1 Basics of combinatorics: counting and permutations

Combinatorics is, roughly speaking, the study of combinations, arrangements and orderings. Here we focus on two aspects: counting and permutations. Combinatorics is also closely related to the theory of graphs that we will discuss towards the end of the course. Orderings of sets will also be considered later in the course.

1.1 Counting

Imagine you're making an important choice, like which CISC course to take as an elective next year. While planning your schedule, you realize you also want to take a mathematics elective next year. There are 5 CISC courses and 4 MATH courses from which you can choose your electives. How many possible pairs of courses exist?

The **product rule** tells you exactly how many possibilities exist in this scenario: if there are i ways to make one choice and j ways to make another choice, then there is a total of ij ways to make both choices. Generalizing from two choices to m choices, we get the product rule.

From our previous scenario, the product rule tells us that you have $5 \cdot 4 = 20$ different elective options available. Aside from scheduling, the product rule appears in many other areas both of computing and of life.

Example 1.1 *Car licence plates in Ontario follow a certain pattern: four uppercase letters followed by three numbers.*

How many possible licence plates can the government produce? Let the set of letters consist of the usual 26 elements $\{A, B, \ldots, Z\}$*, and let the set of numbers consist of the 10 single digits* $\{0, 1, \ldots, 9\}$*. Then, taking four elements from the letter set and three elements from the number set, we get a total of* $26^4 \cdot 10^3 = 456\,976\,000$ *possible licence plates.*

As the previous examples might have revealed, the product rule is actually a disguised result about the Cartesian product of sets. Consider, for example, two sets $\{a, b, c\}$ and

$\{d, e\}$. How many ways can we choose one element from each of these sets? If we take the Cartesian product of both sets, the resultant set gives us every possible pair of elements: $\{(a, d), (b, d), (c, d), (a, e), (b, e), (c, e)\}$. Therefore, we have six ways to choose one element from each set.

Thinking about the product rule in the context of the cardinality of the Cartesian product of sets gives the following formula.

Proposition 1.1 (Product rule) *Let $A_1, A_2, \ldots, A_m$ be disjoint sets each of finite cardinality. Then*

$$|A_1 \times A_2 \times \cdots \times A_m| = |A_1| \cdot |A_2| \cdot \quad \cdots \quad \cdot |A_m|.$$

Note that the sets $A_1, A_2, \ldots, A_m$ need not be disjoint, since our choices from one set don't depend on any of our previous or future choices from other sets, that is, cardinality of the set $A_1 \times \cdots \times A_m$ depends only on the cardinalities of the sets A_i, and not on whether or not they are disjoint.

Imagine again that you're making an important choice. This time, you're deciding on an undergraduate honours project. Two professors in the School of Computing are interested in supervising you; the first professor has three project ideas and the second professor has four project ideas. How many options do you have for choosing your project?

In this scenario, we can't use the product rule because you may only choose a single project; you aren't choosing one project from each of the two professors. Instead, this is where the **sum rule** applies: if there are i ways to make one choice and j ways to make another choice, and the two choices cannot be made simultaneously, then there is a total of $i + j$ ways to make one of these choices. Generalizing from two choices to m choices, we get the sum rule.

From our previous scenario, the sum rule tells us that you have $3 + 4 = 7$ different project options available.

Example 1.2 *Queen's University has a password policy similar to the following: a password must be between 10 and 12 characters long; it must consist of uppercase letters (A–Z), lowercase letters (a–z), and numbers (0–9); and it must contain at least one of each of those* *characters.*

How many passwords exist that meet all of the above criteria? We can determine this using both the sum rule and the product rule. Let P_{10}*,* P_{11}*, and* P_{12} *denote the sets of valid passwords of length 10, 11, and 12, respectively. The sum rule tells us that the total number of valid passwords is* $P_{10} + P_{11} + P_{12}$*.*

Now, let's determine the values of P_{10}*,* P_{11}*, and* P_{12}*. We do this using the product rule. First, we will determine all passwords of length* n *(which gives us* 62^n*), and then we will subtract from this value all passwords that don't contain at least one uppercase letter (*36^n*), lowercase letter (*36^n*), or number (*52^n*). This gives us the following values:*

$$P_{10} = 62^{10} - 36^{10} - 36^{10} - 52^{10} = 687\,431\,943\,039\,157\,248,$$
$$P_{11} = 62^{11} - 36^{11} - 36^{11} - 52^{11} = 44\,256\,451\,766\,801\,594\,368, \text{ and}$$
$$P_{12} = 62^{12} - 36^{12} - 36^{12} - 52^{12} = 2\,825\,912\,993\,235\,006\,394\,368.$$

Therefore, there exists a total of $2\,870\,856\,876\,944\,847\,145\,984$ *(2 sextillion) valid passwords. To put this number into context, astronomers estimate there are about 1 sextillion stars in the universe!*

Again, just like with the product rule, the sum rule is a set theory result in disguise. This time, since we aren't taking elements from every set, we don't care about tuples of elements. Instead, we are considering all sets together—that is, the union of all sets—and selecting our element from the lot, so we just care about the total number of elements. First in the special case where the sets are disjoint we get the following simple formula.

Proposition 1.2 (Sum rule) *Let* $A_1, A_2, \ldots, A_m$ *be disjoint sets each of finite cardinality. Then*

$$|A_1 \cup A_2 \cup \cdots \cup A_m| = |A_1| + |A_2| + \cdots + |A_m|.$$

The sum rule of Proposition 1.2 is obviously dependent on the disjointness of the sets.

With non-disjoint sets we have to modify the formula for counting the cardinality of union, that is, we have to make sure that each element is counted only once. We call this method of including individual elements and excluding common elements the **inclusion-exclusion principle**.

