Basic counting functions

In this section we shall be concerned with finding the cardinality, that is, counting the number of elements in sets specified in various ways. To motivate the ideas, we can consider the following example.

X needs to go from Philadelphia to New York via Trenton. From Philadelphia to Trenton X must use a Septa train and there are exactly three of these, namely, trains 1, 2, and 3. From Trenton to New York, X must use a New Jersey Transit train and there are exactly two of these, namely trains A and B. How manty distinct ways are there of making the journey.

First, we must define the notion of distinctness of journeys. Then we have three separate questions:

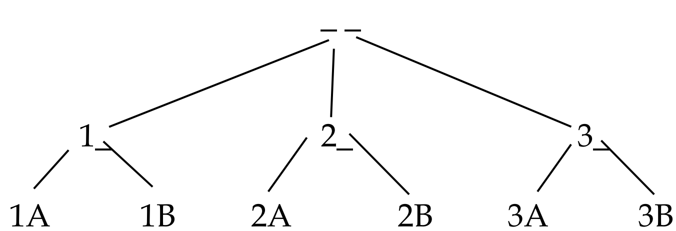
0 how to generate all journeys systematically,

1 how to list all journeys systematically, and

2 how to count the number of distinct ways with or without knowing the answers to 0 and 1.

Note that each journey may be coded as a two-letter word ab where $a \in \{1, 2, 3\}$ and $b \in \{A, B\}$. In other words, each journey may be regarded as an element of the cartesian product $\{1, 2, 3\} \times \{A, B\}$. For example, the word $2B$ codes the journey in which X uses the Septa train 2 in the first leg of the journey and the New Jersey Transit train B in the second leg of the journey. When we studied sets, we already studied a systematic procedure for generating all members of a cartesian product and this was to construct a tree. We shall do exactly the same here. The only difference is that we write the members of the final set as words and not as ordered pairs. We exhibit the tree below.

Tree



We use the tree to systematically generate all the possible journeys and to list them. The set of journeys is $\begin{Bmatrix} 1A, & 1B, \\ 2A, & 2B, \\ 3A, & 3B \end{Bmatrix}$. Now we may count and see that there are six distinct journeys.

Note that two journeys are distinct if and only of their codes are distinct as words. One could have counted the number of distinct journeys as the cardinality of the set $\{1,2,3\} \times \{A,B\}$.

$$\begin{aligned} & \nu\left(\{1,2,3\} \times \{A,B\}\right) \\ = \quad & \left(\nu\left(\{1,2,3\}\right)\right)\left(\nu\left(\{1,2,3\}\right)\right) \\ = \quad & (3)(2) \\ = \quad & 6 \end{aligned}$$

This section is devoted to investigating ways of systematically generating, listing, and counting elements of sets defined in various ways. Certain functions arise in the process of this investigation and we study these functions in the following pages. Trees serve as a useful tool for addressing these questions.

Note that we start off with a certain number of empty slots and then fill them in from left to right, one slot at a time making as many branches as there are possibilities. Each time we move from one slot to the next the tree develops another level.

R In general we shall be concerned with generating, lising, and counting the total number of possible solutions to some problem. Generation has to be systematic and efficient; it is usually acomplished using some sort of a tree. Listing needs to systematic as well and usually invloves ordering the elements of the last level of the tree. Counting can be done by hand at the last level, but sometimes it is possible to count without generating or listing. This requires applying various ideas and this process is not algorithmic.

D We define $\forall p, q \in \mathbb{N} \quad p..q \coloneqq \{n \in \mathbb{N} | p \leq n \leq q\}$and $a \in \{1, 2, 3\}$

E Note $3..6 \coloneqq \{n \in \mathbb{N} | 3 \leq n \leq 6\} = \{3, 4, 5, 6\}$

D BC ≔ Base Case

D RcS ≔ Recursive Step.

D The factorial of a number $n \in \mathbb{N}$, written $n!$, is defined recursively as follows:

BC $0! \coloneqq 1$

RcS $\forall n \in \mathbb{N} \quad (n+1)! \coloneqq (n+1)(n!)$

T $\forall n \in \mathbb{N} \quad n! \coloneqq \prod_{k \in 1..n} k$

D Using the above, we may define the factorial function:

$$Fctrl \coloneqq n \mapsto n! : \mathbb{N} \longrightarrow \mathbb{N}$$

E In this example we use φ as a shorthand for $Fctrl$ and rewrite the recursive definition in the terms of φ and then show how one calculates its values using the recursive procedure.

BC $\varphi(0) \coloneqq 1$

RcS $\forall n \in \mathbb{N} \quad \varphi(n+1) \coloneqq (n+1)\varphi(n)$

To calculate $\varphi(1)$, we may use the recursive definition and proceed as follows. We shall repeatedly use Euclid's rule, namley: **equals substituted into equals yield equals**.

The first task is to match $\varphi(1)$ to the left or the right side of one of the defining equaitons. We shall show the match explicitly.

$$\varphi(1) = \varphi(0+1) = \left(\varphi(n+1)\right)\left\langle n \quad \longleftarrow \quad 0\right\rangle$$

The above means that $\varphi(1)$ may be obtained by substituting 0 in the place of n in the expression $\left(\varphi(n+1)\right)$. Since $\varphi(n+1)$ is equal by definition to $(n+1)\varphi(n)$, we may, using Euclid's rule, substitute 0 in the expression $(n+1)\varphi(n)$ to obtain a new value $\varphi(n+1)$. This will reduce the argument of φ to 0 and this is to be used repeatedly to compute $\varphi(n)$.

