Fun With Functions

Rob Hackman

Fall 2025

University of Alberta

Table of Contents

What is a function?

You should have worked with functions in the past, most likely in a maths course.

What is a function?

You should have worked with functions in the past, most likely in a maths course.

You probably have some concept of what a function is. Most likely your idea of a function is much more limited than what a function actually is.

What is a function?

You should have worked with functions in the past, most likely in a maths course.

You probably have some concept of what a function is. Most likely your idea of a function is much more limited than what a function actually is.

At a high level a function is nothing more than a relationship between two collections of items.

Recall what you know about functions

A function *maps* values from an input set to an output set

Recall what you know about functions

A function *maps* values from an input set to an output set

- The input set is called the *domain* of the function

Recall what you know about functions

A function *maps* values from an input set to an output set

- The input set is called the *domain* of the function
- The output set is called the *codomain* of the function

Recall what you know about functions

A function *maps* values from an input set to an output set

- The input set is called the *domain* of the function
- The output set is called the *codomain* of the function
- The set of values that can actually be produced by the function is called the *range*[1] of the function, a subset of the codomain

[1] Range can be defined as either synonymous with codomain or as above (which is also called the *image* of the function).

Recall what you know about functions

A function *maps* values from an input set to an output set

- The input set is called the *domain* of the function
- The output set is called the *codomain* of the function
- The set of values that can actually be produced by the function is called the *range*[1] of the function, a subset of the codomain
- Most likely you have worked with functions that map from the set of real numbers to the set of real numbers. The set of real numbers is denoted $\mathbb{R}$

[1] Range can be defined as either synonymous with codomain or as above (which is also called the *image* of the function).

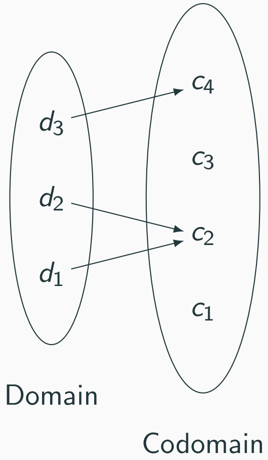
Function rules

Below are a few additional rules, and elaborations on the above slide, to keep in mind

- A function must map *every* value from its domain to a *single* value in its codomain
- Not every value in the codomain must be producible from the function, the values that are producible are the range

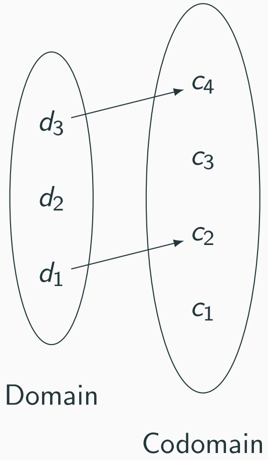
Visualization of a function

Below is a valid function, note how *every* value in the domain is mapped to a value in the codomain. Not every value in the codomain is producible.



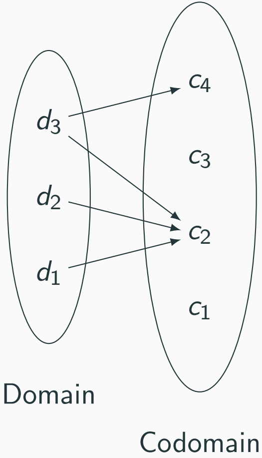
Visualization of a partial function

An entity defined exactly as above but removing the restriction that it must be applicable to *every* value in its domain is called a *partial function*. A partial function is depicted below.



Visualization of something that is not a function

Whether a function or a partial function each value in the domain that is mapped by the function must map to *one* value in the codomain. The visualization below is an example of something that is *not* a function.



The identity function

This function is called the identity function — it is called such because it maps its input value to itself.

$$ident(x) \longrightarrow x$$

The above is called a *function definition*. It defines how a function maps its domain to its codomain.

