Contents

Chapter 1

Introduction

Downcasing types (working draft, 2009nov16)

Eduardo Ochs
LLaRC, PURO/UFF
eduardoochs@gmail.com
http://www.uff.br/llarc/
http://angg.twu.net/

The latest version can be found at:
http://angg.twu.net/LATEX/2009unilog-dnc.pdf

Note: this is a *first, very rough working draft* of a paper on Downcasing Types... let me explain how it came into being. When I started writing down my material about DNC (from "DowNCasing") a few months ago I was with the impression that it would be too hard to publish it as an article in a journal — because of the reasons in section 1.2 — but it occurred to me that I could take an alternative route: to write first a technical report, mainly for a very specific audience — for the people in my research group in Rio das Ostras and for few other groups in Rio de Janeiro (and there are no categorists in the strict sense here!), with all details and all dictionaries of notations, to help these people understand some papers and books in CT...

Then another excuse to finish it appeared: I was trying to help Valeria de Paiva — whom I knew from several events in Logic in Brazil, and who had even watched my very first international talk (see `http://angg.twu.net/math-b.html#PhD`) — to find out how she could become a visiting researcher in Brazil for a few months, and at one point she asked casually if I wasn't going to submit anything to the session on Categorical Logic at UNILOG'10, where she was one of the organizers... I wrote the first version of the abstract — not the one below —, asked for comments ("what she thought the referees could complain about", etc, as we still had ten days before the deadline), and she pointed out that I made liftings look too easy, that the notion of downcasing — and its non-triviality — didn't make enough sense from what I wrote, and a few other things that don't matter now...

Then, in the space of little more than one week, I was able to write all this down (with some bits of cut-and-paste from past seminar notes, of course). It doesn't matter exactly what Valeria said; in some situations what really matters is *what we do* with what other people say to us — and we both wanted these ideas to be written down, and now they are *much* closer to being in a readable form than they were before...

[Expect big changes in the text below over the next few weeks...]

1.1 Abstract

(Taken from: `http://angg.twu.net/LATEX/2009unilog-abs1.pdf`)

When we represent a category $\mathbf{C}$ in Type Theory it becomes a 7-uple: $(\mathbf{C}_0, \mathrm{Hom}_{\mathbf{C}}, \mathrm{id}_{\mathbf{C}}, \circ_{\mathbf{C}}; \mathsf{assoc}_{\mathbf{C}}, \mathsf{idL}_{\mathbf{C}}, \mathsf{idR}_{\mathbf{C}})$, where the first four components are "structure" and the last three are "properties".

We call the "structure" components the "syntactical part", and the "properties" components the "logical part". A *protocategory* is a 4-uple $(\mathbf{C}_0, \mathrm{Hom}_{\mathbf{C}}, \mathrm{id}_{\mathbf{C}}, \circ_{\mathbf{C}})$ — just the "syntactical skeleton" of what a category is, without the components that talk about equality of morphisms. By splitting at the right places the uples that represent functors, natural transformations, isos, adjunctions, limits, etc, we define proto-functors, proto-NTs, and so on.

The operation that takes entities and returns the corresponding proto-entities behaves as a projection, and we say that it goes from the "real world" — where everything has both a "syntactical" and a "logical" part — to the "syntactical world", where only the syntactical parts have been kept.

The opposite of to *project* is to *lift*. We may start with a proto-something, s^-, in the syntactical world, and try to lift it to an s in the real world that projects into s^-. Meta-theorems about lifting are hard to obtain, but we know many interesting liftings — each object r of the (projectable fragment of the) real world projects to an proto-object r^- that can be lifted back to r — and we can start by studying them to understand how liftings behave.

Proto-objects — even proto-proofs — are especially amenable to being represented diagrammatically, and there is a simple way to attribute a precise meaning to each entity — each node, arrow, etc — appearing in these diagrams. We will show how to formalize two such diagrammatic proofs — the Yoneda Lemma and one of the weakest monadicity theorems — as terms in Coq.

For most applications in Categorical Semantics one further trick is needed: "downcasing types", that lets us name entities by what they represent in the "archetypical case". For example, in a hyperdoctrine, if P is an object over $B \times C$ and $f : A \to B$ then Beck-Chevalley Condition for $\forall$ says that the natural morphism from $f^*\Pi_{\pi_{BC}}P$ to $\Pi_{\pi_{AC}}(f \times C)^*P$ should be an iso. In the archetypical hyperdoctrine, $\mathrm{Sub}(\mathbf{Set})$, P "is" a subset $\{\, (b,c) \in B \times C \mid P(b,c) \,\}$ of $B \times C$, and both $f^*\Pi_{\pi_{BC}}P$ and $\Pi_{\pi_{AC}}(f \times C)^*P$ "deserve the name" $\{\, a \in A \mid \forall c \in C.\, P(fa, c) \,\}$. The downcasing of P is $b, c||P$, and the BCC map

becomes a map $a||\forall c.P \mapsto a||\forall c.P$ that is not the identity, whose construction can be read out from a diagram.

Roughly, what is the happening is the following: the formal definition of hyperdoctrine generalizes *some* of the structure of Sub(**Set**); with our way of interpreting diagrams we can define all this structure diagrammatically, in a notation that "suggests" that we are in Sub(**Set**), i.e., "in the archetypical case", and then we can "lift" these definitions to diagrams with the same two-dimensional structure, but in any of the standard notations.

Several categorical theorems become quite clear when we find "archetypical diagrams" for their (proto-)proofs, and then we lift those to standard notations; we will show some examples from Lawvere's "Adjointness in Foundations" (1969) and "Equality in Hyperdoctrines" (1970) papers.

1.2 There Are no Theorems in This Paper

...because the things that we usually call "theorems" in Category Theory belong to the real world — they are a construction *plus something more*.

Take for example the Yoneda Lemma. It says that given a functor $R : \mathbf{B} \to \mathbf{Set}$ and an object B of $\mathbf{B}$ we have a bijection between the set RB and the set of natural transformations $C \dot{\to} ((B \to C) \to RC)$. Here we are working in the syntactical world only — we *mention* liftings, but we don't do any liftings explicitly, and so what we get is just the projection of that bijection, which is a proto-iso between RB and the set of proto-NTs $C \dot{\to} ((B \to C) \to RC)$. Usually a "theorem" involving a such construction would have to either show that it is always a bijection, or to show a case where it is *not* a bijection.

What we do have here is the definition of the two worlds (mostly via examples, but whatever...), of the projections, of the liftings, some ideas of how to work with this splitting of worlds, and examples.

Chapter 2

The syntactical world

2.1 Categories

[Typeset this from my handwritten notes]

2.2 Functors

2.3 Natural Transformations

2.4 Isos

2.5 Epis and Monics

2.6 Adjunctions

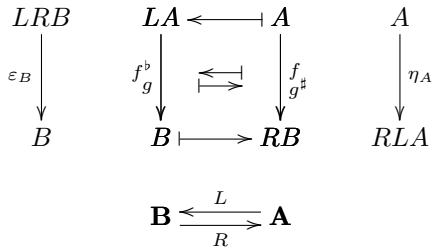
If L and R are functors going in opposite directions between two categories, say,

$$\mathbf{B} \underset{R}{\overset{L}{\leftrightarrows}} \mathbf{A}$$

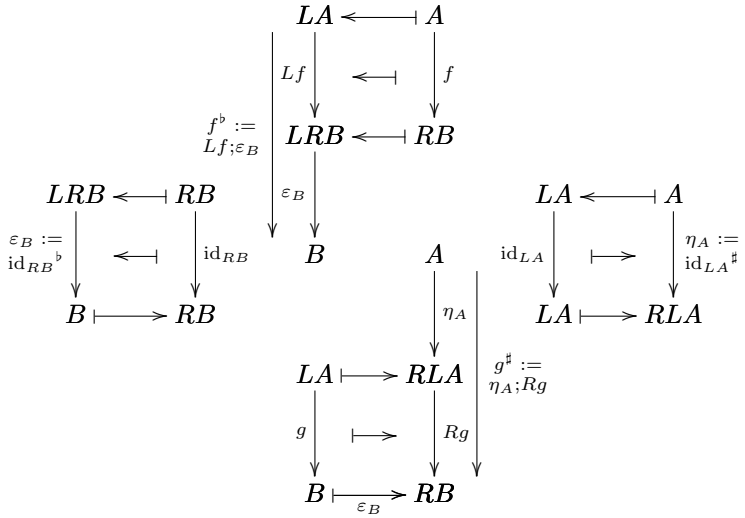
then a *proto-adjunction*, $L \dashv R$, is an 8-uple,

$$(\mathbf{A}, \mathbf{B}, L, R, \flat, \sharp, \eta, \epsilon)$$

that we draw as:



There is some redundancy in this definition, as we may reconstruct some of the entities $\flat$, $\sharp$, η, ϵ in terms of the other ones:



