 1 + 1 + 1 + 1 \\
 = 4 + 3 + 1 & = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1
 \end{array}$$

设 $G_1(x)$ 表示将 n 表示为互不相同的正整数之和的方案数的母函数, $G_2(x)$ 表示将 n 表示为奇数之和的方案数的母函数. 如果 $G_1(x) = G_2(x)$, 那么它们的展开式中 x^n 项的系数就对应相等, 从而两种拆分方案数对任意的 n 都相等.

先求 $G_1(x)$. 使用的正整数互不相同, 则每个正整数至多使用一次, 因此整数 k 对母函数的贡献就是 $(1 + x^k)$. 由此可得

$$\begin{aligned}
 G_1(x) &= (1 + x)(1 + x^2) \cdots (1 + x^k) \cdots \\
 &= \frac{1 - x^2}{1 - x} \frac{1 - x^4}{1 - x^2} \cdots \frac{1 - x^{2k}}{1 - x^k} \cdots \\
 &= \frac{1}{1 - x} \frac{1}{1 - x^3} \cdots \frac{1}{1 - x^{2k-1}} \cdots
 \end{aligned}$$

上述推导的最后一步中, 我们用无穷乘积中每一项的分子约去对应的分母.

再求 $G_2(x)$. 拆分时只能使用奇数. 考虑奇数 $2k - 1$ ($k = 1, 2, 3, \dots$), 它对母函数的贡献的计算方式与之前相同, 为

$$1 + x^{2k-1} + x^{4k-2} + \cdots = \frac{1}{1 - x^{2k-1}}$$

由此可得

$$G_2(x) = \prod_{k=1}^{\infty} \frac{1}{1 - x^{2k-1}} = \frac{1}{1 - x} \frac{1}{1 - x^3} \cdots \frac{1}{1 - x^{2k-1}} \cdots$$

显然 $G_1(x) = G_2(x)$. 因此, 将正整数 n 拆分为互不相同的正整数之和的方案数, 与拆分为奇数之和的方案数相同.

例 3.3.6. 设 $p(n)$ 为正整数 n 的拆分数, 则

$$p(n) < \exp\left(\sqrt{\frac{2\pi^2 n}{3}}\right)$$

其中 $\exp(x)$ 表示指数函数 e^x .

这个例子给出了关于整数拆分数的一个较为宽松的上界, 因此虽然其中出现了常数 e 和 π , 但并不必然意味着 $p(n)$ 与二者之间有某种紧密的联系. 我们使用母函数证明此结论.

证明. 首先写出 $p(n)$ 的母函数, 并将其视为一个正常的幂级数:

$$\begin{aligned} G(x) &= \prod_{n=1}^{\infty} \frac{1}{(1-x^n)} \\ &= \frac{1}{1-x} \frac{1}{1-x^2} \cdots \frac{1}{1-x^k} \cdots \end{aligned}$$

注意到当 $0 < x < 1$ 时, 式

$$\prod_{n=1}^N \frac{1}{(1-x^n)}$$

随 N 的增加严格递减, 且有下界 0; 根据单调有界定理, 此级数收敛. 因此, 我们可任取 $0 < x < 1$, 并在等式两侧分别取自然对数, 然后在等式右侧对每一项作 Maclaurin 展开, 得

$$\begin{aligned} \ln G(x) &= -\ln(1-x) - \ln(1-x^2) - \cdots \\ &= \left(x + \frac{1}{2}x^2 + \frac{1}{3}x^3 + \cdots\right) + \left(x^2 + \frac{1}{2}x^4 + \frac{1}{3}x^6 + \cdots\right) + \cdots \end{aligned}$$

由于 $x > 0$, 等式右侧的级数中每一项均为正数; 又因为其收敛, 因此必为绝对收敛, 调整求和顺序不影响结果. 由此可得

$$\begin{aligned} \ln G(x) &= (x + x^2 + \cdots) + \left(\frac{1}{2}x^2 + \frac{1}{2}x^4 + \cdots\right) + \left(\frac{1}{3}x^3 + \frac{1}{3}x^6 + \cdots\right) + \cdots \\ &= \frac{x}{1-x} + \frac{1}{2} \cdot \frac{x^2}{1-x^2} + \frac{1}{3} \cdot \frac{x^3}{1-x^3} + \cdots \end{aligned}$$

又由 $0 < x < 1$, 注意到

$$\begin{aligned} \frac{x^n}{1-x^n} &= \frac{x}{1-x} \cdot \frac{x^{n-1}}{1+x+x^2+\cdots+x^{n-1}} \\ &< \frac{x}{1-x} \cdot \frac{x^{n-1}}{x^{n-1}+x^{n-1}+\cdots+x^{n-1}} \\ &= \frac{1}{n} \frac{x}{1-x} \end{aligned}$$

将其代回前式, 得到

$$\begin{aligned} \ln G(x) &< \frac{x}{1-x} + \frac{1}{2^2} \cdot \frac{x}{1-x} + \frac{1}{3^2} \cdot \frac{x}{1-x} + \cdots \\ &= \frac{x}{1-x} \left(1 + \frac{1}{2^2} + \frac{1}{3^2} + \cdots\right) \\ &= \frac{\pi^2}{6} \frac{x}{1-x} \end{aligned}$$

最后一步使用了著名的 p 级数求和公式:

$$1 + \frac{1}{2^2} + \frac{1}{3^2} + \cdots = \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

现在我们从幂级数 $G(x)$ 中提取出 $p(n)$. 由于 $p(n) > 0, x > 0$, 因此我们有

$$G(x) = \sum_{k=0}^{\infty} p(k)x^k > p(n)x^n$$

对不等式两侧分别取自然对数, 然后运用 $\ln x \leq x - 1$ 进行放缩, 得到

$$\begin{aligned} \ln p(n) &< \ln G(x) + n \ln \frac{1}{x} \\ &< \ln G(x) + n \left(\frac{1}{x} - 1 \right) \\ &< \frac{\pi^2}{6} \frac{x}{1-x} + n \frac{1-x}{x} \end{aligned}$$

此不等式对任意 $0 < x < 1$ 成立, 因此可令不等式右侧取其极小值, 以获得 $\ln p(n)$ 的一个尽量紧的上界. 为此, 我们利用平均值不等式从不等式右侧式中消去 x , 得到

$$\begin{aligned} \ln p(n) &< 2\sqrt{\frac{\pi^2}{6} \frac{x}{1-x} \cdot n \frac{1-x}{x}} \\ &= \sqrt{\frac{2\pi^2 n}{3}} \end{aligned} \quad \square$$

上述证明过程中进行的放缩较为激进, 导致最终求得的上界非常宽松. 例如, $p(8) = 22$, 但例 3.3.6 给出的上界是 $p(8) \leq 1415$, 二者相差了 60 多倍.

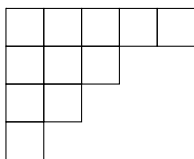
3.3.3 Ferrers 图像

除了母函数以外, 还有多种手段可用于研究整数的无序拆分. 我们下面讨论的 Ferrers 图像即是一种专门用于研究无序拆分的数学工具.

Ferrers 图像是正整数的无序拆分方案的一种图形表示, 它用 n 个边长相同的小正方形方格表示正整数 n . 拆分出的每一个正整数对应 Ferrers 图像的一行, 若拆分出的正整数为 r , 则在 Ferrers 图像的它对应的那一行中左对齐地画出 r 个小方格. 由于拆分是无序的, 为了保证与拆分方案间的一一对应关系, Ferrers 图像中从上到下的每一行方格数单调不减; 换句话说, Ferrers 图像必须以从大到小的顺序依次画出拆分出的每个正整数. 例如, 对正整数 11 的拆分方案

$$11 = 5 + 2 + 3 + 1$$

表示为如下 Ferrers 图像:



拆分出的 3 必须画在 2 的上方, 否则就违反了 Ferrers 图像的规则.

Ferrers 图像看上去像是一个倒过来的台阶：图形在上部和左部对齐，在右下方则形成台阶状的折线。利用 Ferrers 图像，我们能得到一些关于整数拆分方案数的相当有趣的结论。

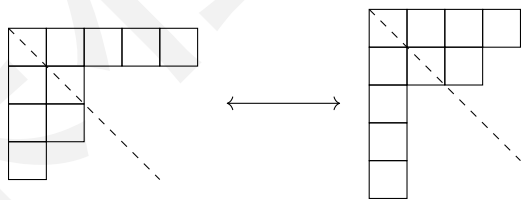
例 3.3.7. 将正整数 n 拆分为不多于 m 个正整数之和的方案数，与将 n 拆分为不超过 m 的正整数之和的方案数相同。

通过 Ferrers 图像，我们可以轻易证明这个结论。对于任何一个不多于 m 个正整数的拆分方案，它对应的 Ferrers 图像最多有 m 行；对于任何一个使用不超过 m 的正整数的拆分方案，它对应的 Ferrers 图像最多有 m 列。我们下面证明这两类 Ferrers 图像间存在一一对应关系，从而二者的数量也就必然相等。

证明. 考虑从 Ferrers 图像的左上角开始，向右下方 45° 方向作一条对角线，然后作 Ferrers 图像关于此对角线的对称图形（可以理解为绕此对角线翻转 180° ）。对称操作使得原来图形中的每一行变成了新图形中的每一列，原来的每一列变成了新图形中的每一行。因此，对一个最多 m 行的 Ferrers 图像作上述对称操作，我们就能得到一个最多 m 列的图形；如果该图形也是一个 Ferrers 图像，那么我们就成功建立了一种从“不超过 m 个正整数的拆分方案”到“使用不超过 m 的正整数的拆分方案”的一一对应关系。

事实上，Ferrers 图像绕对角线作对称操作，得到的一定仍然是一个 Ferrers 图像。首先，经由对称操作得到的图形仍然包含 n 个小方格，并且它们在上部和左部对齐。其次，原本的 Ferrers 图像中每行的小方格都左对齐，因此必然不会出现某一列比它左侧一列更长的情况，否则就意味着多出来的这些方格没有左对齐；因此，在由对称操作得到的新图形中，每一行都不比它上方一行更长，满足 Ferrers 图像的性质。□

例如，考虑将 10 拆分为不超过 4 个正整数的方案，和将 10 拆分为不超过 4 的正整数的方案。我们画出一个满足前者要求的 Ferrers 图像，然后做它关于对角线的对称：



如此，我们就建立了如下两个拆分方案的一一对应关系：

$$10 = 5 + 2 + 2 + 1$$

$$10 = 4 + 3 + 1 + 1 + 1$$

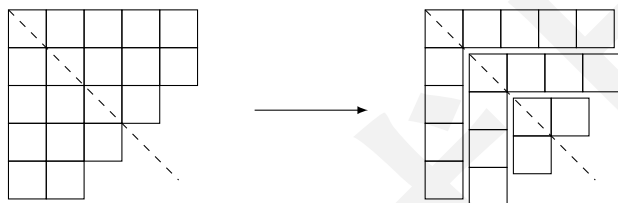
对特定的 n ，如果两个拆分方案对应的 Ferrers 图像关于对角线对称，则我们称这样的两个拆分方案互为共轭。如果一个拆分方案的共轭是它自身，则称这个拆分方案自共轭，它的 Ferrers 图像应当沿对角线轴对称。

例 3.3.8. 正整数 n 的自共轭的拆分方案数与仅使用奇数的拆分方案数相同。

证明. 我们使用 Ferrers 图像证明这个结论. 考虑某个自共轭的拆分方案, 例如

$$19 = 5 + 5 + 4 + 3 + 2$$

画出它的 Ferrers 图像, 这将是一个沿对角线轴对称的图形. 现在设想“切下”这个图形的第一行和第一列. 由于第一行和第一列关于对角线轴对称, 因此切下它们之后, 余下的图形仍然关于对角线轴对称; 简单讨论可知, 余下的图形仍然是一个 Ferrers 图像. 我们继续切下余下的 Ferrers 图像的第一行和第一列, 如此反复, 我们就将原本的 Ferrers 图像切成了若干层, 如下所示:



由于图形沿对角线对称, 因此每次切下的第一行和第一列的长度都相等; 又因为第一行和第一列恰好共享一个小方格, 因此每次切下的小方格的数量必然是个奇数. 另一方面, 每进行一次切割, 剩下的 Ferrers 图像的行数和列数都至少要减少 1, 因此每次切下的小方格数量都必然比上次更少. 因此, 每次切下的小方格的数目将是一系列互不相同的奇数, 而这同样对应于 n 的一个拆分方案. 例如, 上图对应的拆分方案是

$$19 = 9 + 7 + 3$$

拆分出的 3 个正整数分别对应从 Ferrers 图像中切出的 3 层.

上述讨论表明, n 的每一个自共轭的拆分方案都对应一个使用互不相同的奇数的拆分方案. 简单讨论可以得知, 每一个使用互不相同的奇数的拆分方案同样也对应一个自共轭的拆分方案: 我们可以逆向执行上面的流程, 把每个奇数替换为“L”字形的方格, 然后将它们拼在一起, 得到一个沿对角线对称的 Ferrers 图像. 综上所述, n 的自共轭的拆分方案与使用互不相同的奇数的拆分方案之间存在一一对应关系, 因此二者方案数相同. \square

3.4 指数型母函数

回顾 3.1 节中对母函数适用范围的讨论: 母函数适合解决“组合”问题而非“排列”问题, 即将问题分为若干独立的步骤后, 不同的步骤之间没有顺序 (或者也可说是有一预先确定的顺序). 如果不同的步骤间可以有不同的顺序, 则上述母函数形式就相对不适用.

我们以组合数 $\binom{n}{k}$ 和排列数 $P(n, k)$ 为例展示母函数在处理排列类问题时的困难. 对

$\binom{n}{k}$ 而言, 其母函数有简洁的封闭形式:

$$\sum_{k=0}^n \binom{n}{k} x^k = (1+x)^n \quad (3-16)$$

封闭形式 $(1+x)^n$ 有明确的组合意义: n 个元素中的每一个均可以独立地选或不选, 因此对单元素集作组合的方案数的母函数即为 $(1+x)$; 由乘法原理, 它的 n 次幂即为对 n 元素集作组合的方案数的母函数. 现在再考虑排列数 $P(n, k)$, 它的母函数为

$$\sum_{k=0}^n P(n, k) x^k = \sum_{k=0}^n \binom{n}{k} k! \cdot x^k \quad (3-17)$$

一方面, 我们难以将上述和式表示为一个简洁的封闭形式, 这给后续的变形和分析带来了困难; 更关键的是, 排列数的母函数与组合数不同, 不适用乘法原理. 例如, 对单元素集作排列, 其方案数的母函数仍为 $(1+x)$, 但它的 n 次幂则显然不是对 n 元素集作排列的方案数的母函数.

为了解决上述问题, 我们考察式 (3-16) 和式 (3-17) 的区别: 后者在 x^k 这一项的系数上额外乘上了 $k!$. 从组合学意义的角度来看, 二者的展开式中, x^k 项均对应于从一大小为 n 的集合中选出 k 个元素的组合场景, 区别在于组合数不考虑这些元素的顺序, 而排列数考虑元素顺序, 因此需要在组合数的基础上再乘上 $k!$, 代表不同的元素排列方式. 正是这多出来的 $k!$ 使母函数变得难以化简且不满足乘法原理.

考虑母函数本身的含义, 母函数 $G(x)$ 可以视为对数列 $\{a_n\}$ 的一种编码, 它应当满足两个条件: 从 $\{a_n\}$ 出发能够求得 $G(x)$ 展开式中的各项系数, 反过来从 $G(x)$ 出发也能求得 $\{a_n\}$ 的各项. 因此, 母函数的性质

$$[x^n]G(x) = a_n$$

实际上并不必须成立. 理论上, 只需令

$$[x^n]G(x) = f_n(a_n)$$

就能使上述两个条件得到满足, 其中 $f_n (n = 0, 1, 2, \dots)$ 均是可逆函数.

综合以上讨论的结果, 我们发现, 在处理排列问题时, 我们可以不使用原本的母函数形式

$$G(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots$$

改为使用

$$G_e(x) = a_0 + a_1 x + \frac{a_2}{2!} x^2 + \frac{a_3}{3!} x^3 + \dots$$

以消去式 (3-17) 中的 $k!$, 从而使其能够继续化简, 同时也能重新适用乘法原理. 这种形式的母函数称为**指数型母函数**.

定义 3.8 (指数型母函数). 设 $\{a_n\}$ 是实数列. 定义 $\{a_n\}$ 的指数型母函数为

$$G_e(x) = \sum_{n=0}^{\infty} a_n \frac{x^n}{n!} = a_0 + a_1 x + \frac{a_2}{2!} x^2 + \frac{a_3}{3!} x^3 + \dots \quad (3-18)$$

如果令 $a_n \equiv 1$, 则

$$G_e(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x$$

结果是自然指数函数 e^x , 此即为指数型母函数命名的由来.

例 3.4.1. 有 8 颗小球, 其中 3 颗红球、3 颗黄球、2 颗蓝球, 每种颜色的小球之间无任何区别. 从中选取 4 个小球排成一行, 求方案数.

这个例子中, 我们分别考虑每种颜色的小球对应的指数型母函数:

- 红球: $G_R(x) = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3$
- 黄球: $G_Y(x) = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3$
- 蓝球: $G_B(x) = 1 + x + \frac{1}{2!}x^2$

将以上三者相乘, 即得所要求的指数型母函数:

$$\begin{aligned} G_e(x) &= G_R(x)G_Y(x)G_B(x) \\ &= \left(1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3\right)^2 \left(1 + x + \frac{1}{2!}x^2\right) \\ &= 1 + 3x + 9 \cdot \frac{x^2}{2!} + 26 \cdot \frac{x^3}{3!} + 70 \cdot \frac{x^4}{4!} + \dots \end{aligned}$$

可得有 70 种不同的方案.

我们作一些额外的讨论来展示这个例子中指数型母函数的作用. 首先, 为 x 加上 R, Y, B 三种下标之一以区分不同颜色小球的贡献:

$$G_e(x) = \left(1 + x_R + \frac{1}{2!}x_R^2 + \frac{1}{3!}x_R^3\right) \left(1 + x_Y + \frac{1}{2!}x_Y^2 + \frac{1}{3!}x_Y^3\right) \left(1 + x_B + \frac{1}{2!}x_B^2\right)$$

考虑展开式中的 x^4 项, 它有多种表示方式, 例如其中一种为

$$x_R x_Y^2 x_B$$

这代表选出一个红球、两个黄球和一个绿球的情况. 将这四个小球排成一行, 对应于 1.5.1 小节讨论过的多重排列, 其方案数为

$$\frac{4!}{1! \cdot 2! \cdot 1!} \quad (3-19)$$

注意到在三种颜色的小球各自对应的指数型母函数中, x_R, x_Y^2, x_B 的系数分别是 $\frac{1}{1!}, \frac{1}{2!}, \frac{1}{1!}$, 因此这三项乘在一起, 将得到

$$\frac{1}{1! \cdot 2! \cdot 1!} x_R x_Y^2 x_B$$

当利用指数型母函数的定义式 (3-18) 求 a_4 时, 我们需要把 $G_e(x)$ 展开式中 x^4 项的系数乘以 $4!$, 这将使上式中 $x_R x_Y^2 x_B$ 的系数变为式 (3-19), 即真正的多重排列方案数. 由此可见, 在每种颜色的小球对应的指数型母函数中, x^n 项系数除以 $n!$ 的组合学意义在于: n 个相同颜色的小球彼此间交换位置不影响最终的排列结果, 因此需要排除它们之间顺序对结果的影响, 故应除以 $n!$.

例 3.4.2. 求 1, 3, 5, 7, 9 这五个数字组成的 n 位数的个数, 要求 3, 7 两个数字各自出现的次数均为偶数, 其余数字的出现次数则没有限制.

这个例子中, 我们可以设满足条件的 n 位正整数的数量为 a_n . 先写出数列 $\{a_n\}$ 的指数型母函数:

$$G_e(x) = \left(1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots\right)^2 \left(1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots\right)^3$$

其中等式右侧的三次方项对应的是 1, 5, 9 这三个不受限制的数字, 它即等于 e^{3x} . 等式右侧的二次方项对应的是 3, 7 这两个受限的数字, 我们可利用 e^x 的展开式对其进行化简:

$$\begin{aligned} e^x &= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \\ e^{-x} &= 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \cdots \end{aligned}$$

因此,

$$1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots = \frac{1}{2} (e^x + e^{-x})$$

由此, 我们可以化简并展开 $G_e(x)$:

$$\begin{aligned} G_e(x) &= \frac{1}{4} (e^x + e^{-x})^2 e^{3x} \\ &= \frac{1}{4} (e^{2x} + 2 + e^{-2x}) e^{3x} \\ &= \frac{1}{4} (e^{5x} + 2e^{3x} + e^x) \\ &= \frac{1}{4} \left(\sum_{n=0}^{\infty} \frac{5^n x^n}{n!} + 2 \sum_{n=0}^{\infty} \frac{3^n x^n}{n!} + \sum_{n=0}^{\infty} \frac{x^n}{n!} \right) \\ &= \sum_{n=0}^{\infty} \frac{1}{4} (5^n + 2 \cdot 3^n + 1) \frac{x^n}{n!} \end{aligned}$$

最终, 我们得到

$$a_n = \frac{1}{4} (5^n + 2 \cdot 3^n + 1)$$

例 3.4.3. 设 n 为非负整数, 求将 $2n$ 名学生两两分组、各组间没有顺序的方案数.

记分组方案数为 a_{2n} , 我们下面求解数列 $\{a_n\}$ 的指数型母函数. 首先单独考虑一组学生的情况: 由于每组中恰好有 2 名学生, 因此其指数型母函数为

$$G_e^*(x) = \frac{x^2}{2!}$$

总共可以有 $n = 0, 1, 2, \dots$ 组学生, 因此分组方案数的指数型母函数即为

$$\begin{aligned} G_e(x) &= 1 + G_e^*(x) + \frac{1}{2!} G_e^*(x)^2 + \frac{1}{3!} G_e^*(x)^3 + \dots \\ &= e^{G_e^*(x)} \\ &= e^{\frac{x^2}{2}} \end{aligned}$$

要求的是其中 x^{2n} 项的系数, 为 $\frac{1}{2^n \cdot n!}$, 因此分组方案数即为 $\frac{(2n)!}{2^n \cdot n!}$. 这一结果可作如下解释: 首先, 我们将 $2n$ 名同学排成一行, 并将第 1、2 人分为一组、第 3、4 人分为一组, 以此类推, 方案数为 $(2n)!$; 每组内的两人没有顺序, 因此结果要除以 2^n , 另外分出的 n 组间也没有顺序, 因此结果要再除以 $n!$.

表面上, 这个例子中每组内部的两个学生之间没有顺序, 分出的各组间也没有顺序, 看似是一个组合问题, 应该使用普通的母函数. 然而, 实际上, 将上述求解过程中的指数型母函数替换为普通的母函数, 最终只能得到方案数为 1 的错误结论. 这主要是因为普通的母函数无法刻画 n 名学生之间的区别.

▲ 注意

使用指数型母函数时, 需注意使 x^k 项对应的组合场景中恰好包含 k 个可排列顺序的对象. 因此, 若有一个能使用母函数解决的组合问题, 对其增加顺序约束后, 问题不一定能用指数型母函数求解. 例如, 重新考虑例 3.1.3 中的天平和砝码, 将其修改为一个包含顺序约束的版本:

例. 有一架天平 (没有游码) 和若干不同质量的砝码, 天平一侧是放置物品的托盘, 另一侧是用于悬挂砝码的挂钩. 第一枚砝码挂在天平上, 后续砝码依次挂在上一枚砝码下方, 因此砝码可以有不同悬挂顺序. 现有一质量为 m_0 克的物体, 问有多少种悬挂砝码的方式使天平平衡.

这个例子中如不考虑砝码顺序, 则可令 x^n 中的幂次 n 对应砝码的质量, 从而一枚质量为 m 克的砝码就对应于多项式 $(1 + x^m)$, 将所有砝码对应的多项式乘在一起即得到要求的母函数, 其中 x^{m_0} 项的系数即为答案. 但是, 若要考虑砝码的顺序, 则指数型母函数

$$G_e(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

中的 $n!$ 将不具有任何实际意义, 因为我们没有办法将每一克质量分开进行排序. 因此, 指数型母函数不能用于求解这个例子中的问题.

习题

3.1. 设 $a_n = n^3$ ($n \geq 0$), 求数列 $\{a_n\}$ 的母函数, 化简至封闭形式.

3.2. 设 $a_n = \binom{n+3}{3}$ ($n \geq 0$), 求数列 $\{a_n\}$ 的母函数, 化简至封闭形式.

3.3. 设 $a_n = \sum_{k=1}^{n+1} k^3$, 已知

$$\sum_{n=0}^{\infty} (n+1)^3 x^n = \frac{1+4x+x^2}{(1-x)^4}$$

据此求出数列 $\{a_n\}$ 的母函数.

3.4. 设数列 $\{a_n\}$ 的母函数为

$$A(x) = \frac{4-3x}{(1-x)(1+x-x^3)}$$

定义

$$b_0 = a_0, \quad b_1 = a_1 - a_0, \quad \dots, \quad b_n = a_n - a_{n-1}, \quad \dots$$

求数列 $\{b_n\}$ 的母函数.

第四章 线性常系数递推关系

许多组合计数问题实际上并非单独的一个问题，而是一系列形式相同、规模不同的问题实例。问题的规模可以用非负整数 n 表示；若规模为 n 时，计数问题实例的解为 a_n ，则全体 a_n ($n = 0, 1, 2, \dots$) 构成一个计数序列

$$\{a_n\} : a_0, a_1, a_2, \dots$$

递推关系是一种重要的分析计数序列的手段。这是因为许多组合计数问题存在特定的结构，允许我们从它的一个大规模实例中提取出更小规模的实例，从而建立数列 $\{a_n\}$ 中各项间的数学关系。在第三章中，我们展示了几个计数序列并使用母函数分析了它们；本章中，我们继续以母函数为基础，讨论如何分析并求解一类特定形式的递推关系。

4.1 引论

我们首先讨论几个包含递推关系的简单例子。

例 4.1.1. 某水库上游河道的流量为每小时 q 千立方米。假设此水库正在蓄水，水流只进不出，蓄水开始后第 n 小时水库中的水量为 a_n 千立方米，则有递推关系

$$a_n = a_{n-1} + q \quad (n \geq 1)$$

这是因为第 n 小时的蓄水量恰好等于第 $n-1$ 小时的蓄水量加上一小时内流入水库的水量（忽略各种损耗）。设在蓄水开始时，水库中的水量为 a_0 千立方米，则上述递推式有解

$$a_n = a_0 + nq \quad (n \geq 0)$$

上述数列 $\{a_n\}$ 称为算术数列，或也称为等差数列。

例 4.1.2. 某培养皿中初始培养着 a_0 个细菌，每个细菌每小时能分裂一次，产生两个子代细菌。设第 n 个小时末培养皿中的细菌数量为 a_n ，则有递推关系

$$a_n = 2a_{n-1} \quad (n \geq 1)$$

这是因为从第 $n-1$ 小时到第 n 小时的这一小时中，培养皿中的所有细菌都分裂了一次。此递推式有解

$$a_n = 2^n \cdot a_0 \quad (n \geq 0)$$

上述数列 $\{a_n\}$ 称为几何数列, 或也称为等比数列.

这个例子中的问题还可从另一个角度考虑, 从而建立另外一种递推关系. 设 b_n 表示在第 n 个小时中新分裂出的细菌数目. 所有在第 $n-1$ 个小时前诞生的细菌都会在第 n 个小时中分裂一次, 因此有递推关系

$$b_n = \sum_{k=0}^{n-1} b_k \quad (n \geq 1)$$

第 n 个小时末的细菌总数则等于

$$a_n = b_{n+1} = \sum_{k=0}^n b_k \quad (n \geq 1)$$

例 4.1.3. 归并排序是计算机科学领域中的一种常用排序算法. 假设现在要对 n 个整数进行归并排序. 算法首先将这些整数分为两组, 每组 $\frac{n}{2}$ 个整数, 分别使用归并排序算法对这两组整数进行排序. 然后, 该算法从前向后扫描已经排好序的两组整数, 将其合并为一个排好序的数列. 在分组和合并时, 算法平均为每个整数花费 q 条计算机处理器指令.

设执行排序算法总共所需的计算机处理器指令数为 a_n , 则有递推关系

$$a_n = 2a_{n/2} + qn \quad (n > 1)$$

当 $n=1$ 时, 虽然单个元素本身已经有序, 但此时也需执行少量指令, 不妨设 $a_1 = q$. 算法分析中通常使用渐进记号解此递推式; 本例中, 我们为简便起见可设 n 为 2 的幂, 由此得上述递推式的精确解为

$$a_n = qn(\log_2 n + 1) \quad (n = 1, 2, 4, 8, \dots)$$

从上面几个例子可以看出, 通过对计数问题的结构进行分析, 对于计数序列中的某一项 a_n , 我们可用它之前的项将其表示出来, 从而构建数列的递推式. 计数序列的初值 (如 a_0, a_1 等) 一般是已知的, 在此之上, 通过迭代地使用递推式, 我们可计算出序列后面的各项, 以至于推导出数列的通项表达式.

本章中, 我们集中讨论一类特定形式的递推关系的求解方式. 这些递推关系的递推式中包含常数个数列元素的一阶项, 其系数也均为常数, 因此称为**线性常系数递推关系**:

$$a_n + C_1 a_{n-1} + C_2 a_{n-2} + \dots + C_k a_{n-k} = b_n \quad (4-1)$$

其中 k 以及 C_1, C_2, \dots, C_k 均为常数, 并且 $C_k \neq 0$. b_n 是常数或关于 n 的函数, 但不依赖于任何 a_i ; 如果 $b_n = 0$ 恒成立, 则称上述递推关系是**齐次线性常系数递推关系** (linear homogeneous recurrence relation with constant coefficients, 简记为 LHRRCC), 反之则称为**非齐次线性常系数递推关系**. 常数 k 称为此递推关系的阶 (order). 例 4.1.1 和 4.1.2 中展示了两个一阶的线性常系数递推关系, 前者是非齐次的, 后者则是齐次的.

式 (4-1) 称为线性常系数递推关系的递推式. 为了能够使用递推式进行递推, 还需给出数列的初值:

$$a_0, a_1, a_2, \dots, a_{k-1}$$

完整的递推关系需要同时包含初值和递推式.

并非所有递推关系都是线性常系数的, 如例 4.1.3 中的递推式

$$a_n = 2a_{n/2} + qn \quad (n > 1)$$

其中由于出现了 $a_{n/2}$ 这一项, 因此不属于上面定义的线性常系数递推关系. 但是, 有些情况下, 通过一定的变形, 可将这些递推关系转化为线性常系数递推关系. 例如, 对上述递推式而言, 作换元

$$t_k = a_{2^k}$$

可得关于数列 $\{t_k\}$ 的递推式

$$t_k = 2t_{k-1} + q \cdot 2^k \quad (k > 0)$$

这是一个线性常系数递推关系, 可以套用本章后面介绍的方法求解.

4.2 Fibonacci 数列

本节中, 我们介绍 **Fibonacci 数列** $\{F_n\}$, 它包含一个最简单的二阶齐次线性常系数递推关系:

$$F_n = F_{n-1} + F_{n-2} \quad (n \geq 2)$$

1202 年, 意大利数学家斐波那契 (L. Fibonacci) 出版了著作《算盘全书》(Liber Abaci), 他在这本书中提出了一个关于兔子繁殖的问题:

在第一个月的月初, 围栏里有一对刚出生的幼年雌雄兔子. 一对幼年雌雄兔子会在出生一个月后成长为成年兔子, 并且雌兔在随后每个月的月底都生出一对幼年雌雄兔子. 问一年 (12 个月) 后围栏里有多少对雌雄兔子.

设 F_n 表示第 n 个月月底时围栏中有多少对雌雄兔子, 则上述问题即等价于求 F_{12} 的值. 考虑数列 $\{F_n\}$ 应满足的递推关系. 在第 n 个月的月底, 所有的兔子可以分为两类:

- 一类是这个月底刚出生的兔子, 它们的双亲至少要在两个月前出生, 才有足够的时间长大并繁衍, 因此这些兔子有 F_{n-2} 对.
- 另一类是从上个月月底之前就出生、存活到这个月底的, 这些兔子有 F_{n-1} 对.

因此, 我们有

$$F_n = F_{n-1} + F_{n-2} \quad (4-2)$$

第一个月中围栏里有 1 对兔子, 第二个月中它们还无法生育, 围栏里仍然只有这 1 对兔子, 因此我们有初值

$$F_1 = F_2 = 1$$

需要注意, 一般而言, 我们会利用递推关系反向求出

$$F_0 = F_2 - F_1 = 0$$

并将

$$F_0 = 0, F_1 = 1 \quad (4-3)$$

作为数列 $\{F_n\}$ 的初值, 这是因为数列一般都从下标 0 开始, 其初值一般是最初的若干项.

将式 (4-2) 与式 (4-3) 组合在一起, 我们就得到了 Fibonacci 数列的完整递推关系:

$$\begin{cases} F_n = F_{n-1} + F_{n-2} & (n \geq 2) \\ F_0 = 0, F_1 = 1 \end{cases} \quad (4-4)$$

利用此递推关系, 可求得 Fibonacci 提出的问题的答案为 $F_{12} = 144$.

Fibonacci 数列 $\{F_n\}$ 的最初的若干项为

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, \dots$$

此数列满足一系列重要的数学性质. 在讨论 F_n 的通项表示前, 我们先介绍一部分有关 $\{F_n\}$ 的性质.

定理 4.1. Fibonacci 数列中相邻两项互质, 即

$$\gcd(F_n, F_{n+1}) = 1 \quad (n = 1, 2, 3, \dots)$$

其中 $\gcd(a, b)$ 表示 a, b 的最大公约数.

证明. 利用辗转相减法:

$$\begin{aligned} \gcd(F_n, F_{n+1}) &= \gcd(F_{n+1} - F_n, F_n) = \gcd(F_{n-1}, F_n) \\ &= \gcd(F_n - F_{n-1}, F_{n-1}) = \gcd(F_{n-2}, F_{n-1}) \\ &= \dots \\ &= \gcd(F_1, F_2) = 1 \end{aligned}$$

□

定理 4.2. 设 m 和 n 均是正整数, 且 $m \leq n$, 则

$$m \mid n \iff F_m \mid F_n$$

证明. 先证充分性. 当 $m \mid n$ 时, 必定存在正整数 k 使得 $n = km$. 对 k 施加归纳: 当 $k = 1$ 时, 要证的结论是显然的; 当 $k > 1$ 时, 考虑矩阵

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} F_0 & F_1 \\ F_1 & F_2 \end{bmatrix}$$

由于

$$\begin{aligned} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} F_{n-1} & F_n \\ F_n & F_{n+1} \end{bmatrix} &= \begin{bmatrix} F_n & F_{n+1} \\ F_{n-1} + F_n & F_n + F_{n+1} \end{bmatrix} \\ &= \begin{bmatrix} F_n & F_{n+1} \\ F_{n+1} & F_{n+2} \end{bmatrix} \end{aligned}$$

简单讨论可知

$$A^n = \begin{bmatrix} F_{n-1} & F_n \\ F_n & F_{n+1} \end{bmatrix}$$

由于 $k > 1$ 、 $n = km$, 因此我们有

$$A^n = A^m \cdot A^{(k-1)m} = \begin{bmatrix} F_{m-1} & F_m \\ F_m & F_{m+1} \end{bmatrix} \begin{bmatrix} F_{(k-1)m-1} & F_{(k-1)m} \\ F_{(k-1)m} & F_{(k-1)m+1} \end{bmatrix}$$

比较矩阵各位置上的项, 可得

$$F_n = F_{m-1}F_{(k-1)m} + F_mF_{(k-1)m+1}$$

归纳假设给出 $F_m \mid F_{(k-1)m}$, 从而由上式可知 $F_m \mid F_n$, 充分性成立.

再证必要性. 由于 $n \geq m$, 因此可设 $n = m + r$, 其中 r 是非负整数. 对 r 施加归纳: 当 $r = 0$ 时, 结论显然成立; 当 $r > 0$ 时, 仍采用上述的矩阵思路, 我们有

$$A^n = A^{m+r} = \begin{bmatrix} F_{m-1} & F_m \\ F_m & F_{m+1} \end{bmatrix} \begin{bmatrix} F_{r-1} & F_r \\ F_r & F_{r+1} \end{bmatrix}$$

比较矩阵各位置上的项, 可得

$$F_n = F_{m-1}F_r + F_mF_{r+1}$$

F_{m-1} 与 F_m 是 Fibonacci 数列中的相邻两项, 二者必互质, 因此若要使 $F_m \mid F_n$, 则必须有 $F_m \mid F_r$. 归纳假设给出 $m \mid r$, 从而由 $n = m + r$ 可知也有 $m \mid n$, 必要性成立. \square

上述定理指出, 若正整数 m, n 满足 $m \leq n$, 则二者间的整除关系完全等价于 Fibonacci 数 F_m, F_n 间的整除关系. 由此, 我们能得到如下两个推论:

推论 4.3. 设 n, m 是正整数, 则 $\gcd(F_n, F_m) = F_{\gcd(n, m)}$.

推论 4.4. 正整数 $p \geq 5$ 是质数, 当且仅当 F_p 与它之前所有的正 Fibonacci 数均互质, 即

$$\gcd(F_k, F_p) = 1 \quad (k = 1, 2, \dots, p-1)$$

反过来, 若 Fibonacci 数 $F_q \geq 5$ 是质数, 则 q 也必为质数.

上述关于 Fibonacci 数列中质数的推论之所以仅考虑 F_5 及之后的 Fibonacci 数, 主要是为了回避反例 $F_4 = 3$. 虽然 4 不是质数, 但因为 $F_2 = 1$, 所以不能根据 $F_2 \mid F_4$ 就推出 F_4 不是质数.

除了前面几个例子中展示的数论性质之外, Fibonacci 数列还满足一系列恒等式, 统称为 Fibonacci 恒等式. 这些恒等式可看作 Fibonacci 数列另外的递推关系.

定理 4.5. Fibonacci 数列的平方和为

$$\sum_{k=1}^n F_k^2 = F_1^2 + F_2^2 + \dots + F_n^2 = F_n F_{n+1} \quad (4-5)$$

证明. 对 n 施加归纳. 当 $n = 2$ 时, $F_1^2 + F_2^2 = 2 = F_2 F_3$ 成立. 当 $n > 2$ 时, 有

$$\begin{aligned} \sum_{k=1}^n F_k^2 &= (F_1^2 + F_2^2 + \dots + F_{n-1}^2) + F_n^2 \\ &= F_{n-1} F_n + F_n^2 \\ &= F_n (F_{n-1} + F_n) \\ &= F_n F_{n+1} \end{aligned}$$

□

上述证明思路有一个直观的图形表示. 注意到 F_n^2 可以看成是边长为 F_n 的正方形的面积, 因此我们考虑将边长为 F_1, F_2, \dots, F_n 的正方形拼在一起. 图 4.1 展示了 $n = 6$ 时的情形, 对于其他的 n , 图形是完全类似的; 从中可看出, 边长为 F_1, F_2, \dots, F_n 的正方形拼在一起, 正好能形成一个两边长分别为 F_n 和 F_{n+1} 的长方形. 显然, 全部正方形的面积之和等于长方形的面积, 由此能立即得到此定理中的结论.

除了正向使用递推式

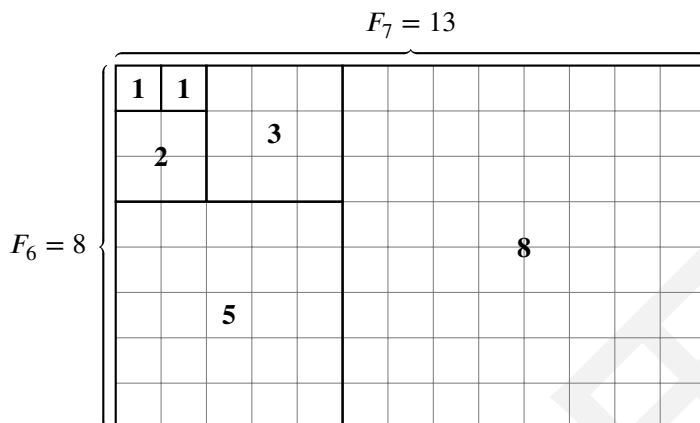
$$F_{n-1} + F_n = F_{n+1}$$

以外, 我们还可对其进行移项, 得到

$$F_n = F_{n+1} - F_{n-1}$$

上式可用于对 F_n^2 裂项, 从而提供此定理的另一种证明思路:

$$\begin{aligned} \sum_{k=1}^n F_k^2 &= (F_1 F_2 - F_0 F_1) + (F_2 F_3 - F_1 F_2) + \dots + (F_n F_{n+1} - F_{n-1} F_n) \\ &= F_n F_{n+1} - F_0 F_1 \\ &= F_n F_{n+1} \end{aligned}$$

图 4.1: $n = 6$ 时 Fibonacci 数列平方和的图形证明

定理 4.6. Fibonacci 数列的和为

$$\sum_{k=1}^n F_k = F_1 + F_2 + \cdots + F_n = F_{n+2} - 1 \quad (4-6)$$

证明. 由 $F_k = F_{k+2} - F_{k+1}$, 可得

$$\begin{aligned} \sum_{k=1}^n F_k &= (F_3 - F_2) + (F_4 - F_3) + \cdots + (F_{n+2} - F_{n+1}) \\ &= F_{n+2} - F_2 \\ &= F_{n+2} - 1 \end{aligned}$$

□

定理 4.7. Fibonacci 数列奇数项的和与偶数项的和为

$$\sum_{k=1}^n F_{2k-1} = F_1 + F_3 + F_5 + \cdots + F_{2n-1} = F_{2n} \quad (4-7)$$

$$\sum_{k=1}^n F_{2k} = F_2 + F_4 + F_6 + \cdots + F_{2n} = F_{2n+1} - 1 \quad (4-8)$$

证明. 由 $F_k = F_{k+1} - F_{k-1}$, 可得

$$\begin{aligned} \sum_{k=1}^n F_{2k-1} &= (F_2 - F_0) + (F_4 - F_2) + \cdots + (F_{2n} - F_{2n-2}) \\ &= F_{2n} - F_0 \\ &= F_{2n} \end{aligned}$$

以及

$$\begin{aligned}
 \sum_{k=1}^n F_{2k} &= (F_3 - F_1) + (F_5 - F_3) + \cdots + (F_{2n+1} - F_{2n-1}) \\
 &= F_{2n+1} - F_1 \\
 &= F_{2n+1} - 1
 \end{aligned}$$

□

定理 4.8. Fibonacci 数的平方与其前后项乘积正好相差 1:

$$F_n^2 + (-1)^n = F_{n-1}F_{n+1}$$

证明. 对 n 施加归纳. 当 $n = 1$ 时, $F_1^2 - 1 = 0 = F_0F_2$ 成立. 当 $n > 1$ 时, 若命题对更小的 n 均成立, 则

$$\begin{aligned}
 (-1)^n + F_n^2 &= (-1)^{n-2} + (F_{n-2} + F_{n-1})^2 \\
 &= ((-1)^{n-2} + F_{n-2}^2) + F_{n-1}^2 + 2F_{n-2}F_{n-1} \\
 &= (F_{n-3}F_{n-1} + F_{n-2}F_{n-1}) + (F_{n-2}F_{n-1} + F_{n-1}^2) \\
 &= F_{n-1}^2 + F_{n-1}F_n \\
 &= F_{n-1}F_{n+1}
 \end{aligned}$$

□

在讨论完 Fibonacci 数列的性质后, 下面我们讨论 Fibonacci 数列 $\{F_n\}$ 的通项表达式. 首先需要知道解的形式. 重新考察 $\{F_n\}$ 满足的递推式

$$F_n = F_{n-1} + F_{n-2}$$

我们将其移项成如下形式:

$$F_n - F_{n-1} - F_{n-2} = 0$$

注意到, 几何数列

$$a_n = q^n$$

可以成为上述递推式的解, 只要其中的参数 q 满足方程

$$q^2 - q - 1 = 0 \quad (4-9)$$

这是因为如果上式成立, 则递推式也必然成立:

$$\begin{aligned}
 a_n - a_{n-1} - a_{n-2} &= q^n - q^{n-1} - q^{n-2} \\
 &= q^{n-2}(q^2 - q - 1) \\
 &= 0
 \end{aligned}$$

解方程 (4-9), 得到如下两个根:

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

容易发现, α, β 的任意线性组合也满足 Fibonacci 数列的递推式. 设

$$b_n = A\alpha^n + B\beta^n \quad (4-10)$$

其中 A, B 是任意实常数, 则我们有

$$\begin{aligned} b_n - b_{n-1} - b_{n-2} &= (A\alpha^n + B\beta^n) - (A\alpha^{n-1} + B\beta^{n-1}) - (A\alpha^{n-2} + B\beta^{n-2}) \\ &= A\alpha^{n-2}(\alpha^2 - \alpha - 1) + B\beta^{n-2}(\beta^2 - \beta - 1) \\ &= 0 + 0 = 0 \end{aligned}$$

如能选取恰当的常数 A, B , 使得上述数列的初值等于 Fibonacci 数列的初值, 则此数列就将满足与 Fibonacci 数列完全相同的递推关系, 从而二者就将是相同的数列. 这时, 式 (4-10) 即成为 Fibonacci 数列的通项表达式. 为了确定 A, B 的值, 我们可列出如下方程组:

$$\begin{cases} A\alpha^0 + B\beta^0 = F_0 = 0 \\ A\alpha^1 + B\beta^1 = F_1 = 1 \end{cases} \quad (4-11)$$

此方程组有解

$$A = \frac{1}{\sqrt{5}}, B = -\frac{1}{\sqrt{5}}$$

将各参数代入式 (4-10) 中, 我们最终得到

$$F_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1 + \sqrt{5}}{2} \right)^n - \left(\frac{1 - \sqrt{5}}{2} \right)^n \right] \quad (4-12)$$

此即为 Fibonacci 数列的通项表达式.

我们现在作一些额外的关于上述求解过程的通用性及逻辑完整性的讨论. 在通用性方面, 方程组 (4-11) 用于从数列初值 a_0, a_1 中解出 A, B , 它的系数矩阵行列式为

$$\begin{vmatrix} 1 & 1 \\ \frac{1 + \sqrt{5}}{2} & \frac{1 - \sqrt{5}}{2} \end{vmatrix} = -\sqrt{5} \neq 0$$

可见, 无论数列初值 a_0, a_1 取何值, $\{a_n\}$ 的通项表达式始终会是式 (4-10) 的形式. 因此, 上述方法具有一定通用性, 可用于根据递推式求出含有待定系数的数列通项表达式, 再根据数列初值求解出各待定系数.

但是, 另一方面, 在上述求解过程中, 我们不加解释地使用了如下事实: 对于特定的 q , 几何数列

$$a_n = q^n$$

是满足齐次线性常系数递推关系

$$F_n = F_{n-1} + F_{n-2}$$

的一个解. 我们利用上述事实构造出了式 (4-10). 问题在于, 整个推理流程中缺少明显的逻辑链条指引我们构造形如 $a_n = q^n$ 形式的解, 我们仅是观察到了可以构造这样的解. 在 4.3 节中, 我们将从另一个角度出发, 利用母函数求解递推关系, 从而填补上述推理中的逻辑空缺部分.

4.3 母函数与递推关系

第三章中介绍的母函数是一种研究数列的有力工具. 回顾其定义式 (3-1):

$$G(x) = \sum_{n=0}^{\infty} a_n x^n = a_0 + a_1 x + a_2 x^2 + \dots$$

母函数 $G(x)$ 通过其展开式的各项系数编码了数列 $\{a_n\}$ 中的各项. 若 $\{a_n\}$ 上存在某种递推关系, 则由于递推关系刻画了 $\{a_n\}$ 中各项之间的数学关系, 因此母函数 $G(x)$ 与其自身间也应满足对应的数学关系; 由此我们可以建立关于 $G(n)$ 的方程、解出 $G(n)$, 并通过展开 $G(n)$ 求出 $\{a_n\}$ 的通项表示. 可见, 母函数具备有效求解数列递推关系的能力.

例 4.3.1. 设数列 $\{a_n\}$ 满足如下非齐次一阶线性常系数递推关系:

$$\begin{cases} a_n = a_{n-1} + 1 & (n \geq 1) \\ a_0 = 1 \end{cases}$$

显然, 此递推式的解为 $a_n = n + 1$.

下面我们展示如何使用母函数求解此递推关系. 我们的目标是将递推式

$$a_n - a_{n-1} - 1 = 0$$

转化为关于 $\{a_n\}$ 的母函数 $G(x)$ 的方程. 由于 a_n 是 $G(x)$ 展开式中 x^n 项的系数, 因此上式等价于

$$[x^n]G(x) - [x^{n-1}]G(x) - 1 = 0 \quad (4-13)$$

我们已知

$$\frac{1}{1-x} = \sum_{n=0}^{\infty} x^n = 1 + x + x^2 + \dots$$

因此式 (4-13) 可转化为如下形式:

$$[x^n]G(x) - [x^n](xG(x)) - [x^n]\left(\frac{1}{1-x}\right) = 0$$

这就相当于

$$G(x) - xG(x) - \frac{1}{1-x} = 0 \quad (4-14)$$

我们可以分别展开 $G(x)$ 、 $xG(x)$ 和 $\frac{1}{1-x}$ 以对上式获得一个更直观的理解:

$$\begin{aligned} G(x) &= a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots \\ xG(x) &= a_0x + a_1x^2 + a_2x^3 + \cdots \\ \frac{1}{1-x} &= 1 + x + x^2 + x^3 + \cdots \end{aligned}$$

式 (4-14) 是一个关于 $G(x)$ 的一次方程. 求解此方程得

$$G(x) = \frac{1}{(1-x)^2}$$

例 3.2.1 中已求解过 $\frac{1}{(1-x)^2}$ 的展开式, 为

$$G(x) = \frac{1}{(1-x)^2} = \sum_{n=0}^{\infty} (n+1)x^n$$

从而由母函数的定义可得

$$a_n = n+1$$

这与我们上面通过直接观察递推关系给出的通项表达式相符.

例 4.3.2. 著名的汉诺塔 (Tower of Hanoi) 问题描述了这样一个场景: 有 A, B, C 三根柱子和 n 个直径不同的圆盘, 初始时圆盘按照小在上、大在下的顺序堆放在 A 柱上, 如图 4.2 所示. 可以按照如下规则移动圆盘:

- (1) 每次只能移动某根柱子顶部的一个圆盘, 将其移至另一根柱子的顶部;
- (2) 直径更大的圆盘不能置于直径更小的圆盘上方.

问将所有圆盘移至 C 柱上所需的移动次数.

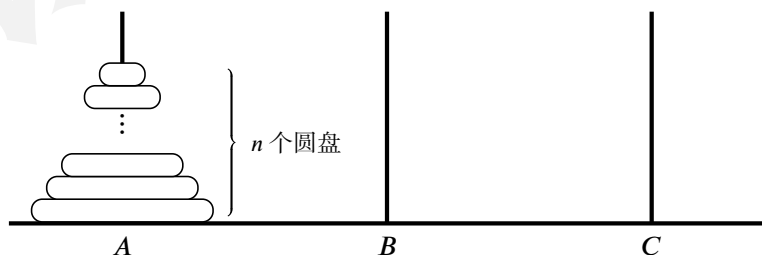


图 4.2: 汉诺塔问题示意图

设将 n 个圆盘移动到 C 柱上需要移动 h_n 次圆盘. 注意到, 在移动最下方的圆盘时, A 柱和 C 柱上不能有其他任何圆盘, 因此这时其他的 $n-1$ 个圆盘一定都在 B 柱上. 由此, 我们可将整个移动圆盘的过程分为三段:

- (1) 将 $n-1$ 个圆盘由 A 柱移至 B 柱, 花费 h_{n-1} 步;
- (2) 将最大的圆盘由 A 柱移动至 C 柱, 花费 1 步;
- (3) 将 $n-1$ 个圆盘由 B 柱移至 C 柱, 花费 h_{n-1} 步.

