2 Functions

Our first experience with functions probably tells us that a function is a rule which, given a number x, defines a number $f(x)$. We're used to drawing graphs which represent functions and to doing various kinds of manipulation (differentiation, integration). But actually a function is something a little more general than that, and the view above is a bit inappropriate for a computer scientist.

Definition 2.1 *Given two sets X and Y, a* <u>*function*</u> *f from X to Y is a rule which, given any element $x \in X$ defines an element $f(x)$ in Y.*

3 things are necessary to define a particular function; the set X (called the <u>domain</u> of the function), the set Y, (called the <u>codomain</u> of the function), and the rule itself. We call $f(x)$ the <u>image</u> of x under f.

Examples of how we might describe a function.

$$f : \mathbb{Z} \to \mathbb{Z}, \quad f(x) = x^2$$

or alternatively

$$f : \mathbb{Z} \to \mathbb{Z}, \quad x \mapsto x^2$$

In programming, the domain corresponds to the type of the input argument of the function, and the codomain to the type of the returned value. We usually use letters like f, g, h as names of functions, or sometimes $f_1, f_2, \ldots$. But it doesn't really matter.

The example probably looks quite familiar, because the function deals with numbers, but there's no reason why either X or Y should be a set of numbers. e.g.

- We might use f to measure temperature, say the body temperature of the people in this room, in which case Y would be a set of numbers, but X wouldn't.
- Alternatively, we might let X be the set of people in this room, and Y a set of female first names, and let $f(x)$ be the first name of the mother of person x.
- Or X might be a set of records, and Y the values of one particular component. Then $f(x)$ might access that component.
- Any relation R which relates sets $A_1, \ldots A_n$ can be thought of as a function from $A_1 \times \ldots A_n$ to the set $\{True, False\}$. We define $R(x_1, \ldots x_n)$ to take the value $True$ when the relation holds on $(x_1, \ldots x_n)$ and $False$ otherwise. This explains why we're using the same notation for functions as we already (sometimes) used for relations. A function with codomain $\{True, False\}$ is called a Boolean valued function.

Any of the above is a perfectly valid function.

Some people use the word <u>mapping</u> instead of function, or the word <u>map</u>.

2.2 *Using diagrams to help us to understand a function.*

Since a function f is specified by the two sets X, Y and all the pairs $(x, f(x))$, it is actually a binary relation on $X \cup Y$. Each element of X is 'related' by f to its image $f(x)$ in Y (but careful, the boolean valued function we associate with that binary relation is not f). So it is natural to use a diagram to represent a function f, consisting of points and directed arcs. One cluster of points represents the elements of X, another cluster of points the elements of Y, and for each element x of X, a directed arc joins it to its image $f(x)$ in Y. (NB. If X and Y have some points in common we see each common points twice, once as an element of X, and once as an element of Y. This is a slight difference between the diagram of a function and that of the associated relation on $X \cup Y$.)

Examples.

We mentioned at the end of the section on sets that a diagram like this is called a directed graph.

Of course we are also used to representing functions by a different kind of graph (it is confusing that the same name is used for two quite different things). The points of X are marked off on an x-axis, the points of Y on a y-axis, defining a coordinate sustem. Then each point $(x, f(x))$ is plotted. We'll call this kind of graph the cartesian graph of the function.

Examples.

Of course we can only use a cartesian graph for representing a function which map a set of numbers to a set of numbers. A directed graph can be used a bit more generally, but in practice we're unlikely to use it if X and Y are infinite.

Examples 2.3 *Some more examples of functions and non-functions.*

Let X be the set of student numbers of people in this room. Let Y be the set of all female first names.

Now consider the following rules.

If $f(x)$ is the first name of the person with student number x, then f doesn't define a function from X to Y, since some people in this room are male.

If $g(x)$ is the name of the girlfriend/boyfriend of the person with student number x, then we may or may not have a function. We will if everyone in this room has exactly one girlfriend and no boyfriend, but we won't if someone in the room has a boyfriend, and we won't if someone has two different partners with different names. For g to be a function we need exactly one value of $g(x)$ for each element x of X, and we need that value to be an element of Y.

Look at this pictorially. (We can make up a set of just 5 students, to save time).

