0.1 Morphism with no inverse in the category of posets and monotone functions

Unlike **Set**, where bijections are isomorphisms, in the category of **posets and monotone functions**, not every bijection is an isomorphism.

For instance, given the following sets:

$A = \{1, 2\}$ where A is discrete

$B = \{1, 2\}$ where $1 \leq 2$

$A \to B$ could map $2_A \mapsto 1_B$ and $1_A \mapsto 2_B$, which is valid because elements in A are disconnected.

$B \to A$ would not be able to map $1_B \mapsto 2_A$ and $2_B \mapsto 1_A$ (in reverse) and keep order because $2_A \not\leq 1_A$

0.2 Initial and terminal objects for the category of posets and monotone functions

(Initial object)

When we regard a **poset** as a category, then the initial object is the **smallest element**, if it exists. This is because there is a single morphism between the smallest object and any other object; the smallest object is always smaller than any other object.

In the **category of posets**, however, we have to consider the poset for which there is a unique monotone function between it and every other object.

Analogous to **Set**, this would be the **empty poset** $(\varnothing, \varnothing)$, which is discrete, indiscrete, and has no relations (is a flat poset).

(Terminal object)

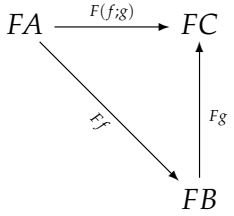
When we regard a **poset** as a category, then the terminal object is the **largest element**, if it exists. This is because there is a single morphism from any object to the largest object; the largest object is always larger than any other object.

In the **category of posets**, analogous to **Set**, this would be the **singleton equality poset** $(1, =)$.

0.3 Functors

A functor $F : \mathcal{C} \to \mathcal{D}$ is a function that maps between categories $\mathcal{C}$ and $\mathcal{D}$ by mapping all definitions from $\mathcal{C}$ to $\mathcal{D}$ such that:

1. all objects in $\mathcal{C}$ are mapped to all objects in $\mathcal{D}$ by F
   $ob\mathcal{C} \xrightarrow{F} ob\mathcal{D}$ i.e. $X \in \mathcal{C}$ then $F(X) \in \mathcal{D}$
2. all morphisms in $\mathcal{C}$ are mapped to a morphism in $\mathcal{D}$
   $\forall A, B \in \mathcal{C} \wedge f : A \to B \,.\, FA \xrightarrow{F_{A,B}f} FB$
3. the identity morphisms are also mapped and maintained
   $\forall A \in \mathcal{C} \,.\, Fid_A = id_{FA}$
4. composition is also mapped and maintained
   $\forall A, B, C \in \mathcal{C} \wedge A \xrightarrow{f} B \xrightarrow{g} C \,.\, F(f \underset{A,B}{;} g) = Ff \underset{FA,FB}{;} Fg$



Functors can also be thought of as homomorphisms between categories.

0.4 More categories

0.4.1 Monoids

Any monoid $(M, e, \bullet)$ gives rise to a category $\tilde{M}$ if:

- $ob\tilde{M} \stackrel{\text{def}}{=} 1$ where 1 is any singleton set. e.g. the unit set $\{()\}$
- $\tilde{M}((),()) \stackrel{\text{def}}{=} M$ morphisms in $() \to () \in \tilde{M}$ are the elements of M
- $id_{()} \stackrel{\text{def}}{=} e$ the identity morphism is the identity element $e \in M$
- $() \xrightarrow{m} () \xrightarrow{m'} () \stackrel{\text{def}}{=} m \bullet m'$ composition is the monoid binary relation $(\bullet)$, e.g. concatenation

0.4.2 Preord and preordered classes

Preord is the category of preordered sets and monotone maps.

Any preordered class (a class can be as big as the universe of all sets) $(A, \leq)$ gives rise to a category $\hat{A}$ if:

- objects in $\hat{A}$ are elements in A
  $ob\hat{A} \stackrel{\text{def}}{=} A$

- the set of morphisms are either the singleton set or the empty set

  $\forall x, y \in A \,.\, \hat{A}(x,y) \stackrel{\text{def}}{=} \begin{cases} 1 & x \leq y \\ \varnothing & x \not\leq y \end{cases}$

- the identity morphism is the element of the singleton set

  $id_x \stackrel{\text{def}}{=} ()$

- composition is the element of the singleton set, so all objects are related by this element.

  $x \xrightarrow{()} y \xrightarrow{()} z \stackrel{\text{def}}{=} ()$

0.4.3 Groupoid

A **groupoid** is a category where every morphism is an isomorphism.

The category of sets and bijective functions, for instance, is a **groupoid**.

