Equivalence of categories

Siddharth Bhat

`##harmless` **Category Theory in Context**

Sun 20, June 2021

A motivating example

- Consider $F : \mathsf{Set}_\partial \to \mathsf{Set}_*$.

A motivating example

- Consider $F : \mathsf{Set}_\partial \to \mathsf{Set}_*$.
- Send a set X to the set $(X \cup \{X\}, X)$ where we view X as the new basepoint.

A motivating example

- Consider $F : \mathsf{Set}_\partial \to \mathsf{Set}_*$.
- Send a set X to the set $(X \cup \{X\}, X)$ where we view X as the new basepoint. (Why X? It's a nice trick, as math lacks `gensym`).

A motivating example

- Consider $F : \mathsf{Set}_\partial \to \mathsf{Set}_*$.
- Send a set X to the set $(X \cup \{X\}, X)$ where we view X as the new basepoint. (Why X? It's a nice trick, as math lacks `gensym`).
- Send a partial function $f : S \xrightarrow{\partial} T$ to a total function which maps to the basepoint where undefined.

A motivating example

- Consider $F : \mathrm{Set}_\partial \to \mathrm{Set}_*$.
- Send a set X to the set $(X \cup \{X\}, X)$ where we view X as the new basepoint. (Why X? It's a nice trick, as math lacks `gensym`).
- Send a partial function $f : S \xrightarrow{\partial} T$ to a total function which maps to the basepoint where undefined.

$$F : \mathrm{Set}_\partial \to \mathrm{Set}_* \quad Ff : FX \to FY = X \cup \{X\} \to Y \cup \{Y\}$$

$$Ff \equiv \lambda x. \begin{cases} f(x) & f \text{ is defined at } x \\ Y & \text{otherwise} \end{cases}$$

- Want an inverse to $F : \mathrm{Set}_\partial \to \mathrm{Set}_*$ (called $G : \mathrm{Set}_* \to \mathrm{Set}_\partial$).

A motivating example

- Consider $F : \mathsf{Set}_\partial \to \mathsf{Set}_*$.
- Send a set X to the set $(X \cup \{X\}, X)$ where we view X as the new basepoint. (Why X? It's a nice trick, as math lacks `gensym`).
- Send a partial function $f : S \xrightarrow{\partial} T$ to a total function which maps to the basepoint where undefined.

$$F : \mathsf{Set}_\partial \to \mathsf{Set}_* \quad Ff : FX \to FY = X \cup \{X\} \to Y \cup \{Y\}$$

$$Ff \equiv \lambda x. \begin{cases} f(x) & f \text{ is defined at } x \\ Y & \text{otherwise} \end{cases}$$

- Want an inverse to $F : \mathsf{Set}_\partial \to \mathsf{Set}_*$ (called $G : \mathsf{Set}_* \to \mathsf{Set}_\partial$). We should probably forget the basepoint (since we added it ourselves).

A motivating example

- Consider $F : \mathrm{Set}_\partial \to \mathrm{Set}_*$.
- Send a set X to the set $(X \cup \{X\}, X)$ where we view X as the new basepoint. (Why X? It's a nice trick, as math lacks `gensym`).
- Send a partial function $f : S \xrightarrow{\partial} T$ to a total function which maps to the basepoint where undefined.

$$F : \mathrm{Set}_\partial \to \mathrm{Set}_* \quad Ff : FX \to FY = X \cup \{X\} \to Y \cup \{Y\}$$

$$Ff \equiv \lambda x. \begin{cases} f(x) & f \text{ is defined at } x \\ Y & \text{otherwise} \end{cases}$$

- Want an inverse to $F : \mathrm{Set}_\partial \to \mathrm{Set}_*$ (called $G : \mathrm{Set}_* \to \mathrm{Set}_\partial$). We should probably forget the basepoint (since we added it ourselves).
- Send a set $(X, x) \in \mathrm{Set}_*$ to $X - \{x\} \in \mathrm{Set}_\partial$.