Example 1.3 *Consider the sets* $A = \{1, 2, 4, 8\}$ *and* $B = \{2, 4, 6, 8\}$*. How many elements are in the union of the two sets,* $A \cup B$*?*

Clearly, $|A \cup B| \neq 8$*, since we would be counting the elements 2, 4, and 8 each twice. (In a set, we do not permit multiple copies of an element.)*

What we can do instead is count the number of elements in each set and sum them together as usual, but also subtract from this sum the number of elements common to both sets. Since A *and* B *share three elements, we have that* $|A \cup B| = 4 + 4 - 3 = 5$*. A quick check reveals that, indeed,* $A \cup B = \{1, 2, 4, 6, 8\}$*.*

The previous example illustrates the inclusion-exclusion principle on two sets. Using set notation, the principle in this case yields

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

You might think this generalizes easily to three sets by adding $|C|$ to the sum and subtracting both $|A \cap C|$ and $|B \cap C|$ from the sum. However, this leads to another issue: doing so would have us subtract more than once any elements common to each of A, B, and C. Instead of overcounting, we're undercounting!

Since we're undercounting by exactly the number of elements common to all three sets, we must add that value back to the sum. As a result, the inclusion-exclusion principle on three sets is written thus:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|.$$

At this point, you might notice a pattern developing with each formulation of the inclusion-exclusion principle. We add cardinalities of single sets, we subtract cardinalities of pairs of sets, we add cardinalities of triples of sets, and so on. In general, when we introduce a term that involves k sets, we add if k is odd and we subtract if k is even.

Using this observation, we can write out the general form of the inclusion-exclusion principle.

Theorem 1.1 (Inclusion-exclusion principle) *Let $A_1, A_2, \dots, A_m$ be sets each of finite cardinality. Then*

$$\left|\bigcup_{i=1}^{m} A_i\right| = \sum_{k=1}^{m}(-1)^{k+1}\left(\sum_{1\le i_1<\dots<i_k\le m} |A_{i_1}\cap\dots\cap A_{i_k}|\right).$$

The previous expression might seem a bit intimidating, so let's break it down into parts. The part of the expression to the left of the equals sign, $|\bigcup_{i=1}^{m} A_i|$, is a shorthand way to say $|A_1 \cup \dots \cup A_m|$. To the right of the equals sign, we have two parts to consider. The sum, $\sum_{k=1}^{m}(-1)^{k+1}$, alternates between adding and subtracting the kth term of the expression; we add the term if k is odd and subtract the term if k is even. The term we are adding or subtracting, $\left(\sum_{1\le i_1<\dots<i_k\le m} |A_{i_1}\cap\dots\cap A_{i_k}|\right)$, is a notation-heavy way of writing "add together the cardinalities of all possible intersections of k sets among $A_1, \dots, A_m$.

If you're worried about having to remember the unwieldy general form of the inclusion-exclusion principle, don't be; most of the time, you will only need to use the cleaner two- or three-set formula. The proof of Theorem 1.1 is presented in section 19 of the textbook. We don't go though the complete proof in this course but may consider special cases like $m = 4$ (in class or in the assignments).

1.2 Pigeonhole Principle

In a post office or mailroom, mail is sorted into small slots in the wall, colloquially called "pigeonholes". (This word wasn't just made up for fun; pigeon holes were originally boxes on walls used to store domesticated birds.) In 1834, a German mathematician named Peter Gustav Lejeune Dirichlet studied combinatorial problems involving n items being placed into m containers where $n > m$. Dirichlet needed a name for his work; his father happened to be a postmaster, and postmasters place mail into pigeonholes, so that term seemed quite fitting. From this, the term pigeonhole principle was coined.

Theorem 1.2 (Pigeonhole principle) *If n elements are partitioned into m subsets, then at least one subset must contain at least $\lceil n/m\rceil$ elements.*

Proof. Suppose we have n elements partitioned into m subsets. By contradiction, suppose that no subset contains more than $\lceil n/m \rceil - 1$ elements. Then the total number of elements is at most

$$m\left(\left\lceil \frac{n}{m} \right\rceil - 1\right) < m\left(\left(\frac{n}{m} + 1\right) - 1\right) = n.$$

Thus, we must have fewer than n elements. However, we assumed we had exactly n elements. Therefore, our assumption was incorrect and at least one subset must contain at least $\lceil n/m \rceil$ elements. $\square$

To illustrate the pigeonhole principle, let's consider an example that involves (what else?) pigeons. Suppose there are nine nests in a tree, and ten pigeons fly to that tree to roost. Nine pigeons can each take one nest, but that leaves one pigeon without a nest. If all ten pigeons are in a nest, then it must be the case that at least one nest contains more than one pigeon.

So what do pigeons and nests have to do with combinatorics? Well, nothing. But the idea behind the pigeonhole principle can be applied to a number of mathematical and computational problems.

Example 1.4 *In the 2017–2018 academic year, there were 11 783 students enrolled in the Faculty of Arts and Science at Queen's University. Assume that each of these students was born in the same four-year period of 1996–1999. The given four-year period consisted of 1 461 days. Therefore, by the pigeonhole principle, at least* $\lceil 11\,783/1\,461 \rceil = 9$ *students in the Faculty of Arts and Science were born on the same day.*

Example 1.5 *Lossless compression algorithms allow users to compress files without losing any of the original data. In a perfect world, a lossless compression algorithm would always make files smaller. However, on some files, the algorithm actually increases the file size. Why?*

By contradiction, suppose there exists a lossless compression algorithm that compresses every file F *into a compressed file* F'*, where* $\mathit{size}(F') \leq \mathit{size}(F)$*. Let* M *be the smallest natural number such that there exists a file* F *with* $\mathit{size}(F) = M$ *that can be compressed to a file* F' *with* $\mathit{size}(F') = N$*.*

Since $N < M$, every size-N file keeps its original size after compression. There are 2^N such files. Since F' is also of size N, we have a total of $2^N + 1$ files of size N. However, $2^N < 2^N + 1$, so the pigeonhole principle tells us that two different compressed files of size N must be identical. Such files cannot be losslessly decompressed, since we don't know which of the two original files they once were. Therefore, our assumption was incorrect and no lossless compression algorithm can always make a file smaller.