Anatomy of a function definition

$$f(x) \longrightarrow x^2 + 5$$

Anatomy of a function definition

$$\underset{\text{Function Name}}{f}(x) \longrightarrow x^2 + 5$$

Anatomy of a function definition

Parameter List

$$f\,(x) \longrightarrow x^2 + 5$$

Function Name

Anatomy of a function definition

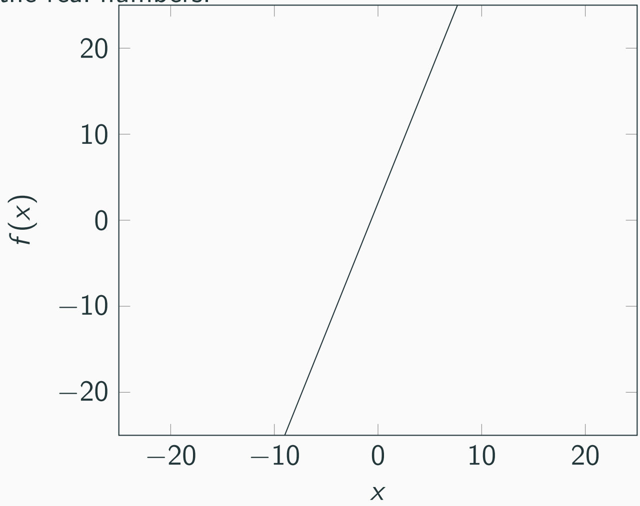
Parameter List

$$f(x) \longrightarrow \underbrace{x^2 + 5}_{\text{Function Body}}$$

Function Name

The identity function

Below is the graph of mapping the function $f(x) \longrightarrow 3x + 2$ over the real numbers.



Function applications

A *function application*[2] is how one applies a function to a specific value from its domain to produce a resultant value from its range.

In a function application *arguments* are provided in order to be substituted for the parameters within the function body.

The evaluation of the function body after substituting the arguments for their respective parameters is the result of the function application.

[2] Often called a *function call* in programming

Anatomy of a function application

$$f\,(5)$$

Anatomy of a function application

$f\,(5)$

Function Name

Anatomy of a function application

Argument List

$$f\,(5)$$

Function Name

Anatomy of a function application

Argument List

$$f\,(5)$$

Function Name

Read as "*f* applied to 5", or often simply "*f* of 5".

Example function application

Consider the following evaluation of function f, per the definition of f given earlier in the slides

$$\begin{aligned} & f(3) \\ = & (3)^2 + 5 \\ = & 14 \end{aligned}$$

Function domain and codomain values

The domain and codomain of a function do not need to be sets of the same type of values, for example the *floor*, denoted as $\lfloor x \rfloor$, and *ceiling*, denoted as $\lceil x \rceil$ functions.

The domain of the floor and ceiling functions is $\mathbb{R}$ (the set of real numbers) while the codomain of the functions is $\mathbb{Z}$ (the set of integers).

Definition of floor

$$\lfloor x \rfloor \longrightarrow max\{m \in \mathbb{Z} \mid m \leq x\}$$

Definition of floor

$$\lfloor x \rfloor \longrightarrow max\{m \in \mathbb{Z} \mid m \leq x\}$$

Read as “is a member of” or simply “in”

Definition of floor

Read as "such that"

$$\lfloor x \rfloor \longrightarrow max\{m \in \mathbb{Z} \mid m \leq x\}$$

Read as "is a member of" or simply "in"

Definition of floor

Read as "such that"

$$\lfloor x \rfloor \longrightarrow max\{m \in \mathbb{Z} \mid m \leq x\}$$

Read as "is a member of" or simply "in"

The above function definition can be read as "$\lfloor x \rfloor$ is the maximal value m in the set of integers such that m is less than or equal to x"

Example applications of the floor function

$$\lfloor 3.7 \rfloor \longrightarrow 3$$
$$\lfloor 231.5328 \rfloor \longrightarrow 231$$
$$\lfloor \pi \rfloor \longrightarrow 3$$

Recall: piecewise functions

You have likely seen at sometime in your mathematics career a *piecewise function*.

A piecewise function is simply a function whose domain is divided into sections, and each section of the domain gets its own definition of the function.

"Piecewise" is just a way of describing the way the function is defined – ultimately it is not a property of the function.

A common example of a piecewise function is the *absolute value* function, denoted as $|x|$.

Absolute value definition

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$

The absolute value function has a domain and codomain of $\mathbb{R}$, and its range is the non-negative real numbers. It maps any real number to is *magnitutde*.

Domains and codomains

You may have only ever worked with functions in your maths courses, and in these cases your domains and codomains were probably always some type of numbers as in our previous examples.

Domains and codomains

You may have only ever worked with functions in your maths courses, and in these cases your domains and codomains were probably always some type of numbers as in our previous examples.

A function maps values from its domain to its codomain, which are two sets of values.

Domains and codomains

You may have only ever worked with functions in your maths courses, and in these cases your domains and codomains were probably always some type of numbers as in our previous examples.

A function maps values from its domain to its codomain, which are two sets of values.

What exactly is a "set"?

Sets

Simply put a set is a collection of items, called *elements*. Each element is distinct, meaning a set contains no duplicates.

Sets

Simply put a set is a collection of items, called *elements*. Each element is distinct, meaning a set contains no duplicates.