A motivating example

- Consider $F : \mathrm{Set}_\partial \to \mathrm{Set}_*$.
- Send a set X to the set $(X \cup \{X\}, X)$ where we view X as the new basepoint. (Why X? It's a nice trick, as math lacks `gensym`).
- Send a partial function $f : S \xrightarrow{\partial} T$ to a total function which maps to the basepoint where undefined.

$$F : \mathrm{Set}_\partial \to \mathrm{Set}_* \quad Ff : FX \to FY = X \cup \{X\} \to Y \cup \{Y\}$$

$$Ff \equiv \lambda x. \begin{cases} f(x) & f \text{ is defined at } x \\ Y & \text{otherwise} \end{cases}$$

- Want an inverse to $F : \mathrm{Set}_\partial \to \mathrm{Set}_*$ (called $G : \mathrm{Set}_* \to \mathrm{Set}_\partial$). We should probably forget the basepoint (since we added it ourselves).
- Send a set $(X, x) \in \mathrm{Set}_*$ to $X - \{x\} \in \mathrm{Set}_\partial$. Define $G(X, x) \equiv X - \{x\}$.

A motivating example

- Consider $F : \mathrm{Set}_\partial \to \mathrm{Set}_*$.
- Send a set X to the set $(X \cup \{X\}, X)$ where we view X as the new basepoint. (Why X? It's a nice trick, as math lacks `gensym`).
- Send a partial function $f : S \xrightarrow{\partial} T$ to a total function which maps to the basepoint where undefined.

$$F : \mathrm{Set}_\partial \to \mathrm{Set}_* \quad Ff : FX \to FY = X \cup \{X\} \to Y \cup \{Y\}$$

$$Ff \equiv \lambda x. \begin{cases} f(x) & f \text{ is defined at } x \\ Y & \text{otherwise} \end{cases}$$

- Want an inverse to $F : \mathrm{Set}_\partial \to \mathrm{Set}_*$ (called $G : \mathrm{Set}_* \to \mathrm{Set}_\partial$). We should probably forget the basepoint (since we added it ourselves).
- Send a set $(X, x) \in \mathrm{Set}_*$ to $X - \{x\} \in \mathrm{Set}_\partial$. Define $G(X, x) \equiv X - \{x\}$.
-

$$f : (S, s) \to (T, t) \quad Gf : GS \to GT = (S - \{s\}) \xrightarrow{\partial} (T - \{t\})$$

$$G(f) \equiv \lambda x. \begin{cases} \texttt{undefined} & \mathrm{f(x) = t} \\ f(x) & \text{otherwise} \end{cases}$$

Are we all good?

$$F : \mathrm{Set}_\partial \to \mathrm{Set}_*; \quad f : X \xrightarrow{\partial} Y; \quad Ff : FX \to FY = X \cup \{X\} \to Y \cup \{Y\}$$

$$Ff \equiv \lambda x. \begin{cases} f(x) & f \text{ is defined at } x \\ Y & \text{otherwise} \end{cases}$$

$$G : \mathrm{Set}_* \to \mathrm{Set}_\partial; \quad f : (S, s) \to (T, t); \quad Gf : (S - \{s\}) \xrightarrow{\partial} (T - \{t\})$$

$$G(f) \equiv \lambda x. \begin{cases} \texttt{undefined} & \mathrm{f(x) = t} \\ f(x) & \text{otherwise} \end{cases}$$

Are we all good?

$$F : \mathrm{Set}_\partial \to \mathrm{Set}_*; \quad f : X \xrightarrow{\partial} Y; \quad Ff : FX \to FY = X \cup \{X\} \to Y \cup \{Y\}$$

$$Ff \equiv \lambda x. \begin{cases} f(x) & f \text{ is defined at } x \\ Y & \text{otherwise} \end{cases}$$

$$G : \mathrm{Set}_* \to \mathrm{Set}_\partial; \quad f : (S, s) \to (T, t); \quad Gf : (S - \{s\}) \xrightarrow{\partial} (T - \{t\})$$

$$G(f) \equiv \lambda x. \begin{cases} \texttt{undefined} & \mathrm{f(x) = t} \\ f(x) & \text{otherwise} \end{cases}$$

- Let $X \equiv (\{1, 2\}, 1) \in \mathrm{Set}_*$; $Y \equiv (\{a, b\}, a) \in \mathrm{Set}_*$; $f : X \to Y \in Hom_*(X, Y)$; $f(1) \equiv a$, $f(2) \equiv b$.