We record the process as follows.

$$\begin{aligned}
& \varphi(1) \\
= \quad & \varphi(0+1) \\
= \quad & \Big(\varphi(n+1)\Big)\Big\langle n \quad \longleftarrow \ 0 \Big\rangle \\
= \quad & \Big((n+1)\varphi(n)\Big)\Big\langle n \quad \longleftarrow \ 0 \Big\rangle \\
= \quad & (0+1)\varphi(0) \\
= \quad & (1)1 && \Big(\because \varphi(0) \coloneqq 1\Big) \\
= \quad & 1
\end{aligned}$$

Similarly,

$$\begin{aligned}
& \varphi(2) \\
= \quad & \varphi(1+1) \\
= \quad & \Big(\varphi(n+1)\Big)\Big\langle n \quad \longleftarrow \ 1 \Big\rangle \\
= \quad & \Big((n+1)\varphi(n)\Big)\Big\langle n \quad \longleftarrow \ 1 \Big\rangle \\
= \quad & (1+1)\varphi(1) \\
= \quad & (2)1 && \Big(\because \varphi(1) \coloneqq 1\Big) \\
= \quad & 2! \\
= \quad & 2
\end{aligned}$$

Similarly, $\quad \varphi(3)$

$$= \quad \varphi(2+1)$$

$$= \quad \Big(\varphi(n+1)\Big)\Big\langle n \quad \leftarrow \; 2\Big\rangle$$

$$= \quad \Big((n+1)\varphi(n)\Big)\Big\langle n \quad \leftarrow \; 2\Big\rangle$$

$$= \quad (2+1)\varphi(2)$$

$$= \quad (3)(2)1 \qquad\qquad \Big(\because \varphi(2) \coloneqq (2)1\Big)$$

$$= \quad 3!$$

$$= \quad 6$$

R Note that if we use the theorem: $\forall n \in \mathbb{N}, \varphi(n) = n! \coloneqq \prod_{k \in 1..n} k$, it is much easier to compute, for instance,

$$3!$$

$$= \quad (3)(2)(1)$$

$$= \quad 6$$

The recursive procedure serves as a method of telling computers how to compute.

D The basic counting functions $P(n,k)$ and $C(n,k)$ are defined as follows:

D $\forall n \in \mathbb{N} \;\; \forall k \in 0..n$

$$P(n,k) \quad \coloneqq \quad \frac{(n!)}{\big((n-k)!\big)}$$

D $\forall n \in \mathbb{N} \;\; \forall k \in 0..n$

$$C(n,k) \quad \coloneqq \quad \frac{(n!)}{(k!)\big((n-k)!\big)} \quad = \quad \frac{P(n,k)}{P(k,k)}$$

D $\forall n \in \mathbb{N} \;\; \forall k \in 0..n$

$$\binom{n}{k} \qquad := \qquad \frac{(n!)}{(k!)\big((n-k)!\big)} \qquad = \qquad C(n,k)$$

R The two examples that follow exhibit two distinct methods of bookkeeping for calculations that involve substitution into formulas.

E Compute $P(3,2)$

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

$$\underline{\hspace{8cm}}\left\langle \begin{matrix} n \leftarrow 3 \\ \\ k \leftarrow 2 \end{matrix} \right\rangle$$

$$P(3,2) = \frac{(3!)}{\big((3-2)!\big)}$$

$$P(3,2) = \frac{3!}{1!}$$

$$P(3,2) = \frac{3(2)1}{1}$$

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

E Compute $C(3,2)$

$$C(3,2)$$

$$= \Big(C(n,k)\Big)\left\langle \begin{matrix} n \leftarrow 3 \\ \\ k \leftarrow 2 \end{matrix} \right\rangle$$

$$= \left(\frac{(n!)}{(k!)\big((n-k)!\big)}\right)\left\langle \begin{matrix} n \leftarrow 3 \\ \\ k \leftarrow 2 \end{matrix} \right\rangle$$

$$= \left(\frac{(3!)}{(2!)(3-2!)}\right)$$

$$= \left(\frac{(3)}{(2!)(1!)}\right)$$

$$= \frac{(3)\cancel{(2)(1)}}{\cancel{(2)(1)}(1)}$$

$$= 3$$

D $\quad \forall n \in \mathbb{N} \;\; \forall k, n_1, n_2, \cdots n_p \in 0..n \quad$ with $\;\sum_{k=1}^{p} n_k = n$

$$P(n; n_1, n_2, \cdots, n_p) \quad \coloneqq \quad \frac{n!}{(n_1!)(n_2!)(\cdots)(n_p!)}$$

$$\binom{n}{n_1 \quad n_2 \quad \cdots \quad n_p} \quad \coloneqq \quad \frac{n!}{(n_1!)(n_2!)(\cdots)(n_p!)}$$

T $\quad$ Given a permutation $\rho: 1..p \to 1..p$,

$$\binom{n}{n_1 \quad n_2 \quad \cdots \quad n_p} = \binom{n}{n_{\rho(1)} \quad n_{\rho(2)} \quad \cdots \quad n_{\rho(p)}}$$

R $\quad$ The above means that value of $\binom{n}{n_1 \quad n_2 \quad \cdots \quad n_p}$ does not change if the entries in the lower row of $\binom{n}{n_1 \quad n_2 \quad \cdots \quad n_p}$ are permuted. For example:

$$\binom{6}{1 \quad 2 \quad 3} = \binom{6}{1 \quad 3 \quad 2} = \binom{6}{2 \quad 1 \quad 3} = \binom{6}{2 \quad 3 \quad 1} = \binom{6}{3 \quad 1 \quad 2} = \binom{6}{3 \quad 2 \quad 1}$$

E $\quad$ Compute $\binom{4}{1 \quad 1 \quad 2 \quad 1}$

Since $1 + 1 + 2 + 1 = 5 \neq 4$, $\binom{4}{1 \quad 1 \quad 2 \quad 1}$ does not make sense.

