Databases: Categories, functors, universal construction

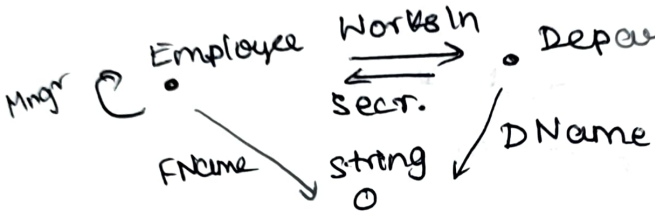
- Integrating data from disparate sources is needed now.

→ A database is a system of interlocking tables.

Employee	FName	W.In	Mngr

Department	DName	Secy

can be represented as



(Hasse diagr. for a database)

*Non-ID columns are arrows here.

(Employee, Dept : ID-columns)

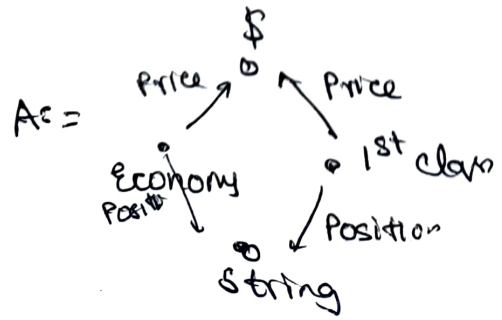
We can add conditions to this schema also. eg: Dept. Secr. WorkIn = Dept

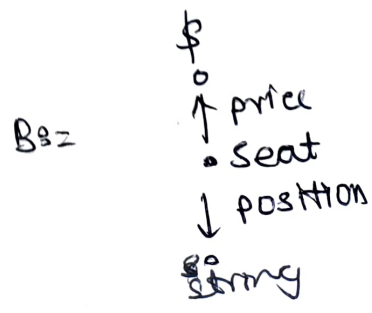
Emp. Mngr. WorIn = Emp. WorIn

We will see that these are categories & and set data is given by a functor $\mathcal{C} \to \mathbf{Set}$

- Category theory provides a mathematical approach for translating between these different organisational forms of tabular

"Data migration"





- A has more detail than B and they can be connected by a functor.
- While moving data from one schema to other, we might fail to satisfy some of its constraints
- So we will discuss a category-method for migrating data

Categories:

- A category $\mathcal{C}$ consists of 4 pieces of data → Objects, → Morphisms, → Identities, → Composition rule

Defn: To specify a category $\mathcal{C}$:

- One specifies a collection $Ob(\mathcal{C})$ (objects)
- $\forall\ c, d \in Ob(\mathcal{C})$ specify a set $\mathcal{C}(c,d)$ called morphisms from c to d
- $\forall\ c \in Ob(\mathcal{C})$ specify a morphism $id_c \in \mathcal{C}(c,c)$
- $\forall\ c, d, e \in Ob(\mathcal{C})$ and $f \in \mathcal{C}(c,d)$ $g \in \mathcal{C}(d,e)$ we must have $(f;g)$ or $(g \circ f) \in \mathcal{C}(c,e)$

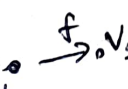
Conditions: a) Unitality: $f: c \to d$ with identities does nothing $id_c; f = f$ $f; id_d = f$

b) Associativity $f: c_0 \to c_1$ $g: c_1 \to c_2$ $h: c_2 \to c_3$

$(f;g);h = f;(g;h)$

Free categories:

- For any quad $G = (V, A, s, t)$, we can define a cat $Free(G)$ whose objects are vertices V and morphisms are paths from $c \to d$. Identity morphism on obj e is trivial. Composition ≅ Concatenation



Obj. $\{v_1, v_2\}$ Hom = $\{id_{v_1}: v_1 \to v_1,\ f: v_1 \to v_2,\ id_{v_2}: v_2 \to v_2\}$

3.9) Unitality: $id_A ; f = f$ (As id_A is a trivial path and concat is just the path itself)

Associativity: $(f;g);h = f;(g;h)$ As concatenation of two paths is irrelevant of order.