2.7 Monads

A *protomonad* for a proto-endofunctor $T : \mathbf{A} \to \mathbf{A}$ is a 4-uple:

$$(\mathbf{A}, T, \eta, \mu)$$

that we draw as:

$$A \xrightarrow{\eta_A} TA \xleftarrow{\mu_A} TTA$$

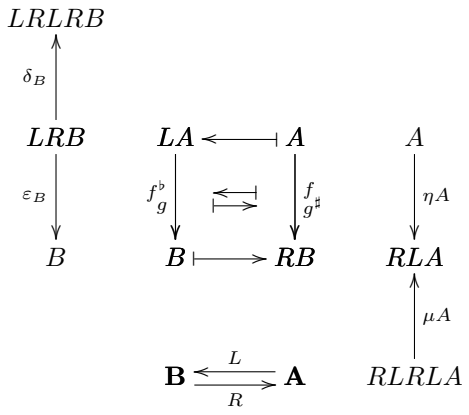
A *proto-comonad* for a proto-endofunctor $S : \mathbf{B} \to \mathbf{B}$ is a 4-uple:

$$(\mathbf{B}, S, \varepsilon, \delta)$$

that we draw as:

$$B \xleftarrow{\varepsilon_B} SB \xrightarrow{\delta_B} SSB$$

Each proto-adjunction induces both a proto-monad and a proto-comonad. We draw all these together as:



We define $\mu_A := R(\mathrm{id}_{RLA}{}^{\flat})$ and $\delta_B := L(\mathrm{id}_{LRB}{}^{\sharp})$:

$$\frac{\dfrac{\mathrm{id}_{RLA} : RLA \to RLA}{\mathrm{id}_{RLA}{}^{\flat} : LRLA \to LA}\ \flat}{\mu_A := R(\mathrm{id}_{RLA}{}^{\flat}) : RLRLA \to RLA}\ R \qquad \frac{\dfrac{\mathrm{id}_{LRB} : LRB \to LRB}{\mathrm{id}_{LRB}{}^{\sharp} : RB \to RLRB}\ \sharp}{\delta_B := L(\mathrm{id}_{LRB}{}^{\sharp}) : LRB \to LRLRB}\ L$$

We have seen how a proto-adjunction induces a proto-monad; now we will see how a proto-monad induces *two* proto-adjunctions.

2.7.1 The Kleisli Category of a Monad

The *Kleisli proto-category* of a proto-monad $(\mathbf{A}, T, \eta, \mu)$ is the proto-category:

$$\mathbf{A}_T := ((\mathbf{A}_T)_0, \mathrm{Hom}_{\mathbf{A}_T}, \mathrm{id}_{\mathbf{A}_T}, \circ_{\mathbf{A}_T})$$

where $(\mathbf{A}_T)_0$ is equal to $\mathbf{A}_0$, but we write the objects of $(\mathbf{A}_T)_0$ in a funny way: an object $A \in \mathbf{A}$ becomes

$$[A - \succ TA]$$

when we regard it as an object of $(\mathbf{A}_T)_0$.

A morphism in $\mathrm{Hom}_{\mathbf{A}_T}([A - \succ TA], [C - \succ TC])$ is just a map $f : A \to TC$ in $\mathrm{Hom}_{\mathbf{A}}(A, TC)$. We write it as $[f] : \mathrm{Hom}_{\mathbf{A}_T}([A - \succ TA], [C - \succ TC])$ to stress that its (formal) type is different from f.

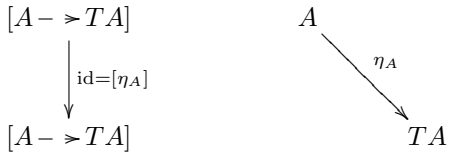
The identity operation, $\mathrm{id}_{\mathbf{A}_T}$, is the η (the "unit") of the monad in disguise:

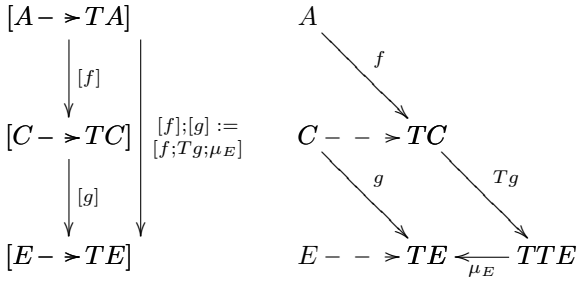
$$\mathrm{id}_{\mathbf{A}_T}([A - \succ TA]) := [\eta_{\mathbf{A}}]$$

Note that:

$$\frac{\dfrac{\dfrac{A : \mathbf{A}_0}{\eta_A : A \to TA}}{[\eta_A] : [A - \succ TA] \to [A - \succ TA]}}{\mathrm{id}_{\mathbf{A}_T}([A - \succ TA]) : \mathrm{Hom}_{\mathbf{A}_T}([A - \succ TA], [A - \succ TA])}\ \mathsf{ren}$$

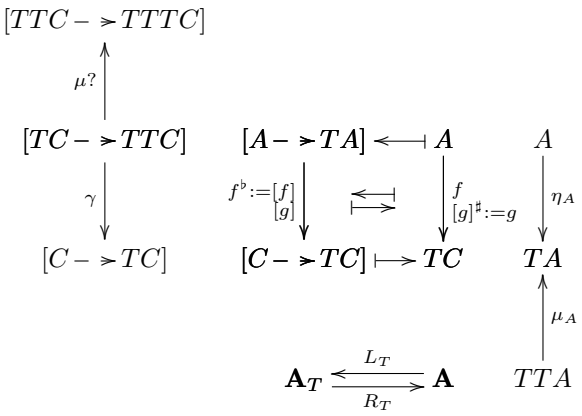
The composition, $\circ_{\mathbf{A}_T}$, needs a trick: if $f : A \to TC$ and $g : C \to TE$ then $[f];[g] := [f; Tg; \mu_E]$. In diagrams:





The dashed arrow in, say, $[A- \succ TA]$, is to suggest three things:
that morphisms in $\mathbf{A}_T$ follow the direction of the '$- \succ$',
that a morphism $A \to TA$ is not part of the definition of an object $[A- \succ TA]$,
that the '$- \succ$' is the ghost of the unit of the monad — the unit would go from A to TA, but it is not used in the definitions; nevertheless, its memory remains.

We can draw the Kleisli (proto-)adjunction as:



2.7.2 The Eilenberg-Moore Category of a Monad

The *Eilenberg-Moore proto-category* for a proto-monad $(\mathbf{A}, T, \eta, \mu)$ is:

$$\mathbf{A}^T := ((\mathbf{A}^T)_0, \mathrm{Hom}_{\mathbf{A}^T}, \mathrm{id}_{\mathbf{A}^T}, \circ_{\mathbf{A}^T})$$

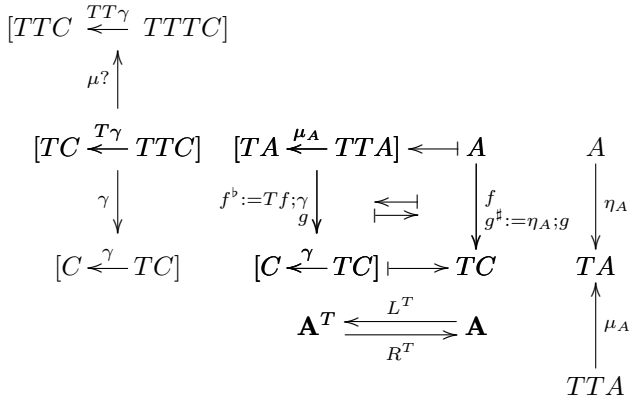
where an object of $(\mathbf{A}^T)_0$ is a pair (A, α) (a "proto-algebra"), that we write as:

$$[A \overset{\alpha}{\leftarrow} TA]$$

We use a non-dashed arrow, '$\leftarrow$', to stress that the map $\alpha : \mathrm{Hom}_{\mathbf{A}}(TA, A)$ is part of the definition of the object.

A (proto-)morphism $f : [A \overset{\alpha}{\leftarrow} TA] \to [C \overset{\gamma}{\leftarrow} TC]$ is just a morphism $f : \mathrm{Hom}_{\mathbf{A}}(A, C)$. The identity $\mathrm{id}_{\mathbf{A}^T}$ and the composition $\circ_{\mathbf{A}^T}$ are defined in the obvious way (inherited from $\mathbf{A}$).

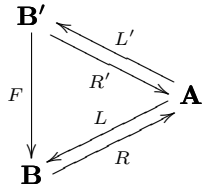
The Eilenberg-Moore adjunction can be drawn as:



where [two triangles showing the transpositions]:

2.7.3 The Comparison Theorem

If $\mathbf{B} \underset{R}{\overset{L}{\leftrightarrows}} \mathbf{A}$ and $\mathbf{B}' \underset{R'}{\overset{L'}{\leftrightarrows}} \mathbf{A}$ are two proto-adjunctions — the full definition with '$\flat$'s, '$\sharp$'s, 'η's and 'ε's will not be relevant now — then a *proto-comparison* from $L' \dashv R'$ to $L \dashv R$ is just a proto-functor $F : \mathbf{B}' \to \mathbf{B}$,

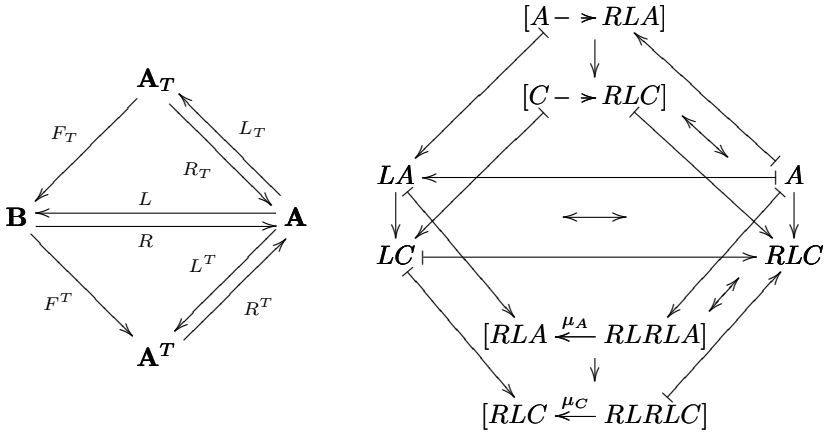


such that $(L'; F)_0 = L_0$ and $(F; R)_0 = R'_0$, i.e., two of the triangles in the figure commute on objects (remember that in the syntactical world equality of morphisms rarely matters). Note that we only need to care for the objects of $\mathbf{B}$ that are images of objects in $\mathbf{B}'$.