当 $n=0$ 时柱子上没有圆盘, 因此无需进行任何移动, 从而有 $h_0=0$. 由此可得关于数列 $\{h_n\}$ 的一个递推关系:

$$\begin{cases} h_n = 2h_{n-1} + 1 & (n \geq 1) \\ h_0 = 0 \end{cases}$$

我们继续使用母函数求解上述递推关系. 设 $\{h_n\}$ 的母函数为

$$H(x) = \sum_{n=0}^{\infty} h_n x^n = h_0 + h_1 x + h_2 x^2 + h_3 x^3 + \cdots$$

与例 4.3.1 类似地, 递推式

$$h_n - 2h_{n-1} - 1 = 0$$

等价于

$$[x^n]H(x) - [x^n](2xH(x)) - [x^n]\left(\frac{1}{1-x}\right) = 0 \quad (n \geq 1)$$

注意由于 $h_0=0 \neq 1$, 上式对 $n=0$ 并不成立; 因此若将其转化为关于 $H(x)$ 的方程, 则常数项不为零. 通过列出 $H(x)$ 、 $2xH(x)$ 和 $\frac{1}{1-x}$ 的展开式并比较, 可以明确看出这一点:

$$\begin{array}{rcl} H(x) & = & h_0 + h_1 x + h_2 x^2 + h_3 x^3 + \cdots \\ 2xH(x) & = & 2h_0 x + 2h_1 x^2 + 2h_2 x^3 + \cdots \\ \frac{1}{1-x} & = & 1 + x + x^2 + x^3 + \cdots \end{array}$$

以横线上式减去横线下两式, 得

$$H(x) - 2xH(x) - \frac{1}{1-x} = -1 \quad (4-15)$$

求解此方程, 得到

$$\begin{aligned} H(x) &= \frac{x}{(1-x)(1-2x)} = \frac{1}{1-2x} - \frac{1}{1-x} \\ &= \sum_{n=0}^{\infty} 2^n x^n - \sum_{n=0}^{\infty} x^n \\ &= \sum_{n=0}^{\infty} (2^n - 1) x^n \end{aligned}$$

从而由母函数的定义可得

$$h_n = 2^n - 1 \quad (4-16)$$

此即为汉诺塔问题的解.

例 4.3.3. 求十进制表示中包含偶数个 5 的 n 位正整数的数目.

诸如 35、25、513 这样的正整数包含奇数个 5, 而 565、255、71 等正整数则包含偶数个 5. 若一正整数包含奇数 (或偶数) 个 5, 那么在它的尾部 (即原先的个位数右侧) 添加一个 5, 就得到一个包含偶数 (或奇数) 个 5 的正整数; 反之, 如果在尾部添加除 5 以外的九种数字之一, 则仍得到包含奇数 (或偶数) 个 5 的正整数. 由此, 我们可构建递推式来描述和求解这个例子中的问题.

设 a_n 表示包含偶数个 5 的 n 位正整数的数目, b_n 表示包含奇数个 5 的 n 位正整数的数目. 为求出 a_n 满足的递推关系, 我们考察 n 位正整数的前 $n-1$ 位中 5 的数目的奇偶性:

- 若有偶数个 5, 则前 $n-1$ 位有 a_{n-1} 种方案, 最后一位数可取 5 以外的所有 9 个数字, 共有 $9a_{n-1}$ 种方案;
- 若有奇数个 5, 则前 $n-1$ 位有 b_{n-1} 种方案, 最后一位数只能取 5 来让 5 的个数变为偶数, 共有 b_{n-1} 种方案.

因此, 我们有

$$a_n = 9a_{n-1} + b_{n-1}$$

对 b_n 作类似的讨论, 可知

$$b_n = 9b_{n-1} + a_{n-1}$$

数列初值 $a_1 = 8, b_1 = 1$ 分别表示不包含 5 和包含 5 的一位正整数的数目. 综上, 我们可列出如下递推关系:

$$\begin{cases} a_n = 9a_{n-1} + b_{n-1} & (n \geq 2) \\ b_n = a_{n-1} + 9b_{n-1} & (n \geq 2) \\ a_1 = 8, b_1 = 1 \end{cases}$$

与之前讨论的情况稍有不同, 上述递推关系虽然是一个齐次的线性常系数递推关系, 但其中却包含两个数列. 对于这种情况, 我们有两种常见的处理手段, 下面分别讨论.

第一种处理手段是消元, 适合用于消元后的递推式不太复杂的情况. 注意到, a_n, b_n 的定义合起来涵盖了全体 n 位正整数, 因此我们有

$$a_n + b_n = 9 \cdot 10^{n-1}$$

代入上述递推关系, 得

$$\begin{cases} a_n = 8a_{n-1} + 9 \cdot 10^{n-2} & (n \geq 2) \\ a_1 = 8 \end{cases}$$

此递推关系中仅包含 $\{a_n\}$ 中的项, 可以使用之前讨论过的方式求解. 设

$$A(x) = \sum_{n=1}^{\infty} a_n x^n = a_1 x + a_2 x^2 + a_3 x^3 + \dots$$

为数列 $\{a_n\}$ 的母函数. 由递推式,

$$[x^n]A(x) - [x^n](8xA(x)) - [x^n]\left(\frac{9x^2}{1-10x}\right) = 0 \quad (n \geq 2)$$

上式中, 我们利用 $\frac{1}{1-ax}$ 的展开式构造了非齐次项对应的母函数. 比较母函数展开式中的各项系数:

$$\begin{array}{rcl} A(x) & = & a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + \dots \\ 8xA(x) & = & 8a_1 x^2 + 8a_2 x^3 + 8a_3 x^4 + \dots \\ \frac{9x^2}{1-10x} & = & 9 \cdot 10^0 x^2 + 9 \cdot 10^1 x^3 + 9 \cdot 10^2 x^4 + \dots \end{array}$$

可得

$$A(x) - 8xA(x) - \frac{9x^2}{1-10x} = 8x$$

求解此方程, 得

$$A(x) = \frac{8x - 71x^2}{(1-8x)(1-10x)}$$

为了展开 $A(x)$, 我们常用的手段是从中拆分成形如 $\frac{1}{1-ax}$ 的分式:

$$A(x) = x \left(\frac{P}{1-8x} + \frac{Q}{1-10x} \right)$$

其中 P, Q 是待定系数, 可通过比较分子中的常数项与一次项系数求解:

$$\begin{cases} P + Q = 8 \\ -10P - 8Q = -71 \end{cases}$$

解得

$$P = \frac{7}{2}, Q = \frac{9}{2}$$

因此, 我们有

$$\begin{aligned} A(x) &= \frac{x}{2} \left(\frac{7}{1-8x} + \frac{9}{1-10x} \right) \\ &= \frac{x}{2} \left[\sum_{n=0}^{\infty} (7 \cdot 8^n) x^n + \sum_{n=0}^{\infty} (9 \cdot 10^n) x^n \right] \\ &= \sum_{n=1}^{\infty} \frac{1}{2} (7 \cdot 8^{n-1} + 9 \cdot 10^{n-1}) x^n \end{aligned}$$

从而由母函数的定义可得

$$a_n = \frac{7 \cdot 8^{n-1} + 9 \cdot 10^{n-1}}{2} \quad (n \geq 1)$$

第二种手段是对数列 $\{a_n\}$ 和 $\{b_n\}$ 分别构造母函数, 解关于两个母函数的二元一次方程组, 这种方法有较好的通用性. 设数列 $\{a_n\}$ 和 $\{b_n\}$ 的母函数分别为

$$A(x) = \sum_{n=1}^{\infty} a_n x^n = a_1 x + a_2 x^2 + a_3 x^3 + \cdots$$

$$B(x) = \sum_{n=1}^{\infty} b_n x^n = b_1 x + b_2 x^2 + b_3 x^3 + \cdots$$

由递推式可得

$$\begin{cases} [x^n]A(x) - [x^n](9xA(x)) - [x^n](xB(x)) = 0 \\ [x^n]B(x) - [x^n](xA(x)) - [x^n](9xB(x)) = 0 \end{cases} \quad (n \geq 2)$$

比较母函数展开式中的各项系数:

$$\begin{array}{rcl} A(x) & = & a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + \cdots \\ 9xA(x) & = & 9a_1 x^2 + 9a_2 x^3 + 9a_3 x^4 + \cdots \\ xB(x) & = & b_1 x^2 + b_2 x^3 + b_3 x^4 + \cdots \\ \hline B(x) & = & b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^4 + \cdots \\ 9xB(x) & = & 9b_1 x^2 + 9b_2 x^3 + 9b_3 x^4 + \cdots \\ xA(x) & = & a_1 x^2 + a_2 x^3 + a_3 x^4 + \cdots \end{array}$$

可得

$$\begin{cases} A(x) - 9xA(x) - xB(x) = 8x \\ B(x) - 9xB(x) - xA(x) = x \end{cases} \quad (4-17)$$

求解此方程组, 得

$$\begin{cases} A(x) = \frac{8x - 71x^2}{1 - 18x + 80x^2} \\ B(x) = \frac{x - x^2}{1 - 18x + 80x^2} \end{cases}$$

这与我们在第一种方法中解出的 $A(x)$ 一致; 后续求解 a_n 通项表示的过程也与第一种方法相同, 在此略去.

在熟悉了使用母函数求解递推关系的流程后, 我们现在重新使用母函数求解 Fibonacci 数列 $\{F_n\}$, 解决 4.2 节末留下的问题. 回顾 $\{F_n\}$ 满足的递推关系:

$$\begin{cases} F_n = F_{n-1} + F_{n-2} & (n \geq 2) \\ F_0 = 0, F_1 = 1 \end{cases}$$

构造 Fibonacci 数列的母函数:

$$G(x) = F_0 + F_1x + F_2x^2 + \cdots = \sum_{n=0}^{\infty} F_n x^n$$

由递推式可得

$$[x^n]G(x) - [x^n](xG(x)) - [x^n](x^2G(x)) = 0 \quad (n \geq 2)$$

比较母函数展开式中的各项系数:

$$\begin{aligned} G(x) &= F_0 + F_1x + F_2x^2 + F_3x^3 + \cdots \\ xG(x) &= F_0x + F_1x^2 + F_2x^3 + \cdots \\ x^2G(x) &= F_0x^2 + F_1x^3 + \cdots \end{aligned}$$

可得

$$(1 - x - x^2)G(x) = F_0 + (F_1 - F_0)x = x$$

由此解得

$$G(x) = \frac{x}{1 - x - x^2} \quad (4-18)$$

上式即为 Fibonacci 数列母函数的表达式. 为了将上式展开为级数形式, 与之前讨论的例子类似, 我们可先将其拆分为形如 $\frac{1}{1-ax}$ 的分式:

$$\frac{x}{1 - x - x^2} = \frac{P}{1 - \alpha x} + \frac{Q}{1 - \beta x}$$

首先需要解出 α, β , 然后再用待定系数法解出 P, Q . α, β 的取值应满足

$$1 - x - x^2 = (1 - \alpha x)(1 - \beta x)$$

作换元 $x = \frac{1}{y}$, 得

$$1 - \frac{1}{y} - \frac{1}{y^2} = \left(1 - \frac{\alpha}{y}\right) \left(1 - \frac{\beta}{y}\right)$$

等式两侧同乘 y^2 , 得

$$y^2 - y - 1 = (y - \alpha)(y - \beta)$$

这表明 α, β 是方程

$$y^2 - y - 1 = 0 \quad (4-19)$$

的两个根. 求解此方程, 得到

$$\alpha = \frac{1 + \sqrt{5}}{2}, \quad \beta = \frac{1 - \sqrt{5}}{2}$$

得到 α, β 后, 再使用待定系数法求解 P, Q :

$$G(x) = \frac{P}{1-\alpha x} + \frac{Q}{1-\beta x} = \frac{(P+Q) - (\beta P + \alpha Q)x}{1-x-x^2}$$

将上式与式 (4-18) 比较, 可得

$$\begin{cases} P+Q=0 \\ \frac{1-\sqrt{5}}{2}P + \frac{1+\sqrt{5}}{2}Q = -1 \end{cases}$$

求解此方程组, 得到

$$P = \frac{1}{\sqrt{5}}, Q = -\frac{1}{\sqrt{5}}$$

综上, 我们可以写出 $G(x)$ 的展开式:

$$\begin{aligned} G(x) &= \frac{1}{\sqrt{5}} \left(1 - \frac{1+\sqrt{5}}{2}x \right)^{-1} - \frac{1}{\sqrt{5}} \left(1 - \frac{1-\sqrt{5}}{2}x \right)^{-1} \\ &= \frac{1}{\sqrt{5}} \sum_{n=0}^{\infty} \left(\frac{1+\sqrt{5}}{2} \right)^n x^n - \frac{1}{\sqrt{5}} \sum_{n=0}^{\infty} \left(\frac{1-\sqrt{5}}{2} \right)^n x^n \\ &= \sum_{n=0}^{\infty} \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right] x^n \end{aligned}$$

从而由母函数的定义可得

$$F_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right]$$

这就是 Fibonacci 数列的通项表达式.

对比上述求解过程与 4.2 节中的求解过程. 首先, 方程 (4-19) 与 (4-9) 具有完全相同的形式, 它们有相同的解 α, β . 在 4.2 节中, 我们通过观察得出结论, F_n 的通项表示应当具有式 (4-10) 所示的形式:

$$F_n = A\alpha^n + B\beta^n$$

在上面的推导中则可以明确看出, α^n, β^n 并非空穴来风, 而是来源于

$$G(x) = \frac{P}{1-\alpha x} + \frac{Q}{1-\beta x}$$

展开式中的 x^n 项系数. 由此我们可以回答 4.2 节末留下的问题: 一个满足递推式的几何数列 $\{q^n\}$, 实际上对应于母函数拆分出的形如 $\frac{1}{1-qn}$ 的分式的展开式中的各项系数.

从上面讨论的几个例子中可以看出, 母函数为求解线性常数系数递推关系提供了一套标准化的解决方案, 本章后面两节将着重讨论这一内容. 在此之前, 我们先展示如何使用母函数求解其他类型的递推关系, 它们并非本章的重点, 所以我们仅以几个例子作简单讨论.

例 4.3.4. 一个内存分配器管理着一段长度为 n 的连续内存，内存单元按地址从低到高编号为 1 到 n ，初始时全部内存为空闲状态。用户通过该分配器不断申请长度为 2 的连续内存区间；对于每次申请，分配器会在所有可能的分配位置中均匀随机地选择一个进行分配。这一过程一直持续到没有长度为 2 的连续内存区间可供分配为止。最终，整段内存中可能会存在若干不连续的、未被分配的内存单元。求未被分配的内存单元数目的期望值。

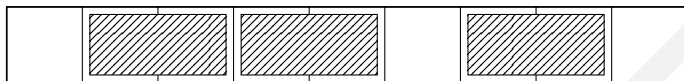


图 4.3: 一个内存分配器外碎片的例子

这个例子涉及到计算机科学中经典的内存外碎片问题，即内存中虽然存在一些空闲区域，但它们分散在各处，导致没有足够大的连续内存区间可供分配。图 4.3 中展示了一个 $n = 9$ 时的例子：内存中已经分配了 3 段长度为 2 的内存区间，剩下 3 个未被分配的单元；虽然空闲内存单元的总数大于 2，但此时已经没有长度为 2 的连续区间可供分配了。

设一段长度为 n 的空闲内存分配到最后，剩余未分配内存单元数的期望为 a_n 。为了构建递推关系，我们考虑在这段内存上执行的第一次内存分配：分配出的内存区间的起始地址均匀随机地在

$$\{1, 2, \dots, n-1\}$$

中取值。若分配的地址为 k ，则此次分配会将原先一整段长度为 n 的空闲内存切分为左、右两段空闲内存，长度分别为 $k-1$ 和 $n-k-1$ 。之后，内存分配会继续在左、右两段空闲内存中分别进行，而最终剩余未分配内存单元数的期望也就等于左、右两段内存中未分配单元数的期望之和。

根据以上推导，我们可以建立关于 a_n 的递推关系：

$$a_n = \frac{1}{n-1} [(a_0 + a_{n-2}) + (a_1 + a_{n-3}) + \dots + (a_{n-2} + a_0)] \quad (n \geq 2)$$

化简：

$$(n-1)a_n = \sum_{k=0}^{n-2} [a_k + a_{(n-2)-k+1}] = 2 \sum_{k=0}^{n-2} a_k$$

上述递推关系中包含一个和式，我们希望消除这个和式以方便求解。注意到递推关系对 a_{n-1} 也同样成立：

$$(n-2)a_{n-1} = \sum_{k=0}^{n-3} a_k$$

以上两式相减，得

$$(n-1)a_n - (n-2)a_{n-1} - 2a_{n-2} = 0$$

这是一个二阶齐次线性变系数递推关系，其初值为 $a_0 = 0$, $a_1 = 1$, $a_2 = 0$ 。

之前的例子中, 我们通过对母函数进行四则运算, 能够求解常系数递推关系. 若递推式中的系数不是常数, 而是与 n 有关, 则可以考虑对母函数进行求导或积分, 从而向系数中引入 n . 设

$$G(x) = \sum_{n=1}^{\infty} a_n x^{n-1} = a_1 + a_2 x + a_3 x^2 + a_4 x^3 + \cdots$$

递推关系中 a_n 前带有系数 $(n-1)$, 为此可对 $G(x)$ 求导:

$$G'(x) = \sum_{n=2}^{\infty} (n-1) a_n x^{n-2} = a_2 + 2a_3 x + 3a_4 x^2 + \cdots$$

比较母函数展开式中的各项系数:

$$\begin{aligned} G'(x) &= a_2 + 2a_3 x + 3a_4 x^2 + 4a_5 x^3 + \cdots \\ xG'(x) &= a_2 x + 2a_3 x^2 + 3a_4 x^3 + \cdots \\ 2xG(x) &= 2a_1 x + 2a_2 x^2 + 2a_3 x^3 + \cdots \end{aligned}$$

由此可得

$$G'(x) - xG'(x) - 2xG(x) = 0$$

即

$$G'(x) - \frac{2x}{1-x} G(x) = 0$$

上式是一个关于 $G(x)$ 的一阶齐次线性微分方程. 它有通解

$$\begin{aligned} G(x) &= C \cdot \exp\left(\int \frac{2x}{1-x} dx\right) \\ &= C \cdot e^{-2x-2\ln(x-1)} \\ &= C \cdot \frac{e^{-2x}}{(x-1)^2} = C \cdot \frac{e^{-2x}}{(1-x)^2} \end{aligned}$$

利用边界条件 $G(0) = a_1 = 1$, 我们可求出通解中的常数 $C = 1$. 因此,

$$G(x) = \frac{e^{-2x}}{(1-x)^2}$$

展开:

$$\begin{aligned} G(x) &= e^{-2x} \cdot \frac{1}{(1-x)^2} \\ &= \left[\sum_{n=0}^{\infty} \frac{(-2)^n}{n!} x^n \right] \left[\sum_{n=0}^{\infty} (n+1)x^n \right] \\ &= \sum_{n=0}^{\infty} \left[\sum_{k=0}^n \frac{(-2)^k}{k!} \cdot (n-k+1) \right] x^n \end{aligned}$$

因此, 我们最终有

$$a_n = \sum_{k=0}^{n-1} \frac{(-2)^k (n-k)}{k!} \quad (4-20)$$

分析式 (4-20) 可知, 当内存空间足够大时, 这个例子中的随机内存分配器的期望效率是

$$1 - \lim_{n \rightarrow \infty} \sum_{k=0}^{n-1} \frac{(-2)^k}{k!} \left(1 - \frac{k}{n}\right) = 1 - e^{-2}$$

这个值大致是 86.47%. 显然, 随机分配并不是一种效率很高的分配算法.

从这个例子可以看出, 若对变系数递推关系套用母函数方法求解, 则需要先对母函数求导或积分凑出递推式中的系数, 再使用高等数学方法解微分方程求出 $G(x)$. 详细的讨论将会超出本书的范畴.

例 4.3.5. 设集合 $\{1, 2, \dots, n\}$ 的一个排列为 $p_1 p_2 \dots p_n$, 若此排列满足

$$p_1 < p_2 > p_3 < p_4 > \dots < p_{n-1} > p_n$$

则称其为交错排列. 求交错排列的方案数.

设交错排列的方案数为 a_n , 数列 $\{a_n\}$ 指数型母函数为

$$G_e(x) = a_0 + a_1 x + \frac{a_2}{2!} x^2 + \frac{a_3}{3!} x^3 + \dots$$

显然, 当 n 为偶数时, 没有满足上述形式的交错排列, 从而 $a_n = 0$. 当 $n > 1$ 时, 设 $p_i = n$, 则 p_i 必定大于交错排列中其相邻的两个元素, 即

$$p_1 < p_2 > p_3 < \dots > p_{i-1} < p_i > p_{i+1} < p_{i+2} > \dots > p_n$$

我们可将排列的前 $i-1$ 位和后 $n-i$ 位分别看成两个独立的交错排列, 将二者组合并在中间插入 $p_i = n$, 即可得到一个 n 阶交错排列. 因此, 我们有

$$a_n = \sum_{i=1}^n \binom{n-1}{i-1} a_{i-1} a_{n-i} = \sum_{i=0}^{n-1} \binom{n-1}{i} a_i a_{n-i-1}$$

在等式两侧同除以 $(n-1)!$, 得

$$\frac{a_n}{(n-1)!} = \sum_{i=0}^{n-1} \frac{a_i}{i!} \cdot \frac{a_{n-i-1}}{(n-i-1)!}$$

注意到等式右侧是 $G_e(x)$ 与自身的卷积形式; 将上式两侧分别对 n 求和, 再利用 $a_1 = 1$, 我们即得到关于 $G_e(x)$ 的微分方程:

$$G'_e(x) = G_e(x)^2 + 1$$

解此微分方程:

$$\begin{aligned}\frac{dG_e}{dx} &= G_e^2 + 1 \\ \frac{dG_e}{G_e^2 + 1} &= dx \\ \arctan G_e &= x + C \\ G_e &= \tan(x + C)\end{aligned}$$

由 $G_e(0) = a_0 = 0$ 可解得 $C = 0$, 因此 a_n 的指数型母函数为 $G_e(x) = \tan x$.

$\tan x$ 的 Maclaurin 展开没有简单的封闭形式, 因此数列 $\{a_n\}$ 也没有对应的简单通项表达式. 这个例子表明, 母函数或指数型母函数有简单的封闭形式, 并不意味着其对应的数列也会有简单的封闭形式, 反之亦然.

4.4 齐次线性常系数递推关系

本节中, 我们讨论一般的齐次线性常系数递推关系的求解方式. 在线性常系数递推关系的定义式 (4-1) 中令非齐次项 $b_n \equiv 0$, 得到

$$a_n + C_1 a_{n-1} + C_2 a_{n-2} + \cdots + C_k a_{n-k} = 0 \quad (4-21)$$

方便起见, 我们令 $C_0 = 1$, 将上式表示为和式形式:

$$\sum_{j=0}^k C_j a_{n-j} = 0$$

这是一个 k 阶递推式, 它表明 a_n 的值依赖于 $a_{n-1}, a_{n-2}, \cdots, a_{n-k}$ 这 k 项的值. 为使递推关系能够成立, 我们必须同时给出数列前 k 项

$$a_0, a_1, a_2, \cdots, a_{k-1}$$

的值, 才能递推出此后每一项的值. 设这些项的初值分别是 $t_0, t_1, \cdots, t_{k-1}$, 则我们得到

$$\begin{cases} a_n + C_1 a_{n-1} + C_2 a_{n-2} + \cdots + C_k a_{n-k} = 0 & (n \geq k) \\ a_n = t_n & (0 \leq n < k) \end{cases}$$

此即为齐次线性常系数递推关系的一般形式.

4.3 节中展示了几个例子, 其中涉及使用母函数求解一些简单的一阶和二阶递推关系. 实际上, 母函数方法可以推广到任意有限高阶的线性常系数递推关系上, 下面我们对此作详细的讨论.

4.4.1 特征多项式

考虑由式 (4-21) 给出的关于数列 $\{a_n\}$ 的 k 阶齐次线性常系数递推关系. 为了求解此递推关系, 首先构造 $\{a_n\}$ 的母函数:

$$G(x) = a_0 + a_1x + a_2x^2 + \cdots = \sum_{n=0}^{\infty} a_n x^n$$

由递推式

$$\sum_{j=0}^k C_j a_{n-j} = 0$$

可得

$$\sum_{j=0}^k [x^{n-j}]G(x) = \sum_{j=0}^k [x^n] (x^j G(x)) = 0 \quad (n \geq k)$$

比较母函数展开式中的各项系数:

$$\begin{array}{rcl} G(x) = & a_0 + & a_1x + a_2x^2 + a_3x^3 + \cdots + a_kx^k + \cdots \\ C_1xG(x) = & C_1a_0x + C_1a_1x^2 + C_1a_2x^3 + \cdots + C_1a_{k-1}x^k + \cdots \\ C_2x^2G(x) = & C_2a_0x^2 + C_2a_1x^3 + \cdots + C_2a_{k-2}x^k + \cdots \\ \cdots & & \cdots \\ C_kx^kG(x) = & & C_ka_0x^k + \cdots \end{array}$$

可见若将上面 $k+1$ 个等式相加, 则只有比 x^k 更低次的项系数非零. 由此, 我们有

$$(1 + C_1x + C_2x^2 + \cdots + C_kx^k) G(x) = \sum_{n=0}^{k-1} \sum_{j=0}^n C_j a_{n-j} x^n$$

简便起见, 我们用 $P(x)$ 表示等式右侧的多项式:

$$P(x) = \sum_{m=0}^{k-1} \sum_{j=0}^m C_j a_{m-j} x^m$$

$P(x)$ 具体取决于数列 $\{a_n\}$ 满足的递推关系中的各项系数, 以及数列的前 k 项的值, 但由上式可知它必定是一个关于 x 的至多 $k-1$ 次的多项式. 利用 $P(x)$, 我们现在可解出 $G(x)$:

$$G(x) = \frac{P(x)}{1 + C_1x + C_2x^2 + \cdots + C_kx^k} \quad (4-22)$$

为了将 $G(x)$ 展开为级数形式, 我们可仿照 4.3 节中的做法, 将 $G(x)$ 表示为若干形如 $\frac{1}{1-ax}$ 的分式之和的形式, 再利用 Maclaurin 展开求出各项系数. 为此, 我们首先在复数域 \mathbb{C} 上对分母作因式分解:

$$1 + C_1x + C_2x^2 + \cdots + C_kx^k = (1 - \alpha_1x)(1 - \alpha_2x) \cdots (1 - \alpha_kx)$$

作换元 $x = \frac{1}{y}$, 得

$$1 + \frac{C_1}{y} + \frac{C_2}{y^2} + \cdots + \frac{C_k}{y^k} = (1 - \frac{\alpha_1}{y})(1 - \frac{\alpha_2}{y}) \cdots (1 - \frac{\alpha_k}{y})$$

等式两侧同乘 y^k , 得

$$y^k + C_1 y^{k-1} + C_2 y^{k-2} + \cdots + C_k = (y - \alpha_1)(y - \alpha_2) \cdots (y - \alpha_k)$$

可见, $\alpha_1, \alpha_2, \cdots, \alpha_k$ 即为方程

$$C(x) = x^k + C_1 x^{k-1} + C_2 x^{k-2} + \cdots + C_k = 0 \quad (4-23)$$

的全部 k 个根.

对比式 (4-23) 和最初的数列递推式 (4-21), 可以看出二者对应项的系数和次数均相同:

$$\begin{aligned} a_n + C_1 a_{n-1} + C_2 a_{n-2} + \cdots + C_k a_{n-k} &= 0 \\ x^k + C_1 x^{k-1} + C_2 x^{k-2} + \cdots + C_k &= 0 \end{aligned}$$

这表明, 给定递推关系式 (4-21), 我们即能直接写出多项式 $C(x)$, 而省略中间较为繁琐的母函数推导. 由此可见, 多项式 $C(x)$ 在求解线性常系数递推关系的过程中发挥重要的桥梁作用, 它称为递推关系的特征多项式 (characteristic polynomial). $C(x) = 0$ 则称为线性常系数递推关系的特征方程.

4.4.2 通过特征多项式求解齐次线性常系数递推关系

多项式 $C(x)$ 共有 k 个根, 其中可能存在多重根, 根的重数决定了将 $G(x)$ 表示为分式之和时结果的形式. 为此, 我们按如下方式改写 $C(x)$:

$$C(x) = (x - \alpha_1)^{k_1} (x - \alpha_2)^{k_2} \cdots (x - \alpha_m)^{k_m}$$

其中

$$\begin{aligned} k_1, k_2, \cdots, k_m &\geq 1 \\ k_1 + k_2 + \cdots + k_m &= k \end{aligned}$$

并且

$$\alpha_1, \alpha_2, \cdots, \alpha_m \neq 0$$

是 $C(x)$ 的互不相同的根. 由此我们有

$$G(x) = \frac{P(x)}{C(x)} = \frac{P(x)}{(1 - \alpha_1 x)^{k_1} (1 - \alpha_2 x)^{k_2} \cdots (1 - \alpha_m x)^{k_m}}$$

由于 $P(x)$ 的次数不高于 $k-1$, 因此 $G(x)$ 是一真分式, 从而其必能分解为形如 $\frac{A_{ij}}{(1-\alpha_i x)^j}$ 的部分分式 (partial fraction) 之和:

$$\begin{aligned} G(x) &= \frac{A_{11}}{1-\alpha_1 x} + \frac{A_{12}}{(1-\alpha_1 x)^2} + \cdots + \frac{A_{1k_1}}{(1-\alpha_1 x)^{k_1}} \\ &\quad + \frac{A_{21}}{1-\alpha_2 x} + \frac{A_{22}}{(1-\alpha_2 x)^2} + \cdots + \frac{A_{2k_2}}{(1-\alpha_2 x)^{k_2}} \\ &\quad + \cdots \\ &\quad + \frac{A_{m1}}{1-\alpha_m x} + \frac{A_{m2}}{(1-\alpha_m x)^2} + \cdots + \frac{A_{mk_m}}{(1-\alpha_m x)^{k_m}} \\ &= \sum_{i=1}^m \sum_{j=1}^{k_i} \frac{A_{ij}}{(1-\alpha_i x)^j} \end{aligned} \quad (4-24)$$

其中 A_{ij} 均是常数. 关于这里的部分分式分解, 我们有如下结论成立:

引理 4.9. 真有理分式

$$G(x) = \frac{P(x)}{C(x)}$$

能唯一地表示为部分分式之和.

证明. 对 $C(x)$ 的次数 k 施加归纳. 当 $k=1$ 时, $P(x)$ 只能是零次多项式 (即常数), 这时 $\frac{P(x)}{C(x)}$ 本身即为一个部分分式, 命题成立. 当 $k>1$ 时, 设 α_i 是 $C(x)$ 的 k_i 重根:

$$C(x) = (x - \alpha_i)^{k_i} C_i(x) \quad (k_i \geq 1)$$

其中 $C_i(x)$ 是一个 $k-k_i$ 次多项式, 且满足 $C_i(\alpha_i) \neq 0$. 由此, 我们可将 $G(x)$ 分解为如下形式:

$$G(x) = \frac{A_{ik_i}}{(x - \alpha_i)^{k_i}} + \frac{Q(x)}{(x - \alpha_i)^{k_i-1} C_i(x)} \quad (4-25)$$

其中 A_{ik_i} 是待定系数, $Q(x)$ 是待定多项式. 整理上式, 可得

$$A_{ik_i} C_i(x) + Q(x)(x - \alpha_i) = P(x)$$

令 $x = \alpha_i$, 有

$$A_{ik_i} C_i(\alpha_i) = P(\alpha_i)$$

由于 $C_i(\alpha_i) \neq 0$, 因此我们有

$$A_{ik_i} = \frac{P(\alpha_i)}{C_i(\alpha_i)}$$

从而也就有

$$Q(x) = \frac{P(x) - \frac{P(\alpha_i)}{C_i(\alpha_i)} C_i(x)}{x - \alpha_i} \quad (4-26)$$

注意到

$$P(\alpha_i) - \frac{P(\alpha_i)}{C_i(\alpha_i)} C_i(\alpha_i) = 0$$

因此式 (4-26) 右侧分式的分子一定含有因式 $(x - \alpha_i)$, 从而 $Q(x)$ 也一定是一个多项式. $Q(x)$ 的次数至少比 $P(x)$ 小 1, 因此它是一个至多 $k - 2$ 次的多项式. 根据归纳假设,

$$\frac{Q(x)}{(x - \alpha_i)^{k_i - 1} C_i(x)}$$

能唯一地表示为部分分式之和, 因此由式 (4-25) 可知 $G(x)$ 也能唯一地表示为部分分式之和. \square

对任意实数 β , 均有 Maclaurin 展开式

$$(1 + x)^\beta = \sum_{n=0}^{\infty} \frac{\beta(\beta - 1) \cdots (\beta - n + 1)}{n!} x^n$$

上式可用于展开 $G(x)$ 的部分分式分解中的每一项:

$$\frac{A_{ij}}{(1 - \alpha_i x)^j} = A_{ij} (1 - \alpha_i x)^{-j} = \sum_{n=0}^{\infty} A_{ij} \binom{n + j - 1}{n} \alpha_i^n x^n$$

结合式 (4-24), 我们有

$$\begin{aligned} G(x) &= \sum_{i=1}^m \sum_{j=1}^{k_i} \frac{A_{ij}}{(1 - \alpha_i x)^j} \\ &= \sum_{i=1}^m \sum_{j=1}^{k_i} \sum_{n=0}^{\infty} A_{ij} \binom{n + j - 1}{n} \alpha_i^n x^n \\ &= \sum_{n=0}^{\infty} \left[\sum_{i=1}^m \sum_{j=1}^{k_i} A_{ij} \binom{n + j - 1}{n} \alpha_i^n \right] x^n \end{aligned}$$

根据母函数的定义, 我们立即得到

$$a_n = \sum_{i=1}^m \sum_{j=1}^{k_i} A_{ij} \binom{n + j - 1}{n} \alpha_i^n \quad (4-27)$$

此即为数列 $\{a_n\}$ 的通项表达式, 其中共有 k 个形如 A_{ij} 的待定系数, 我们可结合数列前 k 项的值列出 k 元一次方程组求解.

下面我们通过几个例子展示如何利用特征多项式以及式 (4-27) 求解齐次线性常系数递推关系.

例 4.4.1. 例 4.3.2 描述的汉诺塔问题中, 递推式为:

$$h_n = 2h_{n-1} + 1$$

注意到上式对 h_{n-1} 同样成立:

$$h_{n-1} = 2h_{n-2} + 1$$

两式相减, 得到一个二阶齐次线性常系数递推关系

$$h_n - 3h_{n-1} + 2h_{n-2} = 0$$

初值为 $h_0 = 0, h_1 = 1$.

上述递推关系的特征多项式为

$$C(x) = x^2 - 3x + 2$$

它有两个一重根, 分别是

$$\alpha_1 = 1, \alpha_2 = 2$$

将 $m = 2, k_1 = k_2 = 1$ 和以上两个根代入式 (4-27), 得

$$h_n = A_1 \cdot 1^n + A_2 \cdot 2^n = A_1 + A_2 \cdot 2^n$$

再利用数列初值建立关于待定系数 A_1, A_2 的方程组:

$$\begin{cases} A_1 + A_2 = h_0 = 0 \\ A_1 + 2A_2 = h_1 = 1 \end{cases}$$

求解此方程组, 得

$$A_1 = -1, A_2 = 1$$

因此,

$$h_n = 2^n - 1$$

这与我们之前使用母函数求得的结果相同.

例 4.4.2. 求解递推关系:

$$\begin{cases} a_n = 3a_{n-1} - 3a_{n-2} + a_{n-3} & (n \geq 3) \\ a_0 = 0, a_1 = 1, a_2 = 2 \end{cases}$$

此递推关系对应的特征多项式为

$$C(x) = x^3 - 3x^2 + 3x - 1 = (x - 1)^3$$

它有一个三重根

$$\alpha = 1$$

将 $m = 1, k_1 = 3$ 和 $\alpha = 1$ 代入式 (4-27) 中, 得

$$\begin{aligned} a_n &= A_1 \cdot 1^3 + \binom{n+1}{n} A_2 \cdot 1^3 + \binom{n+2}{n} A_3 \cdot 1^3 \\ &= A_1 + (n+1)A_2 + \frac{(n+2)(n+1)}{2} A_3 \end{aligned}$$

再利用数列初值建立关于待定系数 A_1, A_2, A_3 的方程组:

$$\begin{cases} A_1 + A_2 + A_3 = 0 \\ A_1 + 2A_2 + 3A_3 = 1 \\ A_1 + 3A_2 + 6A_3 = 2 \end{cases}$$

解得 $A_1 = -1, A_2 = 1, A_3 = 0$. 因此,

$$a_n = -1 + (n+1) = n$$

我们将本小节讨论的一般形式的齐次线性常系数递推关系的求解流程总结如下.

- (1) 利用母函数求出递推式对应的特征多项式 $C(x)$, $C(x)$ 的形式与递推式之间有一个直观的对应关系, 因此实际上可根据递推式直接写出 $C(x)$;
- (2) 求解 $C(x)$ 的根、对母函数作部分分式分解并展开为级数形式, 从而获得式 (4-27) 所示的数列通项表示.

由于式 (4-27) 中含有组合数, 求解其中的待定系数需要一定计算量, 因此实践中我们一般还会对此式作进一步化简. 为此, 我们需要对特征多项式 $C(x)$ 的根的情况作分类讨论. 通常情况下, 特征多项式是实系数多项式, 它可能有若干实数根和若干对共轭复数根; 此外, 实数根和复数根都有可能是多重根.

下面我们针对每种根的情况分别讨论. 为方便起见, 不失一般性地, 在讨论时, 我们假设 $C(x)$ 仅有我们正在讨论的那种类型的根. 若 $C(x)$ 有多种类型的根, 由式 (4-27) 可知, 将每种类型的根对应的解相加即可得到原递推式的解, 再根据数列初值解出待定系数即可.

特征多项式有 k 个一重实根时解的形式

若特征多项式有 k 个两两不同的一重实根, 则我们有

$$C(x) = (x - \alpha_1)(x - \alpha_2) \cdots (x - \alpha_k)$$

母函数则可表示为如下形式:

$$G(x) = \sum_{j=1}^k \frac{A_j}{1 - \alpha_j x} = \frac{A_1}{1 - \alpha_1 x} + \frac{A_2}{1 - \alpha_2 x} + \cdots + \frac{A_k}{1 - \alpha_k x}$$

对每一项分别作 Maclaurin 展开, 即得到 a_n 的通项表达式:

$$a_n = \sum_{j=1}^k A_j \alpha_j^n$$

待定系数 A_1, A_2, \dots, A_k 可由线性方程组

$$\begin{cases} A_1 + A_2 + \dots + A_k = a_0 \\ A_1 \alpha_1 + A_2 \alpha_2 + \dots + A_k \alpha_k = a_1 \\ \dots \\ A_1 \alpha_1^{k-1} + A_2 \alpha_2^{k-1} + \dots + A_k \alpha_k^{k-1} = a_{k-1} \end{cases}$$

解出. 此方程组的系数矩阵是著名的 Vandermonde 矩阵; 由于 $\alpha_1, \alpha_2, \dots, \alpha_k$ 彼此不同且均非零, 因此 Vandermonde 矩阵的行列式值也非零:

$$\begin{vmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_k \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \dots & \alpha_k^{k-1} \end{vmatrix} = \prod_{1 \leq j < i \leq n} (\alpha_i - \alpha_j) \neq 0$$

因此, 由上述方程组必能解出唯一的一组 A_1, A_2, \dots, A_k .

例 4.4.3. 如图 4.4 所示, 一个圆被它的 $n \geq 2$ 条半径切分为 n 个扇形区域, 这些区域依次编号为 D_1, D_2, \dots, D_n . 现有 k 种颜色, 对这 n 个区域进行着色, 要求相邻区域的颜色不同, 求着色方案数.

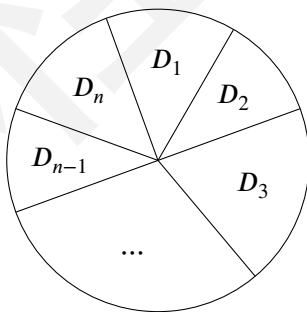


图 4.4: 例 4.4.3 示意图

设染色方案数为 a_n . 为了建立递推关系, 我们讨论 D_1 和 D_{n-1} 两个区域所选用的颜色是否相同, 将染色方案分为如下两类:

- (1) D_1 和 D_{n-1} 颜色相同: 这时 D_n 将有 $k-1$ 种颜色可用; 此时, 从 D_1 到 D_{n-2} 这 $n-2$ 个区域的着色方案, 等价于将圆切分为 $n-2$ 个区域的着色方案;
- (2) D_1 和 D_{n-1} 颜色不同, 这时 D_n 将有 $k-2$ 种颜色可用; 此时, 从 D_1 到 D_{n-1} 这 $n-1$ 个区域的着色方案, 等价于将圆切分为 $n-1$ 个区域的着色方案.

由此我们得到如下递推式:

$$a_n = (k-2)a_{n-1} + (k-1)a_{n-2}$$

当 $n=2$ 或 $n=3$ 时, 每个扇形区域都和其他所有扇形相邻, 因此可以直接通过乘法原理计算染色方案, 分别为

$$a_2 = k(k-1), a_3 = k(k-1)(k-2)$$

为了方便计算, 可利用 a_2, a_3 和递推式, 逆推出 a_0, a_1 作为数列初值:

$$a_0 = k, a_1 = 0$$

需要注意, a_0, a_1 并非真的是数列的最初两项, 它们没有实际的组合意义, 我们仅是为了方便求解递推关系式中的待定系数才计算出这两个值.

递推式的特征多项式为

$$C(x) = x^2 - (k-2)x - (k-1)$$

它的两个根分别为

$$x_1 = k-1, x_2 = 1$$

因此, 我们有

$$a_n = A_1(k-1)^n + A_2(-1)^n$$

代入初值:

$$\begin{cases} A_1 + A_2 = k \\ (k-1)A_1 - A_2 = 0 \end{cases}$$

解得

$$A_1 = 1, A_2 = k-1$$

由此可得

$$a_n = (k-1)^n + (k-1)(-1)^n \quad (n \geq 2)$$

我们最后简单解释这个例子中要从 $n=2$ 开始计算数列初值的原因. 在讨论 D_1 和 D_{n-1} 两个区域的颜色时, 我们实际上假定这两个区域是被其他区域完全分隔开的. 因此, 为了使我们的讨论成立, 至少要有 $n=4$ 个区域; 又因为递推关系是二阶的, 所以数列初值即为 a_2, a_3 . 这提示我们, 递推关系的初值与构建递推式时使用的组合场景密切相关.

例 4.4.4. 在如图 4.5 所示的、由 $3n-2$ 个阻值均为 R 的电阻构成的网络中, 求 x_0, x'_0 两点间的等效电阻值.

这个例子与电学有关, 我们可以建立递推关系求解等效电阻值. 设 x_0, x'_0 两点间的等效电阻值为 R_n . R_n 对应的电阻网络可以看作一个阻值为 R_{n-1} 的电阻和三个阻值为 R 的电阻按如图 4.6 所示的方式串并联构成的.

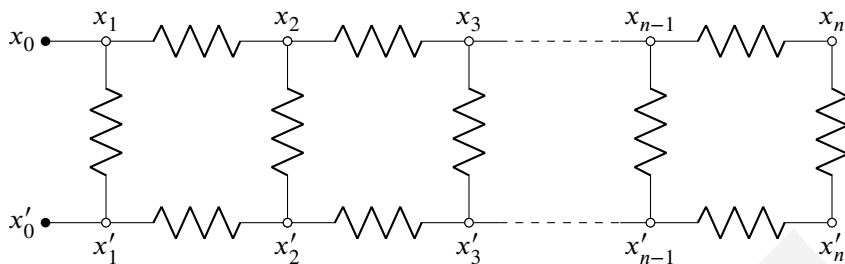


图 4.5: 例 4.4.4 示意图

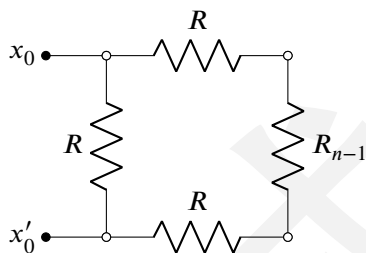


图 4.6: 例 4.4.4 中电阻网络的等效示意图

根据并联电阻计算公式，我们有

$$\frac{1}{R_n} = \frac{1}{R} + \frac{1}{2R + R_{n-1}}$$

这已经是一个关于 R_n 的递推关系，但我们还需对其作进一步处理，将其变形为一个线性常系数递推关系。对上式两侧分别取倒数，得

$$R_n = \frac{R(2R + R_{n-1})}{3R + R_{n-1}}$$

现在我们构造两个辅助数列来帮助求出 R_n 。令

$$R_n = \frac{a_n}{b_n}$$

代入前式中，得

$$\frac{a_n}{b_n} = \frac{R \left(2R + \frac{a_{n-1}}{b_{n-1}} \right)}{3R + \frac{a_{n-1}}{b_{n-1}}} = \frac{2R^2 b_{n-1} + R a_{n-1}}{3R b_{n-1} + a_{n-1}}$$

由此我们得到一组 a_n 、 b_n 可以满足的递推关系：

$$\begin{cases} a_n = 2R^2 b_{n-1} + R a_{n-1} \\ b_n = 3R b_{n-1} + a_{n-1} \end{cases}$$

这是一个涉及两个数列的齐次线性常系数递推关系, 可用消元法求解. 由第二个递推式可知

$$a_{n-1} = b_n - 3Rb_{n-1}$$

代入第一个递推式中, 得

$$b_{n+1} - 3Rb_n = 2R^2b_{n-1} - R(b_n - 3Rb_{n-1})$$

整理, 得

$$b_{n+1} - 4Rb_n + R^2b_{n-1} = 0$$

递推关系初值 b_1, b_2 可由 R_1, R_2 推出. 简单计算可知

$$R_1 = R, R_2 = \frac{3}{4}R$$

结合量纲分析, 我们可选取 $b_1 = 1, b_2 = 4R$ 作为初值. 由此即可求解 b_n . 上述关于 $\{b_n\}$ 的递推式的特征方程为

$$C(x) = x^2 - 4Rx + R^2$$

其根为

$$x_1 = (2 + \sqrt{3})R, x_2 = (2 - \sqrt{3})R$$

这是两个一重实根, 因此我们有

$$b_n = \left[A(2 + \sqrt{3})^n + B(2 - \sqrt{3})^n \right] R^n$$

其中 A, B 是待定系数. 代入初值:

$$\begin{cases} (2 + \sqrt{3})A + (2 - \sqrt{3})B = \frac{1}{R} \\ (7 + 4\sqrt{3})A + (7 - 4\sqrt{3})B = \frac{4}{R} \end{cases}$$

解得

$$A = \frac{1}{2\sqrt{3}R}, B = -\frac{1}{2\sqrt{3}R}$$

因此, 我们有

$$b_n = \frac{R^{n-1}}{2\sqrt{3}} \left[(2 + \sqrt{3})^n - (2 - \sqrt{3})^n \right]$$

由此, 我们可继续解出

$$\begin{aligned} a_n &= b_{n+1} - 3Rb_n \\ &= \frac{R^n}{2\sqrt{3}} \left[(\sqrt{3} - 1)(2 + \sqrt{3})^n + (\sqrt{3} + 1)(2 - \sqrt{3})^n \right] \end{aligned}$$

求得 a_n 和 b_n 的通项表示后, 将二者相除即可得到 R_n :

$$R_n = \frac{(\sqrt{3}-1)(2+\sqrt{3})^n + (\sqrt{3}+1)(2-\sqrt{3})^n}{(2+\sqrt{3})^n - (2-\sqrt{3})^n} \cdot R$$

当 $n \rightarrow \infty$ (即电阻网络无限延长) 时, 我们有

$$\lim_{n \rightarrow \infty} R_n = (\sqrt{3}-1)R$$

上述极限值也可通过证明无穷级数 $\{R_n\}$ 的收敛性, 并求关于 R_∞ 的方程

$$\frac{1}{R_\infty} = \frac{1}{R} + \frac{1}{2R + R_\infty}$$

的解直接得出.

特征多项式有 k 重实根时解的形式

若特征多项式有一个 k 重实根, 即

$$C(x) = (x - \alpha)^k$$

则母函数可表示为如下形式:

$$G(x) = \sum_{j=1}^k \frac{A_j}{(1 - \alpha x)^j}$$

由式 (4-27), 数列的通项表达式为

$$a_n = \sum_{j=1}^k A_j \binom{n+j-1}{n} \alpha^n = \sum_{j=1}^k A_j \binom{n+j-1}{j-1} \alpha^n$$

注意到 $\binom{n+j-1}{j-1}$ 是关于 n 的 $j-1$ 次多项式, 因此

$$\sum_{j=1}^k A_j \binom{n+j-1}{j-1}$$

至多是 n 的 $k-1$ 次多项式. 换句话说, a_n 是 α^n 与 n 的至多 $k-1$ 的多项式的乘积, 我们只需求出这个关于 n 的多项式即可求得 a_n . 为此, 我们可以直接利用待定系数法设出这个多项式:

$$a_n = (A_0 + A_1 n + A_2 n^2 + \cdots + A_{k-1} n^{k-1}) \alpha^n \quad (4-28)$$

上式相当于将二项式系数并入了设出的待定系数中, 减少了计算量.

得到式 (4-28) 后, 我们可继续对其作改进, 从而进一步削减计算量. 为此, 我们需要利用如下结论:

引理 4.10. 设 n 为正整数, k 为非负整数. 任何一个关于 n 的至多 k 次的多项式总是能唯一地表示为

$$\binom{n}{0}, \binom{n}{1}, \binom{n}{2}, \dots, \binom{n}{k}$$

的一个线性组合.

证明. $\binom{n}{m}$ 是关于 n 的 m 次多项式. 记

$$\binom{n}{m} = \sum_{j=0}^m A_{mj} n^j \quad (m = 0, 1, 2, \dots, k)$$

其中各 A_{mj} 均为已知量, 且 $A_{mm} \neq 0$. 考虑任意关于 n 的至多 k 次的多项式

$$A(x) = a_0 + a_1 n + a_2 n^2 + \dots + a_k n^k$$

构造方程

$$a_0 + a_1 n + a_2 n^2 + \dots + a_k n^k = b_0 \binom{n}{0} + b_1 \binom{n}{1} + b_2 \binom{n}{2} + \dots + b_k \binom{n}{k}$$

比较 n 的各次项系数, 可得 $k+1$ 元线性方程组

$$\sum_{j=0}^n A_{jm} b_j = a_m \quad (m = 0, 1, 2, \dots, k)$$

此方程组的系数矩阵是上三角矩阵, 其行列式值为

$$\prod_{m=1}^k A_{mm} \neq 0$$

因此方程组必有唯一解. □

利用上述结论, 我们可以将式 (4-28) 中的 n^j ($j = 0, 1, 2, \dots, k-1$) 直接替换为 $\binom{n}{j}$, 得到另一种含有待定系数的 a_n 的表达式:

$$a_n = \left[A_0 \binom{n}{0} + A_1 \binom{n}{1} + A_2 \binom{n}{2} + \dots + A_{k-1} \binom{n}{k-1} \right] \alpha^n \quad (4-29)$$

这种表示方式的优点在于, 一般情况下, 我们会选取数列的前 k 项

$$a_0, a_1, a_2, \dots, a_{k-1}$$

用于求解待定系数. 将 $n = 0, 1, 2, \dots, k-1$ 分别代入上式, 可以发现只有前 n 个二项式系数的值非零. 这时, 方程组的系数矩阵实际上变为一个下三角矩阵, 从而能够极大地简化求解过程.

例 4.4.5. 求解递推关系:

$$\begin{cases} a_n = 4a_{n-1} - 4a_{n-2} & (n \geq 2) \\ a_0 = 1, a_1 = 4 \end{cases}$$

递推关系的特征多项式为

$$C(x) = x^2 - 4x + 4 = (x - 2)^2$$

它有一个二重根 $x = 2$. 因此我们可以设

$$a_n = (A_1 + A_2 n) 2^n$$

代入初值:

$$\begin{cases} A_1 = 1 \\ 2(A_1 + A_2) = 4 \end{cases}$$

解得

$$a_1 = a_2 = 1$$

因此,

$$a_n = (n + 1) 2^n$$

例 4.4.6. 求解下列 n 阶行列式:

$$d_n = \begin{vmatrix} 2 & 1 & 0 & \cdots & 0 & 0 \\ 1 & 2 & 1 & \cdots & 0 & 0 \\ 0 & 1 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & 1 \\ 0 & 0 & 0 & \cdots & 1 & 2 \end{vmatrix}$$

$\underbrace{\hspace{10em}}_{n \text{ 阶}}$

为了建立递推关系, 我们可按照行列式的定义对其进行展开, 从而将其转化为阶数更小的行列式 (即所谓代数余子式). 对第一行作行展开:

$$d_n = 2A_{11} - A_{12}$$

$$= 2 \begin{vmatrix} 2 & 1 & \cdots & 0 & 0 \\ 1 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 2 & 1 \\ 0 & 0 & \cdots & 1 & 2 \end{vmatrix} - \begin{vmatrix} 1 & 1 & \cdots & 0 & 0 \\ 0 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 2 & 1 \\ 0 & 0 & \cdots & 1 & 2 \end{vmatrix}$$

$\underbrace{\hspace{10em}}_{n-1 \text{ 阶}} \quad \underbrace{\hspace{10em}}_{n-1 \text{ 阶}}$

第一项中的行列式就是 d_{n-1} . 对于第二项, 我们进一步对它的第一列作列展开:

$$\underbrace{\begin{vmatrix} 1 & 1 & \cdots & 0 & 0 \\ 0 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 2 & 1 \\ 0 & 0 & \cdots & 1 & 2 \end{vmatrix}}_{n-1 \text{ 阶}} = \underbrace{\begin{vmatrix} 2 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 2 & 1 \\ 0 & \cdots & 1 & 2 \end{vmatrix}}_{n-2 \text{ 阶}}$$

可见它等于 d_{n-2} . 因此, 我们得到递推关系:

$$d_n = 2d_{n-1} - d_{n-2}$$

分别计算 $n = 1, 2$ 时的一阶和二阶行列式, 可以得到数列初值

$$d_1 = 2, d_2 = 3$$

递推关系的特征多项式为

$$C(x) = x^2 - 2x + 1$$

它有一个二重根 $x = 1$. 因此, 解的形式为

$$d_n = An + B$$

代入初值, 解得

$$d_n = n + 1 \quad (n \geq 1)$$

特征多项式有共轭复根时解的形式

实系数多项式的复数根一定成对出现, 我们可以将一对共轭复根放在一起考虑. 假设特征多项式有一对共轭复根 α_1, α_2 , 我们将其表示为模与辐角的形式:

$$\alpha_1 = \rho(\cos \theta + i \sin \theta)$$

$$\alpha_2 = \rho(\cos \theta - i \sin \theta)$$

由此可以化简 a_n 的通项表达式:

$$\begin{aligned} A_1 \alpha_1^n + A_2 \alpha_2^n &= (A_1 + A_2) \rho^n \cos n\theta + i(A_1 - A_2) \rho^n \sin n\theta \\ &= A \rho^n \cos n\theta + B \rho^n \sin n\theta \end{aligned}$$

其中

$$A = A_1 + A_2, \quad B = i(A_1 - A_2)$$

通过将待定系数 A_1, A_2 转化为 A, B , 我们能够避免在计算实数数列的通项表达式的过程中引入大量涉及复数的计算.