Sets may have a finite or infinite number of elements depending on their definition.

Sets

Simply put a set is a collection of items, called *elements*. Each element is distinct, meaning a set contains no duplicates.

Sets may have a finite or infinite number of elements depending on their definition.

There are no restrictions on the types of elements that can be within a set!

Sets

Simply put a set is a collection of items, called *elements*. Each element is distinct, meaning a set contains no duplicates.

Sets may have a finite or infinite number of elements depending on their definition.

There are no restrictions on the types of elements that can be within a set!

The implication for us is that our functions can map values of any type to values of any type, not just numbers!

The Pig Latin function

We can define a function that operates on anything we like, for example words!

The Pig Latin function

We can define a function that operates on anything we like, for example words!

We will define the Pig Latin function, *pl*, which maps a word to its Pig Latin version.

The Pig Latin function

We can define a function that operates on anything we like, for example words!

We will define the Pig Latin function, *pl*, which maps a word to its Pig Latin version.

The first step to defining a function is defining our domain and codomain. Both our domain and codomain will be "words". However, a more formal definition will help us in defining our function.

A definition for "words"

For our purposes will define a word to be a non-empty sequence of alphabetic characters. That is, a word is some sequence $c_1 c_2 ... c_n$.

We will also define that each character is either a vowel (a, e, i, o, u) or a consonant (any other alphabetic character).

So our definition of a word is a sequence of characters, and our definition of characters specifies they are either consonants or vowels.

Rules of Pig Latin

There are two rules to Pig Latin

- If a word begins with a consonant the first character is moved to the back of the word and "ay" is added after that
- If a word begins with a vowel the word remains the same except "way" is added to the end of it.

Original: This sentence for example

Translated: histay entencesay orfay exampleway

Definition of Pig Latin function

So, given our definition of our domain and our codomain, we can now define our function formally

$$pl(c_1...c_n) = \begin{cases} c_2...c_n c_1 ay & \text{if } c_1 \text{is a consonant} \\ c1_...c_n way & \text{otherwise} \end{cases}$$

Definition of Pig Latin function

So, given our definition of our domain and our codomain, we can now define our function formally

$$pl(c_1...c_n) = \begin{cases} c_2...c_n c_1 ay & \text{if } c_1 \text{is a consonant} \\ c1_...c_n way & \text{otherwise} \end{cases}$$

We can define functions over any domain set we like — we don't even necessarily need to be able to define it as formally above, so long as we specify how it maps the domain values to the codomain values then we have defined a function!

Example of a function defined a little differently

Functions with multiple parameters

Functions can have more than one parameter, the function defined on the previous slide had two parameters — first name and birth month.

Functions with multiple parameters are not unusual. It is why in slides defining the components of a function definition and function application we had both the parameter **list** and argument **list**.

Example function with multiple arguments

When functions with multiple parameters are applied, the arguments are substituted for the parameters in the order they appear.

$$g(x, y) \longrightarrow x^2 + 3y + 2$$

$$\begin{aligned} g(2,5) &= (2)^2 + 3(5) + 2 \\ &= 4 + 15 + 2 \\ &= 21 \end{aligned} \qquad \begin{aligned} g(5,2) &= (5)^2 + 3(2) + 2 \\ &= 25 + 6 + 2 \\ &= 33 \end{aligned}$$

Table of Contents

Notation for a function's domain and codomain

As we've learned each function maps values from its domain set to its codomain set. We have not learned a notation for specifying the sets on which a function operates, so far it has been implied by the behaviour of the function.

If a function f maps from the domain set of X to the domain set of Y then this information can be noted by writing

$$f : X \to Y$$

Defining function composition

Function composition is simply the concept of applying one function to a value, and applying another function to the result of the first function.

Defining function composition

Function composition is simply the concept of applying one function to a value, and applying another function to the result of the first function.

As the result of the first function is used as the argument to the second function it must be the case that the codomain of the first function is the same as the domain of the second function.

Defining function composition

Function composition is simply the concept of applying one function to a value, and applying another function to the result of the first function.

As the result of the first function is used as the argument to the second function it must be the case that the codomain of the first function is the same as the domain of the second function.

That is if function g is composable with f such that we can construct $g \circ f$ then $f : X \to Y$ and $g : Y \to Z$, for some sets X, Y, and Z which may or may not be equivalent.

Defining function composition

Function composition is simply the concept of applying one function to a value, and applying another function to the result of the first function.

As the result of the first function is used as the argument to the second function it must be the case that the codomain of the first function is the same as the domain of the second function.

