EECS 70 Discrete Mathematics and Probability Theory
Fall 2020 **Note 10**

Counting

The next major topic of the course is probability theory. Suppose you toss a fair coin a thousand times. How likely is it that you get exactly 500 heads? And what about 1000 heads? It turns out that the chances of 500 heads are roughly 2.5%, whereas the chances of 1000 heads are so infinitesimally small that we may as well say that it is impossible. But before you can learn to compute or estimate odds or probabilities you must learn to count! That is the subject of this note.

1 Counting Sequences

We start by considering a simple scenario. We pick $k \leq n$ elements from an n-element set $S = \{1, 2, \ldots, n\}$ one at a time while removing the sampled element from S. This type of sampling is called *sampling without replacement*. We wish to count the number of different ways to do this, taking into account the order in which the elements are picked. For example, when $k = 2$, picking 1 and then 2 is considered a different outcome from picking 2 followed by 1. Another way to ask the question is this: we wish to form an ordered sequence of k distinct elements, where each element is picked from the set S. How many different such ordered sequences are there?

If we were dealing cards, the set would be $S = \{1, \ldots, 52\}$, where each number represents a card in a deck of 52 cards. Picking an element of S in this case refers to dealing one card. Note that once a card is dealt, it is no longer in the deck and so it cannot be dealt again. So the hand of k cards that are dealt consists of k distinct elements from the set S.

For the first card, it is easy to see that we have 52 distinct choices. But now the available choices for the second card depend upon what card we picked first. The crucial observation is that regardless of which card we picked first, there are exactly 51 choices for the second card. So the total number of ways of choosing the first two cards is 52×51. Reasoning in the same way, there are exactly 50 choices for the third card, regardless of our choices for the first two cards. It follows that there are exactly $52 \times 51 \times 50$ sequences of three cards. In general, the number of sequences of k cards is $52 \times 51 \times \cdots \times [52 - (k-1)]$.

This is an example of the First Rule of Counting:

First Rule of Counting: If an object can be made by a succession of k choices, where there are n_1 ways of making the first choice, and *for every* way of making the first choice there are n_2 ways of making the second choice, and *for every* way of making the first and second choice there are n_3 ways of making the third choice, and so on up to the n_k-th choice, then the total number of distinct objects that can be made in this way is the product $n_1 \times n_2 \times n_3 \times \cdots \times n_k$.

Here is another way of picturing the First Rule of Counting. Consider the following tree:

计数

本课程的下一个主要主题是概率论。假设你将一枚公平的硬币抛掷一千次。恰好得到500个正面的可能性有多大？那1000个正面呢？结果表明，得到500个正面的概率大约是2.5%，而得到1000个正面的概率小到我们可以认为它几乎不可能发生。但在你学会计算或估计概率之前，你必须先学会计数！这就是本讲义的主题。

1 计算序列

我们从一个简单的场景开始。我们从一个n元素集合$S = \{1,2,\ldots,n\}$中一次选取$k \leq n$个元素，并在抽样后将已选元素从S中移除。这种抽样方式称为不可重复抽样。我们希望计算这样做的不同方式数量，同时考虑元素被选取的顺序。例如，当$k = 2$时，先选1再选2被视为与先选2再选1不同的结果。另一种提问方式是：我们希望形成一个由k个不同元素组成的有序序列，每个元素都从集合S中选取。有多少种不同的这样的有序序列？

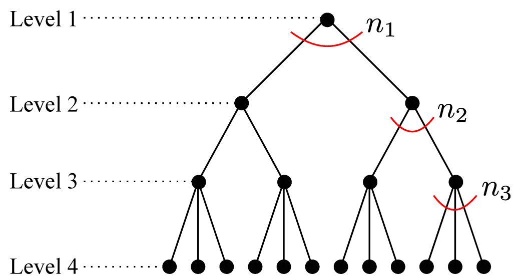
如果我们是在发牌，集合就是$S = \{1,\ldots,52\}$，其中每个数字代表一副52张牌中的一张牌。在这种情况下，从S中选择一个元素意味着发一张牌。注意一旦一张牌被发出，它就不再在牌堆中，因此不能再被发出。所以发出的k张牌组成的手牌是由集合S中k个不同元素构成的。

对于第一张牌，很容易看出我们有52种不同的选择。但现在第二张牌的可用选择取决于我们第一张选的是哪张牌。关键的观察是，无论我们第一张选的是什么牌，第二张牌都有恰好51种选择。因此选择前两张牌的总方式数为52×51。用同样的方式推理，无论前两张牌如何选择，第三张牌都有恰好50种选择。因此，三张牌序列的总数恰好是$52\times 51\times 50$。一般来说，k张牌序列的数量为$52\times 51\times\cdots\times[52-(k-1)]$。

这是计数第一法则的一个例子：

计数第一法则：如果一个对象可以通过连续k次选择构造而成，第一次选择有n_1种方式，对于每一种第一次选择的方式第二次选择有n_2种方式，对于每一种前两次选择的方式第三次选择有n_3种方式，依此类推直到第n_k次选择，那么可以通过这种方式构造的不同对象总数为乘积$n_1 \times n_2 \times n_3 \times\cdots\times n_k$。

以下是计数第一法则的另一种理解方式。考虑下面这棵树：



It has branching factor n_1 at the root, n_2 at every vertex at the second level,..., n_k at every vertex at the k-th level. Each path from the root to a vertex at level $k+1$ (i.e., a leaf vertex) represents one possible way of making the object by making a succession of k choices. So the number of distinct objects that can be made is equal to the number of leaves in the tree. Moreover, the number of leaves in the tree is the product $n_1 \times n_2 \times n_3 \times \cdots \times n_k$. For example, if $n_1 = 2$, $n_2 = 2$, and $n_3 = 3$, then there are 12 leaves (i.e., outcomes).

2 Counting Sets

Consider a slightly different question. We would like to pick k distinct elements of $S = \{1, 2, \dots, n\}$ (i.e., without repetition), but we do not care about the order in which we picked the k elements. For example, picking elements $1, \dots, k$ is considered the same outcome as picking elements $2, \dots, k$ and picking 1 as the last (k-th element). Now how many ways are there to choose these elements to obtain the same outcome?

When dealing a hand of cards, say a poker hand, it is often more natural to count the number of distinct hands (i.e., the set of 5 cards dealt in the hand), rather than the order in which they were dealt. As we have seen in Section 1, if we are considering order, there are $52 \cdot 51 \cdot 50 \cdot 49 \cdot 48 = \frac{52!}{47!}$ outcomes. But how many distinct hands of 5 cards are there? Here is another way of asking the question: each such 5 card hand is just a subset of S of cardinality 5. So we are asking how many 5 element subsets of S are there?

