Chapter 8

Binary Operations

So far we have been throwing around the term *operation*, whether seemingly mathematical in nature or not, frivolously and somewhat carelessly. Jumping into a car, paying for groceries, flipping the levers of a light switch or rigidly rotating a rectangle were all treated as, vague, operations.

We should now become much more precise with this terminology and get familiar with the official definition of *a binary operation on a set*.

Definition 10: *a binary operation on a set* S is a rule according to which to each ordered pair of elements of S, whether distinct or not, there corresponds a uniquely determined element from the same set S.

Observe that the above *rule* is often symbolized, with f or g, for example, and is given formulaically but it doesn't have to be. In grown-up mathematics such a rule can be described purely verbally or depicted with a drawing. As long as there is no ambiguity in the said verbal or pictorial description of a rule then everything is fine.

As an example, a rule that is defined on the pixels of the *Mona Lisa* painting by Leonardo da Vinci (1452-1519) can be given purely verbally as follows: given any two pixels of the Mona Lisa's painting by Leonardo da Vinci, return the pixel that is equidistant from the two given pixels and is colinear with them.

The mathematically savvy readers will surely recognize the operation of division of a line segment in the real or the complex plane in half.

Advanced and Optional

An unordered pair of the following two entities:

- a set G and
- a single binary operation $\circ$ defined on G

is known as *a groupoid*.

We observe in passing that in the grown-up fashion the above definition of a groupoid does not impose any additional requirements on the *properties*, such as an associativity or a commutativity, of the said binary operation.

Separately, in the older texts a slightly less pointy term of *an algebraic operation* was used for what now we refer to as *a binary operation on a set*.

Definition 10 tells us that a binary operation *on a set*, with the emphasis on the binding *on a set*, is not a nebulous and freewheeling rule that acts on two arbitrary elements, an element a of a set A and an element b of a set B, and correlates a particular element c of yet another set C to each ordered pair of such elements, but rather that:

- such a rule acts on two elements, x and y, from *the same set* and
- that *to each ordered pair* of such elements, such as (x, y), for example, that rule assigns a particular element also from *the same set*

Graphically, all the said ordered pairs of the elements of a finite set S can be represented or generated by the means of a square table that captures the essence of an algebraic structure known as a *Cartesian Product Of Two Sets*, A and B, say.

In general, if A and B are two finite sets then visually all the elements of the order-insensitive Cartesian products of these sets, symbolized as $A \times B$, where the individual factors of that product are taken in that order, can be represented as follows.

List all the elements a of the set A in the leftmost column of the $|A|$ by $|B|$ table, a table that has $|A|$ rows and $|B|$ columns, in the one-element-per-row fashion.

Assuming, for demonstration purposes, that the set A has the following three elements:

$$A = \{x, y, z\}$$

we list all the elements of the sample set A in three rows (Figure 8.1):

A	
x	
y	
z	

Next, list all the elements b of the set B in the topmost row of the $|A|$ by $|B|$ table in the one-element-per-column fashion.

Assuming, for demonstration purposes, that the set B has the following five elements:

$$B = \{1, 2, 3, 4, 5\}$$

we list all the elements of the sample set B in five columns (Figure 8.2):

$A \backslash B$	1	2	3	4	5
x					
y					
z					

By doing so we construct the column of three row headers, containing the elements x, y, z of the set A and we construct the row of five column headers, containing the elements $1, 2, 3, 4, 5$ of the set B.

Lastly, at the intersection of every m-th row and k-th column of the above table, keeping the column of row headers and the row of column headers separate, enter an ordered pair (a_m, b_k) of elements of the sets A and B by putting the element a_m of the set A from the m-th row header first and by putting the element b_k of the set B from the k-th column header last (Figure 8.3):

$A \backslash B$	1	2	3	4	5
x	$(x, 1)$	$(x, 2)$	$(x, 3)$	$(x, 4)$	$(x, 5)$
y	$(y, 1)$	$(y, 2)$	$(y, 3)$	$(y, 4)$	$(y, 5)$
z	$(z, 1)$	$(z, 2)$	$(z, 3)$	$(z, 4)$	$(z, 5)$

and voila, we have a visual representation of each and every ordered pair (a, b) of the Cartesian product $A \times B$ of the sets A and B taken in that particular order.