2 Permutations

To order a collection of n objects, we pick one object to be the first, another object to be the second, and so on. There are n different choices for the first object, then $n - 1$ remaining choices of the second object, $n-2$ for the third, and so on, until only one choice remains for the last object. The total number of ways to order the n objects is the product of integers from 1 to n, which is called the *factorial* of n, denoted $n!$. This is illustrated by the following small example.

Example 2.1 *Consider a set A with 4 elements, a_1, a_2, a_3, and a_4. Let π be a permutation of A. The permutation π, roughly speaking, is defined by inserting each element into an ordered list with four empty spots.*

Consider the element a_1. Currently all four spots are empty in the list $(\text{␣}\ \text{␣}\ \text{␣}\ \text{␣})$, so there are four spaces into which a_1 can be inserted. Suppose a_1 is inserted into the second space.

Consider the element a_2. The current state of the list is $(\text{␣}\ a_1\ \text{␣}\ \text{␣})$, so there are three spaces into which a_2 can be inserted. Suppose a_2 is inserted into the fourth space.

Next consider the element a_3. The current state of "the incomplete permutation" is $(\text{␣}\ a_1\ \text{␣}\ a_2)$, so there are two spaces into which a_3 can be inserted. Suppose a_3 is inserted into the first space.

Finally, consider the element a_4. The current state of the list is $(a_3\ a_1\ \text{␣}\ a_2)$, so there is only a single space into which a_4 can be inserted.

Altogether, there are $4 \times 3 \times 2 \times 1 = 24$ *different ways to arrange the elements of* A*, so there are* 24 *possible permutations of* A*.*

An ordering, or rearrangement, of n objects is called a *permutation* of the objects. Formally permutations are defined as bijective functions from a set A to itself (see section 27 in the textbook).

When talking about permutations it is convenient to view a finite set as an ordered list of elements $A = (a_1, a_2, \ldots, a_n)$. (Recall that when talking about unordered sets the elements were enclosed inside braces "$\{\ldots\}$".) Suppose that we first fix an arbitrary order of the n elements of a set A. A bijective mapping $A \longrightarrow A$ then uniquely determines the reordering, or permutation.

Definition 2.1 (Permutation) *A permutation of a set A is a bijective function* $\pi : A \to A$*.*

A permutation is just a function that maps elements of a set A to elements of the same set A. If our informal definition of a permutation is an arrangement of elements, then the permutation π itself is what performs the rearranging of elements.

Example 2.2 *Suppose we have a set* $A = (1, 2, 3, 4, 5, 6)$*, and suppose further that we have a permutation* π *defined as*

$$\begin{aligned} \pi(1) &= 3; & \pi(4) &= 2; \\ \pi(2) &= 4; & \pi(5) &= 1; \\ \pi(3) &= 5; & \pi(6) &= 6. \end{aligned}$$

Then $\pi(A) = (3, 4, 5, 2, 1, 6)$*,* $\pi(\pi(A)) = (5, 2, 1, 4, 3, 6)$*, and so on.*

Naturally the sets A, $\pi(A)$, and $\pi(\pi(A))$ contain the same elements. Permutations deal with ordered of sets. By saying that there are $n!$ permutations of an n elements set A, we mean that there are $n!$ ways to order the elements of A.

We can more compactly represent the permutation π from the previous example using certain notations. The **two-line notation** uses, as you might expect, a two-line matrix; the first line lists the elements of the set A, and the second line lists the permuted elements $\pi(A)$.

Thus,

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 2 & 1 & 6 \end{pmatrix}.$$

Alternatively, we could represent the permutation π using **cycle notation**. In this notation, we choose a starting element and follow a "cycle" through the permutation until we arrive back at the starting element. We perform these steps until we have written each element in A. Using this notation the permutation from Example 2.2 can be written as $\pi = (1\ 3\ 5)(2\ 4)(6)$.

The number of permutations of a set with n elements is equal to the product of every natural number from 1 to n, or n-factorial $n!$. The value of the factorial of n grows very quickly as n increases. We can compare the growth rate of $n!$ to that of a function involving only n and constants using **Stirling's approximation**. For large-enough n, we have

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{\mathrm{e}}\right)^n.$$

Next we consider an extension of permutations, as defined in Definition 2.1. A ***k*-permutation** is a permutation of k elements taken from an n-element set, where $0 \le k \le n$.

Example 2.3 *Consider a set* $A = (1, 2, 3, 4)$. *Some possible* 2*-permutations of* A *are* $(1, 3)$, $(1, 2)$, $(2, 1)$, $(2, 4)$, $(3, 1)$, *and* $(4, 2)$.

Obviously, the 2-permutations of A listed in the previous example are not all that are possible. How many 2-permutations—or, more generally, how many k-permutations—exist for an n-element set?

The number of k-permutations of an n-element set, $0 \le k \le n$, is denoted $P(n, k)$. Next we want to determine this value. We already know that $P(n, n) = n!$ (Note that $P(n, n)$ is simply the number of all permutations of an n-element set.)

Theorem 2.1 *The number of* k*-permutations of a set with* n *distinct elements, where* $0 \le k \le n$, *is*

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

Proof. For all i where $0 \le i \le (k-1)$, there are $n - i$ ways to choose the ith element of the permutation. By the product rule, we have that

$$\begin{aligned} P(n,k) &= n(n-1)(n-2)\cdots(n-(k-1)) \\ &= n(n-1)(n-2)\cdots(n-k+1). \end{aligned}$$

We can rewrite the right-hand side of the above expression as

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

□

Note that, as special cases of Theorem 2.1, we have $P(n,0) = 1$ and $P(n,1) = n$ for all $n \geq 0$. As mentioned above, we also have $P(n,n) = n!$ for all $n \geq 0$.