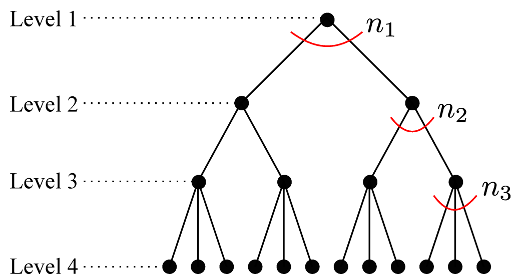
Here is a clever trick for counting the number of distinct subsets of S with exactly 5 elements. Create a bin corresponding to each such 5 element subset. Now take all the sequences of 5 cards and distribute them into these bins in the natural way: each sequence gets placed in the bin corresponding to the set of 5 elements in the sequence. Thus, if the sequence is $(2, 1, 3, 5, 4)$, then it is placed in the bin labeled $\{1, 2, 3, 4, 5\}$. How many sequences are placed in each bin? The answer is exactly 5!, since there are exactly 5! different ways to order 5 cards.

Recall that our goal was to compute the number of 5 element subsets, which now corresponds to the number of bins. We know that there are $\frac{52!}{47!}$ distinct 5-card sequences, and there are 5! sequences placed in each bin. The total number of bins is therefore $\frac{52!}{47!5!}$.

This quantity $\frac{n!}{(n-k)!k!}$ is used so often that there is special notation for it: $\binom{n}{k}$, pronounced "*n choose k*," and it's known as the *binomial coefficient*. This is the number of ways of picking k distinct elements from S, where the order of placement does not matter. Equivalently, it's the number of ways of choosing k objects out of a total of n distinct objects, where the order of the choices does not matter.

The trick we used above is actually our Second Rule of Counting:

Second Rule of Counting: Assume an object is made by a succession of choices, and the order in which the choices are made does not matter. Let A be the set of ordered objects and let B be the set of unordered objects. If there exists an m-to-1 function f from A to B, we can count the number of ordered objects (pretending that the order matters) and divide by m (the number of ordered objects per unordered objects) to obtain $|B|$, the number of unordered objects.



它在根节点的分支因子为n1，在第二层每个顶点为n2，……，在第k层每个顶点为nk。从根节点到第k+1层的一个顶点（即叶节点）的每条路径代表通过连续k次选择构造对象的一种可能方式。因此，可以构造的不同对象数量等于树中叶子的数量。此外，树中叶子的数量等于乘积n1 ×n2 ×n3 ×· · ·×nk。例如，如果n1 = 2，n2 = 2，且n3 = 3，则有12片叶子（即结果）。

2 计算集合

考虑一个略有不同的问题。我们想从S = {1,2,...,n}中挑选k个不同的元素（即不重复），但我们并不关心挑选k个元素的顺序。例如，挑选元素1,...,k被视为与先挑选2,...,k再最后挑选1（第k个元素）相同的结果。现在有多少种方式可以选择这些元素以获得相同的结果？在发牌时，比如一手扑克牌，通常更自然的是计算不同手牌的数量（即手中5张牌的集合），而不是它们被发出的顺序。

正如我们在第1节中看到的，如果我们考虑顺序，会有52·51·50·49·48 = 52!47!种结果。但到底有多少种不同的5张牌手牌？还有另一种提问方式：每种这样的5张牌手牌只是S的一个基数为5的子集。所以我们问的是S中有多少个5元素子集？

这是一个计算S中恰好包含5个元素的不同子集数量的巧妙方法。为每一个这样的5元素子集创建一个桶。现在取所有5张牌的序列，并将它们自然地分配到这些桶中：每个序列放入与其序列中5个元素集合相对应的桶中。因此，如果序列为(2,1,3,5,4)，则它被放入标有{1,2,3,4,5}的桶中。每个桶中有多少个序列？答案正好是5!，因为恰好有5!种不同的方式来排列5张牌。

回想一下，我们的目标是计算恰好包含5个元素的子集数量，现在这对应于桶的数量。我们知道有52!

47! 不同的5张牌序列，每个桶中有5!个序列。

因此桶的总数是52!

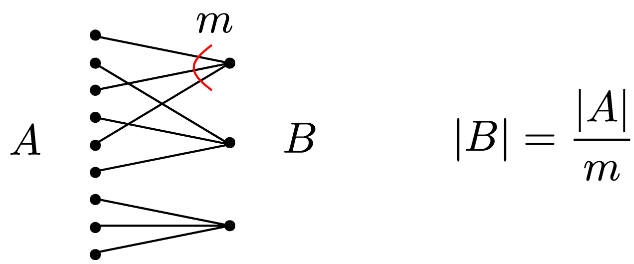
47!5!·

这个量n! $(n-k)!k!$ 因其频繁使用而有专门的记号：n $\binom{}{k}$,读作n选k,以及它被称为二项式系数。这是从S中选取k个不同元素的方式数量，其中放置顺序不重要。等价地说，它是在总共n个不同对象中选择k个对象的方式数量，其中选择的顺序不重要。

我们上面使用的技巧实际上就是计数第二法则：

计数第二法则：假设一个对象通过一系列选择构成，且这些选择的顺序无关紧要。设A为有序对象的集合，B为无序对象的集合。如果存在一个从A到B的m对1函数f，则我们可以先计算有序对象的数量（假设顺序重要），然后除以m（每个无序对象对应的有序对象数量），从而得到|B|，即无序对象的数量。

Note that we are assuming that the number of ordered objects is the same for every unordered object; the rule cannot be applied otherwise. Here is another way of picturing the Second Rule of Counting:



The function f simply places the ordered outcomes into bins corresponding to the unordered outcomes. In our poker hand example, f will map $5!$ elements in the domain of the function (the set of ordered 5 card outcomes) to one element in the range (the set of 5-element subsets of S). The number of elements in the range of the function is therefore $\frac{52!}{47!5!}$.

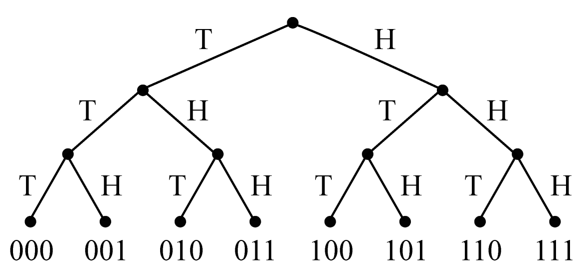
3 Sampling with Replacement

Sometimes we wish to consider a different scenario of sampling where after we choose an element from $S = \{1, 2, \ldots, n\}$, we throw it back into S so that we can choose it again. Assume that we are still picking k elements out of S one at a time and that order matters. How many distinct sequences of k elements can we obtain under this new sampling scheme? We can use the First Rule of Counting. Since we have n choices in each trial, $n_1 = n_2 = \cdots = n_k = n$. Then we have a grand total of n^k sequences.

This type of sampling is called *sampling with replacement*; multiple trials can have the same outcome. Card dealing is a type of *sampling without replacement*, since two trials cannot both have the same outcome (one card cannot be dealt twice).