Moreover, the visual representation of the ordered pairs (b, a) of the Cartesian product $B \times A$ of the sets B and A taken in the reverse, with respect to the above experiment, order will be constructed in a similar way (Figure 8.4):

$B \backslash A$	x	y	z
1	$(1, x)$	$(1, y)$	$(1, z)$
2	$(2, x)$	$(2, y)$	$(2, z)$
3	$(3, x)$	$(3, y)$	$(3, z)$
4	$(4, x)$	$(4, y)$	$(4, z)$
5	$(5, x)$	$(5, y)$	$(5, z)$

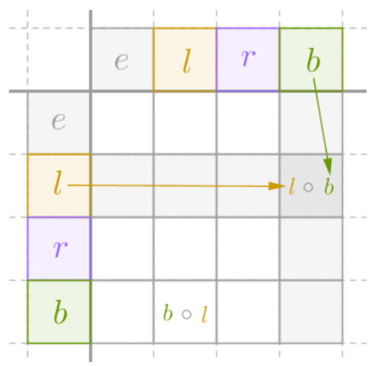
Observe that while the overall number of the ordered pairs across the above two tables is the same, $3 \cdot 5 = 5 \cdot 3 = 15$, the actual ordered pairs (a, b) and (b, a) across these tables are different.

For example, while the ordered pairs $(y, 3)$ and $(3, y)$ are constructed from the same individual components, the element y of the set A and the element 3 of the set B, these two pairs are nonetheless different.

In the case of a binary operation on a set, however, the individual components of each ordered pair of elements that the said operation glues together come not from different sets but from one and the same set, S, say.

Consequently, the main visualization vehicle of these ordered pairs will be the Cartesian product, $S \times S$, in which the individual factors are not different sets but one and the same set, S.

Which, in turn, means that the visual representation of that Cartesian product SS will be necessarily a square $|S|$ by $|S|$ table, a table with $|S|$ rows and $|S|$ columns and, without knowing it back then, in **Chapter 6** we already constructed essentially such a table with a minor exception that in such tables (Figure 6.7):



we did not use the corresponding pairs in the (x, y) format but, instead, constructed such pairs by gluing the elements of a group into a group-theoretic product, $x \circ y$, right away (Figure 6.8):

	e	l	r	b
e	ee	el	er	eb
l	le	ll	lr	lb
r	re	rl	rr	rb
b	be	bl	br	bb

which is an issue that is mostly cosmetic and symbolically it is possible to render a binary operation $\circ$ on a set S as follows:

$$\circ : S \times S \rightarrow S$$

which is easy to decipher. Namely, the above string of symbols says that all possible ordered pairs (x, y) comprised of the elements x, y of the set S constitute the domain of the operation $\circ$ and that the elements s of the same set S constitute the codomain of that operation.

Recall that in Figure 6.9:

	e	l	r	b
e	e	l	r	b
l	l	e	b	r
r	r	b	e	l
b	b	r	l	e

we replaced each such pair $x \circ y$ with exactly the said, sole, element s of S and by doing so we obtained a visual representation of this specific map $G \times G \rightarrow G$, which was the heart and soul of the light switch group.

Now, it so happens that a binary operation defined on a set is one of the central and defining characters of a group and in grown-up mathematics, theoretical physics, computer science and likely elsewhere, there often arises a need to answer the following, popular, question:

does the suspect pair of a candidate set this and that and a candidate operation such and such form a group?

One, but not the only, way to answer this type of questions was already experienced by us firsthand, loosely and informally, in the early exercises, which our readers, no doubt, diligently did themselves - such an approach boils down to *a filtering* or *an elimination process* in which we ask a series of questions and unless *all* these questions, without a single exception, are answered in the positive, the usual suspects are *not* a group:

- is the proposed operation *binary*? Yes or no? If the answer to that question is a *no* then we can short-circuit the filtering process and announce that the candidate set and the candidate operation are not a group. If the answer to that question is a *yes* then we move on to the next question, as more work needs to be done