E $\quad$ Compute $\binom{4}{1 \quad 0 \quad 2 \quad 1}$

Since $1 + 0 + 2 + 1 = 4$, $\quad \binom{4}{1 \quad 0 \quad 2 \quad 1}$

$$= \frac{4!}{(1!)(0!)(2!)(1!)}$$

$$= \frac{(4)(3)\cancel{(2)}\cancel{(1)}}{(1)(1)\left(\cancel{(2)}\cancel{(1)}\right)(1)}$$

$$= \quad 12$$

R In the last example, we did not show substitution explicitly. **On any exam you will have to show substitution explicitly following the procedure explained in the two examples immediately preceding the last one**.

R The following theorem says that these counting functions expectedly have only natural numbers as outputs.

Q Compute the following:

$P(4,2), C(4,2), P(4;1,2,1), P(P(4,2),2), C\big(P(4,2),P(4,2)\big), P\big(C(4,2),C(4,2)\big),$

$\binom{P(4,2)}{C(4,2)}, \begin{pmatrix} & 5 & \\ 2 & 1 & 2 \end{pmatrix}$

R We recall the following two theorems from basic algebra:

$$(0) \quad \frac{a-b \geq 0}{a \geq b} \qquad\qquad (1) \quad \frac{b \neq 0 \quad \frac{a}{b} \geq 1}{a \geq b}$$

We shall use these results in what follows:

T $\quad \forall n \in \mathbb{N} \;\; \forall k, n_1, n_2, \cdots n_p \in 0..n \qquad P(n,k) \in \mathbb{N}$

$$C(n,k) \in \mathbb{N}$$

$$P\big(n; n_1, n_2, \cdots, n_p\big) \in \mathbb{N}$$

T $\quad \forall n \in \mathbb{N} \;\; \forall k \in 0..n \qquad P(n,k) \geq C(n,k)$

SPf

$$\frac{LS}{RS}$$

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

$$= \quad P(n,k) \div C(n,k)$$

$$= \left(\frac{(n!)}{((n-k)!)}\right) \div \left(\frac{(n!)}{(k!)((n-k)!)}\right)$$

$$= \left(\frac{\cancel{(n!)}}{\cancel{((n-k)!)}}\right)\left(\frac{(k!)\cancel{((n-k)!)}}{\cancel{(n!)}}\right)$$

$$= k!$$

$$\geq 1$$

Hence, $LS \geq RS$

Hence, $P(n,k) \geq C(n,k)$

EPf

Q Prove the assertions in the following theorem by mimicking the method of proof in the preceding example. Prove each of the assertions by induction once you cover the material on induction and recursion. Find all the hidden assumptions in the preceding proof and investigate it for possible gaps.

T $\forall n \in \mathbb{N} \;\; \forall k \in 0..n$

0 $P(n,0) = 1$

1 $P(n,1) = n$

2 $P(n,n) = P(n,n-1) = n!$

3 $C(n,0) = C(n,n) = 1$

4 $C(n,1) = C(n,n-1) = n$

D Using the above we may define two counting partial-functions:

0 $P := (n,k) \mapsto P(n,k) : \mathbb{N} \times \mathbb{N} \dashrightarrow \mathbb{N}$

1 $C := (n,k) \mapsto C(n,k) : \mathbb{N} \times \mathbb{N} \dashrightarrow \mathbb{N}$

R The above may be defined as counting functions by truncating the source to the subset of $\mathbb{N} \times \mathbb{N}$, $\{(n,k) \in \mathbb{N} \times \mathbb{N} | k \leq n\}$ on which they are defined; thus we have counting functions:

$$(1) \qquad P := (n,k) \mapsto P(n,k) : \{(n,k) \in \mathbb{N} \times \mathbb{N} | k \leq n\} \longrightarrow \mathbb{N}$$

$$(2) \qquad C := (n,k) \mapsto C(n,k) : \{(n,k) \in \mathbb{N} \times \mathbb{N} | k \leq n\} \longrightarrow \mathbb{N}$$

R An alternative to the above is to define the output at every input belonging to $\{(n,k) \in \mathbb{N} \times \mathbb{N} | k < n\}$ to be $\perp$ in which case we have two counting functions:

$$(1) \qquad P := (n,k) \mapsto P(n,k) : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N} \cup \{\perp\}$$

$$(2) \qquad C := (n,k) \mapsto C(n,k) : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N} \cup \{\perp\}$$

<u>Combinatorial Interpretations</u>

R $P(n,k)$ counts the number of *permutations* of n distinct objects taken k at a time, that is, number of k −letter words in an alphabet consisting of n distinct letters, each in infinite supply, such that order matters (OM) and repetition is not allowed ($\cancel{RA}$).

R $C(n,k)$ is the number of *combinations* of n distinct objects taken k at a time, that is, number of k −letter words in an alphabet consisting of n distinct letters, each in infinite supply, such that order does not matter ($\cancel{OM}$) and repetition is not allowed ($\cancel{RA}$).

R $P\left(n; n_1, n_2, \cdots, n_p\right)$ is the number of *permutations* of p distinct objects where there are $\forall k \in 1..p \;\; n_k$ identical objects of type k, that is, number of n −letter words in an alphabet $\mathcal{A} := \left(a_1 \colon n_1, a_2 \colon n_2, \cdots, a_p \colon n_p\right)$, each in finite supply as indicated, such that order matters and repetition is allowed. Note that we must have $\sum_{k=1}^{p} n_k = n$ for $P\left(n; n_1, n_2, \cdots, n_p\right)$ to be defined.

E Prove that $\forall\, n \,\epsilon \mathbb{N}\ \forall k\ \epsilon\ 0..n$ $\binom{n}{k} = \binom{n}{n-k}$

SPf $\forall\, n \,\epsilon \mathbb{N}\ \forall k\ \epsilon\ 0..n$ RS

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

$$= \binom{n}{k}\left\langle k \longleftarrow (n-k)\right\rangle$$

$$= \left(\frac{(n!)}{(k!)\big((n-k)!\big)}\right)\left\langle k \longleftarrow (n-k)\right\rangle$$

$$= \frac{(n!)}{\big((n-k)!\big)\Big(\big(n-(n-k)\big)!\Big)}$$

$$= \frac{(n!)}{\big((n-k)!\big)\big((n-n+k)!\big)}$$

$$= \frac{(n!)}{\big((n-k)!\big)\big((0+k)!\big)}$$

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

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

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

$$= LS$$

EPf

R The above equality is the first on the list of combinatorial identities.