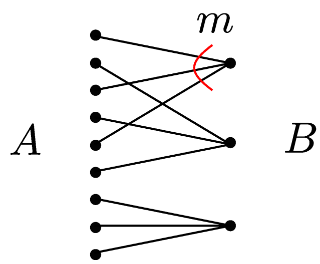
3.1 Coin Tosses

Let us return to coin tosses. How many different outcomes are there if we toss a coin k times? By an outcome, we mean a length-k sequence consisting of heads and tails. Suppose we represent each outcome by a binary string of length k where 0 corresponds to a tail (T) and 1 a head (H). For example, the string 001 would represent an outcome of 2 tails followed by a head. The following tree, in which the leaves are in 1-to-1 correspondence with the set of possible outcomes, illustrates the experiment for $k = 3$:



Here $S = \{0, 1\}$. This is a case of sampling with replacement; multiple coin flips could result in tails (we could pick the element 0 from the set S in multiple trials). Order also matters; e.g., strings 001 and 010 are considered different outcomes. By the reasoning above (using the First Rule of Counting) we have a total of $n^k = 2^k$ distinct outcomes, since $n = 2$.

请注意，我们假设每个无序对象对应的有序对象数量是相同的；否则该规则无法应用。以下是计数第二法则的另一种理解方式：



$$|B| = \frac{|A|}{m}$$

函数f只是将有序结果放入对应的无序结果的桶中。在我们的扑克手牌例子中，f会将函数定义域中的5!个元素（有序的5张牌结果集合）映射到值域中的一个元素（S的5元素子集集合）。因此函数值域中的元素数量是52!

$\overline{47!5!}\cdot$

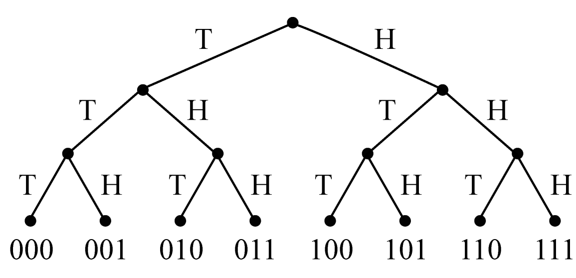
3 可重复抽样

有时我们考虑另一种抽样场景：当我们从S = {1,2,...,n}中选择一个元素后，将其放回S中以便可以再次选择。假设我们仍然每次从S中逐个选取k个元素，并且顺序重要。在这种新的抽样方案下，我们可以得到多少个不同的k元素序列？我们可以使用计数第一法则。由于每次试验都有n种选择，n1 = n2 = · · ·= nk = n。因此，我们总共可以得到nk个序列。

这种抽样称为可重复抽样；多次试验可能出现相同的结果。发牌是一种不可重复抽样，因为两次试验不能有相同的结果（一张牌不能被发两次）。

3.1 抛硬币

让我们回到抛硬币的问题。如果我们抛一枚硬币k次，会有多少种不同的结果？所谓结果，是指由正面和反面组成的长度为k的序列。假设我们用长度为k的二进制字符串表示每个结果，其中0对应反面（T），1对应正面（H）。例如，字符串001将表示两次反面后接一次正面的结果。以下这棵树的叶子与k = 3时所有可能结果的集合一一对应，说明了这个实验：



这里S = {0,1}。这是一种可重复抽样情况；多次抛硬币可能都出现反面（我们可以在多次试验中从集合S中选择元素0）。顺序也重要；例如，字符串001和010被视为不同的结果。根据上述推理（使用计数第一法则），我们总共有nk = 2k个不同的结果，因为n = 2。

3.2 Rolling Dice

Let's say we roll two dice, so $k = 2$ and $S = \{1, 2, 3, 4, 5, 6\}$. How many possible outcomes are there? In this setting, ordering matters; obtaining 1 with the first die and 2 with the second is different from obtaining 2 with the first and 1 with the second. We are sampling with replacement, since we can obtain the same result on both dice.

The setting is therefore the same as the coin flipping example above (order matters and we are sampling with replacement), so we can use the First Rule of Counting in the same manner. The number of distinct outcomes is therefore $n^2 = 6^2 = 36$.

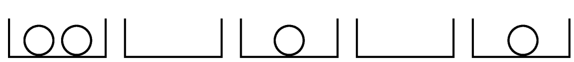
3.3 Sampling with Replacement, but where Order Does Not Matter

Say you have unlimited quantities of apples, bananas and oranges. You want to select 5 pieces of fruit to make a fruit salad. How many ways are there to do this? In this example, $S = \{1, 2, 3\}$, where 1 represents apples, 2 represents bananas, and 3 represents oranges. $k = 5$ since we wish to select 5 pieces of fruit. Ordering does not matter; selecting an apple followed by a banana will lead to the same salad as a banana followed by an apple.

This scenario is trickier to analyze. It may seem natural to apply the Second Rule of Counting because order does not matter. Let's consider this method. We first pretend that order matters and observe the number of ordered objects is 3^5 as discussed above. How many ordered options are there for every unordered option? The problem is that this number differs depending on which unordered object we are considering. Let's say the unordered object is an outcome with 5 bananas. There is only one such ordered outcome. But if we are considering 4 bananas and 1 apple, there are 5 such ordered outcomes (represented as 12222, 21222, 22122, 22212, 22221).

Now that we see the Second Rule of Counting will not help, can we look at this problem in a different way? Let us first generalize back to our original setting: we have a set $S = \{1, 2, \dots, n\}$ and we would like to know how many ways there are to choose multisets (sets with repetition) of size k. Remarkably, we can model this problem in terms of binary strings, as described below.

Assume we have one bin for each element of S, so n bins in total. For example, if we selected 2 apples and 1 banana, bin 1 would have 2 elements and bin 2 would have 1 element. In order to count the number of multisets, we need to count how many different ways there are to fill these bins with k elements. We don't care about the order of the bins themselves, just how many of the k elements each bin contains. Let's represent each of the k elements by a 0 in the binary string, and separations between bins by a 1. The following picture illustrates an example of placement where $S = \{1, \dots, 5\}$ and $k = 4$:



00110110

Counting the number of multisets is now equivalent to counting the number of placements of the k 0's. We have just reduced what seemed like a very complex problem to a question about a binary string, simply by looking at it from a different perspective!