Remark 2.1 *From Definition 2.1, we know that a permutation can be viewed as a bijective function on a finite set. From this, we can conclude that the value $P(n,n)$ counts the number of bijective functions from a set of size n to a set of size n. What, then, does the value $P(n,k)$ count? If $k < n$, then we can't have a bijection (since we can't have a surjection). Because we're mapping each of the n elements in the range to at most one of the k elements in the domain, $P(n,k)$ is counting the number of injective functions from a set of size k to a set of size n.*

Example 2.4 *A certain instructor is creating a midterm exam for his discrete mathematics class. Suppose he has a question bank containing 50 different questions. If the midterm exam consists of 5 questions, then the instructor can create a total of $P(50,5) = \frac{50!}{(50-5)!} = 254\,251\,200$ midterm exams.*

Example 2.5 *The Examinations Office is scheduling three CISC exams and three MATH exams, all of which are different. There are two days available, three exams can be held per day, and the office wants to schedule all CISC exams on one day and all MATH exams on the other day. Under such constraints, the office has $(3!)(3!)(2!)$ ways to schedule all exams: $3!$ ways of scheduling three CISC exams, $3!$ ways of scheduling three MATH exams, and $2!$ ways to arrange the "CISC exam" day and the "MATH exam" day.*

Our final example deals with an algorithmic problem that is important in computing.

Example 2.6 *The traveling salesperson problem is stated as follows: "Given a list of n cities and a list of distances between each pair of cities, what is the shortest route that both visits all cities and ends in the origin city?"*

The traveling salesperson problem is well-known for being a difficult problem for computers to solve. This is because, if we take a brute force approach to solving the problem, our solver must list all routes through each city, determine whether that route ends in the origin city, and then keep only the shortest route. Since we have n cities to visit, in the worst case this approach will require our solver to analyze $n!$ different routes.

For small values of n, this isn't bad; a brute force approach could easily solve the traveling salesperson problem for 3 or 4 cities. However, for larger n, the problem quickly grows out of hand. If we have a list of 20 cities, and if our solver checks 100 solutions per second, it would take $771\,000\,000$ years to check every solution!

2.1 Permutations with Repetition

Earlier, we learned that there exist $n!$ permutations of an n-element set. We got the $n!$ term from the fact that, after selecting the ith element of the permutation, we had $(i-1)$ elements remaining to permute. But what if we could select elements from the set more than once?

Consider the question of how many values can be stored in one byte. Since a byte is 8 bits, and since a bit can be either `0` or `1`, we were essentially constructing permutations by choosing elements from the set $\mathbb{B} = \{\mathtt{0}, \mathtt{1}\}$ more than once. It is possible to represent 2^8 values in one byte; in other words, we make 8 selections from a 2-element set and use the product rule.

Permutations with repetition, then, is just a specific application of the product rule where we're making k selections from k "copies" of the same set. (In reality, we only have one copy of the set, but it can be easier to illustrate the process by imagining multiple copies.)

Theorem 2.2 *The number of k-permutations of a set with n distinct elements, with repeti-*

tion, is n^k*.*

Proof. For all i where $0 \leq i \leq (k-1)$, there are n ways to choose the ith element of the permutation, since repetition is allowed. By the product rule, we have that $\underbrace{n \times \cdots \times n}_{k \text{ times}} = n^k$. $\square$

2.2 Permutations with Indistinguishable Elements

We now know quite a lot about permutations of sets, and we also know that sets contain only one copy of each distinct element. If we modify the definition of a set to allow for multiple copies of each element, then we get what is known as a **multiset**. For instance, $\{1, 2\}$ is a set, but $\{1, 2, 2\}$ is a multiset. Note that with multisets, just as with sets, order does not matter; the multisets $\{1, 2, 2\}$ and $\{2, 1, 2\}$ are the same multiset. However, $\{1, 2\}$ and $\{1, 2, 2\}$ are *distinct* as multisets! What can we say about numbers of permutations of multisets?

We call repeated elements within a multiset **indistinguishable**. When counting permutations of multisets, we must take care that two permutations that look identical (due to the indistinguishable elements) are not both counted as separate permutations.

In order to count the number of permutations of a set with indistinguishable objects, we begin by counting the number of permutations of the set as usual. We then account for the overcounting of identical permutations by dividing the total number of permutations by the number of ways we can permute just the indistinguishable elements.

Example 2.7 *An often used example of a word with indistinguishable letters is* MISSISSIPPI*. In this word, we have four* I*'s, four* S*'s, and two* P*'s, all of which are indistinguishable. We can "distinguish" the letters for illustrative purposes by subscripting them:*

$$\mathrm{M}\ \mathrm{I}_1\ \mathrm{S}_1\ \mathrm{S}_2\ \mathrm{I}_2\ \mathrm{S}_3\ \mathrm{S}_4\ \mathrm{I}_3\ \mathrm{P}_1\ \mathrm{P}_2\ \mathrm{I}_4.$$

If we permute two or more indistinguishable letters, then we should not get two or more different permutations. For instance, $\mathrm{MISSISSIP}_1\mathrm{P}_2\mathrm{I}$ *and* $\mathrm{MISSISSIP}_2\mathrm{P}_1\mathrm{I}$ *should be considered the same permutation.*

We have a total of 11! *ways to permute the word* MISSISSIPPI. *We then divide that by* 4!, 4!, *and* 2! *to account for the indistinguishable letters* I, S, *and* P, *respectively, which gives us a total of* $\frac{11!}{(4!)(4!)(2!)}$ *permutations.*

If we consider each indistinguishable element to be in its own "class", we can formulate this counting technique as follows.

Theorem 2.3 *The number of permutations of a set with n elements, where n_i of the elements are in class i for $1 \leq i \leq r$, is*

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

Proof. Suppose we have n_1 elements in class 1 and we have n positions in which to insert these elements. We can insert all n_1 elements in $P(n, n_1)$ ways, but ordering doesn't matter since the n_1 elements are indistinguishable, so we must divide by $P(n_1, n_1)$ to compensate for overcounting. Thus, we have

$$\begin{aligned}\frac{P(n, n_1)}{P(n_1, n_1)} &= \frac{n!/(n-n_1)!}{n_1!/(n_1-n_1)!} \\ &= \frac{n!}{n_1!(n-n_1)!}\end{aligned}$$

ways to insert the n_1 elements in class 1, and we have $(n - n_1)$ positions remaining.

