Important Definitions
from "Category Theory for Programmers" by Bartosz Milewski

Jessika Rockel

December 9, 2020

1 Categories

A category consists of objects and arrows (morphisms) between them.

Every object must have an identity morphism and for any two composable morphisms the composition must also be a morphism.

1.1 Composition

For two functions $f :: A \to B$ and $g :: B \to C$ their composition is defined as: $g \circ f = \lambda x.g(f(x))$.

Composition is associative: $h \circ (g \circ f) = (h \circ g) \circ f$

The identity morphism is a unit of composition: $f \circ \mathbf{id}_A = f$ and $\mathbf{id}_B \circ f = f$.

1.2 Homset

A set of morphisms from object a to object b in a category $\mathbf{C}$ is called a *homset* and is written as $\mathbf{C}(a, b)$ or sometimes $\mathbf{Hom}_{\mathbf{C}}(a, b)$

1.3 Thin Category

A category is called *thin* when there is at most one morphism going from any object a to any object b.

2 Types

A Type is a set of values. That set may be finite or infinite.
Haskell Types also implicitly include the value *bottom* (written as $\perp$), which represents nontermination.

2.1 Examples of Types

`Void` is the type corresponding to the empty set. It is a type that is not inhabited by any values. A function taking `Void` as an argument can be defined, but it can never be called. The return type of this function (called `absurd` in Haskell) can be anything.

`()` (pronounced "unit") is the type corresponding to a singleton set. It has only one possible value.

`Bool` is the type corresponding to a set with exactly two values, `True` and `False`. Functions to `Bool` are called *predicates*.

3 Pure Functions

A pure function is a function that always produces the same output given the same input and does not have any side effects (such as shared memory modification, I/O,...).

4 Free Construction

The process of completing a given structure by extending it to satisfy its basic laws (such as the laws of a category in the case of a *free category*).

5 Orders

Orders are special types of relations:

A *preorder* is reflexive and transitive.
A *partial order* is a preorder where $a \leq b$ and $b \leq a$ implies $a = b$.

The additional comdition that any two objects are in a relation with each other makes it a *linear order* or *total order*.

6 Monoid

6.1 Set Monoid

A monoid is a set with a binary operation. That operation must be associative and there must be a special element in the set that acts as a unit with respect to the operation.
An example of this are the natural numbers with zero, which form a monoid under addition. Addition is associative ($a + (b + c) = (a + b) + c$) and the neutral element is zero ($a + 0 = a$ and $0 + a = a$).

6.2 Category Monoid

A categorical monoid is a one-object category with some number of morphisms.

We can get a set monoid from a categorical monoid where the set is the homset $\mathbf{M}(m, m)$ and the binary operation is composition.