That is if function g is composable with f such that we can construct $g \circ f$ then $f : X \to Y$ and $g : Y \to Z$, for some sets X, Y, and Z which may or may not be equivalent.

Then it is true that $g \circ f$ is a function such that $g \circ f : X \to Z$

Example of function composition

Let f be the function that maps the name of a colour to its corresponding hexadecimal RGB colour code.

Example of function composition

Let f be the function that maps the name of a colour to its corresponding hexadecimal RGB colour code.

Let g be the function that maps an RGB colour code to a circle filled in with that colour.

Example of function composition

Let f be the function that maps the name of a colour to its corresponding hexadecimal RGB colour code.

Let g be the function that maps an RGB colour code to a circle filled in with that colour.

Then the function $g \circ f$ is the function that maps the English word for a colour to a circle of that colour.

Example of function composition

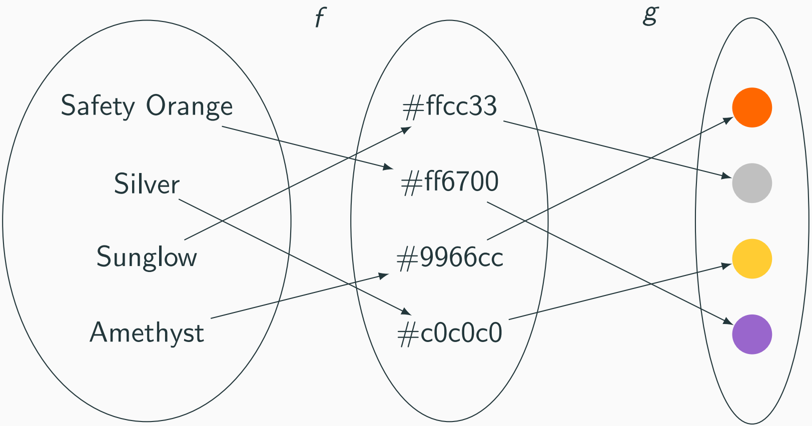
Let f be the function that maps the name of a colour to its corresponding hexadecimal RGB colour code.

Let g be the function that maps an RGB colour code to a circle filled in with that colour.

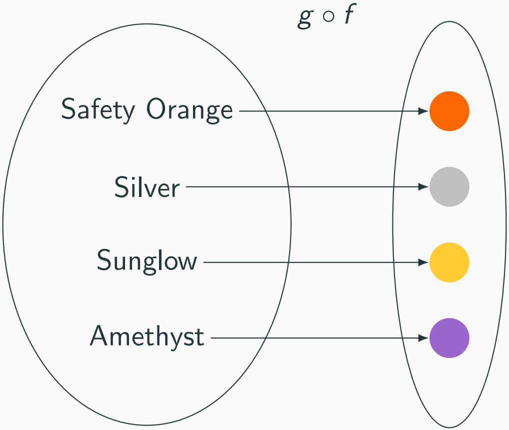
Then the function $g \circ f$ is the function that maps the English word for a colour to a circle of that colour.

Knowledge Check: While $g \circ f$ is a valid function composition $f \circ g$ is not — why?

Visualization of function composition



Visualization of function composition



More simple function composition

The syntax $f \circ g$ is the syntax for when one needs to specifically refer to the newly created function that is the result of composing f with g.

More simple function composition

The syntax $f \circ g$ is the syntax for when one needs to specifically refer to the newly created function that is the result of composing f with g.

However, as function composition really is simply the act of constructing a new function that applies two functions in order, we can achieve the same result by simply applying both functions in the order we desire, rather than defining the new function with the composition operator.

More simple function composition

The syntax $f \circ g$ is the syntax for when one needs to specifically refer to the newly created function that is the result of composing f with g.

However, as function composition really is simply the act of constructing a new function that applies two functions in order, we can achieve the same result by simply applying both functions in the order we desire, rather than defining the new function with the composition operator.

An application like $g \circ f(5)$ can instead be written as $g(f(5))$. Since a function application is an expression to be evaluated, we simply first evaluate $f(5)$ to its result and then use that result as the argument for g.