How many ways can we choose these locations? The length of our binary string is $k + n - 1$, and we are choosing which k locations should contain 0's. The remaining $n - 1$ locations will contain 1's. Once we pick a location for one zero, we cannot pick it again; repetition is not allowed. Picking location 1 followed by location 2 is the same as picking location 2 followed by location 1, so ordering does not matter. It follows

3.2 掷骰子

假设我们掷两个骰子，因此k = 2且S = {1,2,3,4,5,6}。可能的结果有多少种？在这种情况下，顺序是重要的；第一个骰子得到1而第二个得到2，与第一个得到2而第二个得到1是不同的结果。我们是在可重复抽样的条件下进行的，因为我们可能在两个骰子上都得到相同的结果。

因此，该设定与上面的抛硬币例子相同（顺序重要且为可重复抽样），我们可以以同样的方式使用计数第一法则。不同的结果总数为n2 = 62 = 36。

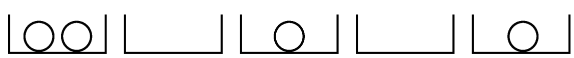
3.3 可重复抽样但顺序无关的情况

假设你有无限数量的苹果、香蕉和橙子。你想从中选出5个水果来制作一份水果沙拉。有多少种选择方式？在这个例子中，S = {1,2,3}，其中1代表苹果，2代表香蕉，3代表橙子。k = 5，因为我们希望选择5个水果。顺序不重要；先选一个苹果再选一个香蕉，与先选香蕉再选苹果得到的沙拉是一样的。

这种情况分析起来更复杂。可能人们会自然想到使用计数第二法则，因为顺序无关。让我们尝试这种方法。我们先假装顺序重要，并注意到有序结果的数量如上所述为35。但每一个无序选项对应多少个有序选项呢？问题是这个数量取决于我们考虑的是哪个无序对象。比如，若无序对象是5根香蕉的结果，则只有一种有序结果。但如果考虑的是4根香蕉和1个苹果，则有5种有序结果（表示为12222, 21222, 22122, 22212, 22221）。

现在我们看到计数第二法则无法适用，能否从另一个角度看待这个问题？让我们先回到原始设定：我们有一个集合S = {1,2,...,n}，想知道从中选择大小为k的多重集（允许重复的集合）有多少种方式。值得注意的是，我们可以将此问题建模为二进制字符串问题，如下所述。

假设S的每个元素都有一个对应的桶，总共n个桶。例如，如果我们选了2个苹果和1个香蕉，桶1中有2个元素，桶2中有1个元素。为了计算多重集的数量，我们需要计算有多少种不同的方式将k个元素分配到这些桶中。我们不关心桶本身的顺序，只关心每个桶包含多少个k元素。让我们用二进制字符串中的0表示每个k元素，用1表示桶之间的分隔符。下图说明了一个例子，其中S = {1,...,5}且k = 4：



00110110

现在计算多重集的数量等价于计算k个0的放置方式数。我们仅仅通过换一种视角，就把一个看似复杂的问题简化成了关于二进制字符串的问题！

有多少种方式可以选择这些位置？我们的二进制字符串长度为k +n−1，我们要选择其中k个位置放置0。剩下的n−1个位置将放置1。一旦我们选定了某个0的位置，就不能再次选择；不允许重复。先选位置1再选位置2与先选位置2再选位置1是相同的，因此顺序无关。由此得出

that all we wish to compute is the number of ways of picking k elements from $k+n-1$ elements, without replacement and where the order of placement does not matter. This is given by $\binom{n+k-1}{k}$, as discussed in Section 2. This is therefore the number of ways in which we can choose multisets of size k from set S.

Returning to our example above, the number of ways of picking 5 pieces of fruit is exactly $\binom{3+5-1}{5} = \binom{7}{5}$.

Notice that we started with a problem which seemed very different from previous examples, but, by viewing it from a certain perspective, we were able to use previous techniques (those used in counting sets) to find a solution!

In a sense, we have found the zeroth rule of counting:

Zeroth Rule of Counting: If a set A can be placed into a one-to-one correspondance with a set B (i.e. you can find a bijection between the two — an invertible pair of maps that map elements of A to elements of B and vice-versa), then $|A| = |B|$.

This is the very heart of what it means to count and is key to many combinatorial arguments as we will explore further in the next section.

4 Combinatorial Proofs

Combinatorial proofs are interesting because they rely on intuitive counting arguments rather than tedious algebraic manipulation. They often feel like proofs by stories — the same story, told from multiple points of view.

As a warmup, let's prove the Binomial Theorem using counting arguments:

Theorem 10.1 (Binomial Theorem)**.** *For all $n \in \mathbb{N}$,*

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

Proof. The left-hand side can be written as the n-fold product $(a+b)(a+b)\cdots(a+b)$. Label these n factors as $f_1, f_2, \ldots, f_n$. When $(a+b)(a+b)\cdots(a+b)$ is expanded out via multiplication, it will result in a sum over 2^n monomials, and to each monomial in the sum, each factor f_i contributes either an a or a b. For example, in $(a+b)(a+b) = a^2+ab+ba+b^2$, a^2 is formed by multiplying a from f_1 and a from f_2; ab is formed by multiplying a from f_1 and b from f_2; ba is formed by multiplying b from f_1 and a from f_2; b^2 is formed by multiplying b from f_1 and b from f_2. In general, the monomial $a^k b^{n-k}$ is obtained when k copies of a's and $n-k$ copies of b's are multiplied. There are exactly $\binom{n}{k}$ ways to choose k copies of a's and $n-k$ copies of b's from the n factors $f_1, f_2, \ldots, f_n$. □

This well-known result has numerous applications. For example, the following combinatorial identity can be easily derived using Theorem 10.1:

Corollary 10.1. *For all $n \in \mathbb{N}$,*

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

Proof. Set $a = -1$ and $b = 1$ in Theorem 10.1. □

我们只需计算从k +n−1个元素中无放回地选取k个元素的方式数，且顺序无关。这由n+k−1
$\binom{}{k}$，如
第2节。因此，这正是我们可以从集合S中选择大小为k的多重集的方式数。回到上面的例子，
选取5个水果的方式数正好是3+5−1 $\binom{}{5} = \binom{7}{5}$.
注意到我们开始的问题似乎与之前的例子完全不同，但通过特定视角观察后，我们能够利用之前的技术（如集合计数的方法）找到解决方案！

在某种意义上，我们发现了计数的第零法则：

计数第零法则：如果集合A可以与集合B建立一一对应关系（即能找到两者之间的双射——一组可逆的映射，将A的元素映射到B，反之亦然），则|A| = |B|。

这正是计数的本质所在，也是我们在下一节将探讨的许多组合论证的关键。

4 组合证明

组合证明很有趣，因为它们依赖于直观的计数推理而不是繁琐的代数操作。它们常常像讲故事一样——同一个故事，从多个角度讲述。

作为热身，让我们用计数方法证明二项式定理：定理10.1（二项式定

理）。对于所有n ∈N，

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} \text{akbn−k。}$$

证明。左边可以写成n重积(a+b)(a+b)···(a+b)。将这n个因子
记作f1, f2,..., fn。当(a+b)(a+b)···(a+b)通过乘法展开时，会产生2n个单项式的和，每个单项式中，每个因子fi贡献一个a或一个b。例如，在(a+b)(a+b) = a2 +ab+ba+b2中，a2由f1中的a和f2中的a相乘得到；ab由f1中的a和f2中的b相乘得到；ba由f1中的b和f2中的a相乘得到；b2由f1中的b和f2中的b相乘得到。一般情况下，单项式akbn−k出现在选择了k个a和n−k个b进行相乘时。从n个因子f1, f2,..., fn中恰好有nk种方式选择k个a和n−k个b。
( )

□

这个众所周知的结果有许多应用。例如，下面的组合恒等式可以很容易地通过定理10.1推导出来：

推论10.1。对于所有n ∈N，

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

证明。在定理10.1中令a = −1且b = 1。 □

In what follows, we will consider more sophisticated versions of combinatorial proofs. The key idea is to show that the two sides of a given equation correspond to two different ways to count the same kind of objects. In more advanced settings, the goal is to establish a bijection between the set of objects counted on the left-hand side of an identity with the set of objects counted on the right-hand side. We illustrate these concepts below.