0.5 Functor examples

0.5.1 Revisiting the inverse of town routes

To solve the problem for whether any non-identity morphism has an inverse in the category of towns and routes, we can define the following functor:

F : **Town** $\to$ **Nat** $= \widetilde{(\mathbb{N}, 0, +)}$ i.e. from **Town** to the monoid of natural numbers

- $\forall$ *town* $\in$ **Town** . *town* $\mapsto () \in$ **Nat**
- $(f : x_0 \to x_n = x_0 \xrightarrow{f=[x_0, \ldots, x_n]} x_n) \mapsto n \in$ **Nat**
- $(id_{x_0} = x_0 \xrightarrow{[x_0]} x_0) \mapsto 0 \in$ **Nat**
- $(f;g = x_0 \xrightarrow{f=[x_0, \ldots, x_n]} x_n \xrightarrow{g=[x_n, \ldots, x_m]} x_m) \mapsto n + m \in$ **Nat**

Isomorphisms are preserved by functors.

Morphisms are equal to the route lengths in **Nat**.

Composition is concatenation of routes in **Town**, which translates to addition in **Nat**.

An identity morphism is the singleton route in **Town** and zero (0) in **Nat**.

$\therefore$ Since adding (composing) any number with a non-zero natural number (morphism) in **Nat** will never produce the identity (0), and **Town** is mapped to **Nat** by F (preserving isomorphisms), concatenating two routes that are not the identity will never produce the identity route either. Thus, no non-identity route has an inverse.

0.5.2 Define a functor $C^{op} \times C \to$Set—the hom-functor

Given category C, define functor $F : C^{op} \times C \to$ **Set**

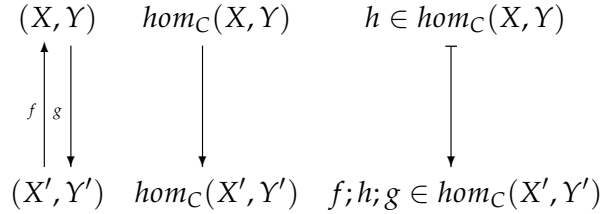
First, we know category $C^{op} \times C$ consists of:

- **(objects)** $(X, Y) \in C^{op} \times C$ where $X \in C^{op}$ and $Y \in C$.
  i.e. $X, Y \in C$ since objects in C are in C^{op}
- **(morphisms)** $(X, Y) \mapsto (X', Y') \in C^{op} \times C$ is (f, g) where
  $f : X \to X' \in C^{op}$
  $g : Y \to Y' \in C$
- **(identities)** $id_{(X,Y)} = (id_X, id_Y)$
- **(composition)** $(f, g); (f', g') = (f; f', g; g')$ where
  $f; f' \in C^{op}$ or $f'; f \in C$
  $g; g' \in C$

Now we define what we map to in **Set**:

- **(objects)** $(X, Y) \in (C \times C^{op}) \mapsto hom_C(X, Y) \in$ **Set**
- **(morphisms)** for **Set**, a morphism is all functions $hom_C(X, Y) \to hom_C(X', Y')$
  Given (f, g) where
  $f : X \to X' \in C^{op}$ i.e. $f : X' \to X \in C$
  $g : Y \to Y' \in C$
  We can define a morphism (functions) $hom_C(X, Y) \to hom_C(X', Y') \in$ **Set** as follows:
  if $h \in hom_C(X, Y)$
  then $f; h; g \in hom_C(X', Y')$
  so $h \mapsto f; h; g$

In short:



Now we check it is a valid functor:

- **(identities)** $(id_X, id_Y) \mapsto h \mapsto id_X; h; id_Y = id_{hom_C(X,Y)}$

- **(composition)** Given $h \in hom_C(X, Y)$

  We want to show that the following composition is valid:

  $hom_C(X, Y) \xrightarrow{A} hom_C(X', Y') \xrightarrow{B} hom_C(X'', Y'')$

  $h' = f; h; g$ and $f'; h'; g' = f'; f; h; g; g'$

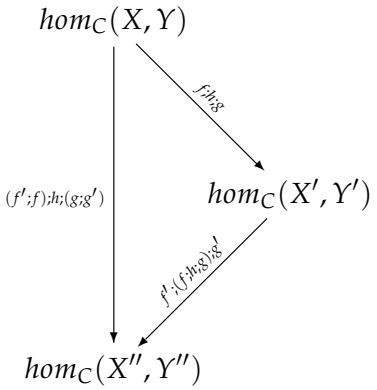
  we know that composition $A; B : hom_C(X, Y) \to hom_C(X'', Y'') = f'; f; h; g; g'$

  We now prove composition is maintained by the functor:

  Given $(f, g) = (X' \to X'', Y \to Y')$ and $(f', g') = (X \to X', Y' \to Y'')$

  $F((f; g)\ ;\ (f'; g'))$
  $= F(f'; f\ ,\ g; g')$
  $= (f'; f; h; g; g')$ where $h \in hom_C(X, Y)$
  $= f'; (f; h; g); g'$
  $= f'; F(f, g); g'$
  $= F(f, g); F(f'; g)$

  i.e.



