Elements of Category Theory

Robin Cockett

Department of Computer Science
University of Calgary
Alberta, Canada

robin@cpsc.ucalgary.ca

Estonia, Feb. 2010

FUNCTORS

A **functor** is a map of categories $F : \mathbb{X} \longrightarrow \mathbb{Y}$ which consists of a map F_0 of the objects and a map F_1 of the maps (we shall drop these subscripts) such that

- $\partial_0(F(f)) = F(\partial_0(f))$ and $\partial_1(F(f)) = F(\partial_1(f))$:

$$\frac{X \xrightarrow{f} Y}{F(X) \xrightarrow[F(f)]{} F(Y)}$$

- $F(1_A) = 1_{F(A)}$, identity maps are preserved.
- $F(fg) = F(f)F(g)$, composition is preserved.

Every category has an identity functor.
Composition of functors is associative. Thus:

Lemma

Categories and functors form a category Cat.

EXAMPLES OF Set FUNCTORS

- The product (with A) functor

$$\_ \times A : \mathsf{Set} \longrightarrow \mathsf{Set}; \quad \begin{array}{c} X \\ {\scriptstyle f}\big\downarrow \\ Y \end{array} \quad \mapsto \quad \begin{array}{c} X \times A \\ \big\downarrow{\scriptstyle f \times 1} \\ Y \times A \end{array} \qquad \begin{array}{c} (x, a) \\ \big\downarrow \\ (f(x), a) \end{array}$$

- The exponential functor:

$$A \Rightarrow \_ : \mathsf{Set} \longrightarrow \mathsf{Set}; \quad \begin{array}{c} X \\ {\scriptstyle f}\big\downarrow \\ Y \end{array} \quad \mapsto \quad \begin{array}{c} A \Rightarrow X \\ \big\downarrow{\scriptstyle A \Rightarrow f} \\ A \Rightarrow Y \end{array} \qquad \begin{array}{c} h \\ \big\downarrow \\ hf \end{array}$$

EXAMPLES OF Set FUNCTORS

- List on A (`data L(A) = Nil | Cons A L(A)`)

$$\mathsf{L} : \mathsf{Set} \longrightarrow \mathsf{Set};\quad \begin{array}{c} X \\ f\big\downarrow \\ Y \end{array} \quad\mapsto\quad \begin{array}{c} \mathsf{L}(X) \\ \big\downarrow \mathsf{L}(f) \\ \mathsf{L}(Y) \end{array} \qquad \begin{array}{c} [x_1, x_2, \ldots] \\ \big\downarrow \\ [f(x_1), f(x_2), \ldots] \end{array}$$

- Trees on A (`data T(A) = Lf A | Node T(A) T(A)`):

$$\mathsf{T} : \mathsf{Set} \longrightarrow \mathsf{Set};\quad \begin{array}{c} X \\ f\big\downarrow \\ Y \end{array} \quad\mapsto\quad \begin{array}{c} \mathsf{T}(X) \\ \big\downarrow \mathsf{T}(f) \\ \mathsf{T}(Y) \end{array} \qquad \begin{array}{c} \mathtt{Node}(\mathtt{Lf}\ x_1)(\mathtt{Lf}\ x_2) \\ \big\downarrow \\ \mathtt{Node}(\mathtt{Lf}\ f(x_1))(\mathtt{Lf}\ f(x_2)) \end{array}$$

EXAMPLES OF Set FUNCTORS

- The covariant powerset functor:

$$\mathcal{P} : \mathrm{Set} \longrightarrow \mathrm{Set}; \quad \begin{array}{c} X \\ {\scriptstyle f}\big\downarrow \\ Y \end{array} \quad \mapsto \quad \begin{array}{c} \mathcal{P}(X) \\ \big\downarrow{\scriptstyle \mathcal{P}(f)} \\ \mathcal{P}(Y) \end{array} \qquad \begin{array}{c} X' \subseteq X \\ \big\downarrow \\ f(X') \subseteq Y \end{array}$$

- The contravariant powerset functor:

$$\mathcal{P} : \mathrm{Set}^{\mathrm{op}} \longrightarrow \mathrm{Set}; \quad \begin{array}{c} X \\ {\scriptstyle f}\big\uparrow \\ Y \end{array} \quad \mapsto \quad \begin{array}{c} \mathcal{P}(X) \\ \big\downarrow{\scriptstyle \mathcal{P}(f)} \\ \mathcal{P}(Y) \end{array} \qquad \begin{array}{c} X' \subseteq X \\ \big\downarrow \\ f^{-1}(X') \subseteq Y \end{array}$$

Note: *covariant* functors are functors, *contravariant* functors are functors BUT starting at the dual category.