- is the, now, binary candidate operation *associative*? Yes or no? If the answer to that question is a *no* then we can short-circuit the filtering process and announce that the candidate set and the candidate operation are not a group. If the answer to that question is a *yes* then we move on to the next question, as more work needs to be done
- does the candidate set possess *a unique identity element under the, now, binary associative candidate operation*? Yes or no? If the answer to that question is a *no* then we can short-circuit the filtering process and announce that the candidate set and the candidate operation are not a group. If the answer to that question is a *yes* then we move on to the next question, as more work needs to be done
- does every element of the candidate set possess *a unique inverse under the, now, binary associative candidate operation*? Yes or no? If the answer to that question is a *no* then we can short-circuit the filtering process and announce that the candidate set and the candidate operation are not a group. If the answer to that question is a *yes* then we move on to the next question, as more work needs to be done

That is, if the answer to *any* of the above questions is a *no* then the suspect duo is *not* a group.

If the answer to *all* of the above questions is a *yes*, then the suspect duo *is* a group.

More technically, the above definition of a binary operation on a set tells us that if the candidate operation $\circ$ defined on the candidate set S is binary then for any two elements a, b of S there (always) exists a uniquely determined element c of S such that $a \circ b = c$.

Later on, when we will be studying the so-called *subgroups* of a group, we will introduce the concept of *closure* of a subset under a binary operation.

The above filtering process is straightforward, mechanical, not very elegant but requires very little special preparation to master and it already allows students to answer a large number of interesting questions.

There exist other, more sophisticated methods of determining whether a structure under investigation is a group or not but for now, in the upcoming exercises, we will stick with the above filtering method and in addition to understanding the tactical meaning of a binary operation on a set, we will also get a formal and rigorous introduction to the said filtering process that is guaranteed to decide one way or another whether the suspect duo is or is not a group.

Exercise 8.1: is the operation of addition defined on the set of integers $\mathbb{Z}$ binary?

Solution: in order to answer this type of questions we need to trawl the unordered pair of the given entities through **Definition 10**. If the given operation and the given set do satisfy that definition then the suggested operation *is* binary, otherwise - it is *not*.

In the case of integer addition we know that the sum $k + l$ of any two integers k, l is a particular integer.

In other words, the sum of any two elements of the set $\mathbb{Z}$ is again a particular element of the same set.

Hence, by **Definition 10**, the operation of addition defined on the set of integers $\mathbb{Z}$ *is* binary. $\square$

Exercise 8.2: is the operation of addition defined on the set of *even* integers binary?

Solution: we know that the sum $2k + 2l$ of any two even integers $2k$ and $2l$ is a particular integer that is also even:

$$2k + 2l = 2(k + l)$$

In other words, the sum of any two elements of the set of even integers is again a particular element of the same set.

Hence, by **Definition 10**, the operation of addition defined on the set of even integers *is* binary. $\square$

Exercise 8.3: is the operation of addition defined on the set of *odd* integers binary?

Solution: we know that the sum $(2k + 1) + (2l + 1) = 2(k + l + 1)$ of any two odd integers $2k + 1$ and $2l + 1$ is, no doubt, an integer that is *particular* but it is an integer that is *even* and the integers that are even do not belong to the set of odd integers.

In other words, it is not the case that the sum of any two elements of the set of odd integers is again a particular element of the same set.

Hence, by **Definition 10**, the operation of addition defined on the set of odd integers is *not* binary. $\square$

The above **exercises 8.1**, **8.2** and **8.3** highlight an interesting phenomenon - every *even* integer is still *an* integer and, thus, belongs to the set of integers $\mathbb{Z}$. Likewise, every *odd* integer is still *an* integer and, thus, belongs to the set of integers $\mathbb{Z}$ also.

With respect to the set of all integers, the even integers are *a subset* of $\mathbb{Z}$.

Likewise, with respect to the set of all integers, the odd integers are also *a subset* of $\mathbb{Z}$ but a subset that is different from the subset of even integers.

In other words, a set A is *a subset* of a set S if *every* element of the set A is also an element of the set S.

The result of the **exercises 8.1** and **8.2** tell us that the operation of addition defined on the set of integers $\mathbb{Z}$ *is* binary and so is the operation of addition defined on the *subset* of *even* integers.

However, the result of the **exercise 8.3** tells us that the operation of addition defined on the subset of *odd* integers is *not* binary!