Q Prove the remaining combinatorial identities in a manner similar to the above.

Combinatorial Identities

$$\forall m, n, p \in \mathbb{N} \quad \forall k \in 0..n \quad \forall l \in 0..k$$

0. Symmetry $\binom{n}{k} = \binom{n}{n-k}$

1. Absorption $\binom{n}{k} = \left(\frac{n}{k}\right)\binom{n-1}{k-1} \quad (k \neq 0)$

2. Trinomial Revision $\binom{n}{l}\binom{n-l}{k-l} = \binom{k}{l}\binom{n}{k}$

3. Parallel Summation $\sum_{k=0}^{n}\binom{m+k}{k} = \binom{m+n+1}{n}$

4. Upper Summation $\sum_{k=0}^{n}\binom{k}{m} = \binom{n+1}{m+1} \qquad 0 \leq m \leq n$

5. Convolution $\sum_{k=0}^{n}\binom{m}{k}\binom{p}{n-k} = \binom{m+p}{n} \quad n \leq m+p$

6. Addition $\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$

R Convolution is also called Vandermonde Convolution

<u>Basic Word Counting</u>

1 Counting the number of q −letter words in the alphabet of p distinct elements: $\{a_1, a_{2,}, \cdots, a_p\}$ such that OM and ~~RA~~.

Since ~~RA~~, once a letter is used, it may not be used again. Therefore,

in the 1st slot we have: p = p – 0 = p – (1 - 1) choices,

in the 2nd slot we have: p – 1 = p – (2 - 1) choices,

in the 3nd slot we have: p – 2 = p – (3 - 1) choices,

.

.

.

in the qth slot we have: p – (q - 1) = p – q + 1 choices.

By the product principle, the total number of choices is:

$$p\ (p-1)(p-2)\ \ldots\ (p-q+1)$$

$$= p\ (p-1)(p-2)\ \ldots\ (p-q+1)(1)$$

$$= p\ (p-1)(p-2)\ \ldots\ (p-q+1)(\frac{((p-q)!)}{((p-q)!)})$$

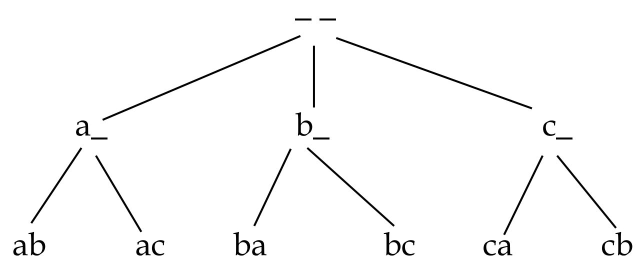
$$= \left(\frac{p\ (p-1)(p-2)\ldots(p-q+1)(p-q)!}{((p-q)!)}\right)$$

$$= \left(\frac{p\ !}{(p-q)!}\right)$$

$$= P(p\ \ q)$$

E We shall now generate, list, and count the set of 2-letter words in the alphabet $\{a, b, c\}$, denoted by $Wrd\left(2, \{a, b, c\}, OM, \cancel{RA}\right)$ such that OM and ~~RA~~. Note that here $p = 3$ and $q = 2$. We use a tree to systematically generate, list, and count all such words.

Tree



To generate the words systematically, we use the tree as shown above.

To list the words as we do below, we use the last row of the tree.

$$Wrd\left(2, \{a, b, c\}, OM, \cancel{RA}\right) = \begin{Bmatrix} ab, & ac, \\ ba, & bc, \\ ca, & cb \end{Bmatrix}$$

To find the cardinality of the set of words, we count the number of elements in the set.

$$\boldsymbol{\nu}\left(Wrd\left(2, \{a, b, c\}, OM, \cancel{RA}\right)\right) = 6$$

Note that we already know the number from the formula because, as we know from previous calculations $P(3,2) = 6$

2 Counting the number of q −letter words in the alphabet of p distinct elements: $\{a_1, a_{2,}, \cdots, a_p\}$ such that ~~OM~~ and ~~RA~~.

In this case, we have to use the result from (1), and identify all the words that are the same up to order.

Since each word has q letters, we may permute the letters of each word in q!, that is: $P(q,\ q)$ ways.

Hence the total number is obtained by dividing the answer for (1) by $P(q,\ q)$.

Therefore, the answer is:

$$\frac{P(p,\ q)}{P(q,\ q)}$$

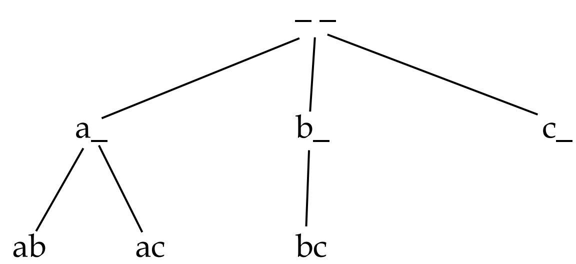
$$= C(p,\ q)$$

$$= \frac{p\,!}{((q!))\big(((p-q)!)\big)}$$

R Note that in this context, the word ab and ba are equivalent. Similarly, $abc, acb, bac, bca, cab, cba$ are all equivalent. In each case the words differ only in the order of the letters and, therefore, they are equivalent.