4.1 Combinatorial Identities

Using combinatorial arguments, we can prove complex facts, such as

$$\binom{n}{k+1} = \binom{n-1}{k} + \binom{n-2}{k} + \cdots + \binom{k}{k}, \tag{1}$$

which is known as the *hockey-stick identity*. You can directly verify this identity by algebraic manipulation. But you can also do this by interpreting what each side means as a combinatorial process. The left-hand side of (1) is just the number of ways of choosing a subset X with $k+1$ elements from a set $S = \{1, \ldots, n\}$. Let us think about a different process that results in the choice of a subset with $k+1$ elements. We start by picking the lowest-numbered element of X. If we picked 1, to finish constructing X, we must now choose k elements out of the remaining $n-1$ elements of S greater than 1; this can be done in $\binom{n-1}{k}$ ways. If instead the lowest-numbered element we picked is 2, then we have to choose k elements from the remaining $n-2$ elements of S greater than 2, which can be done in $\binom{n-2}{k}$ ways. Moreover all these subsets are distinct from those where the lowest-numbered element was 1. So we should add the number of ways of choosing each to the grand total. Proceeding in this way, we split up the process into cases according to the first (i.e., lowest-numbered) object we select, to obtain:

$$\textbf{First element selected is} \begin{cases} \text{element } 1, & \text{leading to } \binom{n-1}{k} \text{ remaining ways,} \\ \text{element } 2, & \text{leading to } \binom{n-2}{k} \text{ remaining ways,} \\ \text{element } 3, & \text{leading to } \binom{n-3}{k} \text{ remaining ways,} \\ \quad\vdots & \qquad\qquad\vdots \\ \text{element } (n-k), & \text{leading to } \binom{k}{k} \text{ remaining ways.} \end{cases}$$

These factors correspond to the terms on the right-hand side of (1). (Note that the lowest-numbered object we select cannot be higher than $n-k$ as we have to select $k+1$ distinct objects.)

Another combinatorial identity we will prove is

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = 2^n. \tag{2}$$

This identity actually follows immediately from setting $a = 1$ and $b = 1$ in the Binomial Theorem above, but we will provide a more direct combinatorial proof. Suppose we have a set S with n distinct elements. On the left-hand side of (2), $\binom{n}{i}$ counts the number of ways of choosing a subset of S of size exactly i; so the sum on the left-hand side counts the total number of subsets (of any size) of S.

We claim that the right-hand side of (2) does indeed also count the total number of subsets. To see this, just identify a subset with an n-bit vector, where in each position j we put a 1 if the j-th element is in the subset, and a 0 otherwise. So the number of subsets is equal to the number of n-bit vectors, which is 2^n (there are 2 options for each bit).

Let us look at an example, where $S = \{1,2,3\}$ (so $n = 3$). Enumerate all $2^3 = 8$ possible subsets of S: $\varnothing, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}$. The eight 3-bit vectors corresponding to these subsets are

接下来，我们将考虑更复杂的组合证明形式。关键思想是说明给定等式的两边对应于对同一类对象的两种不同计数方式。在更高级的情况下，目标是在恒等式左边所计数的对象集合与右边所计数的对象集合之间建立一个双射。我们将在下文说明这些概念。

4.1 组合恒等式

使用组合论证，我们可以证明一些复杂的结论，例如

$$\binom{n}{k+1} = \binom{n-1}{k} + \binom{n-2}{k} + \cdots + \binom{k}{k}, \tag{1}$$

这个恒等式被称为冰球杆恒等式。你可以通过代数运算直接验证该恒等式，但也可以通过组合意义来解释等式两边所代表的过程。等式(1)左边表示的是从集合S = {1,...,n}中选出一个包含k+1个元素的子集X的方法数。让我们考虑另一种能产生k+1个元素子集的方式。我们首先选出X中编号最小的元素。如果我们选了1，那么接下来需要从S中剩余的n−1个大于1的元素中再选出k个元素；这有n−1

(k) 种方式。如果改为

如果我们选中的最小编号元素是2，那么我们需要从S中剩下的n−2个大于2的元素中选择k个元素，这有n−2 (k) 种方式。此外，所有这些子集都与

其中编号最小的元素为1的情况。因此，我们应该将每种情况的选择方式数量加到总数中。以此类推，我们根据首先（即编号最小的）选中的对象将整个过程划分为不同的情形，得到：

首先选择的元素是

元素 1，导致 n−1k 种剩余方式，元素 2，

导致 n−2k 种剩余方式，( ) 元素 3，导致 n−3

(k) 剩余方式，

⋮ ⋮

元素 (n−k)，导致 kk 种剩余方式。

这些因子对应于（1）右边的项。（注意，我们选择的编号最小的对象不能高于 n−k，因为我们必须选择 k+1 个不同的对象。）

我们将证明的另一个组合恒等式是

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = 2^n. \tag{2}$$

这个恒等式实际上可以通过在上面的二项式定理中设 a = 1 和 b = 1 立即得出，但我们将提供一个更直接的组合证明。假设我们有一个包含 n 个不同元素的集合 S。在（2）的左边，n

(i) 计算了从 S 中恰好选择大小为 i 的子集的方式数；所以

左边的求和计算了 S 的所有大小的子集的总数。

我们声称（2）右边确实也计算了子集的总数。要看到这一点，只需将子集与一个n位向量对应起来，在每一位 j 上，如果第 j 个元素在子集中则填1，否则填0。因此子集的数量等于n位向量的数量，即 2n（每位有2种选择）。

让我们看一个例子，其中 S = {1,2,3}（即 n = 3）。枚举 S 的所有 23 = 8 个可能子集：∅ ,{1},{2},{3},{1,2},{1,3},{2,3},{1,2,3}。这八个对应的3位向量分别是

Table 1: Permutations of $\{1,2,3\}$ and the number of fixed points.

(π_1,π_2,π_3)	# fixed points
$(1,2,3)$	3
$(1,3,2)$	1
$(2,1,3)$	1
$(2,3,1)$	0
$(3,1,2)$	0
$(3,2,1)$	1

$(0,0,0),(1,0,0),(0,1,0),(0,0,1),(1,1,0),(1,0,1),(0,1,1),(1,1,1)$, respectively. On the left-hand side of equation (2), the term $\binom{3}{0}$ counts the number of ways to choose a subset of S with 0 elements; there is only one such subset, namely the empty set $\varnothing = \{\}$. There are $\binom{3}{1} = 3$ ways of choosing a subset with 1 element, $\binom{3}{2} = 3$ ways of choosing a subset with 2 elements, and $\binom{3}{3} = 1$ way of choosing a subset with 3 elements (namely, the subset consisting of every element of S). Summing, we get $1+3+3+1 = 8$, as expected.