Thus, the results of the above three exercises tell us that in the context of binary operations we have to be extremely careful when switching from a given set to one of its subsets and trying to predict whether the given operation that is binary on the given set will still be binary if it is defined on one of the subsets of the said set - in general, it will not and we have to undertake the relevant analysis anew.

In other words, the property of an operation defined on a set S of being *binary*, in general, is not conserved and is not automatically propagated when we switch to a subset of S and, thus, cannot be taken for granted to survive the said transition.

In that light, if $\circ$ is a binary operation on a set S and A is such a subset of S that for any two elements a_1 and a_2 of A the element $a_1 \circ a_2$ is also in A then the subset A is (said to be) *closed under the binary operation* $\circ$.

Exercise 8.4: is the operation of *multiplication* defined on the set of odd integers binary?

Solution: about two thousand years ago Euclid, in his monumental work *"Elements"*, in Book 9 Proposition 29, proved that the product of any two odd positive integers is also odd. We now know that the same result holds for negative odd integers as well:

$$(2k+1)(2l+1) = 2(2kl+k+l)+1$$

In other words, it is the case that the product of any two elements of the set of odd integers is again a particular element of the same set.

Hence, by **Definition 10**, the operation of multiplication defined on the set of odd integers *is* binary. $\square$

Exercise 8.5: is the operation of multiplication defined on the set of positive rational numbers binary?

Solution: we know that the product $a \cdot b$ of any two positive rational numbers a, b is a particular positive rational number so long that we agree that equivalent positive rational numbers such as $1/3$ and $8/24$ represent the same positive rational number:

$$a \cdot b = \frac{m}{n} \cdot \frac{k}{l} = \frac{mk}{nl} \in \mathbb{Q}^+$$

In other words, the product of any two elements of the set of positive rational numbers is again a particular element of the same set.

Hence, by **Definition 10**, the operation of multiplication defined on the set of positive rational numbers *is* binary. □

Exercise 8.6: is the operation of multiplication defined on the set of negative even integers binary?

Solution: we know that the product $(-2k) \cdot (-2l)$ of any two negative even integers $(-2k)$ and $(-2l)$ is an integer that is:

- *particular* and
- that is also *even* but

such an integer is not negative:

$$(-2k) \cdot (-2l) = 2(2kl) > 0$$

as **Definition 10** demands.

In other words, it is not the case that the product of any two elements of the set of negative even integers is again a particular element of the same set.

Hence, by **Definition 10**, the operation of multiplication defined on the set of negative even integers is *not* binary. □

Exercise 8.7: is the operation of multiplication defined on the set of negative odd integers binary?

Solution: is left to the reader. Generate your own line of reasoning. Answer: no. □

Exercise 8.8: is the operation of multiplication defined on the set of negative integers binary?

Solution: is left to the reader.

Generate your own line of reasoning. Answer: no. □

Exercise 8.9: is the operation of division defined on the set of real numbers binary?

Solution: we know that the result of the division of a real number r by a real number s is well-defined so long that the real number s is distinct from zero, in which case such a result is a particular real number.

However, it is not the case that the result of division of two real numbers r and s is defined for *all* real numbers r and s.

From the **All, Every, Each, Exists, If P Then Q** chapter we remember that:

"all are" negates into "at least one is not"

and in the case of division and real numbers the number zero is *at least one real number that is not* allowed to be divided by.

Thus, it is not the case that for all the ordered pairs of elements of the set of real numbers the result of the division of one such number by another is a particular element of the same set.

Hence, the operation of division defined on the set of real numbers is *not* binary. □

Exercise 8.10: is the operation of subtraction defined on the set of real numbers binary?

Solution: we know that the result of the subtraction $(r - s)$ of any real number s from any real number r is a specific real number.

In other words, the difference of any two elements of the set of real numbers is again a particular element of the same set.

