Relations and Functions

At the end of this lecture you should be able to:

- explain the meaning of the term **relation**;
- represent a binary relation pictorially and as a set of ordered pairs;
- find the **inverse** of a given relation;
- determine whether a particular relation on a set is an equivalence relation;
- define the term **function**, and determine whether a particular binary relation is or is not a function;
- **apply** a function to a particular input;
- write an appropriate **function signature**;
- where appropriate, specify a function as a mathematical formula;
- describe how a function can effectively have multiple inputs;
- explain the term **function composition**.
- determine whether a function is **injective**, **surjective** or **bijective**.

Relations

A **relation** is a set of connections from one set to another set.

Consider a small college that specialises in science.

Four subjects are taught – physics, chemistry, biology and mathematics.

There are three members of the teaching staff – Juliet, Ade and Meena.

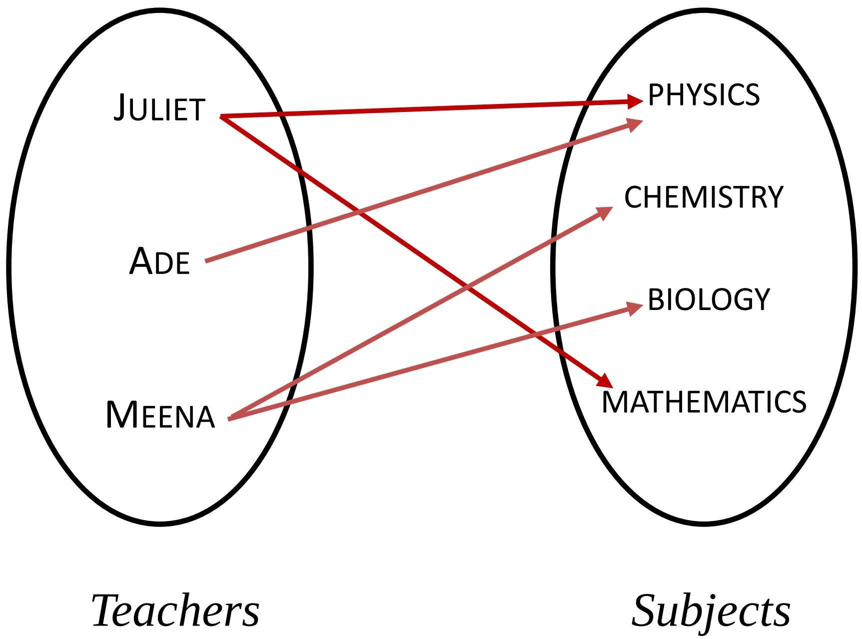
We can define two sets:

$$Teachers = \{ \text{JULIET}, \text{ADE}, \text{MEENA} \}$$

$$Subjects = \{ \text{PHYSICS}, \text{CHEMISTRY}, \text{BIOLOGY}, \text{MATHEMATICS} \}$$

We need to consider which teachers teach which subjects.

One way we can represent this is by means of a diagram:



The diagram shows a **relation** between the set of teachers and the set of subjects.

We often use R to represent a relation, and express R as a *set of ordered pairs*:

In our example:

R = { (JULIET, PHYSICS), (JULIET, MATHEMATICS), (ADE, PHYSICS), (MEENA, CHEMISTRY), (MEENA, BIOLOGY) }

We can express the relationship between the individual pairs as follows:

JULIET $\mathscr{R}$ MATHEMATICS

We read this as: JULIET *is related to* MATHEMATICS

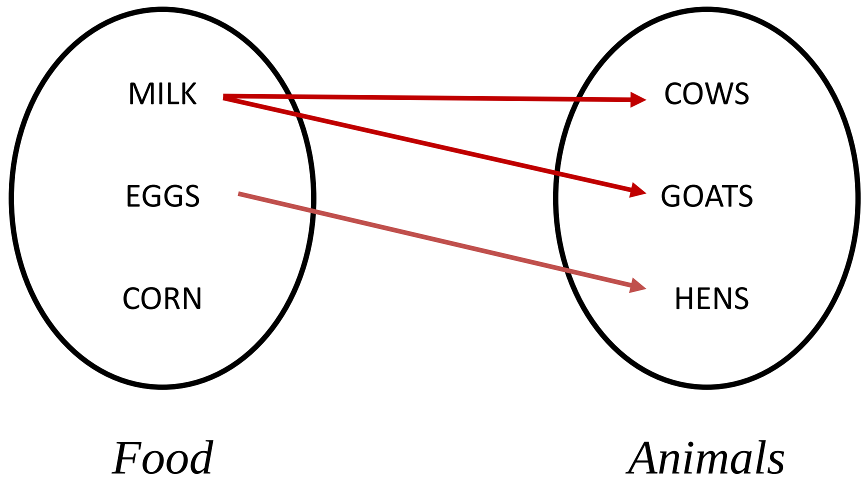
In our case it means: JULIET *teaches* MATHEMATICS

Conversely, the symbol $\not\mathscr{R}$ means *is not related to*.