E We shall now generate, list, and count the set of 2-letter words in the alphabet $\{a, b, c\}$, denoted by $Wrd\left(2, \{a, b, c\}, \cancel{OM}, \cancel{RA}\right)$ such that ~~OM~~ and ~~RA~~. Note that here $p = 3$ and $q = 2$. We use a tree to systematically generate, list, and count all such words.

Tree



To generate the words systematically, we use the tree.

To list the words as we do below, we use the last row of the tree.

$$Wrd\left(2, \{a, b, c\}, \cancel{OM}, \cancel{RA}\right) = \left\{\begin{matrix} ab, ac, \\ bc \end{matrix}\right\}$$

To find the cardinality of the set, we count the number of elements in the set above.

$$\boldsymbol{\nu}\left(Wrd\left(2, \{a, b, c\}, \cancel{OM}, \cancel{RA}\right)\right) = 3$$

Note that we already know the number from the formula because, as we know from previous calculations $C(3,2) = 3$

3 Counting the number of q −letter words in the alphabet of p distinct elements: $\{a_1, a_{2,}, \cdots, a_p\}$ such that OM and RA.

Since RA, once a letter is used, it may be used again. Therefore,

in the 1st slot we have: p choices,

in the 2nd slot we have: p choices,

in the 3nd slot we have: p choices,

.

.

.

in the qth slot we have: p choices,

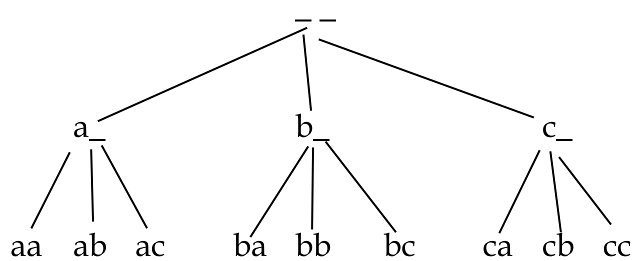
By the product principle, the total number of choices is:

$$(p)\,(p)(p)\ldots(p)$$

$$= p^q$$

E We shall now generate, list, and count the set of 2-letter words in the alphabet $\{a, b, c\}$, denoted by $Wrd\left(2, \{a, b, c\}, OM, RA\right)$ such that OM and RA.

Tree



To generate the words systematically, we use the tree.

To list the words as we do below, we use the last row of the tree.

$$Wrd\left(2, \{a, b, c\}, OM, RA\right) = \begin{Bmatrix} aa, & ab, & ac, \\ ba, & bb, & bc, \\ ca, & cb, & cc \end{Bmatrix}$$

To find the cardinality of the set, we count the number of elements in the set above.

$$\boldsymbol{\nu}\left(Wrd\left(2, \{a, b, c\}, OM, RA\right)\right) = 9$$

Note that we already know the number from the formula because, as we know from previous calculations $3^2 = 9$.

4 Counting the number of q −letter words in the alphabet of p distinct elements: $\{a_1, a_{2,}, \cdots, a_p\}$ such that ~~OM~~ and RA.

<u>The idea of codes</u>

In this section we introduce the idea of codes. Sometimes counting can be easier if one codes the elements of the set whose elements need to be counted in such a way that the set of codes is easier to count. The codes are constructed in such a way that there is a bijective correspondence between the original set and the set of codes. We make up a situation that mimics the procedure that we follow.

We think of the elements of the alphabet $\{a, b, c, d, e\}$ as ingredients for a salad: apples, bananas, cantaloupe pieces, dates, and elderberries. Each salad consists of exactly three helpings; one may take one, two, to three helpings of each ingredient. We may code a salad as a 3-letter word in the alphabet $\{a, b, c, d, e\}$ in which order does not matter and repitition is allowed. For example, the word aac corresponds to two helpings of apple and one helping of elderberries.

We shall now code each salad as a 7-letter word in the alphabet $\{0, 1\}$. To understand the coding we think of walling past a table where apples, bananas, cantaloupe pieces, dates, and elderberries are kept in containers in that order and from which one has to take helpings. For the salad aac, we start at apples and take two helpings, which is coded as 11 (one one not eleven), then we move to the container for bananas, then we move to the container for cantaloupe pieces. Each act of moving to the next container is recorded as a 0. So now we have 1100 (one, one, zero, zero). Then we take one helping of cantaloupe pieces, giving the code 11001 (one, one, zero, zero, one). Then we move to the container for dates, getting 110010, and again we move to the container for elderberries getting 1100100 and finally leave the table. The final code for aac is therefore 1100100.

Note that since one may take exactly three helpings, each code contains exactly three 1's. Since there are five ingredients, a, b, c, d, e there are four acts of moving from one container to the next (a to b, b to c, c to d, d to e), there are exactly four 0's in each code. The set of distinct salads is in bijective correspondence with 7-letter words in the alphabet $\{0, 1\}$ that contain exactly three 1's and four 0's. For example, the code 0101010 corresponds to move over from a to b, one helping of b, move over from b to c, one helping of c, move over from c to d, one helping of d, move over from d to e and then leave. The salad is bcd.

Alphabet := $\{a, b, c, d, e\}$

Want all 3-letter words such that **order does not matter** and **repetition is allowed**.

We shall code each word in exactly one way using 0's and 1's as follows:

Consider the word: aac

a:	Use a:	1
a:	Use a:	1
	Change letter to b:	0
b:		
	Change letter to c:	0
c:	Use c:	1
	Change letter to d:	0
d:		
	Change letter to e:	0
e:		

Word: aac Code: 1100100

Consider the word: bcd

a:		
	Change letter to b:	0
b:	Use b:	1
	Change letter to c:	0
c:	Use c:	1
	Change letter to d:	0

d: Use d: 1

Change letter to e: 0

e:

Word: bcd Code: 0101010

Note that the 0's mark changing a letter, and the 1 is used as many times as a letter is used. The number of changes of letter is 4 = 5 -1, as there are 5 distinct letters. If there are p distinct letters there will be p-1 changes of letter, and therefore p -1 0's.

