1/26/16

Mathematics Between Categories

Recall that given categories C & D, a <u>functor</u> $F: C \to D$ is a map sending objects $c \in C$ to objects $F(c) \in D$, morphisms $f: c \to c'$ in C to morphisms $F(f): F(c) \to F(c')$ in D, preserving composition $F(f' \circ f) = F(f') \circ F(f)$ & identities $F(1_c) = 1_{F(c)}$.

$$
\begin{tikzcd}
\mathrm{Ring} \arrow[dr, "U_3"'] & & \mathrm{Vect}_k \arrow[dl, "U_4"] & & \\
& \mathrm{AbGrp} \arrow[d, "U_2"'] & & & \\
& \mathrm{Grp} \arrow[dr, "U_1"'] & & \mathrm{Top} \arrow[dl, "U_5"] & \mathrm{Set}^2 \arrow[dll, "U_6"] \\
& & \mathrm{Set} & &
\end{tikzcd}
$$

($U :=$ underlying)

There are many "forgetful functors" going from categories of "fancy" mathematical gadgets to categories of less fancy ones, forgetting some extra properties, structure, or stuff.

$\boxed{\text{Ex}}$ $U_1: \mathrm{Grp} \to \mathrm{Set}$ sends any group G to its <u>underlying</u> set, and any homomorphism $f: G \to G'$ to its <u>underlying</u> function.

$\boxed{\text{Ex}}$ Given categories C & D, there's a category $C \times D$, where objects are ordered pairs (c,d) with $c \in C$, $d \in D$, and morphisms are ordered pairs (f,g) with f a morphism in C, g a morphism in D: given $f: c \to c'$ in C and $g: d \to d'$ in D then $(f,g): (c,d) \to (c',d')$. We define $(f',g') \circ (f,g) = (f' \circ f, g' \circ g)$ and $1_{(c,d)} = (1_c, 1_d)$.

In fact $C \times D$ is the product of the objects $C, D \in \mathrm{Cat}$, which is the category with

- (small) categories as objects
- functors as morphisms

Among other things this means we have projections

$$\begin{tikzcd} & C \times D \arrow[dl, "p"'] \arrow[dr, "q"] & \\ C & & D \end{tikzcd}$$

Set is a large category, but we can still define $\text{Set}^2 = \text{Set} \times \text{Set}$, with pairs of sets as objects.

In the chart, let $U_6 : \text{Set}^2 \longrightarrow \text{Set}$ be the projection onto the first component.
$$(S,T) \longmapsto S$$

Functions can be nice in two ways: one-to-one & onto

Functors can be nice in three ways:

Def A functor $F: C \to D$ is <u>faithful</u> if for any $c, c' \in C$
$F: \hom(c,c') \longrightarrow \hom(F(c), F(c'))$ is one-to-one.

Def A functor $F: C \to D$ is <u>full</u> if for any $c, c' \in C$
$F: \hom(c,c') \longrightarrow \hom(F(c), F(c'))$ is onto.

Def A functor $F: C \to D$ is <u>essentially surjective</u> if for any $d \in D$, there exists $c \in C$ such that $F(c) \cong d$ meaning there exists an isomorphism $g: F(c) \to d$ in D.

Ex Compare $\text{FinVect}_{\mathbb{R}}$ (finite dim'l vector spaces) to this category C, with
- $\{0\}, \mathbb{R}, \mathbb{R}^2, \mathbb{R}^3, \ldots$ as objects
- all linear maps between these as morphisms

$F: C \longrightarrow \text{FinVect}$
$\mathbb{R}^n \longrightarrow \mathbb{R}^n$ and similarly for morphisms: $f: \mathbb{R}^n \to \mathbb{R}^m \longmapsto f: \mathbb{R}^n \to \mathbb{R}^m$

This is faithful & full, not surjective on objects, but essentially surjective.

Later we'll define "equivalent" categories & see that if $F: C \to D$ is faithful, full, & essentially surjective then C & D are equivalent.

We say:

$\boxed{\text{Def}}$ • A functor $U: C \to D$ <u>forgets nothing</u> if it's faithful, full, & ess. surj.

• U <u>forgets (at most) properties</u> if it's faithful & full.

• U <u>forgets (at most) structure</u> if it's faithful.

• In general we say U <u>forgets (at most) stuff</u>.

$\boxed{\text{Ex}}$ $U_1: \mathrm{Grp} \to \mathrm{Set}$ forgets (at most) structure.

It's faithful: given $f, f': G \to G'$ in Grp, $U_1(f) = U_1(f') \Rightarrow f = f'$.

It's not full: there are usually functions $f: U_1(G) \to U_1(G')$ that don't come from group homomorphism, e.g.: $f(gh) \neq f(g)f(h)$ or $f(1) \neq 1$.

$\boxed{\text{Ex}}$ $U_2: \mathrm{AbGrp} \to \mathrm{Grp}$ forgets (at most) properties: the comm. law is forgotten.