JULIET $\not\mathscr{R}$ CHEMISTRY means JULIET *does not teach* CHEMISTRY

Worked examples

1. The diagram below represents the relation "*is produced by*" between the set *Food* and the set *Animals*:



a) Represent the relation in terms of a set of ordered pairs.

b) Write in words: MILK $\mathscr{R}$ GOATS

<u>Solution</u>

a) R = { (MILK, COWS), (MILK, GOATS), (EGGS, HENS) }

b) MILK *is produced by* GOATS

2. A and B are two sets and R is a relation from set A to set B .

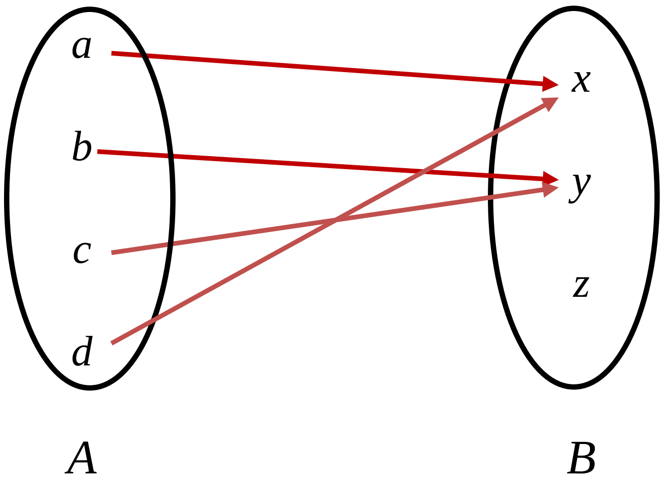
$$A = \{ a, b, c, d\}$$

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

$$R = \{(a, x), (b, y), (c, y), (d, x)\}$$

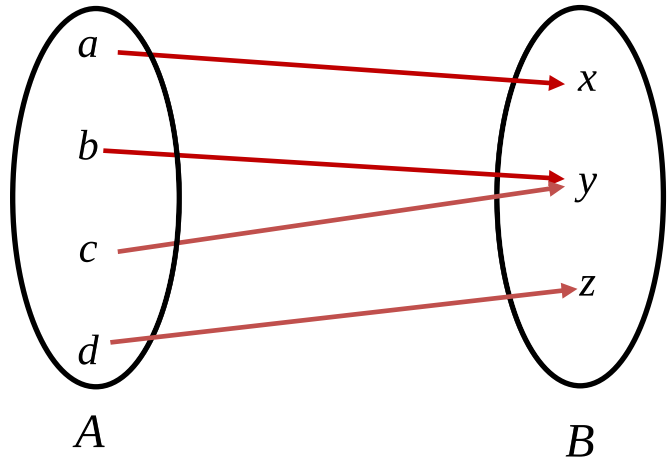
Represent the relation R pictorially.

<u>Solution</u>

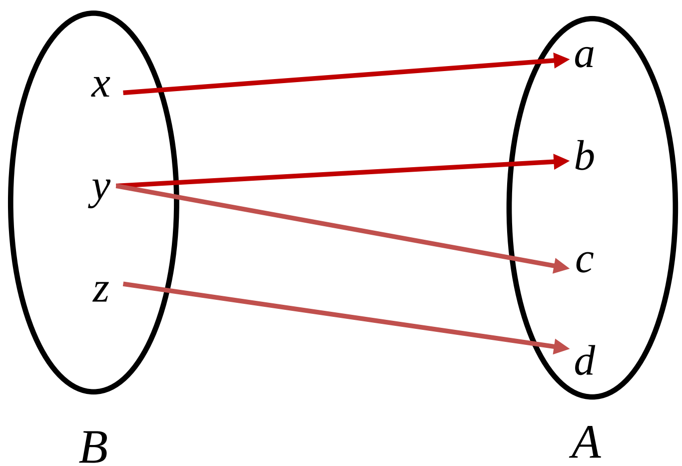


Inverse relations

Consider the following relation, R, from a set A to a set B:



The relation below represents the **inverse** of this relation, written as R^{-1},



Inverse relations (continued)

Expressing R as a set of ordered pairs, we have:

$$R = \{ (a, x), (b, y), (c, y), (d, z) \}$$

$$R^{-1} = \{ (x, a), (y, b), (y, c), (z, d) \}$$

In general, if R is a relation from a set A to set B, then the inverse, R^{-1} , is a relation from a set B to a set A and is found by reversing the pairs.

We can use the notation for set comprehension that we learnt last week to give a formal definition of the inverse:

$$R^{-1} = \{ (b, a) \mid (a, b) \in R \}$$

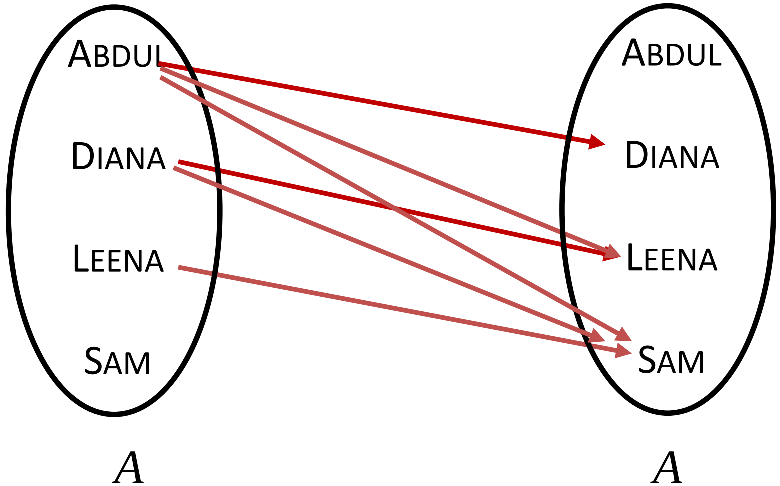
Relations on a set

When a relation is between two identical sets, we talk about a relation on a set.

