Functions

Margaret M. Fleck

25 September 2009

This lecture starts the material in section 2.3 of Rosen. It discusses functions and introduces the concepts of one-to-one and onto.

1 Announcements

Another reminder of the upcoming midterm.

This lecture is brought to you by the number 65535. (This is $2^{16} - 1$, i.e. the largest number you can store in a 16-bit unsigned integer variable.)

Another useful fact is that 1000 is approximately equal to 2^{10}. This is helpful when determining how large a number you will get when trying to access locations in computer memory, especially for big memory sizes.

2 Functions

We all know roughly what functions are, from high school and (if you've taken it) calculus. You've mostly seen functions whose inputs and outputs are numbers, defined by an algebraic formula such as $f(x) = 2x + 3$. We're going to generalize and formalize this idea, so we can talk about functions with other sorts of input and output values.

Suppose that A and B are sets, then a function f from A to B (shorthand: $f : A \rightarrow B$) is an assignment of exactly one element of B (i.e. the output value) to each element of A (i.e. the input value). A is called the *domain* of f and B is called the *co-domain*.

For example, let's define $g : \mathbb{Z} \rightarrow \mathbb{Z}$ by the formula $g(x) = 2x$. The domain and co-domain of this function are both $\mathbb{Z}$.

Notice that the domain and co-domain are part of the definition of the function, just like the input/output type declarations for a function in a programming language. Suppose we define $h : \mathbb{N} \to \mathbb{N}$ such that $h(x) = 2x$. This is a **different function from** g because the declared domain and co-domain are different.

Two functions are equal if they have the same domain, the same co-domain, and assign the same output value to each input value.

The inputs and outputs to functions don't have to be numbers and a function doesn't have to be defined by an algebraic formula. It's sufficient to describe a clear, explicit procedure for finding the output value, given the input value. For example, we can define

$$s : \{\text{CS 173 course staff}\} \to \{\text{letters of the alphabet}\}$$

where $s(x)$ is the first letter in x's name. For example $s(\text{Margaret}) = M$.

For small finite sets like these, we can also just list all the input/output pairs:

Margaret $\mapsto$ M

Lucas $\mapsto$ L

Mike $\mapsto$ M

David $\mapsto$ D

Amir $\mapsto$ A

Andrew $\mapsto$ A

We also show this with a drawing like this. [draw something similar to Figures 1 in Rosen 2.3]

Notice that we use $\mapsto$ when we show the output value for a a single input value, but $\to$ to show the input and output sets for the whole function.

A function is also known as a *map* or *mapping* and we can say that some input value x *maps to* the corresponding output value y.

3 What isn't a function?

Much of the definition of a function is an association of output values with input values. Suppose I give you an association p. When is p not a function?

One possibility is that p doesn't provide an output value for every input value. For example, suppose we defined $p : \mathbb{R} \to \mathbb{R}$ such that $p(x)$ is the multiplicative inverse of x. That is, $p(x)$ is the integer y such that $xy = 1$. This isn't a function because one input value (zero) doesn't have a corresponding output value.[1]

Notice that this depends critically on what we've declared the domain to be. If we revised the type declaration for p to read $p : \mathbb{R} - \{0\} \to \mathbb{R}$, then p would be a function.

An association is also not a function if it assigns two different output values to the same input value. Suppose if I want to define the locations of top-5 CS departments. I.e. the domain is $D = \{\text{MIT}, \text{CMU}, \text{UIUC}, \text{Stanford}, \text{Berkeley}\}$ and the do-domain is the set of all cities C. I might define c as:

MIT $\mapsto$ Cambridge

CMU $\mapsto$ Pittsburgh

UIUC $\mapsto$ Urbana

UIUC $\mapsto$ Champaign

Berkeley $\mapsto$ Berkeley

Stanford $\mapsto$ Palo Alto

This isn't a function because UIUC is mapped to two distinct output values. When we need to have a function return multiple values, we need to return them as sets. For example, we might define $c : D \to \mathbb{P}(C)$ by:

MIT $\mapsto$ {Cambridge}

CMU $\mapsto$ {Pittsburgh}

UIUC $\mapsto$ {Urbana, Champaign}

---

[1] Almost functions that don't provide an output for every input are sometimes useful and are known as *partial functions*. But we won't use them in this class.