In the directed graph which represents a function, we see one directed arc out of each element of X, leading to an element of Y. Some elements of Y may be at the end of more than one arc, and some elements of Y may be at the end of no arc.

A few more examples of directed graphs.

Definition 2.4 *Where $f : X \to Y$ is a function, the set of all elements of Y*

which are of the form $f(x)$ for some x in X is called the range *of f. In set theory, we would describe this set as*

$$\{y \in Y : y = f(x) \quad \text{for} \quad \text{some} \quad x \in X\}$$

Some people use the word image instead of range.

The identity function on a set X is an important example of a function.

Definition 2.5 *Identity function $i_X : X \to X$.*

Sometimes we have more than one function and it makes sense to combine them.

Definition 2.6 *Composite of two functions*

We can't always compose two functions. We can only define the composite $g \circ f$ of functions f and g if the codomain of f is in the domain of g.

Examples

Minimum function We define the function $min : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ by the rule

$$\begin{aligned} min(x, y) &= x \quad \text{if} \quad x < y \\ &= y \quad \text{otherwise.} \end{aligned}$$

Clearly $min(x, y)$ computes the smaller (minimum) of x and y.

By applying the function more than once we can compute the minimum of more than 2 numbers. For instance, the minimum of 3 numbers x, y, zif found as $min(min(x, y), z)$ and the minimum of 4 numbers x, y, z, t as $min(min(x, y), min(z, t))$.

We're not exactly using the composite of min with itself here, because of course the range of min is not in its domain, but it's the same basic idea.

Sometimes we want to undo the effect of a function. We know $f(x)$ but we want to know where it came from, what x was. That could be very important if we wanted to detect an error in a program. Maybe we have a program which runs through a large set of input values x, and for each value of x we have an output value y, computed using some function y? If the program is interrupted we may want to recover x. Maybe memory constraints meant we couldn't store it. Can we recover x?

We can if f has an inverse.

Definition 2.7 *Given a function $f : X \to Y$, then a function $g : Y \to X$ is an* inverse *for f if $g \circ f(x) = x$ for all $x \in X$ and also $f \circ g(y) = y$ for all $y \in Y$.*

Another way of phrasing the above is to say that $g \circ f$ is the identity function on X and $f \circ g$ is the identity function on Y.

We say above that g is an inverse for f. In fact if f has an inverse, it only has one. But it doesn't have to have one at all.

Examples of some functions with inverses, some without. Some numerical, some functions between finite numerical sets.

Actually to recover x from $f(x)$ we only need one of the two properties which define the inverse, the first of the two.

Definition 2.8 *Given a function* $f : X \to Y$*, then a function* $g_1 : Y \to X$ *is an* *<u>left inverse</u>* *for* f *if* $g_2 \circ f(x) = x$ *for all* $x \in X$ *and a function* $g_2 : Y \to X$ *is a* *<u>right inverse</u>* *if* $f \circ g_2(y) = y$ *for all* $y \in Y$*.*

If some function g is both a left inverse and a right inverse for f then it is an inverse for f.

The function $f_1 : \mathbb{Z} \to \mathbb{Z}$ defined by $f_1(x) = 2x$ has a left inverse but no right inverse, while the function $f_2 : \mathbb{Z} \to \mathbb{Z}$ defined by $f_2(x) = [x/2]$ has a right inverse but no left inverse. In fact f_2 is the left inverse of f_1, and f_1 is the right inverse of f_2.

When does a function f have an inverse, or a left inverse, or a right inverse?

Look at some examples.

Definition 2.9 *injective (1-1), surjective (onto) , bijective (and the nouns)*

Relate to the pictures, both directed graphs and coordinate graphs.

We see that f has a left inverse if (and only if) it is injective, and a right inverse if and only if it is surjective. So f has an inverse if and only if it is bijective. So the words bijective and invertible are basically interchangeable.

Some more examples. Actually compute the rules for (left) inverses of a few easy functions, such as

$$f : \mathbb{R} \to \mathbb{R}, \quad x \mapsto (2x + 3)$$

by setting $x = 2y + 3$ and rearranging to get y on its own.