0.6 Homework exercises

0.6.1 Define a functor from C to a preorder

Every category can be turned into a preordered class.

Given category C, write a preorder $X \leq Y$ when $\exists X \to Y$

(Part 1) Prove $\leq$ is a preorder on obC:

A preorder is a reflexive and transitive binary relation $\leq$.

- reflexive because for any $X \in C$, id_X is a morphism $X \to X$, so $X \leq X$ exists and is reflexive.

- transitive because if $X \leq Y \leq Z$ then we can pick a map $f : X \to Y$ and a map $g : Y \to Z$ so $f;g$ is a map $X \to Z$. This morphism is mapped to $X \leq Y$, which means the composition is transitive.

For instance, in **Set**, $A \leq B$ if B is non-empty or A is empty.

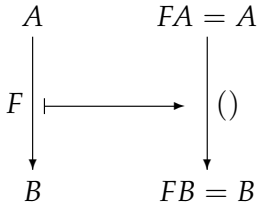
(functions are left-total and single-valued)

(Part 2) Define a functor $C \to \widehat{C} = (\widehat{ob C}, \leq)$

$F : C \to \widehat{ob C}, \leq$

- $ob\widehat{C} \stackrel{\text{def}}{=} obC$ i.e. $\forall X \in C \,.\, X \in C \mapsto X \in \widehat{C}$
- Given $X \xrightarrow{f} Y \in C$, we know $\widehat{C}(X, Y) \stackrel{\text{def}}{=} 1$ i.e. $\forall X \leq Y \in C \,.\, f \mapsto ()$

therefore:



We know this is valid because:

- $id_X \stackrel{\text{def}}{=} ()$ i.e. $\forall x, id_X \in C \,.\, id_X \mapsto ()$
- $X \xrightarrow{()} Y \xrightarrow{()} Z \stackrel{\text{def}}{=} ()$ i.e. composition is maintained trivially,

Additionally, since this definition describes a category that rises from a preordered class (section 3.4.2), we can argue from the resulting category that C is a preordered class.

0.6.2 Full and faithful

Given $F_{X,Y} : hom_C(X, Y) \to hom_D(F(X), F(Y))$

- A **full** mapping is one which is **surjective**. (left-total)

  So $F_{X,Y}$ is full if it is surjective.

  i.e. every morphism in $F\,C$ is mapped to by at least one morphism of C.

  symbolically: $\forall g \in F\,C \,.\, \exists f \in C \,.\, g = F(f)$

- A **faithful** mapping is one which is **injective**.

  *note: injective and single-valued are not the same.

  i.e. $F_{X,Y}$ is faithful when for every $X, Y \in C$, $f : X \to Y$ and $g : X \to Y$, $F\,f = F\,g$ implies $f = g$

  in short: $\forall f, g \in C \,.\, F(f) = F(g) \Rightarrow f = g$

- $F_{X,Y}$ is **fully faithful** if it is both full and faithful.

Note that these definitions only care about morphisms being mapped.

Is the functor defined in 3.6.2 faithful and/or full?

We check:

- The functor is full because:

  $\forall g : FX \to FY \in D$

  $\exists f : X \to Y \in C$

  such that $F_{X,Y}(f) = g$

  since g impies $X \leq Y$ and $X \leq Y$ implies there is an $f : X \to Y$, then $f \mapsto g$, so every f is mapped.

- The functor is not faithful for C if $C(X, Y) = \{f, g, h\}$. This is because all morphisms will be mapped to the same thing, the empty tuple.

  So, $f, g, h \mapsto ()$, meaning, $F_{X,Y}$ is not single valued. However, is this the case for every C?

  To prove this is not the case, we can define a category D where this is an injective mapping. e.g. Given $hom_D(X, Y) = k, k \mapsto ()$. This is injective.

  Therefore, F is **not faithful in general**.

0.6.3 Prove category $(C(X, X), id_X, ;_{X,X})$ is a monoid

Given category C with object X, the monoid $M = (C(X, X), id_X, \underset{X,X}{;})$ is defined as follows:

- id_X is the identity element for morphisms
- $;_{X,X}$ (composition) is an associative binary operator by definition (it comes from a category, where–by definition–composition is associative)