Berkeley $\mapsto$ {Berkeley}

Stanford $\mapsto$ {Palo Alto}

Since we've declared that our output values are sets, we have to make them all sets. So we have to map (say) MIT to {Cambridge} rather than to the bare string Cambridge.

Another example of this problem would be the function $h : \mathbb{R}^+ \to \mathbb{R}$ such that $h(x) = \sqrt{x}$ which maps each positive real onto its square root. We could fix this function by stipulating that we mean the positive square root.

4   Images and Onto

Suppose we have a function $f : A \to B$. If x is an element of A, then the value $f(x)$ is also known as the *image* of x. The image of f is the set

$$Im(f) = \{f(x) : x \in A\}$$

f is *onto* if its image is its whole co-domain. Or, equivalently,

$$\forall y \in B, \exists x \in A, f(x) = y$$

For example, our function from instructors to first letters of names isn't onto, because its image is the set $\{A, M, D, L\}$ which is nowhere near all the letters of the alphabet.

Suppose we define $f : \mathbb{Z} \to \mathbb{Z}$ by $f(x) = x + 2$. This function is onto. If we pick any integer y, let x be $y - 2$. Then $f(x) = f(y - 2) = (y - 2) + 2 = y$.

Now suppose we define $g : \mathbb{N} \to \mathbb{N}$ using the same formula $g(x) = x + 2$. g isn't onto, because the values 0 and 1 have no pre-images.

Warning: whether a function is onto or one-to-one depends on how we've declared its domain and co-domain. When we're discussing these properties, it is absolutely critical to declare the input/output types for all the functions you are using.

5   Negating expressions with multiple quantifiers

Let's use our definition of onto as an excuse to think about negating formal statements containing more than one quantifier. Our definition was:

$$\forall y \in B, \exists x \in A, f(x) = y$$

So a function f is not onto if

$$\neg \forall y \in B, \exists x \in A, f(x) = y$$

To negate this, we proceed step-by-step, moving the negation inwards. You've seen all the identities involved, so this is largely a matter of being careful.

$$\neg \forall y \in B, \exists x \in A, f(x) = y$$

$$\equiv \exists y \in B, \neg \exists x \in A, f(x) = y$$

$$\equiv \exists y \in B, \forall x \in A, \neg(f(x) = y)$$

$$\equiv \exists y \in B, \forall x \in A, f(x) \neq y$$

So, if we want to show that f is not onto, we need to find some value y in B, such that no matter which element x you pick from A, $f(x)$ isn't equal to y.

6   Warning about variations in terminology

It used to be that people used the term "range" to refer to the co-domain. However, formal mathematics has standardized on the term "co-domain" for the declared set of possible output values for the function. When the term "range" is used, it is as a synonym for "image," i.e. the actual output values produced when you feed in all possible input values. (Rosen uses "range" with this meaning.)

We'll try to stick carefully to the newer convention in this class. **Please avoid the term range.** But be aware that older authors and authors outside math/CS may use the terms differently.

The terms "injective" and "surjective" are fancy synonyms for one-to-one and onto. No more and no less. It's important to get used to both versions of each term, because individual mathematicians often alternate, even within a single lecture.

7 Pre-images and one-to-one

A function is *one-to-one* if it never assigns two input values to the same output value. That is

$$\forall x, y \in A, x \neq y \rightarrow f(x) \neq f(y)$$

or, equivalently,

$$\forall x, y \in A, f(x) = f(y) \rightarrow x = y$$

(These two versions are equivalent because they are the contrapositives of one another.)

For example, let $g : \mathbb{Z} \rightarrow \mathbb{Z}$ be defined by $g(x) = 2x$. g is one-to-one. But remember our function s that mapped CS 173 instructors to the first letter of their names. It's not one-to-one because Margaret and Mike share the same output value M.