Now start with a proto-adjunction, $\mathbf{B} \underset{R}{\overset{L}{\leftleftarrows}} \mathbf{A}$, build its protomonad,

$$(\mathbf{A}, T, \eta, \mu) = (A, (R; L), \eta, (\lambda A{:}\mathbf{A}_0.R(\mathrm{id}_{RLA}{}^\flat))),$$

and build its Kleisli proto-adjunction $\mathbf{A}_T \underset{R_T}{\overset{L_T}{\leftleftarrows}} \mathbf{A}$, and its Eilenberg-Moore proto-adjunction, $\mathbf{A}^T \underset{R^T}{\overset{L^T}{\leftleftarrows}} \mathbf{A}$. It turns out that we have proto-comparison functors $F_T : \mathbf{A}_T \to \mathbf{B}$ and $F^T : \mathbf{B} \to \mathbf{A}^T$:



where the functor F_T acts like this on morphisms,

$$\frac{[f] : [A - \succ RLA] \to [C - \succ RLC]}{\dfrac{f : A \to RLC}{f^\flat : LA \to LC}}$$

and the functor F^T, that appears as $LA \mapsto [RLA \overset{\mu_A}{\leftarrow} RLRLA]$ in the diagram, is actually $B \mapsto [RB \overset{R(\mathrm{id}_{RB}{}^\flat)}{\leftarrow} RLRB]$:

$$\frac{\dfrac{\dfrac{\dfrac{A}{LA}\,L}{RLA}\,R}{RLA \to RLA}\,\mathrm{id}}{LRLA \to LA}\,\flat \qquad \frac{\dfrac{\dfrac{\dfrac{B}{RB}\,R}{RB \to RB}\,\mathrm{id}}{LRB \to B}\,\flat}{R(\mathrm{id}_{RB}{}^\flat) : RLRB \to RB}\,R$$
$$R(\mathrm{id}_{RLA}{}^\flat) : RLRLA \to RLA\,R$$

We need to impose the condition $\forall A{:}\mathbf{A}_0.\mu_A = R(\mathrm{id}_{RLA}{}^\flat)$, but this holds in any (non-proto-) category.

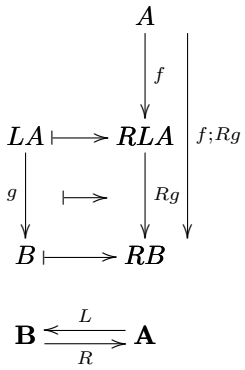
2.7.4 Monadicity

[Define proto-equivalence of categories]
[Define tripleable functor]
[Give examples?]
[Define proto-equalizer]
[Define preservation and reflection of proto-equalizers]
[Prove Theorem 1 of Beck's thesis (p.8 of the reprint)]
`http://www.tac.mta.ca/tac/reprints/articles/2/tr2abs.html`

2.7.5 Beck's Lemma

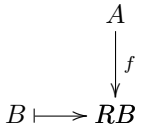
2.8 Universals

In an adjunction, each morphism $A \to RLA$ induces a natural transformation $B \dot{\to} ((A \to B) \to (A \to RB))$, by:

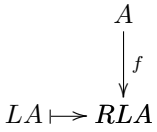


and for $f = \eta_A$ the natural transformation is a natural iso. The notion of "universal" generalizes this, and makes sense when we only have the functor $R : \mathbf{B} \to \mathbf{A}$.

A *preuniversal* for a functor $R : \mathbf{B} \to \mathbf{A}$ is a triple $(A : \mathbf{A}_0, B : \mathbf{B}_0, f : A \to RB)$, that we will draw as:



or as (A, LA, f) i.e.,

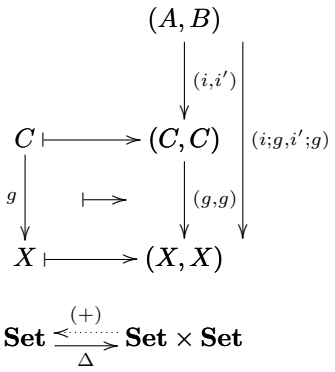


where the 'LA' is just a "long name" for an object of $\mathbf{B}$ — we don't have a functor L anymore, so L is just a letter.

A *universal* is a preuniversal plus "universality", where *universality* is the assertion that the induced natural transformation is a natural iso. In the syntactical world universality looks more like extra structure than like a property: a (proto-)universal is a (proto-)preuniverse $(A, LA, f : A \to RLA)$ plus a proto-inverse for its induced natural transformation.

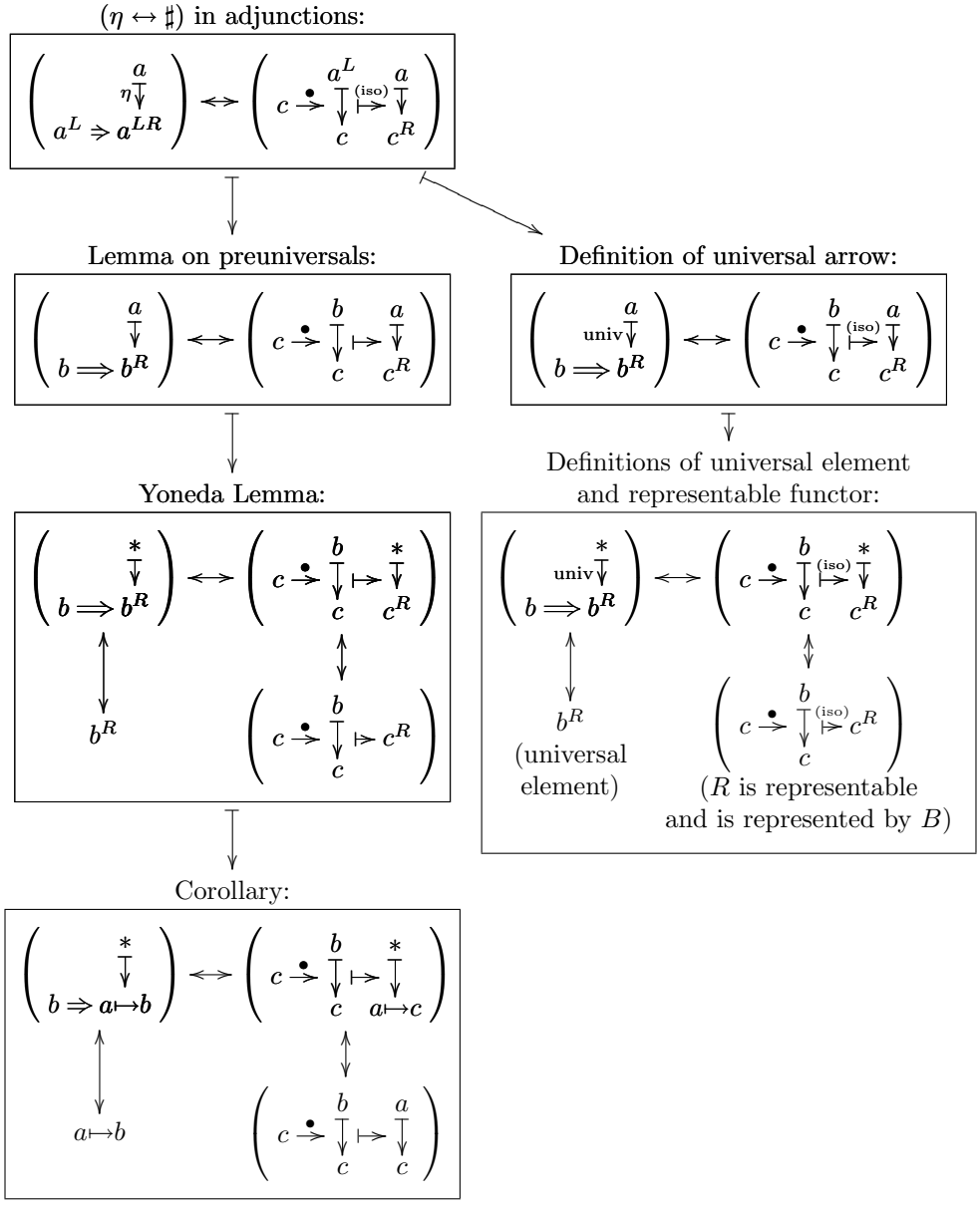
The idea — let me make it clearer — is that in the syntactical world we "prove" than an (A, LA, f) is a universal by *constructing a proto-inverse for it*, where the proto-inverse is an operation that behaves "syntactically" (i.e., in the types) as an inverse to the natural transformation, and checking that this proto-inverse is a real inverse is something that is left to a later stage — the "lifting".