For example consider the following set, A, consisting of four people.

A = {ABDUL, DIANA, LEENA, SAM}

Now consider the relation R, *is older than*. It could look like this:



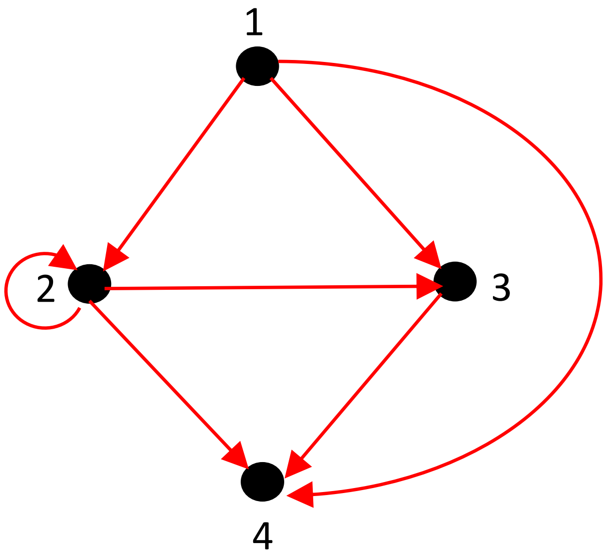
Writing the relation as a set of ordered pairs gives:

R = {(ABDUL, DIANA), (ABDUL, LEENA), (ABDUL, SAM), (DIANA, LEENA), (DIANA, SAM), (LEENA, SAM) }

Digraphs of relations

A **digraph** (a directed graph) is a useful way of representing a relation on a set.

The digraph below represents the relation { (1, 2), (2, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4) } on the set {1, 2, 3, 4}



Symmetric relations

Consider the set, A

$$A = \{ 1, 2, 3, 4 \}$$

Now consider the following relation, R, on this set:

$$R = \{ (1, 2), (2, 1), (2, 3), (3, 2)\ (3, 4), (4, 3) \}$$

We can see that: $1\mathscr{R}2$ and $2\mathscr{R}1$
$2\mathscr{R}3$ and $3\mathscr{R}2$
$3\mathscr{R}4$ and $4\mathscr{R}3$

This is an example of a **symmetric** relation. In such a relation, for every pair (a, b) there is also a pair (b, a).

Put another way, whenever $a\mathscr{R}b$ then we also find that $b\mathscr{R}a$.

We can see that our previous example *is older than* is <u>not</u> symmetric.

For example: ABDUL *is older than* DIANA but DIANA *is not older than* ABDUL

Reflexive relations

Consider again the set, A

$$A = \{ 1, 2, 3, 4 \}$$

Now consider the following relation, R, on this set:

$$R = \{ (1, 1), (2, 1), (2, 2), (2, 4) (3, 3), (4, 4) \}$$

We can see that:
$1\mathscr{R}1$
$2\mathscr{R}2$
$3\mathscr{R}3$
$4\mathscr{R}4$

This is an example of a **reflexive** relation. In such a relation, every element is related to itself.

Put another way, for every element, a, we find that $a\mathscr{R}a$.

We can see that our previous example *is older than* is <u>not</u> reflexive.

ABDUL *is not older than* ABDUL and so on.

Transitive relations

Once again, consider the set, A

$$A = \{ 1, 2, 3, 4 \}$$

Now consider the following relation, R, on this set:

$$R = \{ (1, 2), (1, 3), (1, 4), (2, 3)\ (2, 4), (3, 4) \}$$

We can see that: $1\mathscr{R}2$, $2\mathscr{R}3$ and $1\mathscr{R}3$
$1\mathscr{R}3$, $3\mathscr{R}4$ and $1\mathscr{R}4$
$2\mathscr{R}3$, $3\mathscr{R}4$ and $2\mathscr{R}4$

This is an example of a **transitive** relation. In such a relation, if $a\mathscr{R}b$ and $b\mathscr{R}c$, then $a\mathscr{R}c$.

We can see that our previous example *is older than* is transitive.

ABDUL *is older than* DIANA, DIANA *is older than* LEENA, and ABDUL *is older than* LEENA.
ABDUL *is older than* LEENA, LEENA *is older than* SAM, and ABDUL *is older than* SAM.
DIANA *is older than* LEENA, LEENA *is older than* SAM, and DIANA *is older than* SAM.

Equivalence relations

If a relation is symmetric, reflexive and transitive then it is described as an **equivalence** relation.

Worked example

Consider the relation *is greater than or equal to* ($\geq$) on the set of real numbers.

Is this an equivalence relation?

<u>Solution</u>

It is a reflexive relation, because for every number, a, $a \geq a$.

For example $1 \geq 1$, $3 \geq 3$ and so on.

It is a transitive relation, because whenever $a \geq b$ and $b \geq c$, then $a \geq c$.

For example $7 \geq 6$, $6 \geq 4$ and $7 \geq 4$.

However it is *not* a symmetric relation, because it is *not* true that whenever $a \geq b$ then $b \geq a$.

For example $7 \geq 6$ but $6 \geq 7$.

Is greater than or equal to is therefore *not* an equivalence relation.

Equivalence classes

Consider the set, A

$$A = \{ 1, 2, 3, 4 \}$$

Now consider the following equivalence relation, R, on this set:

$$R = \{ (1, 1), (2, 2), (3, 3), (4, 4), (1, 2), (2, 1), (2, 3), (3, 2), (1, 3), (3, 1)\}$$

For each element of the set there is an **equivalence class**, defined as the set of elements that that element is related to.

