Definition 1. A functor $F : C \to D$ between categories C and D is a mapping of objects to objects and arrows to arrows in such a way that

1. $F(f : A \to B) = F(f) : F(A) \to F(B)$,
2. $F(g \circ f) = F(g) \circ F(f)$,
3. $F(1_A) = 1_{F(A)}$.

For each category C, we have the identity functor $1_C : C \to C$ taking each object and morphism to itself. One can then show that it is possible to form the categories Cat of small categories and functors and CAT, the category of locally-small categories and functors.

Consider the following theorem from Group Theory:

Theorem 1 (Cayley)**.** *Every group* G *is isomorphic to a group of permutations.*

With such a theorem for groups in mind, observe that it is also possible to think of small categories in a "concrete" sense as well:

Theorem 2. *Every (small) category* C *is isomorphic to a category in which the objects are sets and the arrows are functions.*

Sketch proof. Define the Cayley representation $\overline{C}$ of the category C to be the following category:

- objects are sets of the form $\overline{c} := \{f \in C \mid \text{cod } f = c\}$,
- arrows are functions $\overline{g} : \overline{c} \to \overline{c'}$ for $g : c \to c'$ in C, defined for any $f : x \to c$ in $\overline{c}$ by $\overline{g}(f) := g \circ f$.

□

Note that the term concrete is motivated by the following definition:

Definition 2. A category C is called concrete if there is a functor $U : C \to \text{Set}$ such that for any pair of objects $c, c' \in C$, the induced functions

$$U_{c,c'} : \text{Hom}(c, c') \to \text{Hom}(Uc, Uc')$$

are injective. This condition is also referred to as being *faithful*.

In light of this, we see Theorem 2 states that every small category is concrete.

Now, let's produce some new categories out of existing ones.

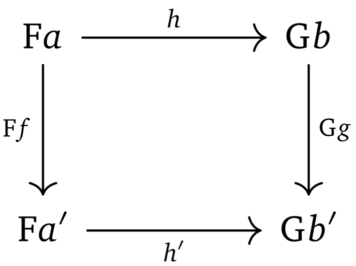
(1) The product of two categories C and D, written $C \times D$ has objects of the form (c, d) for $c \in C, d \in D$ and arrows of the form $(f, g) : (c, d) \to (c', d')$ for $f : c \to c' \in C$ and $g : d \to d' \in D$. The composition and identities are defined componentwise, and notably, the category has two "projection" functors

$$C \xleftarrow{\pi_1} C \times D \xrightarrow{\pi_2} D$$

(2) The opposite (or dual) category C^{op} of a category C has the same objects as C and has arrows $f^* : c^* \to c'^*$ for each arrow $f : c' \to c$ in C; that is, it is the corresponding category with all arrows "formally turned around". Furthermore, the identities are defined to be $1_{c^*} := (1_c)^*$ and composition by

$$f^* \circ g^* := (gf)^*$$

(3) Given two functors $F : A \to C$ and $G : B \to C$, the comma category $F{\downarrow}G$ is the category with objects $(a, b, h : Fa \to Gb)$ where $a \in A$, $b \in B$ and $h \in C$, and arrows $(f, g) : (a, b, h) \to (a', b', h')$ for $f : a \to a'$ and $g : b \to b'$ such that the following diagram commutes:



Notice that this category has two "projection" functors, namely: $F{\downarrow}G \to A$, where we take objects to the corresponding objects in A and arrows to the corresponding arrows in A, i.e., in this case, (a, b, h) to a and $(f, g) : (a, b, h) \to (a', b', h')$ to $f : a \to a'$. We have a similar functor for $F{\downarrow}G \to B$.

As special examples of this category, consider the following:

i The arrow category $C^{\to}$ is obtained by $1_C{\downarrow}1_C$, where 1_C is the identity functor. Notice that the projection functors in this case can be labeled as dom and cod (show this!)

ii Consider the slice and coslice categories for a dedicated object $c \in C$. Notice that for a dedicated object, we have the functor $c : \mathbb{1} \to C$ from the single object category to our category C, taking the single object and its identity morphism to the object c and the respective identity morphism in C. Then, the slice and coslice categories are just $c{\downarrow}1_C$ and $1_C{\downarrow}c$, or more simply written, just $c{\downarrow}C$ and $C{\downarrow}c$.

Definition 3 (Free monoid). Start with an "alphabet" A of "letters"

$$A = \{a, b, \dots\}$$

- A word over A is a finite sequence of letters
- We write "␣" for the empty word.
- The "Kleene closure" of A is defined to be the set A^* of all words over A.

Notice that A^* is a monoid with concatenation and the empty word.