An example should make this clearer. Any diagram $A \xrightarrow{i} C \xleftarrow{i'} B$ (mnemonic: C will "deserve the name" $A + B$, as we will see in the next section) induces a natural transformation $X \dot{\to} ((C \to X) \to (A \to X) \times (B \to X))$, by:



[...]
$[X \dot{\to} (\mathrm{Hom}(C, X) \to \mathrm{Hom}(A, X) \times \mathrm{Hom}(B, X))]$
$[X \dot{\to} (\mathrm{Hom}(C, X) \to \mathrm{Hom}((A, B), (X, X))]$
$[X \dot{\to} (\mathrm{Hom}_{\mathbf{Set}}(L(A, B), X) \to \mathrm{Hom}_{\mathbf{Set}\times\mathbf{Set}}((A, B), RX)]$
[...]

2.9 The Yoneda Lemma



2.10 Products

2.11 To Deserve a Name

In a diagram $A \xrightarrow{i} C \xleftarrow{i'} B$ we say that the 'C' *deserves the name* '$A + B$', and that i and i' deserve the names ι and ι', when $A \xrightarrow{i} C \xleftarrow{i'} B$ is a coproduct diagram.

[Similarly for products, pullbacks, etc...]

We say that the 'C' is a *candidate for* '$A + B$' when... [This is similar to stating: "*Proposition: Foo*", and then proceeding to a proof of Foo.]

2.11.1 Mad Names

In section **??** we hint to a definition for "downcasing" and "uppercasing" that is much stricter than the one that is really useful...

"Downcasing" and "uppercasing" should *not* be precise syntactical transformations on names! Instead, what works well is to keep a set of loosely-defined guidelines for downcasings and uppercasing, that need not be followed very rigorously, and to keep a "dictionary of uppercasings and downcasings" with several entries like "$a' \mapsto b : A \to B$', '$f \equiv a \mapsto b$' (I use '$\equiv$' to mean "change of notation", usually from standard to downcased)... if this dictionary can be reconstructed from very hints, the better.

There is *nothing* — apart from the desire to think clearly and to communicate our thoughts — that prevents us, in the formalization of some construction or theorem in Coq with LaTeX chunks (section **??**), from using "mad names" like, say, '$\$^{\%}_{\#}$' instead of '$a \mapsto b$'. The language of long names and downcasings presented here is not like an internal language, like the ones that we use for toposes (see [BellLST], [LambekScott], etc); rather, it is more like *half* of an internal language: a corpus of sentences and diagrams, as they are "spoken", but without a fixed grammar — we've split the *use of a language* from its *formalization*.

By the way, we can use the idea of "diagrams as dictionaries between notations" (see sec. **??**) to change from the n-th attempt to a good notation to the $(n + 1)$-th attempt; the diagrams keep the same shape...

2.12 Terminals

2.13 Exponentials

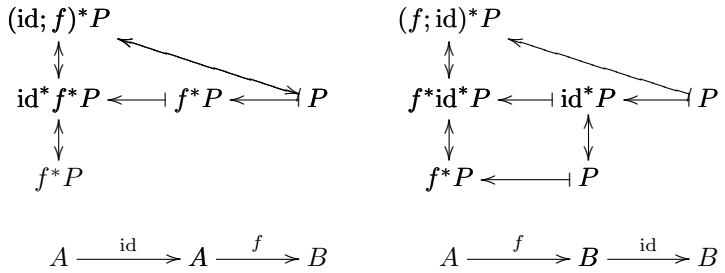
2.14 Cartesian Categories

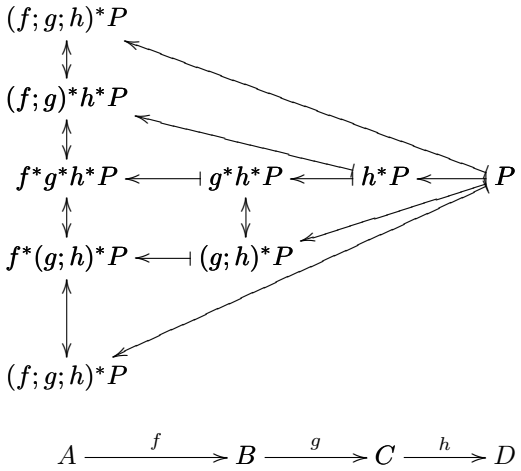
2.15 Cartesian Closed Categories

2.16 Indexed Categories, Fibrations, Hyperdoctrines

2.16.1 Subobjects

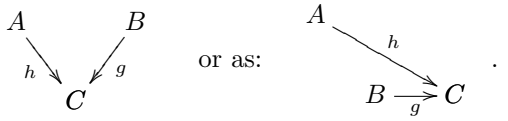
2.16.2 Indexed Categories





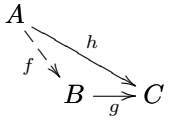
2.16.3 Cartesian Morphisms

A "vee" in a category $\mathbb{A}$ is a pair of maps $(h : A \to C,\, g : B \to C)$ with the same codomain; we will draw vees as:



.

A *completion* for a vee $(h : A \to C,\, g : B \to C)$ is a map $f : A \to B$ such that $f; g = h$. Let's draw the set of all completions for the vee $(h : A \to C,\, g : B \to C)$ as this:



.

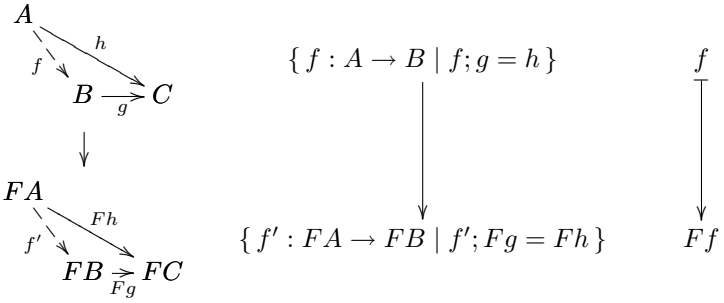
The standard notation for that, using slice categories (see [TTT], p.3, or [Awodey], p.15), would be:

$$\mathrm{Hom}_{\mathbb{A}/C}(h : A \to C,\, g : B \to C).$$

A functor $F : \mathbb{A} \to \mathbb{B}$ takes vees in $\mathbb{A}$ to vees in $\mathbb{B}$,

$$(h : A \to C,\, g : B \to C) \;\mapsto\; (Fh : FA \to FC,\, Fg : FB \to FC)$$

and induces functions from sets of completions to sets of completions:

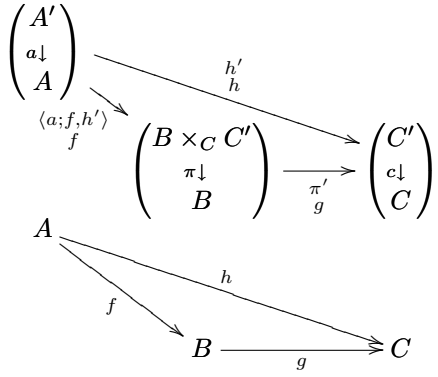


When a map $\left(\begin{array}{ccc} A & & \\ & \searrow^{h} & \\ f \searrow & & C \\ & B \underset{g}{\rightrightarrows} & \end{array} \right) \to \left(\begin{array}{ccc} FA & & \\ & \searrow^{Fh} & \\ f' \searrow & & FC \\ & FB \underset{Fg}{\rightrightarrows} & \end{array} \right)$ is a bijection there is a unique completion $f : A \to B$ corresponding to each completion $f' : FA \to FB$. We will then say that this f is the *lifting* of the corresponding f', and we will say that the vee (h, g) *has unique liftings* (for the functor F; but let's think of F as fixed).

When a map $g : B \to C$ has the property that all vees of the form (h, g) have unique liftings — note that this must hold for all possible choices of domains for h, i.e., now the 'A' is free — then we will say that g is *cartesian*. This property is not so rare as it might seem.

Fact. If $F : \mathbf{Set}^{\to} \to \mathbf{Set}$ is the "codomain" functor, Cod, then pullbacks in **Set** — regarded as morphisms in $\mathbf{Set}^{\to}$ — are cartesian morphisms for F.

The diagram below is the core of the proof. Each completion (f', f) "above" is mapped to a completion f "below"; each completion f below lifts to a completion $(\langle a; f, h' \rangle, f)$ above.



2.16.4 Cleavages

Now let's clean up the notation. The functor $F : \mathbb{A} \to \mathbb{B}$ will become the "projection functor" $p : \mathbb{E} \to \mathbb{B}$, going from the "entire category" $\mathbb{E}$ to the "base category". Projection functors are just normal functors, but they are downcased in a funny way, as we shall see soon.