NATURAL TRANSFORMATIONS

Given two functors $F, G : \mathbb{X} \longrightarrow \mathbb{Y}$ a **(natural) transformation** $\alpha : F \Rightarrow G$ is a family of maps in $\mathbb{Y}$ $\alpha_X : F(X) \longrightarrow G(X)$, indexed by the objects $X \in \mathbb{X}$ such that for every map $f : X \longrightarrow X'$ in $\mathbb{X}$ the following diagram commutes:

$$\begin{array}{ccc} F(X) & \xrightarrow{F(f)} & F(X') \\ {\scriptstyle \alpha_X}\downarrow & & \downarrow{\scriptstyle \alpha_{X'}} \\ G(X) & \xrightarrow[G(f)]{} & G(X') \end{array}$$

This means that $\mathrm{Cat}(\mathbb{X}, \mathbb{Y})$ can be given the structure of a category. In fact, Cat is a Cat-enriched category (a.k.a. a **2-category**).

Lemma

$\mathrm{Cat}(\mathbb{X}, \mathbb{Y})$ *is a category with objects functors and maps natural transformations.*

NATURAL TRANSFORMATION EXAMPLE I

Consider the category:

$$\mathrm{TWO} = \; E \underset{\partial_0}{\overset{\partial_0}{\rightrightarrows}} N$$

A functor $G : \mathrm{TWO} \longrightarrow \mathrm{Set}$ is precisely a directed graph!!

A natural transformation between two functors:

$$\alpha : G_1 \longrightarrow G_2 : \mathrm{TWO} \longrightarrow \mathrm{Set}$$

is precisely a morphism of the directed graphs.

$$\alpha_N G_1(\partial_i)(f) = G_2(\partial_i)(\alpha_E(f)).$$

NATURAL TRANSFORMATION EXAMPLE II

Consider the category $\mathbb{N}^{\mathrm{op}}$:

$$0 \xleftarrow[\partial]{} 1 \xleftarrow[\partial]{} 2 \xleftarrow[\partial]{} \cdots$$

A functor $F : \mathbb{N}^{\mathrm{op}} \longrightarrow \mathsf{Set}$ is a forest. The children of a node $x \in F(n)$ in the forest is given by $\{x' \in F(n+1) | \partial(x') = x\}$.

A natural transformation between two functors

$$\gamma : F_1 \longrightarrow F_2 : \mathbb{N}^{\mathrm{op}} \longrightarrow \mathsf{Set}$$

is precisely a morphism of forests:

$$\gamma_n(F_1(\partial)(x)) = F_2(\partial)(\gamma_{n+1}(x)).$$

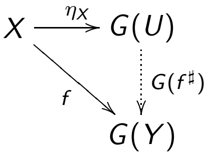
NATURAL TRANSFORMATION ...

If functors define structure ...

Then natural transformation define the (natural) homomorphisms of that structure ...

UNIVERSAL PROPERTY

Let $G : \mathbb{Y} \longrightarrow \mathbb{X}$ be a functor and $X \in \mathbb{X}$, then an object $U \in \mathbb{Y}$ together with a map $\eta : X \longrightarrow G(U)$ is a **universal pair** for the functor G (at the object X) if for any $f : X \longrightarrow G(Y)$ there is a unique $f^\sharp : U \longrightarrow Y$ such that



commutes.

UNIVERSAL PROPERTY – EXAMPLE I

let Graph be the category of directed graphs and Cat the category of categories, let the functor

$$U : \mathrm{Cat} \longrightarrow \mathrm{Graph}$$

be the "underlying functor" which forgets the composition structure of a category.

The map which takes a directed graph and embeds it into the graph underlying the path category as the singleton paths (paths of length one)

$$\eta : G \longrightarrow U(\mathrm{Path}(G)); [n_1 \xrightarrow{a} n_2] \mapsto (n_1, [a], n_2)$$

has the universal property for this "underlying" functor U.