Are we all good?

$$F : \mathrm{Set}_\partial \to \mathrm{Set}_*; \quad f : X \xrightarrow{\partial} Y; \quad Ff : FX \to FY = X \cup \{X\} \to Y \cup \{Y\}$$

$$Ff \equiv \lambda x. \begin{cases} f(x) & f \text{ is defined at } x \\ Y & \text{otherwise} \end{cases}$$

$$G : \mathrm{Set}_* \to \mathrm{Set}_\partial; \quad f : (S, s) \to (T, t); \quad Gf : (S - \{s\}) \xrightarrow{\partial} (T - \{t\})$$

$$G(f) \equiv \lambda x. \begin{cases} \texttt{undefined} & \text{f(x) = t} \\ f(x) & \text{otherwise} \end{cases}$$

- Let $X \equiv (\{1, 2\}, 1) \in \mathrm{Set}_*$; $Y \equiv (\{a, b\}, a) \in \mathrm{Set}_*$; $f : X \to Y \in Hom_*(X, Y)$; $f(1) \equiv a$, $f(2) \equiv b$. (Basepoint-preserving).

Are we all good?

$$F : \mathrm{Set}_\partial \to \mathrm{Set}_*; \quad f : X \xrightarrow{\partial} Y; \quad Ff : FX \to FY = X \cup \{X\} \to Y \cup \{Y\}$$

$$Ff \equiv \lambda x. \begin{cases} f(x) & f \text{ is defined at } x \\ Y & \text{otherwise} \end{cases}$$

$$G : \mathrm{Set}_* \to \mathrm{Set}_\partial; \quad f : (S, s) \to (T, t); \quad Gf : (S - \{s\}) \xrightarrow{\partial} (T - \{t\})$$

$$G(f) \equiv \lambda x. \begin{cases} \texttt{undefined} & \mathrm{f(x) = t} \\ f(x) & \text{otherwise} \end{cases}$$

- Let $X \equiv (\{1, 2\}, 1) \in \mathrm{Set}_*$; $Y \equiv (\{a, b\}, a) \in \mathrm{Set}_*$; $f : X \to Y \in Hom_*(X, Y)$; $f(1) \equiv a$, $f(2) \equiv b$. (Basepoint-preserving).
- $GX = \{1\}$, $GY = \{a\}$, $Gf \equiv 1 \mapsto a$.

Are we all good?

$$F : \mathrm{Set}_\partial \to \mathrm{Set}_*; \quad f : X \xrightarrow{\partial} Y; \quad Ff : FX \to FY = X \cup \{X\} \to Y \cup \{Y\}$$

$$Ff \equiv \lambda x. \begin{cases} f(x) & f \text{ is defined at } x \\ Y & \text{otherwise} \end{cases}$$

$$G : \mathrm{Set}_* \to \mathrm{Set}_\partial; \quad f : (S, s) \to (T, t); \quad Gf : (S - \{s\}) \xrightarrow{\partial} (T - \{t\})$$

$$G(f) \equiv \lambda x. \begin{cases} \texttt{undefined} & \mathrm{f(x) = t} \\ f(x) & \text{otherwise} \end{cases}$$

- Let $X \equiv (\{1, 2\}, 1) \in \mathrm{Set}_*$; $Y \equiv (\{a, b\}, a) \in \mathrm{Set}_*$; $f : X \to Y \in Hom_*(X, Y)$; $f(1) \equiv a$, $f(2) \equiv b$. (Basepoint-preserving).
- $GX = \{1\}$, $GY = \{a\}$, $Gf \equiv 1 \mapsto a$. $FGX = (\{1, \{1\}\}, \{1\})$, $FGY = (\{a, \{a\}\}, \{a\})$, $FGf \equiv 1 \mapsto a, \{1\} \mapsto \{a\}$.

Are we all good?

$$F : \mathrm{Set}_\partial \to \mathrm{Set}_*; \quad f : X \xrightarrow{\partial} Y; \quad Ff : FX \to FY = X \cup \{X\} \to Y \cup \{Y\}$$

$$Ff \equiv \lambda x. \begin{cases} f(x) & f \text{ is defined at } x \\ Y & \text{otherwise} \end{cases}$$

$$G : \mathrm{Set}_* \to \mathrm{Set}_\partial; \quad f : (S, s) \to (T, t); \quad Gf : (S - \{s\}) \xrightarrow{\partial} (T - \{t\})$$

$$G(f) \equiv \lambda x. \begin{cases} \texttt{undefined} & \mathrm{f(x) = t} \\ f(x) & \text{otherwise} \end{cases}$$