例 4.4.7. 求解递推关系:

$$\begin{cases} a_n = a_{n-1} - a_{n-2} & (n \geq 2) \\ a_0 = a_1 = 1 \end{cases}$$

上述递推式的特征多项式为

$$C(x) = x^2 - x + 1$$

它有两个共轭复根:

$$x_{1,2} = \frac{1}{2} \pm \frac{\sqrt{3}}{2}i = \cos \frac{\pi}{3} \pm i \sin \frac{\pi}{3}$$

因此, 解的形式为

$$a_n = A \cos \frac{n\pi}{3} + B \sin \frac{n\pi}{3}$$

代入初值:

$$\begin{cases} \frac{1}{2}A + \frac{\sqrt{3}}{2}B = 1 \\ -\frac{1}{2}A + \frac{\sqrt{3}}{2}B = 0 \end{cases}$$

解得

$$A = 1, B = \frac{\sqrt{3}}{3}$$

因此, 我们有

$$a_n = \cos \frac{n\pi}{3} + \frac{\sqrt{3}}{3} \sin \frac{n\pi}{3}$$

这表明 $\{a_n\}$ 实际上是一个周期数列.

我们最后简单讨论特征多项式 $C(x)$ 有多重共轭复根的情况. 这时, $C(x)$ 至少是一个四次多项式, 对应于四阶以上的递推关系, 在实际问题中相对罕见. 多重共轭复根相当于多重根和共轭复根的组合, 这些根对 a_n 的通项表达式的贡献也就是多重根和共轭复根各自贡献的乘积. 例如, 如果特征多项式有两个二重共轭复根

$$\alpha_{1,2} = \rho(\cos \theta \pm i \sin \theta)$$

则解的形式为

$$\begin{aligned} a_n &= (A_1 + A_2 n)(A_3 \rho^n \cos n\theta + A_4 \rho^n \sin n\theta) \\ &= A \rho^n \cos n\theta + B \rho^n \sin n\theta + C n \rho^n \cos n\theta + D n \rho^n \sin n\theta \end{aligned}$$

代入数列初值, 解出待定系数 A, B, C, D 即可.

4.5 非齐次线性常系数递推关系

相比于 4.4 节讨论的齐次递推关系, 非齐次递推关系中包含额外的非齐次项 b_n :

$$a_n + C_1 a_{n-1} + C_2 a_{n-2} + \cdots + C_k a_{n-k} = b_n$$

由于 b_n 可以是任意关于 n 的函数, 因此我们实际上无法在一般情况下求解此递推式. 本节中, 我们集中讨论一些形式较为简单的 b_n , 其中仅包含关于 n 的多项式或指数.

少数情况下, b_n 的形式非常简单 (如例 4.3.1、4.3.2 中的 $b_n \equiv 1$), 以至于我们能直接根据经验构造出

$$\frac{Q(x)}{R(x)} = \sum_{n=k}^{\infty} b_n x^n$$

其中 $Q(x), R(x)$ 均为关于 x 的多项式. 这时尽管递推式中有非齐次项, 我们仍可直接套用之前讨论的方法求解 $\{a_n\}$:

$$\begin{array}{rcl} G(x) & = & a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_k x^k + \cdots \\ C_1 x G(x) & = & C_1 a_0 x + C_1 a_1 x^2 + C_1 a_2 x^3 + \cdots + C_1 a_{k-1} x^k + \cdots \\ C_2 x^2 G(x) & = & C_2 a_0 x^2 + C_2 a_1 x^3 + \cdots + C_2 a_{k-2} x^k + \cdots \\ & \cdots & \cdots \\ C_k x^k G(x) & = & C_k a_0 x^k + \cdots \end{array}$$

将上面 $k+1$ 个等式相加, 得

$$\begin{aligned} (1 + C_1 x + C_2 x^2 + \cdots + C_k x^k) G(x) &= \sum_{n=0}^{k-1} \sum_{j=0}^n C_j a_{n-j} x^n + \sum_{n=k}^{\infty} b_n x^n \\ &= P(x) + \frac{Q(x)}{R(x)} \end{aligned}$$

由此可以直接求出

$$G(x) = \frac{P(x)R(x) + Q(x)}{C(x)R(x)}$$

将其展开, 即可得到 $\{a_n\}$ 的通项表示.

通常而言, 给定 b_n , 求出其母函数的分式形式是一个较为复杂的任务. 因此, 在求解非齐次线性常系数递推关系时, 我们一般先将问题转化为方便求解的形式, 再使用母函数、特征多项式等方法求解.

4.5.1 差分法

回顾例 4.4.1 中我们将汉诺塔问题的非齐次递推关系转化为齐次递推关系时的做法. 汉诺塔问题中的递推关系为:

$$h_n - 2h_{n-1} = 1$$

其中非齐次项 b_n 是常数 1, 也即关于 n 的零次多项式. 由于递推关系对于 h_{n-1} 同样成立:

$$h_{n-1} - 2h_{n-2} = 1$$

因此, 将以上两式相减, 将能消掉非齐次项 b_n , 得到一个齐次递推关系:

$$(h_n - 2h_{n-1}) - (h_{n-1} - 2h_{n-2}) = h_n - 3h_{n-1} + 2h_{n-2} = 0$$

上述过程可概括为: 将数列中相邻两项 h_n 与 h_{n-1} 对应的递推式相减, 得到一个新的递推式, 这即是所谓的差分 (differentiation) 操作.

差分法是通过差分操作化简非齐次线性常系数递推关系的方法. 一般地, 若 b_n 是关于 n 的 p 次多项式, 则对递推关系运用至多 $p+1$ 次差分操作, 一定能将其变为齐次的递推关系. 这是因为差分操作会使 b_n 的次数降低 1:

$$\begin{aligned} n^p - (n-1)^p &= n^p - \sum_{j=0}^p \binom{p}{j} (-1)^j n^{p-j} \\ &= \sum_{j=1}^p \binom{p}{j} (-1)^j n^{p-j} \end{aligned}$$

因此, p 次差分操作将使 b_n 变为一个零次多项式, 即一个常数; 再作一次差分操作就可使其变为 0, 此时得到的即为一个齐次递推关系.

下面我们讨论一个经典的求和问题, 展示如何运用差分法求解非齐次线性常系数递推关系.

例 4.5.1. 求正整数的平方和:

$$S_n = \sum_{k=1}^n k^2$$

为了计算 S_n , 我们可先构造一个它满足的递推关系, 然后求解此递推关系以得到 S_n 的通项表达式. 注意到

$$S_n = 1^2 + 2^2 + \cdots + (n-1)^2 + n^2$$

$$S_{n-1} = 1^2 + 2^2 + \cdots + (n-1)^2$$

由此我们得到如下一阶非齐次线性常系数递推关系:

$$S_n - S_{n-1} = n^2$$

其中, 非齐次项 $b_n = n^2$ 是关于 n 的二次多项式. 根据之前的讨论, 对上述递推关系作 3 次差分操作, 即可得到一个齐次递推关系. 我们依次写出每一次差分操作后得到的递推式.

(1) 第一次:

$$(S_n - S_{n-1}) - (S_{n-1} - S_{n-2}) = n^2 - (n-1)^2$$

即

$$S_n - 2S_{n-1} + S_{n-2} = 2n - 1$$

(2) 第二次:

$$(S_n - 2S_{n-1} + S_{n-2}) - (S_{n-1} - 2S_{n-2} + S_{n-3}) = (2n-1) - [2(n-1) - 1]$$

即

$$S_n - 3S_{n-1} + 3S_{n-2} - S_{n-3} = 2$$

(3) 第三次:

$$(S_n - 3S_{n-1} + 3S_{n-2} - S_{n-3}) - (S_{n-1} - 3S_{n-2} + 3S_{n-3} - S_{n-4}) = 2 - 2$$

即

$$S_n - 4S_{n-1} + 6S_{n-2} - 4S_{n-3} + S_{n-4} = 0$$

最终得到的齐次递推关系的特征多项式为

$$C(x) = x^4 - 4x^3 + 6x^2 - 4x + 1 = (x-1)^4$$

它有一个四重根 $x=1$. 由此可写出 a_n 的通项表达式:

$$a_n = A_0 + A_1n + A_2n^2 + A_3n^3$$

代入初值

$$S_0 = 0, S_1 = 1, S_2 = 5, S_3 = 14$$

求解:

$$\begin{cases} A_0 = 0 \\ A_0 + A_1 + A_2 + A_3 = 1 \\ A_0 + 2A_1 + 4A_2 + 8A_3 = 5 \\ A_0 + 3A_1 + 9A_2 + 27A_3 = 14 \end{cases}$$

求解此方程组, 得

$$A_0 = 0, A_1 = \frac{1}{6}, A_2 = \frac{1}{2}, A_3 = \frac{1}{3}$$

因此,

$$\begin{aligned} S_n &= \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n \\ &= \frac{1}{6}(2n^3 + 3n^2 + n) \\ &= \frac{1}{6}n(n+1)(2n+1) \end{aligned}$$

从上面的例子中可以看出, 每进行一次差分操作, 递推式中就多出一项, 其特征多项式也就高出一度. 因此, 对一个非齐次项为 p 阶多项式的非齐次递推关系使用差分法, 将使得最终的特征多项式比最初的递推关系高出 $p+1$ 次. 表面上, 这导致我们必须求解高次

多项式的根；但实际上，由差分法得到的特征多项式非常容易作因式分解。例如，考虑递推式

$$a_n + C_1 a_{n-1} + C_2 a_{n-2} + \cdots + C_k a_{n-k} = b_n$$

它的特征多项式为

$$C(x) = x^n + C_1 x^{n-1} + C_2 x^{n-2} + \cdots + C_k$$

对递推式作一次差分操作，得到

$$\begin{aligned} & a_n + (C_1 - 1) a_{n-1} + (C_2 - C_1) a_{n-2} + \cdots + (C_k - C_{k-1}) a_{n-k} - C_k a_{n-k-1} \\ & = b_n - b_{n-1} \end{aligned}$$

特征多项式为

$$\begin{aligned} C^*(x) &= x^{n+1} + (C_1 - 1) x^n + (C_2 - C_1) x^{n-1} + \cdots + (C_k - C_{k-1}) x - C_k \\ &= (x - 1) (x^n + C_1 x^{n-1} + C_2 x^{n-2} + \cdots + C_k) \\ &= (x - 1) C(x) \end{aligned}$$

可见，对递推式作一次差分操作，相当于为原本的特征多项式乘上 $(x - 1)$ 。例如，在例 4.5.1 中，递推式

$$S_n - S_{n-1} = n^2$$

的特征多项式是

$$C(x) = x - 1$$

我们对其连续使用了三次差分法，因此最后得到的特征多项式就是 $(x - 1)^4$ 。

需要注意，差分法也适用于 b_n 并非关于 n 的多项式的情况；在这种情况下，差分操作可能是用 a_n 对应的递推式减去 a_{n-1} 对应的递推式的 q 倍，其对特征多项式的影响也就变为乘上 $(x - q)$ ，而并不一定是 $(x - 1)$ 。

例 4.5.2. 考虑 $n \geq 3$ 位的二进制串，其中每一位数字均为 0 或 1；从左向右扫描该串，每次遇到“010”时就将这三个数字划掉，然后跳过被划掉的三个数字继续扫描。如果最后划掉的“010”刚好是原串中的最后三位，求满足这一要求的二进制串的数目。

这个例子的核心难点在于上述要求不等价于“原串的最后三位为 010”。例如，考虑 11 位二进制串

$$00101001010$$

对其执行上述步骤，那么首先会在第 2 ~ 4 位发现一个“010”并划去，然后从原串的第 5 位继续扫描，在原串的第 7 ~ 9 位发现另一个“010”并划去。最终的结果如下：

$$\overline{00101001010}$$

尽管这个串最后三位是“010”，但因为我们会提前划掉倒数第 5 位至倒数第 3 位的“010”，所以它并不满足这个例子的要求。

设满足要求的二进制串的数目是 a_n . 这些串的最后三位必须是“010”, 其余 $n-3$ 位分别可取 0 或 1, 若不考虑其他约束, 则一共有 2^{n-3} 种方案. 不过, 如果第 $n-4$ 位到 $n-2$ 位也是“010”, 并且这个“010”会被划掉, 那么最后三位就不能被一起划掉, 这样的二进制串不满足要求.

考虑这些不满足要求的二进制串的数目: 如果忽略掉串的最后两位, 那么最后一次划掉的“010”就将是串的最后三位, 因此其数目就应该是 a_{n-2} . 由此可构造递推关系:

$$a_n + a_{n-2} = 2^{n-3} \quad (n \geq 5) \quad (4-30)$$

简单讨论可以得到数列初值:

$$a_3 = 1, a_4 = 2$$

可以使用差分法求解上述递推关系. 对 a_{n-1} 应用递推式:

$$a_{n-1} + a_{n-3} = 2^{n-4}$$

将原本的递推式与上式相减, 得

$$a_n + a_{n-2} - 2(a_{n-1} + a_{n-3}) = 2^{n-3} - 2 \cdot 2^{n-4} = 0$$

由此可得一个三阶齐次线性常系数递推关系:

$$a_n - 2a_{n-1} + a_{n-2} - 2a_{n-3} = 0$$

上面我们作差分操作时, 是用 a_n 对应的递推式减去 a_{n-1} 对应的递推式的 2 倍, 这相当于为特征多项式乘上 $(x-2)$. 因此, 递推关系的特征多项式可作如下因式分解:

$$C(x) = x^3 - 2x^2 + x - 2 = (x-2)(x^2 + 1)$$

解得

$$x_1 = 2, x_2 = i = \cos \frac{\pi}{2} + i \sin \frac{\pi}{2}, x_3 = -i = \cos \frac{\pi}{2} - i \sin \frac{\pi}{2}$$

由此可写出 a_n 的通项表达式:

$$a_n = A_1 \cdot 2^n + A_2 \cos \frac{n\pi}{2} + A_3 \sin \frac{n\pi}{2}$$

为求解上式中的待定系数, 我们可以先从之前求出的两个数列初值开始, 利用递推式 (4-30) 逆推出

$$a_0 = \frac{1}{2}, a_1 = 0, a_2 = 0$$

与例 4.4.3 类似, 这些初值均没有实际的组合意义, 其唯一作用是简化通项表达式中待定系数的计算. 根据数列初值, 我们可以列出方程组:

$$\begin{cases} A_1 + A_2 = \frac{1}{2} \\ 2A_1 + A_3 = 0 \\ 4A_1 - A_2 = 0 \end{cases}$$

求解此方程组, 得

$$A_1 = \frac{1}{10}, A_2 = \frac{2}{5}, A_3 = -\frac{1}{5}$$

因此我们有

$$a_n = \frac{2^n}{10} + \frac{2}{5} \cos \frac{n\pi}{2} - \frac{1}{5} \sin \frac{n\pi}{2} \quad (n \geq 3)$$

例 4.5.3. 平面上有 n 条封闭曲线, 任意两条曲线相交于两点, 任意三条曲线不交于一点. 求这些曲线将平面分为多少个区域.

设 n 条封闭曲线将平面分为 a_n 个区域. 为了构建递推关系, 设想先画出 $n-1$ 条曲线将平面分割为 a_{n-1} 个区域, 然后再画出第 n 条曲线. 这条曲线与之前所画出的每一条曲线均交于两点, 总共产生 $2(n-1)$ 个交点; 这些交点将第 n 条封闭曲线截成 $2(n-1)$ 条弧, 每条弧都将平面上原本的一个区域一分为二. 因此, 第 n 条曲线使平面上增加 $2(n-1)$ 个区域. 由此可以写出递推式:

$$a_n = a_{n-1} + 2(n-1) \quad (n \geq 2)$$

递推关系的初值为

$$a_1 = 2$$

非齐次项 $2(n-1)$ 是关于 n 的一次多项式, 因此作两次差分操作即可使其变为零. 第一次差分操作得到:

$$a_n - 2a_{n-1} + a_{n-2} = 2$$

第二次差分操作得到:

$$a_n - 3a_{n-1} + 3a_{n-2} - a_{n-3} = 0$$

这即是一个齐次线性常系数递推关系, 其特征多项式为

$$C(x) = x^3 - 3x^2 + 3x - 1 = (x-1)^3$$

它有一个三重根 $x=1$. 由此可写出 a_n 的通项表达式:

$$a_n = A_0 + A_1 n + A_2 n^2$$

利用原本的递推式以及 $a_1 = 1$, 可推出

$$a_0 = 2, a_2 = 4$$

代入 a_n 的通项表达式中, 列出如下方程组:

$$\begin{cases} A_0 = 2 \\ A_0 + A_1 + A_2 = 2 \\ A_0 + 2A_1 + 4A_2 = 4 \end{cases}$$

求解此方程组, 得

$$A_0 = 2, A_1 = -1, A_2 = 1$$

因此,

$$a_n = n^2 - n + 2 \quad (n \geq 1)$$

当 $n = 0$ 时, 整个平面属于同一个区域, 因此我们可令

$$a_0 = 1$$

这再次体现出, 我们之前求得的数列初值 $a_0 = 2$ 仅是为了方便求解待定系数的工具, 没有实际意义.

4.5.2 特解法

特解法是一种常用的数学思想, 适用于包括递推关系在内的多种类型的问题. 具体地, 对于非齐次递推关系

$$a_n + C_1 a_{n-1} + C_2 a_{n-2} + \cdots + C_k a_{n-k} = b_n$$

我们可以将其分解为如下两个递推关系:

$$u_n + C_1 u_{n-1} + C_2 u_{n-2} + \cdots + C_k u_{n-k} = 0 \quad (4-31)$$

$$v_n + C_1 v_{n-1} + C_2 v_{n-2} + \cdots + C_k v_{n-k} = b_n \quad (4-32)$$

分别解出 u_n, v_n 并令

$$a_n = u_n + v_n$$

即可得到原本的非齐次递推关系的解.

式 (4-31) 是一个齐次线性常系数递推关系, 可用 4.4 节讨论过的方法求解; 式 (4-32) 则仍然是一个非齐次递推关系, 但与原本的递推关系不同, 没有确定的数列初值. 这意味着我们可以构造任意一个满足递推式的 $\{v_n\}$, 然后令

$$u_i = a_i - v_i \quad (i = 0, 1, 2, \dots, k-1)$$

并以此作为 $\{u_n\}$ 的初值求解齐次递推关系 (4-31). 上述过程中, $\{v_n\}$ 称为递推关系的**特解**, $\{u_n\}$ 称为递推关系的**通解**.

特解法的核心问题是如何根据 b_n 构造数列 $\{v_n\}$. 如同我们在本节开始时论述的那样, 没有通用的方法可从一般的 b_n 中构造 $\{v_n\}$; 我们下面仅讨论一种简单但足够常见的情况:

$$b_n = r^n b(n) \quad (4-33)$$

其中 $b(n)$ 是关于 n 的 p 次多项式, $r \neq 0$ 是递推关系的特征多项式的 m 重根. 如果 r 不是特征多项式的根, 则 $m = 0$.

我们讨论如何构造出特解. 首先, 我们有递推式

$$v_n + C_1 v_{n-1} + C_2 v_{n-2} + \cdots + C_k v_{n-k} = r^n b(n)$$

在上式两侧同时除以 r^n , 得

$$\frac{v_n}{r^n} + \frac{C_1}{r} \frac{v_{n-1}}{r^{n-1}} + \frac{C_2}{r^2} \frac{v_{n-2}}{r^{n-2}} + \cdots + \frac{C_k}{r^k} \frac{v_{n-k}}{r^{n-k}} = b(n)$$

作换元 $v_n = r^n q_n$, 得

$$q_n + \frac{C_1}{r} q_{n-1} + \frac{C_2}{r^2} q_{n-2} + \cdots + \frac{C_k}{r^k} q_{n-k} = b(n) \quad (4-34)$$

上式是一个关于 q_n 的递推式, 等号右侧是一个关于 n 的多项式; 从形式上, 我们猜测 q_n 也可以是一个关于 n 的多项式. 为了构造 q_n , 我们考察上述递推式的特征多项式:

$$\begin{aligned} C_q(x) &= x^k + \frac{C_1}{r} x^{k-1} + \frac{C_2}{r^2} x^{k-2} + \cdots + \frac{C_k}{r^k} \\ &= \frac{1}{r^k} (r^k x^k + C_1 r^{k-1} x^{k-1} + C_2 r^{k-2} x^{k-2} + \cdots + C_k) \\ &= \frac{1}{r^k} [(rx)^k + C_1 (rx)^{k-1} + C_2 (rx)^{k-2} + \cdots + C_k] \\ &= \frac{1}{r^k} C(rx) \end{aligned}$$

注意到 r 是原特征方程 $C(x)$ 的 m 重根:

$$C(x) = (x - r)^m D(x)$$

其中 $D(x)$ 是对 $C(x)$ 因式分解得到的余下部分, 是一个 $k-m$ 次的多项式, 且满足 $D(r) \neq 0$. 代入前式中, 得到

$$\begin{aligned} C_q(x) &= \frac{1}{r^k} (rx - r)^m D(rx) \\ &= \frac{1}{r^{k-m}} (x - 1)^m D(rx) \end{aligned}$$

现在我们考虑多项式

$$C_D(x) = \frac{1}{r^{k-m}} D(rx)$$

它可看作一个 $k-m$ 阶线性常系数递推关系

$$\sum_{j=0}^{k-m} D_j q_{n-j} = B(n) \quad (4-35)$$

的特征多项式, 其中 D_j 为 $C_D(x)$ 中 x^{n-j} 项的系数, $B(n)$ 则是一关于 n 的多项式. 由于差分操作是可逆的 (逆操作即为前缀和), 我们可以从 $b(n)$ 出发, 应用 m 次差分操作的逆, 从而

得到 $B(n)$, 这表明 $B(n)$ 必然存在. 根据我们在 4.5.1 小节中的讨论可知, 对此递推关系作 m 次差分操作, 得到的递推式的特征方程即为 $C_q(x)$. 这表明, 只要 $\{q_n\}$ 满足递推式 (4-35), 则其一定也满足递推式 (4-34).

设想对递推式 (4-35) 作 m 次差分操作, 最终得到式 (4-34). 我们在 4.5.1 小节中指出, 对多项式作一次差分操作, 将使它的次数降低 1; 作 m 次差分操作, 将使多项式的次数降低 m . 由于式 (4-34) 的两侧均为关于 n 的 p 次多项式, 因此式 (4-35) 的两侧均为关于 n 的 $m+p$ 次多项式. 这表明 q_n 也必定是关于 n 的至少 $m+p$ 次多项式. 在此之上, 注意到

$$\sum_{j=0}^{k-m} D_j = C_D(1) = \frac{1}{r^{k-m}} D(r) \neq 0$$

这意味着 q_n 中 n 的最高次项在式 (4-35) 左侧的系数不为零, 因此式 (4-35) 左侧多项式的次数就等于 q_n 的次数, 从而 q_n 应该是一个恰好 $m+p$ 次的多项式.

一个 $m+p$ 次的多项式中最多有 $m+p+1$ 项. 但是, 由于在 m 次差分操作中低于 m 次的项会全部变为 0, 因此我们仅需关注不低于 m 次的项:

$$q_n = A_0 n^m + A_1 n^{m+1} + A_2 n^{m+2} + \cdots + A_p n^{m+p}$$

其中 $A_0, A_1, A_2, \cdots, A_p$ 是待定系数. 将上式代入 v_n 中, 得

$$v_n = r^n (A_0 n^m + A_1 n^{m+1} + \cdots + A_p n^{m+p}) \quad (4-36)$$

此即为我们所求的特解. 特解中的待定系数可通过递推式 (4-32) 求解, 也可留到求出通解、代入 $a_n = u_n + v_n$ 后, 利用数列初值直接求解.

上面我们讨论了 $b_n = r^n b(n)$ 时的特解形式, 它可以简单地推广至有限个 $r^n b(n)$ 的线性组合, 即

$$b_n = r_1^n b_1(n) + r_2^n b_2(n) + \cdots + r_t^n b_t(n)$$

对于上述形式的非齐次项, 我们同样可以运用特解法的思路, 先分别求出以下递推式的特解:

$$\begin{cases} v_n^{(1)} + C_1 v_{n-1}^{(1)} + C_2 v_{n-2}^{(1)} + \cdots + C_k v_{n-k}^{(1)} = r_1^n b_1(n) \\ v_n^{(2)} + C_1 v_{n-1}^{(2)} + C_2 v_{n-2}^{(2)} + \cdots + C_k v_{n-k}^{(2)} = r_2^n b_2(n) \\ \vdots \\ v_n^{(t)} + C_1 v_{n-1}^{(t)} + C_2 v_{n-2}^{(t)} + \cdots + C_k v_{n-k}^{(t)} = r_t^n b_t(n) \end{cases}$$

将所有求出的特解相加, 即得到原递推关系的一个特解:

$$v_n = v_n^{(1)} + v_n^{(2)} + \cdots + v_n^{(t)}$$

例 4.5.4. 求解下列递推关系:

$$\begin{cases} a_n - a_{n-1} - 6a_{n-2} = 5 \cdot 4^n & (n \geq 2) \\ a_0 = 5, a_1 = 3 \end{cases}$$

我们使用特解法求解此递推关系. 令

$$a_n = u_n + v_n$$

其中 u_n 是通解, v_n 是特解. 先写出递推关系特征多项式:

$$C(x) = x^2 - x - 6$$

并解出它的两个一重实根

$$x_1 = 3, x_2 = -2$$

非齐次项中含有 4^n , 4 不是特征多项式的根. 因此, 将 $m = 0, p = 0$ 代入式 (4-36), 即可得到特解的形式:

$$v_n = A_0 \cdot 4^n$$

代入原递推关系式:

$$A_0 \cdot 4^n - A_0 \cdot 4^{n-1} - 6A_0 \cdot 4^{n-2} = 5 \cdot 4^n$$

解得

$$A_0 = \frac{40}{3}$$

因此

$$v_n = \frac{40}{3} \cdot 4^n$$

下一步是求递推关系的通解 u_n . 由特征多项式的两个一重实根可知, 通解为

$$u_n = A_1 \cdot 3^n + A_2 \cdot (-2)^n$$

因此有

$$a_n = u_n + v_n = A_1 \cdot 3^n + A_2 \cdot (-2)^n + \frac{40}{3} \cdot 4^n$$

再利用数列初值建立关于待定系数 A_1, A_2 的方程组:

$$\begin{cases} A_1 + A_2 + \frac{40}{3} = 5 \\ 3A_1 - 2A_2 + \frac{160}{3} = 3 \end{cases}$$

求解此方程组, 得

$$A_1 = -\frac{67}{5}, A_2 = \frac{76}{15}$$

由此得到

$$a_n = -\frac{67}{5} \cdot 3^n + \frac{76}{15} \cdot (-2)^n + \frac{40}{3} \cdot 4^n$$

例 4.5.5. 使用尺寸为 1×1 的方砖和 1×2 的长砖铺设 $1 \times n$ 的路径, 求:

- (1) 所有可能的铺砖方案数;
- (2) 每一种可能的铺砖方案中使用的砖数相加, 得到的砖数的总和.

这个例子中的第一问实际上就是 Fibonacci 数列. 设铺砖方案数为 a_n , 则按最后一块砖的长度分类, 有

$$a_n = a_{n-1} + a_{n-2}$$

讨论可得初值

$$a_1 = 1, a_2 = 2$$

因此我们有

$$a_n = F_{n+1} = \frac{1}{\sqrt{5}} (\alpha^{n+1} - \beta^{n+1})$$

其中

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

下面求所有方案中使用的砖的总数. 设总砖数为 b_n , 仍然按照最后一块砖的长度分类:

- 如果最后一块砖是 1×1 的方砖, 则有 a_{n-1} 种铺砖方案; 在这些方案中, 路径前面 $1 \times (n-1)$ 的部分合计使用了 b_{n-1} 块砖, 最后一块 1×1 的方砖合计使用了 a_{n-1} 次;
- 如果最后一块砖是 1×2 的长砖, 则有 a_{n-2} 种铺砖方案; 在这些方案中, 路径前面 $1 \times (n-2)$ 的部分合计使用了 b_{n-2} 块砖, 最后一块 1×2 的长砖合计使用了 a_{n-2} 次.

因此, 我们有如下递推关系:

$$\begin{aligned} b_n &= b_{n-1} + a_{n-1} + b_{n-2} + a_{n-2} \\ &= b_{n-1} + b_{n-2} + a_n \end{aligned}$$

a_n 上面已经求出, 是一个已知量, 因此这是一个关于 b_n 的二阶非齐次线性常系数递推关系. 令

$$a_n = u_n + v_n$$

其中 u_n 是通解, v_n 是特解. b_n 满足的递推式的特征多项式为 $x^2 - x - 1$, 它的根同样是 α 和 β . 由于

$$a_n = \frac{\alpha}{\sqrt{5}} \cdot \alpha^n - \frac{\beta}{\sqrt{5}} \cdot \beta^n$$

将 $m=1, p=0$ 代入式 (4-36) 可知, 特解的形式应为

$$v_n = A\alpha^n + Cn\beta^n$$

其中 A, C 为待定系数. 通解的形式为

$$u_n = B\alpha^n + D\beta^n$$

其中 B, D 同样为待定系数. 相加, 可得 b_n 的形式为

$$b_n = (An + B)\alpha^n + (Cn + D)\beta^n \quad (4-37)$$

讨论可得数列初值为

$$b_0 = 0, b_1 = 1, b_2 = 3, b_3 = 7$$

虽然将上述初值代入递推式解出待定系数即可求出 $\{b_n\}$ 的通项表达式, 但实践中我们可以作进一步的化简以避免求解复杂的四元一次方程组. 将式 (4-37) 代回递推式中, 得到

$$\begin{aligned} (An + B)\alpha^n + (Cn + D)\beta^n &= [A(n-1) + B]\alpha^{n-1} + [C(n-1) + D]\beta^{n-1} \\ &\quad + [A(n-2) + B]\alpha^{n-2} + [C(n-2) + D]\beta^{n-2} \\ &\quad + \frac{1}{\sqrt{5}}(\alpha^{n+1} - \beta^{n+1}) \end{aligned}$$

注意到等式两侧 α^n, β^n 的系数应当分别相等. 由此可得

$$\begin{aligned} (An + B)\alpha^n &= [A(n-1) + B]\alpha^{n-1} + [A(n-2) + B]\alpha^{n-2} + \frac{1}{\sqrt{5}}\alpha^{n+1} \\ (Cn + D)\beta^n &= [C(n-1) + D]\beta^{n-1} + [C(n-2) + D]\beta^{n-2} - \frac{1}{\sqrt{5}}\beta^{n+1} \end{aligned}$$

注意到 α, β 是方程

$$x^2 - x - 1 = 0$$

的解, 我们可以利用此性质化简以上两式. 以第一条关于 α 的等式为例. 首先可将 B 从等式中消去, 因为其在等式两侧的系数相等:

$$An\alpha^n = A(n-1)\alpha^{n-1} + A(n-2)\alpha^{n-2} + \frac{1}{\sqrt{5}}\alpha^{n+1}$$

再对等式右侧进行变形, 以消去等式左侧的 $An\alpha^n$:

$$An\alpha^n = (An\alpha^{n-1} + An\alpha^{n-2}) - \left(A\alpha^{n-1} + 2A\alpha^{n-2} - \frac{1}{\sqrt{5}}\alpha^{n+1} \right)$$

由此可得

$$A\alpha^{n-1} + 2A\alpha^{n-2} - \frac{1}{\sqrt{5}}\alpha^{n+1} = 0$$

解得

$$A = \frac{\alpha^3}{\sqrt{5}(\alpha + 2)} = \frac{\alpha^3}{5\alpha} = \frac{1}{5}\alpha^2$$

同理, 还可解得

$$C = -\frac{\beta^3}{\sqrt{5}(\beta + 2)} = -\frac{\beta^3}{-5\beta} = \frac{1}{5}\beta^2$$

如此解出 A 和 C 之后, 我们只需将 $b_0 = 0, b_1 = 1$ 代入 b_n 的通项表达式中求出 B 和 D 即可:

$$\begin{cases} B + D = 0 \\ \left(\frac{1}{5}\alpha^2 + B\right)\alpha + \left(\frac{1}{5}\beta^2 + D\right)\beta = 1 \end{cases}$$

由此解得

$$B = \frac{1 - \frac{1}{5}(\alpha^3 + \beta^3)}{\alpha - \beta} = \frac{1}{5\sqrt{5}}$$

$$D = -B = -\frac{1}{5\sqrt{5}}$$

最终, 将解得的 A, B, C, D 四个系数代回 b_n 的表达式中, 我们有

$$b_n = \left(\frac{1}{5}\alpha^2 n + \frac{1}{5\sqrt{5}}\right)\alpha^n + \left(\frac{1}{5}\beta^2 n - \frac{1}{5\sqrt{5}}\right)\beta^n$$

这就是所有铺砖方案中使用的砖的总数.

习题

4.1. 设 $G_n = F_{2n}$ ($n \geq 0$), 其中 F_n 是第 n 个 Fibonacci 数.

(1) 证明: $G_n - 3G_{n-1} + G_{n-2} = 0$ ($n = 2, 3, 4, \dots$);

(2) 求数列 $\{G_n\}$ 的母函数.

4.2. 已知数列 $\{a_n\}$ 的母函数为 $\frac{1}{1-x+x^2}$, 求 $\{a_n\}$ 满足的线性常系数递推式, 并求 a_0, a_1 .

4.3. 已知 $a_n = c \cdot 3^n + d \cdot (-1)^n$ ($n \geq 0$), 其中 c, d 是常数, 求 $\{a_n\}$ 满足的一个不含 c, d 的线性常系数递推关系.

4.4. 求解递推关系:

$$\begin{cases} a_n - 2a_{n-1} + a_{n-2} = 5 & (n \geq 2) \\ a_0 = 1, a_1 = 2 \end{cases} \quad (4-38)$$

4.5. 由 A, B, C, D 四个字母组成允许重复的 n 位字符串, 其中子串 AB 至少出现一次, 求满足要求的字符串数目.

4.6. 考虑如下汉诺塔问题的变种: 有 A, B, C 三根柱子, 初始时 A 柱上有 n 个圆盘, 按直径从小到大的顺序编号为 1 到 n ; 最终目标是将所有偶数编号的盘套在 B 柱上、所有奇数编号的盘套在 C 柱上. 移动圆盘时的规则不变, 求所需的移动次数.

4.7. 从包含 k 类元素的多重集

$$M = \{\infty \cdot x_1, \infty \cdot x_2, \dots, \infty \cdot x_k\}$$

中选出 n 个元素作可重排列，但不允许相同元素在排列中连续出现 3 次，求满足要求的排列数目.

4.8. 计算:

$$\sum_{k=1}^n 4^k = 1^4 + 2^4 + \dots + n^4$$

4.9. 从 1 到 n 的正整数中选取 k 个不同且不相邻的数，方案数记为 $f(n, k)$.

- (1) 求 $f(n, k)$ 满足的线性常系数递推关系;
- (2) 用数学归纳法求 $f(n, k)$ 的通项表示;
- (3) 若规定 1 与 n 是相邻的数，并在此前提下令从 1 到 n 的正整数中选取 k 个不同且不相邻的数的方案数为 $g(n, k)$ ，利用 $f(n, k)$ 求 $g(n, k)$.

4.10. 使用尺寸为 1×1 的方砖、直角边长为 1 的等腰直角三角形砖，以及斜边长为 2 的等腰直角三角形砖，铺设 $1 \times n$ 的路径，求:

- (1) 所有可能的铺砖方案数;
- (2) 每一种可能的铺砖方案中使用的砖数相加，得到的砖数的总和.

第五章 特殊计数序列

本章中,我们讨论几类重要的计数序列,包括 Catalan 数、错位排列数,以及两类 Stirling 数. 这些计数序列对应的计数问题均有很强的实际意义,在组合数学领域中得到广泛应用.

我们采用先建立递推关系、再求解递推关系的方式来研究这些计数序列. 这些递推关系虽然不具备第四章讨论的线性性和齐次性等性质,但我们仍能灵活使用母函数方法求解它们.

5.1 Catalan 数

毫不夸张地说, Catalan 数是组合数学领域中最著名的数字序列.

M. Kauers 与 P. Paule, 奥地利约翰开普勒林茨大学教授

Catalan 数定义为

$$C_n = \frac{1}{n+1} \binom{2n}{n} \quad (5-1)$$

它与大量不同的组合计数问题存在联系. 本节中,我们介绍其中几个代表性的计数问题.

我们首先讨论凸多边形的三角剖分方案数这一经典问题,并通过该问题求解出 Catalan 数的通项表达式. 然后,我们再讨论其他一些形式各异、但答案均为 Catalan 数的问题.

例 5.1.1. 在一个凸 $(n+2)$ 边形中作 $n-1$ 条顶点间的连线,将其剖分为 n 个互不重叠的三角形,求不同的剖分方案数. 图 5.1 展示了 $n=3$ 时五边形的 5 种不同剖分方案.

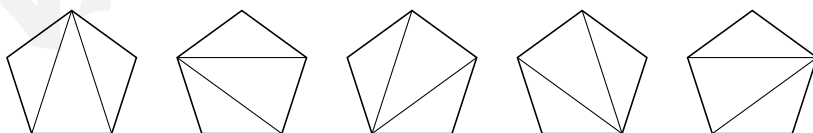


图 5.1: 五边形划分为不重叠三角形的五种方案

设满足要求的剖分方案数为 C_n . 我们简单讨论数列 $\{C_n\}$ 的最初几项的值:

- $n=1$ 时, 多边形本身即为三角形, 因此方案数为 $C_1=1$, 相当于不进行任何剖分;

- $n = 2$ 时, 多边形为四边形, 画出两条对角线中的任何一条, 均能将四边形剖分为两个三角形, 因此方案数为 $C_2 = 2$;
- $n = 3$ 时, 多边形为五边形, 由图 5.1 可知方案数为 $C_3 = 5$.

继续讨论下去, 还能得到 $C_4 = 14$ 、 $C_5 = 42$, 等等.

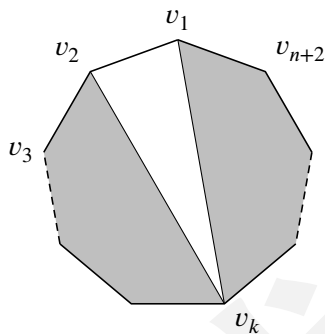


图 5.2: 正多边形划分方案数的递推关系构建

为了求解 $\{C_n\}$ 的通项表达式, 我们可先构建关于它的一个递推关系. 将凸 $(n+2)$ 边形的所有顶点依次标为 $v_1, v_2, v_3, \dots, v_{n+2}$. 考虑 $v_1 v_2$ 这条边, 最终划分出的 n 个三角形中必有恰好一个包含这条边; 这个三角形的两个顶点已经确定是 v_1, v_2 , 余下一个顶点则可能是凸 $(n+2)$ 边形上除 v_1, v_2 外的其他任何一个顶点 v_k ($k = 3, 4, \dots, n+2$), 如图 5.2 所示. 考虑三角形 $\triangle v_1 v_2 v_k$, 它将整个凸 $(n+2)$ 边形分为两部分:

- 边 $v_2 v_k$ 一侧: $(k-1)$ 边形 $v_2 v_3 v_4 \dots v_k$, 其三角剖分方案数为 C_{k-3} ;
- 边 $v_1 v_k$ 一侧: $(n-k+4)$ 边形 $v_1 v_k v_{k+1} \dots v_{n+2}$, 其三角剖分方案数为 C_{n-k+2} .

因此, 由乘法原理, 在剖分出 $\triangle v_1 v_2 v_k$ 的情况下, 剖分方案数为 $C_{k-3} C_{n-k+2}$. 令 k 遍历其所有可能取值, 由加法原理, 我们得到:

$$C_n = \sum_{k=3}^{n+2} C_{k-3} C_{n-k+2} = \sum_{k=0}^{n-1} C_k C_{n-1-k} \quad (5-2)$$

利用 $C_1 = 1$ 和 $C_2 = 2$, 可逆推出数列初值

$$C_0 = 1$$

它对应于多边形的退化情况, 例如当 $k = 3$ 时, 上述讨论指出 $\triangle v_1 v_2 v_3$ 的两侧分别是一个 $(n-1)$ 边形和一个“二边形”; 这个“二边形”可认为是退化的多边形, 其三角剖分方案数 C_0 为 1, 相当于不进行任何剖分.

递推式 (5-2) 可使用母函数方法求解. 设数列 $\{C_n\}$ 的母函数为

$$C(x) = \sum_{n=0}^{\infty} C_n x^n = C_0 + C_1 x + C_2 x^2 + \dots$$

对其进行平方, 得

$$\begin{aligned} C(x)^2 &= \sum_{n=0}^{\infty} \left(\sum_{k=0}^n C_k C_{n-k} \right) x^n \\ &= C_0^2 + (C_0 C_1 + C_1 C_0) x + (C_0 C_2 + C_1 C_1 + C_2 C_0) x^2 + \cdots \\ &= C_1 + C_2 x + C_3 x^2 + \cdots \end{aligned}$$

因为 $C_0 = 1$, 所以

$$C(x) = 1 + xC(x)^2$$

解这个关于 $C(x)$ 的二元一次方程, 得到

$$C(x) = \frac{1 \pm \sqrt{1-4x}}{2x}$$

方程的两个解中只有一个是我们要的母函数. 注意到

$$\begin{aligned} \frac{1 + \sqrt{1-4x}}{2x} &= \frac{1}{2x} \left[1 + \sum_{n=0}^{\infty} \binom{\frac{1}{2}}{n} (-4)^n x^n \right] \\ &= \frac{1}{2x} \left[2 + \sum_{n=1}^{\infty} \binom{\frac{1}{2}}{n} (-4)^n x^n \right] \\ &= \frac{1}{x} + \sum_{n=0}^{\infty} \frac{1}{2} \binom{\frac{1}{2}}{n+1} (-4)^{n+1} x^n \end{aligned}$$

中含有 x 的负数次幂项, 不符合我们定义的母函数形式, 因此只有

$$C(x) = \frac{1 - \sqrt{1-4x}}{2x}$$

是正确的母函数. 对其作 Maclaurin 展开, 得到

$$\begin{aligned} C(x) &= \frac{1}{2x} \left[1 - \sum_{n=0}^{\infty} \binom{\frac{1}{2}}{n} (-4)^n x^n \right] \\ &= -\frac{1}{2x} \sum_{n=1}^{\infty} \left[\frac{(-4)^n}{n!} \prod_{j=0}^{n-1} \left(\frac{1}{2} - j \right) \right] x^n \\ &= \sum_{n=1}^{\infty} \left[-\frac{4^n}{2 \cdot n!} \prod_{j=0}^{n-1} \left(j - \frac{1}{2} \right) \right] x^{n-1} \\ &= \sum_{n=0}^{\infty} \left[-\frac{4^{n+1}}{2 \cdot (n+1)!} \cdot \frac{(-1) \cdot \prod_{j=1}^n (2j-1)}{2^{n+1}} \right] x^n \\ &= \sum_{n=0}^{\infty} \frac{2^n \prod_{j=1}^n (2j-1)}{(n+1)!} x^n \end{aligned}$$

由此我们得到

$$C_n = \frac{2^n \prod_{j=1}^n (2j-1)}{(n+1)!} \quad (5-3)$$

利用二项式系数, 我们可对此式作进一步化简:

$$\begin{aligned} C_n &= \frac{1}{n+1} \cdot \frac{2^n \prod_{j=1}^n (2j-1)}{n!} \\ &= \frac{1}{n+1} \cdot \frac{2^n \cdot n! \cdot \prod_{j=1}^n (2j-1)}{(n!)^2} \\ &= \frac{1}{n+1} \cdot \frac{\prod_{j=1}^n 2j \cdot \prod_{j=1}^n (2j-1)}{(n!)^2} \\ &= \frac{1}{n+1} \cdot \frac{(2n)!}{(n!)^2} = \frac{1}{n+1} \binom{2n}{n} \end{aligned}$$

结果即为 Catalan 数.

例 5.1.2. 考虑 n 个矩阵的链式乘积

$$P = A_1 A_2 \cdots A_n$$

为了计算 P , 一共需要计算 $n-1$ 次矩阵乘法. 矩阵乘法满足结合律, 因此我们可以在上式中添加 $n-1$ 对括号以明确计算次序, 最终得到所谓的**完全括号化方案**. 一个完全括号化方案定义为或者仅包含一个矩阵, 或者是外加一对括号的两个完全括号化方案的乘积. 例如, 当 $n=4$ 时, 共有如下 5 种完全括号化方案:

$$P = (((A_1 A_2) A_3) A_4)$$

$$P = ((A_1 (A_2 A_3)) A_4)$$

$$P = ((A_1 A_2) (A_3 A_4))$$

$$P = (A_1 ((A_2 A_3) A_4))$$

$$P = (A_1 (A_2 (A_3 A_4)))$$

设为矩阵乘积添加括号的不同方案数为 a_n . 考虑最外层括号中的两个元素, 它们分别可视为 $A_1 A_2 \cdots A_k$ 和 $A_{k+1} A_{k+2} \cdots A_n$ 的一个加括号方案 ($k=1, 2, \dots, n-1$). 因此, 我们有

$$a_n = \sum_{k=1}^{n-1} a_k a_{n-k}$$

当 $n=1$ 时, 无需加任何括号, 因此边界条件为 $a_1=1$. 注意到, 上述递推式与 Catalan 数满足的递推式具有相同的形式; 观察边界条件, 可知

$$a_n = C_{n-1} = \frac{1}{n} \binom{2n-2}{n-1}$$

例 5.1.3. 考虑含有 n 个叶子节点和 $n-1$ 个非叶子节点 (或称内部节点) 的二叉树; 求满足要求的不同二叉树结构的数目.

由数学归纳法可知, 在这样的二叉树中, 每个内部节点都必须有两个后代节点. 注意到, 满足要求的包含 n 个节点的二叉树, 与 n 个矩阵链式乘积的完全括号化方案之间存在一一对应关系. 图 5.3 展示了 $n=4$ 时的情况: 以每个节点为根的子二叉树均对应一个完全括号化方案, 叶子节点依次对应每个矩阵, 内部节点则可视作添加的括号. 由此立即得到满足要求的二叉树结构的数目为 $C_{n-1} = \frac{1}{n} \binom{2n-2}{n-1}$.

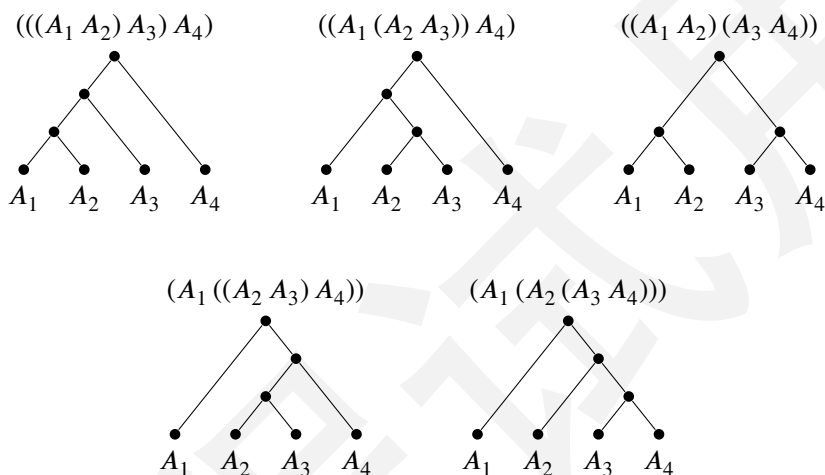


图 5.3: 例 5.1.3 中 $n=4$ 时满足要求的二叉树及其对应的完全括号化方案

例 5.1.4. 有 n 个相同的节点, 用这些节点构成任意二叉树, 可以得到多种树结构. 例如, 当 $n=3$ 时, 共有如图 5.4 所示的 5 种二叉树结构.

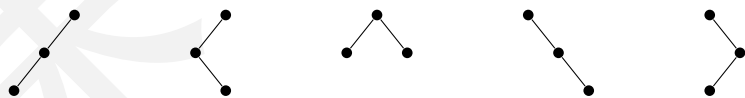


图 5.4: 例 5.1.4 中 $n=3$ 时所有可能的二叉树结构示意图

设包含 n 个节点的二叉树的不同结构数目为 a_n . 考虑此二叉树的根节点, 设其左子树包含的节点数为 k ($k=0, 1, 2, \dots, n-1$), 则右子树包含的节点数必定为 $n-1-k$. 因此, 我们有

$$a_n = \sum_{k=0}^{n-1} a_k a_{n-1-k}$$

当 $n=0$ 时, 子树为空, 因此可视为只有 $a_0=1$ 种结构. 上述递推关系与 Catalan 数的递推

关系完全相同，由此可得

$$a_n = C_n = \frac{1}{n+1} \binom{2n}{n}$$

例 5.1.5. 考虑一个 n 行 n 列的格路模型，起点为左下角 $(0,0)$ ，终点为右上角 (n,n) 。要求路径始终在对角线 $y=x$ 下方（允许接触对角线，但不能穿越），求方案数。图 5.5 中展示了 $n=6$ 时的网格和一条合法路径。

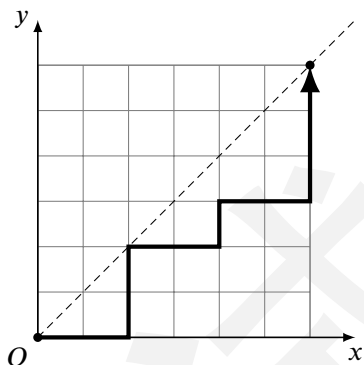


图 5.5: Catalan 数的格路模型

许多实际问题均可抽象为上述格路模型。例如：

栈（stack）是一种先进后出、后进先出的数据结构。现有 n 个不同的元素依次入栈、出栈，使得最终栈内为空；将这些元素按照出栈顺序依次排列，构成的序列称为出栈序列。求有多少种不同的出栈序列。

元素入栈对应格路模型中向右走的一步，元素出栈对应格路模型中向上走的一步。空栈无法出栈，所以出栈的元素总数不能多于入栈的元素总数，这也就意味着格路模型中的行走路径必须始终在对角线 $y=x$ 下方。

一个长度为 $2n$ 的括号序列中包含 n 个左括号 “(” 和 n 个右括号 “)”，求有多少种不同的合法括号序列。

一个合法的括号序列中，每个右括号都必须匹配一个位于它前面的左括号，且左右括号需要一一对应。因此，左括号 “(” 对应格路模型中向右走的一步，右括号 “)” 对应格路模型中向上走的一步。如果路径穿越了对角线 $y=x$ ，意味着右括号的数量多过了左括号，那么多出的右括号就没有了能匹配的左括号，括号序列即为非法。另外，在例 5.1.2 中的矩阵乘积完全括号化方案中，若我们删除所有矩阵 A_1, A_2, \dots, A_n 而保留所有括号，最终得到的就是长度为 $2n-2$ 的合法括号序列。

一家电影院售卖价格为 50 元的电影票, 售票处在开始售票时没有准备任何用于找零的钞票. 有 $2n$ 位顾客排队买票, 其中 n 个人拿着 100 元钞票、 n 个人拿着 50 元钞票. 求不同的排队方案数, 使得售票处能为每一位拿着 100 元钞票的顾客成功找零.