UNIVERSAL PROPERTY – EXAMPLE cont.
Consider a map of directed graphs into the graph underlying a category, $h : G \longrightarrow U(\mathbb{C})$, we can extend it uniquely to a functor from the path category to the category by defining

$$h^\sharp : \mathrm{Path}(G) \longrightarrow \mathbb{C}; (A, [a_1, .., a_n], B) \mapsto h(a_1)..h(a_n) : h(A) \longrightarrow h(B)$$

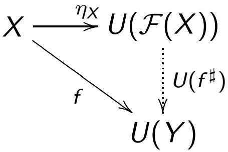
This is uniquely determined by h as where the "generating" arrows go determines where the composite arrows go.

UNIVERSAL PROPERTY – EXAMPLE ...

For those more mathematically inclined:

Consider the category of Group then there is an obvious underlying functor U : Group $\longrightarrow$ Set.

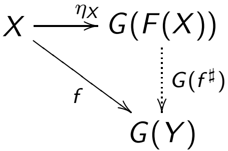
The pair $(\mathcal{F}(X), \eta)$ where $\eta : X \longrightarrow U(\mathcal{F}(X))$ is a universal pair for this underlying functor



The diagram expresses the property of being a "free" group (or more generally "free" algebra).

ADJOINT

Suppose $G : \mathbb{Y} \longrightarrow \mathbb{X}$ has for each $X \in \mathbb{X}$ a universal pair $(F(X), \eta_X)$ so that

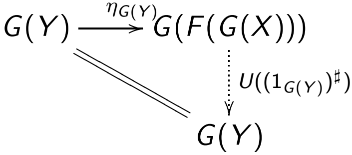


then G is said to be a **right adjoint**.

If $h : X \longrightarrow X' \in \mathbb{X}$ then define $F(h) := (h\eta_{X'})^\sharp$
then F is a functor ...
F is **left adjoint** to G.

$\eta : 1_{\mathbb{X}} \longrightarrow FG$ is a natural transformation ...

ADJOINT Furthermore, $\epsilon_Y := (1_{G(Y)})^\sharp : GF \longrightarrow 1_{\mathbb{Y}}$ is a natural transformation



ADJOINT

This gives the following data (and **adjunction**):

$$(\eta, \epsilon) : F \dashv G : \mathbb{X} \longrightarrow \mathbb{Y}$$

- $F : \mathbb{X} \longrightarrow \mathbb{Y}$ and $G : \mathbb{Y} \longrightarrow \mathbb{X}$ functors
- $\eta : 1_{\mathbb{X}} \longrightarrow FG$ and $\epsilon : GF \longrightarrow 1_{\mathbb{Y}}$ natural transformations
- Triangle equalities:

$$\begin{array}{ccc} G(Y) & \xrightarrow{\eta_{G(Y)}} & G(F(G(Y))) \\ & \searrow{=} & \downarrow{\epsilon_Y} \\ & & G(Y) \end{array} \qquad \begin{array}{ccc} F(X) & \xrightarrow{F(\eta_X)} & F(G(F(X))) \\ & \searrow{=} & \downarrow{\epsilon_{F(X)}} \\ & & F(X) \end{array}$$

This data is purely algebraic and is precisely to ask F be left adjoint to G!

ADJOINT

Another important characterization:

$$\frac{X \xrightarrow{f = g^\flat} G(Y)}{F(X) \xrightarrow[g = f^\sharp]{} Y}$$

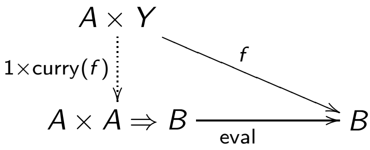
And another important example: **cartesian closed** categories:

$$\frac{A \times X \xrightarrow{f} Y}{X \xrightarrow[\text{curry}(f)]{} A \Rightarrow Y}$$

Semantics of the typed λ-calculus.