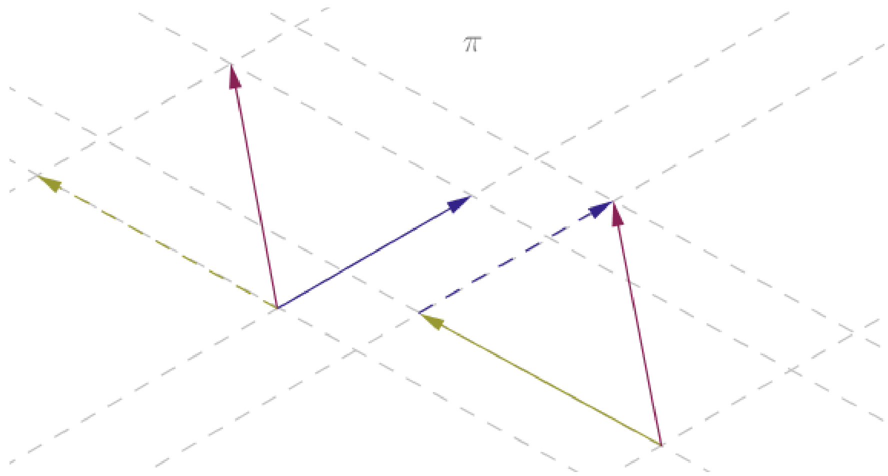
Hence, by **Definition 10**, the operation of subtraction defined on the set of real numbers *is* binary. □

Exercise 8.11: is the operation of addition defined on the set of planar or 2-space vectors binary?

Solution: we know that the sum of any two coplanar vectors is a vector that belongs to the same plane to which the two vectors that are summed belong.

For example, let the blue and the green vectors belong to the parent Euclidean plane, π.

Then, no matter how these two vectors are added together, by using the parallelogram method or by using the parallel translation method, the result of such an addition will be the red vector that always belongs to the same Euclidean plane, π (Figure 8.5):



Hence, the operation of addition defined on the set of planar or 2-space vectors *is* binary. □

Exercise 8.12: is the operation of addition defined on the set S containing only the real number zero:

$$S = \{0\}$$

binary?

Solution: we know that $0 + 0 = 0$ and, thus, to all the ordered pairs of elements of S, which consist of non-distinct elements this time, there corresponds a particular, and, in fact, the same, element of the set S.

Therefore, the operation of addition defined on the set S containing only the real number zero, *is* binary. □

Exercise 8.13: is the operation of multiplication defined on the set S containing only the real number one:

$$S = \{1\}$$

binary?

Is the operation of multiplication defined on the set $S = \{7\}$ containing only the real number seven binary?

Solution: is left to the reader.

Generate your own line of reasoning. Answers: *yes* and *no*. □

Exercise 8.14: once upon a time, in a kingdom far away, there lived three queens who, as a hobby, liked to preside over trials that involved exactly two suspects.

The first queen, named, Andiana, would furnish *a guilty* verdict only when both suspects were proven to be guilty and *a not guilty* verdict in all the other cases.

The second queen, named Origana, would furnish *a not guilty* verdict only when both suspects were proven to be not guilty and *a guilty* verdict in all the other cases.

The third queen, named Xorilla, would furnish *a guilty* verdict only when exactly one of the two suspects was proven to be guilty (which specific one - does not matter), while the other suspect was proven to be not guilty and she would furnish *a not guilty* verdict in all the other cases.

What a duplicitous bunch, ah?

There also was the fourth queen, named Notella, who was the weird one in the family because she liked to preside over trials with just one suspect and if that suspect was proven to be guilty then she would furnish *a not guilty* verdict and if a suspect was proven to be not guilty then she would furnish *a guilty* verdict!

Do these three queens belong to the Cathedral of Binary Operations or not?

In other words, are the computer science operations of AND, OR and XOR defined on the set of binary strings of a finite length $n = 1, 2, 3, \ldots$ binary or not?

Solution: a binary string of a finite length n is a contiguous collection of n same-size characters or *bits*, each of which can have only one of only two mutually exclusive values: zero or one.

Here is a sample binary string of a finite length $n = 5$:

$$10110$$

Consider a binary string of length $n = 1$.

Such a string can only carry one of two possible values: 0 or 1.

Let us investigate the result of the application of the above computer science operators to any two such strings.