If we perform the same steps for every class n_i, where $1 \leq i \leq r$, then by the product rule we have that

$$\frac{n!}{n_1!(n-n_1)!} \cdot \frac{(n-n_1)!}{n_2!(n-n_1-n_2)!} \cdot \; \cdots \; \cdot \frac{(n-n_1-\cdots-n_{r-1})!}{n_r!0!} = \frac{n!}{n_1!n_2!\cdots n_r!}.$$

□

Example 2.8 *The Examinations Office is booking 12 rooms for exams. Five CISC exams will be written (three of which are different sections of the same course), four MATH exams will be written (two of which are different sections of the same course), and three PHYS exams will be written (all of which are different courses). Different sections of the same course write the same exam, so for room-booking purposes each section is indistinguishable.*

There are $12!$ *ways to assign exams to rooms, but the office doesn't care in which room different sections write the same exam. Therefore, they must divide by the number of different sections writing the same exam. This gives a total of* $\frac{12!}{(3!)(2!)}$ *ways to book rooms for the exams.*

3 Combinations

In our discussion on permutations with indistinguishable elements, we arrived at a general formula by dividing the total number of permutations by the number of ways we could permute only the indistinguishable elements. We did so in order to avoid overcounting "identical" permutations.

If we extend the idea of indistinguishability to mean "indistinguishable up to ordering", then we obtain a new counting technique where we care only about the number of elements we take and not the order in which those elements are arranged. Instead of permutations of elements, we are considering **combinations** of elements.

Roughly speaking, combinations are just subsets in disguise.

Definition 3.1 (k-combination) *A k-combination of s set A is a size-k subset of elements from A.*

Since combinations are subsets, and since the arrangement of elements in a set doesn't matter, the arrangement of elements in combinations doesn't matter. This is the key distinction between permutations and combinations that we alluded to in the previous section: ordering matters for permutations, but not for combinations.

Further note that we always use the term ***k*-combination** when referring to a specific value or calculation. There is no such thing as a "combination of a set", since if we followed the same distinction between permutation and k-permutation, a "combination of a set" would just be the set itself. Thus, when we use the word "combination" on its own, we mean it in the broad, non-formal sense of selecting elements from a set without ordering.

Example 3.1 *Suppose we have a set $A = \{1, 2, 3, 4\}$. All of the possible 2-combinations of*

A are $\{1,2\}$, $\{1,3\}$, $\{1,4\}$, $\{2,3\}$, $\{2,4\}$, and $\{3,4\}$; exactly the same as all of the possible size-2 subsets of A.

It is incorrect for us to consider $\{1,2\}$ and $\{2,1\}$ to be distinct 2-combinations of A, since they both contain the same subset of elements from A. Therefore, we only count that particular 2-combination once.

Now that we know what combinations are, how can we count all possible k-combinations of an n-element set? This problem sounds very similar to our previous problem of counting all k-permutations of an n-element set; indeed, we can take almost the exact same approach we took when counting k-permutations. The only difference is that, since ordering doesn't matter with combinations, we need to include one additional term to guard against overcounting.

Theorem 3.1 *The number of k-combinations of a set with n elements, where $0 \leq k \leq n$, is*

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

Proof. We can count the number of k-combinations of a set with n elements by first calculating the number of k-permutations of the same set, and then dividing by the number of permutations of a k-element set. The division is necessary because the ordering of the k elements does not matter. Thus, we have

$$\begin{aligned} C(n,k) &= \frac{P(n,k)}{P(k,k)} \\ &= \frac{n!/(n-k)!}{k!/(k-k)!} \\ &= \frac{n!}{k!(n-k)!}. \end{aligned}$$

□

Note that, as special cases of Theorem 3.1, we have $C(n,0) = 1$, $C(n,1) = n$, and $C(n,n) = 1$ for all $n \geq 0$. The values $C(n,0)$ and $C(n,n)$ should make sense from a set-theoretic standpoint, since for any n-element set A, there is only one zero-element subset ($\emptyset$) and only one n-element subset (the set A itself).

Before we continue, it is worthwhile to point out an interesting symmetry property of combinations that may help us to solve some problems. In a k-combination, we take k elements from an n-element set. However, this is no different from us taking $(n-k)$ elements and leaving them out of our final choice. From this observation, we get the aforementioned property.

Theorem 3.2 *For all natural numbers n and k, where $0 \leq k \leq n$,*

$$C(n,k) = C(n,n-k).$$

Proof. Recall that $C(n,k) = \frac{n!}{k!(n-k)!}$. Substituting $(n-k)$ for k, we get

$$\begin{aligned} C(n,n-k) &= \frac{n!}{(n-k)!(n-(n-k))!} \\ &= \frac{n!}{(n-k)!k!}, \end{aligned}$$

and hence $C(n,k) = C(n,n-k)$. □

A common question to hear from students by this point is "how can we tell whether we need to calculate permutations or combinations in a problem?" It's certainly a reasonable question to ask, and you won't be faulted for wondering this yourself. Unfortunately, there is no surefire trick for determining which counting technique to use, apart from determining whether the problem statement emphasizes ordering of elements. Thus, if a question asked

> In how many ways can we choose 3 faculty members from a department of 10 faculty members?

then we would use combinations, since we care about only the number of faculty members. If, on the other hand, a question asked

> In how many ways can we line up 10 faculty members for a department photo?

then we would use permutations, since we care about both the number and the ordering of faculty members.

When in doubt, remember this mnemonic: **c**ombinations are for **c**hoosing some number of elements, and **p**ermutations are for **p**lacing those elements in a specific order.