This is faithful and also full: if you have any group homomorphism $f: U_2(A) \to U_2(A')$ then $U(f')$ for some homomorphism of abelian groups $f': A \to A'$

But it's not ess. surjective: if G is nonabelian, $G \not\cong U_2(A)$ for some $A \in \mathrm{AbGrp}$.

$\boxed{\text{Ex}}$ $U_6: \mathrm{Set}^2 \to \mathrm{Set}$

forgets stuff: $U_6(S, S') = S$ It forgets the second set in the pair.

Technically, its not faithful:

we can have 2 different morphisms $(f,g), (f,g'): (S,S') \to (T,T')$ with $U_6(f,g) = f = U_6(f,g')$

In our chart, every forgetful functor $U: C \to D$ has a "left adjoint" $F: D \to C$ which "freely creates" the stuff, structure or properties that U forgets.

$\boxed{\text{Ex}}$ $F_1: \mathrm{Set} \to \mathrm{Grp}$ takes a set S and forms the free group on S, $F_1(S)$.

$F_2: \mathrm{Grp} \to \mathrm{AbGrp}$ <u>abelianizes</u> any group G, forming

$$F_2(G) = \frac{G}{\langle xyx^{-1}y^{-1} \rangle}$$

normal subgp gen. by these elements

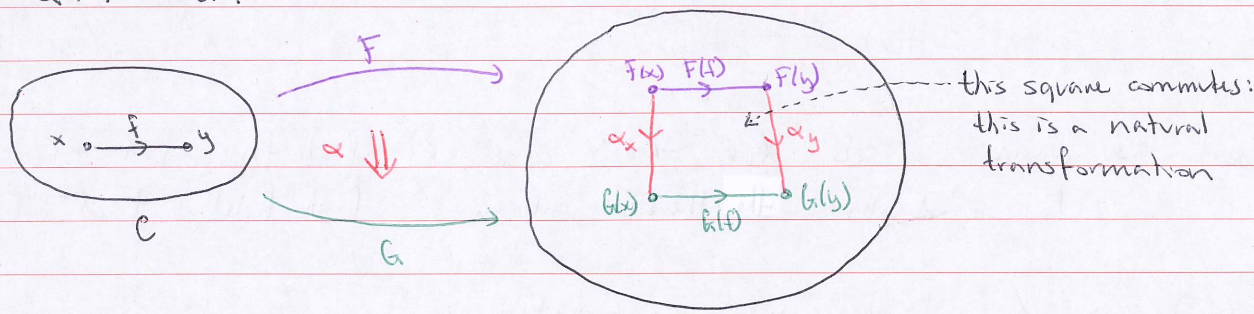
$\boxed{\text{Ex}}$ $L_6: \mathrm{Set} \to \mathrm{Set}^2$

$S \mapsto (S, \emptyset)$

1/28/16

To define adjoint functors (and many other things) we need...
Natural Transformations

Given 2 functors $F, G : C \to D$ we can define a natural transformation $\alpha : F \Rightarrow G$!



$\boxed{\text{Def}}$ Given functors $F, G : C \to D$ a transformation $\alpha : F \Rightarrow G$ is a function sending each object $x \in C$ to a morphism $\alpha_x : F(x) \to G(x)$.
We say α is a natural transformation if for each morphism $f : x \to y$ in C this square commutes:

$$\begin{tikzcd} F(x) \arrow[r, "F(f)"] \arrow[d, "\alpha_x"'] & F(y) \arrow[d, "\alpha_y"] \\ G(x) \arrow[r, "G(f)"] & G(y) \end{tikzcd}$$

$\boxed{\text{Prop}}$ Given categories C & D there's a category, the functor category D^C, with:

- objects being functors $F : C \to D$
- morphisms being natural transformations $\alpha : F \Rightarrow G$

In D^C we compose $\alpha : F \Rightarrow G$, $\beta : G \Rightarrow H$ to get $\beta \circ \alpha : F \Rightarrow H$ as follows:
$(\beta \circ \alpha)_x : F(x) \to H(x)$ $\forall x \in C$ is given by $\beta_x \circ \alpha_x$.
In D^C the identity $1_F : F \Rightarrow F$, $(1_F)_x : F(x) \to F(x)$ $x \in C$ is given by $1_{F(x)}$.

Pf:

Check that the composite $\beta \circ \alpha$ is natural:
given $f : x \to y$ in C, want this to commute:

$$\begin{tikzcd} F(x) \arrow[r, "F(f)"] \arrow[d, "(\beta\circ\alpha)_x"'] & F(y) \arrow[d, "(\beta\circ\alpha)_y"] \\ H(x) \arrow[r, "H(f)"'] & H(y) \end{tikzcd}$$

(continued)

Pf: (continued)

$$\begin{tikzcd}
F(x) \arrow[r, "F(f)"] \arrow[d, "\alpha_x"] \arrow[dd, bend right=60, "(\beta\circ\alpha)_x"'] & F(y) \arrow[d, "\alpha_y"'] \arrow[dd, bend left=60, "(\beta\circ\alpha)_y"] \\
G(x) \arrow[r, "G(f)"] \arrow[d, "\beta_x"] & G(y) \arrow[d, "\beta_y"'] \\
H(x) \arrow[r, "H(f)"] & H(y)
\end{tikzcd}$$