4.2 Permutations and Derangements

Question: Suppose we collect the homeworks of n students, randomly shuffle them, and return them to the students. If we repeated this experiment many times, then on average how many students would receive their own homework?

At first this may seem like a rather challenging problem, but you will be able to answer this question in a few lines later in the course once you have some basic tools of probability theory under your belt. For now, let's translate the question into a mathematical statement and consider a related counting problem.

Label the homeworks as $1,2,\dots,n$ and let π_i denote the homework that is returned to the i-th student. Then, the vector $\pi = (\pi_1,\dots,\pi_n)$ corresponds to a permutation of the set $\{1,\dots,n\}$. Note that $\pi_1,\dots,\pi_n \in \{1,2,\dots,n\}$ are all distinct, so each element in $\{1,\dots,n\}$ appears exactly once in the permutation π. In total, there are $n!$ distinct permutations of $\{1,\dots,n\}$.

In this setting, the i-th student receives their own homework if and only if $\pi_i = i$. Then the question "How many students receive their own homework?" translates into the question of how many indices (i's) satisfy $\pi_i = i$. These are known as fixed points of the permutation π. To illustrate the idea concretely, let us consider the example $n = 3$. Table 1 shows a complete list of all $3! = 6$ permutations of $\{1,2,3\}$, together with the corresponding number of fixed points.

We now consider an interesting counting problem concerning derangements, which are defined as follows.

Definition 10.1 (Derangement)**.** *A permutation with no fixed points is called a* derangement.

Let D_n denote the number of derangements of $\{1,2,\dots,n\}$. As can be seen in Table 1, $D_3 = 2$; specifically, the derangements are $(2,3,1)$ and $(3,1,2)$. Our goal is to find a formula for D_n as a function of n for all $n \in \mathbb{Z}^+$. We will learn two different ways to solve this problem.

The first method involves a combinatorial proof. For concreteness, let's consider the case of $n = 4$. Table 2a lists all derangements of $\{1,2,3,4\}$, with each row corresponding to a derangement. In the first derangement, note that the entries shown in blue corresponds to a derangement of $\{1,2\}$. In the second and third derangements, the entries shown in red are in 1-to-1 correspondence with the derangements of $\{1,2,3\}$, shown in Table 2b. Note that the red 4 in Table 2a satisfies the same constraint as the number 3 in Table 2b; specifically, $\pi_3 \neq 4$ in the former and $\pi_3 \neq 3$ in the latter. In summary, the above discussion suggests that D_4 is related to D_2 and D_3. The following theorem formalizes this relationship.

表1：{1,2,3}的排列及其不动点数量。(π1,π2,π3) 不动点数

$(1,2,3)$	3
$(1,3,2)$	1
$(2,1,3)$	1
$(2,3,1)$	0
$(3,1,2)$	0
$(3,2,1)$	1

(0,0,0),(1,0,0),(0,1,0),(0,0,1),(1,1,0),(1,0,1),(0,1,1),(1,1,1)。在等式（2）的左边，项 30 计算了从 S 中选择一个包含0个元素的子集的方式数；这样的子集只有一个，即空集 ∅= {}。有 3 = 3 种方式选择一个包含1个元素的子集，

$\binom{3}{2}$ = 3 种方式选择一个包含2个元素的子集，3 = 1 种方式选择一个包含3个元素的子集（即 S 的全部元素组成的子集）。相加得 1+3+3+1 = 8，符合预期。

4.2 排列与错位排列

问题：假设我们收集了 n 名学生的作业，随机打乱后返还给他们。如果我们多次重复这个实验，平均有多少学生会收到自己的作业?

起初这似乎是一个相当具有挑战性的问题，但一旦你掌握了一些概率论的基本工具，几行之内你就能回答这个问题。现在，让我们将这个问题转化为数学表述，并考虑一个相关的计数问题。

将作业标记为 1,2,...,n，并令 πi 表示返还给第 i 个学生的作业。
那么，向量 π= (π1,...,πn) 对应于集合 {1,...,n} 的一个排列。注意 π1,...,πn ∈
{1,2,...,n} 都是不同的，因此 {1,...,n} 中的每个元素在排列 π 中恰好出现一次。在
总共有 n! 个 {1,...,n} 的不同排列。

在这种情况下，第 i 个学生收到自己的作业当且仅当 πi = i。那么问题“有多少学生收到了自己的作业？”就转化为有多少个下标（i）满足 πi = i。这些被称为排列 π 的不动点。为了具体说明这一概念，让我们考虑 n = 3 的例子。表1显示了{1,2,3}所有 3! = 6 个排列的完整列表，以及相应的不动点数量。

我们现在考虑一个关于错位排列的有趣计数问题，其定义如下。定义10.1（错位排列）。没有不

动点的排列称为错位排列。

设 Dn 表示 {1,2,...,n} 的错位排列数。从表1可以看出，D3 = 2；具体来说，错位排列是 (2,3,1) 和 (3,1,2)。我们的目标是找到 Dn 作为 n 的函数的公式，适用于所有正整数 n。我们将学习两种不同的方法来解决这个问题。

第一种方法涉及一个组合证明。具体来说，让我们考虑 n = 4 的情况。表2a列出了{1,2,3,4}的所有错位排列，每一行对应一个错位排列。在第一个错位排列中，注意蓝色显示的条目对应于{1,2}的一个错位排列。在第二和第三个错位排列中，红色显示的条目与表2b中所示的{1,2,3}的错位排列一一对应。注意表2a中的红色4满足与表2b中数字3相同的约束；具体来说，前者是 π3 = 4，后者是 π3 = 3。总之，上述讨论表明 D4 与 D2 和 D3 有关。以下定理形式化了这种关系。

Table 2: List of derangements. (a) $n = 4$. (b) $n = 3$.

(a)

π_1	π_2	π_3	π_4
2	**1**	4	3
2	**4**	**1**	3
4	**1**	**2**	3
4	3	2	1
3	4	2	1
2	3	4	1
3	4	1	2
4	3	1	2
3	1	4	2

(b)

π_1	π_2	π_3
2	3	1
3	1	2

Theorem 10.2. *For an arbitrary positive integer $n \geq 3$, the number D_n of derangements of $\{1,\ldots,n\}$ satisfies*

$$D_n = (n-1)(D_{n-1} + D_{n-2}). \tag{3}$$

Proof. We provide a combinatorial proof by establishing relevant bijections. In a derangement $(\pi_1,\ldots,\pi_n)$ of $\{1,\ldots,n\}$, suppose $\pi_n = j \in \{1,\ldots,n-1\}$; i.e., j is what n maps to under the permutation. There are $n-1$ choices for j, and this gives rise to the multiplicative factor $(n-1)$ in (3). (The case of $j = n$ is excluded, since a derangement cannot have $\pi_n = n$.) For a given value of $j \in \{1,\ldots,n-1\}$, there are two cases to consider:

Case 1: If $\pi_j = n$, then j and n get swapped, and there is a 1-to-1 correspondence between the set of derangements of the remaining elements and the set of derangements of $\{1,\ldots,n-2\}$. This gives rise to D_{n-2}.