Example 3.2 *A group of five students are begging their discrete mathematics instructor to come up with examples that don't involve writing or scheduling exams. If the group chooses three representatives to talk with the instructor during office hours, how many possible combinations of representatives are there?*

Suppose we call the students Alice, Bob, Carol, David, and Eve. If, for instance, Alice, Bob, and Carol attend the office hours, then that combination is no different than if Carol, Bob, and Alice attend; the representatives are the same. Altogether, we have the following subsets of three representatives each:

$$ABC,\ ABD,\ ABE,\ ACD,\ ACE,\ ADE,\ BCD,\ BCE,\ BDE,\ CDE.$$

In other terms, we have that $C(5,3) = \frac{5!}{3!(5-3)!} = \frac{5!}{3!2!} = 10$ *combinations.*

Example 3.3 *Recall that a byte consists of eight binary digits. Call a byte "balanced" if it contains an equal number of* `0` *bits and* `1` *bits.*

How many balanced bytes exist? We can frame this problem in the following way. Since a balanced byte contains an equal number of `0`*'s and* `1`*'s, we must have four occurrences of each bit within a balanced byte. Thus, if we start with a blank byte with eight spaces, and we fill four of those spaces with* `0`*'s, then we are forced to fill the remaining four spaces with* `1`*'s.*

In how many ways can we fill four spaces with `0`*'s? This question is equivalent to asking in how many ways we can choose four spaces out of eight. This gives* $C(8,4) = \frac{8!}{4!(8-4)!} = \frac{8!}{4!4!} = 70$*, so there exist a total of 70 balanced bytes.*

3.1 Combinations with Repetition

Just like with permutations, it is possible for us to calculate the number of k-combinations of a set when we are able to select elements from the set more than once. Unlike k-combinations without repetition, here we are able to select more copies of elements in our k-combination than those that are in the original set. Thus, a k-combination with repetition is not necessarily a subset of the original set.

How do we illustrate the process of taking a k-combination with repetition? Instead of taking and replacing elements of the set to form the k-combination, we will work "in reverse" by writing out how many copies of each element our k-combination will contain. We represent this scenario in a uniquely American style: using the so-called **stars and bars** method. (Incidentally, the mathematician who popularized this method—William Feller—was born in Croatia, not America.)

With the stars and bars method, we partition our set into classes using bars (denoted $|$). Each class corresponds to a distinct element in the set. We then represent the number of copies of elements in that class to be included in our k-combination using the appropriate number of stars (denoted $\star$).

Example 3.4 *A student is enrolling in courses for the upcoming academic year. They plan to enrol in 10 courses. If the courses are to be selected from the set* $\{$*CISC, MATH, PHYS, BIOL, CHEM*$\}$*, then we can partition these five classes of course codes using four bars:*

$$\underbrace{\quad}_{\textit{CISC}} \mid \underbrace{\quad}_{\textit{MATH}} \mid \underbrace{\quad}_{\textit{PHYS}} \mid \underbrace{\quad}_{\textit{BIOL}} \mid \underbrace{\quad}_{\textit{CHEM}}$$

The student wants to take five CISC courses, three MATH courses, one PHYS course, and one CHEM course. If we denote one course by one star, our "stars and bars diagram" will look like the following:

$$\star\star\star\star\star \mid \star\star\star \mid \star \mid\mid \star$$

The above "star and bars" diagram illustrates one possible 10-combination (with repetition) of a set with 5 elements.

As the previous example illustrates, the number of elements in the set and the number of selections we make dictate the number of stars and bars that are available to us. If our set contains i elements and we wish to make j selections, then we will have j stars and $(i-1)$ bars. As we also saw in the previous example, it is perfectly fine to have zero stars in a given partition; this just means we made no selections of elements from the corresponding class.

Using stars and bars in this way, it becomes evident that the number of ways to form a k-combination with repetition from a set of n elements is exactly the same as the number of ways to arrange k stars and $(n-1)$ bars in a row; that is, $(k+n-1)!$. However, since the stars and bars are indistinguishable, we must divide by both $k!$ and $(n-1)!$ to avoid overcounting.

Theorem 3.3 *The number of k-combinations of a set with n elements, with repetition, is*

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

Proof. Let $A = \{a_1, a_2, \ldots, a_n\}$. Consider a string of $(k+n-1)$ blank spaces

$$\underbrace{\sqcup\ \sqcup\ \sqcup\ \cdots\ \sqcup\ \sqcup\ \sqcup}_{k+n-1 \text{ times}}$$

and a set containing k $\star$'s and $(n-1)$ $|$'s. Each arrangement of $\star$'s and $|$'s into the blank spaces constitutes a k-combination with repetition, with the number of $\star$'s between the start of the string and the first $|$ counting the number of selections of a_1, the number of $\star$s between the first $|$ and second $|$ counting the number of selections of a_2, and so on. There is a total of $C(k+n-1, n-1)$ ways to place the $(n-1)$ $|$'s into the blank spaces, and from this we force placement of the k $\star$s. Thus, there is a total of $C(k+n-1, n-1) = C(k+n-1, k) = \frac{(k+n-1)!}{k!(n-1)!}$ possible k-combinations of a set with n elements where repetition is allowed. □

Returning to our course enrolment example, we see that if the student had no constraints on the 10 courses they wanted to take, then they would have a total of $C(10+5-1, 10) = C(14, 10) = C(14, 4) = 1001$ course combinations to choose from. Note that the 10-combination counts the number of ways to select the "types" of the 10 courses from the set:

$$\{\text{CISC, MATH, PHYS, BIOL, CHEM}\}$$

as opposed to selecting specific courses.

We can use k-combinations with repetition to calculate many other interesting things; consider, for example, a computer algebra system that needs to find solutions to a given equation. The system could naïvely check every possible solution, but this can be slow. Using combinations, certain possibilities can be ruled out based on constraints or other conditions.

Example 3.5 *Let x, y, and z be natural numbers. How many solutions exist for the equation $x + y + z = 16$?*

We can frame this problem as a combinatorial problem, since any solution to the given equation corresponds to a selection of 16 elements from a set of size 3 where we have x elements from class 1, y elements from class 2, and z elements from class 3. Drawing a "stars and bars diagram", we have the following scenario:

$$\underbrace{\qquad}_{x} \mid \underbrace{\qquad}_{y} \mid \underbrace{\qquad}_{z}$$

Since we are calculating the number of 16-combinations of a set with 3 elements, we get that the total number of solutions to the equation is $C(16+3-1, 16) = C(18, 16) = C(18, 2) = 153$.