- Let $X \equiv (\{1, 2\}, 1) \in \mathrm{Set}_*$; $Y \equiv (\{a, b\}, a) \in \mathrm{Set}_*$; $f : X \to Y \in Hom_*(X, Y)$; $f(1) \equiv a$, $f(2) \equiv b$. (Basepoint-preserving).
- $GX = \{1\}$, $GY = \{a\}$, $Gf \equiv 1 \mapsto a$. $FGX = (\{1, \{1\}\}, \{1\})$, $FGY = (\{a, \{a\}\}, \{a\})$, $FGf \equiv 1 \mapsto a, \{1\} \mapsto \{a\}$.
- Define $\eta : Id_{\mathrm{Set}_*} \to GF$; $\eta((X, x)) \equiv (X, \{X\})$;

Are we all good?

$$F : \mathrm{Set}_\partial \to \mathrm{Set}_*; \quad f : X \xrightarrow{\partial} Y; \quad Ff : FX \to FY = X \cup \{X\} \to Y \cup \{Y\}$$

$$Ff \equiv \lambda x. \begin{cases} f(x) & f \text{ is defined at } x \\ Y & \text{otherwise} \end{cases}$$

$$G : \mathrm{Set}_* \to \mathrm{Set}_\partial; \quad f : (S, s) \to (T, t); \quad Gf : (S - \{s\}) \xrightarrow{\partial} (T - \{t\})$$

$$G(f) \equiv \lambda x. \begin{cases} \texttt{undefined} & \mathrm{f(x) = t} \\ f(x) & \text{otherwise} \end{cases}$$

- Let $X \equiv (\{1, 2\}, 1) \in \mathrm{Set}_*$; $Y \equiv (\{a, b\}, a) \in \mathrm{Set}_*$; $f : X \to Y \in Hom_*(X, Y)$; $f(1) \equiv a$, $f(2) \equiv b$. (Basepoint-preserving).
- $GX = \{1\}$, $GY = \{a\}$, $Gf \equiv 1 \mapsto a$. $FGX = (\{1, \{1\}\}, \{1\})$, $FGY = (\{a, \{a\}\}, \{a\})$, $FGf \equiv 1 \mapsto a, \{1\} \mapsto \{a\}$.
- Define $\eta : Id_{\mathrm{Set}_*} \to GF$; $\eta((X, x)) \equiv (X, \{X\})$; Remap the basepoint.
- Let $S \equiv \{c, d\} \in \mathrm{Set}_\partial$; $T \equiv \{3, 4\} \in \mathrm{Set}_\partial$; $g \in Hom_\partial(S, T)$; $g(c) \equiv 3$, $g(d) \not\equiv \_$
- $FS \equiv \{1, 2, \{1, 2\}_*\}$; $FT \equiv \{3, 4, \{3, 4\}_*\}$; $Fg \equiv c \mapsto 3, d \mapsto \{3, 4\}, \{1, 2\} \mapsto \{3, 4\}$.
- $GFS \equiv \{1, 2\}$; $GFT \equiv \{3, 4\}$; $GFg \equiv c \mapsto 3, d \not\mapsto$.

Are we all good?

$$F : \text{Set}_\partial \to \text{Set}_*; \quad f : X \xrightarrow{\partial} Y; \quad Ff : FX \to FY = X \cup \{X\} \to Y \cup \{Y\}$$

$$Ff \equiv \lambda x. \begin{cases} f(x) & f \text{ is defined at } x \\ Y & \text{otherwise} \end{cases}$$

$$G : \text{Set}_* \to \text{Set}_\partial; \quad f : (S, s) \to (T, t); \quad Gf : (S - \{s\}) \xrightarrow{\partial} (T - \{t\})$$

$$G(f) \equiv \lambda x. \begin{cases} \texttt{undefined} & \text{f(x) = t} \\ f(x) & \text{otherwise} \end{cases}$$