It is written with square brackets:

So: $[1] = \{1, 2, 3\}$ $[2] = \{1, 2, 3\}$ $[3] = \{1, 2, 3\}$ $[4] = \{4\}$

Notice here that we have arrived at two disjoint sets {1, 2, 3} and {4}.

Equivalence classes always partition the set. In other words, equivalence classes are either equal or disjoint.

Formally, given a set S and an equivalence relation R on S, the *equivalence class* of an element a in S is the set:

$$\{ x \in S \mid x \mathcal{R} a\}$$

Worked example

Consider the set, A

$$A = \{ a, b, \mathrm{c} \}$$

Now consider the following equivalence relation, R, on this set:

$$R = \{ (a, a), (a, b), (b, a), (b, b), (c, c) \}$$

Find the equivalence classes for this relation.

<u>Solution</u>

$$[a] = \{a, b\} \qquad [b] = \{a, b\} \qquad [c] = \{c\}$$

Binary relations

The relations we have just seen are **binary** relations: they are relations between *two* sets.

In our first worked example we had: Food = { MILK, EGGS, CORN}
$\mathit{Animals}$ = { COWS, GOATS, HENS}

and we looked at the particular relation:

R = { (MILK, COWS), (MILK, GOATS), (EGGS, HENS) }

In a binary relation, all the possible relations that could exist are given by the Cartesian product of the two sets.

In this case:

Food X $\mathit{Animals}$ = { (MILK, COWS), (MILK, GOATS), (MILK, HENS), (EGGS, COWS), (EGGS, GOATS), (EGGS, HENS), (CORN, COWS), (CORN, GOATS), (CORN, HENS) }

A particular relation, R, is a subset of the Cartesian product.

n-ary relations

We saw that in our previous example we had two sets:

$$Food = \{\text{MILK, EGGS, CORN}\}$$
$$Animals = \{\text{COWS, GOATS, HENS}\}$$

Let us add another set into the mix:

$$Farms = \{\text{MANOR FARM, CITY FARM}\}$$

Imagine that they keep goats and hens at Manor Farm, cows and hens at City Farm. A relation that shows which animal produces which food at which farm would look like this:

$$R = \{(\text{MILK, GOATS, MANOR FARM}), (\text{MILK, COWS, CITY FARM}),$$
$$(\text{EGGS, HENS, MANOR FARM}), (\text{EGGS, HENS, CITY FARM})\}$$

A relation like this, which relates three sets together, is called a **ternary** relation. In general we can have 2-ary, 3-ary, 4-ary relations and so on.

Application to computing

Relational databases

Databases are of enormous importance to computing – and nowadays most databases are based on the *relational* model.

Relational databases use tables to store information. Each row of the table is called a **record** and each column is called a **field**.

Employee_No	Name	Department_No	Date_of_Birth	Salary
108765	Patty O'Furniture	3	14.09.1960	43,000
282098	Isadore Open	1	30.08.1981	28,000
291073	Justin Case	2	28.02.1975	55,000
365498	Anne T Body	4	31.10.1990	22,000

Each row of the table is one element in a *relation* (in this case a 5-ary relation).

A typical database consists of many related tables: the table on the right is connected to the table above by the *Department_No* field.

Department_No	Department_Name
1	Accounts
2	Sales
3	Human Resources
4	Marketing

Relational databases are created and queried by a special language based on the theory of relations.

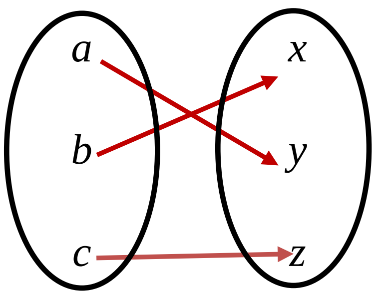
This is known as **Structured Query Language** (**SQL**).

Functions

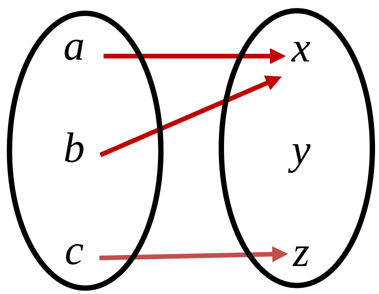
A function is a special sort of binary relation. It is a relation in which each element of the first set relates to one and only one element in the second set.

This means we can assign more than one element of the first set to the same element of the second set, but never the other way round.

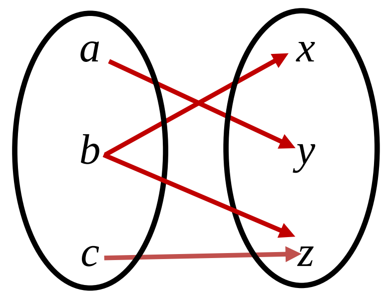
It also means that every element of the first set must be assigned to a member of the second set.