Example 3.6 *Let x, y, and z be natural numbers, this time with the constraints that $x \geq 2$, $y \geq 5$, and $z \geq 3$. How many solutions exist for the equation $x + y + z = 16$?*

This example is very similar to the previous example, but with added constraints. Thus, we can follow the same procedure as before while keeping in mind that each of x, y, and z must take on certain values; namely, we must have at least two elements from class 1, at least five elements from class 2, and at least three elements from class 3. Drawing a "stars and bars diagram", we have the following scenario:

$$\underbrace{\star\star}_{x} \mid \underbrace{\star\star\star\star\star}_{y} \mid \underbrace{\star\star\star}_{z}$$

Since ten stars are preassigned to the diagram, we must place the remaining six stars ourselves. This is equivalent to us calculating the number of 6-combinations of a set with 3 elements, which leads us to conclude that the total number of constrained solutions to the equation is $C(6 + 3 - 1, 6) = C(8, 6) = C(8, 2) = 28$.

4 Binomial Theorem

Let's now take a brief step back from counting and look at an algebraic problem: expanding binomials. As you likely learned in your first-year math classes (or even earlier), we can use the FOIL method—first, outer, inner, last—to expand the binomial $(x + y)^2$. This gives us the following result:

$$\begin{aligned}(x+y)^2 &= (x+y)(x+y)\\ &= x^2+xy+xy+y^2\\ &= x^2+2xy+y^2.\end{aligned}$$

Can we follow a similar technique for binomials with larger exponents? Of course; the FOIL method is just a special case of the distributive property of multiplication, which tells us that $a(b+c)=(ab+ac)$ for values a, b, and c. To see how this works, let's consider $(x+y)^3$:

$$\begin{aligned}(x+y)^3 &= (x+y)(x+y)(x+y)\\ &= (x^2+2xy+y^2)(x+y)\\ &= x^3+x^2y+2x^2y+2xy^2+xy^2+y^3\\ &= x^3+3x^2y+3xy^2+y^3.\end{aligned}$$

Just for fun, let's also consider $(x+y)^4$ while we're at it:

$$\begin{aligned}(x+y)^4 &= (x+y)(x+y)(x+y)(x+y)\\ &= (x^3+3x^2y+3xy^2+y^3)(x+y)\\ &= x^4+x^3y+3x^3y+3x^2y^2+3x^2y^2+3xy^3+xy^3+y^4\\ &= x^4+4x^3y+6x^2y^2+4xy^3+y^4.\end{aligned}$$

By this point, you might notice a certain pattern is developing. Each term in the expansion of the binomial $(x+y)^n$ is of the form ax^by^c, where $b+c=n$ and where a is some coefficient.

How can we calculate the value of the coefficient a for some arbitrary term without writing the entire expansion? Let's begin by determining what this value a is counting. We can write the general binomial $(x+y)^n$ as

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

By the distributivity property of multiplication, the expansion of this binomial will contain one term for each possible choice of x and y, and we take n choices. As an illustration, assume we are only choosing x. If we take x a total of n times, then we will add the term x^n to our expansion. On the other hand, if we take x a total of $n-4$ times, then we must take y a total of 4 times in order to collect n terms overall. Thus, we will add the term $x^{n-4}y^4$ to our expansion.

Generalizing this idea to us choosing x a total of b times and y a total of c times, where $b+c=n$, we see that the idea is equivalent to us calculating the number of ways we can choose b occurrences of x from n binomials (equivalently, choosing c occurrences of y). In other words, we're taking a b-combination from a set of binomials of size n (equivalently, a c-combination), and therefore, the value a is equal to $C(n,b)=C(n,c)$.

Before we continue, we will introduce a new notation used specifically in the context of binomials. We say that the **binomial coefficient** $\binom{n}{k}$ is the number of ways to choose k elements from an n-element set, where $0 \le k \le n$. Sound familiar? It should; the binomial coefficient is exactly the same as a k-combination, but written using a different notation.

Definition 4.1 (Binomial coefficient) *The binomial coefficient* $\binom{n}{k}$*, read as "n choose k", is defined for* $0 \le k \le n$ *as*

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

Now that we are familiar with binomial coefficients, we may use this notation to obtain the general form of a binomial expansion. We obtain the general form by way of the **binomial theorem**.

Theorem 4.1 (Binomial theorem) *Let x and y be variables, and let n be a natural number. Then*

$$\begin{aligned}(x+y)^n &= \sum_{i=0}^{n} \binom{n}{i} x^{n-i} y^i \\ &= \binom{n}{0}x^n + \binom{n}{1}x^{n-1}y + \binom{n}{2}x^{n-2}y^2 + \cdots + \binom{n}{n-1}xy^{n-1} + \binom{n}{n}y^n.\end{aligned}$$

Proof. We prove by induction. Let $P(n)$ be the statement "$(x+y)^n = \sum_{i=0}^{n}\binom{n}{i}x^{n-i}y^i$".

When $n=1$, we have $(x+y)^1 = x+y = \binom{1}{0}x^1 + \binom{1}{1}y^1 = \sum_{i=0}^{1}\binom{1}{i}x^{1-i}y^i$. Therefore, $P(1)$ is true.

Assume that $P(k)$ is true for some $k \in \mathbb{N}$. That is, assume that $(x+y)^k = \sum_{i=0}^{k} \binom{k}{i} x^{k-i} y^i$.