ADJOINT

Here is the couniversal property for $A \Rightarrow B$:



$$\text{curry}(f) = y \mapsto \lambda a.f(a, x)$$

MONADS (briefly)

Given an adjunction

$$(\eta, \epsilon) : F \dashv G : \mathbb{X} \longrightarrow \mathbb{Y}$$

consider $T := FG$ we have two transformations:

$$\eta_X : X \longrightarrow T(X) = G(F(X))$$

$$\mu_X : T(T(X)) \longrightarrow T(X) = G(F(G(F(X)))) \xrightarrow{G(\epsilon_{F(X)})} G(F(X))$$

and one can check these satisfy:

$$\begin{array}{ccccc} T(X) & \xrightarrow{\eta_{T(X)}} & T(T(X)) & \xleftarrow{T(\eta_X)} & T(X) \\ & \searrow\!\!= & \downarrow \mu & =\!\!\swarrow & \\ & & T(X) & & \end{array} \qquad \begin{array}{ccc} T(T(T(X))) & \xrightarrow{\mu} & T(T(X)) \\ \downarrow T(\mu) & & \downarrow \mu \\ T(T(X)) & \xrightarrow[\mu]{} & T(X) \end{array}$$

Such a (T, η, μ) is called a **monad**.

ADJUNCTIONS AND MONADS

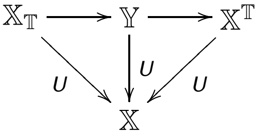
Any adjunction

$$(\eta, \epsilon) : F \dashv G : \mathbb{X} \longrightarrow \mathbb{Y}$$

generates an monad on $\mathbb{X}$ and a comonad on $\mathbb{Y}$.

Furthermore, every monad arises through an adjunction ...

Given a monad $\mathbb{T} = (T, \eta, \mu)$ on a category $\mathbb{X}$ we may construct two categories with underlying right adjoints to $\mathbb{X}$ which generate $\mathbb{T}$:
the **Kleisli** category $\mathbb{X}_{\mathbb{T}}$
and the **Eilenberg-Moore** category $\mathbb{X}^{\mathbb{T}}$
so that any $U : \mathbb{Y} \longrightarrow \mathbb{X}$ a right adjoint which also generates $\mathbb{T}$ sits canonically between these categories:



MONADS AND EFFECTS

Computational *effects* (exceptions, state, continuations, non-determinism ...) can be generated by using the composition of Kleisli categories.

Here is the definition of $\mathbb{X}_{\mathbb{T}}$ (e.g. think list monad):

Objects:

$$X \in \mathbb{X}$$

Maps:

$$\frac{X \xrightarrow{f} T(Y) \in \mathbb{X}}{X \xrightarrow{f} Y \in \mathbb{X}_{\mathbb{T}}}$$

Identities:

$$\frac{X \xrightarrow{\eta_X} T(X) \in \mathbb{X}}{X \xrightarrow{1_X} X \in \mathbb{X}_{\mathbb{T}}}$$

Composition:

$$\frac{X \xrightarrow{f} T(Y) \xrightarrow{T(f)} T^2(Z) \xrightarrow{\mu} T(Z) \in \mathbb{X}}{X \xrightarrow{f} Y \xrightarrow{g} Z \in \mathbb{X}_{\mathbb{T}}}$$

MONADS AND EFFECTS

Incomplete history of monads:

- Named by Mac Lane (Categories for Working Mathematician)
- Known first as "standard construction" (Eilenberg, Moore) also "triple" (Barr)
- Kleisli discovered the "Kleisli category"
- Ernie Manes introduced the form of a monad used in Haskell
- Moggi developed computer Science examples (rediscovered Manes form for monad) and calculi for monads (probably motivated by the partial map classifier – a very well behaved monad),
- Wadler made the connection to list comprehension and uses in programming,
- ... do syntax.

MATHEMATICS CAME FIRST ON THIS ONE ...

FUNCTORIAL CALCULUS

The functorial calculus has turned out to be a useful practical and theoretical tool in programming language semantics and implementation ...

Everyone should know it!!

Although very important this is not the focus of these talks!

INITIAL AND FINAL OBJECTS

An **initial object** in a category $\mathbb{X}$ is an object which has exactly one map to every object (including itself) in the category.

Denote the initial object by 0 and the unique map as $?_A : 0 \longrightarrow A$.

Dual to an initial object is a **final object**: a final object in a category $\mathbb{X}$ is an object to which every object has exactly one map.