3.10) $v_1 \xrightarrow{f_1} v_2 \xrightarrow{f_2} v_3$

$Obj(e) = \{v_1, v_2, v_3\}$

$e( ) = \{id_{v_1}, id_{v_2}, id_{v_3}, f_1, f_2, f_1;f_2\}$

3.12) $n + {}^nC_2$ $\quad$ v_1 $\quad$ $Obj = \{v_1\}$ $\quad e( ) = \{id_{v_1}\}$

$n + \frac{n(n-1)}{2} = \frac{n(n+1)}{2}$ $\quad \phi$ $\quad$ Obj: $\{\}$ $\quad e( ) = \phi$

3.13) $\mathbb{N}$ as free categories:

one vertex, one arrow but infinitely many paths.

Path = $\{z, s, (s;s), (s;s;s) \ldots\}$ length of 0 is z.

- Example of category with 1 object (monoid)

$(M, *, e) \cong (\mathbb{N}, ;, id_0)$ (Associativity & unitality is satisfied)

symmetry? (Only 1 object ~~so more possible~~!)

3.15) Concat of m, n will give $m+n$

Presenting categories with path eqns:

∴ For any graph G, we have a free category on G. But we will add some more props. Add eqns between paths and still get a cat.

- Category after the add^n of equation is called finitely-presented category.

1) $A \xrightarrow{f} B$, $g: A \to C$, $h: B \to D$, $i: C \to D$

2) Same with $f;h = g;i$ (Parallel paths are same)

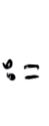
1) has 10 morphisms but 2 has only 9 ($id_A, id_B, id_C, id_D, f, g, h, i, f;h$)

3.16) $id_A, id_B, id_C, id_D, f, g, h, i, f;h, g;i$

3.18)

$C :=$  $s;s=z \quad s;s;s;s=z$

Set of morphism is $\{z, s\}$

3.19) $D :=$  $s;s;s;s = s;s$

Morph: $\{z, s, s;s, s;s;s\}$

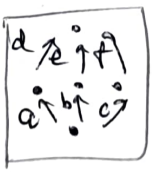
3.23 Preorders and free categories = 2 ends of a spectrum.

"A preorder is a category where "every" two parallel arrows are the same"

→ $(P, \leq)$ be a preorder. P category is defined from it. $Ob(P) = P$ and has exactly 1 morphism from $p \to q$ if $p \leq q$ and no morphism if $p \not\leq q$.

(Satisfies unitality, composition, associativity

preorder



with $a;d = b;e = c;f$

But free category need not need this edt

Exc: 3.21) G_1: 

G_2:  $f = f;id_V$

G_3:  $f;h = g;i$

G_4: 

NONE

Preorder from category: Given $\mathcal{C}$, we define $(C, \leq)$. $C = \text{Obj}(\mathcal{C})$ and $c_1 \leq c_2$ iff $\mathcal{C}(c_1, c_2) \neq \phi$ (And if exist it ~~could~~ can be ~~[illegible]~~ infinite)

[Inclusion map]

3.22) ☺

Considering a preorder as a category is rightadjoint to turning a category into a preorder by preorder reflection!! (Inc-map)

இங்கிவனை யாண்பெற என்ன தவம் செய்தேனோ!

☺

3.23) Both preorders and free cats are denoted by graph but denoted opposite ends of spectrum.

Any category representation $\in$ [Free cat, Preorder]

Free cat ↓ No eqns, $\frac{n(n+1)}{2}$ morphs

Preorder ↓ All eqns, $n + f(n)$ morphs, $f(n) << {}^nC_2$

<u>Important categories</u> (<u>sets</u>):

Defn: Category of sets, "Set"

(i) Ob(Set) = collection of sets

(ii) If $S, T \in$ Ob(Set), then $\text{Set}(S, T) = \{f : S \to T \mid f \text{ is a fn}\}$

(iii) $\forall S \in$ Ob(Set), $id_S : S \to S$ de $id_S(s) = s$, $s \in S$.

(iv) $f : S \to T$ and $g : T \to U$, $f ; g : S \to U$, $f ; g := g(f(s))$

Some cat. related to it is FinSet where objects are finite

Eg: $\underline{2} = \{1,2\}$ $\underline{3} = \{1,2,3\}$ No. of morphisms $= 3^2 = 9$

3.26) "Categories are V-categories for a special choice of V."

Study of Set (cat.) ⇒ categories
Study of Cost (cat.) ⇒ Lawvere metric spaces.