Definition 2.10 *Given two sets* X *and* Y*, a* *<u>partial function</u>* f *from* X *to* Y *is a rule which defines an element* $f(x) \in Y$ *for some elements* x *of* X*, but may be undefined for others.*

We write $f : X \not\to Y$ *to indicate that* f *is a partial function (but not a function) from* X *to* Y*.*

We call the subset of X *of those elements* x *for which* $f(x)$ *is defined the* *<u>domain</u>* *of* f*.*

Examples:

f from $\mathbb{R}$ to $\mathbb{R}$ defined by

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

wherever this makes sense. This is a partial function with domain $\{x \in \mathbb{R} : x \geq 0\}$.

g from $\mathbb{R}$ to $\mathbb{R}$ defined by

$$g(x) = \frac{1}{x^2 - 1}$$

wherever this makes sense. This is a partial function with domain $\mathbb{R} \setminus \{1, -1\}$.

Find partial function from $\mathbb{R}$ to $\mathbb{R}$ which is left inverse of

$$f : \mathbb{R} \to \mathbb{R}, \quad x \mapsto (2x + 3)^3$$

by setting $x = 2y + 3$ and rearranging to get y on its own.

If we have a partial function f from X to Y with domain Z, then we can treat f as a function with the smaller domain Z, so a lot of what we said already about functions remains true for partial functions. We call the function $g : Z \to Y$ defined by the rule $g(z) = f(z)$ the restriction of f to Z.

We sometimes use the term total function instead of function, to stress the fact that we have a function, not just a partial function.

A partial function isn't much less than a function, and sometimes it's good enough for our purposes. Notice that when $f : X \to Y$ is a function, a partial function from Y to X, rather than a (full) function, is all we really need to 'undo' the effect of f, i.e. the left inverse only really needs to be defined on the range of f.

2.11 *Recursive functions*

The rule which defines a function is not always given explicitly. Sometimes it *could* be given explicitly, but can be given much more simply in another form, which describes the value of the function at each value of x in terms of its value at smaller values of x. In many examples, the function is defined on the positive integers, and its value at x can be defined in terms of its value at positive integers less than x.

e.g. The factorial function, $fact(n)$, also written $n!$, etc. This is a function from $\mathbb{N}$ to $\mathbb{N}$, which is defined for any positive integer n to be the product $n(n-1)\ldots 1$. It's defined very simply recursively by the two rules

$$fact(n) = n \times fact(n-1)$$

and

$$fact(1) = 1$$

So the easiest way to program it is as follows:-

(in C)

```
int factorial(int n) /* n assumed >0  */
{
  if (n==1 ) return 1;
  return n*factorial(n-1);
}
```

e.g. The function $f : \mathbb{N} \to \mathbb{N}$ defined by $f(n) = 2^n$ can be defined recursively by the rules

$$f(1) = 1, \quad f(n) = 2f(n-1)$$

and similarly easily programmed.

e.g. We can also define addition of two positive integers recursively, by the rules

$$x + y = (x+1) + (y-1)$$

and

$$x + 0 = x$$

Provided we know how to evaluate $x = 1$ and $y - 1$, we can calculate from these two rules the sum of any two positive integers.

$$\begin{aligned} 5+3 &= (5+1)+(3-1) = 6+2 \\ &= (6+1)+(2-1) = 7+1 = 8 \end{aligned}$$

Example: Where $f : \mathbb{N} \to \mathbb{N}$ is the function given recursively by the rule $f(n) = 2f(n-1)$ and $f(1) = 2$, we have

$$f(2) = 2 \times 2 = 4, \quad f(3) = 2 \times 4 = 8, \quad f(4) = 2 \times 8 = 16$$

In fact the function can be described very simply by the rule $f(n) = 2^n$.

Example: Where $f : \mathbb{N} \to \mathbb{N}$ is the function given recursively by the rule $f(n) = 2nf(n-1)$ and $f(1) = 2$, we have

$$\begin{aligned} f(2) &= 2 \times 2 \times 2 = 8. \\ f(3) &= 2 \times 3 \times 8 = 48. \\ f(4) &= 2 \times 4 \times 48 = 384 \end{aligned}$$

This function is in fact $2^n fact(n)$ (aka $2^n n!$).