Denote the final object by the numeral 1 and the unique map by $!_A : A \longrightarrow 1$.

What are these in Set, Mat(R), and Cat?

Initial and final objects

- In Set the initial object is the empty set and the final object is any one element set.
- In Mat(R) the initial object *and* the final object is the 0-dimensional object.
- In Cat the initial object is the empty category and the final category is any category with one object and one arrow.

INITIAL AND FINAL OBJECTS

A simple observation is:

Lemma

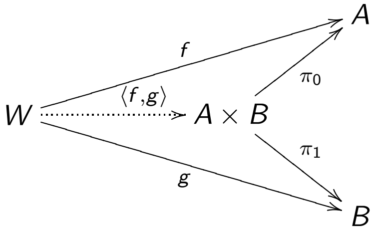
If K and K' are initial in $\mathbb{C}$ then there is a unique isomorphism $\alpha : K \longrightarrow K'$.

PROOF: As K is initial there is exactly one map $\alpha : K \longrightarrow K'$. Conversely, as K' is initial there is a unique map $\alpha' : K' \longrightarrow K$. This map is the inverse of α as $\alpha\alpha' : K \longrightarrow K$ is the unique endo-map on K namely the identity and similarly we obtain $\alpha'\alpha = 1'_K$. □

Thus initial objects (and by duality final objects) are unique up to unique isomorphism.

PRODUCTS AND COPRODUCTS

Let A and B be objects in a category then a **product** of A and B is an object, $A \times B$, equipped with two maps $\pi_0 : A \times B \longrightarrow A$ and $\pi_1 : A \times B \longrightarrow B$ such that given any object W with two maps $f : W \longrightarrow A$ and $g : W \longrightarrow B$ there is a unique map $\langle f, g \rangle : W \longrightarrow A \times B$, such that $\langle f, g \rangle \pi_0 = f$ and $\langle f, g \rangle \pi_1 = g$. That is:



The maps π_0 and π_1 are called **projections**.

Coproducts are dual.

PRODUCTS AND COPRODUCTS

- In Set the product is the cartesian product and the coproduct is the disjoint union.
- In Mat(R) the product of n and m *and* the coproduct is $n + m$.
- In Cat the product puts the categories in parallel the coproduct puts them side-by-side.

Are projections epic? In Set consider $A \times 0 \xrightarrow[\pi_0]{} A$.

PRODUCTS AS ADJOINTS

Given any category there is always a "diagonal" functor:

$$\Delta : \mathbb{X} \longrightarrow \mathbb{X} \times \mathbb{X}; \quad \begin{array}{c} X \\ {\scriptstyle f}\downarrow \\ Y \end{array} \quad \mapsto \quad \begin{array}{c} X \times X \\ \downarrow{\scriptstyle f \times f} \\ Y \times Y \end{array} \qquad \begin{array}{c} (x, y) \\ \downarrow \\ (f(x), f(y)) \end{array}$$

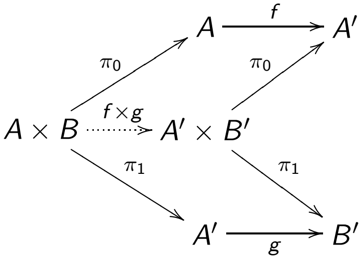
having products amounts to requiring that this functor is a left adjoint (namely $(Y, Z) \mapsto X \times Z$)!

$$\begin{array}{ccc} X & \xrightarrow{\Delta} & X \times X \\ & {\scriptstyle \langle f, g \rangle}\searrow & \vdots\, {\scriptstyle f \times g} \\ & & Y \times Z \end{array}$$

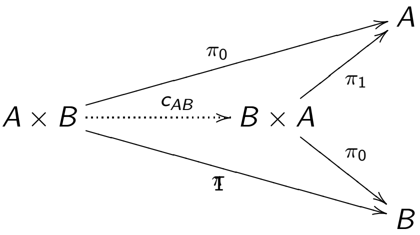
Here $\Delta = \langle 1_X, 1_X \rangle$ is the diagonal map in the category ...