The objects of $\mathbb{B}$ will be called A, B, C, D, ... the objects of $\mathbb{E}$ will be called P, Q, R, S, ...; in the archetypal case — namely, $p \equiv \mathrm{Cod} : \mathrm{Sub}(\mathbf{Set}) \to \mathbf{Set}$ — they will stand for *propositions* over *sets*. For example, an object Q of $\mathbb{E}$ over an object B of $\mathbb{B}$ will stand for the subobject $\{\, b \in B \mid Q(b) \,\} \rightarrowtail B$:

$$\begin{array}{ccc} \mathrm{Sub}(\mathbf{Set}) & \qquad & Q \equiv (\{\, b \in B \mid Q(b) \,\} \rightarrowtail B) \\ {\scriptstyle p \equiv \mathrm{Cod}}\downarrow & & \downarrow \\ \mathbf{Set} & & pQ = \mathrm{Cod}\, Q = B \end{array}$$

An object Q of $\mathbb{E}$ is said to be *over* its projection pQ; similarly, morphisms in $\mathbb{E}$ are said to be *over* their images. For each object B of $\mathbb{B}$ the subcategory of $\mathbb{E}$ formed by the objects and morphisms over B and id_B is called the *fiber over* B, and denoted by $\mathbb{E}_B$. Also, we will tend to draw objects and morphisms of $\mathbb{E}$

over their images and omit the funtor arrows for p. So

$$\begin{array}{ccc} Q & \xrightarrow{g'} & R \\ & & \\ B & \xrightarrow{g} & C \end{array} \qquad \text{is shorthand for} \qquad \begin{array}{ccc} Q & \xrightarrow{g'} & R \\ \downarrow & & \downarrow \\ B = pQ & \xrightarrow{g=pg'} & C = pR \end{array} \qquad \begin{array}{c} \mathbb{E} \\ \downarrow p \\ \mathbb{B} \end{array} .$$

Morphisms in a fiber are said to be *vertical*.

A *cleavage* for a projection funtor p is a pair of operations, $(\cdot^*, \bar{\cdot})$, that produces, for each morphism $g : B \to C$ in $\mathbb{B}$ and for each R in $\mathbb{E}$ over C, a cartesian morphism $\overline{g}_R : g^*R \to R$ over g.

A *cartesian lifting* for $g : B \to C$ at an object R over C is a cartesian morphism $Q \to R$ over g with codomain R. A cleavage $(\cdot^*, \bar{\cdot})$ chooses a particular cartesian lifting for each pair $(R, g : B \to pR)$.

A *cloven fibration* is a projection functor $p : \mathbb{E} \to \mathbb{B}$ plus a cleavage $(\cdot^*, \bar{\cdot})$. A *fibration* is slightly less than this: a projection functor $p : \mathbb{E} \to \mathbb{B}$ plus the guarantee that each pair $(R, g : B \to pR)$ has at least one cartesian lifting.

We will not use "plain" fibrations here; cleavages are too convenient.

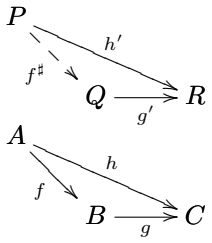
The following fact may at first look too technical to be useful.

Fact (technical). In a cloven fibration $p : \mathbb{E} \to \mathbb{B}$, for each object R in $\mathbb{E}$ the "projection" operation $p_R : \mathbb{E}/R \to \mathbb{B}/pR$ is left adjoint to the "cartesian lifting" operation $\bar{\cdot}_R : \mathbb{B}/pR \to \mathbb{E}/R$. In diagrams:

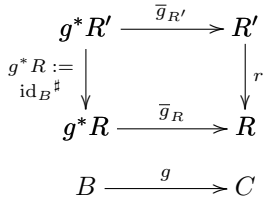
$$\begin{array}{ccc} \mathbb{E}/R & & P \\ {\scriptstyle p_R}\downarrow\uparrow{\scriptstyle \bar{\cdot}_R} & & {\scriptstyle f'}\searrow \quad \nearrow{\scriptstyle h'} \\ \mathbb{B}/pR & & g^*R \xrightarrow[\overline{g}_R]{} R \\ & & pP \\ & & {\scriptstyle f}\searrow \quad \nearrow{\scriptstyle ph'} \\ & & B \xrightarrow[g]{} pR \end{array} \qquad \begin{array}{ccc} (P \xrightarrow{h'} R) & \xrightarrow{f'} & (g^*R \xrightarrow{\overline{g}_R} R) \\ {\scriptstyle p_R}\downarrow & \flat\downarrow\uparrow\sharp & \downarrow{\scriptstyle \bar{\cdot}_R} \\ (pP \xrightarrow{ph'} pR) & \xrightarrow[f]{} & (B \xrightarrow{g} pR) \end{array}$$

The transposition '$\flat$' is a familiar operation: it take each f' to its projection pf'. The inverse transposition, '$\sharp$', is something new, and extremely important — it factors h' through the cartesian morphism $\overline{g}_R$, returning a morphism f' over f.

In a vee $(h' : P \to R, g' : Q \to R)$ over $(h : A \to C, g : B \to C)$ in which g' is cartesian each completion $f : A \to B$ of the lower vee lifts in a unique way to a completion $f' : P \to Q$ of the upper vee. When the upper vee is clear from the context we will denote this f' by $f^\sharp$, and we will call $f^\sharp$ the "factorization of h' through g' (over f)". Note that here g' is any cartesian morphism — not necessarily one returned by the cleavage.



For any morphism $g : B \to C$ in the base, the operation $R \mapsto g^*R$ given by the cleavage is the action of a functor on objects. To construct the action of the functor $g^* : \mathbb{E}_C \to \mathbb{E}_B$ on morphisms we use the factorization '$\sharp$': if $r : R' \to R$ is a morphism in $\mathbb{E}_C$, then $g^*r : g^*R' \to g^*R$ is the factorization of $\overline{g}_{R'}; r$ through $\overline{g}_R$ (over id_B).



Each g^* turned out to be a functor; it turns also out that each $\overline{g}$ is a natural transformation. More precisely: for each morphism $g : B \to C$ in the base the operation $R \mapsto (g^*R \stackrel{\overline{g}_R}{\to} \overline{g})$ is the action of a natural transformation $\overline{g}$ from $g^* : \mathbb{E}_C \to \mathbb{E}_B$ to $\mathrm{id}_{\mathbb{E}_C} : \mathbb{E}_C \to \mathbb{E}_C$.

A *proto-vee* is just a vee.

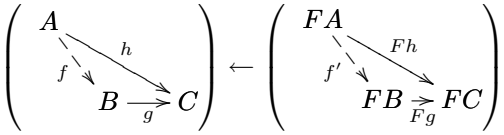
A *proto-completion* for a vee $(h : A \to C, g : B \to C)$ is just a morphism $f : A \to B$.

A morphism $f' : P \to Q$ is *proto-above* a morphism $f : A \to B$ if $pP = A$ and $pQ = B$. This is like being above, but here we check only the domain and the codomain; the condition $pf' = f$ has been dropped.

The condition of "having unique liftings" for a vee becomes an operation in the syntactical world: a proto-inverse for the

For a (proto-)vee

A *vee with proto-unique liftings* is a vee $(h : A \to C, g : B \to C)$ plus an operation that takes each proto-completion f' of $(Fh : FA \to FC, Fg : FB \to FC)$ to a proto-completion f of $(h : A \to C, g : B \to C)$. Note that there may be many more proto-completions of (Fh, Fg) than completions. We may draw a proto-unique lifiting for (h, g) as



i.e., as a proto-inverse for the natural arrow '$\rightarrow$' from the first set of proto-completions to the second.

A *proto-cartesian arrow* $g : B \to C$ is an arrow $g : B \to C$ plus an operation

$$(A,\ A \xrightarrow{h} C) \mapsto \left(\begin{array}{ccc} A & & \\ & \searrow^{h} & \\ {\scriptstyle f}\searrow & & \\ & B \xrightarrow[g]{} & C \end{array} \right) \leftarrow \left(\begin{array}{ccc} FA & & \\ & \searrow^{Fh} & \\ {\scriptstyle f'}\searrow & & \\ & FB \xrightarrow[Fg]{} & FC \end{array} \right)$$

that takes the each possible other leg for the vee and returns a corresponding proto-unique lifting.

A *proto-projection functor* is just a proto-functor.

