Combinatorics Review

Daniel H. Luecking

February 14, 2024

Rule of sum

Rule

If one task can be done in m *ways while another task can be performed in* k *ways, and the two tasks cannot be performed simultaneously, then there are* $m + k$ *possible ways to perform one or the other task.*

Rule of sum

Rule

If one task can be done in m ways while another task can be performed in k ways, and the two tasks cannot be performed simultaneously, then there are $m + k$ possible ways to perform one or the other task.

We use this to count collections of things, where the tasks result in an object in the set we are counting. For this to work, there should be a one-to-one correspondence between the ways of completing the tasks and the collection we are counting.

Rule of sum

Rule

If one task can be done in m ways while another task can be performed in k ways, and the two tasks cannot be performed simultaneously, then there are $m + k$ possible ways to perform one or the other task.

We use this to count collections of things, where the tasks result in an object in the set we are counting. For this to work, there should be a one-to-one correspondence between the ways of completing the tasks and the collection we are counting.

The rule of sum extends to any number of tasks, as long as no two tasks can be performed simultaneously.

Rule of product

Rule

Suppose one task can be performed in m ways and a second task can be performed in k ways. Then the number of ways to perform the first followed by the second is mk.

Rule of product

Rule

Suppose one task can be performed in m ways and a second task can be performed in k ways. Then the number of ways to perform the first followed by the second is mk.

This rule extends to a sequence of any number of tasks.

Rule of product

Rule

Suppose one task can be performed in m ways and a second task can be performed in k ways. Then the number of ways to perform the first followed by the second is mk.

This rule extends to a sequence of any number of tasks.

It is important, in applications, that the following are satisfied

1. The number of ways to perform a task does not depend on the outcome of the other tasks.

Rule of product

Rule

Suppose one task can be performed in m ways and a second task can be performed in k ways. Then the number of ways to perform the first followed by the second is mk.

This rule extends to a sequence of any number of tasks.

It is important, in applications, that the following are satisfied

1. The number of ways to perform a task does not depend on the outcome of the other tasks.
2. If we use this to count final outcomes, we must have
    - If any one task is done differently, the final outcome is different.

Rule of product

Rule

Suppose one task can be performed in m ways and a second task can be performed in k ways. Then the number of ways to perform the first followed by the second is mk.

This rule extends to a sequence of any number of tasks.

It is important, in applications, that the following are satisfied

1. The number of ways to perform a task does not depend on the outcome of the other tasks.
2. If we use this to count final outcomes, we must have
   - If any one task is done differently, the final outcome is different.
   - All final outcomes are produced by some way of performing the sequence of tasks.

Permutations

A *permutation* on a set A is a subset of A arranged in order.

Permutations

A *permutation* on a set A is a subset of A arranged in order.

We are interested in the number of possible permutations. That number doesn't depend on what is in the set A, only on how many elements A has and on how many elements we are choosing to put in order.

Permutations

A *permutation* on a set A is a subset of A arranged in order.

We are interested in the number of possible permutations. That number doesn't depend on what is in the set A, only on how many elements A has and on how many elements we are choosing to put in order.

Definition: $P(n, k)$ stands for the number of permutations possible when choosing k elements from a set of size n.

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

There is a one-to-one correspondence between permutations of k things from an n-set and one-to-one functions from any k-set A into and n-set B. that is, $P(n, k)$ is the number of one-to-one functions from A to B.

Permutations with repetition (arrangements)

How many ways can we arrange the letters of the string `"BOOKKEEPER"`?

Permutations with repetition (arrangements)

How many ways can we arrange the letters of the string `"BOOKKEEPER"`?

To be clear what this means, imagine we write each letter on a tile, `B` `O` `O` `K` `K` `E` `E` `P` `E` `R`, then we shuffle these tiles and line them up, *recording the resulting string* of letters.

Permutations with repetition (arrangements)

How many ways can we arrange the letters of the string `"BOOKKEEPER"`?

To be clear what this means, imagine we write each letter on a tile, `B` `O` `O` `K` `K` `E` `E` `P` `E` `R`, then we shuffle these tiles and line them up, *recording the resulting string* of letters.

If we count each tile as different, we get $10!$ permutations. But if, for example, all we do is exchange the two `O`-tiles we don't get different strings.

Permutations with repetition (arrangements)

How many ways can we arrange the letters of the string `"BOOKKEEPER"`?