Case 2: If $\pi_j \neq n$, then n satisfies the same constraint as j, so there is a 1-to-1 correspondence between the set of rearrangements of $\{1,\ldots,j-1,j+1,\ldots,n\}$ and the set of derangements of $\{1,\ldots,n-1\}$. This gives rise to D_{n-1}.

This completes our derivation of (3). □

Together with the boundary conditions $D_1 = 0$ and $D_2 = 1$ (check this), D_n can be determined efficiently (in linear time in n) using the recursion in (3) and memoization. In fact, the recursion can be solved analytically to yield (see Exercise 2) an explicit formula:

$$D_n = n! \sum_{k=0}^{n} \frac{(-1)^k}{k!}. \tag{4}$$

Coming up with the recursion in (3) and solving it require some creativity. As mentioned earlier, there is an alternate way to tackle this counting problem, which turns out to be a more straightforward approach. This alternate method utilizes a powerful tool called the principle of inclusion-exclusion, which we now describe.

5 The Principle of Inclusion-Exclusion

In this section we will learn a useful formula that can be employed to count objects without over-counting. We start with a familiar example. Let A_1 and A_2 be two subsets of the same finite set A, and suppose we want

表2：错位排列列表。（a）n = 4。（b）n = 3。

(a)

π_1	π_2	π_3	π_4
2	**1**	4	3
2	**4**	**1**	3
4	**1**	**2**	3
4	3	2	1
3	4	2	1
2	3	4	1
3	4	1	2
4	3	1	2
3	1	4	2

(b)

π_1	π_2	π_3
2	3	1
3	1	2

定理 10.2。对于任意正整数 n ⩾3，{1,...,n} 的错位排列数 Dn 满足 Dn = (n−1)(Dn−1 +Dn−2)。
（3）证明。我们通过建立相关的双射提供一个组合证明。在错位排列 (π1,...,πn) 中

对于 {1,...,n} 的一个错位排列 (π1,...,πn)，假设 πn = j ∈{1,...,n−1}；即 j 是 n 在该排列下的映射值。j 有 n−1 种选择，这在（3）中产生了乘法因子 (n−1)。（j = n 的情况被排除，因为错位排列不能有 πn = n。）对于给定的 j ∈{1,...,n−1}，有两种情况需要考虑：

情况1：如果 πj = n，则 j 和 n 被交换，剩余元素的错位排列集合与 {1,...,n−2} 的错位排列集合之间存在一一对应关系。这导致了 Dn−2。

情况2：如果 ̸πj = n，则 n 满足与 j 相同的约束，因此 {1,..., j−1, j+1,...,n} 的重排集合与 {1,...,n−1} 的错位排列集合之间存在一一对应关系。这导致了 Dn−1。

这完成了我们对（3）的推导。 □

结合边界条件 D1 = 0 和 D2 = 1（请验证），可以利用（3）中的递推关系和记忆化高效地确定 Dn（在 n 上为线性时间）。事实上，该递推关系可以解析求解，得到（见练习2）一个显式公式：

$$D_n = n!\sum_{k=0}^{n}\frac{(-1)^k}{k!}. \tag{4}$$

想出（3）中的递推关系并求解需要一些创造力。如前所述，解决这个计数问题还有另一种方法，结果证明这是一种更直接的方法。这种替代方法使用了一种称为容斥原理的强大工具，我们现在就来描述它。

5 容斥原理

在本节中，我们将学习一个有用的公式，可用于计数对象而不重复计数。我们从一个熟悉的例子开始。设 A1 和 A2 是同一有限集 A 的两个子集，并假设我们想要

to count the number of elements in $A_1 \cup A_2$. If A_1 and A_2 are disjoint, then clearly $|A_1 \cup A_2| = |A_1| + |A_2|$. If A_1 and A_2 have common elements, however, then those elements would be counted twice in the previous formula. To correct for this overcounting, we need to subtract $|A_1 \cap A_2|$ from $|A_1| + |A_2|$, thus yielding

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|.$$

The principle of inclusion-exclusion generalizes this concept to an arbitrary number of subsets.

Theorem 10.3 (Inclusion-Exclusion)**.** *Let $A_1, \ldots, A_n$ be arbitrary subsets of the same finite set A. Then,*

$$|A_1 \cup \cdots \cup A_n| = \sum_{k=1}^{n} (-1)^{k-1} \sum_{S \subseteq \{1,\ldots,n\}: |S|=k} |\cap_{i \in S} A_i|. \tag{5}$$

Remarks:

1. The inner summation in (5) is over all size-k subsets of $\{1, 2, \ldots, n\}$. More explicitly, (5) can be written as

$$\begin{aligned} |A_1 \cup \cdots \cup A_n| = \sum_{i=1}^{n} |A_i| \; - \sum_{i<j} |A_i \cap A_j| \; + \sum_{i<j<k} |A_i \cap A_j \cap A_k| \; - \sum_{i<j<k<l} |A_i \cap A_j \cap A_k \cap A_l| \; + \cdots \\ + (-1)^{n-1} |A_1 \cap A_2 \cap \cdots \cap A_n|, \end{aligned} \tag{6}$$

   where $\sum_{i<j}$ denotes summing over all $i, j \in \{1, \ldots, n\}$ such that $i < j$, and so on.

2. The standard way to prove Theorem 10.3 is via induction on n. Here, we will provide a combinatorial proof.

Proof. First, note that if $a \notin A_1 \cup \cdots \cup A_n$, then $a \notin A_i$ for all $i \in \{1, \ldots, n\}$ and hence $a \notin \cap_{i \in S} A_i$ for any $S \subseteq \{1, \ldots, n\}$, so it is not counted by the right-hand side (RHS) of (5). We now show that if $a \in A_1 \cup \cdots \cup A_n$, then it is counted exactly once by the RHS. Define the index set $M \subseteq \{1, \ldots, n\}$ as

$$M = \{i \in \{1, \ldots, n\} \mid a \in A_i\};$$

that is, $a \in A_i$ if and only if $i \in M$. Note that $m := |M| \geq 1$, since $a \in A_1 \cup \cdots \cup A_n$. Clearly, if $S \not\subseteq M$, then $a \notin \cap_{i \in S} A_i$ (i.e., if S contains an index i not in M, then $a \notin A_i$, so $a \notin \cap_{i \in S} A_i$). Furthermore, $a \in \cap_{i \in S} A_i$ for any non-empty subset $S \subseteq M$. Therefore, the number of times that a is counted on the RHS of (5) is

$$\sum_{k=1}^{m} (-1)^{k-1} \sum_{S \subseteq M: |S|=k} 1 \; = \; \sum_{k=1}^{m} (-1)^{k-1} \binom{m}{k} \; = \; 1,$$

where the first equality follows from the fact that the number of distinct k-element subsets of M is $\binom{m}{k}$, while the second equality follows from Corollary 10.1. □

Application: Derangements Revisited

We now illustrate the power of Theorem 10.3 by applying it to derive the formula shown in (4) for the number D_n of derangements of $\{1, \ldots, n\}$.

