Functions

Informally, a function is a mapping that assigns to each element from a given set some element from another (or the same) set.

Well-known mathematical functions include the logarithm function(s), the addition and multiplication functions, and exponentiation functions. In computing applications one often finds functions that operate on arrays, strings, lists, or other data structures.

Formally, a function f from a set A to a set B may be defined as a subset of the Cartesian product $A \times B$ that satisfies the following two properties.

- Completeness

  For each element x of the set A, there exists an element y of B, such that the pair (x, y) is in f.

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

- Uniqueness

  The set f does not contain two pairs (x, y) and (x, z), where y and z are different.

  $$\forall x \in A\, \forall y \in B\, \forall z \in B\ [(x, y) \in f \wedge (x, z) \in f \rightarrow y = z]$$

For example, the set

$$\{(a, 1), (b, 2), (c, 1)\}$$

defines a function from $\{a, b, c\}$ to $\{1, 2\}$, that maps a to 1, b to 2, and c to 1.

Note. Subsets of $A \times B$ that satisfy the uniqueness property but not the completeness property are sometimes called *partial functions* from A to B.

Domains and Codomains

We use the notation $f : A \rightarrow B$ to indicate that f is a function from A to B and denote by $f(a)$ the (unique) element in B for which $(a, f(a)) \in f$.

We also say that $f(a)$ is the result of applying the function f to the *argument* a.

For example, the *squaring function* is a function $f : \mathbf{R} \rightarrow \mathbf{R}$ such that $f(x) = x^2$, for all real numbers x.

A *constant function* (on the integers) is a function $f : \mathbf{Z} \rightarrow \mathbf{Z}$ such that $f(n) = k$, for all integers n, where k is a fixed value, e.g., $k = 2$.

The *identity function* on a set A is the function $id : A \rightarrow A$ such that $id(x) = x$, for all $x \in A$.

We call the set A the *domain* of the function f, and the set B, the *codomain*.

Note that a function must assign a value to each domain element, but that not all elements of the codomain need to be equal to $f(a)$, for some element a in the domain of the function.

The set $\{y \in B : y = f(x) \text{ for some } x \in A\}$ is called the *range* of the function f.

Range and codomain of a function may be different.

Multiple-Argument Functions

Functions of two or more arguments may be viewed as standard one-argument functions where the domain is a set of tuples (e.g., pairs or triples).

For example, a *binary* function is a function f of type $f : A_1 \times A_2 \rightarrow B$.

Example. The addition function on the integers is a binary function that maps each pair of integers (m, n) to their sum $m + n$.

In general, by an *n-ary function* we mean a function of type

$$f : A_1 \times A_2 \times \cdots \times A_n \rightarrow B$$

the domain of which is a set of n-tuples.

It is of course also possible for the codomain of a function to be a set of pairs or tuples.

For example, we may define a function $f : \mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{Z} \times \mathbf{Z}$ such that $f(m, n) = (q, r)$, where q and r are the quotient and remainder, respectively, of the integer division of m by n.

Equality of Functions

Lemma.

> Two functions f and g from A to B are equal if they agree on all arguments, i.e., $f(x) = g(x)$ for all $x \in A$.

For example, let f and g be functions on the integers such that $f(n) = n^2 - 1$ and $g(n) = (n+1)(n-1)$. Then $f = g$.

The set-theoretic definition reflects an abstract view of functions that does not capture essential computational aspects, but can be very useful in reasoning about computationally defined functions.

Consider, for example, the ML function,

```
fun f(n) = if n > 100 then n-10 else 91;
```

What is the set-theoretic description of this function?

$$f = \{(n, 91) \ : \ n \in \mathbf{Z}, \ n \leq 100\} \\ \cup\{(n, n-10) \ : \ n \in \mathbf{Z}, \ n > 100\}$$

Compare f with the following ML function, known as "McCarthy's 91 function:"

```
fun ff(n) = if n > 100 then n-10 else ff(ff(n+11));
```

In which way differ the two functions?

Take further examples of ML functions:

```
fun f(n) = if n = 0 orelse n=1 then 1
=              else f(n-1) + f(n-2);

fun g(i,j,k,n) = if k = n then j
=              else g(j,i+j,k+1,n);

fun h(n) = g(0,1,0,n);
```

What are domains and codomains?

Can you describe the range of each functions?

Can you give suitable set-theoretic definitions?

Multisets

Informally, a *multiset* or *bag* is a collection, possibly with repeated occurrences of elements, in which the order of elements does not matter.

Multisets are often described by listing the elements, though brackets are used instead of braces to differentiate them from sets.

For example, $M = [1, 1, 1, 2, 3, 3]$ denotes a bag with three occurrences of 1, one occurrence of 2, and two occurrences of 3.

Formally, a multiset of elements from a set A can be defined as a function $M : A \to \mathbf{N}$; the intuition being that for each element $x \in A$, $M(x)$ denotes the number of occurrences of x in M.

Multiset Operations

Let M and N denote multisets over a set A.

Element relationship

> $x \in M$ if and only if $M(x) > 0$.

Submultiset relationship

> $M \subseteq N$ if and only if $M(x) \leq N(x)$, for all $x \in A$.

Binary operations for the sum, union, and intersection, respectively, of two multisets M and N over A are defined by specifying suitable functions in terms of M and N.

Sum of multisets

> $(M + N)(x) = M(x) + N(x)$, for all $x \in A$.

Union of multisets

> $(M \cup N)(x) = \max(M(x), N(x))$, for all $x \in A$.

Intersection of multisets

> $(M \cap N)(x) = \min(M(x), N(x))$, for all $x \in A$.