To be clear what this means, imagine we write each letter on a tile, `B` `O` `O` `K` `K` `E` `E` `P` `E` `R`, then we shuffle these tiles and line them up, *recording the resulting string* of letters.

If we count each tile as different, we get $10!$ permutations. But if, for example, all we do is exchange the two `O`-tiles we don't get different strings. Same for the `K`-tiles and `E`-tiles.

Permutations with repetition (arrangements)

How many ways can we arrange the letters of the string `"BOOKKEEPER"`?

To be clear what this means, imagine we write each letter on a tile, `B` `O` `O` `K` `K` `E` `E` `P` `E` `R`, then we shuffle these tiles and line them up, *recording the resulting string* of letters.

If we count each tile as different, we get $10!$ permutations. But if, for example, all we do is exchange the two `O`-tiles we don't get different strings. Same for the `K`-tiles and `E`-tiles.

We have $2!2!3!$ different permutations of tiles that produce the same string.

Permutations with repetition (arrangements)

How many ways can we arrange the letters of the string `"BOOKKEEPER"`?

To be clear what this means, imagine we write each letter on a tile, `B` `O` `O` `K` `K` `E` `E` `P` `E` `R`, then we shuffle these tiles and line them up, *recording the resulting string* of letters.

If we count each tile as different, we get $10!$ permutations. But if, for example, all we do is exchange the two `O`-tiles we don't get different strings. Same for the `K`-tiles and `E`-tiles.

We have $2!2!3!$ different permutations of tiles that produce the same string. We have a set (all permutations of tiles) divided into clusters of equal size ($2!2!3!$) and we want the number of clusters:

Permutations with repetition (arrangements)

How many ways can we arrange the letters of the string `"BOOKKEEPER"`?

To be clear what this means, imagine we write each letter on a tile, `B` `O` `O` `K` `K` `E` `E` `P` `E` `R`, then we shuffle these tiles and line them up, *recording the resulting string* of letters.

If we count each tile as different, we get $10!$ permutations. But if, for example, all we do is exchange the two `O`-tiles we don't get different strings. Same for the `K`-tiles and `E`-tiles.

We have $2!2!3!$ different permutations of tiles that produce the same string. We have a set (all permutations of tiles) divided into clusters of equal size ($2!2!3!$) and we want the number of clusters: divide the number of permutations by the size of the clusters: $\dfrac{10!}{2!2!3!}$.

Permutations with (sub)strings attached

We know there are $8!/2!$ arrangements of `"GEODESIC"`. Now how many contain the substring `"DOG"`?

Permutations with (sub)strings attached

We know there are $8!/2!$ arrangements of `"GEODESIC"`. Now how many contain the substring `"DOG"`? If we imagine the letters on tiles again, we can glue the `D`, `O` and `G` tiles together to get
`DOG` `E` `E` `S` `I` `C`

Permutations with (sub)strings attached

We know there are $8!/2!$ arrangements of `"GEODESIC"`. Now how many contain the substring `"DOG"`? If we imagine the letters on tiles again, we can glue the `D`, `O` and `G` tiles together to get

`DOG` `E` `E` `S` `I` `C`

If we arrange these and record the resulting string we are guaranteed to get one with the substring `"DOG"`.

Permutations with (sub)strings attached

We know there are $8!/2!$ arrangements of `"GEODESIC"`. Now how many contain the substring `"DOG"`? If we imagine the letters on tiles again, we can glue the `D`, `O` and `G` tiles together to get

`DOG` `E` `E` `S` `I` `C`

If we arrange these and record the resulting string we are guaranteed to get one with the substring `"DOG"`. Since there are 6 objects with 2 of them identical, there are $6!/2!$ arrangements.

Permutations with (sub)strings attached

We know there are $8!/2!$ arrangements of `"GEODESIC"`. Now how many contain the substring `"DOG"`? If we imagine the letters on tiles again, we can glue the `D`, `O` and `G` tiles together to get
`DOG` `E` `E` `S` `I` `C`
If we arrange these and record the resulting string we are guaranteed to get one with the substring `"DOG"`. Since there are 6 objects with 2 of them identical, there are $6!/2!$ arrangements.

Note that we can subtract to get the number of arrangements that do not have the substring `"DOG"`: $8!/2 - 6!/2$.