This is a function

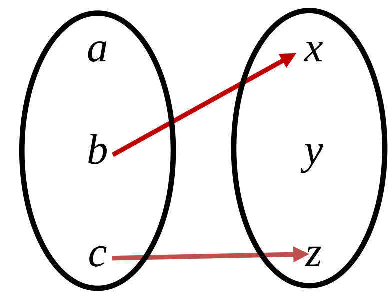


This is a function



This is not a function

(b is assigned to 2 elements)



This is not a function

(a is not assigned to any element)

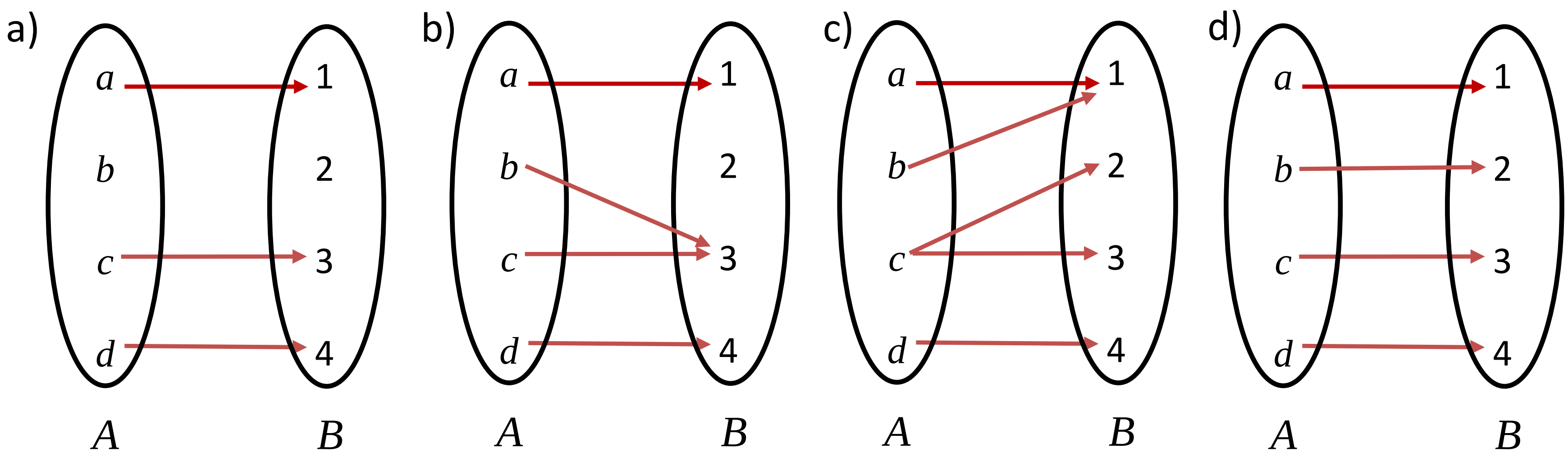
The first set is known as the **domain** of the function, the second set is known as the **range** (or **codomain**).

The element in the second set that corresponds to a particular element of the first set is called the **image** of that element. In the first diagram, for example, y is the image of a.

In a function, every element of the domain must have *one and only one* image in the codomain.

Worked example

Which of the diagrams below represents a function?



<u>Solution</u>

a) is not a function – b has no image in the codomain.

b) is a function.

c) is not a function – c maps to two elements in the codomain.

d) is a function.

Functions as mappings

We see that a function from set A to set B is a set of assignments from A to B.

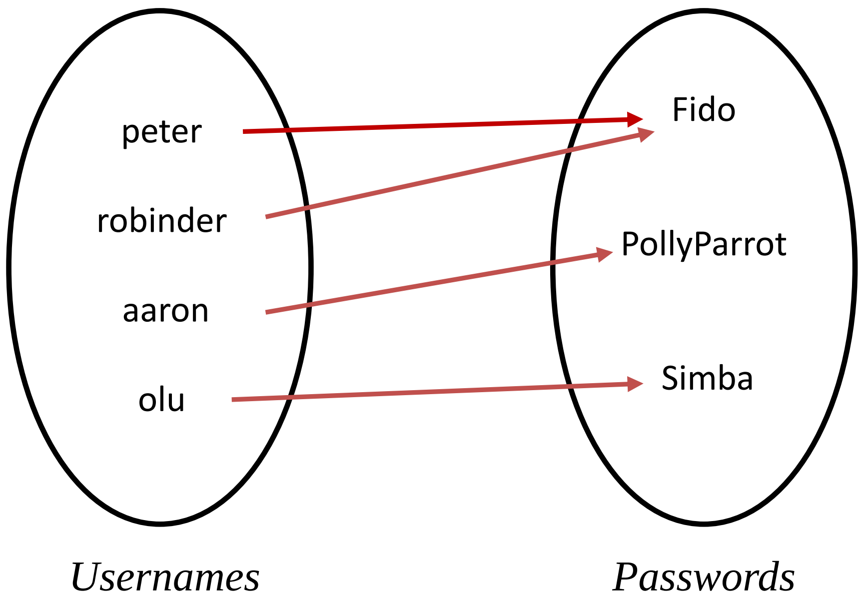
It can also be described as a *mapping* from A to B.

In fact it is a *many-to-one* mapping, because many elements of A can map to one element of B.

Practical example

On a computer network we need to keep records of users of the system and their corresponding passwords.

We would need a function that maps names of users to passwords. This is represented below:



Note

This is a function because more than one user can have the same password, but each user can have only one password.

Using the password function

Let's name our function f.

We can now **apply** our function to each element of $Usernames$ in order to find the user's password.