Top: Topological ; Grph: Graphs ...
Cat: Categories

3.27) Let $\mathcal{C}$ be a category. Its opp. $\mathcal{C}^{op}$ is defined by $Obj(\mathcal{C}^{op}) := Obj(\mathcal{C})$
$\mathcal{C}^{op}(d,c) := \mathcal{C}(c,d)$

Isomorphisms in a category:

- In category, there is an idea of interchanging objects. Eg: {●, □} with {0,2} can be interchanged
- Notion of isomorphism generalises it.

Defn: An isomorphism is a morphism $f: A \to B$ s.t $\exists\ g: B \to A$ satisfying $f;g = id_A$ $g;f = id_B$. A, B are isomorphic objects. f and g are inverses of each other

→ For any finite set $A \in FinSet$, its cardinality is the number $n \in \mathbb{N}$ s.t $\exists$ an isomorphism $A \cong \underline{n}$

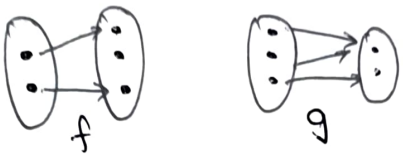
3.30)
1) $f^{-1}: 3 \to A$ $f(1) = a$ $f(2) = b$ $f^{-1}(3) = c$
2) No. of isomorphisms $A \to \underline{3}$ are $3!$

3.31) For a categ. $\mathcal{C}$ $\forall\ c \in \mathcal{C}$ id_c is an isomorphism w. its inverse being id_c itself. $id_c ; id_c = id_c$

3.32] A monoid in which every morphism is an isomorphism is called group.

1] $\circlearrowright_{2}^{S}$ $sos = id_2$ is a group.

3.33) Let G be a graph and Free(G) be the corresponding free category. Only isomorphisms in Free(G) are identity. As it does not satisfy any equations, and the path is unidirectional. (lengths add when composed)

3.34] 

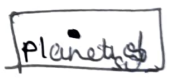
$f;g = id_2$ but $g;f \neq id_3$

Therefore it is not identity but f and g form retraction ($r \circ f = id_A$)

3.3] Functors, natural transformations, and databases:

- Set can be understood as a table with only one column.

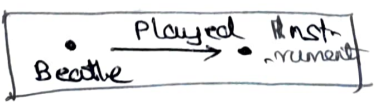
Planets of Sol
Mercury
Venus
Earth
⋮

This can be depicted as 

- A function $f: A \to B$ can almost be depicted as a two-column table. 

Beatle	Played
George	Guit
John	Rh..
Paul	Drums
Ringo	Key.

Instrument
Drums
Keyboard
⋮
(∞)



Database can be presented as a category that could be transformed to category of set

Functors: Let $\mathcal{C}$ and $\mathcal{D}$ be categories.

The functor $F: \mathcal{C} \to \mathcal{D}$ is defined as

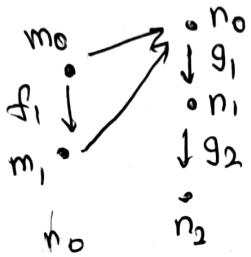
(i) $\forall\, c \in Obj(\mathcal{C})$, $F(c) \in Obj(\mathcal{D})$

(ii) $\forall\, f: c_1 \to c_2$ in $\mathcal{C}$, $F(f): F(c_1) \to F(c_2)$ in $\mathcal{D}$

- Conditions: (i) $\forall\, c \in Obj(\mathcal{C})$, $F(id_c) = id_{F(c)}$

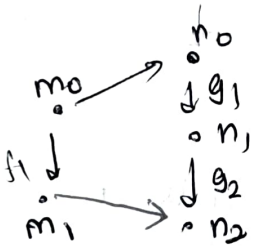
(ii) $\forall c_1, c_2, c_3 \in Obj(\mathcal{C})$, $f, g, h \in \mathcal{C}(c_1, c_2)$ $F(f;g) = F(f); F(g)$ holds true

Eg:



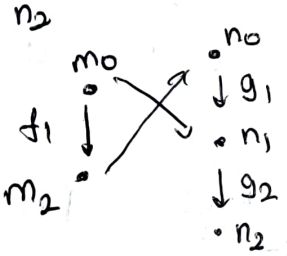
Every path to trivial path id_{n_0}

$F(f_1) = id_{n_0}$



$F(f_1) = g_1; g_2$

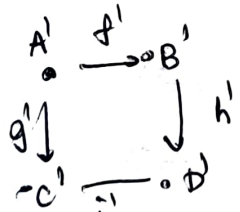
3-38)



(NOT a functor)

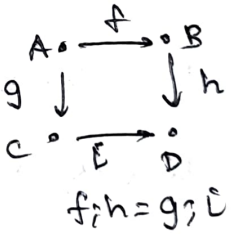
- Functions must be preserved.