A *proto-cleavage* for a proto-projection functor $p : \mathbb{E} \to \mathbb{B}$

[Missing: proto-above]./

2.16.5 Change-of-Base Functors

2.16.6 The Logical Structure in Each Fiber

2.16.7 Preservations

In the semi-logical notation, if $f \equiv (a, b \mapsto a)$, then $\mathsf{P}\wedge^{\natural}$, $\mathsf{P}\supset^{\natural}$ e $\mathsf{Frob}^{\natural}$ are:

$$\frac{f \equiv (a, b \mapsto a) \quad \mathbf{O}[a; P] \quad \mathbf{O}[a; Q]}{a, b; f^*(P \wedge Q) \vdash f^*P \wedge f^*Q}\ \mathsf{P}\wedge^{\natural} \qquad \frac{f \equiv (a, b \mapsto a) \quad \mathbf{O}[a; P] \quad \mathbf{O}[a; Q]}{a, b; f^*P \wedge f^*Q \vdash f^*(P \wedge Q)}\ \mathsf{P}\wedge$$

$$\frac{f \equiv (a, b \mapsto a) \quad \mathbf{O}[a; Q] \quad \mathbf{O}[a; R]}{a, b; f^*P \wedge f^*Q \vdash f^*(Q \supset R)}\ \mathsf{P}\supset^{\natural} \qquad \frac{f \equiv (a, b \mapsto a) \quad \mathbf{O}[a; Q] \quad \mathbf{O}[a; R]}{a, b; f^*(Q \supset R) \vdash f^*P \wedge f^*Q}\ \mathsf{P}\supset$$

$$\frac{f \equiv (a, b \mapsto a) \quad \mathbf{O}[a, b; P] \quad \mathbf{O}[a; Q]}{a; \exists_f(P \wedge f^*Q) \vdash (\exists_f P) \wedge Q}\ \mathsf{Frob}^{\natural} \qquad \frac{f \equiv (a, b \mapsto a) \quad \mathbf{O}[a, b; P] \quad \mathbf{O}[a; Q]}{a; \exists_f(P \wedge f^*Q) \vdash (\exists_f P) \wedge Q}\ \mathsf{Frob}$$

$$\dfrac{\dfrac{\dfrac{\mathbf{O}[a; P] \quad \mathbf{O}[a; Q]}{a; P \wedge Q \vdash P}\ \pi}{a, b; f^*(P \wedge Q) \vdash f^*P}\ f^* \qquad \dfrac{\dfrac{\mathbf{O}[a; P] \quad \mathbf{O}[a; Q]}{a; P \wedge Q \vdash Q}\ \pi'}{a, b; f^*(P \wedge Q) \vdash f^*Q}\ f^*}{a, b; f^*(P \wedge Q) \vdash f^*P \wedge f^*Q}\ \langle , \rangle$$

$$\dfrac{\dfrac{f \qquad \dfrac{\dfrac{\mathbf{O}[a; Q] \quad \mathbf{O}[a; R]}{\mathbf{O}[a; Q \supset R]}\ \supset}{a; Q \supset R \vdash Q \supset R}\ \mathrm{id}}{a; (Q \supset R) \wedge Q \vdash R}\ \mathrm{Uncur}}{a, b; f^*((Q \supset R) \wedge Q) \vdash f^*R}\ f^* \qquad \dfrac{\mathbf{O}[a; Q] \quad \mathbf{O}[a; R]}{\mathbf{O}[a; Q \supset R]}\ \supset \qquad \mathbf{O}[a; Q]}{a, b; f^*(Q \supset R) \wedge f^*Q \vdash a, b; f^*((Q \supset R) \wedge Q)}\ \mathsf{P}\wedge \quad ;$$

$$\frac{a, b; f^*(Q \supset R) \wedge f^*Q \vdash f^*R}{a, b; f^*(Q \supset R) \vdash f^*Q \wedge f^*R}\ \mathrm{Cur}$$

$$\dfrac{\dfrac{\dfrac{\mathbf{O}[a,b;P] \quad \dfrac{\mathbf{O}[a;Q]}{\mathbf{O}[a,b;f^*Q]}\,f^*}{a,b;P\wedge f^*Q\vdash P}\,\pi}{a;\exists_f(P\wedge f^*Q)\vdash \exists_f P}\,\exists_f \qquad \dfrac{\dfrac{\mathbf{O}[a,b;P] \quad \dfrac{\mathbf{O}[a;Q]}{\mathbf{O}[a,b;f^*Q]}\,f^*}{a,b;P\wedge f^*Q\vdash f^*Q}\,\pi'}{a,b;\exists_f(P\wedge f^*Q)\vdash Q}\,\exists_f{}^\flat}{a;\exists_f(P\wedge f^*Q)\vdash (\exists_f P)\wedge Q}\,\langle,\rangle$$

2.16.8 Adjoints to Change-of-Base Functors

Chapter 3

Downcasing Types

3.1 Simple Proofs

In the BHK interpretation propositions P, Q R are seen as the sets of their proofs, and a proof of $P \wedge Q$ is a pair made of a proof of P and a proof of Q and a proof of $P \supset Q$ is a function that receives proofs of P and returns proofs of Q.

As P, Q and R are sets the default choices for names of variables ranging over them are p, q and r. Let's forget temporarily the syntactical distinctions between variables and terms, and let their names also stand for names when needed; also, let's allow "long names": "p, q" will be our default choice of name for a term or variable whose type is $P \times Q = P \wedge Q$, "$p \mapsto q$" the default for a term or variable in $P \to Q = Q^P = P \supset Q$. We call the passage from $P \wedge Q \to R$ to $p, q \mapsto r$ "downcasing", and the opposite direction "uppercasing".

The tree at the left below is a proof of $Q \supset R \vdash P \wedge Q \supset P \wedge R$ in Natural Deduction; the tree at the right is its downcasing,

$$\dfrac{\dfrac{\dfrac{[P \wedge Q]^1}{P} \quad \dfrac{\dfrac{[P \wedge Q]^1}{Q} \quad Q \supset R}{R}}{P \wedge R}}{P \wedge Q \supset P \wedge R}\ 1 \qquad \dfrac{\dfrac{\dfrac{[p, q]^1}{p} \quad \dfrac{\dfrac{[p, q]^1}{q} \quad q \mapsto r}{r}}{p, r}}{p, q \mapsto p, r}\ 1$$

and we may interpret each of its bars in a way that parallels the logic: for example,

$$\dfrac{Q \quad Q \supset R}{R}$$

can be read as: "if we know that Q and that $Q \supset R$ (i.e., that both Q and $Q \supset R$ are true) then we know that R (is true too)"; in the downcased tree, this is: "if we know q and $q \mapsto r$ (in the sense that we have values/definitions/meanings for these terms, i.e., we know that their corresponding types are inhabited), then

we know r (we have a "natural definition" for an $r : R$ in terms q and $q \mapsto r$, and so we know that R is inhabited)".

The downcased tree induces these definitions:

$$\begin{array}{rcl} p & := & \pi\,(p,q) \\ q & := & \pi'\,(p,q) \\ r & := & (q \mapsto r)\,q \\ p,r & := & \langle p,r\rangle \\ p,q \mapsto p,r & := & \lambda(p,q){:}(P \wedge Q).\langle p,r\rangle \end{array}$$

and we can formalize this in Coq as:

(...)

The "⟪...⟫"s above are long names, in their ascii form. A simple preprocessor reads files with these "⟪...⟫"s and converts them to tokens that Coq can treat, like:

(...)

The same translator generates another kind of output, in which a few ascii abbreviations within "⟪...⟫"s are expanded, — '`|->`' becomes '`\mapsto `', etc; we can add more, and the result of these expansions is treated as LaTeX input in mathematical mode, and the rest is typeset in verbatim mode. By LaTeXing this output we get:

(...)

We will use this pretty-printed representation in this paper. Note that we can put arbitrary TeX code in long names — even macros that generate diagrams.

If A, B, C are sets, a similar downcasing interprets:

$$\dfrac{\dfrac{\dfrac{[a,b]^1}{a} \quad \dfrac{\dfrac{[a,b]^1}{b} \quad b \mapsto c}{c}}{a,c}}{a,b \mapsto a,c}\,1 \qquad \dfrac{\dfrac{\dfrac{[p : A \times B]^1}{\pi p : A} \quad \dfrac{\dfrac{[p : A \times B]^1}{\pi' p : B} \quad f : B \to C}{f(\pi' p) : C}}{\langle \pi p, f(\pi' p)\rangle : A \times C}}{\lambda p : A \times B.\langle \pi p, f(\pi' p)\rangle : A \times B \to A \times C}$$

3.2 Functors

Now fix a set A. The functor $(A\times) : \mathbf{Set} \to \mathbf{Set}$ takes each set B to the set $A \times B$, and each function $f : B \to C$ to $(A\times)f : A \times B \to A \times C$, where $(A\times)f = \lambda p : A \times B.\langle \pi p, f(\pi' p)\rangle$.

We downcase the diagram of the functor,

$$\begin{array}{ccc} B & \overset{A\times}{\longmapsto} & A \times B \\ \Big\downarrow f & \longmapsto & \Big\downarrow (A\times)f := \lambda p{:}A\times B.\langle \pi p, f(\pi' p)\rangle \\ C & \underset{A\times}{\longmapsto} & A \times C \end{array}$$

as:

$$\begin{array}{ccc} b & \Longrightarrow & a, b \\ \Big\downarrow & \longmapsto & \Big\downarrow \\ c & \Longrightarrow & a, c \end{array}$$

We use the arrow '$\Rightarrow$' for functors to suggest that functors have two actions — one on objects, one on morphisms.

It is also useful to think that a functor $b \Rightarrow a, b$ has a "syntactical action", which is to prepend an 'a,' to the names of objects. The functor $b \Rightarrow a, b$ takes the "space of 'b's" to the "space of 'a, b's", and takes each arrow $b \mapsto c$ to an arrow $a, b \mapsto a, c$ (here we applied its syntactical action to both the source and the target).