Consider the word: eee

a:

Change letter to b: 0

b:

Change letter to c: 0

c:

Change letter to d: 0

d:

Change letter to e: 0

e: Use e: 1

e: Use e: 1

e: Use e: 1

Word: eee Code: 0000111

Consider the word: cdd

a:

Change letter to b: 0

b:

Change letter to c: 0

c:	Use c:	1
	Change letter to d:	0
d:	Use d:	1
d:	Use d:	1
	Change letter to e:	0

Word: cdd Code: 0010110

Consider the code: 1000101

Word: ade

Consider the code: 0100110

Word: bdd

Note that there are as many 1's as there are letters in the word. Since we are considering 3-letter words, there are 3 1's in each code. In the general case, since we shall be considering q-letter words, there will be q 1's.

Thus each code is of length 7

= (5 - 1) + 3

= (Number of changes of letter possible) + (number of slots)

= $(p - 1) + q$, in the general case

= $p + q - 1$

Thus, each word has exactly one code, and each code represents exactly one word.

The total number of codes is found by choosing exactly which q of the p + q – 1 slots should have 1's in them, that is: C (p + q – 1, q).

Equivalently, the total number of codes is found by choosing exactly which p-1 of the p + q – 1 slots should have 0's in them, that is: $C\ (p + q - 1, p - 1)$.

These two numbers are equal: $C\ (p + q - 1, q) = C\ (p + q - 1, p - 1)$

So, the number of q-letter words in the alphabet of p distinct elements: $\{a_1, a_2, \ldots, a_p\}$ such that ~~OM~~ and RA, is precisely:

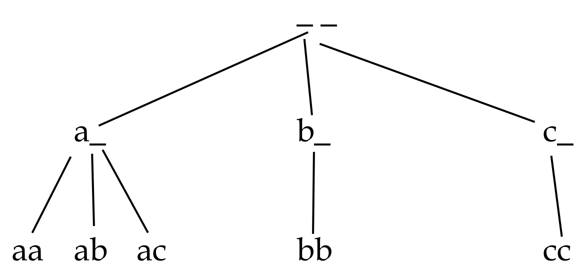
$$= C\,(p + q - 1, p - 1)$$

5 Summary

Number of q −letter words on anlphabet: $\{a_1, a_2, \ldots, a_p\}$ with p distinct letters	OM	~~OM~~
~~RA~~	$P(p,q)$	$C(p,q)$
RA	p^q	$C\,(p + q - 1, q) = C\,(p + q - 1, p - 1)$

E We shall now generate, list, and count the set of 2-letter words in the alphabet $\{a, b, c\}$, denoted by $Wrd\left(2, \{a, b, c\}, \cancel{OM}, RA\right)$ such that ~~OM~~ and RA.

Tree



To generate the words systematically, we use the tree.

To list the words as we do below, we use the last row of the tree.

$$Wrd\left(2, \{a, b, c\}, OM, RA\right) = \left\{\begin{matrix} aa, & ab, & ac, \\ & bb, & \\ , & & cc \end{matrix}\right\}$$

To find the cardinality of the set, we count the number of elements in the set above.

$$\boldsymbol{\nu}\left(Wrd\left(2,\{a,b,c\},OM,RA\right)\right)=6$$

Note that we already know the number from the formula because, as we know from previous calculations $C\ (3\ +\ 2 - 1,2) = C(4,2) = \frac{4!}{(2!)((4-2)!)} = \frac{4!}{(2!)(2!)} = \frac{\cancel{4}(3)2(1)}{\cancel{(2)(2)}} = 6$

Counting

D Given a universe of sets $\mathcal{U}$ and a finite set S we define a cardinality function as follows:

$$\nu := S \ \longmapsto \ \nu(S) : \mathcal{U} \longrightarrow \mathbb{N},$$

R $\nu(S)$ is the number of elements in S and provides one possible formalization, among many, of the notion of 'size of S'.

Comparing Cardinalities

Given two sets S and T

T $$\frac{S \subseteq T}{\nu(S) \leq \nu(T)}$$

T $$\frac{InjFnc(S,T) \neq \{\}}{\nu(S) \leq \nu(T)}$$

T $$\frac{SrjFnc(S,T) \neq \{\}}{\nu(S) \geq \nu(T)}$$

T $$\frac{BijFnc(S,T) \neq \{\}}{\nu(S) = \nu(T)}$$

R The previous four theorems are based on certain simple observations. If S is included in T, then T must have at least as many elements as S. If the set of injective functions from S to T is not empty, we have at least one way of mapping S to T preserving distinctness of elements; hence, T must have at least as many elements as S. If the set of surjectve functions from S to T is not empty, then there is at least one way of mapping S onto T so that every element of T is assigned to some input; hence, S must have at least as many elements as T. If there is a bijection from S to T, then distinctness of inputs is preserved and every target element is realised as an output; hence, S and T must have the same number of elements. **The reader must complete these heuristics and produce proofs.**

R The following four theorems are concerned with determining the cardinalities of sets constructed using set-theoretic operations from set of known cardinalities.