PRODUCTS AND COPRODUCTS

It follows $\_ \times \_$ is a functor $f \times g$ is define as $\langle \pi_0 f, \pi_1 g \rangle$:



Any binary product has a symmetry map:



Note that $c_{AB} c_{BA} = 1_{A \times B}$ and so it is an isomorphism.

LIMITS AND COLIMITS

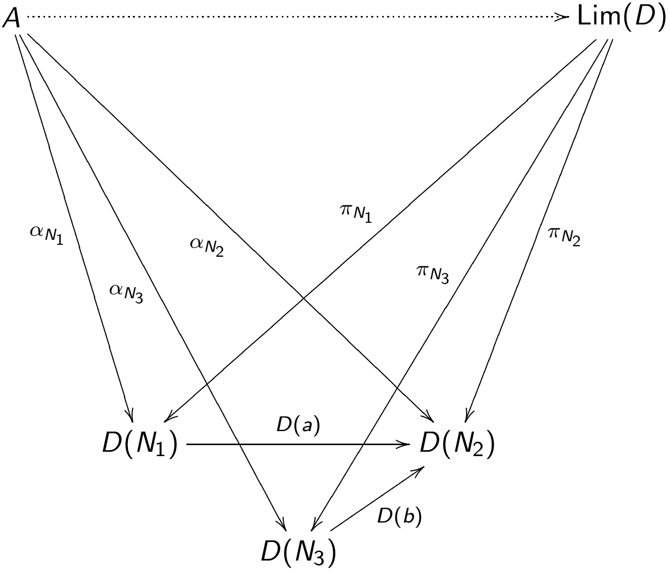
A **diagram**in $\mathbb{X}$ is a functor $D : \mathbb{G} \longrightarrow \mathbb{X}$ from a small category $\mathbb{G}$. A D–**cone** over this diagram consists of an object A, called the **apex** of the cone together with for each node N of $\mathbb{G}$ a map $\alpha_N : A \longrightarrow D(N)$ such that for each arrow of $\mathbb{G}$, $a : N_1 \longrightarrow N_2$, we have $\alpha_{N_1} G(a) = \alpha_{N_2}$.

A morphism of cones $(\alpha, h, \beta) : \alpha \longrightarrow \beta$ is given by a map in $\mathbb{C}$, $h : A \longrightarrow B$ between the apexes of the cones such that $\alpha_N = h\beta_N$ for all the nodes of the diagram.

Lemma

The cones over $D : \mathbb{G} \longrightarrow \mathbb{C}$ form a category, $\text{Cone}_D(\mathbb{C})$, *with objects the cones and maps the morphisms of cones.*

A **limit** of a diagram is a final object in $\mathrm{Cone}_D(\mathbb{C})$. The apex of this cone is written $\mathrm{Lim}(D)$ with projections $\pi_N : \mathrm{Lim}(D) \longrightarrow G(N)$.



ADJOINTS and LIMITS

Because a limit is given by a couniversal property

RIGHT ADJOINTS PRESERVE LIMITS

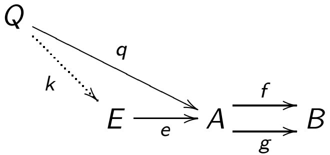
Dually

LEFT ADJOINTS PRESERVE COLIMITS

EQUALIZERS An **equalizer diagram** is a parallel pair of arrows:

$$A \underset{g}{\overset{f}{\rightrightarrows}} B$$

a cone for the above equalizer diagram is determined by a map to $q : Q \longrightarrow A$. Such a map is said to equalize f and g as $qf = qg$. A limit (E, e) is called **the equalizer** (even though it is not unique) and satisfies the property



that there is a unique k such that $ke = q$.

Lemma

Suppose (E, e) *is the equalizer of* $A \underset{g}{\overset{f}{\rightrightarrows}} B$ *then* e *is monic.*

COMPLETENESS AND COCOMPLETENESS
Final objects $\mathbb{G} = 0$, products $\mathbb{G} = \mathbf{1} + \mathbf{1}$.

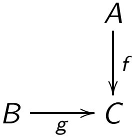
A category is **complete** when it has limits for all small diagrams. Dually it is **cocomplete** if it has colimits for all small diagrams.

There is an important theorem:

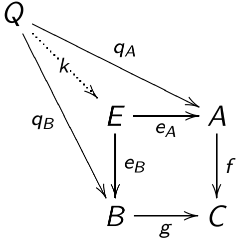
Theorem
A category is complete if and only if it has all products and equalizers.

