Category Theory in Haskell

Kenneth Watkins

June 5, 2022

1 What is a Category

A Category consist of two things

- Objects
- Arrows - Morphisms

To form a category objects and arrows must satisfy the following

- Objects must have a an arrow originating from itself to itself as the Identity morphism. This arrow serves as a unit of composition such that when composed with any Arrow that either starts at A or ends at A it gives back the same arrow. So if f is an arrow f . (idA) = f
- Given an object A, B, C and two arrows, one from A to B and B to C there must exist a third arrow from A to C.
- Given 3 arrows f, g, h, the they must be associative h . (g . f) = (h . g) . f = h . g . f

2 Haskell Types and the Category of Set

A type is a set of values. As an example the Haskell type Bool is a two element set of True and False.

Sets can be finite or infinite

x :: Integer is saying x is an element of Integer

The category of sets is called Set. Its special because we can peak in at its objects.

In Set...

- Objects are sets
- Arrows are morphisms

Intutions about Set

- empty set has no elements

- there are special one element sets
- functions map elements of one set to elements of another set
- functions can map two elements to one but not one element to two
- there exist an identity function that maps each element of a set to itself

  In Haskell the category of haskell types and functions is referred to as Hask. By forgetting the bottom hask is Set

Set	Haskell Type	Typescript Type	Description
Empty Set	Void	never	Type not inhibited by any values
Singleton Set	() "Unit"	void	Type has one value that always exist
Two Element Set	Bool	boolean	true or false functions to this type are called pred