3-38)



No eqn

$\mathcal{F}$



$f;h = g;E$

$\mathcal{C}$

One functor $F: \mathcal{F} \to \mathcal{C}$ exists given $F(A') = A$ $F(B') = B$ $F(C') = C$ and $F(D') = D$

All ten paths are forced to go to their assigned path

3.39) $\underline{8} \to \underline{8}$ $f';g' \to f;h$ $g';i' \to f;h$

3.40) $\mathcal{C}$: [•] $\mathcal{D}$: $\bullet \overset{f'}{\underset{g'}{\rightrightarrows}} \bullet$ Two functors.

$F(f) = f'$ or g'

3.41] There are functors from Comm → Free but nothing matches $A \to A', B \to B', C \to C'$ and $D \to D'$ because if matched $F(g;h) = F(f;k)$ which fails.

∴ Comm has addⁿ constraints and therefore has more things to get matched.

3.42] Functors btw preorders are nothing but monotone maps.

$(\mathbb{N}, \leq)$ considered as a category $\mathbb{N}$. Functor $F: \mathbb{N} \to \mathbb{N}$ sends each object $n \in \mathbb{N}$ and morphisms means $F(m) \to F(n)$ iff $m \to n$ Similar to $F(m) \leq F(n)$ iff $m \leq n$ ☺

∴ A functor $F: \mathbb{N} \to \mathbb{N}$ is same as a monotone map.

3.43) "Cat": Categories of categories. Construct this category.

1] $Id_{\mathcal{C}}: \mathcal{C} \to \mathcal{C}$ functor that just maps each obj / morph to itself.

2] Given: $F: \mathcal{C} \to \mathcal{D}$ and $G: \mathcal{D} \to \mathcal{E}$

$(F;G)(obj(\mathcal{C})) = F(G(F(obj(\mathcal{C}))) \in \mathcal{E}$

$(F;G)(f) =$

$c_1, c_2 \in \mathcal{C}$ · $F(f)$ will be a fn in $\mathcal{D}$ btw d_1 & d_2

Database instances as set-valued functions;

- Let $\mathcal{C}$ be a category, A functor $F: \mathcal{C} \to Set$ is known as set valued functor on $\mathcal{C}$. Much of database theory (dealing with it) can be cast in this light.

→ Any instance of database is given by a Set-valued functor $I: \mathcal{C} \to Set$. The only additional detail is any white node ○ : string should be mapped with a set of strings.

Defn: Let $\mathcal{C}$ be a schema, a finitely-presented category. A $\mathcal{C}$ instance is a functor $I: \mathcal{C} \to Set$

Exc 3.45) 1 category with one object called 1, and 1 isomorphism $F: 1 \to Set$ one can extract a set $F(1)$. You can define

$F(id_1) = id_S$

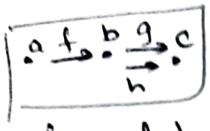
∴ Any set of 1 can be defined with a set-valued functors.

eg: $\mathcal{C} :=$  $(S \circlearrowright \; s \circ s = s)$

- $Z =$ {citizens} $S : Z \to Z$ (President)
  $s \circ s = s$ (President of president = president)
- $Z = \mathbb{N}$ $S: \mathbb{N} \to \mathbb{N}$ just 0 converges.
- $Z =$ {Arithmetic expressions} → S = result.
- $Z = \mathbb{N}_{\geq 2}$ S: n to its smallest prime factor.

3.48) 1. 

$Z = \{$ set of $\circlearrowright \; \circlearrowright$ (Two cycle loops)

2. 

$f;g = f;h$

this is a section

If f is a section $g=h$ so it is not

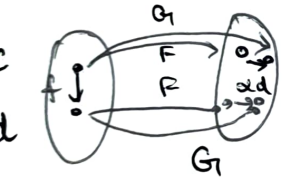
Natural Transformations

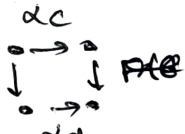
. If $\mathcal{C}$ is a schema, then there are many database instances on it.