拿着 50 元钞票的顾客对应格路模型中向右走的一步, 拿着 100 元钞票的顾客则对应格路模型中向上走的一步. 如果路径穿越了对角线 $y = x$, 意味着售票处收到的 100 元钞票多过了 50 元钞票, 这时售票处将无法为顾客找零. 不过, 与前两个问题相比, 这个问题的答案要额外乘上 $(n!)^2$, 因为还需考虑到不同顾客之间的差异.

下面我们求解这个例子中给出的格路模型. 先不考虑禁止穿越对角线 $y = x$ 的限制, 总的路径数为

$$\binom{2n}{n}$$

再考虑违反此要求的非法路径的数目. 非法的路径穿越了直线 $y = x$, 等价于接触了直线 $y = x + 1$. 我们找到路径第一次接触直线 $y = x + 1$ 的点, 将路径在这一点之前的部分绕此直线翻转. 原先的起点 $(0, 0)$ 经过如上翻转将变为 $(-1, 1)$, 因此我们将得到一条从 $(-1, 1)$ 到 (n, n) 的路径, 如图 5.6 所示.

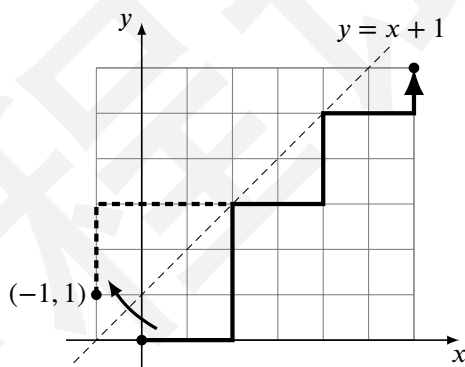


图 5.6: Catalan 数格路模型中非法路径的示意图

由于 $(-1, 1)$ 和 (n, n) 位于 $y = x + 1$ 的两侧, 因此任何一条从 $(-1, 1)$ 到 (n, n) 的路径均必定与 $y = x + 1$ 接触. 若将路径第一次与直线 $y = x + 1$ 接触之前的部分绕此直线翻转, 则我们能重新得到一个从 $(0, 0)$ 到 (n, n) 、直线 $y = x + 1$ 接触的路径. 由此可见, 从 $(0, 0)$ 到 (n, n) 的非法路径与从 $(-1, 1)$ 到 (n, n) 的所有路径一一对应, 这些路径的数量为

$$\binom{2n}{n-1}$$

根据减法原理, 合法的路径数为

$$\binom{2n}{n} - \binom{2n}{n-1} = \binom{2n}{n} - \frac{n}{n+1} \binom{2n}{n} = \frac{1}{n+1} \binom{2n}{n}$$

结果同样是 Catalan 数.

例 5.1.6. $2n$ 位宾客围在一张圆桌旁, 现在要让他们两两握手, 要求握手方案中不能存在交叉. 例如, 图 5.7 中展示了当 $n = 3$ 时的一种满足要求的方案和一种不满足要求的方案, 其中每条虚线都代表一次握手. 求满足要求的握手方案总数.

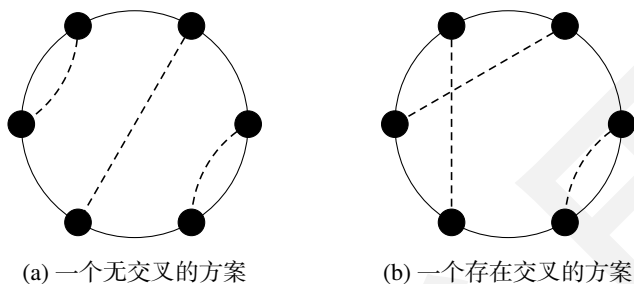


图 5.7: $n = 3$ 时可能出现的握手方案

我们可以首先从 $2n$ 位宾客中选择一位, 从这个人开始依次将所有人编号为 $1, 2, \dots, 2n$, 并考察他们的握手对象. 若某人和比他编号更大的宾客握手, 则想象让他先保持伸出手的状态原地等候; 后面我们必然会遇到这样的宾客 (记作 A), 他要和比他编号更小的宾客握手. 注意到, A 的握手对象只能是目前那些伸出手等待的宾客中编号最大的那一位 (记作 B); 否则, 假设他选择了编号更小的人 C , 则 B 会被 A 、 C “夹在中间”; 后续另一位宾客 D 和 B 握手时, AC 和 BD 就必然相交, 从而违反我们的要求.

上面这个过程可以看做是出栈序列问题: 我们依次扫描每一位宾客, 入栈代表此人将和编号更大的宾客握手, 出栈则代表此人要和目前栈顶的宾客握手. 一个合法的握手方案包含 n 次进栈和 n 次出栈, 因此总的方案数即为 C_n .

5.2 错位排列

错位排列 (derangement) 指的是每个元素都不在其原位上的排列.

定义 5.1 (错位排列). 设 $a_1 a_2 \dots a_n$ 是 $\{1, 2, \dots, n\}$ 的排列, 满足 $a_i \neq i$ ($i = 1, 2, \dots, n$), 则此排列称为 $\{1, 2, \dots, n\}$ 的错位排列.

给定 n , 不同的错位排列的数目称为错位排列数, 记为 D_n .

例如, 当 $n = 3$ 时, 共有如下两个错位排列:

$$\{2, 3, 1\}, \{3, 1, 2\}$$

下面我们构建数列 $\{D_n\}$ 满足的一个递推关系. 考虑错位排列的首个元素 a_1 . 根据要求, 我们有 $a_1 \neq 1$, 因此不妨设

$$a_1 = k \quad (k = 2, 3, \dots, n)$$

这时, 考虑 a_k 的取值情况:

- 若 $a_k = 1$, 则 a_1 和 a_k 均已经满足错位排列的要求, 余下 $n-2$ 个元素的错位排列方案数为 D_{n-2} ;
- 若 $a_k \neq 1$, 考察 a_2, a_3, \dots, a_n 这些元素原本受到的取值约束:

$$a_2 \neq 2, a_3 \neq 3, \dots, a_k \neq k, \dots, a_n \neq n$$

其中, 由于 $a_1 = k$ 且 $a_1 a_2 \dots a_n$ 是 $\{1, 2, \dots, n\}$ 的一个排列, 约束 $a_k \neq k$ 已经自动满足. 同时, 我们新引入了约束 $a_k \neq 1$, 使得上述约束变为如下形式:

$$a_2 \neq 2, a_3 \neq 3, \dots, a_k \neq 1, \dots, a_n \neq n$$

注意到, 上面两组约束是完全等价的 (可以设想将 a_k “重命名” 为 a_1 以更清楚地观察到这一点), 因此两组约束下 a_2, a_3, \dots, a_n 的取值方案数相同, 均为 D_{n-1} .

综上所述, 若固定 $a_1 = k$, 则有 $D_{n-2} + D_{n-1}$ 种错位排列方案. 由于 k 有 $n-1$ 种可能的取值, 因此我们有

$$D_n = (n-1)(D_{n-2} + D_{n-1}) \quad (5-4)$$

讨论可知数列初值为

$$D_1 = 0, D_2 = 1$$

利用上述初值和递推式还可倒推出 $D_0 = 1$. 关于这个值的组合意义, 我们可作如下解释: 0 个元素的排列中不含任何元素, 从而也就没有元素在其原位上, 因此有唯一一种满足要求的不含任何元素的排列.

递推式 (5-4) 不太容易直接求解, 我们可对其作进一步变形和简化. 注意到

$$\begin{aligned} D_n - nD_{n-1} &= (n-1)(D_{n-2} + D_{n-1}) - nD_{n-1} \\ &= (-1)^1 [D_{n-1} - (n-1)D_{n-2}] \\ &= (-1)^2 [D_{n-2} - (n-2)D_{n-3}] \\ &= \dots \\ &= (-1)^{n-1}(D_1 - D_0) = (-1)^n \end{aligned}$$

因此我们有

$$D_n = nD_{n-1} + (-1)^n \quad (5-5)$$

此式是错位排列数 D_n 满足的另一个递推式, 我们可利用它求解 $\{D_n\}$ 的指数型母函数. 设 $\{D_n\}$ 的指数型母函数为

$$G_e(x) = D_0 + D_1x + D_2\frac{x^2}{2!} + D_3\frac{x^3}{3!} + \dots$$

利用式 (5-5), 我们有

$$\begin{aligned} G_e(x) &= D_0 + D_1x + \frac{D_2}{2!}x^2 + \frac{D_3}{3!}x^3 + \dots \\ xG_e(x) &= D_0x + \frac{2D_1}{2!}x^2 + \frac{3D_2}{3!}x^3 + \dots \\ e^{-x} &= 1 + (-1)x + (-1)^2x^2 + (-1)^3x^3 + \dots \end{aligned}$$

比较母函数展开式中的各项系数, 可得

$$G_e(x) - xG_e(x) - e^{-x} = 0$$

由此解得

$$G_e(x) = \frac{e^{-x}}{1-x} \quad (5-6)$$

利用前缀和公式 (3-7) 展开上式:

$$\begin{aligned} G_e(x) &= \frac{1}{1-x} \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} x^n \\ &= \sum_{n=0}^{\infty} \sum_{k=0}^n \frac{(-1)^k}{k!} x^n \\ &= \sum_{n=0}^{\infty} \left[\sum_{k=0}^n (-1)^k \frac{n!}{k!} \right] \frac{x^n}{n!} \end{aligned}$$

最终, 我们得到:

$$D_n = \sum_{k=0}^n (-1)^k \frac{n!}{k!} = \sum_{k=0}^n (-1)^k \binom{n}{k} (n-k)! \quad (5-7)$$

此即为错位排列数的通项表达式.

本节中我们通过使用指数型母函数求解递推关系计算出了错位排列数 D_n . 第六章中我们将介绍容斥原理, 并利用容斥原理重新计算出 D_n .

5.3 第二类 Stirling 数

第二类 Stirling 数 $S(n, k)$ 对将有限集合划分为特定数目的非空子集的方案进行计数.

定义 5.2. 设 A 是 n 元素集, 将其划分为 k 个非空子集 A_1, A_2, \dots, A_k , 满足

$$\begin{aligned} A &= A_1 \cup A_2 \cup \dots \cup A_k \\ A_i \cap A_j &= \emptyset \quad (1 \leq i, j \leq k, i \neq j) \end{aligned}$$

则集合 $\{A_1, A_2, \dots, A_k\}$ 的数目称为第二类 Stirling 数 $S(n, k)$.

例如, 当 $n = 4, k = 2$ 时, 不妨设 $A = \{1, 2, 3, 4\}$, 则 A 有如下 $S(4, 2) = 7$ 种划分方式:

$$\begin{aligned} & \{\{1\}, \{2, 3, 4\}\} \\ & \{\{2\}, \{1, 3, 4\}\} \\ & \{\{3\}, \{1, 2, 4\}\} \\ & \{\{4\}, \{1, 2, 3\}\} \\ & \{\{1, 2\}, \{3, 4\}\} \\ & \{\{1, 3\}, \{2, 4\}\} \\ & \{\{1, 4\}, \{2, 3\}\} \end{aligned}$$

第二类 Stirling 数对应的组合场景有如下等价球盒模型:

将 n 个不同的小球放入 k 个相同的盒子, 禁止出现空盒, 求不同的放球方案数.

这种表述相较定义 5.2 中基于集合概念的表述更为简洁.

我们首先讨论 k 取几个特殊值时 $S(n, k)$ 的值.

推论 5.3. 设 n 是正整数, 则

- (1) $S(n, 1) = 1$;
- (2) $S(n, n) = 1$;
- (3) $S(n, k) = 0$ (若 $k > n$).

证明. 针对每种 k 的取值, 分别讨论对应的球盒模型:

- (1) 将 n 个小球放入 1 个盒子中, 仅有唯一的 1 种方案, 即将所有小球全放入这个盒子中, 因此有

$$S(n, 1) = 1$$

- (2) 将 n 个小球放入 n 个盒子中, 由于禁止出现空盒, 因此只能每个盒子中放一个小球, 又因为盒子是相同的, 因此有

$$S(n, n) = 1$$

- (3) 将 n 个小球放入 $k > n$ 个盒子中, 必然出现空盒, 因此没有合法方案, 从而有

$$S(n, k) = 0 \quad (k > n)$$

□

推论 5.4. 设 $n \geq 2$ 是正整数, 则

- (1) $S(n, 2) = 2^{n-1} - 1$;
- (2) $S(n, n-1) = \binom{n}{2}$.

证明. 针对每种 k 的取值, 分别讨论对应的球盒模型:

- (1) 将 n 个小球放入 2 个盒子中. 我们可任选一个小球, 将其记作 a , 然后将 a 放入一个盒子中. 对于其余的 $n-1$ 个小球, 每个小球既可以放入 a 所在的盒子中, 也可以放入另一个盒子中. 但是, 它们不能全部放入 a 所在的盒子中, 否则另一个盒子将是空盒, 不符合要求. 因此, 总方案数即为

$$S(n, 2) = 2^{n-1} - 1$$

- (2) 将 n 个小球放入 $n-1$ 个盒子中. 显然, 恰好会有一个盒子容纳 2 个小球, 而其余盒子均容纳 1 个小球. 因此, 最终的放球方案仅取决于哪两个小球放在同一个盒子中, 从而方案数即为

$$S(n, n-1) = \binom{n}{2} \quad \square$$

下面我们讨论第二类 Stirling 数 $S(n, k)$ 应满足的递推关系. 我们可任选一个小球, 将其记作 a , 然后考虑 a 所在的盒子中的小球:

- 若 a 所在的盒子中无其他小球, 则其余 $n-1$ 个小球需要放入 $k-1$ 个盒子, 使得每个盒子非空, 方案数为 $S(n-1, k-1)$;
- 若 a 所在的盒子中有其他小球, 则可设想先将其余 $n-1$ 个小球放入 k 个盒子, 使得每个盒子非空, 有 $S(n-1, k)$ 种方案; 再将小球 a 放入 k 个盒子中的任意一个, 有 k 种方案, 因此总的方案数为 $k \cdot S(n-1, k)$.

由加法原理可得

$$S(n, k) = S(n-1, k-1) + k \cdot S(n-1, k) \quad (5-8)$$

上面我们讨论了 n, k 为正整数的情况, 实际上 $S(n, k)$ 对 n, k 为非负整数的情况同样有定义. 我们规定

$$S(n, 0) = 0 \quad (n > 0) \quad (5-9)$$

$$S(0, 0) = 1 \quad (5-10)$$

其中, 式 (5-9) 表明不可能划分一个非空集合却不得到任何非空子集. 式 (5-10) 表明若将空集划分为 0 个非空子集, 则认为有 1 种方案, 即什么都不做, 这可看作一种方便起见的规定. 后面我们介绍第二类 Stirling 数的代数定义时, 将会看到式 (5-10) 能自然地满足代数定义.

下面我们讨论 $S(n, k)$ 的母函数. 对 $S(n, k)$ 既可构造普通的母函数, 也可构造指数型母函数. 先讨论普通的母函数, 它可通过递推式 (5-8) 构造. $S(n, k)$ 依赖两个参数 n, k , 我们固定 k 的取值, 令

$$G_k(x) = \sum_{n=0}^{\infty} S(n, k)x^n = \sum_{n=k}^{\infty} S(n, k)x^n$$

则由递推式可得

$$[x^n]G_k(x) - [x^n](kxG_k(x)) - [x^n](xG_{k-1}(x)) = 0 \quad (n \geq 2, k \geq 2)$$

即

$$G_k(x) - kxG_k(x) - xG_{k-1}(x) = 0$$

解得

$$G_k(x) = \frac{xG_{k-1}(x)}{1 - kx}$$

此式可看作一个关于母函数列 $\{G_k(x)\}$ 的一阶递推关系, 可通过迭代应用递推式的方法求解:

$$\begin{aligned} G_k(x) &= \frac{x}{1 - kx} G_{k-1}(x) \\ &= \frac{x^2}{(1 - kx)[1 - (k-1)x]} G_{k-2}(x) \\ &= \dots \\ &= \frac{x^k}{\prod_{j=1}^k (1 - jx)} G_0(x) \end{aligned}$$

由于 $S(0, 0) = 1$, 且当 $k > 0$ 时 $S(0, k) = 0$, 我们有

$$G_0(x) = 1$$

由此可得

$$G_k(x) = \frac{x^k}{\prod_{j=1}^k (1 - jx)} \quad (5-11)$$

此即为第二类 Stirling 数的母函数.

从母函数 (5-11) 求解 $S(n, k)$ 的通项表达式需要一定技巧. 首先, 从 $G_k(x)$ 中提取因式 x^k , 并对余下的部分作部分分式分解:

$$G_k(x) = x^k \left(\frac{A_1}{1-x} + \frac{A_2}{1-2x} + \dots + \frac{A_k}{1-kx} \right) \quad (5-12)$$

其中 A_1, A_2, \dots, A_k 是待定系数. 下面我们求解这些待定系数. 对任意 $m \in \{1, 2, \dots, k\}$, 我们有

$$\begin{aligned} \lim_{x \rightarrow \frac{1}{m}} (1 - mx)G_k(x) &= \lim_{x \rightarrow \frac{1}{m}} x^k \left(\frac{1 - mx}{1 - x} A_1 + \frac{1 - mx}{1 - 2x} A_2 + \dots + \frac{1 - mx}{1 - mx} A_m + \dots + \frac{1 - mx}{1 - kx} A_k \right) \\ &= \frac{A_m}{m^k} \end{aligned}$$

另一方面, 从式 (5-11) 出发, 也可求出上述极限值:

$$\begin{aligned}
 \lim_{x \rightarrow \frac{1}{m}} (1 - mx) G_k(x) &= \lim_{x \rightarrow \frac{1}{m}} \frac{x^k (1 - mx)}{\prod_{j=1}^k (1 - jx)} \\
 &= \lim_{x \rightarrow \frac{1}{m}} \frac{x^k}{\prod_{j=1}^{m-1} (1 - jx) \prod_{j=m+1}^k (1 - jx)} \\
 &= \frac{1}{m^k} \left(\prod_{j=1}^{m-1} \frac{m}{m-j} \right) \left(\prod_{j=m+1}^k \frac{m}{m-j} \right) \\
 &= \frac{m^{k-1}}{m^k} \frac{1}{(m-1)!} \frac{(-1)^{k-m}}{(k-m)!} \\
 &= \frac{(-1)^{k-m}}{m!(k-m)!}
 \end{aligned}$$

比较上述两式, 我们立即得到

$$A_m = (-1)^{k-m} \frac{m^k}{m!(k-m)!}$$

将上式代回部分因式分解式 (5-12) 中并展开, 即可得到 $S(n, k)$ 的通项表达式:

$$\begin{aligned}
 S(n, k) &= [x^n] G_k(x) = \sum_{j=1}^k A_j \cdot j^{n-k} \\
 &= \sum_{j=1}^k (-1)^{k-j} \frac{1}{j!(k-j)!} j^n \\
 &= \frac{1}{k!} \sum_{j=1}^k (-1)^{k-j} \frac{k!}{j!(k-j)!} j^n \\
 &= \frac{1}{k!} \sum_{j=1}^k (-1)^{k-j} \binom{k}{j} j^n
 \end{aligned} \tag{5-13}$$

下面我们从另一个角度出发, 尝试构造 $S(n, k)$ 的指数型母函数, 以求解其通项表达式. 具体地, 我们首先考虑如下球盒模型:

将 n 个不同的小球放入 k 个不同的盒子, 禁止出现空盒, 求不同的放球方案数 $S^*(n, k)$.

上述模型与第二类 Stirling 数的球盒模型仅有一点区别, 即盒子彼此不同. 因此, 可以设想在第二类 Stirling 数的球盒模型中, 按要求放置小球后, 再为每个盒子标上不同的编号. 由于盒子内已经放入了不同的小球, 盒子间即使不标号也已经存在区别, 因此不同的

标号方法数即为 $k!$. 由此, 我们有

$$S^*(n, k) = k! \cdot S(n, k)$$

现在我们从另一个角度考虑此球盒模型. 注意到, 它实际上等价于如下可重排列问题:

从多重集

$$M = \{\infty \cdot x_1, \infty \cdot x_2, \dots, \infty \cdot x_n\}$$

中选出 n 个元素作可重排列, 要求每类元素都至少选用 1 个, 求不同的可重排列方案数.

此可重排列问题的解与球盒模型的解一一对应: 可重排列中第 i 个位置上的元素为 x_j , 对应于在球盒模型中将 i 号小球放入 j 号盒子.

转化为可重排列后的问题可以使用指数型母函数解决. 考虑任意一类元素 x_j ($j = 1, 2, \dots, n$), 此类元素必须选用至少一次, 因此其指数型母函数为

$$\sum_{n=1}^{\infty} \frac{x^n}{n!} = x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots = e^x - 1$$

由乘法原理, 将全部 n 类元素对应的指数型母函数相乘, 即得上述多重排列问题的解的母函数:

$$\begin{aligned} G_e^*(x) &= (e^x - 1)^k \\ &= \sum_{j=0}^k \binom{k}{j} (-1)^j e^{(k-j)x} \\ &= \sum_{j=0}^k \binom{k}{j} (-1)^j \sum_{n=0}^{\infty} \frac{(k-j)^n}{n!} x^n \\ &= \sum_{n=0}^{\infty} \left[\sum_{j=0}^k \binom{k}{j} (-1)^j (k-j)^n \right] \frac{x^n}{n!} \end{aligned} \quad (5-14)$$

由此我们得到

$$S^*(n, k) = \sum_{j=0}^k (-1)^j \binom{k}{j} (k-j)^n$$

这表明

$$S(n, k) = \frac{1}{k!} \sum_{j=0}^k (-1)^j \binom{k}{j} (k-j)^n \quad (5-15)$$

此即为 $S(n, k)$ 的通项表达式. 比较上式与式 (5-13), 容易发现二者是完全等价的.

注意

上面的推导指出, 第二类 Stirling 数的指数型母函数是

$$G_e(x) = \frac{(e^x - 1)^k}{k!}$$

第二类 Stirling 数对应的球盒模型是将 n 个小球放入恰好 k 个盒子. 上面我们已经求出, 每个盒子中的小球对应的指数型母函数是

$$G_{e, \text{球}}(x) = e^x - 1$$

现在我们考虑盒子数目对应的指数型母函数. 由于盒子的数目必须不多不少恰好是 k , 根据母函数的思想, 我们有

$$G_{e, \text{盒}}(x) = \frac{x^k}{k!}$$

观察上面三个式子, 可以发现

$$G_e(x) = G_{e, \text{盒}}(G_{e, \text{球}}(x))$$

即, 球盒模型的母函数等于盒与球的母函数的复合. 本书第一章中讨论了一些球盒模型, 读者可自行验证它们均符合这条性质. 此性质非常重要, 我们将在 5.4 节中再次用到它.

利用通项表达式, 我们可对任意正整数 n, k 计算 $S(n, k)$. 表 5.1 展示了 $n \leq 7$ 时 $S(n, k)$ 的值. 由于 $k > n$ 时 $S(n, k) = 0$, 相应的项可以略去不写, 因此这是一个三角形数表.

$S(n, k) \backslash k$	1	2	3	4	5	6	7
n							
1	1						
2	1	1					
3	1	3	1				
4	1	7	6	1			
5	1	15	25	10	1		
6	1	31	90	65	15	1	
7	1	63	301	350	140	21	1

表 5.1: $n \leq 7$ 时第二类 Stirling 数 $S(n, k)$ 的值

上面我们基于第二类 Stirling 数的组合学定义讨论了其性质. 实际上, 第二类 Stirling 数除可使用组合方法定义外, 还可使用代数方法定义. 我们先引入下降幂 (falling factorial)

的概念:

定义 5.5 (下降幂). 设 x 是实数, n 是非负整数. x 的 n 次下降幂定义为

$$(x)_n = \prod_{j=1}^n (x - j + 1) = x(x-1)(x-2) \cdots (x-n+1) \quad (5-16)$$

当 $n=0$ 时, 与普通的乘幂类似, 定义 $(x)_0 = 1$.

考虑由关于 x 的全体不高于 n 次的多项式构成的线性空间. 此线性空间的自然基为

$$\{x^0, x^1, x^2, \dots, x^n\}$$

注意到, x^n 与 $(x)_n$ 均是关于 x 的 n 次多项式. 这意味着

$$\{(x)_0, (x)_1, (x)_2, \dots, (x)_n\}$$

同样是上述线性空间的一组基. 因此, 自然基中的每一个元素必定能表示为上述基的线性组合, 其系数即定义为第二类 Stirling 数.

定义 5.6 (第二类 Stirling 数的代数定义). 设 n 是非负整数. 定义第二类 Stirling 数 $S(n, k)$ 为将 x^n 表示为

$$\{(x)_0, (x)_1, (x)_2, \dots, (x)_n\}$$

的线性组合时 $(x)_k$ 项的系数:

$$x^n = \sum_{k=0}^n S(n, k)(x)_k \quad (5-17)$$

$S(n, k)$ 的组合定义和代数定义间没有较为直观的联系, 但这两种定义确实是等价的. 为了说明这一点, 我们下面证明代数定义下的 $S(n, k)$ 满足与组合定义完全相同的递推关系. 使用数学归纳法, 对 n 施加归纳, 由归纳假设, 我们有

$$\begin{aligned} x^n &= x \cdot x^{n-1} \\ &= x \sum_{k=0}^{n-1} S(n-1, k)(x)_k \\ &= \sum_{k=0}^{n-1} (x-k)S(n-1, k)(x)_k + \sum_{k=0}^{n-1} k \cdot S(n-1, k)(x)_k \\ &= \sum_{k=0}^{n-1} S(n-1, k)(x)_{k+1} + \sum_{k=0}^{n-1} k \cdot S(n-1, k)(x)_k \\ &= \sum_{k=1}^n S(n-1, k-1)(x)_k + \sum_{k=1}^n k \cdot S(n-1, k)(x)_k \\ &= \sum_{k=1}^n [S(n-1, k-1) + k \cdot S(n-1, k)](x)_k \end{aligned}$$

由此可得

$$S(n, k) = S(n-1, k-1) + k \cdot S(n-1, k)$$

对于 $n=0$ 或 $k=0$ 时的边界条件, 考察式 (5-17) 的两侧, 我们有

$$\begin{aligned} S(n, 0) &= 0 \quad (n > 0) \\ S(0, 0) &= 1 \end{aligned}$$

可见, 在由第二类 Stirling 数的代数定义推导出的递推关系中, $S(n, k)$ 的初值和递推式均与组合定义相同, 因此这两种定义等价.

本节的最后, 我们讨论第二类 Stirling 数满足的一些性质.

定理 5.7. 设 n, k 是正整数, 则

$$S(n, k) = \sum_{j=k-1}^{n-1} \binom{n-1}{j} S(j, k-1) \quad (5-18)$$

此式可看作 $S(n, k)$ 满足的另一种递推式.

证明. 我们使用球盒模型证明此结论.

任选一个小球, 将其记作 a , 然后考虑不包含 a 的所有 $k-1$ 个盒子; 这些盒子中至少放入 $k-1$ 个小球 (否则将出现空盒), 至多放入除 a 以外的全部 $n-1$ 个小球. 假设我们将 j ($k-1 \leq j \leq n-1$) 个小球放入了这 $k-1$ 个盒子中, 此时一共有 $\binom{n-1}{j}$ 种方式选出这 j 个小球, 又有 $S(j, k-1)$ 种方式将其放入盒子中. 再由乘法原理和加法原理, 即可得到要证的结论. \square

定理 5.8. $S(n, k)$ 在 n 取固定值时是单峰的. 形式化地, 给定任意正整数 n , 数列

$$\{S(n, 0), S(n, 1), S(n, 2), \dots, S(n, n)\}$$

先单调递增再单调递减. 这即是说, 存在非负整数 $K(n)$, 使得

$$\begin{aligned} S(n, 0) &\leq S(n, 1) \leq \dots \leq S(n, K(n)-1) \leq S(n, K(n)) \\ S(n, n) &\leq S(n, n-1) \leq \dots \leq S(n, K(n)+1) \leq S(n, K(n)) \end{aligned}$$

证明. 对 n 施加归纳. 当 $n=1, 2, 3$ 时, 观察表 5.1 可知, 令

$$K(1) = 1, K(2) = 1, K(3) = 2$$

即可满足单峰性要求. 当 $n > 3$ 时, 我们的目标是分别证明如下两个命题:

- 单调递增部分: $S(n+1, 0) \leq S(n+1, 1) \leq \dots \leq S(n+1, K(n))$
- 单调递减部分: $S(n+1, n+1) \leq S(n+1, n) \leq \dots \leq S(n+1, K(n)+1)$

从而完成归纳.

先证单调递增部分命题, 即

$$S(n+1, k) - S(n+1, k-1) > 0 \quad (k = 1, 2, \dots, K(n))$$

当 $k = 1$ 时, 我们有

$$S(n+1, 1) = 1 > S(n+1, 0) = 0$$

当 $k = 2, 3, \dots, K(n)$ 时, 利用递推式 (5-8) 与归纳假设, 我们有

$$\begin{aligned} & S(n+1, k) - S(n+1, k-1) \\ &= [S(n, k-1) + k \cdot S(n, k)] - [S(n, k-2) + (k-1)S(n, k-1)] \\ &= k[S(n, k) - S(n, k-1)] + [S(n, k-1) - S(n, k-2)] + S(n, k-1) \\ &> k \cdot 0 + 0 + 0 = 0 \end{aligned}$$

因此单调递增部分命题成立.

再证单调递减部分命题, 即

$$S(n+1, k) - S(n+1, k+1) > 0 \quad (k = K(n) + 1, K(n) + 2, \dots, n)$$

当 $k = n$ 时, 我们有

$$S(n+1, n) = \binom{n+1}{2} > S(n+1, n+1) = 1$$

当 $k = K(n) + 1, K(n) + 2, \dots, n-1$ 时, 利用递推式 (5-18) 与归纳假设, 我们有

$$\begin{aligned} & S(n+1, k) - S(n+1, k+1) \\ &= \sum_{j=k-1}^n \binom{n}{j} S(j, k-1) - \sum_{j=k}^n \binom{n}{j} S(j, k) \\ &= \binom{n}{k-1} S(k-1, k-1) + \sum_{j=k}^n \binom{n}{j} [S(j, k) - S(j, k+1)] \\ &> 0 + \sum_{j=k}^n \binom{n}{j} \cdot 0 = 0 \end{aligned}$$

因此单调递减部分命题也成立. 由上述两个命题可知, 无论 $S(n+1, K(n))$ 和 $S(n+1, K(n)+1)$ 的大小关系如何, 其中必有一个是数列的峰值, 即

$$K(n+1) \in \{K(n), K(n)+1\}$$

这表明归纳步骤成立, 从而由数学归纳法, 原命题得证. □

5.4 第一类 Stirling 数

第一类 Stirling 数 $s(n, k)$ 对具有特定循环数目的置换进行计数. 第七章中我们将详细讨论置换和循环的概念, 本节中, 我们先使用集合语言描述第一类 Stirling 数的定义.

定义 5.9. 设 A 是一 n 元素集. 将集合 A 划分为 k 个非空子集, 并对每个子集中的元素作圆排列, 总方案数定义为第一类 Stirling 数的绝对值:

$$|s(n, k)| = (-1)^{n-k} s(n, k) \quad (5-19)$$

即, 第一类 Stirling 数的真实值等于上述方案数乘上符号项 $(-1)^{n-k}$.

例如, 当 $n = 3$ 时, 不妨设 $A = \{1, 2, 3\}$, 则我们有

$$s(3, 1) = 2, \quad s(3, 2) = -3, \quad s(3, 3) = 1$$

下面依次解释每个值的含义:

- 当 $k = 1$ 时, A 中的元素分入同一个非空子集中, 这只有 1 种方案; 这些元素则有两种不同的圆排列方式:



因此, 总方案数为 $s(3, 1) = 2$;

- 当 $k = 2$ 时, A 中的 3 个元素划分为 2 个非空子集, 结果必然是划分出一个单元元素集 A_1 与一个 2 元素集 A_2 , A_1 中的元素可以任取 $\{1, 2, 3\}$ 之一, 从而有 3 种不同的方案; 由于 A_1, A_2 中的元素数目均不超过 2, 对每个集合内的元素作圆排列均只会有一种方案. 因此, 总方案数为 $-s(3, 2) = 3$, 即 $s(3, 2) = -3$.
- 当 $k = 3$ 时, A 中的 3 个元素划分为 3 个非空子集, 只有一种方案; 每个子集均为单元元素集, 因此对其中元素作圆排列时也只会有一种方案. 因此, 总方案数为 $s(3, 3) = 1$.

第一类 Stirling 数 $s(n, k)$ 满足和第二类 Stirling 数类似的边界条件:

$$\begin{aligned} s(n, 0) &= 0 & (n > 0) \\ s(n, n) &= 1 & (n \geq 0) \\ s(n, k) &= 0 & (k > n) \end{aligned}$$

下面我们通过定义推导 $|s(n, k)|$ 满足的一个递推式. 我们可任选一个 $a \in A$, 然后考虑与 a 共同参与圆排列的元素:

- 若 a 单独形成一个圆排列, 则集合 A 中其余的 $n - 1$ 个元素需要划分为 $k - 1$ 个非空子集并各自进行圆排列, 方案数为 $|s(n - 1, k - 1)|$;

- 若 a 和其他元素共同形成圆排列, 则可设想先将 $n-1$ 个元素划分为 k 个非空子集并进行圆排列, 有

$$|s(n-1, k)|$$

种方案; 再将 a 插入某个圆排列中, 对于每个已在圆排列中的元素, 其右侧都有一个位置可供 a 插入, 一共有 $n-1$ 个可选的位置, 因此总的方案数为 $(n-1)|s(n-1, k)|$.

由加法原理可得

$$|s(n, k)| = |s(n-1, k-1)| + (n-1)|s(n-1, k)| \quad (5-20)$$

注意到 $s(n, k)$ 与 $s(n-1, k-1)$ 具有相同的符号, 而与 $s(n-1, k)$ 具有相反的符号, 由此我们可从上式中去掉绝对值, 得到关于 $s(n, k)$ 的递推式:

$$s(n, k) = s(n-1, k-1) - (n-1)s(n-1, k) \quad (5-21)$$

与第二类 Stirling 数类似, 第一类 Stirling 数也有代数定义. 仍然考虑由 x 的全体不高于 n 次的多项式构成的线性空间, 它有一组自然基与一组由下降幂构成的基:

$$\{x^0, x^1, x^2, \dots, x^n\}$$

$$\{(x)_0, (x)_1, (x)_2, \dots, (x)_n\}$$

下降幂基中的每个元素必能表示为自然基的线性组合, 其系数即定义为第一类 Stirling 数.

定义 5.10 (第一类 Stirling 数的代数定义). 设 n 是非负整数. 定义第一类 Stirling 数 $s(n, k)$ 为将 $(x)_n$ 表示为

$$\{x^0, x^1, x^2, \dots, x^n\}$$

的线性组合时 x^k 项的系数:

$$(x)_n = \sum_{k=0}^n s(n, k)x^k \quad (5-22)$$

为了说明上述代数定义与组合定义等价, 我们下面证明代数定义下的 $s(n, k)$ 满足与组合定义完全相同的递推关系. 使用数学归纳法, 对 n 施加归纳, 由归纳假设, 我们有

$$\begin{aligned} (x)_n &= (x - n + 1) \cdot (x)_{n-1} \\ &= (x - n + 1) \sum_{k=0}^{n-1} s(n-1, k)x^k \\ &= \sum_{k=0}^{n-1} s(n-1, k)x^{k+1} - \sum_{k=0}^{n-1} (n-1)s(n-1, k)x^k \\ &= \sum_{k=1}^n s(n-1, k-1)x^k - \sum_{k=1}^n (n-1)s(n-1, k)x^k \\ &= \sum_{k=1}^n [s(n-1, k-1) - (n-1)s(n-1, k)]x^k \end{aligned}$$

由此可得

$$s(n, k) = s(n-1, k-1) - (n-1)s(n-1, k) \quad (5-23)$$

对于 $n=0$ 或 $k=0$ 时的边界条件, 考察式 (5-22) 的两侧, 我们有

$$\begin{aligned} s(n, 0) &= 0 \quad (n > 0) \\ s(0, 0) &= 1 \end{aligned}$$

可见此递推关系的初值和递推式均与组合定义相同, 因此两种定义等价.

为了在 $s(n, k)$ 与 $|s(n, k)|$ 之间作出区分, $s(n, k)$ 又可称为有符号的第一类 Stirling 数, $|s(n, k)|$ 则称为无符号的第一类 Stirling 数. 值得一提的是, $|s(n, k)|$ 本身也有代数定义. 和下降幂相对应地, 我们可定义 x 的上升幂 (rising factorial):

$$x^{(n)} = \prod_{j=1}^n (x+j-1) = x(x+1)(x+2) \cdots (x+n-1)$$

由此我们可给出如下定义:

定义 5.11 (无符号的第一类 Stirling 数的代数定义). 设 n 是非负整数. 定义无符号的第一类 Stirling 数 $|s(n, k)|$ 为将 $x^{(n)}$ 表示成:

$$\{x^0, x^1, x^2, \dots, x^n\}$$

的线性组合时 x^k 项的系数:

$$x^{(n)} = \sum_{k=0}^n |s(n, k)| x^k \quad (5-24)$$

我们略去此定义与组合定义间的等价性证明, 感兴趣的读者可自行推导.

下面我们求解第一类 Stirling 数的母函数. 如果固定 n 的取值, 则式 (5-22) 与 (5-24) 实际上就是 $s(n, k)$ 和 $|s(n, k)|$ 的母函数. 下面讨论固定 k 的取值时的情况. 简便起见, 我们先讨论无符号的第一类 Stirling 数 $|s(n, k)|$; 它的定义涉及圆排列, 因此我们使用适合解决排列问题的指数型母函数:

$$G_{e, \text{unsigned}}(x) = \sum_{n=0}^{\infty} \frac{|s(n, k)|}{n!} = \sum_{n=k}^{\infty} \frac{|s(n, k)|}{n!}$$

在 5.3 节中我们指出, 球盒模型的母函数等于盒与球的母函数的复合. 第一类 Stirling 数的组合场景可以描述为如下球盒模型:

将 n 个不同的小球放入 k 个相同的盒子, 禁止出现空盒; 每个盒子中的小球分别作圆排列, 求方案数.

在此模型中, 盒子数目对应的指数型母函数仍为

$$G_{e,\text{盒}}(x) = \frac{x^k}{k!}$$

每个盒子中的小球作圆排列, 因此其指数型母函数即为圆排列数的指数型母函数. 每个圆排列中至少包含一个元素, 长度为 $n \geq 1$ 的圆排列共有 $(n-1)!$ 种, 因此其母函数为

$$G_{e,\text{球}}(x) = \sum_{n=1}^{\infty} \frac{(n-1)!}{n!} x^n = \sum_{n=1}^{\infty} \frac{x^n}{n!} = \ln\left(\frac{1}{1-x}\right)$$

由此可得

$$G_{e,\text{unsigned}}(x) = \frac{1}{k!} \ln^k\left(\frac{1}{1-x}\right) \tag{5-25}$$

此即为无符号的第一类 Stirling 数的指数型母函数. 讨论上式展开式中各项的符号, 可知对于有符号的第一类 Stirling 数, 其指数型母函数为

$$G_{e,\text{signed}}(x) = \frac{(-1)^k}{k!} \ln^k\left(\frac{1}{1+x}\right) \tag{5-26}$$

与第二类 Stirling 数不同, 即使获得了上述指数型母函数, 我们也难以从中求出 $s(n, k)$ 的通项表达式. 因此, 对任意正整数 n, k , 我们一般仍然是使用递推式 (5-21) 计算 $s(n, k)$. 表 5.2 展示了 $n \leq 7$ 时 $s(n, k)$ 的值.

$s(n, k) \backslash k$	1	2	3	4	5	6	7
n							
1	1						
2	-1	1					
3	2	-3	1				
4	-6	11	-6	1			
5	24	-50	35	-10	1		
6	-120	274	225	85	-15	1	
7	720	-1764	1624	-735	175	-21	1

表 5.2: $n \leq 7$ 时第一类 Stirling 数 $s(n, k)$ 的值

第六章 容斥原理

容斥原理是离散概率和组合理论中最有用的列举原则之一。通过巧妙地应用容斥原理，人们找到了许许多多组合问题的解决方案。

G.-C. Rota, 意大利裔美籍数学家

本章中，我们介绍一条重要的计数法则，称为**容斥原理**（Inclusion-exclusion Principle，简称为 IEP）。容斥原理可对若干已知集合的并集中的元素进行计数。本书的第一章中讨论了加法法则，它指出若有限集 A 可表示为一系列集合的并：

$$A = A_1 \cup A_2 \cup \cdots \cup A_n$$

则在这些集合两两交集为空的前提下， A 的大小等于这些集合的大小之和，即

$$|A| = |A_1 \cup A_2 \cup \cdots \cup A_n| = |A_1| + |A_2| + \cdots + |A_n|$$

容斥原理对上述情况作出了推广，它允许 A_1, A_2, \dots, A_n 之间的交集非空。容斥原理指出， A_1, A_2, \dots, A_n 的并集中的元素数目（即 $|A|$ ），可以通过它们之间的交集的元素数目间接计算得出。

6.1 容斥原理及其证明

容斥原理的核心思想是“不重不漏”，每个需要被计数的元素均恰好被统计 1 次，每个不需要被计数的元素均恰好被统计 0 次，这一原则与我们之前介绍的计数问题求解手段一脉相承。下面我们讨论一些简单的例子以展示容斥原理如何运用这种思想。

例 6.1.1. 求 1 到 20 的整数中，能被 2 或 3 整除的数的数目。

这个例子中，我们先分别考虑能被 2 整除的正整数，以及能被 3 整除的正整数。令前者构成集合 A_1 ，此集合的大小是

$$|A_1| = \left\lfloor \frac{20}{2} \right\rfloor = 10$$

后者构成集合 A_2 ，此集合的大小是

$$|A_2| = \left\lfloor \frac{20}{3} \right\rfloor = 6$$

设能被 2 或 3 整除的正整数构成集合 A , 则显然有

$$A = A_1 \cup A_2$$

考虑 $|A_1| + |A_2|$ 的组合含义: 它没有统计集合 A 以外的任何元素, 同时对 A 中的所有元素都统计了至少 1 次. 问题在于, 有一部分 A 中的元素同时属于 A_1, A_2 两个集合, 因此被统计了 2 次; 换句话说, $A_1 \cap A_2$ 中的元素被重复统计了. 我们可用一种称为 Venn 图的图形展示上述场景, 其中我们用两个圆分别表示集合 A_1, A_2 , 圆之间重叠的部分代表两个集合的交集 $A_1 \cap A_2$, 如图 6.1 所示. 显然, 若我们将两个圆形分别统计 1 次, 则两圆重叠的部分将被统计 2 次.

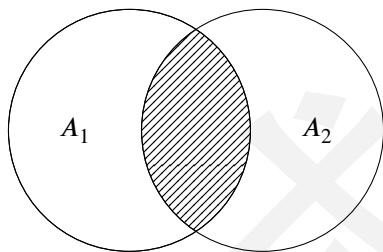


图 6.1: 两个集合 A_1, A_2 对应的 Venn 图

由于 $|A_1| + |A_2|$ 将 $A_1 \cap A_2$ 中的元素统计了 2 次, 因此我们从中减去

$$|A_1 \cap A_2| = \left\lfloor \frac{20}{6} \right\rfloor = 3$$

即可使 A 中所有元素均恰好被统计一次. 这表明

$$|A| = |A_1| + |A_2| - |A_1 \cap A_2| \quad (6-1)$$

此式即为容斥原理在两个集合上的形式. 利用式 (6-1), 我们可计算出 $|A| = 10 + 6 - 3 = 13$; 即, 1 到 20 的正整数中有 13 个可被 2 或 3 整除.

例 6.1.2. 求 1 到 600 的整数中, 能被 2 或 3 或 5 整除的数的数目.

这个例子与上个例子基本相同, 不过要考虑的集合数目变为了 3 个, 我们用这个例子展示容斥原理在 3 个集合上的形式. 首先仍然分别考虑由 1 到 600 的整数中能被 2、3、5 整除的数构成的集合:

$$|A_1| = \left\lfloor \frac{600}{2} \right\rfloor = 300$$

$$|A_2| = \left\lfloor \frac{600}{3} \right\rfloor = 200$$

$$|A_3| = \left\lfloor \frac{600}{5} \right\rfloor = 120$$

我们令

$$A = A_1 \cup A_2 \cup A_3$$

并考虑 $|A_1| + |A_2| + |A_3|$ 的组合含义, 该式将 A 中的所有元素都统计了至少一次. 问题在于, 恰好同时属于 A_1, A_2, A_3 中两个集合的元素被统计了 2 次, 同时属于三个集合的元素则被统计了 3 次. 这时的 Venn 图如图 6.2 所示.

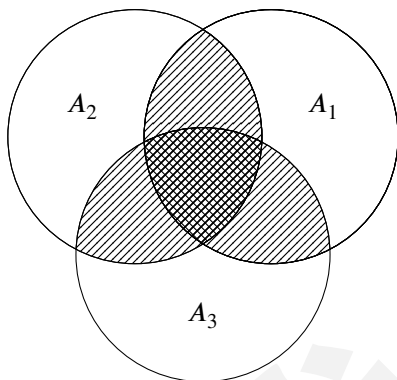


图 6.2: 三个集合 A_1, A_2, A_3 对应的 Venn 图

我们先处理被统计了 2 次的元素. 根据定义, 我们有

$$|A_1 \cap A_2| = \left\lfloor \frac{600}{6} \right\rfloor = 100$$

$$|A_1 \cap A_3| = \left\lfloor \frac{600}{10} \right\rfloor = 60$$

$$|A_2 \cap A_3| = \left\lfloor \frac{600}{15} \right\rfloor = 40$$

现在, 考虑式

$$|A_1| + |A_2| + |A_3| - |A_1 \cap A_2| - |A_2 \cap A_3| - |A_1 \cap A_3|$$

的组合含义: 由于我们从 $|A_1| + |A_2| + |A_3|$ 中减去了 A_1, A_2, A_3 两两之间交集的大小, 因此在上式中, 恰好同时属于 A_1, A_2, A_3 中两个集合的元素被计算了 $2 - 1 = 1$ 次, 符合“不重不漏”的原则了. 然而, 对于那些同时属于三个集合的元素, 它们在 $|A_1| + |A_2| + |A_3|$ 中被重复计算 3 次, 在 $-|A_1 \cap A_2| - |A_2 \cap A_3| - |A_1 \cap A_3|$ 中又被扣减 3 次, 最终被统计的次数变为 0 次, 反而被遗漏了. 因此, 我们还需在上式的基础上加上 A_1, A_2, A_3 三个集合的交集大小:

$$|A_1 \cap A_2 \cap A_3| = \left\lfloor \frac{600}{30} \right\rfloor = 20$$

从而得到

$$|A| = |A_1| + |A_2| + |A_3| - |A_1 \cap A_2| - |A_2 \cap A_3| - |A_1 \cap A_3| + |A_1 \cap A_2 \cap A_3| \quad (6-2)$$

此式即为容斥原理在三个集合上的形式. 利用式 (6-2), 我们可计算出

$$|A| = 300 + 200 + 120 - 100 - 60 - 40 + 20 = 440$$

即, 1 到 600 的整数中有 440 个能被 2 或 3 或 5 整除.

以上两个例子讨论的计数方法可总结为如下一般规律:

定理 6.1 (容斥原理的并集形式). 设 A_1, A_2, \dots, A_n 是一系列有限集, 则

$$\begin{aligned} |A_1 \cup A_2 \cup \dots \cup A_n| &= \sum_i |A_i| - \sum_{i < j} |A_i \cap A_j| + \sum_{i < j < k} |A_i \cap A_j \cap A_k| - \dots \\ &= \sum_{I \subseteq \{1, 2, \dots, n\}, I \neq \emptyset} (-1)^{|I|-1} \left| \bigcap_{i \in I} A_i \right| \end{aligned} \quad (6-3)$$

上述定理给出了求一系列集合并集中元素数量的方法. 但是, 在实际问题中, 这种形式的容斥原理较少直接使用, 一般还会在式 (6-3) 的两侧分别取补集. 应用 De Morgan 定律 (参见附录 A.2), 我们立即得到如下推论:

定理 6.2 (容斥原理). 设 U 是有限的全集, A_1, A_2, \dots, A_n 是一系列集合, 则

$$\begin{aligned} \left| \overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_n} \right| &= |U| - \sum_i |A_i| + \sum_{i < j} |A_i \cap A_j| - \sum_{i < j < k} |A_i \cap A_j \cap A_k| + \dots \\ &= \sum_{I \subseteq \{1, 2, \dots, n\}} (-1)^{|I|} \left| \bigcap_{i \in I} A_i \right| \end{aligned} \quad (6-4)$$

下面我们按照定理 6.2 中的表述证明容斥原理. 证明遵循“不重不漏”的核心思想: 我们证明, 全集 U 中的任何一个元素 x 在式 (6-4) 两侧被统计了相同的次数; 若 x 属于 A_1, A_2, \dots, A_n 中的任何一个集合, 则其被统计的次数为 0, 否则为 1.

先以 $n = 2$ 的最简单情形来举例说明. 这时, 式 (6-4) 为如下的形式:

$$\left| \overline{A_1} \cap \overline{A_2} \right| = |U| - |A_1| - |A_2| + |A_1 \cap A_2| \quad (6-5)$$

在上式左侧, 所有既不在 A_1 也不在 A_2 中的元素都被统计 1 次, 而其他元素被统计 0 次. 在上式右侧, 我们考虑元素 $x \in U$ 被统计的次数, 可以根据 x 是否属于 A_1 或 A_2 分成四类讨论:

- (1) $x \notin A_1, x \notin A_2$: x 被 $|U|$ 统计 1 次, 被 $|A_1|$ 、 $|A_2|$ 和 $|A_1 \cap A_2|$ 各统计 0 次, 总和为 $1 - 0 - 0 + 0 = 1$;
- (2) $x \in A_1, x \notin A_2$: x 被 $|U|$ 和 $|A_1|$ 各统计 1 次, 被 $|A_2|$ 和 $|A_1 \cap A_2|$ 各统计 0 次, 总和为 $1 - 1 - 0 + 0 = 0$;
- (3) $x \notin A_1, x \in A_2$: x 被 $|U|$ 和 $|A_2|$ 各统计 1 次, 被 $|A_1|$ 和 $|A_1 \cap A_2|$ 各统计 0 次, 总和为 $1 - 0 - 1 + 0 = 0$;
- (4) $x \in A_1, x \in A_2$: x 被 $|U|$ 、 $|A_1|$ 、 $|A_2|$ 和 $|A_1 \cap A_2|$ 各统计 1 次, 总和为 $1 - 1 - 1 + 1 = 0$.

可见, 在等式右侧, 只有既不在 A_1 也不在 A_2 中的元素恰好被统计了 1 次, 与等式左侧相同. 这表明容斥原理在 $n = 2$ 时确实成立.

上述讨论可推广至任意正整数 $n > 2$, 从而得到如下证明:

证明. 在式 (6-4) 左侧, 所有不属于 A_1, A_2, \dots, A_n 中任何一个集合的元素均被统计 1 次, 而其他元素均被统计 0 次. 在等式右侧, 元素 $x \in U$ 被统计的次数, 可以根据 x 是否属于 A_1, A_2, \dots, A_n 中的某些集合分成两类讨论:

(1) x 不属于 A_1, A_2, \dots, A_n 中的任何一个集合: x 被 $|U|$ 统计 1 次, 被其余 $2^n - 1$ 项的各统计 0 次, 总和为 $1 - 0 + 0 - \dots + (-1)^n 0 = 1$;

(2) x 恰好属于 A_1, A_2, \dots, A_n 中的 m ($m = 1, 2, \dots, n$) 个集合: x 被 $|U|$ 统计 $1 = \binom{m}{0}$

次; 由于 x 恰好属于 m 个集合, 因此它被 $\sum_i |A_i|$ 统计 $\binom{m}{1}$ 次、被 $\sum_{i < j} |A_i \cap A_j|$ 统计 $\binom{m}{2}$ 次, 等等. x 在等式右侧被统计的总次数为

$$\binom{m}{0} - \binom{m}{1} + \binom{m}{2} - \dots + (-1)^n \binom{m}{n} = \sum_{j=0}^n (-1)^j \binom{m}{j}$$

注意到当 $m < n$ 时 $\binom{m}{n} = 0$, 因此上式又等于

$$\binom{m}{0} - \binom{m}{1} + \binom{m}{2} - \dots + (-1)^m \binom{m}{m} = \sum_{j=0}^m (-1)^j \binom{m}{j} = (1 - 1)^m = 0$$

即, x 在等式右侧被统计 0 次.

综上, 在等式右侧, 只有不属于 A_1, A_2, \dots, A_n 中的任何一个集合的元素恰好被统计 1 次, 其余元素均被统计 0 次, 与等号左侧相同. 因此, 式 (6-4) 恒成立. \square

例 6.1.3. 假设一个学校只有数学、物理和化学三门课程. 已知选修这三门课程的学生分别有 170、130 和 120 人; 同时修数学、物理的学生有 45 人, 同时修数学、化学的学生有 20 人, 同时修物理、化学的学生有 22 人; 同时修全部三门课程的学生有 3 人. 求这学校中上课的学生的数量.