α natural $\Rightarrow$ this commutes

β natural $\Rightarrow$ this commutes

$\blacksquare$

So just as given 2 sets X, Y there's a set Y^X of all fns $f: X \to Y$
2 categories X, Y there's a category Y^X of all functors $F: X \to Y$

Given 2 sets X & Y, they have a product:

$$X \times Y = \{(x,y) : x \in X, y \in Y\}$$

Notice $X \times Y \neq Y \times X$

but $X \times Y \cong Y \times X$

and there's a specific "good" isomorphism

$$\alpha_{X,Y}: X \times Y \longrightarrow Y \times X$$
$$(x,y) \longmapsto (y,x)$$

It's "good" because it's natural, in the sense we just defined.

There are 2 functors from Set^2 to Set,

$F: (X,Y) \longmapsto X \times Y$

$G: (X,Y) \longmapsto Y \times X$

and α is a natural transformation from F to G.

In fact it's a "natural isomorphism":

$\boxed{\text{Def}}$ If $F, G: C \longrightarrow D$ are functors and $\alpha: F \Rightarrow G$ is a nat. tran., we say α is a <u>natural isomorphism</u> if $\alpha_x: F(x) \longrightarrow G(x)$ is an isomorphism $\forall x \in C$.

$\boxed{\text{Prop}}$ $\alpha: F \Rightarrow G$ is a natural isomorphism iff it has an inverse $\alpha^{-1}: G \Rightarrow F$ in D^C.

Pf:

Key idea: $(\alpha^{-1})_x = (\alpha_x)^{-1}$ $\blacksquare$

Prop Suppose C is a category with binary products: any pair of objects $x, y \in C$ has a product. Then we can choose, for any pair $x, y \in C$, a specific product:

$$\begin{tikzcd} & x \times y \arrow[dl, "p_{x,y}"'] \arrow[dr, "q_{x,y}"] & \\ x & & y \end{tikzcd}$$

and then there is a functor $\times : C^2 \longrightarrow C$
$$(x,y) \longmapsto x \times y$$

In fact there are 2 functors:

$F : C^2 \longrightarrow C$ (this is the functor $\times$)
$$(x,y) \longmapsto x \times y$$
$G : C^2 \longrightarrow C$
$$(x,y) \longmapsto y \times x$$
and these are naturally isomorphic.

We say "products are commutative up to natural isomorphism."

Also, products are associative up to natural isomorphism:

$$\alpha_{x,y,z} : (x \times y) \times z \xrightarrow{\sim} x \times (y \times z) \qquad \begin{tikzcd} C^3 \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "\Downarrow\alpha"] & C \end{tikzcd}$$

(Just keep using universal property of product.)

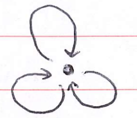
Def A cartesian category is a category with binary products and a terminal object. (I.e. it's a category where any finite set of objects has a product — a finite products category).

One can show that in a cartesian category we have natural isomorphisms
$$l_x : 1 \times x \xrightarrow{\sim} x$$
$$r_x : x \times 1 \xrightarrow{\sim} x$$

All this works similarly in a cat. w/ finite coproducts:
$$B_{x,y} : x + y \xrightarrow{\sim} y + x$$
$$\alpha_{x,y,z} : (x+y)+z \xrightarrow{\sim} x + (y+z)$$
$$l_x : 0 + x \xrightarrow{\sim} x$$
$$r_x : x + 0 \xrightarrow{\sim} x$$

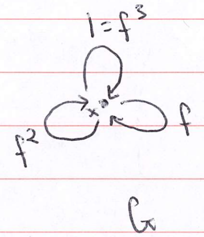
In the case $C = \text{FinSet}$ (finite sets & functions) these give familiar laws of arithmetic: $\mathbb{N}$ is the set of isomorphism classes of objects in FinSet.

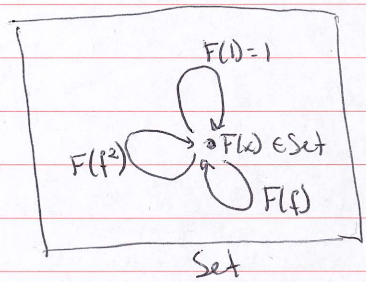
Another example: A group is a category G with one object and with all morphisms invertible:



$\mathbb{Z}/3$

What's a functor $F: G \longrightarrow \text{Set}$?





F picks out a set $X = F(x)$ and for each group element f it picks out a function $F(f): X \longrightarrow X$ s.t. $F(ff') = F(f)F(f')$ & $F(1) = 1_X$.

So: X is a set acted by the group G, or a G-set.

So: a functor $F: G \rightarrow \text{Set}$ is a G-Set. What's a natural transformation between 2 such functors?