Defn: Let $\mathcal{C}$ and $\mathcal{D}$ be categories, $F, G : \mathcal{C} \to \mathcal{D}$ be functors. To specify a natural transformation $\alpha : F \Rightarrow G$

(1) $\forall\, c \in \mathcal{C}$, specify a morphism $\alpha_c : F(c) \to G(c)$ called c-component of α

Cdt: $\forall\, f : c \to d$ in $\mathcal{C}$, $F(f); \alpha_d = \alpha_c; G(f)$



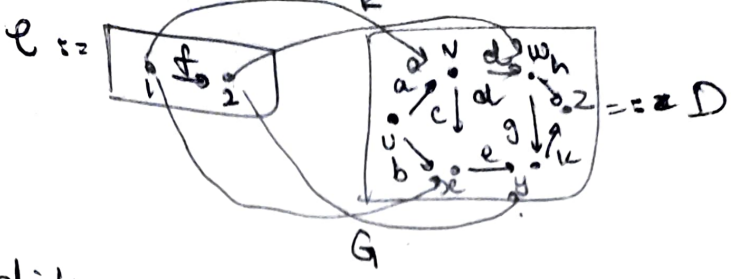


comm. square

$\alpha : F \to G$ is called natural isomorphism if each comp. of α_c is an isomorphism in $\mathcal{D}$

Defn: A "diagram" D in $\mathcal{C}$ is a functor $D : \mathcal{J} \to \mathcal{C}$ from any category $\mathcal{J}$ (indexing category) of diagram D.

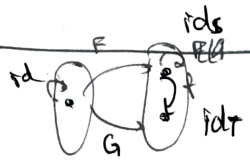
D commutes if $D(f) = D(f')$ holds for every parallel pair of morphisms $f, f' : a \to b$



Condition for natural trans is

$$c;e = d;g$$

•Functors are ways of looking category C as lying inside D and natural transformation α is the way of relating these two views using the morphisms in D



GOLDEN

353) We said the functor $1 \to Set$ can be identified as a set. If A, B are functors $A, B : 1 \to Set$. The fn between A and B is just a Natural transformation

Defn: Let C, D are categories. We denote by D^C the category whose objects are functors $F : C \to D$ and morphisms $D^C(F, G)$ are natural transformations $\alpha : F \to G$. D^C is called functor category.

355) 1) How to compose natural trans.

Let $\alpha_1 : F \to G$ $\alpha_2 : G \to H$ $\alpha_1 ; \alpha_2$ is given as morphism between $F(C)$ and $(F ; G)(C)$

2) Id_C $id : F \to F$ is defined $Id_C(C) = e$

As diagram

$$\begin{array}{ccccc} F(C) & \xrightarrow{\alpha_C} & G(C) & \xrightarrow{\beta_C} & H(C) \\ {\scriptstyle F(f)}\downarrow & & {\scriptstyle G(f)}\downarrow & & \downarrow{\scriptstyle H(f)} \\ F(C') & \xrightarrow[\alpha_{C'}]{} & G(C') & \xrightarrow[\beta_{C'}]{} & H(C') \end{array}$$

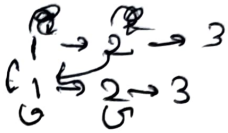
(comp:

$$\begin{array}{ccc} F(C) & \xrightarrow{\alpha_C ; \beta_C} & H(C) \\ {\scriptstyle F(f)}\downarrow & & \downarrow{\scriptstyle H(f)} \\ F(C') & \xrightarrow[\alpha_{C'} ; \beta_{C'}]{} & H(C') \end{array}$$

3.56) With new language (category of functors)

$\mathbf{Set}^{\mathbf{1}}$ ($\mathbf{1} \to \mathbf{Set}$) $\cong$ Set

functors



3.57) Let $\mathbb{N}$ denote the category associated with the preorder $(\mathbb{N}, \leq)$. A functor $F: \mathbb{N} \to \mathbb{N}$ with a non-decreasing sequence $(F_0, F_1, \ldots F_n)$. G is another functor. What is a $\alpha: F \Rightarrow G$? The fn exists iff the ordering exists meaning $F_n \leq G_n$. ∴ The category $\mathbb{N}^{\mathbb{N}}$ is itself a preorder

Preserves the fn

3.58) $\mathcal{C}$ is an arbitrary category and $\mathbb{P}$ be a preorder thought of as category.

1) For any two functors $F, G: \mathcal{C} \to \mathbb{P}$, there is atmost one natural transformation.