T $$\nu\big(\mathcal{P}(S)\big) = 2^{(\nu(S))}$$

T $$\nu(S \times T) = \nu(S) \times \nu(T)$$

T $$\nu(S \sqcup T) = \nu(S) + \nu(T)$$

T $$\nu(S \cup T) = \nu(S) + \nu(T) - \nu(S \cap T)$$

R The previous four theorems are not very difficult to establish. In the first three cases, try to devise a code for the elements of the set and count the codes. For the power set, one may use a modification of the idea of the salad which was introduced as a way of thinking in counting q-letter words on an alphabet with p distinct letters such that order does not matter and repetition is allowed. Think of the elements of S as names of salad ingredients. A salad is defined as the subset of ingredients selected; the amount does not matter. We recognise the empty set as a degenerate sort of salad. As one pass the ingredients, one either accepts and ingredient (1) or rejects it (0). This process produces a code for a salad. You must complete this argument. For the cartesian product the last row of the tree already produces the necessary codes. For the fourth one, one notes that adding the cardinalities counts the overlap twice; hence, one must remove one instance of the overlap to get the correct answer. For the third one, one notes that a coproduct basically disjointifies the two sets by placing the members on 'different floors'; this is what is achieved by taking the cartesian product with {0} and {1}. Then one needs to use the fact that if two sets are disjoint, then the cardinality of the intersection is 0. **The reader must complete these heuristics and produce proofs.**

R The fourth theorem can be vastly generalised and produces a counting principle known as the inclusion-exclusion principle. The idea is to find the cardinality of a set by a certain systematic alternation of overcounting and undercounting that eventually arrives at the correct value. **It will be helpful to draw Venn diagrams for the first few 'small' cases**.

T
$$\nu(S \cup T \cup U) = \begin{array}{cccccc} & \nu(S) & + & \nu(T) & + & \nu(U) \\ - & \nu(S \cap T) & - & \nu(S \cap U) & - & \nu(T \cap U) \\ + & S \cap T \cap U & & & & \end{array}$$

R The formula generalizing the above formulas to the case of an indefinitely finite family of finite sets is given below; it is more intricate and is said to embody the inclusion-exclusion principle. **The reader should first try to find a formula for the union of four sets and for the union of five sets**. The general formula is proved by induction.

Inclusion-exclusion principle

D Given a family of finite sets $\left(A_p \in \mathcal{P}(\mathcal{U}) \middle| p \in 1..n\right)$ we define

$$\forall k \in 1..n \;\; s_k \coloneqq \sum_{1 \le i_1 < i_2 \cdots < i_k \le n} \left(\nu\left(\left(A_{i_1}\right) \cap \left(A_{i_2}\right) \cap \cdots \cap \left(A_{i_k}\right)\right)\right)$$

T $$\nu\left(\bigcup_{p=1}^{n}\left(A_p\right)\right) = \sum_{p=1}^{n}(-1)^{p-1}\left(s_p\right)$$

R The above may be written in 'longhand' as:

$$\nu(A_1 \cup A_2 \cup \cdots \cup A_n) = s_1 - s_2 + s_3 - \cdots + (-1)^{n-1}(s_n)$$

R De Morgan's law allows us to write the above in a different way.

T
$$\begin{aligned} \nu\left(\bigcap_{p=1}^{n}\left(\left(A_p\right)^c\right)\right) &= \nu\left(\left(\bigcup_{p=1}^{n}\left(A_p\right)\right)^c\right) \\ &= \nu(\mathcal{U}) - \nu\left(\bigcup_{p=1}^{n}\left(A_p\right)\right) \\ &= \nu(\mathcal{U}) - \sum_{p=1}^{n}(-1)^{p-1}\left(s_p\right) \end{aligned}$$

R The above may be written in 'longhand' as:

$$\nu((A_1)^c \cap (A_2)^c \cap \cdots \cap (A_n)^c) = \nu(\mathcal{U}) - \left(s_1 - s_2 + s_3 - \cdots + (-1)^{n-1}(s_n)\right)$$

Number of derangements

D A permutation π of an ordered set $(k|k \in 1..n)$ is called a derangement

$:\Leftrightarrow \quad \pi$ has no fixed points, that is $\forall k \in 1..n \quad \pi(k) \neq k$

D The set of derangements of $1..n$ is denoted by: $Drngmnt(1..n)$.

T $$\nu\big(Drngmnt(1..n)\big) = (n!)\left(\sum_{k=0}^{n}((-1)^n)\left(\frac{1}{n!}\right)\right)$$

Pigeonhole Principle

T Given $k, n \in \mathbb{N}\backslash\{0\}$

1 if n pigeonholes are occupied by $n+1$ pigeons then some pigeonhole is occupied by at least 2 pigeons, and

2 if n pigeonholes are occupied by $kn+1$ where pigeons then some pigeonhole is occupied by at least $k+1$ pigeons.

Counting Relations and Functions

Given two sets S and T with $\nu(S) = s \in \mathbb{N}$ and $\nu(T) = t \in \mathbb{N}$ we have the following:

T $\nu\big(Rln(S,T)\big) = 2^{st}$

T $\nu\big(RflxRln(S)\big) = 2^{(s^2-s)}$

T $\nu\big(SymRln(S)\big) = 2^{\left(\frac{s^2+s}{2}\right)}$

T $\nu\big(TrnsRln(S)\big) =$

T $\nu\big(EqvRln(S)\big) =$

T $\nu\big(Pst(S)\big) =$

R **The reader is invited to prove the formulas that are given and also to find the formulas that are not given**. The first one is very easy if one notes that every relation from S to T is just a subset of $S \times T$. One can approach the problems by drawing a grid with the elements of S along the horizontal axis and the elements of T along the vertical axis. Then every element of the cartesian product $S \times T$ may be located on a grid by its coordinates. A relation is a subset of these points and each type of relation determines some geometric feature of the set of points that qualify. **This is a very good exercise for the brain**.

T $\nu\big(Fnc(S,T)\big) = t^s$

T $$\frac{s \le t}{\nu\big(InjFnc(S,T)\big) = \big(C(t,s)\big)\big(P(s,s)\big) = \binom{t}{s}(s!)}$$

T $$\frac{s \ge t}{\nu\big(SrjFnc(S,T)\big) = t^s - \sum_{k=1}^{n-1}\binom{n}{k}(n-k)^s}$$

T $$\frac{s = t}{\nu\big(BijFnc(S,T)\big) = P(s,s) = (s!)}$$

R **The reader is invited to prove these formulas**. The first, the second, and the fourth are fairly easy. One should prove them in the order first, third, second. The third requires the use of the inclusion-exclusion principle.