It is straightforward to verify that the result of the application of the operators AND, OR and XOR to any two bit strings of length 1 regardless of the actual values that they carry at the moment of evaluation is exactly one bit that is:

- for the AND operator is 1 only if both single-bit strings carry the same value of 1 and is 0 otherwise
- for the OR operator is 0 only if both single-bit strings carry the same value of 0 and is 1 otherwise
- for the XOR operator is 1 only if the values of the single-bit strings disagree, one is 0 and the other is 1 or vice versa, and is 0 otherwise

The result of the application of the above computer science operators to any two binary strings s_1 and s_2 of length $n > 1$ will again be a binary string that is exactly the same $n > 1$ bits long and that is populated by the consecutive application of the respective rule described above to all $n > 1$ corresponding pairs of bits of the strings s_1 and s_2 bit by bit.

For example:

```
AND:    OR:     XOR:
10110   10110   10110
11001   11001   11001
-----   -----   -----
10000   11111   01111
```

and so on.

Thus, the computer science operations of AND, OR and XOR defined on the set of binary strings of a finite length $n = 1, 2, 3, \ldots$ *are* all binary. □

Since we did not yet study permutations officially, in the next exercise we shall limit ourselves to the work that we have done in the **Exercise 5.3.2** on the congruence motions of a rectangle.

Exercise 8.15: is the operation of composition defined on the set of two-row tables that contain the names of the vertices of a rectangle and that are generated by the congruence motions of a rigid rectangle binary?

Solution: in the **Exercise 5.3.1** we already constructed a descriptive and wordy list of all 16 compositions of the congruence motions of a rectangle and, after the **Difficult, Giant, Leap Forward**, we now know that officially we constructed a table of all 16 *group products* of the Klein four-group in its much more efficient, compact and concise *symbolic* form.

In other words, we have shown that a composition of any two congruence motions of a rectangle is again a congruence motion of that rectangle.

In the **Exercise 5.3.2** we learned that each congruence motion of a rectangle can be recorded as a certain two-row table and we also found/discovered a rule according to which the result of a composition of any two congruence motions of a rectangle can be recorded effortlessly by stacking one such two-row table on top of the other such two-row table with the consequent removal of all the intermediate rows:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{pmatrix} \circ \begin{pmatrix} 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix}$$

Such two-row tables will soon officially become the so-called *permutations* and we already see that the composition of two two-row tables or two permutations is again a two-row table or a permutation that has exactly the same domain as the two-row tables comprising the said composition.

We, thus, conclude that the operation of composition defined on the set of two-row tables that contain the names of the vertices of a rectangle and that are generated by the congruence motions of a rigid rectangle *is* binary. □

For the next few exercises, marked with one and two asterisks, we will need some familiarity with the set of *complex numbers*, $\mathbb{C}$, which can be defined as the set of all ordered pairs (a, b) where a and b are real numbers $\mathbb{R}$ and where the said binary operations of addition and multiplication obey the following rules:

$$(a, b) + (c, d) = (a + c, b + d) \tag{1}$$

and:

$$(a, b) \cdot (c, d) = (a \cdot c - b \cdot d, b \cdot c + a \cdot d) \tag{2}$$

and where the signs $+$, $-$ and $\cdot$ on the RHSs of (**1**) and (**2**) stand, respectively, for *addition*, *subtraction* and *multiplication* operations as they apply to *real* numbers.

From the above formal definition of complex numbers it can be deduced that the set of complex numbers is equipped with the so-called *imaginary unit* that is symbolized as $i = (0, 1)$ and that has this peculiar property that:

$$i \cdot i = i^2 = -1 = (-i)^2 = (-i) \cdot (-i)$$

where the above symbol $\cdot$ means the multiplication of these particular type of objects, complex numbers, and not a generic group product.

Even if some of our readers are not familiar with the set of complex numbers yet, they still can do this exercise perfectly fine by mechanically using the rule shown above for the multiplication of the imaginary unit i by itself - if the other factor of a product is not i then the result of such a product is deduced in a predictable way:

$$i \cdot 1 = i,\ i \cdot (-1) = -i,\ (-i) \cdot (-1) = i$$

and so on.

Exercise 8.16*: is the operation of multiplication defined on the following set of complex numbers:

$$S = \{1, i, -1, -i\}$$

binary?

Solution: in general, the number of linear permutations $P(n, r)$ of n distinct items taken r items at a time is known to us:

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

Since in our case $n = 4$ and $r = 2$, it seems that we would have to analyze:

$$P(4, 2) = \frac{4!}{(4-2)!} = 12$$

order-sensitive products of distinct factors plus 4 products where the factors repeat.

