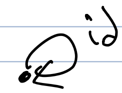
The Null Category $\varnothing$

Contains no objects

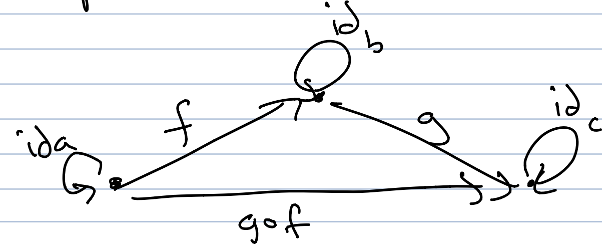
Category of 1 Object $\mathbf{1}$

Since it contains an object, it must have atleast 1 arrow, the identity



A category is a graph w/ certain arrows, namely:

1. Identities
2. Compositions ($f_{ab} \wedge g_{bc} \to g \circ f$)



Any graph that is not a category can be augmented into a category. This process is called **free construction**

An **Order** is a category where arrows represent relations, not functions.

example

$$a \xrightarrow{\le} b$$

Notice that order relations still must be composable and contain identities.

$$\begin{tikzcd} a \arrow[loop above, "\le"] \arrow[r, "\le"] \arrow[rr, bend right, "\le"'] & b \arrow[r, "\le"] & c \end{tikzcd}$$

Categories that can only contain one arrow between two objects is a **thin category**. There is a one-to-one correspondence between any thin category and a preorder.

A **Hom-set** is the set of arrows of one object to another.

$$\begin{tikzcd} a \arrow[r, bend left, "f"] \arrow[r, "g" description] \arrow[r, bend right, "h"'] & b \end{tikzcd} \qquad C(a,b) = \{f, g, h\}$$

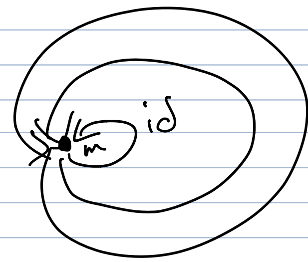
A **partial order** <u>corresponds</u> to a category where an arrow from 'a' to 'b' implies no arrow from 'b' to 'a'.

A **total order** <u>corresponds</u> to a category where there is an arrow btwn all objects

A thin category by definition contains only epic and monic arrows. However, if the category corresponds to a partial order, then a given arrow is not invertible by definition.

This example demonstrates that unlike functions, arrows can be bijective yet still non-invertible.

A **monoid** is a category with one object



Notice any 2 arrows are composable

$f: m \to m \quad g: m \to m \quad g \circ f: m \to m$

This definition matches the set theory definition:

Associativity: $(ab)c = a(bc)$
Identity : $\exists e \forall a \quad ea = ae = a$

Infact the hom-set $C(m,m)$ is an isomorphism to the multiplication monoid.

A programming analogy
Strong types systems are represented through composition. Not any two functions compose

$f: (int \to float) \quad g: (string \to bool)$

$g \not\circ f$

A monoid represents a weakly typed system because any function composes with any other.