3.3 Categories

Now let A, B, C be objects in a category $\mathbf{C}$ — which no longer needs to be **Set**. If $\mathbf{C}$ has products, we can speak of objects $A \times B$ and $A \times C$, and now we have:

$$b \mapsto c : B \to C = \mathrm{Hom}_{\mathbf{C}}(B, C)$$
$$a, b \mapsto a, c : A \times B \to A \times C = \mathrm{Hom}_{\mathbf{C}}(B, C)$$

We still have a construction for $a, b \to a, c$ starting from $b \mapsto c$, but now it's a different one:

$$\dfrac{\dfrac{A \quad B}{a, b \mapsto a}\ \pi \qquad \dfrac{\dfrac{A \quad B}{a, b \mapsto b}\ \pi' \qquad b \mapsto c}{a, b \mapsto c}}{a, b \mapsto a, c}\ ;$$

that uppercases to:

$$\dfrac{\dfrac{A \quad B}{\pi_{AB} : A \times B \to A} \qquad \dfrac{\dfrac{A \quad B}{\pi'_{AB} : A \times B \to B} \qquad f : B \to C}{\pi'_{AB}; f : A \times B \to C}}{A \times f := \langle \pi_{AB}, \pi'_{AB}; f\rangle : A \times B \to A \times C}$$

Note that if someone says "take *the* functor whose downcasing is $b \Rightarrow a, b$" then we just have to understand its action on objects (which is $A\times$) and its

action on morphisms (that we have just constructed, in slightly different ways, in **Set** and in **C**); the other person is assuring us, through the "the", that these actions exist, are not too hard to find, are functorial, and are well-defined; if we were just conjecturing that a functor named "$b \Rightarrow a, b$" should exist we would have to do all the steps. And in the syntactical world some of the steps — functoriality, well-definedness — are simply not relevant; the introduction of the syntactical world permits stating results that are mathematical precise and coherent doing only a fraction of the work that is needed for results in the real world — and proving functorialities, naturalities, etc, can be seen as a separate step.

3.4 Pseudopoints

If **C** is an arbitrary category with finite products and A, B, C are objects of **C** then we have clear meanings for $b \mapsto c$ and $a, b \mapsto a, c$ — as seen above — but what are "b", "a, b", etc? Does the typing $a : A$ mean something?

One solution is to say that 'b', 'a, b', etc are "pseudopoints", and treat them as syntactical devices that we can use in building larger expressions, which do have meaning (or: "have semantics" — $b \mapsto c$ and $a, b \mapsto a, c$ have semantics because they can appear in the Coq code as variables, constants or terms).

In the case where A, B C were sets we could say:

"a is an element of A",
"A is the space of 'a's",
"an 'a, b' is a pair made of an 'a' and a 'b' ",
"$A \times B$ is the space of 'a, b's",
"a $b \mapsto c$ is a function that takes 'b's to 'c's"
(Note that we tend to use indefinite articles when we speak of downcasings!)

but when A, B and C are objects of **C** it is better to say:

"a $b \mapsto c$ is a morphism from B to C",
"$\mathrm{Hom}_{\mathbf{C}}(B, C)$ is the space of '$b \mapsto c$'s",
"B is the object of 'b's",
"$B \times C$ is the object of 'b, c's",
"a b, c is a pseudopoint of $B \times C$".

So the idea of "pseudopoints" exists only to let us have a name for the syntactical function of the 'a, b' and the 'a, c' in a morphism $a, b \mapsto a, c$.

Pseudopoints may have a concrete semantics in the archetypal category that motivates certain kinds of categories; but that will be treated in section ___.

Here's one example in which pseudopoints have no reasonable semantics. Let A^{op} and B^{op} be objects of $\mathbf{Set}^o p$; an arrow $A^{\mathrm{op}} \to B^{\mathrm{op}}$ is an arrow $B \to A$ in disguise. If we downcase $f : A^{\mathrm{op}} \to B^{\mathrm{op}}$ to $a^{\mathrm{op}} \mapsto b^{\mathrm{op}}$, then what is an a^{op}?

3.5 Natural Transformations

3.6 Contexts

3.6.1 Single Hypothesis: CCs

In a cartesian category we have a natural way to interpret judgments involving only pairings and projections, like, for example,

$$p : A \times B, q : A \times C| - \langle\langle \pi q, \pi p\rangle, \langle \pi' p, \pi' p\rangle\rangle : (A \times A) \times (B \times B)$$

They become clearer with our long names and dictionaries:

$$(a, b) : A \times B, (a', c) : A \times C| - ((a', a), (b, b))$$

$$\begin{array}{lcl@{\qquad\qquad}rcl} p & \equiv & a, b & a & := & \pi\,(a, b) \\ q & \equiv & a', c & a' & := & \pi\,(a', c) \\ & & & b & := & \pi'\,(a, b) \\ & & & a', a & := & \langle a', a\rangle \\ & & & b, b & := & \langle b, b\rangle \\ & & & (a', a), (b, b) & := & \langle (a', a), (b, b)\rangle \end{array}$$

We can omit the types, as they can be reconstructed from the downcasings. We can reduce the judgment above to:

$$(a, b), (a', c) \vdash (a', a), (b, b)$$

A standard trick in Categorical Semantics is to represent the context as the product of the types of its variables, and the '$\vdash$' as a morphism from that product to the type of the result. So our judgment becomes this:

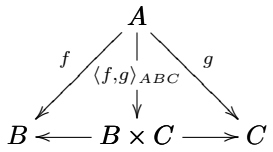
$$\begin{array}{rcl} (A \times B) \times (A \times C) & \to & (A \times A) \times (B \times B) \\ (a, b), (a', c) & \mapsto & (a', a), (b, b) \\ & & \text{where} \\ & & (a, b) := \pi\,((a, b), (a', c)) \\ & & (a', c) := \pi'\,((a, b), (a', c)) \\ & & \text{etc...} \end{array}$$

In a cartesian category we have projection maps,

$$A \times B \xleftarrow{\pi_{(A\times B)(C\times D)}} (A \times B) \times (C \times D) \xrightarrow{\pi'_{(A\times B)(C\times D)}} C \times D$$

$$A \xleftarrow{\pi_{AB}} A \times B \xrightarrow{\pi'_{AB}} B \qquad\qquad A \xleftarrow{\pi_{AC}} A \times C \xrightarrow{\pi'_{AC}} C$$

and factorizations through products, like:



The construction for the map $(A \times B) \times (A \times C) \to (A \times A) \times (B \times B)$ can be extracted from these trees:

$$\dfrac{\dfrac{}{(a,b),(a',c) \vdash a,b}\,\pi_{(A\times B)(A\times C)} \quad \dfrac{}{a,b \vdash a}\,\pi_{AB}}{(a,b),(a',c) \vdash a};$$

$$\dfrac{\dfrac{}{(a,b),(a',c) \vdash a',c}\,\pi'_{(A\times B)(A\times C)} \quad \dfrac{}{a',c \vdash a'}\,\pi_{AC}}{(a,b),(a',c) \vdash a'};$$

$$\dfrac{\dfrac{}{(a,b),(a',c) \vdash a,b}\,\pi_{(A\times B)(A\times C)} \quad \dfrac{}{a,b \vdash b}\,\pi'_{AB}}{(a,b),(a',c) \vdash b};$$

$$\dfrac{\dfrac{\dfrac{}{(a,b),(a',c) \vdash a'}}{(a,b),(a',c) \vdash a',a}\quad \dfrac{\dfrac{}{(a,b),(a',c) \vdash a}}{(a,b),(a',c) \vdash a',a}}{(a,b),(a',c) \vdash a',a}\,\langle,\rangle_{((A\times B)\times(A\times C))AA}$$

$$\dfrac{\dfrac{\dfrac{}{(a,b),(a',c) \vdash b}}{(a,b),(a',c) \vdash b,b}\quad \dfrac{\dfrac{}{(a,b),(a',c) \vdash b}}{(a,b),(a',c) \vdash b,b}}{(a,b),(a',c) \vdash b,b}\,\langle,\rangle_{((A\times B)\times(A\times C))BB}$$

$$\dfrac{\dfrac{\dfrac{}{(a,b),(a',c) \vdash a',a}}{(a,b),(a',c) \vdash (a',a),(b,b)}\quad \dfrac{\dfrac{}{(a,b),(a',c) \vdash b,b}}{(a,b),(a',c) \vdash (a',a),(b,b)}}{(a,b),(a',c) \vdash (a',a),(b,b)}\,\langle,\rangle_{((A\times B)\times(A\times C))(A\times A)(B\times B)}$$

The final result is this, if we omit the typings in the '$\pi_{\_\_}$'s, '$\pi'_{\_\_}$'s and '$\langle,\rangle_{\_\_\_}$'s to make it smaller:

$$\begin{array}{rcl} \langle\langle\pi';\pi,\pi;\pi'\rangle,\langle\pi';\pi,\pi;\pi'\rangle\rangle : (A\times B)\times(A\times C) & \to & (A\times A)\times(B\times B) \\ ((a,b),(a',c)) & \mapsto & ((a',a),(b,b)) \end{array}$$

What matters here is: with the downcasings and the operations in a protocartesian category — projections, pairing — we can build the categorical interpretations of judgments like

$$\begin{array}{rcl} (a',a) : (A\times B),(b,b) : (A\times C) & \vdash & ((a',a),(b,b)) : (A\times A)\times(B\times B) \\ (a,b),(a',c) & \vdash & (a',a),(b,b) \end{array}$$

These categorical constructions behave as expected. For example, the composite of two "flip" functions,

$$\dfrac{\dfrac{}{a,a' \vdash a',a}\quad \dfrac{}{a',a \vdash a,a'}}{a,a' \vdash a,a'}$$

is the identity — but the proof of this lives in the real world.