- Let $X \equiv (\{1,2\}, 1) \in \text{Set}_*$; $Y \equiv (\{a,b\}, a) \in \text{Set}_*$; $f : X \to Y \in Hom_*(X, Y)$; $f(1) \equiv a$, $f(2) \equiv b$. (Basepoint-preserving).
- $GX = \{1\}$, $GY = \{a\}$, $Gf \equiv 1 \mapsto a$. $FGX = (\{1, \{1\}\}, \{1\})$, $FGY = (\{a, \{a\}\}, \{a\})$, $FGf \equiv 1 \mapsto a, \{1\} \mapsto \{a\}$.
- Define $\eta : Id_{\text{Set}_*} \to GF$; $\eta((X, x)) \equiv (X, \{X\})$; Remap the basepoint.
- Let $S \equiv \{c, d\} \in \text{Set}_\partial$; $T \equiv \{3, 4\} \in \text{Set}_\partial$; $g \in Hom_\partial(S, T)$; $g(c) \equiv 3$, $g(d) \not\equiv \_$
- $FS \equiv \{1, 2, \{1,2\}_*\}$; $FT \equiv \{3, 4, \{3,4\}_*\}$; $Fg \equiv c \mapsto 3, d \mapsto \{3,4\}, \{1,2\} \mapsto \{3,4\}$.
- $GFS \equiv \{1, 2\}$; $GFT \equiv \{3, 4\}$; $GFg \equiv c \mapsto 3, d \not\mapsto$.
- In general, may have needed a $\epsilon : FG \to Id_{\text{Set}_\partial}$

Equivalence of categories

- Equality is too strong. Equivalence ought to be just right.

Equivalence of categories

- Equality is too strong. Equivalence ought to be just right.
- Two categories, C, D are said to be equivalent iff we have functors $F : C \to D$ and $G : D \to C$ such that there exist natural isomorphisms: $\eta : 1 \simeq GF$ and $\epsilon : FG \simeq 1$.

Equivalence of categories

- Equality is too strong. Equivalence ought to be just right.
- Two categories, C, D are said to be equivalent iff we have functors $F : C \to D$ and $G : D \to C$ such that there exist natural isomorphisms: $\eta : 1 \simeq GF$ and $\epsilon : FG \simeq 1$.
- Recall: natural isomorphisms is a natural transformation $\eta : F \Rightarrow G$ where each component $\eta_x : Fx \to Gx$ is an isomorphism.

Equivalence of categories

- Equality is too strong. Equivalence ought to be just right.
- Two categories, C, D are said to be equivalent iff we have functors $F : C \to D$ and $G : D \to C$ such that there exist natural isomorphisms: $\eta : 1 \simeq GF$ and $\epsilon : FG \simeq 1$.
- Recall: natural isomorphisms is a natural transformation $\eta : F \Rightarrow G$ where each component $\eta_x : Fx \to Gx$ is an isomorphism.

Equivalence of categories: Characterization

- A functor is full, faithful, and essentially surjective iff an equivalence of categories.

Equivalence of categories: Characterization

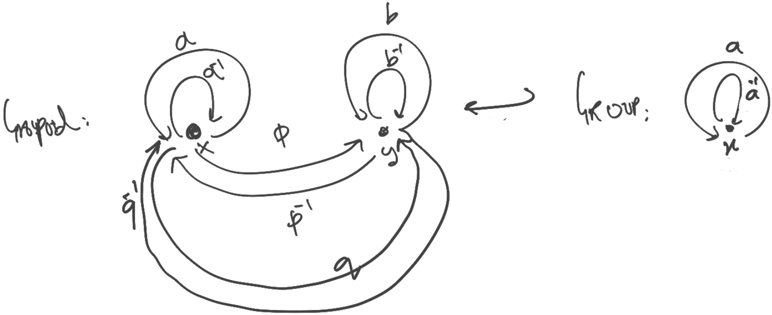
- A functor is full, faithful, and essentially surjective iff an equivalence of categories.
- Full: If the map $Hom(x, y) \to Hom(Fx, Fy)$ is surjective for all $x, y \in C$.
- Faithful: If the map $Hom(x, y) \to Hom(Fx, Fy)$ is injective for all $x, y \in C$.
- Essentially surjective: For any object $d \in D$, there is some object $y \in Im(F)$ such that y is isomorphic to d ($y \simeq d$).

Equivalence of categories: Characterization

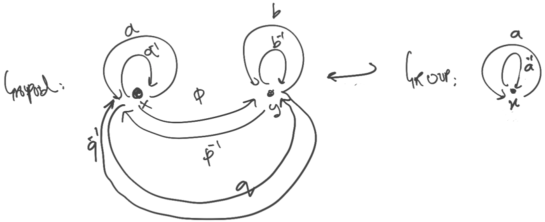
- A functor is full, faithful, and essentially surjective iff an equivalence of categories.
- Full: If the map $Hom(x, y) \to Hom(Fx, Fy)$ is surjective for all $x, y \in C$.
- Faithful: If the map $Hom(x, y) \to Hom(Fx, Fy)$ is injective for all $x, y \in C$.
- Essentially surjective: For any object $d \in D$, there is some object $y \in Im(F)$ such that y is isomorphic to d ($y \simeq d$). functor F is surjective "upto isomorphism".