令 M 、 P 、 C 分别为修数学、物理和化学的学生构成的集合. 根据这个例子中的假设, 我们有

- $|M| = 170, |P| = 130, |C| = 120$;
- $|M \cap P| = 45, |M \cap C| = 20, |P \cap C| = 22$;
- $|M \cap P \cap C| = 3$.

上课的学生数量等于 $|M \cup P \cup C|$, 可利用容斥原理求解:

$$|M \cup P \cup C| = (170 + 130 + 120) - (45 + 20 + 22) + 3 = 336$$

故学校中共有 336 名上课的学生.

例 6.1.4. 求不超过 120 的质数的数目.

由于 $11^2 = 121 > 120$, 因此不超过 120 的合数必然是 2、3、5、7 的倍数. 令 A_p ($p = 2, 3, 5, 7$) 表示不超过 120 的 p 的倍数构成的集合. 我们有

- 单个集合的大小:

$$|A_2| = \left\lfloor \frac{120}{2} \right\rfloor = 60, \quad |A_3| = \left\lfloor \frac{120}{3} \right\rfloor = 40,$$

$$|A_5| = \left\lfloor \frac{120}{5} \right\rfloor = 24, \quad |A_7| = \left\lfloor \frac{120}{7} \right\rfloor = 17$$

- 两个集合交集的大小:

$$|A_2 \cap A_3| = \left\lfloor \frac{120}{6} \right\rfloor = 20, \quad |A_2 \cap A_5| = \left\lfloor \frac{120}{10} \right\rfloor = 12,$$

$$|A_2 \cap A_7| = \left\lfloor \frac{120}{14} \right\rfloor = 8, \quad |A_3 \cap A_5| = \left\lfloor \frac{120}{15} \right\rfloor = 8,$$

$$|A_3 \cap A_7| = \left\lfloor \frac{120}{21} \right\rfloor = 5, \quad |A_5 \cap A_7| = \left\lfloor \frac{120}{35} \right\rfloor = 3$$

- 三个集合交集的大小:

$$|A_2 \cap A_3 \cap A_5| = \left\lfloor \frac{120}{30} \right\rfloor = 4, \quad |A_2 \cap A_3 \cap A_7| = \left\lfloor \frac{120}{42} \right\rfloor = 2,$$

$$|A_2 \cap A_5 \cap A_7| = \left\lfloor \frac{120}{70} \right\rfloor = 1, \quad |A_3 \cap A_5 \cap A_7| = \left\lfloor \frac{120}{105} \right\rfloor = 1$$

- 四个集合交集的大小:

$$|A_2 \cap A_3 \cap A_5 \cap A_7| = \left\lfloor \frac{120}{210} \right\rfloor = 0$$

利用容斥原理, 可得

$$\begin{aligned} & \left| \overline{A_2} \cap \overline{A_3} \cap \overline{A_5} \cap \overline{A_7} \right| \\ &= 120 - (60 + 40 + 24 + 17) + (20 + 12 + 8 + 8 + 5 + 3) - (4 + 2 + 1 + 1) \\ &= 27 \end{aligned}$$

注意此结果并不等于 120 以内质数的数目. 首先, 1 不是质数, 但它不是任何质数的倍数, 因此被统计在内; 其次, 2, 3, 5, 7 这四个质数本身分别是它们自身的倍数, 因此均没有得到统计. 由此可知, 不超过 120 的质数的实际数目为

$$\pi(120) = 27 - 1 + 4 = 30$$

其中 $\pi(n)$ 称为质数计数函数, 表示 1 到 n 的正整数中质数的数目.

例 6.1.5. 求完全由 n 个布尔变量确定的布尔函数

$$f(x_1, x_2, \dots, x_n) : \{0, 1\}^n \mapsto \{0, 1\}$$

的数目. 所谓“完全”指的是不存在某个输入变量 x_i 对输出无任何影响; 换句话说, 对任意 $k \in \{1, 2, \dots, n\}$, 方程

$$f(x_1, x_2, \dots, x_{k-1}, 0, x_{k+1}, \dots, x_n) \neq f(x_1, x_2, \dots, x_{k-1}, 1, x_{k+1}, \dots, x_n)$$

均有解.

这个例子中, 我们要求布尔函数的每个输入均能对输出产生影响; 满足条件的函数数目不易直接计数, 因此可使用容斥原理进行间接计数. 设满足条件的函数构成集合 A , 由 n 个布尔变量 x_1, x_2, \dots, x_n 确定的全体布尔函数 $f(x_1, x_2, \dots, x_n)$ 构成全集 U , 则我们有

$$|U| = 2^{2^n}$$

其含义是每个布尔函数都可由其真值表唯一确定, 真值表中包含函数的所有 2^n 种输入, 每种输入的输出生可以独立地选择为 0 或 1. U 中有一部分函数, 其输出并不依赖于某些输入变量. 令 A_i 表示输出不依赖于输入 x_i 的布尔函数 f 构成的集合, 则有

$$A = \overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_n}$$

现在考虑各 A_i 及其交集的大小. 注意到, 若函数的输出不依赖于输入中的 k ($k = 1, 2, \dots, n$) 个变量, 则它就相当于一个由 $n - k$ 个输入变量确定的布尔函数, 这样的函数有 $2^{2^{n-k}}$ 个. 利用容斥原理, 可求出满足条件的函数数目为

$$\begin{aligned} |A| &= \left| \overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_n} \right| \\ &= \sum_{I \subseteq \{1, 2, \dots, n\}} (-1)^{|I|} \left| \bigcap_{i \in I} A_i \right| \\ &= \sum_{I \subseteq \{1, 2, \dots, n\}} (-1)^{|I|} \cdot 2^{2^{n-|I|}} \\ &= 2^{2^n} - \binom{n}{1} \cdot 2^{2^{n-1}} + \binom{n}{2} \cdot 2^{2^{n-2}} - \dots + (-1)^n \cdot 2^{2^0} \end{aligned}$$

例如, 当 $n = 2$ 时, 上式给出

$$|A| = \left| \overline{A_1} \cap \overline{A_2} \right| = 16 - 8 + 2 = 10$$

因此, 在全部 $2^{2^2} = 16$ 个两输入的布尔函数中, 有 10 个函数的输出由两个输入共同决定; 其余 6 个不满足此要求的函数的输出分别是 0 和 1、 x_1 和 x_2 , 以及 $\overline{x_1}$ 和 $\overline{x_2}$.

6.2 带约束的排列问题

容斥原理可用于解决一大类带有约束的排列问题. 这些问题可能要求排列中不出现特定图象 (或称为“子串”), 或者要求某些元素不能出现在排列的特定位置上. 当多个这样的约束共同存在导致难以直接统计满足要求的排列数目时, 通常可使用容斥原理进行间接计数.

例 6.2.1. a、b、c、d、e、f 六个字母作全排列, 禁止排列中出现子串 ace 或 df, 求满足要求的排列数.

令 U 表示上述六个字母组成的全体全排列构成的集合, A 表示满足要求的排列构成的集合; A_1 表示出现了 ace 的排列构成的集合, A_2 表示出现了 df 的排列构成的集合. 则我们有

$$A = \overline{A_1} \cap \overline{A_2}$$

集合 A_1 中的排列固定出现 ace, 因此实际上相当于 ace、b、d、f 这四个元素的排列, 从而有

$$|A_1| = 4!$$

同理, 集合 A_2 中的排列实际上相当于 a、b、c、df、e 这五个元素的排列, 从而有

$$|A_2| = 5!$$

集合 $A_1 \cap A_2$ 中的排列要求 ace 和 df 同时出现, 因此实际上相当于 ace、b、df 这三个元素的排列, 从而有

$$|A_1 \cap A_2| = 3!$$

利用容斥原理, 可得

$$\begin{aligned} |A| &= |\overline{A_1} \cap \overline{A_2}| \\ &= |U| - |A_1| - |A_2| + |A_1 \cap A_2| \\ &= 6! - 4! - 5! + 3! = 582 \end{aligned}$$

因此, 满足要求的排列一共有 582 个.

例 6.2.2. 用 26 个小写英文字母作无重复的全排列, 其中禁止出现下列子串:

dog, god, gum, depth, thing

求满足要求的排列数.

这个例子中, 我们需要注意多个子串同时出现时的字母重叠问题. 令 A_1, A_2, A_3, A_4, A_5 分别表示出现 dog、god、gum、depth 和 thing 这五种图象的排列数目. 使用与上例类似的思想, 可得

$$|A_1| = |A_2| = |A_3| = 24!, |A_4| = |A_5| = 22!$$

对于两种图象同时出现的情况, 尽管一共有 10 种不同的组合, 但由于要求字母不能重复使用, 实际上仅有 6 种组合是可能出现的, 它们分别是:

$$\begin{aligned} \text{dog} + \text{gum} : & |A_1 \cap A_3| = 22! \\ \text{god} + \text{depth} : & |A_2 \cap A_4| = 20! \\ \text{thing} + \text{god} : & |A_2 \cap A_5| = 20! \\ \text{thing} + \text{gum} : & |A_3 \cap A_4| = 20! \\ \text{thing} + \text{god} : & |A_3 \cap A_5| = 20! \\ \text{depth} + \text{thing} : & |A_4 \cap A_5| = 22! \end{aligned}$$

若三种图象同时出现, 则仅有一种可能性:

$$\text{depth} + \text{thing} + \text{gum} : |A_3 \cap A_4 \cap A_5| = 17!$$

任意四种或五种图象则不可能同时出现. 利用容斥原理可知, 所求的排列数为

$$26! - (3 \cdot 24! + 2 \cdot 22!) + (22! + 4 \cdot 20! + 19!) - 17!$$

例 6.2.3. 使用 a、b、c、d 四种字符组成一个 n 位字符串, 要求 a、b、c 分别至少出现一次, 求满足条件的字符串数.

令 A_1, A_2, A_3 分别表示不出现字母 a、b、c 的字符串构成的集合. A_1 中的字符串不包含字母 a, 因此就相当于用 3 种字母进行可重复的排列; 对集合 A_2, A_3 我们有类似的结论, 因此

$$|A_1| = |A_2| = |A_3| = 3^n$$

类似地, 我们有

$$|A_1 \cap A_2| = |A_1 \cap A_3| = |A_2 \cap A_3| = 2^n$$

$$|A_1 \cap A_2 \cap A_3| = 1^n = 1$$

利用容斥原理, 我们有

$$\begin{aligned} |\overline{A_1} \cap \overline{A_2} \cap \overline{A_3}| &= 4^n - 3 \cdot 3^n + 3 \cdot 2^n - 1 \\ &= 4^n - 3^{n+1} + 3 \cdot 2^n - 1 \end{aligned}$$

下面我们讨论带约束的排列问题中一类重要的特例. 我们关注 n 个元素

$$\{1, 2, \dots, n\}$$

构成的全排列 $a_1 a_2 \dots a_n$, 要求它满足一系列形如

$$a_i \notin X_i \quad (1 \leq i \leq n, X_i \subseteq \{1, 2, \dots, n\})$$

的约束. 上述形式的问题统称为**有禁止位置的排列**, 它相当于要求每个元素都不能出现在排列中特定的位置上.

例 6.2.4. 求 $\{1, 2, \dots, n\}$ 构成的所有排列 $a_1 a_2 \dots a_n$ 中, 满足 $a_1 \neq 1, a_2 \neq 2, \dots, a_n \neq n$ 的排列数.

这个例子实际上就是在 5.2 节中讨论过的错位排列问题; 错位排列可视为有禁止位置的排列的一个特例, 其约束条件为

$$a_i \notin \{i\} \quad (i = 1, 2, \dots, n)$$

我们展示如何使用容斥原理求解错位排列问题.

令 U 为全集, 表示由全体 $\{1, 2, \dots, n\}$ 的排列构成的集合; D 为所有错位排列构成的集合, A_i 表示满足 $a_i = i$ 的所有排列的集合 ($i = 1, 2, \dots, n$). 显然,

$$D = \overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_n}$$

考虑各 A_i 及其交集的大小: A_i 中的排列固定了一个位置上的元素, 对于其余 $n-1$ 个元素则没有任何约束, 因此相当于对这 $n-1$ 个元素作任意的排列, 从而我们有

$$|A_i| = (n-1)! \quad (1 \leq i \leq n)$$

类似地, $A_i \cap A_j$ 中的排列固定了两个位置上的元素, 因此相当于对其余 $n-2$ 个元素作任意的排列, 即

$$|A_i \cap A_j| = (n-2)! \quad (1 \leq i < j \leq n)$$

以此类推, A_1, A_2, \dots, A_n 中任意 m 个不同集合的交集大小为 $(n-m)!$. 利用容斥原理, 我们有

$$\begin{aligned} |D| &= |\overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_n}| \\ &= |U| - \sum_i |A_i| + \sum_{i < j} |A_i \cap A_j| - \dots + (-1)^n |A_1 \cap A_2 \cap \dots \cap A_n| \\ &= n! - \binom{n}{1}(n-1)! + \binom{n}{2}(n-2)! - \dots + (-1)^n \binom{n}{n}(n-n)! \\ &= n! - \frac{n!}{1!} + \frac{n!}{2!} + \dots + \frac{n!}{n!} \\ &= \sum_{k=0}^n (-1)^k \frac{n!}{k!} \end{aligned}$$

这与我们在 5.2 节中得出的结果相同.

下面我们讨论如何求解一般形式的有禁止位置的排列问题. 我们采用与错位排列相同的思路, 令全集 U 表示全体长度为 n 的排列构成的集合, A_i ($i = 1, 2, \dots, n$) 表示第 i 个位

置上的元素违反约束（即 $a_i \in X_i$ ）的排列数目，则满足所有约束的排列构成的集合 A 即可表示为

$$A = \overline{A_1} \cap \overline{A_2} \cap \cdots \cap \overline{A_n}$$

我们用容斥原理求解 $|A|$ 。考虑容斥原理式中的某一项

$$\left| A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_k} \right| \quad (1 \leq i_1 < i_2 < \cdots < i_k \leq n, k = 1, 2, \dots, n)$$

此项的含义是：对于排列中 i_1, i_2, \dots, i_k 这 k 个位置上的元素，需满足

$$a_{i_j} \in X_j \quad (j = 1, 2, \dots, n)$$

而其余 $n - k$ 个元素可以任意作排列，在 i_1, i_2, \dots, i_k 这些位置上的元素已经确定的情况下，方案数即为 $(n - k)!$ 。因此，我们有

$$\sum_{i_1 < i_2 < \cdots < i_k} \left| A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_k} \right| = r_k(n - k)!$$

其中 r_k 表示选出排列中的任意 k 个位置 i_1, i_2, \dots, i_k ，并在每个位置上放置不同的元素，使这些元素均不满足约束条件（即使得 $a_i \in X_i$ ）的方案数。

r_k 可通过一种称为**棋盘多项式**（rook polynomial）的工具有效求解。考虑一个 n 行 n 列的棋盘 C ，我们规定在棋盘上第 i 行、第 j 列的位置放上一枚棋子，代表令排列的第 i 个位置上的元素为 j ，即 $a_i = j$ 。利用棋盘的概念，我们可重新定义 r_k ：

设 C 是一 n 行 n 列的棋盘，其中仅有如下坐标处允许落子：

$$(i, j) \quad (1 \leq i \leq n, j \in X_i)$$

则 $r_k = r_k(C)$ 定义为将 k 枚棋子放入棋盘 C ，且任何两枚棋子不在同一行或同一列的方案数。

对于 $k = 0, 1, 2, \dots, n$ ，各 $r_k(C)$ 可构成一个数列，而我们可利用母函数方法求解此数列。 $\{r_k(C)\}$ 的母函数称为 C 的棋盘多项式：

定义 6.3. 设 C 是一 n 行 n 列的棋盘。定义 C 的棋盘多项式为

$$R(C) = \sum_{k=0}^n r_k(C) x^k \quad (6-6)$$

规定对于任意 C ，均有 $r_0(C) = 1$ 。

利用棋盘多项式，我们即可利用容斥原理给出带禁止位置的排列的解：

定理 6.4. 考虑任一带禁止位置的排列问题, 其中排列的长度为 n , 排列的所有禁止位置构成棋盘 C :

$$C = \{(i, j)\} \quad (1 \leq i \leq n, j \in X_i)$$

若 C 的棋盘多项式为

$$R(C) = \sum_{k=0}^n r_k(C) x^k$$

则原带禁止位置的排列问题的解为

$$\sum_{k=0}^n (-1)^k r_k(C) (n-k)! = n! - r_1(C) (n-1)! + r_2(C) (n-2)! - \cdots + (-1)^n r_n(C) \quad (6-7)$$

下面我们讨论如何求出 $r_k(C)$. 对于较为复杂的棋盘, 合法的落子方案数量不易直接求解, 因此我们转而研究 $r_k(C)$ 应当满足的递推关系. 假设已经决定在 C 的某一个棋盘格放置一枚棋子, 那么这个格子所在的行和列就都无法放置其他棋子了, 在研究后续棋子的布置时, 即可从棋盘 C 中去掉这个格子所在的行和列. 相反, 如果决定不在这个棋盘格放置棋子, 则可从棋盘 C 中去掉这个格子本身. 因此, 根据某一棋盘格是否落子, 可以将 $r_k(C)$ 分为两部分考虑, 其中每一部分都研究一个简化过 (即可落子位置更少) 的棋盘:

推论 6.5. 设 C 是一棋盘, 其中有某一可落子的棋盘格. 令 C_i 是从 C 中移除此棋盘格所在的行、列后所得的棋盘, C_e 则是从 C 中仅移除该棋盘格后所得的棋盘. 则

$$r_k(C) = r_{k-1}(C_i) + r_k(C_e)$$

利用上述推论, 我们可进一步求出 $R(C)$ 满足的递推式:

$$\begin{aligned} R(C) &= \sum_{k=0}^n r_k(C) x^k \\ &= 1 + \sum_{k=1}^n [r_{k-1}(C_i) + r_k(C_e)] x^k \\ &= x \sum_{k=0}^{n-1} r_k(C_i) x^k + \sum_{k=0}^n r_k(C_e) x^k \\ &= x R(C_i) + R(C_e) \end{aligned} \quad (6-8)$$

式 (6-8) 允许我们将复杂棋盘逐步分解为相对比较简单棋盘, 求出其棋盘多项式, 进而求出复杂棋盘上的棋子放置方案数. 我们通过几个例子展示如何进行这种化简.

例 6.2.5. 有如下棋盘多项式:

- (1) $R(\square) = 1 + x$;
- (2) $R\left(\begin{smallmatrix} \square \\ \square \end{smallmatrix}\right) = 1 + 2x$;

$$(3) R\left(\begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array}\right) = (1+x)^2.$$

证明. 分别讨论每个棋盘上的落子方案:

- (1) 对于 1×1 的棋盘, 其上仅有一个可落子位置, 其上最多落一个子; 根据棋盘多项式的定义, 可以直接得出 $R(\square) = 1+x$;
- (2) 对于 1×2 的棋盘, 我们可指定其上一个棋盘格 (以 \star 记号标出), 并应用递推式 (6-8):

$$R\left(\begin{array}{|c|c|} \hline \star & \square \\ \hline \square & \square \\ \hline \end{array}\right) = xR(\emptyset) + R(\square) = x + (1+x) = 1+2$$

- (3) 对于一个仅能在对角线上落子的 2×2 棋盘, 我们同样可指定其上一个棋盘格, 并应用递推式 (6-8):

$$R\left(\begin{array}{|c|c|} \hline \star & \square \\ \hline \square & \square \\ \hline \end{array}\right) = xR(\square) + R(\square) = x(1+x) + (1+x) = (1+x)^2 \quad \square$$

定理 6.6. 设 C 是一棋盘. 如果 C 能被划分为 C_1 和 C_2 两部分, 并且 C_1 的任何一个棋盘格所在的行、列均没有 C_2 的棋盘格, 则称 C_1, C_2 不相交, 且有

$$R(C) = R(C_1)R(C_2) \quad (6-9)$$

证明. 由于 C_1 和 C_2 不共享任何一行或一列, 因此在 C_1 中放置棋子对 C_2 完全没有影响. 这表明, C 中任何一个棋子放置方案均可以分解为 C_1, C_2 两个棋盘上独立的棋子放置方案:

$$r_k(C) = \sum_{i=0}^k r_i(C_1)r_{k-i}(C_2)$$

注意到, 若 $i+j > n$, 则必有 $r_i(C_1)r_j(C_2)x^{i+j} = 0$, 这是因为 C 仅有 n 行 n 列, 无法放置多于 n 个棋子. 由此我们有

$$\begin{aligned} R(C) &= \sum_{k=0}^n \left(\sum_{i=0}^k r_i(C_1)r_{k-i}(C_2) \right) x^k \\ &= \sum_{i+j \leq n} r_i(C_1)r_j(C_2)x^{i+j} \\ &= \sum_{i,j \leq n} r_i(C_1)r_j(C_2)x^{i+j} \\ &= \left(\sum_{i=0}^n r_i(C_1)x^i \right) \left(\sum_{j=0}^n r_j(C_2)x^j \right) \\ &= R(C_1)R(C_2) \quad \square \end{aligned}$$

例 6.2.6. 求解棋盘多项式: $R\left(\begin{array}{ccc} \square & \square & \\ & \square & \square \\ & & \square \end{array}\right), R\left(\begin{array}{ccc} \square & & \\ \square & \square & \\ & \square & \square \end{array}\right).$

求解这类较为复杂的棋盘多项式时, 核心技巧是指定合适的单元格应用递推式 (6-8), 然后再使用递推式 (6-9) 将大棋盘分解为不相交的小棋盘.

$$\begin{aligned} R\left(\begin{array}{ccc} \square & \square & \\ & \star & \square \\ & & \square \end{array}\right) &= xR\left(\begin{array}{cc} \square & \\ & \square \end{array}\right) + R\left(\begin{array}{ccc} \square & \square & \\ & & \square \end{array}\right) \\ &= x(1+x)^2 + (1+2x)^2 \\ &= 1 + 5x + 6x^2 + x^3 \end{aligned}$$

$$\begin{aligned} R\left(\begin{array}{ccc} \square & & \\ \square & \star & \\ & \square & \square \end{array}\right) &= xR\left(\begin{array}{cc} \square & \\ & \square \end{array}\right) + R\left(\begin{array}{ccc} \square & & \\ \square & & \square \\ & \square & \square \end{array}\right) \\ &= x(1+x)(1+2x) + (1+2x)(1+3x+x^2) \\ &= 1 + 6x + 10x^2 + 4x^3 \end{aligned}$$

例 6.2.7. 求解棋盘多项式: $R\left(\begin{array}{ccc} & & \square \\ \square & & \square \\ & \square & \square \end{array}\right).$

$$\begin{aligned} R\left(\begin{array}{ccc} & \square & \\ \square & & \star \\ & \square & \square \end{array}\right) &= xR\left(\begin{array}{cc} & \square \\ \square & \square \end{array}\right) + R\left(\begin{array}{ccc} & \square & \\ \square & \star & \square \end{array}\right) \\ &= xR(\square)R(\square) + xR\left(\begin{array}{cc} & \square \\ \square & \square \end{array}\right) + R\left(\begin{array}{ccc} & \square & \\ \square & & \square \\ & \square & \square \end{array}\right) \\ &= x(1+x)(1+2x) + x(1+x)^2 + (1+x)^4 \\ &= 1 + 6x + 11x^2 + 7x^3 + x^4 \end{aligned}$$

例 6.2.8. 考虑 $\{1, 2, 3, 4\}$ 的全排列 $a_1 a_2 a_3 a_4$, 分别求满足下列两组约束的排列数目:

- (1) $a_1 \notin \{3\}, a_2 \notin \{1, 4\}, a_3 \notin \{2, 4\}, a_4 \notin \{2\};$
- (2) $a_1 \notin \{2\}, a_2 \notin \{2, 3\}, a_3 \notin \{3, 4\}, a_4 \notin \{4\}.$

这两组约束分别对应于如图 6.3 所示的两个棋盘, 阴影代表禁区.

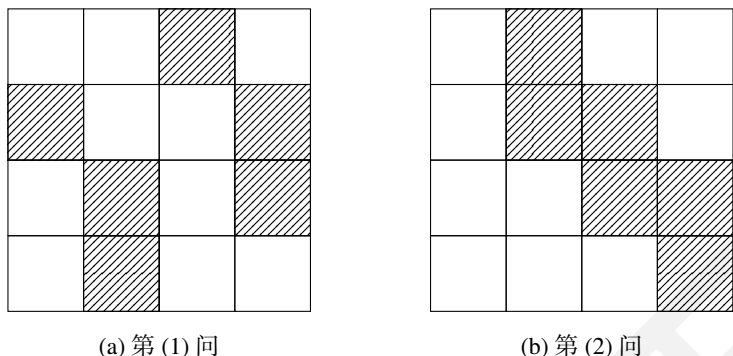


图 6.3: 例 6.2.8 对应的棋盘

两个禁区对应的棋盘多项式在前面的例子中已经计算过，分别为

$$R\left(\begin{array}{c} \square \\ \square \quad \square \\ \square \end{array}\right) = 1 + 6x + 11x^2 + 7x^3 + x^4$$

$$R\left(\begin{array}{c} \square \\ \square \quad \square \\ \square \quad \square \end{array}\right) = 1 + 6x + 10x^2 + 4x^3$$

由此可得，对于两组禁止位置约束，合法的排列方案数分别为

$$(1) 4! - 6 \cdot (4-1)! + 11 \cdot (4-2)! - 7 \cdot (4-3)! + 1 = 4;$$

$$(2) 4! - 6 \cdot (4-1)! + 10 \cdot (4-2)! - 4 \cdot (4-3)! = 4.$$

因此，两组约束下分别只有 4 种合法的排列。

在本节的最后，我们讨论一个经典的有禁止位置的排列问题，它称为**夫妻围坐问题** (ménage problem)。

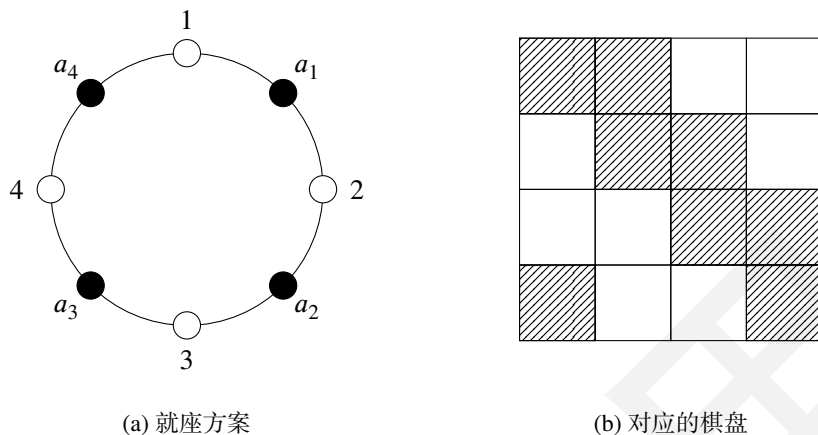
例 6.2.9. n 对夫妻围坐在一张固定的圆桌旁（不允许旋转），要求男性和女性间隔就座，同时每对夫妻的座位都不能相邻，求有多少种不同的就座方案。

这个例子中的问题有许多种解法，我们下面介绍其中一种较为直观的方法，称为“女士优先”方法。首先将每对夫妻分别编号为 $1, 2, \dots, n$ ，并为 n 位女性分配座位。由于男女间隔就座，因此这 n 位女性只可能坐在两组位置上，每组位置上的就座方案可视为对 n 位女性的一个排列，因此方案数为 $2 \cdot n!$ 。女性全部就座后，男性的就座即成为一个带约束的排列问题。按顺时针方向，设 i 号女性的下一个座位上安排 a_i 号男性就座，如图 6.4a 所示；则对应的约束条件为 $a_i \notin X_i$ ，其中

$$X_i = \{i, i+1\} \quad (i = 1, 2, \dots, n-1)$$

$$X_n = \{1, n\}$$

对应的棋盘如图 6.4b 所示。固定女性座位时，男性的就座方案数记为 M_n ，称为 Ménage 数。

图 6.4: $n = 4$ 时的夫妻围坐问题示意图

下面我们求解 Ménage 数. 考虑在图 6.4b 所示的禁区中放置 k 枚棋子的一个放置方案, 这相当于从

$$(1, 2), (2, 3), \dots, (n-1, n), (n, 1)$$

这 n 对数的 k 对中各取一个, 并且要求取出的数互不相同. 注意到, 在上面这个数对构成的序列中, 除 1 以外相同的数均相邻; 因此, 这等价于从数列

$$1, 2, 2, 3, \dots, (n-1), n, n, 1$$

中选取不相邻的 k 个数, 并且不能同时选取首尾的两个 1. 利用 1.6.2 节提到的不相邻组合的计算方法, 可知有

$$\binom{2n-k+1}{k}$$

种方案选取 k 个不相邻的数. 其中, 同时选取首尾的 1 的方案不满足要求; 这些方案由于必须同时选取首尾的 1, 因此就相当于从中间的 $n-2$ 对数中选取 $k-2$ 个不相邻的数, 方案数为

$$\binom{(2n-4)-(k-2)+1}{k-2} = \binom{2n-k-1}{k-2}$$

将以上两式相减, 可得

$$r_k = \binom{2n-k+1}{k} - \binom{2n-k-1}{k-2} = \frac{2n}{2n-k} \binom{2n-k}{k}$$

将 r_k 代入式 (6-7) 中, 即可得到女性位置固定的情况下男性的就座方案数:

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

上式再乘上女性就座的方案数 $2 \cdot n!$ ，即得到原问题所求的方案数：

$$2 \cdot n! \cdot M_n = 2 \cdot n! \cdot \sum_{k=0}^n (-1)^k \frac{2n}{2n-k} \binom{2n-k}{k} (n-k)! \quad (6-10)$$

6.3 带约束的组合问题

与排列问题类似，带有额外约束的组合问题也可用容斥原理解决。本节中我们讨论几个典型的例子。

例 6.3.1. 考虑如图 6.5 所示的格路模型，其中 AB 、 CD 、 EF 、 GH 四条线段禁止通行。求起点为 $(0,0)$ 、终点为 $(10,5)$ 、每一步均向右或向上走，且不经上述四条线段的合法路径数目。

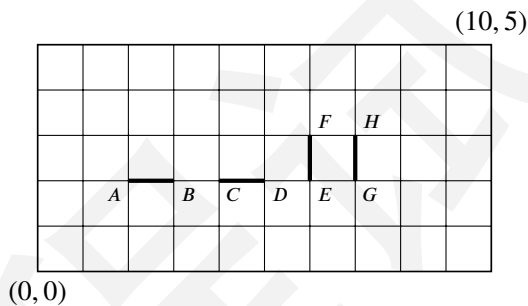


图 6.5: 例 6.3.1 中的格路示意图

这个例子中，禁止通行的四条线段的各端点坐标分别为

$$\begin{aligned} &A(2,2), B(3,2), \\ &C(4,2), D(5,2), \\ &E(6,2), F(6,3), \\ &G(7,2), H(7,3) \end{aligned}$$

直接求解不经过 AB 、 CD 、 EF 、 GH 四条线段的合法路径数目较为困难；但是，若反过来要求路径必须经过 AB ，则整条路径可分解为由起点至 A 和 B 至终点的两条子路径，方案数等于两条子路径的方案之积，容易计算。这提示我们使用容斥原理间接求解。

令 S 表示所有从起点至终点的路径构成的集合，则显然有

$$|S| = \binom{15}{5}$$

令 A_1, A_2, A_3, A_4 分别表示经过 AB 、 CD 、 EF 、 GH 的路径构成的集合。我们有

- 单个集合的大小:

$$|A_1| = \binom{4}{2} \binom{10}{3}, \quad |A_2| = \binom{6}{2} \binom{8}{3},$$

$$|A_3| = \binom{8}{2} \binom{6}{2}, \quad |A_4| = \binom{9}{2} \binom{5}{2}$$

- 两个集合交集的大小:

$$|A_1 \cap A_2| = \binom{4}{2} \binom{8}{3}, \quad |A_1 \cap A_3| = \binom{4}{2} \binom{6}{2},$$

$$|A_1 \cap A_4| = \binom{4}{2} \binom{5}{2}, \quad |A_2 \cap A_3| = \binom{6}{2} \binom{6}{2},$$

$$|A_2 \cap A_4| = \binom{6}{2} \binom{5}{2}, \quad |A_3 \cap A_4| = 0$$

- 三个集合交集的大小:

$$|A_1 \cap A_2 \cap A_3| = \binom{4}{2} \binom{6}{2}, \quad |A_1 \cap A_2 \cap A_4| = \binom{4}{2} \binom{5}{2},$$

$$|A_1 \cap A_3 \cap A_4| = 0, \quad |A_2 \cap A_3 \cap A_4| = 0$$

- 四个集合交集的大小:

$$|A_1 \cap A_2 \cap A_3 \cap A_4| = 0$$

利用容斥原理可知, 合法路径的数目为

$$|\overline{A_1} \cap \overline{A_2} \cap \overline{A_3} \cap \overline{A_4}| = 2049$$

例 6.3.2. 将 n 个不同的小球放入 k 个相同的盒子, 禁止出现空盒, 求方案数.

这个例子中的球盒模型实际上就对应于 5.3 节中介绍的第二类 Stirling 数 $S(n, k)$. 下面我们讨论如何使用容斥原理求解此球盒模型. 仍然先考虑 m 个盒子互不相同的情况, 此时问题的解为

$$S^*(n, k) = k! \cdot S(n, k)$$

令 S 表示所有可能的放球方案构成的集合, 则显然有

$$|S| = m^n$$

令 A_i 表示第 i 个盒子为空的放球方案构成的集合. 考虑这些放球方案: 由于一个盒子必须为空, 因此这些方案相当于将 n 个球放入其余的 $m-1$ 个盒子中, 从而我们有

$$|A_i| = (m-1)^n \quad (1 \leq i \leq m)$$

类似地, $A_i \cap A_j$ ($i \neq j$) 中的放球方案要求两个盒子必须为空, 实际上相当于将 n 个球放入 $m-2$ 个盒子中, 从而我们有

$$|A_i \cap A_j| = (m-2)^n \quad (1 \leq i < j \leq n)$$

以此类推, A_1, A_2, \dots, A_n 中任何 k 个不同集合的交集大小均为 $(m-k)^n$. 利用容斥原理, 可得

$$\begin{aligned} S^*(n, k) &= |\overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_k}| \\ &= |S| - \sum_i |A_i| + \sum_{i < j} |A_i \cap A_j| - \dots + (-1)^n |A_1 \cap A_2 \cap \dots \cap A_m| \\ &= k^n - \binom{k}{1}(k-1)^n + \binom{k}{2}(k-2)^n - \dots + (-1)^k \binom{k}{k}(k-k)^n \\ &= \sum_{j=0}^k (-1)^j \binom{k}{j} (k-j)^n \end{aligned}$$

这表明

$$S(n, k) = \frac{1}{k!} \sum_{j=0}^k (-1)^j \binom{k}{j} (k-j)^n$$

上式与 5.3 节中给出的第二类 Stirling 数的通项表达式相同.

例 6.3.3. 求方程

$$x_1 + x_2 + x_3 = 16 \quad (0 \leq x_1 \leq 5, 2 \leq x_2 \leq 8, -1 \leq x_3 \leq 6)$$

的整数解数目.

在具体求解这个例子中的问题前, 我们先简单讨论一般情况下的方程

$$x_1 + x_2 + \dots + x_n = M$$

的整数解数目的问题, 其中 x_1, x_2, \dots, x_n 中的每个变量均在一个连续的范围内取值. 如果所有的变量取值范围均为 $\mathbb{Z}^* = [0, +\infty)$, 即非负整数集, 则上述方程就相当于 1.6.1 节中介绍的可重组问题. 否则, 我们可讨论 x_i ($i = 1, 2, \dots, n$) 的取值范围:

(1) 若 x_i 取值范围有下界、无上界, 即 $x_i \in [a_i, +\infty)$, 则可作换元

$$x'_i = x_i - a_i$$

由此得到的变量 x'_i 的取值范围就变为非负整数集, 转化为我们在介绍可重组时讨论过的情况.

- (2) 若 x_i 取值范围无下界、有上界, 即 $x_i \in (-\infty, b_i]$, 则这时其他所有变量均必须同样有上界, 否则方程组将有无穷多组整数解. 这时, 我们可对所有变量同时作换元

$$x'_i = -x_i$$

得到

$$x'_1 + x'_2 + \cdots + x'_n = -M$$

从而将方程组中的变量转化为均有下界的情况.

- (3) 若 x_i 取值范围有下界、有上界, 即 $x_i \in [a_i, b_i]$, 则仍然可作换元

$$x'_i = x_i - a_i$$

由此得到的变量 x'_i 的取值为非负整数, 且有一正的上界 $b_i - a_i$. 注意到, 若 x'_i 不满足它的取值范围约束, 则应当有

$$x'_i \in [b_i - a_i + 1, +\infty)$$

这时其取值范围变为我们上面讨论的有下界、无上界的情况, 可以通过再次换元使取值范围变为非负整数集, 转化为可重组合问题. 上述讨论表明, x_i 不在取值范围内的解是容易计数的, 这提示我们使用容斥原理间接地求解问题.

下面我们求解这个例子中的方程. 根据上面的讨论, 先作换元

$$x'_2 = x_2 - 2, \quad x'_3 = x_3 + 1$$

得到方程

$$x_1 + x'_2 + x'_3 = 15 \quad (0 \leq x_1 \leq 5, 0 \leq x'_2 \leq 6, 0 \leq x'_3 \leq 7)$$

此方程的整数解与原方程的整数解一一对应. 令全集 U 表示不考虑各变量的取值约束时上述方程的全体非负整数解构成的集合, A 表示此方程的全体整数解构成的集合; A_1, A_2, A_3 分别代表变量 x_1, x'_2, x'_3 不满足各自的取值约束时的非负整数解构成的集合, 即

$$A_1 = \{x_1 + x_2 + x_3 = 15 \mid x_1, x'_2, x'_3 \in \mathbb{Z}^*, x_1 \geq 6\}$$

$$A_2 = \{x_1 + x'_2 + x'_3 = 15 \mid x_1, x'_2, x'_3 \in \mathbb{Z}^*, x'_2 \geq 7\}$$

$$A_3 = \{x_1 + x'_2 + x'_3 = 15 \mid x_1, x'_2, x'_3 \in \mathbb{Z}^*, x'_3 \geq 8\}$$

则我们有

$$A = \overline{A_1} \cap \overline{A_2} \cap \overline{A_3}$$

考虑集合 A_1 , 其中的解均有 $x_1 \geq 6$, 于是可以作换元

$$y_1 = x_1 - 6$$

从而将方程化为

$$y_1 + x'_2 + x'_3 = 15 - 6 = 9$$

此方程的非负整数解数目为

$$|A_1| = \binom{9+3-1}{9} = \binom{11}{2}$$

同理, 我们有

$$|A_2| = \binom{10}{2}, |A_3| = \binom{9}{2}$$

A_1, A_2, A_3 中两个集合的交集代表 x_1, x_2, x_3 中的某两个变量同时不满足其上界约束, 因此可同时对这两个变量作换元, 然后继续使用上述思路求解. 由此, 我们有

$$|A_1 \cap A_2| = \binom{4}{2}, |A_2 \cap A_3| = \binom{3}{1}, |A_1 \cap A_3| = 1$$

最后, 我们还有

$$|A_1 \cap A_2 \cap A_3| = 0$$

这是因为若三个变量同时不满足其上界约束, 则三者之和至少为 18, 大于 15, 从而原方程不可能有非负整数解. 综上, 利用容斥原理, 我们有

$$\begin{aligned} |A| &= |\overline{A_1} \cap \overline{A_2} \cap \overline{A_3}| \\ &= \binom{17}{2} - \left[\binom{11}{2} + \binom{10}{2} + \binom{9}{2} \right] + \left[\binom{4}{2} + \binom{3}{1} + 1 \right] \\ &= 10 \end{aligned}$$

因此, 原方程一共有 10 组满足约束的整数解.

实际上, 这个例子还有另一种更加方便的解法. 重新观察换元后得到的方程:

$$x_1 + x_2 + x_3 = 15 \quad (0 \leq x_1 \leq 5, 0 \leq x_2 \leq 6, 0 \leq x_3 \leq 7)$$

注意到, 各变量的取值范围约束条件非常严格; 即使 x_1, x_2, x_3 均取上界值, 三者之和也仅有 18, 与方程右侧的 15 仅相差 3, 这表明各变量的取值都应该与上界值相差不多. 为此, 我们可作换元

$$\xi_1 = 5 - x_1, \xi_2 = 6 - x_2, \xi_3 = 7 - x_3$$

则原方程变为

$$\xi_1 + \xi_2 + \xi_3 = 18 - (x_1 + x_2 + x_3) = 3$$

原问题等价于求该方程的非负整数解个数, 结果为 $\binom{3+3-1}{3} = 10$. 要注意, 这种换元法不具备通用性, 仅在变量取值范围约束较为严格时才有助于简化问题. 例如, 考虑如下方程:

$$x_1 + x_2 + x_3 = 15 \quad (0 \leq x_1, x_2, x_3 \leq 10)$$

则按上述方法进行换元不会对求解方程有任何帮助.

6.4 广义容斥原理

本节中, 我们讨论容斥原理的一个简单推广. 给定全集 U 和一系列集合 A_1, A_2, \dots, A_n , 容斥原理可用于计算不属于其中任何一个集合的元素数目, 即恰好属于 A_1, A_2, \dots, A_n 中 0 个集合的元素数目. 通过本节所要讨论的推广形式, 我们可计算恰好属于 A_1, A_2, \dots, A_n 中 k 个集合的元素数目 ($k = 0, 1, 2, \dots, n$).

形式化地, 给定参数 k , 我们希望求出

$$Q_k \triangleq \sum_{I \subseteq \{1, 2, \dots, n\}, |I|=k} \left| \bigcap_{i \in I} A_i \cap \bigcap_{i \notin I} \overline{A_i} \right| \quad (6-11)$$

等式右侧的求和记号对指标集 $\{1, 2, \dots, n\}$ 的所有大小为 k 的子集 I 求和, 被求和的对象则是恰好属于 A_i ($i \in I$) 这 k 个集合的元素. 我们给出如下定理:

定理 6.7 (广义的容斥原理). 设 U 是有限的全集, A_1, A_2, \dots, A_n 是一系列集合, 则恰好属于其中 k 个集合的元素数目为

$$Q_k = \sum_{I \subseteq \{1, 2, \dots, n\}} (-1)^{|I|-k} \binom{|I|}{k} \left| \bigcap_{i \in I} A_i \right| \quad (6-12)$$

证明. 在等式左侧, 所有恰好属于 k 个集合的元素都被统计 1 次, 而其他元素被统计 0 次. 在等式右侧, 对任意元素 $x \in U$, 可以根据 A_1, A_2, \dots, A_n 中包含 x 的集合的数目, 分成三类讨论:

(1) 不足 k 个集合包含 x : 这意味着若

$$x \in \bigcap_{i \in I} A_i$$

则必然有 $|I| < k$, 从而有 $\binom{|I|}{k} = 0$, 因此 x 被统计的次数也必定为 0 次;

(2) 恰好 k 个集合包含 x : x 被统计的次数为

$$(-1)^{k-k} \binom{k}{k} = 1$$

(3) 多于 k 个集合包含 x : 设包含 x 的集合数目为 $k+j$ 个 ($j > 0$), 则当 $|I| = k+i$ (其中 $i = 0, 1, 2, \dots, j$) 时, x 总共被统计的次数为 $\binom{k+j}{k+i}$. 因此, x 在等号右侧被统计

的总次数为

$$\begin{aligned}\sum_{i=0}^j (-1)^i \binom{k+i}{k} \binom{k+j}{k+i} &= \sum_{i=0}^j (-1)^i \binom{k+j}{k} \binom{j}{i} \\ &= \binom{k+j}{k} \sum_{i=0}^j (-1)^i \binom{j}{i} \\ &= \binom{k+j}{k} \cdot 0 = 0\end{aligned}$$

综上, 在等式右侧, 只有恰好属于 A_1, A_2, \dots, A_n 中 k 个集合的元素恰好被统计 1 次, 与等号左侧相同. 因此, 式 (6-4) 恒成立. \square

将 $k=0$ 代入到式 (6-12) 中即得到

$$Q_0 = \sum_{I \subseteq \{1, 2, \dots, n\}} (-1)^{|I|} \left| \bigcap_{i \in I} A_i \right|$$

这即是容斥原理. 因此, 之前讨论的容斥原理可看做上述定理的一个特殊情形.

例 6.4.1. 假设一个学校只有数学、物理和化学三门课程. 已知选修这三门课程的学生分别有 170、130 和 120 人; 同时修数学、物理的学生有 45 人, 同时修数学、化学的学生有 20 人, 同时修物理、化学的学生有 22 人; 同时修全部三门课程的学生有 3 人. 求恰好上一门课和两门课的学生数目.

至少上一门课的学生有 $170+130+120=420$ 人次, 至少上两门课的学生有 $45+20+22=87$ 人次, 上三门课的学生有 3 人次. 由此我们有

$$Q_1 = 420 - 2 \times 87 + 3 \times 3 = 255$$

$$Q_2 = 87 - 3 \times 3 = 78$$

因此, 恰好上一门课的学生有 255 名, 恰好上两门课的学生有 78 名. 上课的学生总共有 $255 + 78 + 3 = 336$ 名, 这与例 6.1.3 给出的结果一致.

6.5 Möbius 反演

Möbius 反演是一种重要的数论工具, 在组合数学领域中可用于解决一大类复杂的计数问题. 该定理可表述如下:

定理 6.8 (Möbius 反演定理). 设 $f: \mathbb{Z}^+ \mapsto \mathbb{R}$ 是定义在正整数集上的实值函数 (这样的函数称为数论函数); $g: \mathbb{Z}^+ \mapsto \mathbb{R}$ 是依赖于 f 的函数, 定义为

$$g(n) = \sum_{d|n} f(d) \quad (6-13)$$

则从 g 可反推出 f :

$$f(n) = \sum_{d|n} \mu\left(\frac{n}{d}\right) g(d) = \sum_{d|n} \mu(d) g\left(\frac{n}{d}\right) \quad (6-14)$$

其中 $\mu(n)$ 称为 **Möbius 函数**, 定义为

$$\mu(n) = \begin{cases} 1 & (n = 1) \\ (-1)^k & (n \text{ 是 } k \text{ 个互异质数的积}) \\ 0 & (\text{其他情况}) \end{cases} \quad (6-15)$$

Möbius 反演与本章前面介绍的容斥原理有着密切的联系, 它可看作容斥原理的一个推广, 尽管二者在形式上有较大的差异. 下面我们从容斥原理出发, 逐步证明 Möbius 反演定理. 证明的基本思路是先将容斥原理推广到实值函数上和多重集上, 然后利用质因子分解将多重集转化为正整数.

首先回顾 6.1 节中证明容斥原理时采用的思路. 容斥原理的表达式是

$$\left| \overline{A_1} \cap \overline{A_2} \cap \cdots \cap \overline{A_n} \right| = \sum_{I \subseteq \{1, 2, \dots, n\}} (-1)^{|I|} \left| \bigcap_{i \in I} A_i \right|$$

我们证明了全集 U 中的每个元素在上式左右两侧被统计的次数均相同; 若元素 x 不属于 A_1, A_2, \dots, A_n 中的任何一个集合, 则其被统计的次数均为 1, 否则次数均为 0. 由此, 我们即能证明上式左右两侧相等.

上述证明思路的重点在于全集 U 中的每个元素对等式两侧的贡献相等. 容斥原理中讨论的是集合中元素的数目, 因此我们在证明容斥原理时, 规定任意元素 x 每被统计一次则产生 1 的贡献. 容易发现, x 的贡献值实际上可以为任意实数, 这对 x 在等式两侧的贡献的相等性没有任何影响. 由此, 我们可将容斥原理推广到定义在 U 上的实值函数 f 上:

$$f(x) : U \mapsto \mathbb{R}$$

由此, 容斥原理可以推广为如下形式:

推论 6.9. 设 U 是有限的全集, A_1, A_2, \dots, A_n 是一系列集合, $f(x) : U \mapsto \mathbb{R}$ 是定义在 U 上的实值函数, 则

$$\sum_{x \in \overline{A_1} \cap \overline{A_2} \cap \cdots \cap \overline{A_n}} f(x) = \sum_{I \subseteq \{1, 2, \dots, n\}} (-1)^{|I|} \sum_{x \in \bigcap_{i \in I} A_i} f(x) \quad (6-16)$$

证明. 注意到 U 中的任意元素 x 在式 (6-16) 左右两侧产生的贡献均相同: 若 x 不属于 A_1, A_2, \dots, A_n 中任何一个集合, 则贡献为 $f(x)$, 否则贡献为 0. 因此, 式 (6-16) 恒成立. \square

显然, 如果令 $f(x) \equiv 1$, 则对任意集合 A 有

$$\sum_{x \in A} f(x) = \sum_{x \in A} 1 = |A|$$

上述推论也就退化为普通的容斥原理.

下面考虑有限集

$$X = \{x_1, x_2, \dots, x_n\}$$

我们基于 X 构建一个容斥原理的场景. 令全集 U 为 X 的所有子集构成的集合 $P(X)$ (称为 X 的幂集). 令 A_i ($i = 1, 2, \dots, n$) 表示 X 的全体不包含元素 x_i 的子集所构成的集合. 在这一场景下, 我们重新考察权值函数 f 以及式 (6-16):

- f 成为一个将 X 的子集映射到实数的函数:

$$f(K) : P(X) \mapsto \mathbb{R}$$

- 在式 (6-16) 左侧, 根据 A_i 的定义, 满足求和条件

$$x \in \overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_n}$$

的 x 应当是一个包含 X 中每个元素的 X 的子集. 这样的子集只有一个, 即 $x = X$, 因此等式左侧即为 $f(X)$;

- 在式 (6-16) 右侧, 根据定义, $\bigcap_{i \in I} A_i$ 是由所有不包含 $\{x_i \mid i \in I\}$ 中任何一个元素的 Y 的子集构成的集合; 容易看出, 它实际上就是 $\{x_i \mid i \notin I\}$ 的幂集. 定义

$$g(K) = \sum_{A \subseteq K} f(A) \quad (6-17)$$

结合上一条结论, 式 (6-16) 在此场景下可以写成

$$f(X) = \sum_{K \subseteq X} (-1)^{|K|} g(\overline{K}) = \sum_{K \subseteq X} (-1)^{|X|-|K|} g(K) \quad (6-18)$$

观察式 (6-17) 与 (6-18), 二者揭示了函数 f 和 g 之间的一项一般性的关系. 我们将其总结为下列定理:

定理 6.10. 设 K 是有限集 X 的子集, $f(K) : P(X) \mapsto \mathbb{R}$ 是定义在 X 的幂集上的实值函数, $g(K) : P(X) \mapsto \mathbb{R}$ 是依赖于 f 的函数, 定义为

$$g(K) = \sum_{I \subseteq K} f(I) \quad (6-19)$$

则从 g 可反推出 f :

$$f(K) = \sum_{I \subseteq K} (-1)^{|K|-|I|} g(I) \quad (6-20)$$

例 6.5.1. 定理 6.10 可用于推出容斥原理. 具体地, 设 U 是全集, A_1, A_2, \dots, A_n 是一系列集合. 定义

$$f(K) = \left| \bigcap_{i \notin K} A_i \cap \bigcap_{i \in K} \overline{A_i} \right|$$

即 $f(K)$ 统计了恰好同时属于 A_i ($i \notin K$) 这 $n - |K|$ 个集合的元素数目. 根据定义, $g(K)$ 相当于统计了同时属于 A_i ($i \notin K$) 这些集合 (是否属于其他集合则无所谓) 的元素数目:

$$g(K) = \left| \bigcap_{i \notin K} A_i \right|$$

将 $K = \{1, 2, \dots, n\}$ 和以上两式代入式 (6-20) 中, 得到

$$\begin{aligned} \left| \bigcap_{i=1}^n \overline{A_i} \right| &= \sum_{I \subseteq \{1, 2, \dots, n\}} (-1)^{n-|I|} \left| \bigcap_{i \notin I} A_i \right| \\ &= \sum_{I \subseteq \{1, 2, \dots, n\}} (-1)^{|I|} \left| \bigcap_{i \in I} A_i \right| \end{aligned}$$

这即为容斥原理的表达式.

下面我们将定理 6.10 推广到多重集上. 考虑多重集

$$X = \{a_1 \cdot x_1, a_2 \cdot x_2, \dots, a_n \cdot x_n\} \quad (a_1, a_2, \dots, a_n \geq 1)$$

它的子多重集将具有如下形式:

$$K = \{b_1 \cdot x_1, b_2 \cdot x_2, \dots, b_n \cdot x_n\} \subseteq X \quad (0 \leq b_i \leq a_i) \quad (6-21)$$

X 的所有子多重集共同构成 X 的幂集 $P(X)$; 注意 $P(X)$ 不再是多重集, 而是一个普通的集合, 它的元素是互不相同的多重集.