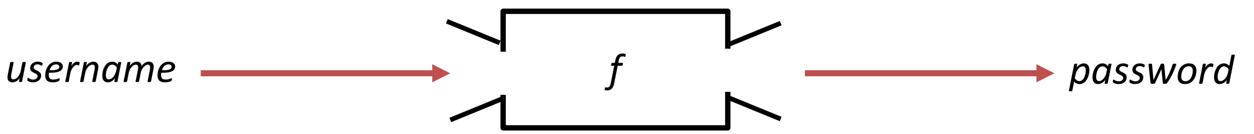
write this as follows:

$$f(peter) = Fido$$
$$f(robinder) = Fido$$
$$f(aaron) = PollyParrot$$
$$f(olu) = Simba$$

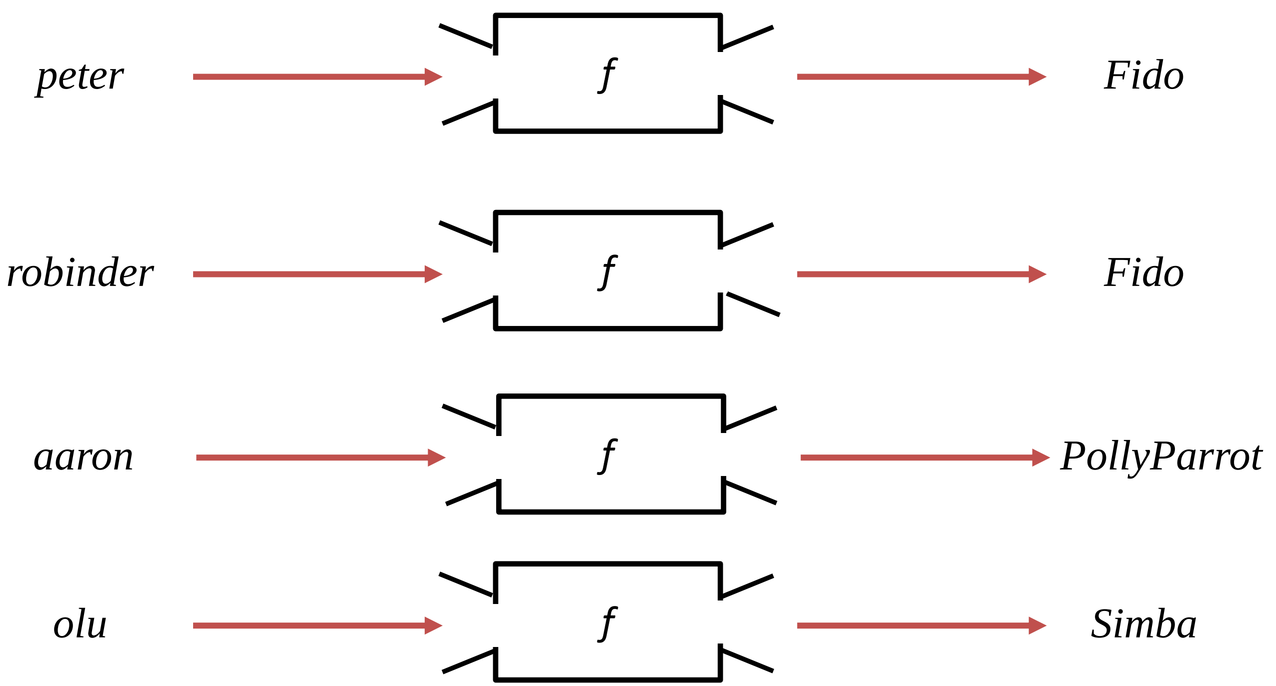
We pronounce this as *f* of *peter*, *f* of *robinder* and so on.

A function as an input/output device

A function is effectively a device that transforms an input to an output. Our password example, for instance, took a username as an input and transformed it into a password.



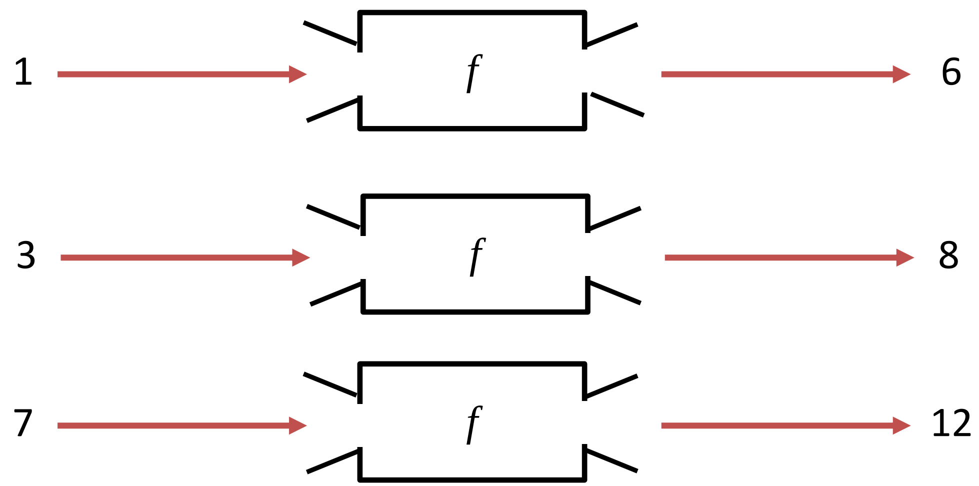
For each individual input, we get the following outputs:



Functions as formulae

When a function takes a number and outputs another number, it can often be specified as a mathematical formula.

Imagine a function, f, that behaves like this for three different inputs:



In this case, the function takes an input and adds 5 to it. If the function behaves like this for every input, we can express our function as a formula:

$$f(x) = x + 5$$

Applying our function

Now that we have our function expressed like this: $f(x) = x + 5$, we can apply it to any input.