<u>Realising arithmetical expressions as cardinalities of sets</u>

R Suppose we have an arithmetical expression which takes natural numbers as inputs and yields natural numbers as outputs. The first problem is to realise the expression as the cardinality of a set. The second problem is to realise the expression as an answer to a question about some fragment of day-to-day experience.

E We consider the expression ab. There are two finite sets A and B such that $\nu(A) = a$, and $\nu(B) = b$. First, we note that :

$$
\begin{aligned}
& ab \\
= \quad & \big(\nu(A)\big)\big(\nu(B)\big) \\
= \quad & \nu(A \times B)
\end{aligned}
$$

The expression ab, is, therefore, realised as the cardinality of the set $A \times B$.

Our next task is to realise the ab as the answer to some question as a fragment of day-today existence.

There are infinitely many possibilities. The following is just an example.

We define :

A $:=$ The set of Septa trains running between Philadelphia and Trenton on day D.

B $:=$ The set of NJT trains running between Trenton and NY on day D.

Situation

X wishes to travel from Philadelphia to NY via Trenton on day D and needs to take exactly one of the above trains in each leg of the journey.

Question

How many distinct ways does X have of making the journey ?

R It is clear from previous discussion that there are exactly ab distinct jouneys.

E We consider the expression $(a+b)(c-d)$. There are four finite sets A, B, C and D such that $\nu(A)=a$, $\nu(B)=b$, $\nu(C)=c$, and $\nu(D)=d$. First, we note that :

$$
\begin{aligned}
& (a+b)(c-d) \\
= \quad & \Big(\nu(A)+\nu(B)\Big)\Big(\nu(C)-\nu(D)\Big) \\
= \quad & \Big(\nu(A \dot{\cup} B)\Big)\Big(\nu(C \backslash D)\Big) \qquad \qquad \Big(A \cap B = \{\ \ \},\ \ D \subseteq C\Big) \\
= \quad & \nu\left(\Big(A \dot{\cup} B\Big) \times \Big(C \backslash D\Big)\right)
\end{aligned}
$$

The expression $(a+b)(c-d)$, is, therefore, realised as the cardinality of the set $\Big(A \dot{\cup} B\Big) \times \Big(C \backslash D\Big)$.

Our next task is to realise the $(a+b)(c-d)$ as the answer to some question as a fragmen of day-today existence.

There are infinitely many possibilities. The following is just an example.

We define :

A := The set of Septa trains running between Philadelphia and Trenton on day D.

B := The set of Megabuses running between Philadelphia and Trenton on day D.

Note that $A \cap B$=$\{\ \ \}$.

C := The set of NJT trains running between Trenton and NY on day D.

D := The set of NJT trains without airconditioning running between Trenton and NY on day D.

Note that $D \subseteq C$

<u>Situation</u>

X wishes to travel from Philadelphia to NY via Trenton on day D and needs to take exactly one of the above trains or buses from Philadelphia to Trenton on the first leg of the journey and an airconditioned train from Trenton to NY on the second leg of the journey.

<u>Question</u>

How many distinct ways does X have of making the journey ?

<u>Homework</u>

0 Compute the following :

00 $P(3{,}2), C(3,2), P(P(3{,}2),2), C(P(3{,}2),2), P\big(P(3{,}2),P(3{,}2)\big), C\big(P(3{,}2),P(3{,}2)\big)$

$P\big(P(3{,}2),C(3{,}2)\big), C\big(P(3{,}2),C(3{,}2)\big), P\big(C(3{,}2),C(3{,}2)\big), C\big(C(3{,}2),C(3{,}2)\big)$

01 $P\big(P(n,0),C(n,0)\big), C\big(P(n,0),C(n,0)\big), P\big(C(n,1),C(n,0)\big), C\big(C(n,n),C(n,0)\big)$

03 $\binom{3}{2}, \binom{3}{\binom{3}{2}}, \binom{\binom{3}{2}}{2}, \binom{\binom{3}{2}}{\binom{3}{2}}, \binom{\binom{3}{2}}{\binom{3}{1}}, \binom{n}{n-1}, \binom{n}{\binom{n}{0}}, \binom{\binom{n}{0}}{n}, \binom{\binom{n}{1}}{\binom{n}{0}}, \binom{\binom{n}{k}}{\binom{n}{n-k}}$

04 $\binom{3}{2 \quad 1}, \binom{3}{2 \quad 0 \quad 1}, \binom{3}{1 \quad 1 \quad 1}, \binom{3}{0 \quad 1 \quad 0 \quad 2}, \binom{10}{1 \quad 2 \quad 3 \quad 4}, \binom{2n}{n \quad n}, \binom{2n}{n+1 \quad n-1}$

1 Prove each of the combinatorial identities.

2 **D** $(a,b,c,d) = (p,q,r,s) :\Leftrightarrow (a = p) \wedge (b = q) \wedge (c = r) \wedge (d = s)$

D $(a,b,c,d), (p,q,r,s)$ are said to be **distinct** $:\Leftrightarrow (a,b,c,d) \neq (p,q,r,s)$

Generate, list, and count:

the number of **distinct** quadruples (a,b,c,d) such that:

$a,b,c,d \in 1..9$

$10 \nmid \ (a+b+c+d)$ and

20 Order matters and repetition is not allowed

21 Order does not matter and repetition is not allowed

22 Order matters and repetition is allowed

23 Order does not matter and repetition is allowed

24 a, b, c, and d are all even

25 a, b, c, and d are all odd

26 exactly one of a, b, c, and d is even

27 exactly two of a, b, c, and d are even

28 exactly three of a, b, c, and d are even

29 a, b, c, and d are in increasing order

200 a, b, c, and d are in strictly increasing order

201 a, b, c, and d are in palindromic order

202 a, b, c, and d are all prime

203 exactly two of a, b, c, and d are equal