我们仿照之前的做法构建一个 X 是多重集时的容斥原理场景. 令全集 U 为 $P(X)$. 令 A_i ($i = 1, 2, \dots, n$) 表示 X 的全体包含不足 a_i 个元素 x_i 的子多重集所构成的集合. 重新考察权值函数 f 以及式 (6-16), 可得与之前相似的结论:

- f 成为一个将 X 的子多重集映射到实数的函数:

$$f(K) : P(X) \mapsto \mathbb{R}$$

- 在式 (6-16) 左侧, 根据 A_i 的定义, 满足求和条件

$$x \in \overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_n}$$

的 x 应当是一个包含 a_1 个 x_1 、 a_2 个 x_2 、……、 a_n 个 x_n 的 X 的子多重集. 这样的子多重集只有一个, 即 $x = X$, 因此等式左侧即为 $f(X)$;

- 在式 (6-16) 右侧, 与 X 是普通集合时的结论类似, $\bigcap_{i \in I} A_i$ 实际上是

$$\bigcup_{i \in I} \{(a_i - 1) \cdot x_i\} \cup \bigcup_{i \notin I} \{a_i \cdot x_i\}$$

的幂集. 仍然定义

$$g(K) = \sum_{A \subseteq K} f(A) \quad (6-22)$$

结合上一条结论, 式 (6-16) 在此场景下可以写成

$$\begin{aligned} f(X) &= \sum_{I \subseteq \{1, 2, \dots, n\}} (-1)^{|I|} g\left(\bigcup_{i \in I} \{(a_i - 1) \cdot x_i\} \cup \bigcup_{i \notin I} \{a_i \cdot x_i\}\right) \\ &= \sum_{I \subseteq \{1, 2, \dots, n\}} (-1)^{n-|I|} g\left(\bigcup_{i \in I} \{a_i \cdot x_i\} \cup \bigcup_{i \notin I} \{(a_i - 1) \cdot x_i\}\right) \end{aligned} \quad (6-23)$$

式 (6-23) 过于复杂, 并且等式右侧是对 $\{1, 2, \dots, n\}$ 的所有子集求和, 而并非如式 (6-18) 那样对 X 的所有子多重集求和. 为了化简该式并在形式上取得统一, 我们引入定义在 X 的幂集上的函数 $\mu(K) : P(X) \mapsto \{-1, 0, 1\}$, 其中 K 的定义同式 (6-21):

$$\mu(K) = \begin{cases} (-1)^{|K|} & (0 \leq b_1, b_2, \dots, b_n \leq 1) \\ 0 & (\text{其他情况}) \end{cases} \quad (6-24)$$

即, $\mu(K)$ 仅在 K 为普通集合时取非零值. 利用函数 μ , 式 (6-23) 可以改写为对 X 的所有子集求和的形式:

$$f(X) = \sum_{K \subseteq X} \mu(K) g(X - K) = \sum_{K \subseteq X} \mu(X - K) g(K) \quad (6-25)$$

最后我们将多重集转化为正整数. 正整数的质因子分解与多重集之间具有良好的一一对应关系. 我们取 n 个不同的质数 p_1, p_2, \dots, p_n , 然后按以下方式将多重集映射为正整数:

$$\{a_1 \cdot x_1, a_2 \cdot x_2, \dots, a_n \cdot x_n\} \mapsto p_1^{a_1} p_2^{a_2} \dots p_n^{a_n}$$

由于任何正整数都有唯一的质因子分解方式, 上述映射显然是可逆的.

经由上述映射, 多重集 X 转化为一个正整数

$$N = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n}$$

X 的任意一个子集 K 均以一一对应的形式被映射为 N 的约数 D :

$$K = \{b_1 \cdot x_1, b_2 \cdot x_2, \dots, b_n \cdot x_n\} \subseteq \{a_1 \cdot x_1, a_2 \cdot x_2, \dots, a_n \cdot x_n\} = X$$

\Downarrow

$$D = p_1^{b_1} p_2^{b_2} \dots p_n^{b_n} \mid p_1^{a_1} p_2^{a_2} \dots p_n^{a_n} = N$$

函数 $\mu(K)$ 则成为一个定义在 N 的全体约数构成的集合上的函数:

$$\mu(n) = \begin{cases} 1 & (n = 1) \\ (-1)^k & (n \text{ 是 } k \text{ 个互异质数的积}) \\ 0 & (\text{其他情况}) \end{cases}$$

此即为 Möbius 函数. 式 (6-22) 和 (6-25) 则分别变为如下形式:

$$g(M) = \sum_{D|M} f(D)$$

$$f(M) = \sum_{D|M} \mu(D)g\left(\frac{M}{D}\right) = \sum_{D|M} \mu\left(\frac{M}{D}\right)g(D)$$

将上面两式中的变量替换为更加常用的小写字母, 我们得到如下结论:

定理 6.11. 设 n 是正整数, $f(m), g(m)$ 是定义在 n 的全体约数构成的集合上的实值函数, 其中 g 是依赖于 f 的函数, 定义为

$$g(m) = \sum_{d|m} f(d)$$

则从 g 可反推出 f :

$$f(m) = \sum_{d|m} \mu(d)g\left(\frac{m}{d}\right) = \sum_{d|m} \mu\left(\frac{m}{d}\right)g(d)$$

注意到函数 f, g, μ 的形式实际上均与 N 的取值无关; 通过让 N 取遍所有的正整数, 函数 f, g, μ 的定义域可扩展为正整数集, 由此得到的即为 Möbius 反演定理. 综上所述, Möbius 反演可看作容斥原理推广到实值函数和多重集上的形式, 但是其多重集的本质被隐藏在正整数的质因数分解之下.

下面我们讨论 Möbius 函数 μ 的性质, 以及 Möbius 反演的一些应用.

推论 6.12. 设 μ 是 Möbius 函数, 则

$$\sum_{d|n} \mu(d) = \begin{cases} 1 & (n=1) \\ 0 & (n>1) \end{cases}$$

证明. 若 $n=1$, 则

$$\sum_{d|n} \mu(d) = \mu(1) = 1$$

若 $n>1$, 设 $n = p_1^{a_1} p_2^{a_2} \cdots p_m^{a_m}$, 令 $n' = p_1 p_2 \cdots p_m$, 则

$$\begin{aligned} \sum_{d|n} \mu(d) &= \sum_{d|n'} \mu(d) = \sum_{k=0}^m \sum_{I \subseteq \{1,2,\dots,n\}, |I|=k} \mu\left(\prod_{i \in I} p_i\right) \\ &= \sum_{k=0}^m \binom{m}{k} (-1)^k \\ &= (1-1)^m = 0 \end{aligned}$$

□

推论 6.13.

$$\mu(n) = \begin{cases} 1 & (n = 1) \\ - \sum_{d|n, d \neq 1} \mu(d) & (n > 1) \end{cases} \quad (6-26)$$

例 6.5.2. 欧拉函数 (Euler Phi Function) $\varphi(n)$ 定义为小于等于 n 且与 n 互质的正整数的数目, 求 $\varphi(n)$.

我们先展示如何使用经典的容斥原理计算 $\varphi(n)$. 令 n 的所有质因子分别为 p_1, p_2, \dots, p_m , 令 A_i 表示所有小于等于 n 且能被 p_i 整除的正整数构成的集合. 由容斥原理, 我们有

$$\begin{aligned} \varphi(n) &= |\overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_m}| \\ &= \sum_{I \subseteq \{1, 2, \dots, m\}} (-1)^{|I|} \frac{n}{\prod_{i \in I} p_i} \\ &= n - \left(\frac{n}{p_1} + \frac{n}{p_2} + \dots + \frac{n}{p_m} \right) + \dots + (-1)^m \frac{n}{p_1 p_2 \dots p_m} \\ &= n \prod_{i=1}^m \left(1 - \frac{1}{p_i} \right) \end{aligned}$$

下面我们使用 Möbius 反演重新求解 $\varphi(n)$. 令

$$f(n) = \varphi(n), \quad g(n) = \sum_{d|n} \varphi(d) = \sum_{d|n} \varphi\left(\frac{n}{d}\right)$$

$\varphi\left(\frac{n}{d}\right)$ 是小于等于 $\frac{n}{d}$ 的正整数中与 $\frac{n}{d}$ 互质的数的数目, 因此即相当于小于等于 n 的正整数中与 n 的最大公约数为 d 的数的数目. 由于任何正整数与 n 的最大公约数一定是 n 的约数, 因此 $g(n)$ 中对所有满足 $d | n$ 的 $\varphi\left(\frac{n}{d}\right)$ 求和, 实际上不重不漏地统计了所有小于等于 n 的正整数, 即 $g(n) = n$. 应用 Möbius 反演可得

$$\varphi(n) = \sum_{d|n} \mu\left(\frac{n}{d}\right) \cdot d = \sum_{d|n} \mu(d) \cdot \frac{n}{d}$$

由于 $\mu(d) \neq 0$ 当且仅当 $d = 1$ 或 d 是互异质数的乘积, 在等式右侧仅保留使 $\mu(d) \neq 0$ 的项, 则上式又可以写成

$$\begin{aligned} \varphi(n) &= n - \left(\frac{n}{p_1} + \frac{n}{p_2} + \dots + \frac{n}{p_m} \right) + \dots + (-1)^m \frac{n}{p_1 p_2 \dots p_m} \\ &= n \prod_{i=1}^m \left(1 - \frac{1}{p_i} \right) \end{aligned}$$

结果与用容斥原理求得的相同.

我们最后讨论使用 Möbius 反演解决可重圆排列问题. 此问题的形式为:

从多重集

$$\{\infty \cdot x_1, \infty \cdot x_2, \dots, \infty \cdot x_m\}$$

中取 n 个元素作圆排列, 求不同的排列方案数 $T(n)$.

1.5.2 节已经讨论过从集合中取元素作圆排列 (即元素不可重复) 的问题, 那时的结论是: 在 n 个元素中取 r 个作圆排列, 方案数是 $\frac{P(n, r)}{r}$. 但是, 集合圆排列的思路不适用于元素可重复的情况.

为了解决可重圆排列问题, 我们首先引入周期的概念. 如果一个圆排列可由一个长度为 d 的线排列重复若干次, 然后首尾拼接形成, 则其中最小的 d 称为此圆排列的周期. 记长度为 n 、周期为 d 的全体圆排列的数目为 $M(d)$. 显然, 若要使 $M(d) \neq 0$, 则 d 必须是 n 的因子.

现在设想将圆排列从某一位置切开, 展开为线排列. 我们从两个角度考察不同的线排列方案数. 一方面, 得到的线排列就相当于从多重集

$$\{\infty \cdot x_1, \infty \cdot x_2, \dots, \infty \cdot x_m\}$$

中直接任取 n 个元素作排列, 总方案数为 m^n . 另一方面, 每个周期为 d 的圆排列都有 d 个本质不同的切开位置, 从这些位置展开后可得到 d 个不同的线排列. 因此,

$$m^n = \sum_{d|n} M(d) \cdot d \quad (6-27)$$

我们令

$$f(n) = M(n) \cdot n, \quad g(n) = m^n$$

此时式 (6-27) 即成为 $g(n) = \sum_{d|n} f(d)$ 的形式, 我们可对其作 Möbius 反演:

$$M(n) \cdot n = \sum_{d|n} \mu\left(\frac{n}{d}\right) m^d$$

从而可得函数 M 的表达式:

$$M(n) = \frac{1}{n} \sum_{d|n} \mu\left(\frac{n}{d}\right) m^d$$

对所有 $M(d)$ 求和, 即得到可重圆排列的方案总数:

$$\begin{aligned}
 T(n) &= \sum_{d|n} M(d) \\
 &= \sum_{d|n} \frac{1}{d} \sum_{k|d} \mu\left(\frac{d}{k}\right) m^k \\
 &= \sum_{k|n} \left[\sum_{l|(n/k)} \mu(l) \cdot \frac{1}{kl} \right] m^k \quad (\text{令 } d = kl) \\
 &= \sum_{k|n} \left[\sum_{l|(n/k)} \mu\left(\frac{n/k}{l}\right) \cdot \frac{l}{n} \right] m^k \quad (\text{将 } l \text{ 代换为 } \frac{n}{kl}) \\
 &= \sum_{k|n} \frac{\varphi(n/k)}{n} m^k \\
 &= \frac{1}{n} \sum_{k|n} \varphi\left(\frac{n}{k}\right) m^k
 \end{aligned}$$

综上所述, 从 m 种不同的元素中允许重复地取 n 个作圆排列, 不同的排列方案数为

$$T(n) = \frac{1}{n} \sum_{k|n} \varphi\left(\frac{n}{k}\right) m^k \quad (6-28)$$

我们将在第七章通过另一种方式重新推导出这一结论.

例 6.5.3. 从 3 种元素中选出 4 个均匀排列在一个圆的四周. 不同的圆排列的数目为

$$T(4) = \frac{1}{4} [\varphi(4) \cdot 3^1 + \varphi(2) \cdot 3^2 + \varphi(1) \cdot 3^4] = \frac{1}{4} (6 + 9 + 81) = 24$$

可能的周期一共有 1, 2, 4 三种, 每种周期的圆排列的数目分别为

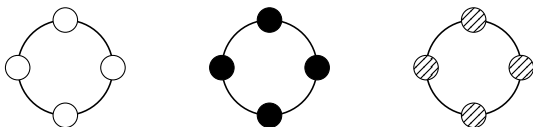
$$M(1) = 3^1 = 3$$

$$M(2) = \frac{1}{2} [\mu(2) \cdot 3^1 + \mu(1) \cdot 3^2] = 3$$

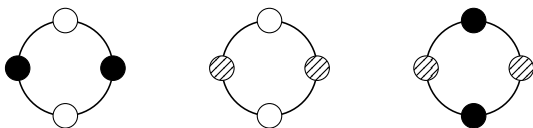
$$M(4) = \frac{1}{4} [\mu(4) \cdot 3^1 + \mu(2) \cdot 3^2 + \mu(1) \cdot 3^4] = \frac{1}{4} (3^4 - 3^2) = 18$$

最后, 我们按照周期分类列出所有 24 种不同的圆排列. 我们使用白色、黑色和阴影分别代表三种类型的元素:

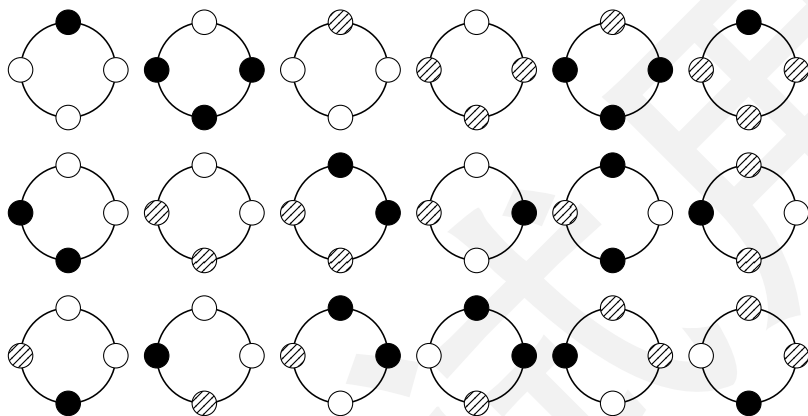
- 周期为 1:



- 周期为 2:



- 周期为 4:



习题

6.1. 解释以下恒等式的组合意义:

$$(1) \binom{n-m}{n-k} = \sum_{j=0}^m \binom{m}{j} \binom{n-j}{k}$$

$$(2) \binom{n-1}{m-1} = \sum_{j=0}^m (-1)^j \binom{m}{j} \binom{n+m-j-1}{n}$$

$$(3) \binom{n+m-1}{n-1} = \sum_{j=0}^m (-1)^j \binom{n+m}{n+j}$$

6.2. 在由 $a, a, a, b, b, b, c, c, c$ 这 9 个元素的全排列中, 分别求满足下列要求的排列数:

- (1) 任何连续 3 个元素不能全相同;
- (2) 任何相邻 2 个元素不能相同.

6.3. 求方程

$$x_1 + x_2 + x_3 = 40 \quad (6 \leq x_1 \leq 15, 5 \leq x_2 \leq 20, 10 \leq x_3 \leq 25)$$

的整数解数目.

第七章 Pólya 计数理论

对许多组合学问题而言，问题的一个解可通过对称操作变换为问题的另一个解。例如，考虑第一章讨论过的环形排列问题：设想有一个将 n 个元素等距排列在一个圆的圆周上的排列方案，则无论是将这个圆绕圆心旋转 $\frac{k}{n} \cdot 360^\circ$ ($k = 1, 2, \dots, n-1$)，或是绕圆的一条直径翻转 180° ，又或是既作旋转又作翻转，均能重新得到一个 n 个元素在圆周上的排列方案。这里的旋转或翻转即称为对称操作。不同的问题可允许不同的对称操作，例如圆排列问题中仅允许绕圆心旋转，而手镯排列问题中同时允许旋转和翻转。在给定的对称操作下，从问题的一个解（例如，一个环形排列）出发，经由对称操作可得到一组其他的解，我们称这些解在给定的对称操作下等价。

对称操作的出现使得部分解之间存在等价性，我们自然希望知道原问题有多少个互不等价的解。本章中我们将介绍这类问题的解决手段。我们依次介绍置换群、Burnside 引理和 Pólya 计数定理；其中，前者用于描述解集上所有可能的对称操作，后两者则基于置换群理论计算不等价的解的数目。

7.1 群论基础

本节中，我们介绍群 (group) 的基本概念。群是一类定义了一种二元运算的代数结构。

定义 7.1 (群). 给定集合 G 以及作用在 G 中元素上的一个二元运算 “ \cdot ”，若其满足以下四条性质，则称 G 在运算 “ \cdot ” 下为一个群：

- (1) 封闭性：对任意 $a, b \in G$ ，有 $a \cdot b \in G$ ；
- (2) 结合律：对任意 $a, b, c \in G$ ，有 $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ ；
- (3) 有单位元：存在 $e \in G$ ，使得对任意 $a \in G$ ，有 $a \cdot e = e \cdot a = a$ ；
- (4) 有逆元：对任意 $a \in G$ ，均存在 $b \in G$ ，使得 $a \cdot b = b \cdot a = e$ (记为 $b = a^{-1}$)。

由集合 G 与二元运算 “ \cdot ” 所确定的群记为 (G, \cdot) ；在不引发歧义的情况下，可直接将群记为 G 。

仅满足封闭性的代数结构 (G, \cdot) 称为原群 (magma)，满足封闭性和结合律的 (G, \cdot) 称为半群 (semigroup)，满足封闭性、结合律且有单位元的 (G, \cdot) 称为么半群 (monoid)。这三种代数结构与群形成了依次递进的关系。

定义 7.2 (群的有限性、群的阶数). 设 G 是一个群, 如果 G 中元素数目是有限的, 则称 G 是一个有限群 (finite group); 反之, 称 G 是一个无限群 (infinite group). 有限群 G 中的元素数目称为群 G 的阶 (order), 记为 $|G|$.

定义 7.3 (子群). 设 G 是一个群, $H \subseteq G$, 若 H 在原有运算下也构成一个群, 则称 H 是 G 的一个子群 (subgroup). 设 $e \in G$ 是 G 的单位元, 则 $\{e\}$ 在原有运算下也构成一个群, 称为 G 的平凡子群.

定义 7.4 (群同态、群同构). 设 (G, \cdot) 和 $(H, *)$ 均是群. 如果存在函数 $f: G \mapsto H$ 使得对 G 中的任意元素 u, v , 均有

$$f(u \cdot v) = f(u) * f(v)$$

则称 f 是从 G 到 H 的一个群同态 (homomorphism). 若 f 是单 (满) 射函数, 则称 f 为一个单 (满) 同态. 若 f 同时是单射和满射函数 (即双射函数), 则称 f 为一个群同构 (isomorphism), 此时称群 (G, \cdot) 与 $(H, *)$ 同构, 记作 $(G, \cdot) \cong (H, *)$.

下面我们展示几个简单的群. 大部分常见场景下, 我们不说 G 在某运算下是群, 而习惯说 G 是某运算群. 例如:

例 7.1.1. $G_1 = \{1\}$ 与 $G_2 = \{-1, 1\}$ 是乘法群.

二元运算 “ \cdot ” 为乘法运算的群称为乘法群. 我们以 $G_2 = \{-1, 1\}$ 为例证明它是乘法群:

- (1) 封闭性: $1 \times 1 = 1$, $1 \times (-1) = -1$, $(-1) \times 1 = -1$, $(-1) \times (-1) = 1$;
- (2) 结合律: 乘法天然满足结合律;
- (3) 有单位元: 1 是乘法单位元, 因此也是此群的单位元;
- (4) 有逆元: 1 的逆元是 1, -1 的逆元是 -1 .

综上, G_2 是乘法群. 证明 G_1 是乘法群的过程类似, 此处略去. 这两个群均为有限群.

例 7.1.2. $G = \{0, 1, 2, \dots, n-1\}$ 是模 n 加法群.

这个例子中, 二元运算 “ \cdot ” 是模 n 加法运算, 即

$$a \cdot b \triangleq (a + b) \bmod n$$

证明 G 是模 n 加法群的思路与前一个例子类似:

- (1) 封闭性: 任何非负整数模 n 的余数只能是 $0, 1, 2, \dots, n-1$ 中的一个, 因此封闭性一定成立;
- (2) 结合律: 加法天然满足结合律, 在模 n 后结合律仍然保持;
- (3) 有单位元: 0 是加法单位元, 因此也是此群的单位元;
- (4) 有逆元: 对任意 $a \in G$, $((n-a) \bmod n) \in G$, 并且

$$(a + ((n-a) \bmod n)) \bmod n = (a + (n-a)) \bmod n = 0$$

所以 $a^{-1} = (n-a)$.

综上, G 是模 n 加法群. G 中仅包含 n 个元素, 因此是有限群.

例 7.1.3. 令 α 取遍 $[0, 2\pi)$ 中的全体实数, 则二维欧几里得空间中的全体旋转矩阵

$$T_\alpha = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}$$

构成集合 $G = \{T_\alpha\}$, 该集合与矩阵乘法运算共同构成一个乘法群, 记为 $SO(2)$. 此群的单位元是 2 阶单位矩阵

$$T_0 = I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

群中任何一个元素 T_α 的逆元则是 $T_{-\alpha}$. 由于 $SO(2)$ 中含有无穷多个矩阵, 因此这是一个无限群.

例 7.1.4. 令 S 为全体行列式为 ± 1 的 3 阶方阵构成的集合, 则 S 构成一个乘法群. S 满足群的四条性质:

- (1) 封闭性: 3 阶方阵互乘仍然得到 3 阶方阵; 另外, 矩阵的积的行列式等于矩阵行列式的积, 因此行列式为 ± 1 的方阵相乘, 得到的矩阵行列式仍为 ± 1 ;
- (2) 结合律: 矩阵乘法天然满足结合律;
- (3) 有单位元: 3 阶单位矩阵 I_3 是矩阵乘法的单位元, 且其行列式为 1, 因此属于 S ;
- (4) 有逆元: 行列式非零的矩阵可逆, 且逆矩阵的行列式为原矩阵行列式的倒数, 因此对任意 $A \in S$ 有 $|A^{-1}| = \pm 1$, 从而 $A^{-1} \in S$.

综上, S 是乘法群, 且它显然是一个无限群.

由于群运算满足结合律, 因此“幂”“次方”等术语也用于指代对群内的某个元素反复进行群运算的操作. 幂运算中的指数只能是整数:

$$a^n = \begin{cases} \underbrace{a \cdot a \cdot \cdots \cdot a}_{n \uparrow a} & (n > 0) \\ e & (n = 0) \\ (a^{-1})^{-n} & (n < 0) \end{cases}$$

与数的乘法类似, 群运算 $a \cdot b$ 中的运算符 “ \cdot ” 在无歧义的情况下可以省略不写, 变为 ab .

下面我们介绍群的一些普适性质.

定理 7.5 (消去律). 设 (G, \cdot) 是群, $a, b, c \in G$, 则

$$a \cdot b = a \cdot c \implies b = c$$

相应地, 有逆否命题

$$b \neq c \implies a \cdot b \neq a \cdot c$$

证明. 若 $a \cdot b = a \cdot c$, 则在等式两侧左乘 a^{-1} , 得:

$$a^{-1} \cdot a \cdot b = a^{-1} \cdot a \cdot c$$

这表明 $e \cdot b = e \cdot c$, 从而 $b = c$. 逆否命题同时得证. \square

定理 7.6. 设 G 是有限群, $a \in G$, 则存在最小的正整数 r , 使得 $a^r = e$, 亦即 $a^{-1} = a^{r-1}$.

证明. 设 $|G| = g$, 则由群的封闭性,

$$a, a^2, a^3, \dots, a^g, a^{g+1} \in G$$

由鸽巢原理, 其中必有两项相同, 设这两项为 a^h 和 a^l ($1 \leq h < l \leq g+1$). 令 $l-h=r$, 则我们有 $a^h \cdot a^r = a^h$, 由消去律即有 $a^r = e$. 既然存在 r 使 $a^r = e$, r 又是正整数, 因此必定存在一个最小的 r . \square

根据子群的定义, $C_n = \{e, a, a^2, \dots, a^{r-1}\}$ 是 G 的一个子群, 其中所有元素都可以表示为 a^k 的形式. C_n 称为 n 次循环群 (cyclic group), 元素 a 称为此群的一个生成元.

定理 7.7. 设 (G, \cdot) 是群, G 的封闭子结构指的是 G 的一个子集 H , 满足 (H, \cdot) 是原群. 这时, H 必定是 G 的子群.

证明. 我们依次证明 H 满足群的四条性质.

- (1) 封闭性: (H, \cdot) 作为原群天然满足封闭性;
- (2) 结合律: 原有二元运算在 G 上满足结合律, 在 H 上仍满足结合律;
- (3) 有单位元: 设 G 的单位元为 e , 由推论 7.6, 任取 H 中一元素 a , 必存在最小的正整数 r 使得 $a^r = e$; 由 H 的封闭性知 $e \in H$, 从而 e 即为 H 的单位元;
- (4) 有逆元: 对任意 $a \in H$, 若 $a^r = e$, 则 $a^{-1} = a^{r-1} \in H$, 这表明 H 中所有元素的逆元也在 H 中.

综上所述, H 是 G 的子群. \square

定理 7.7 的意义在于, 如果我们发现在某种运算下具有封闭性的集合是一个群的子集, 则可以立即断定该集合在相同的运算下也构成一个群.

定义 7.8 (陪集). 设 H 是 G 的子群, 给定 $a \in G$, 集合

$$aH = \{ax \mid x \in H\} \quad (7-1)$$

称为 H 由 a 生成的左陪集. 类似地,

$$Ha = \{xa \mid x \in H\} \quad (7-2)$$

称为 H 由 a 生成的右陪集.

由消去律可知, $|aH| = |Ha| = |H|$.

7.2 置换群

置换群即置换 (permutation) 构成的群. 所谓置换即有限集到其自身的双射. 我们首先给出置换的定义.

定义 7.9 (置换). 有限集 $D = \{1, 2, \dots, n\}$ 到自身的一个双射 $\sigma: D \mapsto D$ 称为 D 上的一个 n 阶置换, 记作

$$\sigma = \begin{pmatrix} 1 & 2 & \cdots & n \\ \sigma(1) & \sigma(2) & \cdots & \sigma(n) \end{pmatrix} \quad (7-3)$$

上式中, 数阵的第一行是集合 D 中的所有元素, 一般以升序排列, 但也可以任意顺序排列; 第二行则是第一行中对应元素在映射 σ 下的像.

不失一般性地, 本节后续在讨论置换时, 除非特别说明, 否则始终假定讨论的是有限集 $D = \{1, 2, \dots, n\}$ 上的置换.

在不产生歧义的情况下, n 阶置换中没有发生变化的元素可以省略不写. 例如, 考虑一个只交换了 i, j ($1 \leq i < j \leq n$) 两数的 n 阶置换:

$$\begin{pmatrix} 1 & 2 & \cdots & i & \cdots & j & \cdots & n \\ 1 & 2 & \cdots & j & \cdots & i & \cdots & n \end{pmatrix}$$

若省略掉没有发生变化的元素, 则该置换可以简写为

$$\begin{pmatrix} i & j \\ j & i \end{pmatrix}$$

这仍然是一个 n 阶置换, 只是除 i, j 外的 $n-2$ 个元素经过该置换都保持不变.

集合 D 上的置换是该集到自身的双射, 两个双射复合后仍然得到一个双射, 因此置换之间可以复合. 设有置换

$$\sigma = \begin{pmatrix} 1 & 2 & \cdots & n \\ \sigma(1) & \sigma(2) & \cdots & \sigma(n) \end{pmatrix}$$

$$\tau = \begin{pmatrix} 1 & 2 & \cdots & n \\ \tau(1) & \tau(2) & \cdots & \tau(n) \end{pmatrix}$$

则二者的复合定义为

$$\tau \circ \sigma = \begin{pmatrix} 1 & 2 & \cdots & n \\ \tau(\sigma(1)) & \tau(\sigma(2)) & \cdots & \tau(\sigma(n)) \end{pmatrix} \quad (7-4)$$

置换的复合运算一般直接称为置换乘法.

例 7.2.1. 设

$$\sigma = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}, \tau = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$$

则因为

$$\sigma(\tau(1)) = \sigma(2) = 3, \sigma(\tau(2)) = \sigma(3) = 2, \sigma(\tau(3)) = \sigma(1) = 1$$

所以有

$$\sigma \cdot \tau = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

类似地, 有

$$\tau \cdot \sigma = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$$

这表明 $\sigma\tau \neq \tau\sigma$, 即置换乘法运算一般不满足交换律. 另一方面, 由于映射的复合运算满足结合律, 因此置换乘法运算也相应地满足结合律.

定义 7.10 (对称群). 设有限集 $D = \{1, 2, \dots, n\}$, S_n 是 D 上的全体 $n!$ 个置换构成的集合, 则 S_n 在置换乘法运算下构成一个群, 称为 n 次对称群 (symmetric group).

证明. 集合 S_n 包含全体 n 阶置换, 因此天然对置换乘法运算封闭, 此外置换乘法也满足结合律. 因此, 要使 S_n 是群, 只需其有单位元, 并且其中所有元素有逆元.

- 单位元: 恒等置换

$$e = \begin{pmatrix} 1 & 2 & \cdots & n \\ 1 & 2 & \cdots & n \end{pmatrix} \quad (7-5)$$

是置换乘法的单位元, 满足对任意置换 $\sigma \in S_n$, 有 $e\sigma = \sigma e = \sigma$;

- 逆元: 任一双射均有逆映射, 因此 S_n 中的每个置换均存在逆置换. 具体地, 若

$$\sigma = \begin{pmatrix} 1 & 2 & \cdots & n \\ \sigma(1) & \sigma(2) & \cdots & \sigma(n) \end{pmatrix}$$

则其逆置换为

$$\sigma^{-1} = \begin{pmatrix} \sigma(1) & \sigma(2) & \cdots & \sigma(n) \\ 1 & 2 & \cdots & n \end{pmatrix} \quad (7-6)$$

综上所述, S_n 是群. □

任何由 n 阶置换构成的群 G 均为 S_n 的子群, 因为 S_n 中包含了 G 中所有可能出现的置换. 由此, 我们可按如下方式定义置换群:

定义 7.11 (置换群). 若 G 是 S_n 的子群, 则称 G 为 n 次置换群.

例 7.2.2. 全体 3 阶置换构成 3 次对称群 S_3 :

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}, \\ \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

例 7.2.3. 设想将一个大正方形划分为四个边长为原先二分之一的小正方形, 小正方形顺时针编号为 1、2、3、4. 将大正方形绕其中心顺时针旋转 90° , 可与旋转前的大正方形重合, 如图 7.1 所示.



图 7.1: 正方形旋转 90° 后与自身重合

通过考察四个小正方形所处的位置上编号的变化, 我们可以将“顺时针旋转 90° ”这一操作表示为如下置换:

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix}$$

其含义是, 旋转前 1 号小正方形占据的位置在旋转后被 4 号小正方形占据、旋转前 2 号小正方形占据的位置在旋转后被 1 号小正方形占据, 等等. 同理, “顺时针旋转 180° ”、“顺时针旋转 270° ”, 以及“旋转 0° ”这三个操作, 也可分别表示为如下置换:

$$\sigma^2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{pmatrix} \\ \sigma^3 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix} \\ e = \sigma^4 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

以上四个置换涵盖了正方形绕其中心旋转后与自身完全重合的所有可能情况, 它们构成一个群 $\{e, \sigma, \sigma^2, \sigma^3\}$, 这实际上是一个四次循环群.

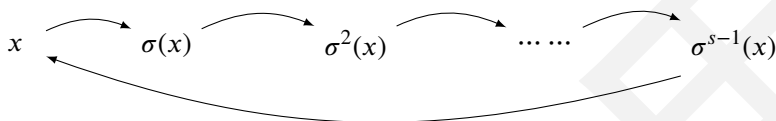
下面介绍置换的循环表示. 考虑任意 n 阶置换

$$\sigma = \begin{pmatrix} 1 & 2 & \cdots & n \\ \sigma(1) & \sigma(2) & \cdots & \sigma(n) \end{pmatrix}$$

定理 7.6 指出存在正整数 r 使得 σ^r 是恒等置换. 考虑 $\{1, 2, \dots, n\}$ 中的任意一个元素 x : 不断对其施加置换 σ , 将得到 $\sigma(x)$ 、 $\sigma(\sigma(x))$ 、 \dots , 以此类推; 由于 σ^r 是恒等置换, 因此必然存在最小的正整数 $s \leq r$, 使得 $x = \sigma^s(x)$. 考察

$$x, \sigma(x), \sigma^2(x), \dots, \sigma^{s-1}(x)$$

这 s 个互异元素在置换 σ 下的变化. $\sigma^k(x)$ ($k = 0, 1, 2, \dots, s-1$) 将被置换 σ 映射为 $\sigma^{k+1}(x)$, $\sigma^{s-1}(x)$ 则被映射为 x :



容易看出, 以上 s 个互异元素在置换 σ 下形成一个循环 (cycle), 置换 σ 将此循环中的每一个元素映射为其在循环中的下一个元素. 我们将此循环表示为如下形式:

$$(x \ \sigma(x) \ \sigma^2(x) \ \dots \ \sigma^{s-1}(x)) = \begin{pmatrix} x & \sigma(x) & \dots & \sigma^{s-2}(x) & \sigma^{s-1}(x) \\ \sigma(x) & \sigma^2(x) & \dots & \sigma^{s-1}(x) & x \end{pmatrix} \quad (7-7)$$

上式中, 等式右侧是一个 n 阶置换, 它描述了置换 σ 对元素 x 及其所在的循环中的元素的映射关系, 不在此循环中的元素则保持不变. 等式左侧是循环的记号, 它将循环中的元素依次写在一对小括号中, 除最后一个元素外的每个元素均被映射为其在括号中的下一个元素, 最后一个元素则被映射为第一个元素.

显然, 置换 σ 下的任意两个不同的循环都不可能拥有相同的元素 (称它们不相交), 同时 $\{1, 2, \dots, n\}$ 中的每个元素也均属于某一个循环. 由此我们可将置换 σ 分解为一系列不相交的形如式 (7-7) 的循环之积, 使得每个元素都恰好出现在一个循环中, 此即为置换的循环表示. 将置换表示为不相交循环之积的过程称为置换的循环分解.

循环中包含的元素的数目称为其阶数或长度. 考虑任意 s 阶循环

$$(a_1 a_2 \dots a_s)$$

注意到

$$(a_1 a_2 \dots a_s) = (a_2 a_3 \dots a_s a_1) = \dots = (a_s a_1 \dots a_{s-1})$$

因此每个 s 阶循环都有 s 种表示方式, 取决于将循环中的哪个元素写在括号中的首位. 这些表示方式彼此等价.

例 7.2.4. 设有 7 阶置换

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 6 & 1 & 4 & 2 & 5 & 7 \end{pmatrix}$$

则其循环表示为

$$\sigma = (1 \ 3)(2 \ 6 \ 5)(4)(7)$$

在上述循环表示中, (4), (7) 这两个一阶循环实际上均等同于恒等置换, 但是不能因此就将其省略, 也不能写成 (4), (7) 以外的一阶循环形式.

在置换 σ 的循环分解中, k 阶循环的出现次数记为 $c_k(\sigma)$. 特别地, $c_1(\sigma)$ 是置换 σ 的不动点数量, 所有一阶循环中的元素均被 σ 映射到其自身而不发生改变. 置换 σ 的循环分解中各种长度的循环的出现次数合称为 σ 的循环结构, 记作

$$(1)^{c_1(\sigma)}(2)^{c_2(\sigma)} \dots (n)^{c_n(\sigma)}$$

其中若某个 $c_k(\sigma) = 0$ ($k = 1, 2, \dots, n$), 则 $(k)^{c_k(\sigma)}$ 这一项可省略不写. σ 的循环结构中的循环总数记作 $c(\sigma)$:

$$c(\sigma) = \sum_{k=1}^n c_k(\sigma) \quad (7-8)$$

若置换 σ, τ 满足

$$c_k(\sigma) = c_k(\tau) \quad (k = 1, 2, \dots, n)$$

则称它们具有相同的循环结构. S_n 中具有某种相同循环结构的全体置换构成一个共轭类 (conjugacy class).

定理 7.12 (不相交循环之积可交换). 设 $\sigma = (a_1 a_2 \dots a_s)$ 和 $\tau = (b_1 b_2 \dots b_t)$ 是两个不相交的循环, 则 $\sigma\tau = \tau\sigma$.

证明. 因为 σ, τ 不相交, 所以 a_1, a_2, \dots, a_s 与 b_1, b_2, \dots, b_t 这 $s+t$ 个元素互不相同. 由此可得

$$\begin{aligned} \sigma\tau &= \begin{pmatrix} a_1 & a_2 & \cdots & a_s \\ a_2 & a_3 & \cdots & a_1 \end{pmatrix} \begin{pmatrix} b_1 & b_2 & \cdots & b_t \\ b_2 & b_3 & \cdots & b_1 \end{pmatrix} \\ &= \begin{pmatrix} a_1 & a_2 & \cdots & a_s & b_1 & b_2 & \cdots & b_t \\ a_2 & a_3 & \cdots & a_1 & b_2 & b_3 & \cdots & b_1 \end{pmatrix} \\ &= \begin{pmatrix} b_1 & b_2 & \cdots & b_t \\ b_2 & b_3 & \cdots & b_1 \end{pmatrix} \begin{pmatrix} a_1 & a_2 & \cdots & a_s \\ a_2 & a_3 & \cdots & a_1 \end{pmatrix} = \tau\sigma \quad \square \end{aligned}$$

定理 7.12 表明, 在置换的循环分解中, 不同的循环可以任意顺序出现. 这表明, 若不考虑循环之间的顺序, 以及每个循环内元素的排列顺序, 则置换的循环分解是唯一的.

定理 7.13. 设 $\sigma = (a_1 a_2 \dots a_s)$, 则 $\sigma^s = e$.

证明. 循环 σ 中的任一元素 x 在施加 s 次置换 σ 后, 将被映射为其在循环中的后第 s 个元素; 由于循环的长度即为 s , 因此这意味着 x 将被映射为其自身. 因此, σ^s 将循环中的每一个元素映射为其自身, 不在循环中的元素则始终保持不变, 从而 σ^s 即为恒等置换 e . \square

推论 7.14. 设置换 σ 可分解为 k 个不相交循环之积, 各循环的长度分别为 s_1, s_2, \dots, s_k , 则

$$\sigma^{\text{lcs}(s_1, s_2, \dots, s_k)} = e$$

其中 $\text{lcs}(s_1, s_2, \dots, s_k)$ 表示 s_1, s_2, \dots, s_k 的最小公倍数.

证明. 不妨设

$$\sigma = (a_{11} \cdots a_{1s_1})(a_{21} \cdots a_{2s_2}) \cdots (a_{k1} \cdots a_{ks_k})$$

由于互不相交的置换的乘法可以交换, 我们有

$$\begin{aligned} \sigma^{\text{lcs}(s_1, s_2, \dots, s_k)} &= (a_{11} \cdots a_{1s_1})^{\text{lcs}(s_1, s_2, \dots, s_k)} \cdots (a_{k1} \cdots a_{ks_k})^{\text{lcs}(s_1, s_2, \dots, s_k)} \\ &= e^{\text{lcs}(s_1, s_2, \dots, s_k)/s_1} \cdots e^{\text{lcs}(s_1, s_2, \dots, s_k)/s_k} \\ &= e \end{aligned}$$

□

例 7.2.5. 一副去除了大小王的扑克牌包含 52 张牌. 所谓混洗是指如下操作: 将牌堆从中央分开得到牌数相等 (均为 26 张) 的两叠, 然后交错互相插入合为一叠, 使得牌堆顶、牌堆底的两张牌保持不变.

牌堆中的每张牌可从顶到底依次编号为 $1, 2, \dots, 52$; 一次混洗首先将其分为 $1, 2, \dots, 26$ 和 $27, 28, \dots, 52$ 两叠, 然后交错插入, 得到编号依次为

$$1, 27, 2, 28, 3, 29, \dots, 26, 52$$

的一叠. 因此, 混洗操作可表示为如下置换:

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & \cdots & 51 & 52 \\ 1 & 27 & 2 & 28 & 3 & 29 & \cdots & 26 & 52 \end{pmatrix}$$

置换 σ 有如下循环分解:

$$\begin{aligned} \sigma &= (1)(2 \ 27 \ 14 \ 33 \ 17 \ 9 \ 5 \ 3)(4 \ 28 \ 40 \ 46 \ 49 \ 25 \ 13 \ 7)(6 \ 29 \ 15 \ 8 \ 30 \ 41 \ 21 \ 11) \\ &\quad (10 \ 31 \ 16 \ 34 \ 43 \ 22 \ 37 \ 19)(12 \ 32 \ 42 \ 47 \ 24 \ 38 \ 45 \ 23)(18 \ 35) \\ &\quad (20 \ 36 \ 44 \ 48 \ 50 \ 51 \ 26 \ 39)(52) \end{aligned}$$

等式右侧包含 2 个 1 阶循环、1 个 2 阶循环和 6 个 8 阶循环; 所有循环长度的最小公倍数为 8, 因此重复进行 8 次混洗就能让牌堆复原.

例 7.2.6. 5.4 节中讨论了无符号的第一类 Stirling 数 $|s(n, k)|$, 其组合意义是将一个 n 元素集分为 k 个非空子集, 并对每个子集中的元素作圆排列的方案数. 注意到非空的圆排列等价于循环, 因此 $|s(n, k)|$ 即表示可分解为恰好 k 个不相交循环之积的 n 阶置换的数目. 由于任何一个 n 阶置换作循环分解, 得到的循环数目只可能是 $1, 2, \dots, n$ 其中之一, 因此

$$\sum_{k=1}^n |s(n, k)| = n!$$

下面我们研究一类特殊的置换, 称为**对换** (transposition), 或称**换位**. 对换即 2 阶循环, 它仅交换两个元素的位置, 对其他元素没有影响.

定理 7.15. 设对换 $\sigma = (a \ b)$, 则 $\sigma^{-1} = \sigma$.

证明. 两次施加同一对换, 相当于将两个元素交换两次, 这时两个元素均回到原先的位置上. 因此, 任意对换 $(a \ b)$ 的逆均是其自身. \square

定理 7.16. 任何循环均能表示为对换之积:

$$(a_1 \ a_2 \ a_3 \ \cdots \ a_n) = (a_1 \ a_n) (a_1 \ a_{n-1}) \cdots (a_1 \ a_2)$$

规定 0 个对换的积是恒等置换 e (可看作 1 阶循环).

证明. 对循环长度 n 施加归纳. 当 $n = 2$ 时, 命题显然成立. 当 $n > 2$ 时, 根据归纳假设, 我们有

$$\begin{aligned} & (a_1 \ a_n) (a_1 \ a_{n-1}) \cdots (a_1 \ a_2) \\ &= (a_1 \ a_n) (a_1 \ a_2 \ a_3 \ \cdots \ a_{n-1}) \\ &= \begin{pmatrix} a_1 & a_2 & a_3 & \cdots & a_{n-1} & a_n \\ a_n & a_2 & a_3 & \cdots & a_{n-1} & a_1 \end{pmatrix} \begin{pmatrix} a_1 & a_2 & a_3 & \cdots & a_{n-1} & a_n \\ a_2 & a_3 & a_4 & \cdots & a_1 & a_n \end{pmatrix} \\ &= \begin{pmatrix} a_1 & a_2 & a_3 & \cdots & a_{n-1} & a_n \\ a_2 & a_3 & a_4 & \cdots & a_n & a_1 \end{pmatrix} \\ &= (a_1 \ a_2 \ a_3 \ \cdots \ a_n) \end{aligned}$$

\square

任何置换均能分解为不相交循环之积, 从而也就能分解为对换之积.

推论 7.17. 设置换

$$\sigma = (a_1 \ b_1) (a_2 \ b_2) \cdots (a_n \ b_n)$$

则其逆元为

$$\sigma^{-1} = (a_n \ b_n) (a_{n-1} \ b_{n-1}) \cdots (a_1 \ b_1)$$

证明.

$$\begin{aligned} \sigma \sigma^{-1} &= (a_1 \ b_1) (a_2 \ b_2) \cdots (a_n \ b_n) (a_n \ b_n) (a_{n-1} \ b_{n-1}) \cdots (a_1 \ b_1) \\ &= (a_1 \ b_1) (a_2 \ b_2) \cdots (a_{n-1} \ b_{n-1}) (a_{n-1} \ b_{n-1}) \cdots (a_1 \ b_1) \\ &= \cdots \\ &= (a_1 \ b_1) (a_1 \ b_1) = e \end{aligned}$$

\square

定理 7.18. 给定置换 σ , 将 σ 分解为对换之积的所有方式中, 对换数目的奇偶性相同.

证明. 用反证法. 假设

$$\sigma = \begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}$$

并且它拥有两种对换数目奇偶性不同的分解方式:

$$\sigma = (a_1 b_1)(a_2 b_2) \cdots (a_{m_1} b_{m_1}) \quad (m_1 \text{ 为奇数})$$

$$\sigma = (c_1 d_1)(c_2 d_2) \cdots (c_{m_2} d_{m_2}) \quad (m_2 \text{ 为偶数})$$

考察函数

$$F(\sigma) = \prod_{j < k} (i_j - i_k)$$

显然 $F(\sigma) \neq 0$. 任取两个互异整数 $r, s \in \{1, 2, \cdots, n\}$ (其中 $r < s$), 提取出上式右侧与 i_r, i_s 相关的项, 可将其变形为

$$F(\sigma) = (-1)^{s-r-1} \left[\prod_{j < k, \{j, k\} \cap \{r, s\} = \emptyset} (i_j - i_k) \right] (i_r - i_s) \prod_{j \neq r, s} (i_j - i_r) (i_j - i_s)$$

考虑置换

$$\sigma' = \sigma \cdot (rs) = \begin{pmatrix} 1 & 2 & \cdots & r & \cdots & s & \cdots & n \\ i_1 & i_2 & \cdots & i_s & \cdots & i_r & \cdots & i_n \end{pmatrix}$$

此置换相当于在 σ 中将 i_r 与 i_s 对调. 比较 $F(\sigma')$ 与 $F(\sigma)$, 可以发现只有 $(i_r - i_s)$ 一项的符号发生变化, 而其他项均不变; 这表明, 在 σ 上右乘一个对换, 将会使函数 F 的值反号.

现在考察 $F(\sigma\sigma^{-1}\sigma)$. 一方面, 这即等价于 $F(\sigma)$; 另一方面, 我们可利用 σ 的对换分解, 将上式展开为:

$$F(\sigma\sigma^{-1}\sigma) = F(\sigma \cdot (a_{m_1} b_{m_1}) \cdots (a_1 b_1)(c_1 d_1) \cdots (c_{m_2} d_{m_2}))$$

在上式右侧, σ 总共右乘了 $m_1 + m_2$ 个对换; 由于 $m_1 + m_2$ 是奇数, 因此我们有 $F(\sigma\sigma^{-1}\sigma) = -F(\sigma)$, 矛盾.

综上所述, 将置换分解为对换之积, 对换数量的奇偶性必定相同. \square

通过考察将置换分解为对换之积时对换数目的奇偶性, 我们可将全体 n 阶置换分为两大类: 能分解为奇数个对换之积的称为**奇置换**, 能分解为偶数个对换之积的称为**偶置换**.

推论 7.19. n 阶奇置换与 n 阶偶置换的数量相同, 均为 $\frac{n!}{2}$.

证明. 设 A_n 、 B_n 分别为全体 n 阶奇置换、偶置换构成的集合. 显然,

$$A_n \cap B_n = \emptyset, |A_n| + |B_n| = n!$$

根据消去律的逆否命题, 对群中的任意三个元素 a, b, c , 若 $b \neq c$, 则必有 $ab \neq ac$. 因此, 任取两个互异元素 i, j , 我们有

- 对任意 $\sigma \in A_n$, $(i j) \cdot \sigma \in B_n$, 故 $|B_n| \leq |A_n|$;
- 对任意 $\tau \in B_n$, $(i j) \cdot \tau \in A_n$, 故 $|A_n| \leq |B_n|$.

由此即有

$$|A_n| = |B_n| = \frac{n!}{2} \quad \square$$

定义 7.20 (交错群). 全体 n 阶偶置换构成一个群, 称为 n 次交错群 (alternating group), 记为 A_n .

证明. 依次证明群的四条性质成立即可:

- (1) 封闭性: 偶置换之积仍然是偶置换;
- (2) 结合律: 自然满足;
- (3) 有单位元: n 阶恒等置换 e 是偶置换, 因此 $e \in A_n$ 是单位元;
- (4) 有逆元: 偶置换

$$p = (a_1 b_1)(a_2 b_2) \cdots (a_m b_m)$$

的逆元为

$$p^{-1} = (a_m b_m)(a_{m-1} b_{m-1}) \cdots (a_1 b_1)$$

仍然是一个偶置换.

综上所述, A_n 是群. □

7.3 Burnside 引理

本节中, 我们讨论 Burnside 引理, 它可用于计算群 G 上的等价类数目. 我们首先给出相关概念, 然后介绍 Burnside 引理并给出其证明.

定义 7.21 (等价类). 设 G 是有限集 C 上的置换群. 若 G 中存在一个置换 σ 将 C 中某个元素 x 映射为元素 y , 则称 x, y 属于同一个等价类 (orbit), 或直译为轨道. x 所属的等价类记为 E_x :

$$E_x = \{\sigma(x) \mid \sigma \in G\} \quad (7-9)$$

称同属于一个等价类的元素在群 G 作用下等价. 由群 G 所确定的 C 上的全体等价类合称为 G 诱导的等价类.

若 C 是某一组合问题的解集, G 是由解的对称操作所确定的置换群, 则 G 诱导的等价类的数目即为我们关心的不等价的解的数目.

例 7.3.1. 一个圆沿直径分为左右两个半圆, 为每个半圆染上黑、白两种颜色之一. 根据乘法原理, 显然共有 $2 \times 2 = 4$ 四种染色方案, 将其分别编号为 1, 2, 3, 4, 如图 7.2 所示. 由此可得解集 $C = \{1, 2, 3, 4\}$.

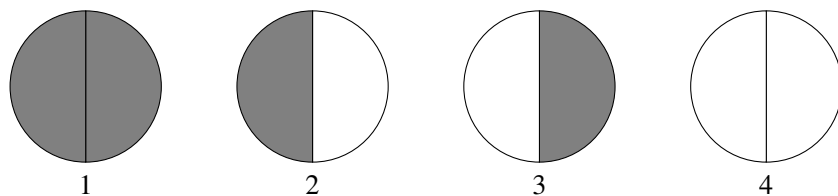


图 7.2: 例 7.3.1 中的四种染色方案

考虑如下对称操作: 将圆绕直径翻转 180° , 若一种染色方案经翻转后可成为另一种染色方案, 则认为这两种染色方案等价. 上述对称操作在解集 $C = \{1, 2, 3, 4\}$ 上确定了如下 4 次 2 置换群:

$$G = \left\{ e = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}, \sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 2 & 4 \end{pmatrix} \right\}$$

此群中的两个置换分别为恒等置换 e , 表示若不进行翻转, 则所有染色方案均不会变化; 以及一个对换 $\sigma = (2\ 3)$, 代表若沿直径将圆翻转 180° , 将使两个半圆一黑一白的 2 个染色方案分别变为另一个, 而两个半圆相同颜色的 2 个方案不变.

G 在 C 上诱导出如下 3 个等价类:

$$E_1 = \{1\}, E_2 = E_3 = \{2, 3\}, E_4 = \{4\}$$

因此, 在允许沿直径作 180° 翻转的情况下, 这个例子中的染色问题有 3 个不等价染色方案.