$\alpha: F \to G$ $\quad$ $\beta: F \to G$ $\quad$ To prove: $\alpha = \beta$

As it is a preorder all parallel fns must be same $\Rightarrow \alpha = \beta$ ☺

2) Not true map it to id.

359) Category "PrO": Obj: Preorders
Morp: Monotones

Category "Bool-Cat": Obj: Bool cats
Morp: Bool functors.

→ These two are equivalent.

Preorder ≅ Bool category

∃ functors $F: PrO \to Bool\text{-}Cat$ and $G: Bool\text{-}Cat \to PrO$

s.t $F;G \cong id_{PrO}$ $G;F \cong id_{BoolCat}$

Category of instance on a schema:

Defn: Suppose $\mathcal{C}$ is a database schema and $I, J : \mathcal{C} \to Set$ are database instances. An instance homomorphism between them is a natural transformation $\alpha: I \to J$. $\mathcal{C}\text{-Inst} := Set^{\mathcal{C}}$

Eg: Category of graphs as a functor category-

$Gr := z$ Arrows ⇉ (source, target) • Vertex

Gr-instance = $I: Gr \to Set$

Arrow	source (1)	Rest (2)
a		
b	1	3
c	2	3

Vertex
1
2
3

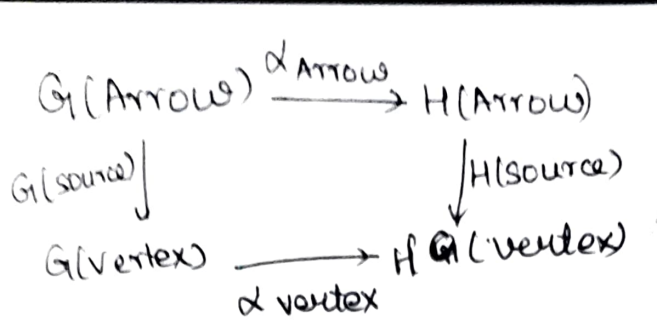
$I(Arrow) = \{a, b, c, d, e\}$ $I(Vertex) = \{1, 2, 3, 4\}$

* Objects in Gr-Inst are graphs.
Morphisms in Gr-Inst are graph homo-morphisms.

$G, H \in Gr \to Set$ are functors $\forall G, H \in Gr\text{-}Inst$
A morphism $G \to H$ is a natural transformation

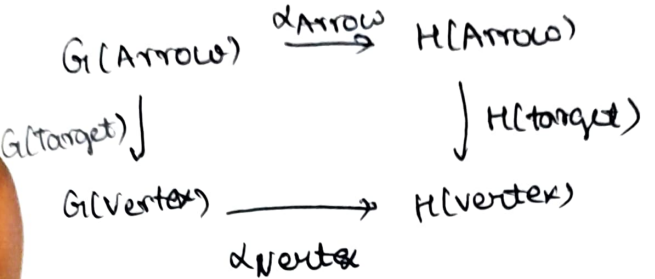
$\alpha_{vertex}: G(vertex) \to H(vertex)$

$\alpha_{arrow}: G(arrow) \to H(arrow)$

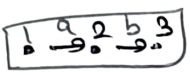


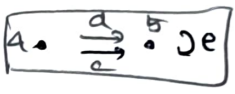
AND

Start with an arrow in G. Use α to the arrow and take a source or take a source and find its transform



(+)

363) G: 

H:= 

Arrow	Src	Trg
a	1	2
b	2	3

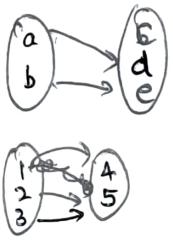
Vert
1
2
3

Arrow	Src	Trg
c	4	5
d	4	5
e	5	5

Vert
4
5

364) α_{Arrow} = ?

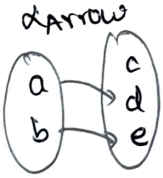
α_{vertex}



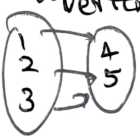
$\alpha_{Arrow}(b) = e$

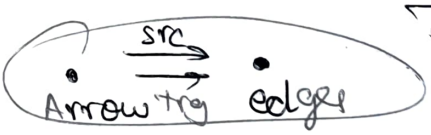
α_A

α_{Arrow}



α_{Vertex}



Gr is just 

That's all

You map it to a set in set category

Using this you can map it to more than 1 set and have natural trans. satisfying top (+) odt. This is the simplest form of migration.