PULLBACKS

Another important limit is the pullback (especially to these talks).
A **pullback diagram** is a binary fan of arrows:



a cone is given by a Q together with maps $q_A : Q \longrightarrow A$ and $q_B : Q \longrightarrow B$ such that $q_A f = q_B g$. A limit (E, e_A, e_B) is called **the pullback**:



and has a unique comparison map k from any cone such that $k e_A = q_A$ and $k e_B = q_B$.

PULLBACKS Products and equalizers imply pullbacks:

$$\begin{array}{ccc} P & \xrightarrow{g'} & A \\ {\scriptstyle f'}\downarrow & & \downarrow{\scriptstyle f} \\ B & \xrightarrow[g]{} & C \end{array}$$

is a pullback if and only if

$$P \xrightarrow{\langle f', g' \rangle} A \times B \underset{\pi_0 f}{\overset{\pi_1 g}{\rightrightarrows}} C$$

is an equalizer.

In Set the pullback is a subset of the product:

$$\{(a, b) | f(a) = g(b)\} \subseteq A \times B$$

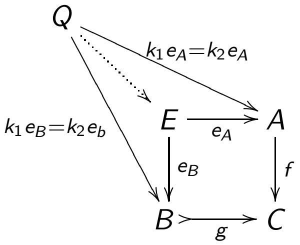
Lemma

In any category the pullback of a monic along any map is a monic.

PROOF: Suppose g is monic and $k_1 e_A = k_2 e_A$ then

$$k_1 e_B g = k_1 e_A f = k_2 e_A f = k_2 e_B g$$

so as $g$is monic $k_1 e_B = k_2 e_B$.

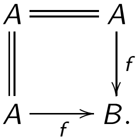


However, this makes k_1 and k_2 comparison maps from the outer square to the pullback. □

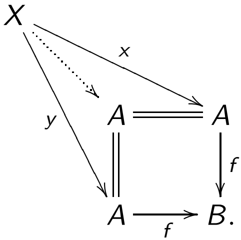
PULLBACKS

Lemma

In any category $f : A \longrightarrow B$ is monic iff the followings is a pullback:



PROOF: If $xf = yf$ there is a unique comparison map



which shows $x = y$. Conversely if f is monic then whenever we form the outer square $x = y$, so this gives a comparison map, whose uniqueness is forced by the fact that f is monic. $\square$

PULLBACKS
As right adjoints preserve pullbacks

RIGHT ADJOINTS PRESERVE MONICS

and dually ..

PULLBACKS

Lemma

In the following (commuting) diagram:

$$\begin{array}{ccccc} A & \xrightarrow{f} & B & \xrightarrow{g} & C \\ {\scriptstyle a}\downarrow & & {\scriptstyle b}\downarrow & & \downarrow{\scriptstyle c} \\ A' & \xrightarrow[f']{} & B' & \xrightarrow[g']{} & C' \end{array}$$

(i) if the two inner squares are pullbacks the outer square is a pullback;

(ii) if the rightmost square and outer square is a pullback the leftmost square is a pullback.

PULLBACKS

Products and pullbacks imply equalizers:

Lemma

The following square is a pullback

$$\begin{array}{ccc} E & \xrightarrow{e} & X \\ {\scriptstyle e'}\big\downarrow & & \big\downarrow{\scriptstyle \langle f, g\rangle} \\ Y & \xrightarrow[\Delta]{} & Y \times Y \end{array}$$

if and only if

$$E \xrightarrow{e} X \underset{g}{\overset{f}{\rightrightarrows}} Y$$

is the equalizer.

PULLBACKS

Pullbacks and a final object imply products:

Lemma

The following square is a pullback

$$\begin{array}{ccc} A \times B & \xrightarrow{\pi_0} & A \\ {\scriptstyle \pi_1}\downarrow & & \downarrow{\scriptstyle !} \\ B & \xrightarrow[!]{} & 1 \end{array}$$

if and only if

$$A \xleftarrow{\pi_0} A \times B \xrightarrow{\pi_1} B$$

is a product.

And so one has all finite limits when one has pullbacks and a final object ...