定义 7.22 (不动置换类). 设 G 是有限集 C 上的置换群. 对 C 中的某个元素 x , G 中使 x 保持不变的全体置换构成 x 的**不动置换类** (stabilizer), 或直译为**稳定集**, 记为 Z_x :

$$Z_x = \{\sigma \in G \mid \sigma(x) = x\} \quad (7-10)$$

例 7.3.2. 考虑例 7.3.1 中的集合 $C = \{1, 2, 3, 4\}$ 以及其上的置换群

$$G = \{e, \sigma\}$$

C 中每个元素的不动置换类分别为

$$Z_1 = Z_4 = \{e, \sigma\}, Z_2 = Z_3 = \{e\}$$

定理 7.23. 给定任意 $x \in C$, 不动置换类 Z_x 构成一个 C 上的置换群.

证明. 依次证明群的四条性质成立即可:

- (1) 封闭性: 若 $\sigma, \tau \in Z_x$, 则 $\sigma(\tau(x)) = \sigma(x) = x$, 故 $\sigma\tau \in Z_x$;
- (2) 结合律: Z_x 中的元素均为置换, 置换本身即满足结合律;

(3) 有单位元: 恒等置换 e 使 C 中的任何元素均保持不变, 因此必然有 $e \in Z_x$, 从而 e 即为 Z_x 的单位元;

(4) 有逆元: 若 $\sigma \in Z_x$, 则 $\sigma(x) = x$, 由逆映射的定义有 $\sigma^{-1}(x) = x$, 因此 $\sigma^{-1} \in Z_x$.

综上所述, Z_x 是群. □

等价类与不动置换类这两个概念间存在着紧密的联系, 我们有如下定理:

定理 7.24 (轨道-稳定集定理). 设 G 是有限集 C 上的置换群. 对任意 $x \in C$, 设 E_x 是 x 所属的等价类, Z_x 是 x 的不动置换类, 则

$$|E_x| |Z_x| = |G| \quad (7-11)$$

证明. 设

$$E_x = \{x_1, x_2, \dots, x_m\}$$

由于 $x \in E_x$, 不妨设 $x_1 = x$. 根据等价类的定义, G 中的任何一个置换作用于 x 上, 必定将其映射为 x_1, x_2, \dots, x_m 中的某一个元素; 同时, 对 E_x 中的任意元素 x_i ($i = 1, 2, \dots, m$), G 中都存在至少一个置换 σ_i 能将 x 映射为 x_i . 由此, 我们可以将 G 中的所有置换, 按照元素 x 在该置换下的像, 划分为 m 个非空子集 G_1, G_2, \dots, G_m :

$$G_i = \{\sigma \in G \mid \sigma(x) = x_i\}$$

G_1, G_2, \dots, G_m 显然是 G 的一个划分:

- 互斥性: 若 $i \neq j$ ($i, j = 1, 2, \dots, m$), 则 $G_i \cap G_j = \emptyset$, 否则将存在 $\sigma \in G$ 使得 $\sigma(x) = x_i = x_j$, 矛盾;
- 完备性: 若 $G \neq G_1 \cup G_2 \cup \dots \cup G_m$, 则存在 $\tau \in G$ 使得 $\tau(x) \notin E_x$, 与 E_x 的定义矛盾.

因此, 我们有

$$|G| = |G_1| + |G_2| + \dots + |G_m|$$

由于 G_1, G_2, \dots, G_m 均非空, 我们可从每个集合中均选出一个置换:

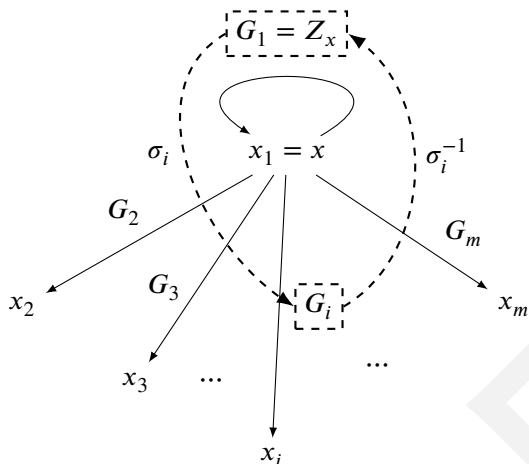
$$\sigma_1 \in G_1, \sigma_2 \in G_2, \dots, \sigma_m \in G_m$$

根据不动置换类的定义, $Z_x = G_1$, 从而 G_1 是群. 下面我们证明所有 G_i ($i = 1, 2, \dots, m$) 的大小均与 G_1 相同. 证明思路如图 7.3 所示. 考虑 G_1 由 σ_i 生成的左陪集 $\sigma_i G_1$, 对任意 $\tau \in G_1$, 因为 $\sigma_i(\tau(x)) = \sigma_i(x) = x_i$, 所以 $\sigma_i G_1 \subseteq G_i$. 再考虑 G_1 由 σ_i^{-1} 生成的左陪集 $\sigma_i^{-1} G_1$, 对任意 $\tau \in G_i$, 因为 $\sigma_i^{-1}(\tau(x)) = \sigma_i^{-1}(x_i) = x$, 所以 $\sigma_i^{-1} G_1 \subseteq G_1$. 根据陪集的性质及以上两个子集关系, 我们有

$$|G_1| = |\sigma_i G_1| \leq |G_i| = |\sigma_i^{-1} G_1| \leq |G_1| \quad (i = 1, 2, \dots, m)$$

由此可得

$$|G_1| = |G_2| = \dots = |G_m|$$

图 7.3: G_1 和 G_i 的大小必定相同

又因为 $|G_1| = |Z_x|$, 我们有

$$|G| = \sum_{i=1}^m |G_i| = \sum_{i=1}^m |Z_x| = |E_x| |Z_x| \quad \square$$

下面我们介绍 Burnside 引理并给出证明.

定理 7.25 (Burnside 引理). 设 G 是有限集 C 上的置换群, 则 G 诱导的 C 的等价类数目为

$$N = \frac{1}{|G|} \sum_{\sigma \in G} c_1(\sigma) \quad (7-12)$$

其中 $c_1(\sigma)$ 是置换 σ 的不动点数目.

证明. 设 G 诱导的 C 的等价类为 E_1, E_2, \dots, E_N , 这些等价类构成对 C 的一个划分. 我们首先证明, 等价类的数量 N 等于 C 中每个元素的不动置换类大小之和除以 G 的阶, 这可以利用轨道-稳定集定理证得:

$$N = \frac{1}{|G|} \sum_{i=1}^N |E_i| \cdot \frac{|G|}{|E_i|} = \frac{1}{|G|} \sum_{i=1}^N \sum_{x \in E_i} |Z_x| = \frac{1}{|G|} \sum_{x \in C} |Z_x|$$

下面我们证明, C 中元素的不动置换类大小之和等于 G 中所有置换的不动点数量之和. 令 $|G| = g$ 和 $|C| = c$, 并设 $G = \{\sigma_1, \sigma_2, \dots, \sigma_g\}$ 和 $C = \{x_1, x_2, \dots, x_c\}$, 我们定义

$$S_{jk} = \begin{cases} 1 & (\sigma_j(x_k) = x_k) \\ 0 & (\sigma_j(x_k) \neq x_k) \end{cases}$$

现在考虑由全体 S_{jk} 构成的 g 行 c 列表格:

$S_{jk} \backslash x_k$	x_1	x_2	\cdots	x_c	$c_1(\sigma_j)$
σ_j					
σ_1	S_{11}	S_{12}	\cdots	S_{1c}	$c_1(\sigma_1)$
σ_2	S_{21}	S_{22}	\cdots	S_{2c}	$c_1(\sigma_2)$
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
σ_g	S_{g1}	S_{g2}	\cdots	S_{gc}	$c_1(\sigma_g)$
$ Z_k $	$ Z_1 $	$ Z_2 $	\cdots	$ Z_c $	$\sum_{k=1}^c Z_k = \sum_{j=1}^g c_1(\sigma_j)$

考察该表格每行及每列的和的意义:

- 第 j 行所有 S_{jk} 的和表示在 σ_j 下不变的 C 中元素的数目, 即 $c_1(\sigma_j)$;
- 第 k 列所有 S_{jk} 的和表示使 x_k 不变的 G 中置换的数目, 即 $|Z_k|$.

对全体 S_{jk} 求和可视为先对行、再对列求和, 也可视为先对列、再对行求和, 二者的结果

必定相等, 即 $\sum_{k=1}^c |Z_k| = \sum_{j=1}^g c_1(\sigma_j)$. 结合上面证得的结论, 我们即有

$$N = \frac{1}{|G|} \sum_{x \in C} |Z_x| = \frac{1}{|G|} \sum_{\sigma \in G} c_1(\sigma) \quad \square$$

例 7.3.3. 设想将一个大正方形划分为四个边长为原先二分之一的小正方形, 使用黑、白两种颜色对每个小正方形染色. 若一种染色方案在大正方形绕其中心旋转后能与另一种染色方案完全重合, 则认为两种方案等价.

设所有染色方案构成的集合为 C ; 用两种颜色给四个小正方形染色, 一共有 $|C| = 2^4 = 16$ 种染色方案; 我们依次将这些染色方案编号为 $1, 2, \dots, 16$, 如图 7.4 所示.

大正方形绕中心逆时针旋转 0° 、 90° 、 180° 和 270° 四个操作对应置换群

$$G = \{e, \sigma_{90^\circ}, \sigma_{180^\circ}, \sigma_{270^\circ}\}$$

中的四个置换, 其中置换 σ 代表逆时针旋转 90° . 我们可将 G 中的每个置换分解为循环, 从而统计其不动点数目:

- 旋转 0° : $e = (1)(2) \cdots (16)$, 共 16 个不动点;
- 旋转 90° : $\sigma_{90^\circ} = (1)(2)(3\ 4\ 5\ 6)(7\ 8\ 9\ 10)(11\ 12)(13\ 14\ 15\ 16)$, 共 2 个不动点;
- 旋转 180° : $\sigma_{180^\circ} = (1)(2)(3\ 5)(4\ 6)(7\ 9)(8\ 10)(11)(12)(13\ 15)(14\ 16)$, 共 4 个不动点;
- 旋转 270° : $\sigma_{270^\circ} = (1)(2)(6\ 5\ 4\ 3)(10\ 9\ 8\ 7)(11\ 12)(16\ 15\ 14\ 13)$, 共 2 个不动点.

由 Burnside 引理, G 诱导的 C 中等价类的数目为

$$N = \frac{1}{4} \times (16 + 2 + 2 + 4) = 6$$

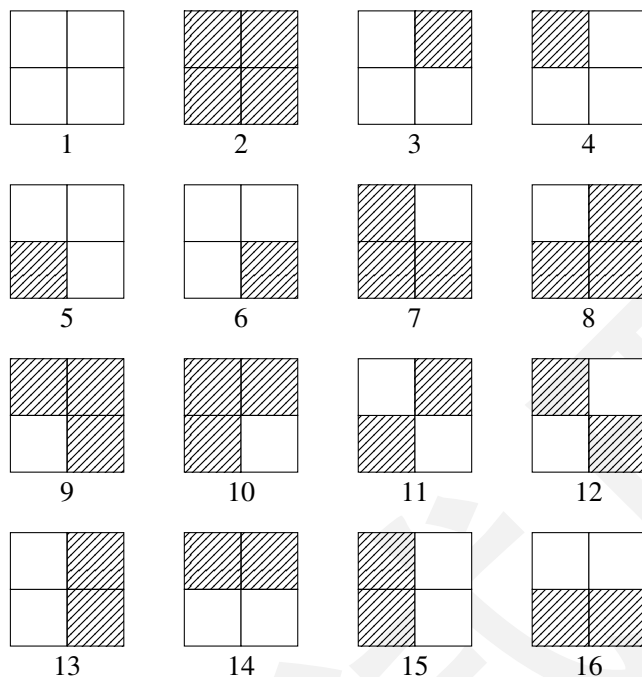


图 7.4: 四个小正方形的 16 种染色方案

因此, 一共有 6 种不等价的染色方案.

7.4 Pólya 计数定理

考虑对 n 元集合 D 中的元素进行 m 染色的问题. 令此问题的解集为 C , 则根据乘法原理显然有 $|C| = m^n$. 若 D 上存在一个置换群 G , 则 C 上也将存在一个与 G 同构的置换群 G' . 利用 Burnside 引理, 我们可求出 G' 诱导的 C 上的等价类数目, 但当 m^n 很大时, 讨论 G' 中每个置换的不动点数目会较为复杂. Pólya 计数定理可显著简化这一计算过程, 它允许我们通过 G 中置换的循环表示计算 G' 中置换的不动点数目.

我们先讨论一个简单的例子以展示 Pólya 计数定理的基本思想.

例 7.4.1. 设 $|D| = 6$, 置换

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 5 & 1 & 2 & 4 & 6 \end{pmatrix} = (1\ 3)(2\ 5\ 4)(6)$$

是 D 上的置换群 G 中的一个元素, 它可分解为 3 个不相交循环之积.

对 D 中的元素进行 m 染色, 全体染色方案构成集合 C , 其中的每个元素均代表 D 中元素的一个染色方案; 令元素 k 的颜色为 c_k , 则我们可用序列

$$c_1 c_2 c_3 c_4 c_5 c_6 \quad (1 \leq c_k \leq m, k = 1, 2, \dots, 6)$$

表示 C 中的一个元素. D 上的置换群 G 在 C 上有一个同构置换群 G' , 置换 σ 在 G' 中对应的置换设为 σ' , 其含义是将每个染色方案中的第 k 个元素换至第 $\sigma(k)$ 个位置处, 得到一个新的染色方案:

$$\sigma'(c_1 c_2 c_3 c_4 c_5 c_6) = c_3 c_4 c_1 c_5 c_2 c_6 \quad (1 \leq c_k \leq m, k = 1, 2, \dots, 6)$$

为了使用 Burnside 引理求出 G' 诱导的 C 上的等价类数目, 我们需求出 σ' 的不动点数目. 观察上式可知, 若要使 $c_1 c_2 c_3 c_4 c_5 c_6$ 是 σ' 的不动点, 则应有 $c_1 c_2 c_3 c_4 c_5 c_6 = c_3 c_4 c_1 c_5 c_2 c_6$, 即

$$c_1 = c_3, c_2 = c_5 = c_4, c_6 = c_6$$

上式表明, σ 的循环表示中, 每个循环内的元素必须染为相同的颜色. 此结论的正确性是显然的: 对 σ 的循环表示中的任一个循环而言, 其中的每个元素均会经 σ 依次映射为此循环中的其他元素 (除非循环长度为 1), 因此同一个循环中的元素颜色必须相同, 才能使染色方案成为 σ 的不动点.

上述结论表明, σ' 的不动点数目实际上等价于对 σ 的每个循环分别染一种颜色的方案数. σ 有 3 个循环, 每个循环均可染 m 种颜色中的任意一种, 因此我们就有

$$c_1(\sigma') = m^{c(\sigma)} = m^3$$

其中的 $c(\sigma)$ 在式 (7-8) 中定义, 表示 σ 的循环表示中循环的数目.

在 Burnside 引理中应用上面这个例子得出的结论, 我们即得到 **Pólya 计数定理** (Pólya Enumeration Theorem, 简称为 PET).

定理 7.26 (Pólya 计数定理). 设 G 是有限集 D 上的置换群, C 是由全体用 m 种颜色为 D 中元素染色的方案构成的集合, 则 G 诱导的 C 中的等价类数目为

$$N = \frac{1}{|G|} \sum_{\sigma \in G} m^{c(\sigma)} \quad (7-13)$$

例 7.4.2. 设想将一个大正方形划分为四个边长为原先二分之一的小正方形, 使用黑、白两种颜色对每个小正方形染色. 若一种染色方案在大正方形绕其中心旋转后能与另一种染色方案完全重合, 则认为两种方案等价.

与例 7.3.3 不同, 利用 Pólya 计数定理, 我们可直接考虑大正方形的旋转群 C_4 . 若将 4 个小正方形按顺时针编号为 1, 2, 3, 4, 则 C_4 中的四个置换分别为

$$e = (1)(2)(3)(4), \sigma_{90^\circ} = (4\ 3\ 2\ 1), \sigma_{180^\circ} = (1\ 3)(2\ 4), \sigma_{270^\circ} = (1\ 2\ 3\ 4)$$

由 Pólya 计数定理, 不等价的染色方案数为

$$N = \frac{1}{4} \times (2^4 + 2^2 + 2 \times 2^1) = 6$$

例 7.4.3. 为正 n 边形的顶点染上 m 种颜色, 绕正 n 边形中心旋转能完全重合的两种染色方案视为等价, 求不等价的染色方案的数量.

上个例子中讨论的四个小正方形二染色的问题实际上就是 $n = 4, m = 2$ 时的情况. 6.5 节中讨论过了与这个例子完全等价的可重圆排列问题; 当时, 利用 Möbius 反演, 我们求得的方案数为

$$T(n) = \frac{1}{n} \sum_{k|n} \varphi\left(\frac{n}{k}\right) m^k$$

现在我们换用 Pólya 计数定理解决这个问题.

正 n 边形的旋转群是 n 阶循环群 C_n , 需要讨论的是其中每个置换分解为多少个不相交循环的乘积. 将正 n 边形的顶点依次编号为 $0, 1, 2, \dots, n-1$, 则正 n 边形绕中心旋转 $\frac{k}{n} \cdot 360^\circ$ ($k = 1, 2, \dots, n$) 这一操作对应于置换

$$\sigma_k(i) = (i + k) \bmod n \quad (i = 0, 1, 2, \dots, n)$$

由对称性可知, σ_k 的循环表示中, 每个循环的大小均应相同, 设为 l . 考虑 0 号节点, 对此节点施加 l 次置换 σ_k 后, 该节点首次回到其原本位置, 这意味着 l 是

$$kl \equiv 0 \pmod{n}$$

的最小正整数解. 由此我们可直接解出

$$l = \frac{n}{\gcd(k, n)}$$

其中 $\gcd(k, n)$ 表示 k 与 n 的最大公约数.

上述推导指出, G 中的置换 σ_k 可被分解为 $\gcd(k, n)$ 个不相交循环之积. 根据 Pólya 计数定理, G 诱导的染色方案集 C 上的等价类数目为

$$N = \frac{1}{n} \sum_{k=1}^n m^{\gcd(k, n)} = \frac{1}{n} \sum_{d|n} \sum_{1 \leq k \leq n, \gcd(k, n)=d} m^d$$

若 $k \in \{1, 2, \dots, n\}$ 满足 $\gcd(k, n) = d$, 则 $\gcd\left(\frac{k}{d}, \frac{n}{d}\right) = 1$. 例 6.5.2 中介绍了欧拉函数 $\varphi(n)$, 它定义为小于等于 n 且与 n 互质的正整数的数目, 因此满足上述要求的 k 的数目即为 $\varphi\left(\frac{n}{d}\right)$. 将此结果代入上式, 即得

$$N = \frac{1}{n} \sum_{d|n} \varphi\left(\frac{n}{d}\right) m^d$$

这与我们之前使用 Möbius 反演求得的结果相同.

在染色问题中, 若需要关注每种颜色的使用次数, 则可以使用母函数形式的 Pólya 计数定理. 重新考察 Pólya 计数定理的推导过程: 对于某个循环 p , 将其表示为不相交循环乘

积的形式后, 其中每个循环上的元素都必须染相同的颜色, 不相交的循环则可以独立染色. 这启示我们使用母函数表示每个循环的颜色, 并在最终结果中通过提取多项式系数的方式求出使用特定次数某一颜色的染色方案数. 具体地, 在 Pólya 计数定理中, 被求和项是

$$m^{c(\sigma)}$$

为了体现每种颜色的使用情况, 我们引入变量 x_1, x_2, \dots, x_m 代表 m 种颜色, 用每个变量的幂次代表颜色的使用次数, 将被求和项改写为多项式

$$\prod_{k=1}^n \left(\sum_{j=1}^m x_j^k \right)^{c_k(\sigma)}$$

上式相当于将置换 σ 的循环结构中的 $(k)^{c_k(\sigma)}$ 替换为了 $(x_1^k + x_2^k + \dots + x_m^k)^{c_k(\sigma)}$.

定理 7.27 (母函数形式的 Pólya 计数定理). 设 G 是有限集 D 上的置换群, C 是由全体用 m 种颜色为 D 中元素染色的方案构成的集合, 则 G 诱导的 C 中等价类数目的母函数为

$$P(G) = \frac{1}{|G|} \sum_{\sigma \in G} \prod_{k=1}^n \left(\sum_{j=1}^m x_j^k \right)^{c_k(\sigma)} \quad (7-14)$$

其中 $c_k(\sigma)$ 是置换 σ 的循环表示中大小为 k 的循环的数量, 变量 x_1, x_2, \dots, x_m 分别代表 m 种不同的颜色. 若要求恰好使用 p_1 次颜色 x_1 、 p_2 次颜色 x_2 、……、 p_m 次颜色 x_m 对 D 中的元素染色, 则不等价染色方案的数目即为

$$[x_1^{p_1} x_2^{p_2} \dots x_m^{p_m}] P(G)$$

母函数形式的 Pólya 计数定理是普通的 Pólya 计数定理的一个更为精细的形式. 若令 $x_1 = x_2 = \dots = x_m = 1$, 则求和项即退化为 $m^{c(\sigma)}$, 上述定理退化为普通的 Pólya 计数定理.

7.5 空间多面体的染色问题

本节中, 我们基于 Pólya 计数定理, 具体地讨论一类染色方案计数问题. 此类问题中被染色的对象是空间多面体, 包括其顶点、棱和面; 多面体允许在空间中任意旋转, 经由旋转可完全重合的两种染色方案视为等价. 本节的内容可看作上一节内容的应用, 但因为涉及空间多面体群, 因此我们单独在这一节中作介绍.

我们首先讨论一个简单例子.

例 7.5.1. 设想有一正四面体, 用 3 种颜色对它的每个顶点进行染色; 若正四面体在空间中旋转可使一种染色方案与另一种方案完全重合, 则两个染色方案视为等价. 求不等价的染色方案数.

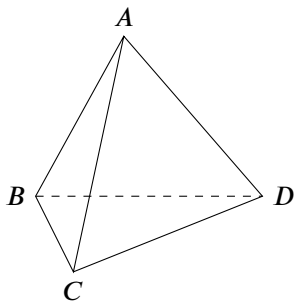


图 7.5: 正四面体示意图

图 7.5 展示了一个正四面体 $ABCD$, 我们先考察它的旋转群 G . 为此, 我们不妨考虑顶点 A , 该顶点可被旋转至图中 A, B, C, D 中的任一位置上, 因此 A 所在的等价类为 $E_A = \{A, B, C, D\}$, 即 $|E_A| = 4$; 另一方面, 若保持顶点 A 的位置不变, 则正四面体只能沿 A 到 $\triangle BCD$ 中心的连线旋转, 此旋转群即为 3 次循环群, 即 $|Z_A| = |C_3| = 3$. 根据轨道-稳定集定理, 我们有

$$|G| = |E_A| |Z_A| = 4 \times 3 = 12$$

按照 A 被旋转至的位置, 我们可分别列出 G 中的全部 12 个置换:

$$\begin{aligned} G = \{ & (A)(B)(C)(D), (A)(BCD), (A)(DCB), \\ & (AB)(CD), (ABD)(C), (ABC)(D), \\ & (AC)(BD), (ACB)(D), (ACD)(B), \\ & (AD)(BC), (ADB)(C), (ADC)(B) \} \end{aligned}$$

可见, G 中除 1 个恒等置换可分解为 4 个循环之积外, 其余 11 个置换均可分解为 2 个循环之积. 由 Pólya 计数定理, 不等价的染色方案数为

$$N = \frac{1}{12} \times (3^4 + 11 \times 3^2) = 15$$

除此之外, 注意到 G 是一个 4 次 12 阶置换群, 并且其中所有元素均为偶置换; 这表明 G 实际上即为 4 次交错群 A_4 .

对于较为复杂的多面体而言, 我们难以如上个例子这样列出其旋转群中的每一个置换; 但是, 通过分析不同类型的旋转操作, 我们可求出不同类型置换的循环结构, 从而同样能计算出每个置换的循环表示中循环的数目.

我们首先不加证明地给出如下结论:

定理 7.28. 任意多面体的旋转群必定与以下某种多面体的旋转群同构:

- 正棱锥 (对应于 n 次循环群 C_n);

- 正棱柱 (对应于 n 次二面体群 D_n);
- 正多面体.

其中, 正棱锥和正棱柱的旋转群取决于其底面的边数 n (允许出现 $n = 1, 2$ 的退化情况). 简单起见, 本节中, 我们仅讨论正多面体的染色问题.

正多面体即每个面均为全等的正多边形, 同时所有多面角也都相等的多面体. 三维空间中共有 5 种正多面体, 分别是正四面体、正六面体、正八面体、正十二面体和正二十面体, 如图 7.6 所示.

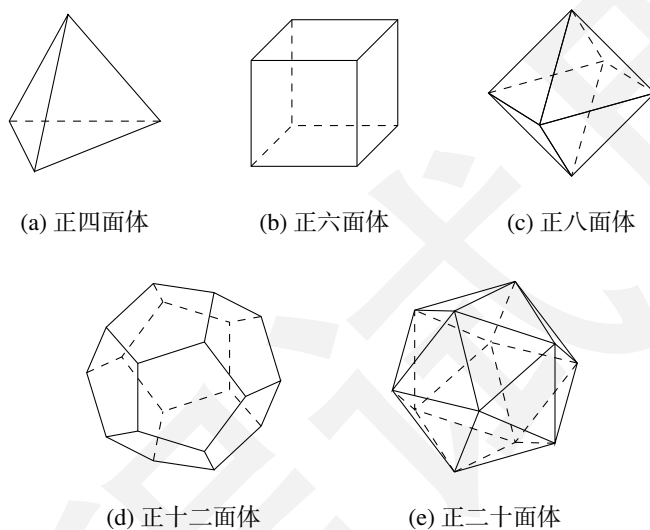


图 7.6: 五种正多面体

正多面体类型	面数	棱数	顶点数	面的形状
正四面体	4	6	4	正三角形
正六面体	6	12	8	正方形
正八面体	8	12	6	正三角形
正十二面体	12	30	20	正五边形
正二十面体	20	30	12	正三角形

表 7.1: 正多面体的基本参数

染色的对象通常可能是多面体的顶点、棱或面, 因此首先应该把握每种正多面体的点数、棱数和面数这三个基本参数, 如表 7.1 所示. 注意到, 正六面体、正八面体的面数和顶点数分别是另外一方的顶点数和面数, 且二者的棱数完全相同. 实际上, 取正六面体每个面的中心点为顶点, 在相邻的两个面的中心点间连边, 即得到一个内接于此正六面体的正八面体; 反过来对正八面体这样做, 则得到一个内接正六面体. 这样的两个多面体互称为

对偶多面体. 正十二面体、正二十面体之间也存在这种关系.

下面我们讨论每种正多面体的旋转群. 我们寻找正多面体的所有对称轴, 并考虑它绕每条对称轴旋转与自身重合时对应的置换的循环结构. 通过轨道-稳定集定理可以验证, 这种方式可以考虑到旋转群中的每一个置换.

正四面体

正四面体共有两类对称轴, 如图 7.7 所示:

- (1) AO 代表顶点到其正对面中心的连线, 正四面体绕其旋转 $\pm 120^\circ$ 可与自身重合;
- (2) EF 代表两条对棱中点的连线, 正四面体绕其旋转 180° 可与自身重合.

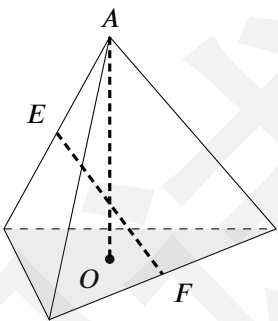


图 7.7: 正四面体的两类对称轴

正四面体共有 4 个顶点, 每个顶点均对应一条上述第一类对称轴, 绕每条对称轴旋转可产生 2 个置换; 正四面体共有 6 条棱, 每组对棱均对应一条上述第二类对称轴, 绕每条对称轴旋转产生 1 个置换. 分析顶点、棱和面分别构成的集合在每种置换下的循环结构, 可得表 7.2, 其中我们共列出了 12 个置换, 数量与正四面体旋转群的阶相同, 因此我们列出了旋转群中所有的置换.

旋转类型	顶点	棱	面	数量
不动	$(1)^4$	$(1)^6$	$(1)^4$	1
绕顶点与面心连线 $\pm 120^\circ$	$(1)^1(3)^1$	$(3)^2$	$(1)^1(3)^1$	$4 \times 2 = 8$
绕对棱中点连线 180°	$(2)^2$	$(1)^2(2)^2$	$(2)^2$	3
合计				12

表 7.2: 正四面体旋转群中各置换的循环结构

正六面体

正六面体（即正方体）共有三类对称轴, 如图 7.8 所示:

- (1) O_1O_2 代表两个对面中心的连线，正六面体绕其旋转 $\pm 90^\circ$ 或 180° 可与自身重合；
- (2) AH 代表两个对顶点的连线，即正六面体的对角线，正六面体绕其旋转 $\pm 120^\circ$ 可与自身重合；
- (3) IJ 代表两条对棱中点的连线，正六面体绕其旋转 180° 可与自身重合。

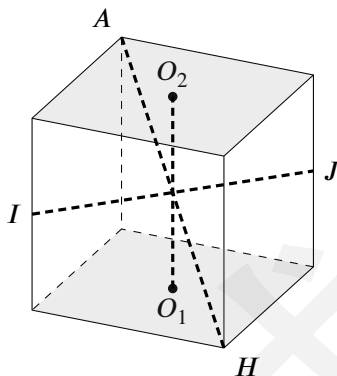


图 7.8: 正六面体的三类对称轴

分析每种旋转操作对应的置换及其循环结构，可得表 7.3，其中列出了 24 个置换。现在考察正六面体的旋转群：考虑正六面体的顶点 A ， A 可以被旋转到原先 8 个顶点所处位置中的任何一个，因此 $|E_A| = 8$ ；若保持顶点 A 的位置不变，则正六面体只能沿 AH 旋转，此旋转群即为 3 次循环群，即 $|Z_A| = |C_3| = 3$ 。根据轨道-稳定集定理，正六面体旋转群 G 的阶为

$$|G| = |E_A| |Z_A| = 8 \times 3 = 24$$

因此，表 7.3 中列出了正六面体旋转群中的全部置换。实际上，正六面体的旋转群同构于 4 次对称群 S_4 。

旋转类型	顶点	棱	面	数量
不动	$(1)^8$	$(1)^{12}$	$(1)^6$	1
绕面心连线 $\pm 90^\circ$	$(4)^2$	$(4)^3$	$(1)^2(4)^1$	$3 \times 2 = 6$
绕面心连线 180°	$(2)^4$	$(2)^6$	$(1)^2(2)^2$	3
绕体对角线 $\pm 120^\circ$	$(1)^2(3)^2$	$(3)^4$	$(3)^2$	$4 \times 2 = 8$
绕对棱中点连线 180°	$(2)^4$	$(1)^2(2)^5$	$(2)^3$	6
合计				24

表 7.3: 正六面体旋转群中各置换的循环结构

正八面体

正八面体共有三类对称轴，如图 7.9 所示：

- (1) AF 代表两个对顶点的连线，即正八面体的对角线，正八面体绕其旋转 $\pm 90^\circ$ 或 180° 可与自身重合；
- (2) O_1O_2 代表两个对面中心的连线，正八面体绕其旋转 $\pm 120^\circ$ 可与自身重合；
- (3) GH 代表两条对棱中点的连线，正八面体绕其旋转 180° 可与自身重合。

容易看出，由于正八面体和正六面体是对偶多面体，因此其对称轴之间也有对应关系：正六面体中面心连线得到的对称轴变为正八面体中对顶点连线的对称轴，反之亦然。

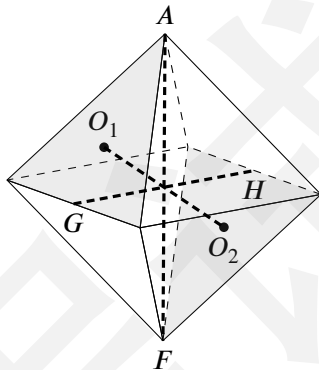


图 7.9: 正八面体的三类对称轴

分析每种旋转操作对应的置换及其循环结构，可得表 7.4，其中列出了 24 个置换。正六面体与正八面体互为对偶多面体，意味着正八面体的内接正六面体在旋转后也与其自身重合，从而二者的旋转群应当同构。因此，正八面体的旋转群中同样包含 24 个置换，并且此旋转群同构于 4 次置换群 S_4 。

旋转类型	顶点	棱	面	数量
不动	$(1)^6$	$(1)^{12}$	$(1)^8$	1
绕体对角线 $\pm 90^\circ$	$(1)^2(4)^1$	$(4)^3$	$(4)^2$	$3 \times 2 = 6$
绕体对角线 180°	$(1)^2(2)^2$	$(2)^6$	$(2)^4$	3
绕面心连线 $\pm 120^\circ$	$(3)^2$	$(3)^4$	$(1)^2(3)^2$	$4 \times 2 = 8$
绕对棱中点连线 180°	$(2)^3$	$(1)^2(2)^5$	$(2)^4$	6
合计				24

表 7.4: 正八面体旋转群中各置换的循环结构

正十二面体

正十二面体共有三类对称轴，如图 7.10 所示：

- (1) O_1O_2 代表两个对面中心的连线，正十二面体绕其旋转 $\pm 72^\circ$ 或 $\pm 144^\circ$ 可与自身重合；
- (2) AB 代表两个对顶点的连线，即正十二面体的对角线，正十二面体绕其旋转 $\pm 120^\circ$ 可与自身重合；
- (3) CD 代表两条对棱中点的连线，正十二面体绕其旋转 180° 可与自身重合。

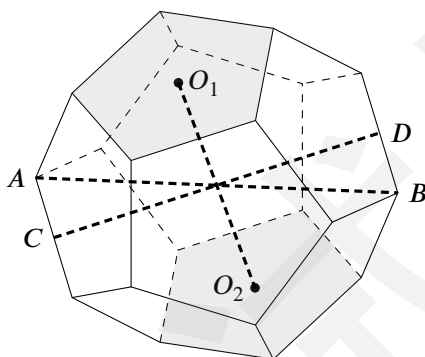


图 7.10: 正十二面体的三类对称轴

分析每种旋转操作对应的置换及其循环结构，可得表 7.5，其中列出了 60 个置换。现在考察正十二面体的旋转群：考虑正十二面体的顶点 A ， A 可以被旋转到原先 20 个顶点所处位置中的任何一个，因此 $|E_A| = 20$ ；若保持顶点 A 的位置不变，则正十二面体只能沿 AB 旋转，此旋转群即为 3 次循环群，即 $|Z_A| = |C_3| = 3$ 。根据轨道-稳定集定理，正十二面体旋转群 G 的阶为

$$|G| = |E_A| |Z_A| = 20 \times 3 = 60$$

因此，表 7.3 中列出了正十二面体旋转群中的全部置换。实际上，正十二面体的旋转群同构于 5 次交错群 A_5 。

旋转类型	顶点	棱	面	数量
不动	$(1)^{20}$	$(1)^{30}$	$(1)^{12}$	1
绕面心连线 $\pm 72^\circ, \pm 144^\circ$	$(5)^4$	$(5)^6$	$(1)^2(5)^2$	$6 \times 4 = 24$
绕体对角线 $\pm 120^\circ$	$(1)^2(3)^6$	$(3)^{10}$	$(3)^4$	$10 \times 2 = 20$
绕对棱中点连线 180°	$(2)^{10}$	$(1)^2(2)^{14}$	$(2)^6$	15
合计				60

表 7.5: 正十二面体旋转群中各置换的循环结构

正二十面体

正二十面体共有三类对称轴，如图 7.11 所示：

- (1) AB 代表两个对顶点的连线，即正二十面体的对角线，正二十面体绕其旋转 $\pm 72^\circ$ 或 $\pm 144^\circ$ 可与自身重合；
- (2) O_1O_2 代表两个对面中心的连线，正二十面体绕其旋转 $\pm 120^\circ$ 可与自身重合；
- (3) CD 代表两条对棱中点的连线，正二十面体绕其旋转 180° 可与自身重合。

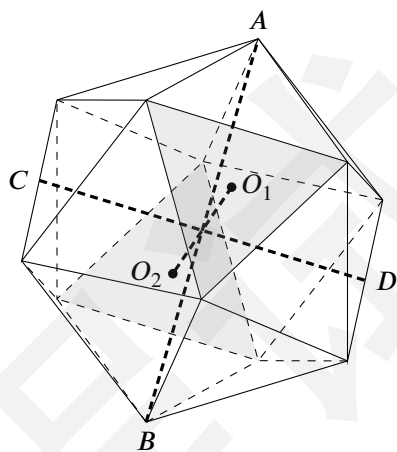


图 7.11: 正二十面体的三类对称轴

分析每种旋转操作对应的置换及其循环结构，可得表 7.6，其中列出了 24 个置换。正十二面体与正二十面体互为对偶多面体，意味着正二十面体的内接正十二面体在旋转后也与其自身重合，从而二者的旋转群应当同构。因此，正二十面体的旋转群中同样包含 60 个置换，并且此旋转群同构于 5 次交错群 A_5 。

旋转类型	顶点	棱	面	数量
不动	$(1)^{12}$	$(1)^{30}$	$(1)^{20}$	1
绕体对角线 $\pm 72^\circ, \pm 144^\circ$	$(1)^2(5)^2$	$(5)^6$	$(5)^4$	$6 \times 4 = 24$
绕面心连线 $\pm 120^\circ$	$(3)^4$	$(3)^{10}$	$(1)^2(3)^6$	$10 \times 2 = 20$
绕对棱中点连线 180°	$(2)^6$	$(1)^2(2)^{14}$	$(2)^{10}$	15
合计				60

表 7.6: 正二十面体旋转群中各置换的循环结构

其他复杂多面体

对于较复杂多面体的染色问题, 首先仍然应当把握其顶点数、棱数和面数这三个基本参数. 在三维空间中, 所有多面体的顶点数 V 、棱数 E 和面数 F 都满足欧拉公式:

$$V + F - E = 2 \quad (7-15)$$

从欧拉公式出发, 可推导出如下定理:

定理 7.29 (欠角和定理). 多面体上一个顶点的欠角定义为 360° 减去与该顶点相关的所有面角之和, 则凸多面体各顶点的欠角之和等于 720° .

以正十二面体为例, 它的每个顶点周围有三个正五边形. 正五边形的每个内角为 108° , 因此每个顶点上的欠角为

$$360^\circ - 3 \times 108^\circ = 36^\circ$$

由欠角和定理, 正十二面体的顶点数为 $\frac{720^\circ}{36^\circ} = 20$.

利用欧拉公式和欠角和定理, 我们一般就能推导出复杂多面体的基本参数. 下面我们则要求多面体的旋转群. 定理 7.28 指出, 无论多面体如何复杂, 其旋转群只可能同构于某一正棱锥、正棱柱或正多面体. 这意味着若一个多面体高度对称, 且与棱锥或棱柱不类似, 则其旋转群的阶只可能是 12、24 或 60.

例 7.5.2. 足球可以看作一个由正五边形和正六边形组成的多面体. 用若干相同的火柴 (可看作两端不相同的线段) 搭一个足球的所有棱, 求方案数.

我们首先计算足球多面体的各基本参数. 在一个经典的足球上, 每个黑色的正五边形面周围是 5 个白色的正六边形面, 每个正六边形面周围间隔排列 3 个正六边形面与 3 个正五边形面. 简单讨论可知, 此多面体的每个顶点周围均有两个正六边形和一个正五边形, 因此每个顶点上的欠角均为

$$360^\circ - 2 \times 120^\circ - 108^\circ = 12^\circ$$

由欠角和定理, 多面体的顶点数为 $\frac{720^\circ}{12^\circ} = 60$. 由于每个顶点均属于一个正五边形面和两个正六边形面, 因此正五边形面有 $\frac{60}{5} = 12$ 个, 正六边形面有 $\frac{60 \times 2}{6} = 20$ 个, 合计 32 个面. 最后, 根据欧拉公式:

$$60 + 32 - E = 2$$

我们可解得棱数 $E = 90$.

足球多面体的棱的旋转群如下:

旋转类型	循环结构	数量
不动	$(1)^{90}$	1
绕五边形面中心连线 $\pm 72^\circ$ 、 $\pm 144^\circ$	$(5)^{18}$	$6 \times 4 = 24$
绕六边形面中心连线 $\pm 120^\circ$	$(3)^{30}$	$10 \times 2 = 20$
绕相邻六边形共棱及其对棱中点连线 180°	$(1)^2(2)^{44}$	15
合计		60

利用轨道-稳定集定理, 我们可验证上表确实列出了足球多面体旋转群中所有的置换: 任取一个顶点 A , A 可以被旋转到原先 60 个顶点所处位置中的任何一个, 因此 $|E_A| = 60$; 若保持顶点 A 的位置不变, 则因为其周围是两个正六边形和一个正五边形, 因此无法再旋转, 从而有 $|Z_A| = 1$. 将二者相乘, 可知足球多面体旋转群的阶即为 $|E_A| |Z_A| = 60$.

求出旋转群中各置换的循环数目后, 我们即可使用 Pólya 计数定理. 需要注意, Pólya 计数定理的本质与 Burnside 引理相同, 是对染色方案集合在置换下的不动点进行计数. 尽管两个相邻的六边形面的共棱及其对棱中点的连线是足球的对称轴, 绕该轴旋转 180° 能使足球和旋转前的自身重合, 但因为这将导致旋转轴所经过的两条棱上的火柴反向, 因此这类置换实际上无不动点.

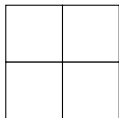
最终, 由 Pólya 计数定理, 用火柴搭出足球的不等价的方案数为

$$N = \frac{1}{60} (2^{90} + 24 \times 2^{18} + 20 \times 2^{30})$$

习题

7.1. n 个人围绕圆桌就座, 允许旋转圆桌, 用 Burnside 引理求方案数.

7.2. 设想将一个大正方形划分为四个边长为原先二分之一的小正方形, 使用黑、白两种颜色对每个小正方形染色.



允许的对称操作包括:

- (1) 大正方形绕其中心旋转;
- (2) 将染色方案中的黑色改为白色、白色改为黑色.

求不等价的染色方案数.

7.3. 在正四面体的每个面上引一条高, 允许旋转, 求不等价的方案数.

7.4. 一幅正方形肖像的边长与一个正方体的棱长相同. 将 6 幅相同的肖像贴在正方体的 6 个面上, 允许旋转, 求不等价的贴法数目.

7.5. 给定 12 种不同的彩色. 将一个经典足球的每个正六边形面染成白色, 每个正五边形面则染成给定的彩色之一, 且不同的正五边形面颜色也不同; 允许旋转, 求不等价的染色方案数.

7.6. 设想一个正八面体, 用红、蓝两色对顶点染色, 用黄、绿两色对面染色. 要求恰有 4 个顶点染为红色、恰有 4 个面染为黄色, 允许旋转, 求不等价的染色方案数.

附录 A 离散数学基础

严格来说，整个组合数学学科都可归入离散数学这个大类之下。本章将简单介绍可作为本书内容基础的离散数学法则和工具，并从中拓展出偏序集的相关知识，进而对本书在第六章中讨论的 Möbius 反演作进一步的讨论和推广。不过，本章所讨论的内容仍然较为初等，读者如有兴趣，可在一些专门教材中找到更为清晰和体系化的讨论，包括集合论公理系统、无穷偏序集的性质、以及偏序关系派生出的抽象代数领域内容等。

A.1 特殊数学运算与记号法则

本书正文中出现的绝大部分计算均遵循常见的运算法则。然而，在遇到边界情况时，本书——以至于整个离散数学学科——采用的处理方式可能与数学的其他分支不同。本节中我们介绍几种常见边界情况的处理方式。

空和与空积

本书的公式中经常出现求和或求积符号。例如，

$$\sum_{k=1}^n k^2$$

表示令变量 k 取遍大于等于 1、小于等于 n 的所有正整数，对 k 的平方求和。当 k 的取值范围是非空有限集时，求和（求积）操作的含义是显然的。然而，一些情况下， k 的取值范围是空集。例如，在上式中若令 $n = 0$ ，则没有任何正整数大于等于 1 且小于等于 n 。这时，求和（求积）即成为空和（空积）。

定义 A.1. 在复数域上，空和的值为 0，空积的值为 1。

上述定义可作如下理解：给定任何一个非空的求和（求积）符号，在其上加（乘）一个空和（空积），因为没有引入任何新的被求和（求积）元素，所以求和（求积）的结果也不应改变。因此，空和（空积）的值应当为加法（乘法）的单位元。在复数域上，加法的单位元即为 0，乘法的单位元即为 1。

0 的 0 次幂

数学分析领域常常认为 0^0 无定义, 原因是二元函数 $f(x, y) = x^y$ 在 $(0, 0)$ 处不连续且奇异. 但在离散数学中, 我们一般直接规定

$$0^0 = 1 \quad (\text{A-1})$$

上式背后的含义可从理论和实践两个角度分别理解. 在理论方面, 由于当 $x \neq 0$ 时总有 $x^0 = 1$, 因此为保证 $g(x) = x^0$ 在 $x = 0$ 处的连续性, 可以规定 $x^0 = 1$. 在实践方面, 规定 $0^0 = 1$ 可简化对一大类公式的边界条件的讨论. 例如, 考虑二项式定理的推论:

$$(1+x)^n = \sum_{k=0}^n \binom{n}{k} x^k$$

当取 $x = 0$ 时, 只有令 $0^0 = 1$ 才能使此式成立.

隐式扩展定义域

当函数的自变量取值超出预设的取值范围时, 有时可将函数值直接视为 0, 从而在一定程度上简化计算和推导. 这种情况常见于递推关系的建立过程中. 典型的例子是第二类 Stirling 数 $S(n, k)$, 它满足递推式

$$S(n, k) = S(n-1, k-1) + k \cdot S(n-1, k)$$

$S(n, k)$ 可视为一个二元函数, 它对一切非负整数 n, k 有定义, 边界条件为

$$S(n, 0) = 0 \quad (n > 0)$$

$$S(0, 0) = 1$$

若假设 n, k 超出定义域时函数值为 0, 则我们有

$$S(n, 0) = S(n-1, -1) + 0 \cdot S(n-1, 0) = 0 + 0 = 0 \quad (n > 0)$$

由此, 我们可将边界条件简化至仅含 $S(0, 0) = 1$ 这一条.

另一个常见实例是二项式系数 $\binom{n}{k}$. 若规定当 $n < 0$ 或 $k < 0$ 时视为 $\binom{n}{k} = 0$, 将有助于简化含有二项式系数的递推式推导过程中的边界条件讨论.

隐式求和范围

在部分情况下, 求和符号的下标的取值范围足够明确 (这一般意味着求和下标的范围覆盖了其定义域内的全部有效取值范围); 为了简化推导时的表达, 我们可以将求和下标的范围省略不写. 例如, 假定当 n 为正整数, 二项式系数 $\binom{n}{k}$ 仅在 k 为非负整数时有定义,

并且当 $k > n$ 时有 $\binom{n}{k} = 0$, 因此 k 的有效取值范围恰好为 $0, 1, 2, \dots, n$. 由此, 我们可将二项式定理写作

$$(1+x)^n = \sum_k \binom{n}{k} x^k$$

而无需以上下标的形式标注 k 的取值范围.

A.2 集合与偏序关系

集合是由不同对象聚集成一个整体, 该整体中的每一个对象称为集合的一个**元素**. 若对象 x 是集合 A 的元素, 则称 A 包含 x , 记作 $x \in A$, 否则记作 $x \notin A$. **空集** \emptyset 是不包含任何元素的集合, 包含至少一个元素的集合称为**非空集合**. 若集合 A 中的所有元素都是集合 B 的元素, 则称 A 是 B 的**子集**, 记作 $A \subseteq B$. 空集是任何集合的子集. 若 A 是 B 的子集, 且 B 也是 A 的子集, 则称 A 和 B **相等**, 记作 $A = B$. 给定任意集合 A , 其所有子集也构成一个集合, 称为 A 的**幂集**, 记作 $P(A)$ 或 2^A .

由有限多个对象构成的集合 A 称为**有限集**, 由无限多个对象构成的集合称为**无限集**. 对于有限集而言, 我们可以在一对大括号中列出集合中所有的元素以表示这个集合. 例如,

$$A = \{0, 1, 2, \dots, n\}$$

是由所有大于等于 0 且小于等于 n 的正整数构成的集合, 这些正整数均是 A 的元素, 而除此之外的任何对象均不是 A 的元素. 一些常见的无限集有特殊的符号表示:

- \mathbb{Z} 表示由全体整数构成的集合, 称为**整数集**;
- \mathbb{Z}^* 表示由全体非负整数构成的集合, 称为**非负整数集**; 该集合也可称为**自然数集**, 记作 \mathbb{N} ;
- \mathbb{Z}^+ 表示由全体正整数构成的集合, 称为**正整数集**;
- \mathbb{R} 表示由全体实数构成的集合, 称为**实数集**;
- \mathbb{C} 表示由全体复数构成的集合, 称为**复数集**.

这些集合之间满足子集关系 $\mathbb{Z}^+ \subseteq \mathbb{Z}^* \subseteq \mathbb{Z} \subseteq \mathbb{R} \subseteq \mathbb{C}$. 若 $Q(x)$ 是关于 x 的命题, 且集合 A 恰好包含了使 $Q(x)$ 为真的所有元素, 则可记作 $A = \{x \mid Q(x)\}$, 这种表示方式对有限集或无限集均适用.

给定两个集合 A, B , 我们可定义如下三种集合操作:

- A 和 B 的**交集**定义为由全体同时属于 A 和 B 的元素构成的集合, 记作 $A \cap B$:

$$A \cap B = \{x \mid x \in A \text{ 且 } x \in B\} \quad (\text{A-2})$$

- A 和 B 的**并集**定义为由全体属于 A 或 B 的元素构成的集合, 记作 $A \cup B$:

$$A \cup B = \{x \mid x \in A \text{ 或 } x \in B\} \quad (\text{A-3})$$

- A 和 B 的差集定义为全体属于 A 但不属于 B 的元素构成的集合, 记作 $A - B$:

$$A - B = \{x \mid x \in A \text{ 且 } x \notin B\} \quad (\text{A-4})$$

在一部分语境中, 讨论的所有集合均是某一更大集合 U 的子集, 这时若给定 $A \subseteq U$, 则 $U - A$ 又称为 A 在 U 中的补集, 记作 \bar{A} . 上述集合操作满足如下几条运算律:

- 自反律 (或称幂等律):

$$A \cap A = A$$

$$A \cup A = A$$

- 结合律:

$$(A \cap B) \cap C = A \cap (B \cap C)$$

$$(A \cup B) \cup C = A \cup (B \cup C)$$

- 分配律:

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

- 吸收律:

$$(A \cap B) \cup A = A$$

$$(A \cup B) \cap A = A$$

- De Morgan 定律:

$$A - (B \cap C) = (A - B) \cup (A - C)$$

$$A - (B \cup C) = (A - B) \cap (A - C)$$

当给定全集 U 时, De Morgan 定律又可写成如下补集形式:

$$\overline{A \cap B} = \bar{A} \cup \bar{B}$$

$$\overline{A \cup B} = \bar{A} \cap \bar{B}$$

给定非空集合 A , 若 \mathcal{A} 是由 A 的非空子集构成的集合, 且

- 对 \mathcal{A} 中的任意两个不同元素 $A_i, A_j \in \mathcal{A}$, 有 $A_i \cap A_j = \emptyset$;
- \mathcal{A} 中全部集合的并集为 A , 即 $\bigcup_{A_i \in \mathcal{A}} A_i = A$.

则 \mathcal{A} 称为对 A 的一个划分. 若 \mathcal{A} 中各元素的大小之和为 n , 则 $|A| = n$, 这实际上就是加法原理.

给定两个集合 A, B , 则全体由 A 中的一个元素与 B 中的一个元素构成的有序对也构成一个集合, 称为 A 和 B 的笛卡尔积, 记作 $A \times B$:

$$A \times B = \{(x, y) \mid x \in A, y \in B\} \quad (\text{A-5})$$

对于两个非空集合 A, B , 若存在一个对应法则 f , 使得对任意 $x \in A$ 均有唯一的 $y \in B$ 与 x 对应, 则称 f 是从 A 到 B 的一个映射, 记作 $f: A \mapsto B$. 法则 f 将 $x \in A$ 映射到 $y \in B$, 记作 $y = f(x)$. 形式化地, f 可视为 $A \times B$ 的一个子集:

$$f = \{(x, f(x)) \mid x \in A\} \quad (\text{A-6})$$

对 A 中的元素 x 与 B 中的元素 $f(x)$, $f(x)$ 称为 x 在 f 下的像, x 称为 $f(x)$ 在 f 下的原像. 对映射 $f: A \mapsto B$, 若集合 B 中的每个元素均只有至多一个原像, 则称 f 是单射; 若集合 B 中的每个元素均有原像, 则称 f 是满射. 若 f 同时是单射和满射, 则称 f 是双射.

