

BASICS

A function f is a special kind of relation if: $\forall x, x', x''. ([x, x'] \in f \text{ and } [x, x''] \in f) \Rightarrow x' = x''$ i.e. for every member of the domain of the relation there is only one member of the range that it can be paired with. We can write the functional relation in many equivalent ways:

- $f x = y$
- $f_x = y$
- f maps x to y
- $[x, y] \in f$

Functions can be defined by formulas on the range. e.g. the squares of the natural numbers can be written as square: $x \rightarrow x^2$ and the successor function can be written as successor: $x \rightarrow x + 1$. The *functionality* of a function can be written as a mapping between the domain and range sets. Thus, square: $\mathbb{R} \rightarrow \mathbb{R}$ (mapping real numbers to real numbers), and successor: $\mathbb{Z} \rightarrow \mathbb{Z}$ (mapping integers to integers).

PROPERTIES

- I_S and $\{\}$ are functions (the identity function on any set S and the empty set).
- If f and g are functions, $g \circ f$ is a function: $(g \circ f)x = g(f x)$. This is function *composition*.

LAMBDA ABSTRACTIONS

A function may be written using only its parameter and an expression that computes its range as: $\lambda x \in S. E \triangleq \{[x, E] \mid x \in S\}$. Note that this is strictly a *typed* lambda abstraction, since x is constrained as an element of the set S . The expression E must be defined for all $x \in S$.

In this formulation, the identity function is:

$$I_S = \lambda x \in S. x$$

and composition as:

$$g \cdot f = \lambda x \in \text{dom } f. g(f x), \text{ if } \text{ran } f \subseteq \text{dom } g$$

The square function is written as $\lambda x. x^2$ and the successor function is $\lambda x. x + 1$. The typed versions are: $\lambda x \in \mathbb{R}. x^2$ and $\lambda x \in \mathbb{Z}. x + 1$.

UPDATING A FUNCTION

$$\text{If } f \text{ is a function: } ([x \mapsto y]f)z = \begin{cases} y, & \text{if } z = x \\ f z, & \text{otherwise} \end{cases}$$

The domain of this form is: $(\text{dom } f) \cup \{x\}$ and its range is: $((\text{ran } f) - \{u \mid [x, u] \in f \wedge \forall w. [w, u] \notin f\}) \cup \{y\}$

We can then define a multiple update: $[x_n \mapsto y_n] \cdots [x_1 \mapsto y_1] f \triangleq [x_n \mapsto y_n](\cdots([x_1 \mapsto y_1]f)\cdots)$

We can then define a sequence by starting with an empty function, and using the natural numbers as the domain: $\langle x_0, \dots, x_{n-1} \rangle \triangleq [0 \mapsto x_0] \cdots [n-1 \mapsto x_{n-1}] \{\}$

We can index this sequence to retrieve the *ith* element: $\langle x_0, \dots, x_{n-1} \rangle i = x_i$ when $i \in \{0, \dots, n-1\}$

CURRYING A FUNCTION

A function with two inputs (parameters) can be transformed into a nesting of two functions in the following manner.

$$\text{Assume } f(x, y) = 2x + y$$

This could be written in lambda form as $\lambda x, y. 2x + y$. However, it is more useful to split the function into two parts: the first function when applied to x produces a function that when applied to y gives the value of $2x + y$. This can be written as $\lambda x. (\lambda y. 2x + y)$, or just as $\lambda x. \lambda y. 2x + y$. The single parameter form is a “curried” function.

As an example, take $f(3, 4)$. This is $((\lambda x. (\lambda y. 2x + y))3)4$, which gives $(\lambda y. 6 + y)4$ and then the same result, 10, as the uncurried version.

Currying was named for Haskell Curry, but was discovered by Frege, and later by Schonfinkel. It can be seen to work in any case by appealing to the isomorphism of two sets. The first is the extension of the function $A \times B \rightarrow C$, which can be written as $\{[[a_0, b_0], c_0], [[a_1, b_1], c_1] \dots\}$, i.e. as a set of pairs where the first element of each pair is a pair of values taken from $A \times B$, and the second is an element of C . This set is isomorphic to the set $\{[a_0, [b_0, c_0]], [a_1, [b_1, c_1]] \dots\}$, which can be seen as just a rearrangement of each element by pairing the bs with the cs , instead of pairing the as with the bs . The cardinality of the two sets is the same (same number of members) and there is a one-to-one correspondance between them. This second set can be written as $A \rightarrow (B \rightarrow C)$, or simple as $A \rightarrow B \rightarrow C$.

The result can be generalized to any number of parameters:

$$A_0 \times A_1 \times \dots \times A_n \rightarrow B \text{ can be rewritten as } A_0 \rightarrow A_1 \rightarrow \dots \rightarrow A_n \rightarrow B$$