



Why

We want to have language for applying two functions one after the other. We apply a first function then a second function.

Definition

Consider two functions. Suppose the range of the first is a subset of the domain of the second. In other words, every value of the first is in the domain (and so can be used as an argument) for the second. In this case we say that the second function is *composable* with the first.

The *composite* (or *composition*) of the second function *with* the first function is the function which associates to an element in the first's domain the element in the second's codomain that the second function associates with the result of the first.

In other words, we take an element in the first's domain. We apply the first function to it. We obtain an element in the first's codomain, which is also an element in the second's domain. We apply the second function to this result. We obtain an element in the second's codomain. The composition of the second function with the first is the function so constructed. Of course the order of composition is important.

Notation

Let A, B, C be non-empty sets. Let $f : A \rightarrow B$ and $g : B \rightarrow C$. We denote the composition of g with f by $g \circ f$ read aloud as "g composed with f." To make clear the domain and codomain, we denote the composition $g \circ f : A \rightarrow C$. $g \circ f$ is defined by

$$(g \circ f)(a) = g(f(a)).$$

for all $a \in A$. Sometimes the notation gf is used for $g \circ f$.

Basic Properties

Function composition is associative but not commutative.¹ Indeed, even if $f \circ g$ is defined, $g \circ f$ may not be.

Proposition 1 (Associative). *Let $f : X \rightarrow Y$, $g : Y \rightarrow Z$ and $h : Z \rightarrow U$. Then $(f \circ g) \circ h = f \circ (g \circ h)$* ²

¹Future editions will include a counterexample.

²The proof is straightforward. Future editions will include it.