集合中对象的数目称为它的大小或基数, 记作 $|A|$. 当 A 是有限集时, 通常使用“大小”指代 $|A|$; A 是有限集等价于 $|A|$ 是非负整数. 当 A 是无限集时, 通常使用“基数”指代 $|A|$. 若存在一个从 \mathbb{Z}^* 到 A 的双射, 则称 A 是可数无限集合, 否则称 A 是不可数集合. 若存在一个从 A 到 B 的双射, 则称 A 和 B 等势, 记作 $|A| = |B|$.

给定非空集合 A . 一个 $A \times A$ 的子集 R 称为 A 上的一个二元关系; 若 $(x, y) \in R$, 则记作 $x R y$, 否则记作 $x \not R y$. 例如, 实数间的小于等于关系 “ \leq ” 在实数集 \mathbb{R} 上定义了二元关系 $\leq_{\mathbb{R}}$:

$$\leq_{\mathbb{R}} = \{(x, y) \mid x, y \in \mathbb{R} \text{ 且 } x \leq y\}$$

一个从 $A \times A$ 到 A 的映射 f 称为 A 上的一个二元运算; 若 $f((x, y)) = z$, 则记作 $z = x f y$. 例如, 实数间的加法运算 “ $+$ ” 在实数集 \mathbb{R} 上定义了二元运算 $+_{\mathbb{R}}$:

$$x +_{\mathbb{R}} y = x + y \quad (x, y \in \mathbb{R})$$

要注意的是, 集合关系、集合运算的符号与其实际意义并没有必然的关联.

给定非空集合 A 以及其上的二元关系 R , R 有可能具有如下几种性质:

- 自反: 对任意 $x \in A$, 有 $x R x$;
- 反自反: 对任意 $x \in A$, 有 $x \not R x$;
- 对称: 对任意 $x, y \in A$, 若 $x R y$, 则 $y R x$;
- 反对称: 对任意 $x, y \in A$, 若 $x R y$ 且 $y R x$, 则 $x = y$;
- 传递: 对任意 $x, y, z \in A$, 若 $x R y$ 且 $y R z$, 则 $x R z$;
- 全关系: 对任意 $x, y \in A$, $x R y$ 和 $y R x$ 二者至少成立其一.

一些特殊的关系同时满足以上的多种性质.

定义 A.2. 设 “ \sim ” 是非空集合 A 上的二元关系, 若其是自反的、对称的和传递的, 则称 “ \sim ” 是 A 上的一个等价关系.

集合 A 上的等价关系 “ \sim ” 会将其划分为若干等价类. 形式化地, 对 A 中的任意元素 x , x 所属的等价类记为 $[x]$, 定义为

$$[x] = \{y \mid y \sim x\}$$

由集合 A 中每个元素的等价类构成的集合 \mathcal{A} 是对 A 的一个划分, 称为 A 关于 “ \sim ” 的商集, 记作 $\mathcal{A} = A/\sim$; 一般还会从 \mathcal{A} 中的每个元素 A_i 中选出一个元素 $x_i \in A_i$, 构成集合 $B = \{x_i\}$, 称为商集的代表集.

例 A.2.1. 设复数集 \mathbb{C} 上的二元关系 “ \sim ” 定义为 $a+bi \sim a+ci$ ($a, b, c \in \mathbb{R}$), 则 “ \sim ” 是 \mathbb{C} 上的等价关系, 即具有相同实部的复数彼此等价. \mathbb{C} 关于 “ \sim ” 的商集的一个代表集即为 \mathbb{R} .

定义 A.3. 设 “ \leq ” 是非空集合 A 上的二元关系, 若其是自反的、反对称的和传递的, 则称 “ \leq ” 是 A 上的一个偏序关系. 集合 A 与偏序关系 “ \leq ” 共同记作 $\langle A, \leq \rangle$, 称为偏序集. 若 “ \leq ” 同时还是一个全关系, 则称其是 A 上的一个全序关系, $\langle A, \leq \rangle$ 称为全序集.

例 A.2.2. 实数集 \mathbb{R} 上的小于等于关系 “ \leq ” 和大于等于关系 “ \geq ” 均是偏序关系. 这两个关系同时也是全序关系.

例 A.2.3. 给定非空集合 $A \subseteq \mathbb{Z}^+$, A 中元素间的整除关系 “ $|$ ” 是偏序关系.

给定偏序集 $\langle A, \leq \rangle$, 对任意 $x, y \in A$, 若 $x \leq y$ 且 $x \neq y$, 则记 $x < y$. 若 $x \leq y$, 则记 $y \geq x$; 若 $x < y$, 则记 $y > x$. 若 $x \leq y$ 不成立, 则记 $x \not\leq y$; $x \not< y$, $x \not\geq y$, $x \not> y$ 有类似的定义. 要注意的是, $x \not\leq y$ 并不必然意味着 $y \leq x$; 仅在 $\langle A, \leq \rangle$ 是全序集时, 才有 $x \not\leq y \Rightarrow y \leq x$.

定义 A.4 (偏序集的直积). 设 $\langle A, \leq_A \rangle$ 、 $\langle B, \leq_B \rangle$ 是两个偏序集, 则二者的直积定义为偏序集 $\langle A \times B, \leq \rangle$, 其中偏序关系 “ \leq ” 定义为

$$(x_1, y_1) \leq (x_2, y_2) \iff x_1 \leq_A x_2 \text{ 且 } y_1 \leq_B y_2$$

上述直积记作 $\langle A, \leq_A \rangle \times \langle B, \leq_B \rangle$.

定义 A.5 (局部有限性). 设 $\langle A, \leq \rangle$ 是偏序集, 若对任意 $x, y \in A$, 集合

$$\{z \in A \mid x \leq z \leq y\}$$

总是有限集, 则称 A 局部有限.

例如, $\langle \mathbb{Z}, \leq \rangle$ 是无限偏序集, 但它局部有限. 另一方面, $\langle \mathbb{R}, \leq \rangle$ 则不满足局部有限性. 所有有限偏序集都是局部有限的.

定义 A.6 (最大元、最小元). 设 $\langle A, \leq \rangle$ 是偏序集:

- 若存在 $x \in A$ 使得对任意 $y \in A$ 均有 $x \leq y$, 则 x 称为 A 的最小元, 记作 $x = \min A$;
- 若存在 $x \in A$ 使得对任意 $y \in A$ 均有 $y \leq x$, 则 x 称为 A 的最大元, 记作 $x = \max A$.

偏序集不一定拥有最小元或最大元, 全序集总是拥有最小元和最大元. 偏序集若有最大元 (最小元), 则一定上有限 (下有限), 但反过来则不一定.

在有最小元的局部有限偏序集 $\langle A, \leq \rangle$ 上可以定义数学归纳法. 具体地, 对任意 $x \in A$, 集合

$$D_x = \{y \in A \mid y \leq x\}$$

均是有限集. 于是, 我们可以构造映射 f , 将 $x \in A$ 映射成任意一个最高位为 1 的 $|D_x|$ 位二进制串, 再将这些二进制串看作二进制正整数, 从而得到一个定义在 A 上的函数. 这时, $f(x)$ 从小到大的顺序 (若 $f(x) = f(y)$, 则 x, y 可按任意顺序) 即是一个可行的归纳顺序:

- 基础: 注意到对任意 $x \in A$, 有 $x \leq x$, $\min A \leq x$, 因此对任意 $x \neq \min A$ 有 $f(x) \geq 2$, 而 $f(\min A) = 1$, 因此 $\min A$ 可作为数学归纳法的基础;
- 归纳: 若 $x < y$ 蕴涵 $D_x \subset D_y$, 从而必然有 $f(x) < f(y)$, 因此在对元素 x 执行归纳时, 所有满足 $y \leq x$ 的元素 y 均已处理完毕.

由此, 我们可在 $\langle A, \leq \rangle$ 上通过数学归纳法证明关于 A 中元素的命题 $P(x)$: 首先证明 $P(\min A)$ 成立, 其次证明只要 $P(y)$ 对 $y < x$ 均成立则 $P(x)$ 也成立.

定义 A.7 (上下界、上下确界). 设 $\langle A, \leq \rangle$ 是偏序集, 对它的一个子集 $B \subseteq A$:

- 若存在 $x \in A$ 使得对任意 $y \in B$ 均有 $x \leq y$, 则称 x 是 B 的下界;
- 若存在 $x \in A$ 使得对任意 $y \in B$ 均有 $y \leq x$, 则称 x 是 B 的上界;
- 若 B 的全体下界构成的集合中存在最大元, 则称其为 B 的下确界, 记作 $\inf B$;
- 若 B 的全体上界构成的集合中存在最小元, 则称其为 B 的上确界, 记作 $\sup B$;

给定 $B \subseteq A$, B 不一定拥有上界或下界; 即使 B 有上界或下界, 也不一定拥有上确界或下确界.

定义 A.8 (格). 设 $\langle A, \leq \rangle$ 是偏序集, 若对任意 $x, y \in A$, $\{x, y\}$ 均有上确界和下确界, 则称 $\langle A, \leq \rangle$ 是格.

给定集合 U , $P(U)$ 中元素间的子集关系 “ \subseteq ” 是偏序关系. 同时, 给定任意 $A, B \in P(U)$, 我们有

$$\sup \{A, B\} = A \cup B$$

$$\inf \{A, B\} = A \cap B$$

因此, $\langle P(U), \subseteq \rangle$ 是格. 考虑 $P(U)$ 的任意子集 \mathcal{X} , 我们有

$$\bigcup_{X_i \in \mathcal{X}} X_i = \sup \mathcal{X}$$

$$\bigcap_{X_i \in \mathcal{X}} X_i = \inf \mathcal{X}$$

由定义可知, 若 $\mathcal{X} = \emptyset$, 则 $P(U)$ 中的每个元素均为 \mathcal{X} 的上界和下界, 因此我们有

$$\sup \emptyset = \min P(U) = \emptyset$$

$$\inf \emptyset = \max P(U) = U$$

定义空交为零个集合的交, 空并为零个集合的并. 上述讨论表明空交的值为全集 U , 而空并的值为 \emptyset .

A.3 偏序卷积与 Möbius 反演

本书的 6.5 节中讨论了 Möbius 反演, 它可以视为容斥原理作用在有限集 X 的幂集 $P(X)$ 上的一个实例. 在上一节的讨论中, 我们已经知道 $\langle P(X), \subseteq \rangle$ 是一个偏序集. 实际上, Möbius 反演可推广至任意偏序集上, 本节我们即讨论这种推广后的 Möbius 反演.

考虑任意一个局部有限的偏序集 $\langle A, \leq \rangle$. 考虑 A 上的二元实值函数

$$f(x, y) : A \times A \mapsto \mathbb{R}$$

该函数满足若 $x \not\leq y$, 则 $f(x, y) = 0$. 记 $F(A)$ 为全体满足条件的函数构成的集合.

对任意 $f, g \in F(A)$, 定义二者的偏序卷积为

$$(f * g)(x, y) = \sum_{x \leq z \leq y} f(x, z)g(z, y) \quad (\text{A-7})$$

根据我们在 A.1 节的讨论, 等式右侧也可简写为 $\sum_z f(x, z)g(z, y)$. 容易发现, 偏序卷积运算对 $F(A)$ 封闭.

偏序卷积运算有如下三条性质. 首先, 它满足结合律. 设 $f, g, h \in F(A)$, 则

$$\begin{aligned} ((f * g) * h)(x, y) &= \sum_z (f * g)(x, z) \cdot h(z, y) \\ &= \sum_z \sum_w f(x, w)g(w, z)h(z, y) \\ &= \sum_{w, z} f(x, w)g(w, z)h(z, y) \\ &= \sum_w \sum_z f(x, w)g(w, z)h(z, y) \\ &= \sum_w f(x, w) \cdot (g * h)(w, y) \\ &= (f * (g * h))(x, y) \end{aligned}$$

其次, 偏序卷积运算拥有单位元. 考虑如下的名为 **Kronecker δ 函数**的二元函数

$$\delta(x, y) = \begin{cases} 1 & (x = y) \\ 0 & (x \neq y) \end{cases} \quad (\text{A-8})$$

该函数仅在其两个输入相等时输出 1, 否则输出 0. 注意到

$$(\delta * f)(x, y) = \sum_z \delta(x, z) f(z, y) = \delta(x, x) f(x, y) = f(x, y)$$

$$(f * \delta)(x, y) = \sum_z f(x, z) \delta(z, y) = f(x, y) \delta(y, y) = f(x, y)$$

根据定义, 显然有 $\delta \in \mathcal{F}(A)$, 这表明 δ 即为运算 “*” 的单位元. 最后, 若 $f \in \mathcal{F}(A)$ 满足

$$f(x, x) \neq 0 \quad (x \in A)$$

则 f 在偏序卷积运算下有逆元. 具体地, 对满足上述要求的 $f \in \mathcal{F}(A)$, 我们可按如下方式递归地构造函数 $g \in \mathcal{F}(A)$:

$$g(x, y) = \begin{cases} \frac{1}{f(y, y)} & (x = y) \\ -\frac{1}{f(y, y)} \sum_{x \leq z < y} g(x, z) f(z, y) & (x \neq y) \end{cases}$$

注意此处求和下标 z 的取值范围不再能省略. 显然, $(g * f)(x, x) = 1$ 对任意 $x \in A$ 成立. 当 $x \neq y$ 时, 有

$$\begin{aligned} (g * f)(x, y) &= \sum_{x \leq z \leq y} g(x, z) f(z, y) \\ &= g(x, y) f(y, y) + \sum_{x \leq z < y} g(x, z) f(z, y) \\ &= \left(- \sum_{x \leq z < y} g(x, z) f(z, y) \right) + \sum_{x \leq z < y} g(x, z) f(z, y) \\ &= 0 \end{aligned}$$

因此我们有 $g * f = \delta$, 从而 g 是 f 的左逆元. 类似地, 还可构造 f 的右逆元 h :

$$h(x, y) = \begin{cases} \frac{1}{f(x, x)} & (x = y) \\ -\frac{1}{f(x, x)} \sum_{x < z \leq y} f(x, z) h(z, y) & (x \neq y) \end{cases}$$

容易证明 $g = h$. 综上, 若 $f(x, x) \neq 0$, 则 f 在偏序卷积运算下有逆元, 记为 f^{-1} .

注意到 $\mathcal{F}(A)$ 中的函数可以自然地定义加法. 于是, $(\mathcal{F}(A), +)$ 是交换群, $(\mathcal{F}(A), *)$ 是么半群. 因此, $(\mathcal{F}(A), +, *)$ 是一个环.

我们下面考虑 $\mathcal{F}(A)$ 中一个特别的函数, 称为 A 的 ζ 函数:

$$\zeta(x, y) = \begin{cases} 1 & (x \leq y) \\ 0 & (x \not\leq y) \end{cases} \quad (\text{A-9})$$

容易看出, 该函数完整地刻画了整个偏序关系. 根据偏序关系的定义可知 $\zeta(x, x) \neq 0$ 始终成立, 从而 ζ 有逆元.

定义 A.9 (偏序集上的 Möbius 函数). 设 $\langle A, \leq \rangle$ 是局部有限的偏序集, ζ 是此偏序集的 ζ 函数. 定义 A 的 **Möbius 函数** 为 ζ 的逆元:

$$\mu = \zeta^{-1} \quad (\text{A-10})$$

根据 Möbius 函数的定义, 我们有

$$\delta(x, y) = \sum_z \mu(x, z) \zeta(z, y) = \sum_{z \leq y} \mu(x, z)$$

由此可得

$$\mu(x, y) = \begin{cases} 1 & (x = y) \\ -\sum_{z < y} \mu(x, z) & (x \neq y) \end{cases} \quad (\text{A-11})$$

下面我们利用这一性质推导几个典型偏序集的 Möbius 函数.

例 A.3.1. 设 $X = \{1, 2, \dots, n\}$, 则 $\langle P(X), \subseteq \rangle$ 是偏序集, 其 Möbius 函数为

$$\mu(A, B) = \begin{cases} (-1)^{|B|-|A|} & (A \subseteq B \subseteq X) \\ 0 & (\text{其他情况}) \end{cases}$$

证明. 对 $n = |B| - |A|$ 施加归纳. 当 $n = 0$ 时, 有 $A = B$, 此时根据定义有 $\mu(A, A) = 1$, 命题成立. 当 $n > 0$ 时, 我们有

$$\begin{aligned} \mu(A, B) &= - \sum_{A \subseteq C \subset B} \mu(A, C) \\ &= - \sum_{A \subseteq C \subset B} (-1)^{|C|-|A|} \\ &= - \sum_{k=0}^{n-1} \binom{n}{k} (-1)^k \\ &= \binom{n}{n} (-1)^n = (-1)^n \end{aligned} \quad \square$$

例 A.3.2. $\langle \mathbb{Z}^*, \leq \rangle$ 是偏序集, 其 Möbius 函数为

$$\mu(x, y) = \begin{cases} 1 & (x = y) \\ -1 & (x + 1 = y) \\ 0 & (\text{其他情况}) \end{cases}$$

且该函数与其他函数的卷积等价于定理 6.10.

证明. 对 $n = y - x$ 施加归纳. 当 $n = 0$ 时, 有 $x = y$, 此时根据定义有 $\mu(x, x) = 1$, 命题成立. 当 $n = 1$ 时, 有 $x + 1 = y$, 此时

$$\mu(x, x + 1) = - \sum_{x \leq z < x+1} \mu(x, z) = -\mu(x, x) = -1$$

命题成立. 当 $n > 1$ 时, 我们有

$$\begin{aligned} \mu(x, y) &= - \sum_{x \leq z < y} \mu(x, z) \\ &= - \sum_{k=0}^{n-1} \mu(x, x + k + 1) \\ &= -(\mu(x, x) + \mu(x, x + 1)) = 0 \end{aligned} \quad \square$$

$\langle \mathbb{Z}^*, \leq \rangle$ 是一个全序集. 由于 Möbius 函数本质上刻画的是偏序关系, 偏序集中的元素仅决定其定义域, 因此上述结论实际上适用于所有全序集. 具体地, 考虑任意全序集 $\langle A, \leq \rangle$; 对任意非最大元的 $x \in A$, 定义

$$x + 1 = y \quad (y \in A, x < y, \{z \mid x < z < y\} = \emptyset)$$

则 $\langle A, \leq \rangle$ 的 Möbius 函数将同样是这个例子中给出的形式.

例 A.3.3. 设 $\langle A, \leq_A \rangle, \langle B, \leq_B \rangle$ 是两个局部有限的偏序集, 其 Möbius 函数分别为 μ_A, μ_B , 则偏序集 $\langle A, \leq_A \rangle \times \langle B, \leq_B \rangle$ 的 Möbius 函数为

$$\mu((x_1, y_1), (x_2, y_2)) = \mu_A(x_1, x_2) \mu_B(y_1, y_2)$$

证明. 若 $(x_1, y_1) = (x_2, y_2)$, 则

$$\mu((x_1, y_1), (x_1, y_1)) = 1 = \mu_A(x_1, x_1) \mu_B(y_1, y_1)$$

命题成立. 否则, 利用数学归纳法, 我们有

$$\begin{aligned} \mu((x_1, y_1), (x_2, y_2)) &= - \sum_{(x_1, y_1) \leq (x, y) < (x_2, y_2)} \mu((x_1, x), (y_1, y)) \\ &= - \sum_{(x_1, y_1) \leq (x, y) < (x_2, y_2)} \mu_A(x_1, x) \mu_B(y_1, y) \\ &= \mu_A(x_1, y_1) \mu_B(x_2, y_2) - \sum_x \sum_y \mu_A(x_1, x) \mu_B(y_1, y) \\ &= \mu_A(x_1, y_1) \mu_B(x_2, y_2) - \left(\sum_x \mu_A(x_1, x) \right) \left(\sum_y \mu_B(y_1, y) \right) \\ &= \mu_A(x_1, y_1) \mu_B(x_2, y_2) - \delta(x_1, x_2) \delta(y_1, y_2) \\ &= \mu_A(x_1, y_1) \mu_B(x_2, y_2) \end{aligned} \quad \square$$

下面我们讨论偏序集上的 Möbius 反演，它和本书 6.5 节中介绍的数论函数的 Möbius 反演具有类似的形式. 我们首先直接给出 Möbius 反演定理，然后给出证明，然后以此为基础重新推导出数论函数的 Möbius 反演形式.

定理 A.10 (偏序集上的 Möbius 反演定理). 设 $\langle A, \leq \rangle$ 是有最小元的局部有限偏序集； $f: A \rightarrow \mathbb{R}$ 是定义在 A 上的实值函数， $g: A \rightarrow \mathbb{R}$ 是依赖于 f 的函数，定义为

$$g(x) = \sum_{y \leq x} f(y) \quad (\text{A-12})$$

则从 g 可反推出 f :

$$f(x) = \sum_{y \leq x} \mu(y, x) g(y) \quad (\text{A-13})$$

证明.

$$\begin{aligned} \sum_{y \leq x} \mu(y, x) g(y) &= \sum_y \mu(y, x) \sum_{z \leq y} f(z) \\ &= \sum_y \mu(y, x) \sum_z \zeta(z, y) f(z) \\ &= \sum_{y, z} \mu(y, x) \zeta(z, y) f(z) \\ &= \sum_z f(z) \sum_y \zeta(z, y) \mu(y, x) \\ &= \sum_z f(z) \delta(z, x) \\ &= f(x) \end{aligned}$$

□

数论函数的 Möbius 反演可看作偏序集上的 Möbius 反演的一个特例，原因是正整数间的整除关系是一种偏序关系. 下面我们作具体的推导. 设 n 是正整数，其质因数分解为

$$n = p_1^{a_1} p_2^{a_2} \cdots p_m^{a_m} \quad (a_1, a_2, \dots, a_m \geq 1)$$

考虑由 n 的全体约数构成的集合 X_n . 令

$$A_k = \{1, p_k, p_k^2, \dots, p_k^{a_k}\} \quad (k = 1, 2, \dots, m)$$

显然， X_n 可视为 A_1, A_2, \dots, A_m 的笛卡尔积，从而我们有

$$\langle X_n, | \rangle = \langle A_1, | \rangle \times \langle A_2, | \rangle \times \cdots \times \langle A_m, | \rangle$$

记 $\langle X_n, | \rangle$ 的 Möbius 函数为 μ ， $\langle A_k, | \rangle$ 的 Möbius 函数为 μ_k ($k = 1, 2, \dots, m$)，则我们就有

$$\mu(x, y) = \prod_{k=1}^m \mu_k(x, y)$$

对于任一个 A_k 而言, 其中的每个元素或者为 1, 或者只含质因数 p_k , 因此各个 $\langle A_k, | \rangle$ 均是全序集. 基于例 A.3.2 中的讨论, 我们可以直接写出它们的 Möbius 函数:

$$\mu_k(p_k^x, p_k^y) = \begin{cases} 1 & (x = y) \\ -1 & (x + 1 = y) \\ 0 & (\text{其他情况}) \end{cases} \quad (0 \leq x, y \leq a_k)$$

显然, $\mu_k(u, v)$ 的值仅和 $\frac{v}{u}$ 有关:

$$\mu_k(u, v) = \mu_k\left(1, \frac{v}{u}\right) = \begin{cases} 1 & \left(\frac{v}{u} = 1\right) \\ -1 & \left(\frac{v}{u} = p_k\right) \\ 0 & (\text{其他情况}) \end{cases} \quad (u, v \in A_k)$$

将各 μ_k 相乘即可得到 μ . 容易发现, μ 也继承了各 μ_k 的上述性质, 即仅和其两个输入的商有关:

$$\begin{aligned} \mu(u, v) &= \mu\left(1, \frac{v}{u}\right) \\ &= \begin{cases} (-1)^{\sum_{i=1}^m k_i} & \left(\frac{v}{u} = p_1^{k_1} p_2^{k_2} \cdots p_m^{k_m} \text{ 且 } k_1, k_2, \dots, k_m \in \{0, 1\}\right) \\ 0 & (\text{其他情况}) \end{cases} \end{aligned} \quad (\text{A-14})$$

记 $\mu(n) = \mu(1, n)$, 我们即得到 $\langle X_n, | \rangle$ 上的 Möbius 反演定理.

推论 A.11. 设 X_n 为 n 的全体约数构成的集合, $f: X_n \mapsto \mathbb{R}$ 是定义在 X_n 上的实值函数, $g: X_n \mapsto \mathbb{R}$ 是依赖于 f 的函数, 定义为

$$g(x) = \sum_{y|x} f(y)$$

则从 g 可反推出 f :

$$f(x) = \sum_{y|x} \mu\left(\frac{x}{y}\right) g(y)$$

为了将 f, g 的定义域从 X_n 扩展到 \mathbb{Z}^+ , 我们可采取与 6.5 节类似的做法, 令 n 取遍所有正整数. 除此以外, 我们还可先构造出质数集 P , 并对 P 中的每一个质数 p_k 构造其非负整数次幂构成的集合:

$$P_k = \{1, p_k, p_k^2, \dots, p_k^n, \dots\} \quad (k = 1, 2, 3, \dots)$$

我们指出, 正整数集与 $\{P_k\}$ 的有限前缀的笛卡尔积的并等势 (后者可以视为非负整数集的 Kleene 闭包):

$$|\mathbb{Z}^+| = \left| \bigcup_{N \in \mathbb{Z}^+} \prod_{k=1}^N P_k \right|$$

这是因为对任意有限的 N , $\prod_{k=1}^N P_k$ 中的元素均唯一对应一个正整数的质因数分解. 因此,

Möbius 函数 (A-14) 的定义域也可扩展到全体正整数.

本节的最后, 我们讨论 **Dirichlet 卷积**, 它是 $\langle \mathbb{Z}^+, | \rangle$ 上的偏序卷积的一个特例. 设 $f: \mathbb{Z}^+ \mapsto \mathbb{R}$ 是任意一元实值函数, 我们定义二元函数 $F: \mathbb{Z}^+ \times \mathbb{Z}^+ \mapsto \mathbb{R}$ 为:

$$F(x, y) = \begin{cases} f\left(\frac{y}{x}\right) & (x | y) \\ 0 & (x \nmid y) \end{cases}$$

显然, $F \in \mathcal{F}(\mathbb{Z}^+)$. 类似地, 给定 $g: \mathbb{Z}^+ \mapsto \mathbb{R}$, 我们能定义出 $G: \mathbb{Z}^+ \times \mathbb{Z}^+ \mapsto \mathbb{R}$. 现在考虑 F 和 G 的偏序卷积:

$$\begin{aligned} (F * G)(1, n) &= \sum_{1|d|n} F(1, d)G(d, n) \\ &= \sum_{d|n} f(d)g\left(\frac{n}{d}\right) \end{aligned}$$

这实际上即是一种定义域为 \mathbb{Z}^+ 的一元实值函数间的二元运算.

定义 A.12 (Dirichlet 卷积). 设 $f, g: \mathbb{Z}^+ \mapsto \mathbb{R}$, 则二者的 *Dirichlet 卷积* 定义为

$$(f * g)(n) = \sum_{d|n} f(d)g\left(\frac{n}{d}\right) \quad (\text{A-15})$$

作为偏序卷积的一个特例, Dirichlet 卷积继承了偏序卷积运算的性质, 包括:

- 结合律: $(f * g) * h = f * (g * h)$;
- 有单位元: 如下函数 ε 满足 $\varepsilon * f = f * \varepsilon = f$:

$$\varepsilon(n) = \begin{cases} 1 & (n = 1) \\ 0 & (n > 1) \end{cases}$$

- 若 $f(1) \neq 0$, 则 f 在 Dirichlet 卷积运算下有逆元:

$$f^{-1}(n) = \begin{cases} \frac{1}{f(1)} & (n = 1) \\ -\frac{1}{f(1)} \sum_{d|n, d \neq n} f^{-1}(d)f\left(\frac{n}{d}\right) & (n > 1) \end{cases}$$

除此之外, 根据定义, Dirichlet 卷积显然也满足交换律.

Dirichlet 卷积可以一种更加简明的方式解释数论函数的 Möbius 反演. 利用 Dirichlet 卷积, Möbius 反演定理可表述为:

$$g = \mathbf{1} * f \iff f = \mu * g$$

其中函数 $\mathbb{1}$ 定义为 $\mathbb{1}(n) \equiv 1$, 即一个值恒为 1 的函数. 令 $\mathbb{1}$ 的逆元为 μ , 即 $\mu * \mathbb{1} = \varepsilon$, 则我们就直接得到数论函数的 Möbius 反演定理. 具体地, 若已知 $g = \mathbb{1} * f$, 则我们有

$$\mu * g = \mu * (\mathbb{1} * f) = (\mu * \mathbb{1}) * f = \varepsilon * f = f$$

若已知 $f = \mu * g$, 则我们有

$$\mathbb{1} * f = \mathbb{1} * (\mu * g) = (\mathbb{1} * \mu) * g = \varepsilon * g = g$$

以上两式即为 Möbius 反演式. 现在考察函数 μ . 因为 $\mu * \mathbb{1} = \varepsilon$, 根据二者的定义可得

$$\sum_{d|n} \mu(d) = \begin{cases} 1 & (n=1) \\ 0 & (n>1) \end{cases}$$

此即为 6.5 节中给出的推论 6.12.

对于任意形式的卷积运算 $f * g = h$ 而言, 其思想均体现为: f, g 中的某一项系数通过特定运算贡献到结果的某一项系数. 常规的多项式乘法

$$F(x)G(x) = H(x)$$

也被称为加法卷积, F 中 x^a 项的系数和 G 中 x^b 项的系数的乘积贡献到 H 中 x^{a+b} 项的系数, a 和 b 在此处作加法运算. 在 Dirichlet 卷积

$$f(n) * g(n) = h(n)$$

中, 我们可将数论函数 f, g, h 分别视作 n 取 $1, 2, 3, \dots$ 时其值构成的数列, 则 f 的第 a 项和 g 的第 b 项的乘积贡献到 h 中的第 ab 项, a 和 b 在此处作乘法运算.

数列母函数的乘法与卷积操作等价. 根据上述讨论, 我们可以合理推测 Dirichlet 卷积也与某一类母函数的乘法操作等价. 定义数论函数 f 的 **Dirichlet 级数母函数**为

$$F_d(s) = \sum_{n=1}^{\infty} \frac{f(n)}{n^s}$$

其乘法满足 Dirichlet 卷积:

$$F_d(s)G_d(s) = \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} \frac{f(n)g(m)}{(nm)^s} = \sum_{n=1}^{\infty} \frac{(f * g)(n)}{n^s}$$

函数 $\mathbb{1}$ 的 Dirichlet 级数母函数即为著名的 Riemann ζ 函数 (它与偏序集上的 ζ 函数重名, 这并不是巧合):

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

ζ 的倒数则应当是 1 在 Dirichlet 卷积下的逆元——即 Möbius 函数——的 Dirichlet 级数母函数：

$$\frac{1}{\zeta(s)} = \sum_{n=1}^{\infty} \frac{\mu(n)}{n^s}$$

有关 Dirichlet 级数的内容涉及解析数论，已超出本书范围，不再进一步讨论。

附录 B 代数结构

本书正文第七章介绍了群；本章中，我们继续讨论群的性质，以及一种名为环的基于群的代数结构。群是一种重要的代数结构，尽管其许多性质可能与组合数学并无直接关系，但作一些深入的讨论有助于对第七章相关内容的理解。环则是母函数的重要理论基础，通过将母函数视为形式幂级数环中的元素，我们可完全回避对母函数收敛性及各种数学运算（在数学分析角度上的）合法性的讨论。读者如有兴趣，可阅读抽象代数领域的专门教材。

B.1 群

先对群的定义作简单的回顾。群指的是一个集合 G 及其上的二元运算“ \cdot ”，运算本身满足结合律， G 中包含该运算的单位元，并且 G 中所有元素在该运算下有逆元。从群出发能衍生出多种代数结构。7.1 节中已经介绍了三种和群有关的代数结构，包括原群、半群和么半群。以上三种代数结构可看作群的性质放宽后得到的结果。若对群的性质进行加强，也能得到新的代数结构。

定义 B.1. 设 (G, \cdot) 是群，若运算“ \cdot ”在 G 上满足交换律：

$$a \cdot b = b \cdot a \quad (a, b \in G)$$

则称 (G, \cdot) 是一个交换群（commutative group），或称阿贝尔群（Abelian group）。反之，若运算“ \cdot ”在 G 上不满足交换律，则称 (G, \cdot) 是一个非交换群（non-commutative group）或非阿贝尔群（non-Abelian group）。

实数加法和乘法均满足交换律，因此实数加法群和实数乘法群天然都是交换群。一些原本不满足交换律的运算可能在特定的集合 G 上满足交换律，例如 7.1 节中的例 7.1.3 介绍的二维旋转群 $SO(2)$ ，其中的运算为矩阵乘法；矩阵乘法虽然本身不满足交换律，但在 $SO(2)$ 上却是满足的：

$$\begin{aligned} \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \cos \beta & \sin \beta \\ -\sin \beta & \cos \beta \end{bmatrix} &= \begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) \\ -\sin(\alpha + \beta) & \cos(\alpha + \beta) \end{bmatrix} \\ &= \begin{bmatrix} \cos \beta & \sin \beta \\ -\sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \end{aligned}$$

例 7.1.4 中给出了一个由全体行列式为 ± 1 的三阶矩阵构成的矩阵乘法群 S ，它是一个非交换群的实例。例如，我们有如下两个行列式为 -1 的矩阵：

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \in S, B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \in S$$

这两个矩阵的积为

$$AB = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, BA = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

显然， S 上的矩阵乘法运算不满足交换律。

下面我们简单讨论由子群刻画群的性质的方法。设 G 是群， H 是 G 的子群。考虑 G 上的二元关系“ \sim ”：

$$a \sim b \iff ab^{-1} \in H$$

回顾附录 A.2 节中的相关讨论，可以注意到“ \sim ”是一个等价关系：

- (1) 自反性： $aa^{-1} = e \in H$ ，故 $a \sim a$ ；
- (2) 对称性：若 $a \sim b$ ，即 $ab^{-1} \in H$ ，则由 H 中元素有逆元可得 $ba^{-1} = (ab^{-1})^{-1} \in H$ ，故 $b \sim a$ ；
- (3) 传递性：若 $a \sim b$ ， $b \sim c$ ，即 $ab^{-1}, bc^{-1} \in H$ ，则 $ab^{-1}bc^{-1} = ac^{-1} \in H$ ，故 $a \sim c$ 。

在该等价关系下，若 $b \sim a$ ，则 $ba^{-1} \in H$ ，即 $b \in Ha$ ；由此可得 $[a] = Ha$ ，即 G 中的任意元素 a 在该等价关系下的等价类是 H 由 a 生成的右陪集。同时，我们可定义 G 关于等价关系“ \sim ”的商集 G/\sim ，称为 G 关于 H 的右商集，记作 $(G/H)_r$ ； $(G/H)_r$ 中的元素是 G 的子集，它们共同构成一个对 G 的划分

$$G = \bigcup_{a \in H} Ha$$

此划分称为 G 关于 H 的右陪集分解。

类似地，给定子群 H ，我们可以定义另一种 G 上的等价关系

$$a \sim b \iff a^{-1}b \in H$$

在该等价关系下，我们有 $[a] = aH$ ，即 G 中的任意元素 a 在该等价关系下的等价类是 H 由 a 生成的左陪集。此时， G 关于等价关系“ \sim ”的商集 G/\sim ，称为 G 关于 H 的左商集，记作 $(G/H)_l$ ，其中的元素构成对 G 的左陪集分解。

由消去律可知，对任意 $a \in G$ ，有 $|aH| = |Ha| = |H|$ 。因此， $(G/H)_r$ 、 $(G/H)_l$ 中各元素的大小均为 $|H|$ ；又因为它们均是对 G 的划分，因此必然有

$$|(G/H)_r| = |(G/H)_l|$$

我们将 $(G/H)_r$ 或 $(G/H)_l$ 的大小称为 H 在 G 中的指数, 记作 $[G:H]$. 由此我们即得到如下定理.

定理 B.2 (Lagrange 定理). 设 G 是群, H 是 G 的子群, 则

$$|G| = [G:H]|H| \quad (\text{B-1})$$

Lagrange 定理指出, 若 H 是 G 的子群, 则 H 的阶数必定是 G 的阶数的约数. 由此我们立即得到如下推论.

推论 B.3. 质数阶群是循环群.

证明. 设 G 是质数阶群, e 为其单位元. 任取 G 中一个非单位元的元素 a , 以 a 作为生成元将能得到一个循环群

$$H = \{e, a, a^2, \dots, a^{r-1}\} \quad (r \geq 2)$$

显然, $|H| \geq 2$, 并且 H 是 G 的子群. 由 Lagrange 定理, G 只有 G 和 $\{e\}$ 两个子群, 因此必然有 $H = G$, 从而 G 是循环群. \square

循环群均是交换群. 根据上述推论, 2、3、5 阶群均是交换群; 1 阶的平凡群显然是交换群, 此外简单讨论可知 4 阶群也一定是交换群. 因此, 非交换群的最小阶数为 6, 典型例子是 3 次对称群.

B.2 环与形式幂级数环

环 (ring) 是一类定义了两种二元运算的代数结构.

定义 B.4 (环). 设有非空集合 R 及其上的两个二元运算 “+” 和 “·”, 若其满足以下三条性质, 则称 R 在运算 “+”、“·” 下为一个环:

- (1) $(R, +)$ 是交换群 (单位元记作 0);
- (2) (R, \cdot) 是么半群 (单位元记作 1);
- (3) 运算 “+” 对运算 “·” 满足分配律, 即对任意的 $a, b, c \in R$, 有

$$\begin{aligned} a \cdot (b + c) &= a \cdot b + a \cdot c \\ (b + c) \cdot a &= b \cdot a + c \cdot a \end{aligned}$$

由集合 R 与二元运算 “+”、“·” 所确定的环记为 $(R, +, \cdot)$; 在不引发歧义的情况下, 可直接将环记为 R . 运算 “+” 称为 R 上的加法, 运算 “·” 称为 R 上的乘法; 元素 0 称为 R 的加法单位元, 元素 1 称为 R 的乘法单位元.

要注意的是,在上述定义中,“+”、“·”、0、1 均是符号,与数的加法、数的乘法、实数 0 和实数 1 没有必然的关联.

环的定义存在一定的分歧.少数定义认为乘法无需满足结合律和有单位元,即仅要求 (R, \cdot) 是原群,在此之上,乘法满足结合律的环被称为**结合环** (associative ring). 另有许多定义要求乘法必须满足结合律但无需拥有单位元,即要求 (R, \cdot) 是半群,在此之上,含有单位元的环被称为**幺环** (unit ring). 要指出的是,这些分歧是纯粹概念性的,无论认为 (R, \cdot) 是原群、半群或幺半群,得到的代数结构均有实际意义和研究价值. 本书采用的定义要求环上的乘法必须满足结合律且有单位元. 作为概念上的补充,在环的上述定义下,若仅要求 (R, \cdot) 是半群,得到的代数结构可称为**伪环** (rng, 读作 /rɒŋ/);

定义 B.5 (交换环). 设 $(R, +, \cdot)$ 是环,若运算 “·” 在 R 上满足交换律,则称 $(R, +, \cdot)$ 是**交换环** (commutative ring).

定义 B.6 (除环). 设 $(R, +, \cdot)$ 是环,若 $(R - \{0\}, \cdot)$ 是群,则称 $(R, +, \cdot)$ 是**除环** (division ring).

定义 B.7 (域). 设 $(R, +, \cdot)$ 是交换环和除环,则称 $(R, +, \cdot)$ 是**域** (field). 有限除环一定是域.

例 B.2.1. 考虑实数集 \mathbb{R} 、实数加法运算 “+” 和实数乘法运算 “·”. 容易验证 $(\mathbb{R}, +)$ 是交换群、 (\mathbb{R}, \cdot) 是幺半群,并且乘法对加法满足分配律. 因此, $(\mathbb{R}, +, \cdot)$ 是环. 除此之外,乘法满足交换律,并且一切非零实数均有乘法逆元,因此 $(\mathbb{R}, +, \cdot)$ 是域,称为**实数域**.

环的一个典型实例是**形式幂级数环**,它是**形式幂级数** (formal power series) 构成的环.

定义 B.8 (形式幂级数). 设 R 是环. R 上的形式幂级数 $f(x)$ 是如下形式的表达式:

$$f(x) = \sum_{n=0}^{\infty} a_n x^n = a_0 + a_1 x + a_2 x^2 + \cdots \quad (a_0, a_1, a_2, \cdots \in R)$$

其中 x 称为**不定元**,不是 R 中的元素. 若 $a_n \neq 0$ 且对所有的 $m > n$ 有 $a_m = 0$,则称 $f(x)$ 是一个**形式多项式**,其次数为 n ,记作 $\deg f(x) = n$; 定义 $\deg 0 = -\infty$.

对形式幂级数可以定义加法和乘法运算. 若

$$\begin{aligned} A(x) &= \sum_{n=0}^{\infty} a_n x^n = a_0 + a_1 x + a_2 x^2 + \cdots \\ B(x) &= \sum_{n=0}^{\infty} b_n x^n = b_0 + b_1 x + b_2 x^2 + \cdots \end{aligned}$$

为两个 R 上的形式幂级数, 则我们有

$$A(x) + B(x) = \sum_{n=0}^{\infty} (a_n + b_n)x^n \quad (\text{B-2})$$

$$A(x) \cdot B(x) = \sum_{n=0}^{\infty} \left(\sum_{k=0}^n a_k b_{n-k} \right) x^n \quad (\text{B-3})$$

定义 B.9 (形式幂级数环). 设 R 是环, $R[[x]]$ 是 R 上的全体形式幂级数构成的集合, 则 $(R[[x]], +, \cdot)$ 是环. 由于形式幂级数乘法满足交换律, 因此这这也是一个交换环.

形式幂级数环的主要意义在于为母函数提供理论基础. 例如, 考虑实数环 \mathbb{R} 上的一个普通幂级数

$$G(x) = \sum_{n=0}^{\infty} a_n x^n = a_0 + a_1 x + a_2 x^2 + \cdots$$

此级数仅在 x 取值在其收敛域内时才有意义, 这导致对 $G(x)$ 进行的任何变形均必须考虑对收敛性的影响. 但实际上, 在母函数方法中, 我们关注的一般是幂级数中各项的系数的值, 而非级数本身的价值, 级数是否收敛并不重要. 因此, 我们可将母函数视为形式幂级数, 从而完全回避 $G(x)$ 的收敛性问题.

有些情况下, 母函数必须视为形式幂级数, 我们通过如下例子展示这一点.

例 B.2.2. 设有无穷实数列 $\{a_k\}$, 其通项为

$$a_n = n!$$

其母函数为

$$G(x) = \sum_{n=0}^{\infty} n! \cdot x^n = 1 + x + 2x^2 + 6x^3 + 24x^4 + \cdots$$

当 $x \neq 0$ 时, $n! \cdot x^n$ 在 $n \rightarrow \infty$ 时无界, 所以此级数对一切 $x \neq 0$ 均不收敛. 因此, 若认为 x 是使 $G(x)$ 收敛的某一实数, 则只能有 $x = 0$.

现在, 对 $G(x)$ 作一步变形:

$$\frac{G(x) - 1}{x} = \sum_{n=0}^{\infty} (n+1)! \cdot x^n = 1 + 2x + 6x^2 + 24x^3 + \cdots$$

结果仍然是一个合法的形式幂级数, 对应于数列 $\{(n+1)!\}$. 但是, 若认为 $x = 0$, 则上述变形由于分母上出现了 x , 因此并不合法. 要解决这个问题, 唯一的方案就是规定 x 不对应任何实数值, 从而回避收敛性和运算合法性等问题.

下面我们推导 $R[[x]]$ 中的形式幂级数满足的各项运算律, 以建立母函数理论的完整基

础. 首先考虑形式幂级数上的四则运算. 设

$$A(x) = \sum_{n=0}^{\infty} a_n x^n = a_0 + a_1 x + a_2 x^2 + \cdots$$

$$B(x) = \sum_{n=0}^{\infty} b_n x^n = b_0 + b_1 x + b_2 x^2 + \cdots$$

$A(x), B(x)$ 间的加法和乘法已在 $R[[x]]$ 的定义中给出. 减法可视为加法的逆, 由于 $(R[[x]], +)$ 是群, 因此加法逆元总是存在. 具体地, $B(x)$ 的加法逆元是

$$-B(x) = \sum_{n=0}^{\infty} -b_n x^n = -b_0 + (-b_1)x + (-b_2)x^2 + \cdots$$

除法可视为乘法的逆, 由于 $(R[[x]], \cdot)$ 只是么半群, 因此乘法逆元不一定存在. $B(x)$ 有乘法逆元 $B^{-1}(x)$ 的充要条件是 $b_0 \neq 0$. 具体地, 设

$$B^{-1}(x) = \sum_{n=0}^{\infty} \tilde{b}_n x^n$$

满足 $B(x)B^{-1}(x) = 1$, 则有无穷线性方程组

$$\begin{cases} b_0 \tilde{b}_0 = 1 \\ \sum_{k=0}^n b_k \tilde{b}_{n-k} = 0 \quad (n = 1, 2, \cdots) \end{cases}$$

注意到该方程组的系数矩阵是下三角矩阵, 对角线元素均为 a_0 . 因此, 任取 $n \geq 0$, 选取系数矩阵的前 $n+1$ 行和 $n+1$ 列构成一个 $n+1$ 元线性方程组, 其系数矩阵的行列式为

$$\begin{vmatrix} b_0 & 0 & 0 & \cdots & 0 \\ b_1 & b_0 & 0 & \cdots & 0 \\ b_2 & b_1 & b_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_n & b_{n-1} & b_{n-2} & \cdots & b_0 \end{vmatrix} = b_0^{n+1} \neq 0$$

因此在 $b_0 \neq 0$ 时方程组总有唯一解, 从而我们能求解出 $B^{-1}(x)$ 的各项系数.

例 B.2.3. 设有形式幂级数

$$B(x) = 1 - x$$

令其乘法逆元为

$$B^{-1}(x) = \sum_{n=0}^{\infty} \tilde{b}_n x^n = \tilde{b}_0 + \tilde{b}_1 x + \tilde{b}_2 x^2 + \cdots$$

则有

$$(1-x)(\tilde{b}_0 + \tilde{b}_1x + \tilde{b}_2x^2 + \cdots) = 1$$

任取 $n \geq 0$, 建立关于 $\tilde{b}_0, \tilde{b}_1, \dots, \tilde{b}_n$ 的 $n+1$ 元线性方程组:

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \begin{bmatrix} \tilde{b}_0 \\ \tilde{b}_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_{n-1} \\ \tilde{b}_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

解得

$$\tilde{b}_0 = \tilde{b}_1 = \cdots = \tilde{b}_n = 1$$

因此我们有 $B^{-1}(x) = \sum_{n=0}^{\infty} x^n = 1 + x + x^2 + \cdots$.

可以反过来验证这一结论. $B^{-1}(x)$ 遵循与收敛的幂级数相同的运算法则, 因此可以在其上运用无穷等比数列求和公式:

$$B^{-1}(x) = 1 + x + x^2 + \cdots = \frac{1}{1-x}$$

结果明确显示 $B^{-1}(x)$ 是 $B(x) = 1-x$ 的乘法逆元.

有了乘法逆元后, 我们也能计算 $\frac{A(x)}{B(x)}$:

- 若 $b_0 \neq 0$, 则 $\frac{A(x)}{B(x)} = A(x)B^{-1}(x)$;
- 若 $b_0 = 0$, 则先进行如下因式分解:

$$B(x) = x^k \sum_{n=0}^{\infty} b_{n+k} x^n \quad (b_k \neq 0)$$

若 $A(x)$ 中也能提取出因式 x^k , 则可先从 $A(x)$ 、 $B(x)$ 中共同约去公因式 x^k , 之后再使用乘法逆元方法求解:

$$\frac{A(x)}{B(x)} = \frac{x^k \sum_{n=0}^{\infty} a_{n+k} x^n}{x^k \sum_{n=0}^{\infty} b_{n+k} x^n} = \left(\sum_{n=0}^{\infty} a_{n+k} x^n \right) \left(\sum_{n=0}^{\infty} b_{n+k} x^n \right)^{-1}$$

否则, $\frac{A(x)}{B(x)}$ 的展开式中将会出现 x 的负整数次幂, 从而不满足我们上面定义的形式幂级数的形式.

与普通的幂级数类似, 形式幂级数上可以定义导数和积分. 令

$$A(x) = \sum_{n=0}^{\infty} a_n x^n = a_0 + a_1 x + a_2 x^2 + \dots$$

则其形式导数和形式不定积分 (省略常数项) 分别为:

$$A'(x) = \sum_{n=1}^{\infty} n a_n x^{n-1} \quad (\text{B-4})$$

$$\int A(x) dx = \sum_{n=0}^{\infty} \frac{a_n}{n+1} x^{n+1} + C \quad (\text{B-5})$$

其中 C 是环 R 中的任意元素.

形式导数遵循我们熟悉的各项导数运算律:

$$[A(x) + B(x)]' = A'(x) + B'(x) \quad (\text{B-6})$$

$$[A(x) - B(x)]' = A'(x) - B'(x) \quad (\text{B-7})$$

$$[A(x)B(x)]' = A'(x)B(x) + A(x)B'(x) \quad (\text{B-8})$$

$$\left[\frac{A(x)}{B(x)} \right]' = \frac{A'(x)B(x) - A(x)B'(x)}{B^2(x)} \quad (\text{B-9})$$

有了形式导数的概念后, 我们可进一步定义形式幂级数的 Maclaurin 展开. 我们作如下规定:

$$A(0) = a_0 \quad (\text{B-10})$$

这一规定是符合直觉的, 并且也与收敛的幂级数的运算规律相符. 由此可得

$$A(x) = \sum_{n=0}^{\infty} \frac{A^{(n)}(0)}{n!} x^n = A(0) + A'(0)x + \frac{A''(0)}{2!} x^2 + \frac{A'''(0)}{3!} x^3 + \dots$$

Maclaurin 展开用于将分式形式的形式幂级数转化为级数形式. 例如, 考虑如下分式:

$$G(x) = \frac{1}{1-ax}$$

此式可看作我们之前讨论的 $\frac{1}{1-x}$ 展开式的一个推广. 对其作 Maclaurin 展开:

$$\begin{aligned} G(x) &= \sum_{n=0}^{\infty} \frac{x^n}{n!} \cdot \left(\frac{d^n}{dx^n} \frac{1}{1-ax} \right) \Big|_{x=0} \\ &= \sum_{n=0}^{\infty} \frac{x^n}{n!} \cdot \frac{n! \cdot a^n}{(1-x)^{n+1}} \Big|_{x=0} \\ &= \sum_{n=0}^{\infty} a^n x^n = 1 + ax + a^2 x^2 + \dots \end{aligned}$$

由此也能得到分式的展开式. 和先求分母乘法逆元、再计算分子与乘法逆元的积的过程相比, 有时这种方式会更简便.

此外, 形式不定积分还可由微积分基本定理推广为形式定积分. 具体地, 令

$$\mathcal{A}(x) = \int A(x) dx$$

则我们定义

$$\int_l^r A(x) dx = \mathcal{A}(r) - \mathcal{A}(l) \quad (\text{B-11})$$

需要注意的是, 与正常函数的定积分不同, 上述形式定积分中的 l, r 或者取 0 (此时 $\mathcal{A}(l), \mathcal{A}(r)$ 经由式 (B-10) 获得良好定义), 或者是与 x 类似的不能代表任何实数值的代数符号.

最后, 对形式幂级数可以定义极限. 给定一系列形式幂级数

$$A_1(x), A_2(x), A_3(x), \dots$$

其极限定义为

$$\lim_{i \rightarrow \infty} A_i(x) = \sum_{n=0}^{\infty} \left(\lim_{i \rightarrow \infty} [x^n] A_i(x) \right) x^n \quad (\text{B-12})$$

即, 形式幂级数的极限等于各项系数的极限组成的数列对应的形式幂级数. 若有某项系数不存在极限, 则形式幂级数的极限也不存在.

利用上述极限, 我们可通过形式多项式表示出形式幂级数:

$$\sum_{n=0}^{\infty} a_n x^n = \lim_{n \rightarrow \infty} \sum_{k=0}^n a_k x^k \quad (\text{B-13})$$

上式的意义在于建立母函数方法中加法原理的一个逻辑基础. 如果直接通过加法原理构造一个形式幂级数, 则可能产生加法原理运用于无限集上的逻辑漏洞; 若将形式幂级数视为一系列形式多项式的极限, 由于形式多项式中只含有有限多项, 因此总是可以适用加法原理, 从而也就回避了上述问题.

类似地, 我们还可定义形式多项式的无穷和与无穷积. 若 $A_1(x), A_2(x), A_3(x), \dots$ 是一列形式幂级数, 则

$$\sum_{n=1}^{\infty} A_n(x) = \lim_{n \rightarrow \infty} \sum_{k=1}^n A_k(x) \quad (\text{B-14})$$

$$\prod_{n=1}^{\infty} A_n(x) = \lim_{n \rightarrow \infty} \prod_{k=1}^n A_k(x) \quad (\text{B-15})$$

上面两式可认为是建立了母函数中加法原理和乘法原理的一个逻辑基础.