For example:

$$f(2) = 7$$
$$f(0) = 5$$
$$f(-2) = 3$$

Another example

Imagine a function that always outputs the square of the input.

We can write this as:

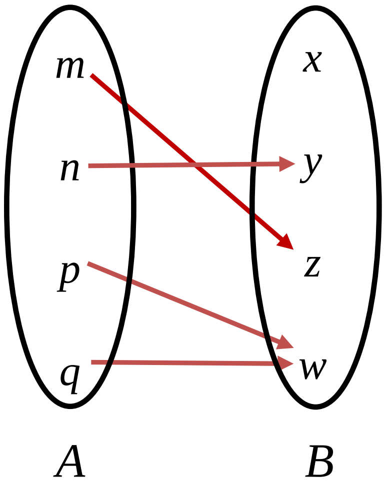
$$f(x) = x^2$$

Some sample inputs would give us:

$$f(2) = 4$$
$$f(3) = 9$$
$$f(-2) = 4$$

Worked examples

1. A function f, which maps from a set A to a set B, is represented pictorially below:



What is the value of the following?

a) $f(m)$ b) $f(n)$ c) $f(p)$ d) $f(q)$

<u>Solution</u>

a) $f(m) = z$

b) $f(n) = y$

c) $f(p) = w$

d) $f(q) = w$

2. A function f is specified as follows:

$$f(x) = 2x - 1$$

What is the value of the following?

a) $f(3)$ b) $f(-2)$ c) $f(0)$

Solution

a) $f(3) = 2 \times 3 - 1$
$= 6 - 1$
$= 5$

b) $f(-2) = 2 \times (-2) - 1$
$= -4 - 1$
$= -5$

c) $f(0) = 2 \times 0 - 1$
$= 0 - 1$
$= -1$

3. A function f is specified as follows:

$$f(x) = 3x^2 - 10$$

What is the value of the following?

a) $f(5)$ b) $f(-4)$ c) $f(0)$

Solution

a) $f(5) = 3 \times 5^2 - 10$
$= 3 \times 25 - 10$
$= 75 - 10$
$= 65$

b) $f(-4) = 3 \times (-4)^2 - 10$
$= 3 \times 16 - 10$
$= 48 - 10$
$= 38$

c) $f(0) = 3 \times 0 - 10$
$= 0 - 10$
$= -10$

The signature of a function

We usually give a function a name. In the password example we called our function f.

When we specify a function, we first need to provide its **signature**:

$$f : Usernames \rightarrow Passwords$$

This means that the function f maps from the set of $Usernames$ to the set of $Passwords$.

This is sometimes read as f maps $Usernames$ into $Passwords$.

Specifying a function

When we specify a function, we need to do two things: provide its signature, and say what it does.

In general, if f is a function from set A to set B we write its signature as:

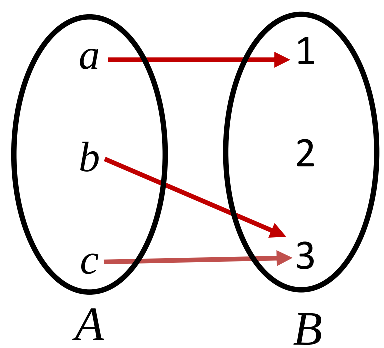
$$f : A \to B$$

If an element x in the domain maps onto an element y in the range, we write:

$$f(x) = y$$

As we saw earlier, we pronounce $f(x)$ as f of x.

An example of the above function could be:



Applying the function to each element of A gives us:

$$f(a) = 1$$
$$f(b) = 3$$
$$f(c) = 3$$

This, together with the signature, fully specifies the function.

Specifying formulaic functions

Consider one of the previous examples, $f(x) = x^2$

Consider the following inputs:

$$f(3) = 9$$
$$f(4) = 16$$
$$f(-2) = 4$$
$$f(-3) = 9$$

The input is an integer, but the output is always a natural number (we never produce a negative number by taking the square).

We could write our function like this:

$$f : \mathbb{Z} \to \mathbb{N}$$

$$f(x) = x^2$$

So when we specify functions which are formulae, the signature will involve number sets such as the above.

From now on, when working with formulae and equations, we will assume that unless otherwise stated we are dealing with real numbers - in other words the function maps from $\mathbb{R}$ to $\mathbb{R}$.

Not all mappings are functions

Consider the following mapping:

$$f : \mathbb{N} \rightarrow \mathbb{R}$$

$$f(x) = \sqrt{x}$$

A number in the domain such as 4 will have two images in the codomain, namely 2 and -2.

So f is not a function.

Functions with more than one input

In computer graphics, images are formed from pixels made up of three primary colours - red, green and blue.