We now show that $P(k+1)$ is true. Multiply each side of the equation by $(x+y)$ to get

$$\begin{aligned}
(x+y)^{k+1} &= \left(\sum_{i=0}^{k} \binom{k}{i} x^{k-i} y^i \right) (x+y) \\
&= x \left(\sum_{i=0}^{k} \binom{k}{i} x^{k-i} y^i \right) + y \left(\sum_{i=0}^{k} \binom{k}{i} x^{k-i} y^i \right) \\
&= \sum_{i=0}^{k} \binom{k}{i} x^{k+1-i} y^i + \sum_{i=0}^{k} \binom{k}{i} x^{k-i} y^{i+1} \\
&= \binom{k}{0} x^{k+1} + \binom{k}{k} y^{k+1} + \sum_{i=1}^{k} \binom{k}{i} x^{k+1-i} y^i + \sum_{i=0}^{k-1} \binom{k}{i} x^{k-i} y^{i+1} \\
&= \binom{k}{0} x^{k+1} + \binom{k}{k} y^{k+1} + \sum_{i=1}^{k} \binom{k}{i} x^{k+1-i} y^i + \sum_{i=1}^{k} \binom{k}{i-1} x^{k-i} y^{i+1} \\
&= \binom{k}{0} x^{k+1} + \binom{k}{k} y^{k+1} + \sum_{i=1}^{k} \left(\binom{k}{i} + \binom{k}{i-1} \right) x^{k+1-i} y^i \\
&= \binom{k+1}{0} x^{k+1} + \binom{k+1}{k+1} y^{k+1} + \sum_{i=1}^{k} \binom{k+1}{i} x^{k+1-i} y^i \\
&= \sum_{i=0}^{k+1} \binom{k+1}{i} x^{k+1-i} y^i.
\end{aligned}$$

Therefore, $P(k+1)$ is true.

By the principle of mathematical induction, $P(n)$ is true for all $n \in \mathbb{N}$. □

Remark 4.1 *It is possible to generalize binomial coefficients and the binomial theorem to polynomials with more than two terms, such as $(x+y+z)^n$. These generalizations are called multinomial coefficients and the multinomial theorem, respectively. As an exercise, think about how to formulate these generalizations.*

Immediately from the statement of the binomial theorem, we get a variant of the theorem as a corollary.

Corollary 4.1 *Let x be a variable and let n be a natural number. Then*

$$(x+1)^n = \sum_{i=0}^{n} \binom{n}{i} x^i.$$

Proof. Follows from the binomial theorem when $y = 1$. $\square$

In the proof of the binomial theorem, we require a particular identity that tells us something about the value of a binomial coefficient in terms of smaller binomial coefficients. Using this identity, which was named after the French mathematician Blaise Pascal, we are able to define binomial coefficients recursively, which is a great help in computational applications.

Theorem 4.2 (Pascal's identity) *For all* $1 \leq k \leq n$*,*

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

Proof. By the definition of the binomial coefficient, we have

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

$\square$

We can reason about Pascal's identity in the following way. Suppose S is a set containing $n+1$ elements, and denote one "special" element as a. Let T be the subset of S not containing a. There are $\binom{n+1}{k}$ size-k subsets of S, and these subsets either (i) do not contain a, but only contain k elements from T, or (ii) contain both a and $k-1$ elements from T. In scenario (i), there are $\binom{n}{k}$ possible subsets of T, and in scenario (ii), there are $\binom{n}{k-1}$ possible subsets of T, so altogether we have $\binom{n}{k} + \binom{n}{k-1}$ size-k subsets of S.

With Pascal's identity, we can draw a beautiful triangular arrangement of binomial coefficients where the kth term in row $n+1$, $\binom{n+1}{k}$, is derived from the sum of the two terms $\binom{n}{k-1}$ and $\binom{n}{k}$ written directly above it. (Blank or nonexistent entries are taken to be zero.) This arrangement is called **Pascal's triangle**, in spite of the fact that it had been studied centuries before by other mathematicians. The first few rows of Pascal's triangle are as follows:

$$
\begin{array}{ccccccccccccc}
 & & & & & & \binom{0}{0} & & & & & & \\
 & & & & & \binom{1}{0} & & \binom{1}{1} & & & & & \\
 & & & & \binom{2}{0} & & \binom{2}{1} & & \binom{2}{2} & & & & \\
 & & & \binom{3}{0} & & \binom{3}{1} & & \binom{3}{2} & & \binom{3}{3} & & & \\
 & & \binom{4}{0} & & \binom{4}{1} & & \binom{4}{2} & & \binom{4}{3} & & \binom{4}{4} & & \\
 & \binom{5}{0} & & \binom{5}{1} & & \binom{5}{2} & & \binom{5}{3} & & \binom{5}{4} & & \binom{5}{5} & \\
\binom{6}{0} & & \binom{6}{1} & & \binom{6}{2} & & \binom{6}{3} & & \binom{6}{4} & & \binom{6}{5} & & \binom{6}{6}
\end{array}
\quad = \quad
\begin{array}{ccccccccccccc}
 & & & & & & 1 & & & & & & \\
 & & & & & 1 & & 1 & & & & & \\
 & & & & 1 & & 2 & & 1 & & & & \\
 & & & 1 & & 3 & & 3 & & 1 & & & \\
 & & 1 & & 4 & & 6 & & 4 & & 1 & & \\
 & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & \\
1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1
\end{array}
$$

Aside from looking nice, Pascal's triangle reveals many hidden intricacies in the structure of and relationships between binomial coefficients. We present a few of these interesting results here without proof.

Proposition 4.1 *The following identities hold:*

1. $\binom{r}{k} = \binom{r}{r-k}$ *for all* $0 \leq k \leq r$ *(row symmetry);*
2. $\sum_{k=0}^{r} \binom{r}{k} = 2^r$ *for all* $r \geq 0$ *(row sum);*
3. $\sum_{k=0}^{r} \binom{r}{k}^2 = \binom{2r}{r}$ *for all* $r \geq 0$ *(row sum of squares);*
4. $\sum_{k=0}^{r} \binom{k}{c} = \binom{r+1}{c+1}$ *for all* $r, c \geq 0$ *(column sum).*

All things considered, what do binomial coefficients and the binomial theorem have to do with computing? For one, just as we saw in our discussion on combinations with repetition, the binomial theorem can speed up certain calculations in computer algebra systems. Multiplication is an intensive operation for a computer, and many multiplications at once can slow down a computation considerably. With the binomial theorem, we can expand certain expressions or calculate certain terms within an expression much more efficiently.