But we also know that the multiplication of complex numbers is an operation that is *commutative*. That is, for any complex numbers a and b it is the case that $a \cdot b = b \cdot a$.

Hence, for our purposes it is sufficient to analyze the products of the order-**in**sensitive pairs of elements of the given set S.

The number of ways $C(n, r)$ in which r distinct items can be picked from a set of n items in such an order-insensitive way can be obtained from the number $P(n, r)$ by reducing that number by the number of ways in which r distinct items can be permuted linearly:

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

As such, in our case we have to analyze only:

$$C(4, 2) = \frac{P(4, 2)}{2!} = \frac{12}{2} = 6$$

pairs of non repeating elements of S:

$$1 \cdot i,\ 1 \cdot (-1),\ 1 \cdot (-i)$$

$$i \cdot (-1),\ i \cdot (-i),\ (-1) \cdot (-i)$$

as follows:

$$1 \cdot i = i,\ 1 \cdot (-1) = -1,\ 1 \cdot (-i) = -i$$

$$i \cdot (-1) = -i,\ i \cdot (-i) = 1,\ (-1) \cdot (-i) = i$$

Likewise, we compute the four products of the repeating elements of the set S:

$1 \cdot 1 = 1,\ i \cdot i = -1, (-i) \cdot (-i) = -1,\ (-1) \cdot (-1) = 1$

observing that in *all* the cases the respective products are the elements of the original set S.

Hence, the operation of multiplication defined on the given set S *is* binary. □

Exercise 8.17:** is the operation of composition defined on the set of linear real-to-real functions binary?

Is the operation of composition defined on the set of linear complex-to-complex functions binary?

Solution: a linear real-to-real function is symbolized as follows:

$$y = ax + b,\ a \neq 0 \tag{3}$$

where otherwise a and b are arbitrary *constant* real numbers and x represents *a variable* real number also known as *the independent variable*.

A composition of two such linear functions is constructed by substituting another linear function symbolized as $cx + d$ into the independent variable x in (**3**):

$$a(cx + d) + b \tag{4}$$

Now that we have a composition of two linear real-to-real functions in (**4**), we want to see if such a composition can be massaged into a shape that aligns with the definition of a linear function shown in (**3**).

A straightforward manipulation of multiplying the real number a by the real numbers cx and d and the consequent re-grouping the resultant summands shows that *yes*, the object shown in (**4**) can be made to look like the object shown in (**3**):

$$a(cx + d) + b = (ac)x + (ad + b)$$

But we know that by the very definition of a linear function shown in (**3**) neither a nor c are equal to zero. Hence, the product ac is also distinct from zero: $ac \neq 0$.

Moreover, the product of any two real numbers, such as the product ad of the real numbers a and d, is a real number and the sum of any two real numbers, such as the sum $ad + b$ of the real numbers ad and b, is a real number. Hence:

$$(ax + b) \circ (cx + d) = (ac)x + (ad + b),\ ac \neq 0$$

which is to say that a linear substitution of an independent variable in a linear function does not take us out of the set of linear functions - a composition of any two linear real-to-real functions is a linear real-to-real function.

Therefore, the operation of composition defined on the set of linear real-to-real functions *is* binary.

Some of our readers may know that over the field of complex numbers it is possible to distinguish two slightly different but similar linear complex-to-complex functions.

The linear functions of *the first kind*:

$$f(z) = az + b, a \neq 0$$

and the linear functions of *the second kind*:

$$f(z) = a\overline{z} + b, a \neq 0$$

where a and b are arbitrary constant complex numbers and z is the independent complex variable.

A straightforward manipulation not unlike the manipulation that we carried out over the set of real numbers shows that *yes*, the linear substitution of the independent variable in a linear complex-to-complex function of *the first* kind:

$$(az + b) \circ (cz + d) = (ac)z + (ad + b),\ ac \neq 0$$

does not take us out of the respective set and, hence, the operation of composition defined on the set of linear complex-to-complex functions of *the first* kind *is* binary.

However, the operation of composition defined on the set of linear complex-to-complex functions of *the second* kind is *not* binary.