Function composition example

Consider the following two function definitions, and the application of a composition of the two of them.

$$\text{let } f(n) \longrightarrow 3n + 1 \qquad\qquad \text{let } g(n) \longrightarrow n/2$$

$$f \circ g(5) = f(g(6))$$

Function composition example

Consider the following two function definitions, and the application of a composition of the two of them.

let $f(n) \longrightarrow 3n + 1$ let $g(n) \longrightarrow n/2$

$$\begin{aligned} f \circ g(5) &= f(g(6)) \\ &= f(6/2) \end{aligned}$$

Function composition example

Consider the following two function definitions, and the application of a composition of the two of them.

let $f(n) \longrightarrow 3n+1$ let $g(n) \longrightarrow n/2$

$$\begin{aligned} f \circ g(5) &= f(g(6)) \\ &= f(6/2) \\ &= f(3) \end{aligned}$$

Function composition example

Consider the following two function definitions, and the application of a composition of the two of them.

let $f(n) \longrightarrow 3n + 1$ let $g(n) \longrightarrow n/2$

$$\begin{aligned} f \circ g(5) &= f(g(6)) \\ &= f(6/2) \\ &= f(3) \\ &= 3(3) + 1 \end{aligned}$$

Function composition example

Consider the following two function definitions, and the application of a composition of the two of them.

$$\text{let } f(n) \longrightarrow 3n+1 \qquad\qquad \text{let } g(n) \longrightarrow n/2$$

$$\begin{aligned} f \circ g(5) &= f(g(6)) \\ &= f(6/2) \\ &= f(3) \\ &= 3(3)+1 \\ &= 10 \end{aligned}$$

Further Composition

Since the domain and codomain of f and g are all the same, we can compose these functions as many times as we like.

let $f(n) \longrightarrow 3n + 1$ let $g(n) \longrightarrow n/2$

$$f \circ g \circ f \circ g(5) = f(g(f(g(5))))$$

Further Composition

Since the domain and codomain of f and g are all the same, we can compose these functions as many times as we like.

let $f(n) \longrightarrow 3n+1$ let $g(n) \longrightarrow n/2$

$$\begin{aligned} f \circ g \circ f \circ g(5) &= f(g(f(g(5)))) \\ &= f(g(f(6/2))) \end{aligned}$$

Further Composition

Since the domain and codomain of f and g are all the same, we can compose these functions as many times as we like.

let $f(n) \longrightarrow 3n + 1$ let $g(n) \longrightarrow n/2$

$$
\begin{aligned}
f \circ g \circ f \circ g(5) &= f(g(f(g(5)))) \\
&= f(g(f(6/2))) \\
&= f(g(3(3) + 1))
\end{aligned}
$$

Further Composition

Since the domain and codomain of f and g are all the same, we can compose these functions as many times as we like.

let $f(n) \longrightarrow 3n+1$ let $g(n) \longrightarrow n/2$

$$\begin{aligned} f \circ g \circ f \circ g(5) &= f(g(f(g(5)))) \\ &= f(g(f(6/2))) \\ &= f(g(3(3)+1)) \\ &= f(10/2) \end{aligned}$$

Further Composition

Since the domain and codomain of f and g are all the same, we can compose these functions as many times as we like.

let $f(n) \longrightarrow 3n+1$ let $g(n) \longrightarrow n/2$

$$\begin{aligned}
f \circ g \circ f \circ g(5) &= f(g(f(g(5)))) \\
&= f(g(f(6/2))) \\
&= f(g(3(3)+1)) \\
&= f(10/2) \\
&= 3(5)+1
\end{aligned}$$

Further Composition

Since the domain and codomain of f and g are all the same, we can compose these functions as many times as we like.

let $f(n) \longrightarrow 3n+1$ let $g(n) \longrightarrow n/2$

$$\begin{aligned}
f \circ g \circ f \circ g(5) &= f(g(f(g(5)))) \\
&= f(g(f(6/2))) \\
&= f(g(3(3)+1)) \\
&= f(10/2) \\
&= 3(5)+1 \\
&= 16
\end{aligned}$$

A note on application syntax

Very complex compositions of functions can get lousy with parentheses. Some notations omit parentheses around arguments and simply apply the function to the next expression to the right.

$$\begin{aligned} f \circ g \circ f \circ g(5) &= f\ g\ f\ g\ 5 \\ &= f\ g\ f\ 6/2 \\ &= f\ g\ 3(3) + 1 \\ &= f\ 10/2 \\ &= 3(5) + 1 \\ &= 16 \end{aligned}$$

A note on application syntax

Very complex compositions of functions can get lousy with parentheses. Some notations omit parentheses around arguments and simply apply the function to the next expression to the right.

$$\begin{aligned} f \circ g \circ f \circ g(5) &= f\ g\ f\ g\ 5 \\ &= f\ g\ f\ 6/2 \\ &= f\ g\ 3(3) + 1 \\ &= f\ 10/2 \\ &= 3(5) + 1 \\ &= 16 \end{aligned}$$

Note: We will not typically use this notation in this class. It is mentioned here for your awareness only. There are certain programming languages that use this syntax such as Haskell!