Mixing these can produce three secondary colours - cyan, magenta, yellow:

CYAN = BLUE + GREEN

MAGENTA = RED + BLUE

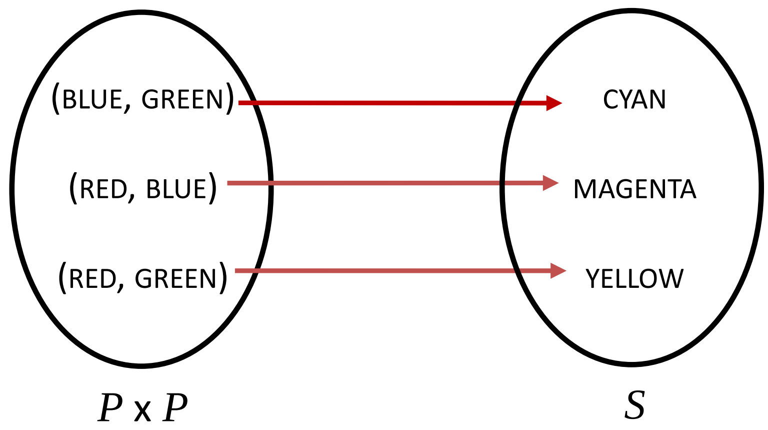
YELLOW = RED + GREEN

We can define two sets, P to represent primary colours and S to represent secondary colours:

P = {RED, GREEN, BLUE}

S = {CYAN, MAGENTA, YELLOW}

A function that produces secondary colours from primary colours would look like this:



The first set is actually the Cartesian product of two sets. In this case the two sets are the same, P and P – but they could be two different sets.

If we call the above function f, then we see that the signature of f is:

$$f : P \times P \rightarrow S$$

What we have done here is effectively to extend the notion of a function to be able to accept more than one input.

When applying the function we should write, for example: f((RED, BLUE)) = MAGENTA

However, we normally drop the double brackets and write: f(RED, BLUE) = MAGENTA

In general: $f(x,y) = z$

Worked example

Consider the following function:

$$f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$$

$$f(x, y) = x^2 + y - 1$$

State the value of:

a) $f(1, 2)$ b) $f(0, 0)$

Solution

a) $f(1, 2) = 1^2 + 2 - 1 = 2$

b) $f(0, 0) = 0^2 + 0 - 1 = -1$

Function composition

Function composition means applying one function to the result of another function.

Consider two functions: $g(x)$ and $f(x)$

$f(g(x))$ is the composition of f and g.

It means apply g, then apply f to the result.

This is also written as: $(f \circ g)(x)$

Worked Example

$g(x) = 2x+3$ $f(x) = x^2$

Calculate: $f(g(2))$

<u>Solution</u>

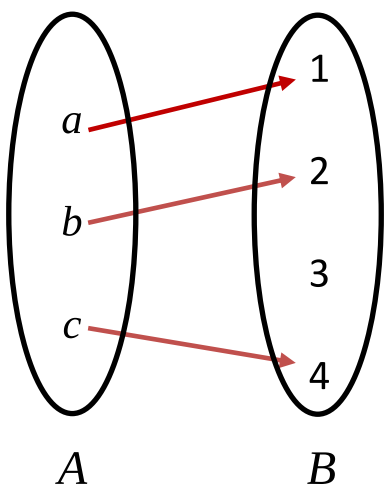
$g(2) = 2 \text{ x } 2 + 3 = 7$

$f(7) = 7^2 = 49$

One-to-one, onto and bijective functions

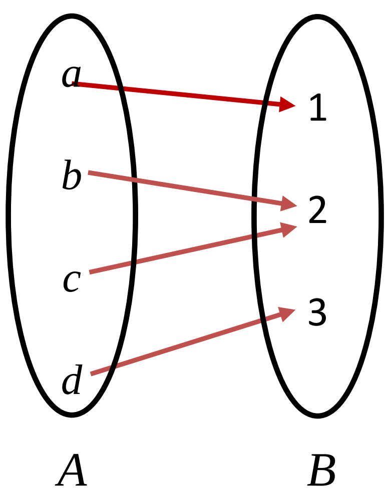
<u>One-to-one</u>

A function from A to B is **one-to-one** (or **injective**) if different elements in the domain all have distinct images.



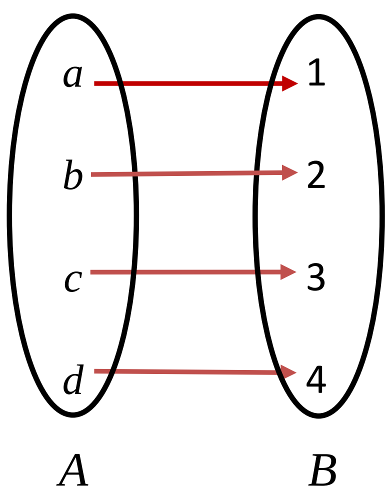
<u>Onto</u>

A function from A to B is **onto** (or **surjective**) if each element of B is an image of some element of A.



<u>Bijective</u>

A function which is both one-to-one and onto is described as **bijective**.



Application to computing

1. Computer programming

Every programming language provides a means of grouping lines of code together to perform a particular task.

Such routines are given different names in different languages – in Java for example they are called *methods*.

A method can accept multiple inputs and optionally can output a value. Any method that outputs a value is equivalent to a function in mathematics.

Consider a mathematical function (which we will call *myFunction*) that is specified as follows:

$$myFunction: \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$$

$$myFunction(x, y) = 2x + y$$

In Java we would code a corresponding method as follows:

```
int myFunction(int firstIn, int secondIn)
{
    return 2 * firstIn + secondIn;
}
```

2. Business Software

In everyday life we often need to do such things as adding several numbers together or finding an average.

This is particularly true in the world of business – and therefore business software such as databases and spreadsheets comes packed with built-in functions to perform tasks such as these.

A commercial application such as Microsoft Excel™ contains literally hundreds of pre-defined functions that accept a number of inputs (often as cells in the spreadsheet) and output the result.

In Excel™, such functions include *sum*, *average*, *max*, *min* and many, many more.

To find the average of the numbers in the cell range A1:A4, for example, we would enter:

=AVERAGE(A1:A4)