Equivalence of categories: Characterization

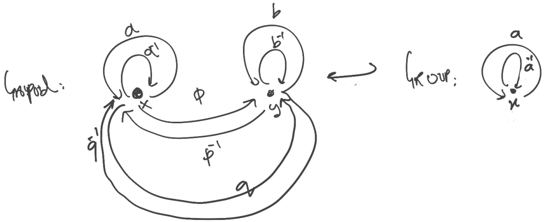
- A functor is full, faithful, and essentially surjective iff an equivalence of categories.
- Full: If the map $Hom(x, y) \to Hom(Fx, Fy)$ is surjective for all $x, y \in C$.
- Faithful: If the map $Hom(x, y) \to Hom(Fx, Fy)$ is injective for all $x, y \in C$.
- Essentially surjective: For any object $d \in D$, there is some object $y \in Im(F)$ such that y is isomorphic to d ($y \simeq d$). functor F is surjective "upto isomorphism".



Equivalence of categories is Unintuitive (to me)

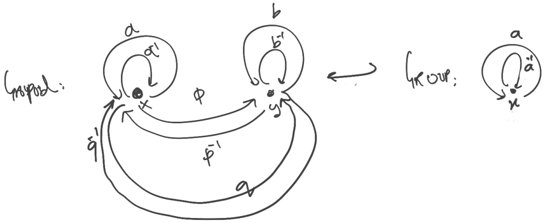


Equivalence of categories is Unintuitive (to me)



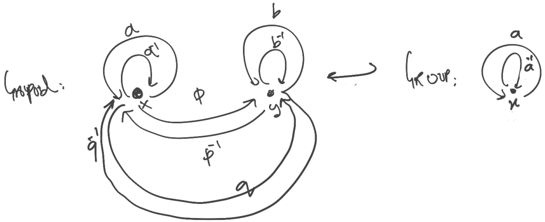
- Let O be a connected groupoid, and let $x \in O$ be some object of the groupoid. Exract out a single object of the groupoid, by considering the subcategory consisting of only the object $x \in O$. Label this subcategory G.

Equivalence of categories is Unintuitive (to me)



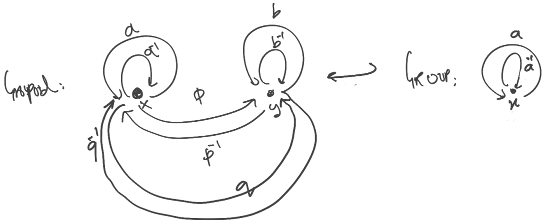
- Let O be a connected groupoid, and let $x \in O$ be some object of the groupoid. Exract out a single object of the groupoid, by considering the subcategory consisting of only the object $x \in O$. Label this subcategory G.
- The embedding functor $F : G \to O$ is full and faithful since its image contains a single object (x) where it preserves all arrows.

Equivalence of categories is Unintuitive (to me)



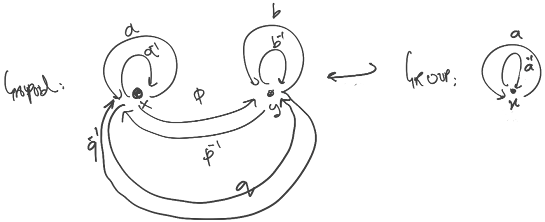
- Let O be a connected groupoid, and let $x \in O$ be some object of the groupoid. Exract out a single object of the groupoid, by considering the subcategory consisting of only the object $x \in O$. Label this subcategory G.
- The embedding functor $F : G \to O$ is full and faithful since its image contains a single object (x) where it preserves all arrows.
- Also see that it is essentially surjective.