Selections without order

A *combination of n things taken k at a time* is subset of size k from an n-set. The distinction between a permutation and a combination is that 2 combinations are the same if they have the same elements, while 2 permutations are the same if they have the same elements in the same order.

Selections without order

A *combination of n things taken k at a time* is subset of size k from an n-set. The distinction between a permutation and a combination is that 2 combinations are the same if they have the same elements, while 2 permutations are the same if they have the same elements in the same order.

More generally, viewing permutations as a one-to-one functions, then to get a permutation we have to choose the correct number of elements and then associate each of those with an element of some set. With a combination, we are done when we have chosen the elements.

Selections without order

A *combination of n things taken k at a time* is subset of size k from an n-set. The distinction between a permutation and a combination is that 2 combinations are the same if they have the same elements, while 2 permutations are the same if they have the same elements in the same order.

More generally, viewing permutations as a one-to-one functions, then to get a permutation we have to choose the correct number of elements and then associate each of those with an element of some set. With a combination, we are done when we have chosen the elements.

That is, for combinations we just choose them, for permutations we choose them and then label them.

Selections without order

A *combination of n things taken k at a time* is subset of size k from an n-set. The distinction between a permutation and a combination is that 2 combinations are the same if they have the same elements, while 2 permutations are the same if they have the same elements in the same order.

More generally, viewing permutations as a one-to-one functions, then to get a permutation we have to choose the correct number of elements and then associate each of those with an element of some set. With a combination, we are done when we have chosen the elements.

That is, for combinations we just choose them, for permutations we choose them and then label them.

Definition: $C(n, k)$ stands for the number of combinations possible when choosing k elements from a set of size n. It has the formula

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

Combinations with repetition

If we are selecting k times with repetition from a set of size n, then number of ways to do this is

$$C(k+n-1,k) = \frac{(k+n-1)!}{k!(n-1)!}.$$

Examples of this: giving k identical prizes to a set of n contestants, with no limit on what each contestants may get, is like selecting k times, with repetition allowed, from a set of size n.

Since all that matters is how many prizes each contestant gets, thus is equivalent to the solutions of the equation

$$p_1 + p_2 + \cdots + p_n = k$$

where each p_j is the number of prizes the jth contestant receives.

Inclusion-exclusion

Suppose we have a set with N elements and these elements might or might not satisfy come conditions $c_1, c_2, c_3, \ldots, c_r$. If we let $N(c_j)$ be the number that satisfy c_j, then we let $S_1 = \sum_{j=1}^{r} N(c_j)$.

Inclusion-exclusion

Suppose we have a set with N elements and these elements might or might not satisfy come conditions $c_1, c_2, c_3, \ldots, c_r$. If we let $N(c_j)$ be the number that satisfy c_j, then we let $S_1 = \sum_{j=1}^{r} N(c_j)$.

If $N(c_j c_k)$ be the number that satisfy c_j and c_k, then we let $S_2 = \sum_{1 \leq j < k \leq r} N(c_j c_k)$. And $S_3 = \sum_{1 \leq j < k < m \leq r} N(c_j c_k c_m)$.

Inclusion-exclusion

Suppose we have a set with N elements and these elements might or might not satisfy come conditions $c_1, c_2, c_3, \ldots, c_r$. If we let $N(c_j)$ be the number that satisfy c_j, then we let $S_1 = \sum_{j=1}^{r} N(c_j)$.

If $N(c_j c_k)$ be the number that satisfy c_j and c_k, then we let $S_2 = \sum_{1 \leq j < k \leq r} N(c_j c_k)$. And $S_3 = \sum_{1 \leq j < k < m \leq r} N(c_j c_k c_m)$.

Then the number of elements that satisfy none of the conditions is

$$E_0 = N - S_1 + S_2 - S_3 + \cdots \pm S_r$$

The number of elements that satisfy at least one of the conditions is

$$L_1 = S_1 - S_2 + S_3 - \cdots \mp S_r$$

The number of elements that satisfy at least one of the conditions is

$$L_1 = S_1 - S_2 + S_3 - \cdots \mp S_r$$

The number of elements that satisfy exactly one of the conditions is

$$E_1 = S_1 - 2S_2 + 3S_3 - \cdots \mp rS_r$$

If L_k is the number that satisfy at least k conditions and E_k is the number that satisfy exactly k conditions, the book has formulas for these as well

It is useful to know that the number of terms in each sum S_k is $C(r, k)$ if there are r conditions.