When we conjugate a linear function of the second kind that is being substituted into the independent argument of another linear function of the second kind, because the operation of conjugation distributes over multiplication and addition, the conjugate images of the constants c and d will still be constant complex numbers – they will different complex numbers, $\overline{c}$ and $\overline{d}$ respectively, but they still will be constant.

However, the independent variable $\overline{z}$ will lose its original shape:

$$\overline{\overline{z}} = z$$

in such a composition:

$$(a\overline{z} + b) \circ (c\overline{z} + d) = (a\overline{c})z + (a\overline{d} + b)$$

which *will* take us out of the respective set. As a matter of fact, we see that a composition of any two linear functions of the second kind becomes a linear function of *the first* kind.

Hence, the operation of composition defined on the set of linear complex-to-complex functions of the second kind is *not* binary. □

Exercise 8.18:** is the operation of composition defined on the set of linear-fractional real-to-real functions binary?

Is the operation of composition defined on the set of linear-fractional complex-to-complex functions binary?

Solution: a linear fractional real-to-real function is symbolized as follows:

$$f(x) = \frac{ax + b}{cx + d},\ ad - bc \neq 0 \tag{5}$$

where a, b, c and d are otherwise arbitrary constant real numbers and x is the independent real variable.

If $c = 0$ in (**5**) then the linear fractional function $f(x)$ degenerates into a linear function shown in (**3**) and we will not be interested in this case since we already studied it.

Thus, from now on we assume that in addition to the constraint in (**5**) it is also the case that $c \neq 0$.

A composition of two linear fractional functions, the function $f(x)$ shown in (**5**) and the function $g(x)$ such that:

$$g(x) = \frac{a_g x + b_g}{c_g x + d_g},\ a_g d_g - b_g c_g \neq 0, c_g \neq 0 \tag{6}$$

is constructed in exactly the same way in which a composition of two linear functions was constructed earlier:

$$f \circ g = \frac{a \cdot \frac{a_g x + b_g}{c_g x + d_g} + b}{c \cdot \frac{a_g x + b_g}{c_g x + d_g} + d}$$

Multiplying both the numerator and the denominator of the above fraction by the denominator $(c_g x + d_g)$ of the function $g(x)$:

$$f \circ g = \frac{a a_g x + a b_g + b c_g x + b d_g}{c a_g x + c b_g + d c_g x + d d_g}$$

and collecting the terms in the result that are x-bound and x-free, we have:

$$f \circ g = \frac{(a a_g + b c_g)x + (a b_g + b d_g)}{(c a_g + d c_g)x + (c b_g + d d_g)}$$

which is to say that with the following designations:

$$A = a a_g + b c_g,\ B = a b_g + b d_g$$

$$C = c a_g + d c_g,\ D = c b_g + d d_g$$

we have a function that *looks* almost like a linear fractional one:

$$f \circ g = \frac{Ax + B}{Cx + D}$$

In order to settle the matter one way or the other, all we have to do is check the value of the difference $AD - BC$.

Doing so, we see that the first and the last summands of the respective product AD will cancel out with the first and the last summands of the respective product BC:

$$AD - BC = a d a_g d_g + b c b_g c_g - a d b_g c_g - b c a_g d_g$$

and, thus, that the resultant sum can be factorized by grouping its like ad-summands and its like bc-summands like so:

$$AD - BC = ad(a_g d_g - b_g c_g) - bc(a_g d_g - b_g c_g)$$

which is to say that:

$$AD - BC = (ad - bc)(a_g d_g - b_g c_g)$$

But the above factors represent the combinations of the respective multipliers that constitute the constraints shown in (**5**) and (**6**).

Thus, it is the case that $AD - BC \neq 0$.

As such, the operation of composition defined on the set of linear fractional real-to-real functions *is* binary.

Our readers are encouraged to construct their own lines of reasoning that show that the operation of composition defined on the set of linear fractional complex-to-complex functions of *the first* kind:

$$w(z) = \frac{az + b}{cz + d},\ ad - bc \neq 0$$

is binary, while the operation of composition defined on the set of linear fractional complex-to-complex functions of *the second* kind:

$$w(z) = \frac{a\overline{z} + b}{c\overline{z} + d},\ ad - bc \neq 0$$

is *not* binary. □