Equivalence of categories is Unintuitive (to me)



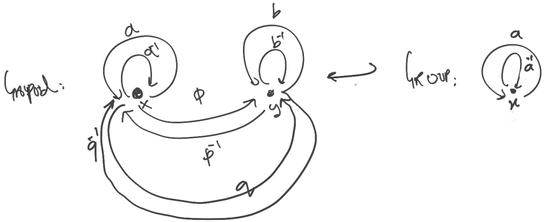
- Let O be a connected groupoid, and let $x \in O$ be some object of the groupoid. Exract out a single object of the groupoid, by considering the subcategory consisting of only the object $x \in O$. Label this subcategory G.
- The embedding functor $F : G \to O$ is full and faithful since its image contains a single object (x) where it preserves all arrows.
- Also see that it is essentially surjective. For any other $y \in O$, we have a path $x \xrightarrow{p} y$ (as O is connected).

Equivalence of categories is Unintuitive (to me)



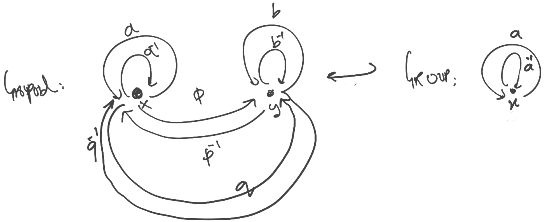
- Let O be a connected groupoid, and let $x \in O$ be some object of the groupoid. Exract out a single object of the groupoid, by considering the subcategory consisting of only the object $x \in O$. Label this subcategory G.
- The embedding functor $F : G \to O$ is full and faithful since its image contains a single object (x) where it preserves all arrows.
- Also see that it is essentially surjective. For any other $y \in O$, we have a path $x \xrightarrow{p} y$ (as O is connected).

Equivalence of categories is Unintuitive (to me)



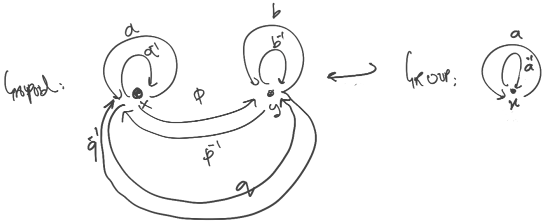
- Let O be a connected groupoid, and let $x \in O$ be some object of the groupoid. Exract out a single object of the groupoid, by considering the subcategory consisting of only the object $x \in O$. Label this subcategory G.
- The embedding functor $F : G \to O$ is full and faithful since its image contains a single object (x) where it preserves all arrows.
- Also see that it is essentially surjective. For any other $y \in O$, we have a path $x \xrightarrow{p} y$ (as O is connected). since it is a groupoid, all morphisms are isos, and thus $y \simeq x$.

Equivalence of categories is Unintuitive (to me)



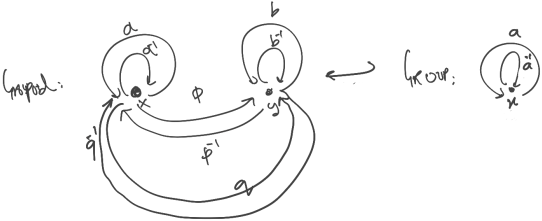
- Let O be a connected groupoid, and let $x \in O$ be some object of the groupoid. Exract out a single object of the groupoid, by considering the subcategory consisting of only the object $x \in O$. Label this subcategory G.
- The embedding functor $F : G \to O$ is full and faithful since its image contains a single object (x) where it preserves all arrows.
- Also see that it is essentially surjective. For any other $y \in O$, we have a path $x \xrightarrow{p} y$ (as O is connected). since it is a groupoid, all morphisms are isos, and thus $y \simeq x$.
- Soo, this is an equivalence of categories?!

Equivalence of categories is Unintuitive (to me)



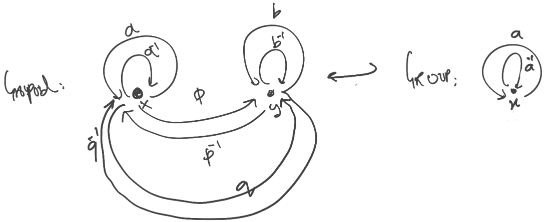
- Let O be a connected groupoid, and let $x \in O$ be some object of the groupoid. Exract out a single object of the groupoid, by considering the subcategory consisting of only the object $x \in O$. Label this subcategory G.
- The embedding functor $F : G \to O$ is full and faithful since its image contains a single object (x) where it preserves all arrows.
- Also see that it is essentially surjective. For any other $y \in O$, we have a path $x \xrightarrow{p} y$ (as O is connected). since it is a groupoid, all morphisms are isos, and thus $y \simeq x$.
- Soo, this is an equivalence of categories?!
- Philosophically, equivalence of categories does not need to preserve size.

