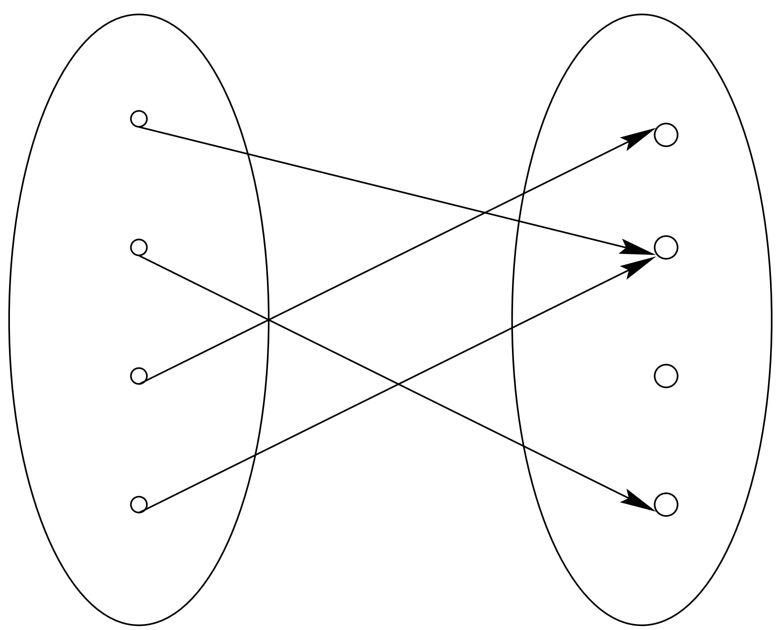
Functions

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

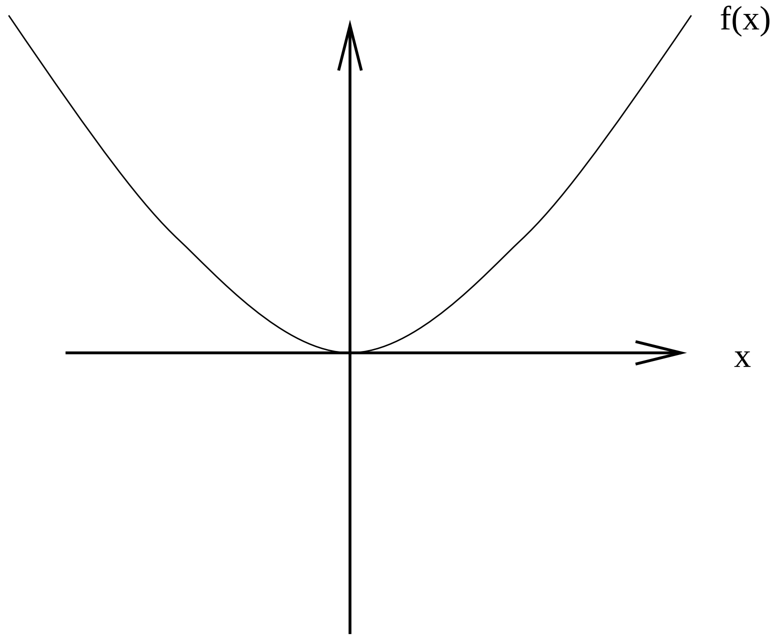
Well-known mathematical functions include the logarithm function(s), the addition and multiplication functions, and exponentiation functions. In computing applications one often finds functions that operate on arrays, strings, lists, or other data structures.

Functions on finite sets can be described by arrow diagrams.



Functions on Infinite Domains

Mathematical functions defined on infinite sets are usually drawn as graphs, e.g., the function that maps each real number x to x^2, or the function that maps x to $x^3 + x$, etc. The values along the y-coordinate are defined by an expression in terms of x, e.g., $y = x^2$ or $y = x^3 + x$.



But not all identities in variables x and y define a function. For instance, the formula for a circle, $x^2 + y^2 = 1$, does not define a function.

Set-Theoretic Definition of Functions

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

Domains and Codomains

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

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

A *constant function* (on the integers) is a function $f : \mathbf{Z} \to \mathbf{Z}$ 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 $id : A \to A$ such that $id(x) = x$, for all $x \in A$.

We call A the *domain* of the function f, and B, the *codomain*. We also speak of "a function from A to B" or, especially if $A = B$, "a function on A."

Note that a function must assign a value to each domain element, but 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.

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

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 $f : A_1 \times A_2 \to B$.

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

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

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.

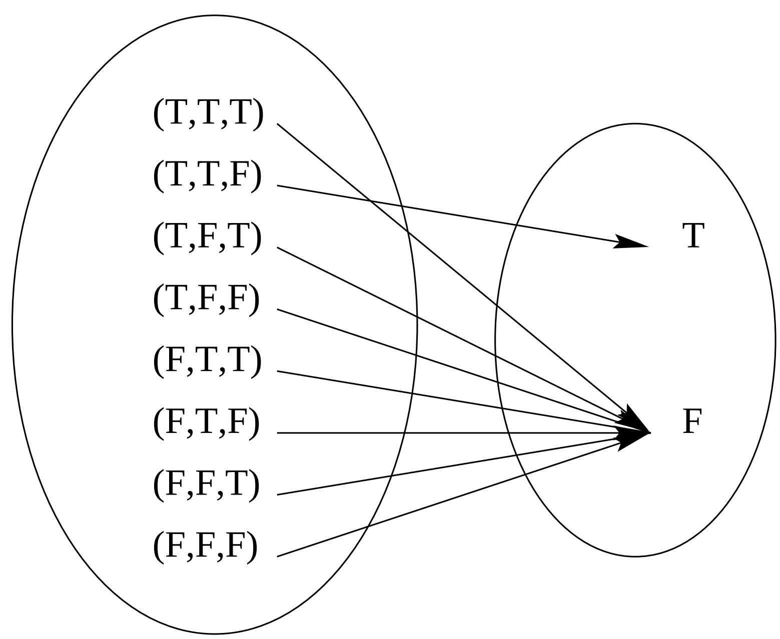
For example, we may define a function $f : \mathbf{Z} \times \mathbf{Z} \to \mathbf{Z} \times \mathbf{Z}$ such that $f(m, n) = (q, r)$, where q and r are the quotient and remainder, respectively, of the integer division of m by n.

Boolean Functions

Truth tables describe functions, called *Boolean functions*, that map n-tuples of truth values to single truth values.

On the other hand, every propositional formula A defines a truth table, and hence a (unique) Boolean function.

For example, the formula $p \wedge q \wedge \neg r$ defines the following Boolean function.



Partial Functions

Is the (multiplicative) *inverse mapping*, which assigns to each rational number m/n the number n/m a function on the rational numbers?

> *No*, because n/m is not defined if $m = 0$.

It is a function on the non-zero rational numbers, though.

The formal definition we gave above characterizes so-called *total* functions, that is, a mapping f that assigns a value from B to *each* argument from the specified domain A.

In computer science one usually has to deal with "partial functions," such as division, which may be "undefined" for certain arguments.

In set-theoretic terms, a *partial function* is a subset of $A \times B$ that satisfies the uniqueness property, but not the completeness property.

Each partial function can of course be viewed as a total function on a more restricted domain, but there are a number of (theoretical and practical) issues about partial functions and function domains that are especially important in computing applications.

Equality of Functions

Lemma.

> Two (total) 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?

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

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?