来计算 A1 ∪A2 中元素的数量。如果 A1 和 A2 不相交，则显然 |A1 ∪A2| = |A1|+|A2|。然而，若 A1 和 A2 有公共元素，则这些元素在前述公式中会被重复计算两次。为了纠正这种重复计数，我们需要从 |A1|+|A2| 中减去 |A1 ∩A2|，从而得到

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|.$$

容斥原理将这一概念推广到任意数量的子集。定理 10.3（容斥原理）。设 A1,...,An 是同一个

有限集 A 的任意子集。则有

$$|A_1 \cup \cdots \cup A_n| = \sum_{k=1}^{n} (-1)^{k-1} \sum_{S \subseteq \{1,\ldots,n\}:|S|=k} |\cap_{i \in S} A_i|. \tag{5}$$

备注：

1. (5) 式中的内层求和是对 {1,2,...,n} 的所有大小为 k 的子集进行的。更明确地，(5) 可以写成

$$\begin{aligned}|A_1 \cup \cdots \cup A_n| = \sum_{i=1}^{n} |A_i| \; - \sum_{i<j} |A_i \cap A_j| \; + \sum_{i<j<k} |A_i \cap A_j \cap A_k| \; - \sum_{i<j<k<l} |A_i \cap A_j \cap A_k \cap A_l| \; + \cdots \\ + (-1)^{n-1} |A_1 \cap A_2 \cap \cdots \cap A_n|,\end{aligned} \tag{6}$$

   其中 Σi<j 表示对所有满足 i < j 的 i, j ∈{1,...,n} 求和，依此类推。

2. 证明定理 10.3 的标准方法是对 n 进行归纳法。这里我们将提供一个组合证明。

证明。首先注意到，如果 a ∈A1 ∪· · ·∪An，则对所有 i ∈{1,...,n} 都有 a ∈Ai，因此对于任意 S ⊆{1,...,n}，因此不被 (5) 式右边（RHS）计算。现在我们证明如果 a ∈A1∪· · ·∪An，那么它在右边恰好被计算一次。定义索引集 M ⊆{1,...,n} 为

$$M = \{i \in \{1,\ldots,n\} \mid a \in A_i\};$$

即 a ∈Ai 当且仅当 i ∈M。注意 m := |M| ⩾1，因为 a ∈A1 ∪· · ·∪An。显然，如果 S ⊆M，则 a ∈∩i∈SAi（也就是说，如果 S 包含一个不在 M 中的索引 i，则 a ∈Ai，因此 a ∈∩i∈SAi）。此外，对于任何非空子集 S ⊆M，都有 a ∈∩i∈SAi。因此，a 在 (5) 式右边被计算的次数为

$$\sum_{k=1}^{m} (-1)^{k-1} \sum_{S \subseteq M:|S|=k} 1 \; = \; \sum_{k=1}^{m} (-1)^{k-1} \binom{m}{k} \; = \; 1,$$

其中第一个等式成立是因为 M 的大小为 k 的不同子集数量是 m $\binom{}{k}$, 而第二个等式由推论 10.1 得出。 □

应用：错位排列再探

我们现在通过应用定理 10.3 来推导 (4) 式中关于 {1,...,n} 的错位排列数 Dn 的公式，以此说明该定理的强大之处。

Let A_i denote the set of all permutations of $\{1,\ldots,n\}$ with i being a fixed point. Then, since there are $n!$ distinct permutations of $\{1,\ldots,n\}$ and $|A_1 \cup A_2 \cup \cdots \cup A_n|$ counts the number of permutations with at least one fixed point, we have

$$D_n = n! - |A_1 \cup A_2 \cup \cdots \cup A_n|. \tag{7}$$

What is the cardinality of A_i? We know that i maps to i, while the remaining $n-1$ elements can be arranged in an arbitrary way. So, $|A_i| = (n-1)!$. Similarly, for $i \neq j$, i maps to i and j maps to j, while the remaining $n-2$ elements can be arranged in an arbitrary way, so we conclude $|A_i \cap A_j| = (n-2)!$. In general, for any subset $S \subseteq \{1,\ldots,n\}$ of size $|S| = k$, $|\cap_{i\in S} A_i| = (n-k)!$, while there are $\binom{n}{k}$ such subsets. Thus, the inclusion-exclusion formula (5) gives

$$|A_1 \cup A_2 \cup \cdots \cup A_n| = \sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} (n-k)! = \sum_{k=1}^{n} (-1)^{k-1} \frac{n!}{k!(n-k)!}(n-k)! = n! \sum_{k=1}^{n} \frac{(-1)^{k-1}}{k!}.$$

Substituting this result into (7) gives

$$D_n = n! - n! \sum_{k=1}^{n} \frac{(-1)^{k-1}}{k!} = n! \sum_{k=0}^{n} \frac{(-1)^k}{k!}.$$

6 Exercises

1. Provide a direct combinatorial proof of Corollary 10.1.
2. Using (3) and the boundary conditions $D_1 = 0$ and $D_2 = 1$, derive (4). [Hint: For $n \geq 2$, define $Q_n = D_n - nD_{n-1}$ and obtain a recursion for Q_n.]
3. Prove Theorem 10.3 using induction.

令 A_i 表示 {1,...,n} 的所有排列组成的集合，其中 i 是一个不动点。那么，由于共有 n! {1,...,n}的不同排列总数为n!，而$|A_1 \cup A_2 \cup \cdots \cup A_n|$统计的是至少有一个不动点的排列数量，因此我们有

$$D_n = n! - |A_1 \cup A_2 \cup \cdots \cup A_n|。 \tag{7}$$

A_i的基数是多少？我们知道i映射到i，而其余n−1个元素可以任意排列，因此$|A_i| = (n-1)!$。类似地，对于i = j，i映射到i，j映射到j，其余n−2个元素可以任意排列，所以我们得出$|A_i \cap A_j| = (n-2)!$。一般来说，对于任意大小为$|S| = k$的子集$S \subseteq \{1,\ldots,n\}$，有$|\cap_{i\in S} A_i| = (n-k)!$，而这样的子集共有$\binom{n}{k}$个。因此，容斥公式(5)给出

$$|A_1 \cup A_2 \cup \cdots \cup A_n| = \sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} (n-k)! = \sum_{k=1}^{n} (-1)^{k-1} \frac{n!}{k!(n-k)!} (n-k)! = n! \sum_{k=1}^{n} \frac{(-1)^{k-1}}{k!}.$$

将此结果代入(7)式可得

$$D_n = n! - n! \sum_{k=1}^{n} \frac{(-1)^{k-1}}{k!} = n! \sum_{k=0}^{n} \frac{(-1)^k}{k!}.$$

6 练习题

1. 为推论10.1提供一个直接的组合证明。

2. 利用(3)式以及边界条件$D_1 = 0$和$D_2 = 1$，推导出(4)式。[提示：对于$n \geq 2$，定义$Q_n = D_n - nD_{n-1}$，并得到Q_n的递推关系。]

3. 用归纳法证明定理10.3。