Equivalence of categories is Unintuitive (to me)



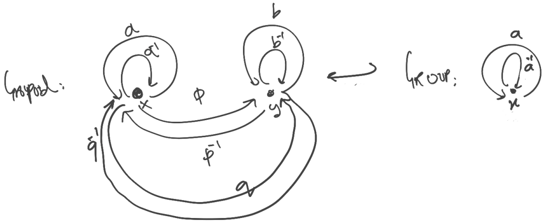
- Let O be a connected groupoid, and let $x \in O$ be some object of the groupoid. Exract out a single object of the groupoid, by considering the subcategory consisting of only the object $x \in O$. Label this subcategory G.
- The embedding functor $F : G \to O$ is full and faithful since its image contains a single object (x) where it preserves all arrows.
- Also see that it is essentially surjective. For any other $y \in O$, we have a path $x \xrightarrow{p} y$ (as O is connected). since it is a groupoid, all morphisms are isos, and thus $y \simeq x$.
- Soo, this is an equivalence of categories?!
- Philosophically, equivalence of categories does not need to preserve size. It only needs to preserve a "copy of information".

Equivalence of categories is Unintuitive (to me)



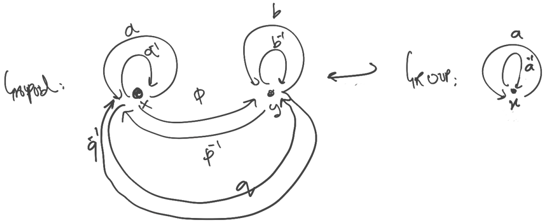
- Let O be a connected groupoid, and let $x \in O$ be some object of the groupoid. Exract out a single object of the groupoid, by considering the subcategory consisting of only the object $x \in O$. Label this subcategory G.
- The embedding functor $F : G \to O$ is full and faithful since its image contains a single object (x) where it preserves all arrows.
- Also see that it is essentially surjective. For any other $y \in O$, we have a path $x \xrightarrow{p} y$ (as O is connected). since it is a groupoid, all morphisms are isos, and thus $y \simeq x$.
- Soo, this is an equivalence of categories?!
- Philosophically, equivalence of categories does not need to preserve size. It only needs to preserve a "copy of information". Connected Groupoid contains many copies of the same information.

Equivalence of categories is Unintuitive (to me)



- Let O be a connected groupoid, and let $x \in O$ be some object of the groupoid. Exract out a single object of the groupoid, by considering the subcategory consisting of only the object $x \in O$. Label this subcategory G.
- The embedding functor $F : G \to O$ is full and faithful since its image contains a single object (x) where it preserves all arrows.
- Also see that it is essentially surjective. For any other $y \in O$, we have a path $x \xrightarrow{p} y$ (as O is connected). since it is a groupoid, all morphisms are isos, and thus $y \simeq x$.
- Soo, this is an equivalence of categories?!
- Philosophically, equivalence of categories does not need to preserve size. It only needs to preserve a "copy of information". Connected Groupoid contains many copies of the same information. It is safe for forget these.

Equivalence of categories is Unintuitive (to me)



- Let O be a connected groupoid, and let $x \in O$ be some object of the groupoid. Exract out a single object of the groupoid, by considering the subcategory consisting of only the object $x \in O$. Label this subcategory G.
- The embedding functor $F : G \to O$ is full and faithful since its image contains a single object (x) where it preserves all arrows.
- Also see that it is essentially surjective. For any other $y \in O$, we have a path $x \xrightarrow{p} y$ (as O is connected). since it is a groupoid, all morphisms are isos, and thus $y \simeq x$.
- Soo, this is an equivalence of categories?!
- Philosophically, equivalence of categories does not need to preserve size. It only needs to preserve a "copy of information". Connected Groupoid contains many copies of the same information. It is safe for forget these.

Skeleta

- A category is *skeletal* iff each isomorphism class has a single object.

Skeleta

- A category is *skeletal* iff each isomorphism class has a single object.
- *Mat*, category of numbers and materices is the skeleton of FinVectBasis, category of finite vector spaces with bases, and morphisms as matrices encoding linear operators relative to the basis.
- Can build $sk(C)$ (Skeleton of C). Crush each isomorphism class into a single object.
- The inclusion $sk(C) \hookrightarrow C$ defines an equivalence of categories.