3.6.2 Context Morphisms

3.6.3 Several Hypotheses: CCs (With a Trick)

We can also construct the term for $(a,b),(a',c) \vdash (a',a),(b,b)$ from a tree in Natural Deduction-style, to which we add the contexts:

$$\dfrac{\dfrac{a',c}{a'}\,\pi \quad \dfrac{a,b}{a}\,\pi}{a',a}\,\langle,\rangle \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\overline{(a',c)\vdash a',c}\ \text{id}}{(a',c)\vdash a'}\ ;\pi}{(a,b),(a',c)\vdash a'}\ \pi'; \quad \dfrac{\dfrac{\overline{(a,b)\vdash a,b}\ \text{id}}{(a,b)\vdash a}\ ;\pi}{(a,b),(a',c)\vdash a}\ \pi;}{(a,b),(a',c)\vdash a',a}\ \langle,\rangle$$

(*Note:* think of trees with judgments at the nodes as being in "sequent calculus style"... so for us "Natural Deduction" is when the contexts are hidden, and "sequent calculus" is when they are explicit.)

This is easy to do when all the nodes have exactly the same single hypothesis — we would have that in the case $\gamma : (A \times B) \times (A \times C) \vdash \ldots$, but not in the case $p : (A \times B), q : (A \times C) \vdash \ldots$, where we have some cases where contexts have to be merged; a nice way to handle the mergings is to represent them as extra steps in the tree of judgments:

$$\dfrac{\dfrac{a',c}{a'}\,\pi \quad \dfrac{a,b}{a}\,\pi}{a',a}\,\langle,\rangle \qquad \dfrac{\dfrac{\dfrac{\overline{(a',c)\vdash a',c}\ \text{id}}{(a',c)\vdash a'}\ ;\pi}{(a,b),(a',c)\vdash a'}\ \pi'; \quad \dfrac{\dfrac{\overline{(a,b)\vdash a,b}\ \text{id}}{(a,b)\vdash a}\ ;\pi}{(a,b),(a',c)\vdash a}\ \pi;}{(a,b),(a',c)\vdash a',a}\ \langle,\rangle$$

The order of the variables in the context is immaterial, in a sense — the ND tree at the left above can also be expanded to a sequent calculus "proof" (i.e., construction) of $(a',c),(a,b) \vdash a',a$,

[But one can be obtained from the other one by composing with (iso)morphisms that scramble the variables in the context...]

[Explain the convention for parentheses in the context: $(a,b),(a',c) \vdash \ldots$ is $(a,b) : A \times B, (a',c) : A \times C \vdash \ldots$, $((a,b),(a',c)) \vdash \ldots$ is $((a,b),(a',c)) : (A \times B) \times (AxC) \vdash \ldots$, and $a,b,a',c \vdash \ldots$ is $a : A, b : B, a' : A, c : C \vdash \ldots$.]

[The internal language: examples]

[Adding $+$ and $\cdot$ to the language with morphisms $+ : A \times A \to A$ and $\cdot : A \times A \to A$, that we downcase to $a,a' \mapsto a + a$ and $a,a' \mapsto aa'$; the downcasing of the composite of $x,y \mapsto xy, y$ and $z,w \mapsto z + w$ is $x,y \mapsto xy + y$]

3.6.4 Discharges: CCCs

3.7 Adjunctions

3.8 Currying Functors

If $\mathbb{A}$, $\mathbb{B}$, and $\mathbb{C}$ and categories, then we can form the product category $\mathbb{A} \times \mathbb{B}$ and the category of functors $\mathbb{B} \to \mathbb{C}$. It turns out that for any fixed category $\mathbb{B}$, $(\times\mathbb{B}) : \mathbf{Cat} \to \mathbf{Cat}$ and $(\mathbb{B} \to) : \mathbf{Cat} \to \mathbf{Cat}$ are functors — this is easy to prove — and we have $(\times\mathbb{B}) \dashv (\mathbb{B} \to)$; the adjunction part is left as an exercise in most basic Category Theory books, and its proof is quite hairy. We will look at its syntactical skeleton — which is just the construction of the two transpositions, $\flat$ and $\sharp$. The figure is:

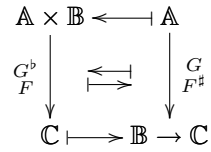
$$\begin{array}{ccc} \mathbb{A} \times \mathbb{B} & \longleftarrow\!\dashv & \mathbb{A} \\ {\scriptstyle G^{\flat}_{F}}\downarrow & \dashv\!\longrightarrow & \downarrow{\scriptstyle {}^{G}_{F^{\sharp}}} \\ \mathbb{C} & \longmapsto \mathbb{B} \to \mathbb{C} & \end{array}$$

In the syntactical world the '$\sharp$' takes each protofunctor $F \equiv (F_0, F_1) : \mathbb{A} \times \mathbb{B} \to \mathbb{C}$ and produces a protofunctor $F^\sharp \equiv (F^\sharp{}_0, F^\sharp{}_1) : \mathbb{A} \to (\mathbb{B} \to \mathbb{C})$. The action of $F^\sharp$ on morphisms, $F^\sharp{}_0$, takes each object A of $\mathbb{A}$ to an object $F^\sharp{}_0 A \equiv F^\sharp A$ of $\mathbb{B} \to \mathbb{C}$; this $F^\sharp A : \mathbb{B} \to \mathbb{C}$ is itself a (proto)functor, and so it is made of two components: $F^\sharp A \equiv ((F^\sharp A)_0, (F^\sharp A)_1)$. The action of $F^\sharp$ on morphisms, $F^\sharp{}_1$, takes each morphism $\alpha : A \to A'$ of $\mathbb{A}$ to a morphism $F^\sharp{}_1 \alpha \equiv F^\sharp \alpha : F^\sharp A \to F^\sharp A'$ between two functors $F^\sharp A, F^\sharp A' : \mathbb{B} \to \mathbb{C}$. This is a natural transformation, that we can write as $B \dot{\to} (F^\sharp A B \overset{F^\sharp \alpha B}{\to} F^\sharp A' B))$. The full construction is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[A]^3 \quad [B]^1}{(A,B)}\,\langle,\rangle \qquad F}{F^\sharp A B \equiv F(A,B)} \Rightarrow E_0}{(F^\sharp A)_0 \equiv \lambda B.F(A,B)}\,1 \qquad \dfrac{\dfrac{\dfrac{[A]^3}{\mathrm{id}_A}\,\mathrm{id} \quad [\beta]^2}{(\mathrm{id}_A,\beta)} \qquad F}{F^\sharp A \beta \equiv F(\mathrm{id}_A,\beta)} \Rightarrow E_1}{(F^\sharp A)_1 \equiv \lambda\beta.F(\mathrm{id}_A,\beta)}\,2\,\langle,\rangle}{F^\sharp A \equiv (\lambda B.F(A,B), \lambda\beta.F(\mathrm{id}_A,\beta))}}{F^\sharp{}_0 \equiv \lambda A.(\lambda B.F(A,B), \lambda\beta.F(\mathrm{id}_A,\beta))}\,3 \qquad \dfrac{\dfrac{\dfrac{\dfrac{[\alpha]^5 \quad \dfrac{[B]^4}{\mathrm{id}_B}}{(\alpha,\mathrm{id}_B)} \qquad F}{F^\sharp \alpha B \equiv F(\alpha,\mathrm{id}_B)} \Rightarrow E_1}{F^\sharp \alpha \equiv \lambda B.F(\alpha,\mathrm{id}_B)}\,4}{F^\sharp{}_1 \equiv \lambda\alpha.\lambda B.F(\alpha,\mathrm{id}_B)}\,5}{F^\sharp := (\lambda A.(\lambda B.F(A,B), \lambda\beta.F(\mathrm{id}_A,\beta)), \lambda\alpha.\lambda B.F(\alpha,\mathrm{id}_B))}\,\langle,\rangle$$

but how could anyone have arrived at this?... and how could anyone check whether this construction is right?

One possible answer is: "start from the types".



$$\dfrac{\dfrac{(A,B)}{B}\ \pi' \qquad \dfrac{\dfrac{(A,B)}{A}\ \pi \qquad a \Rightarrow (b \Rightarrow c)}{b \Rightarrow c} \Rightarrow E_0}{C} \Rightarrow E_0 \qquad \dfrac{\dfrac{(A,B)}{B}\ \pi' \qquad \dfrac{\dfrac{(A,B)}{A}\ \pi \qquad G}{GA} \Rightarrow E_0}{G^\flat(A,B) \equiv GAB} \Rightarrow E_0$$

$$\dfrac{\dfrac{\dfrac{(\alpha,\beta)}{\beta}\ \pi'}{b \mapsto b'}\ \text{ren} \qquad \dfrac{\dfrac{(\alpha,\beta)}{\alpha}\ \pi}{A}\ \mathbf{src} \qquad \dfrac{G}{a \Rightarrow (b \Rightarrow (a;b)^F)}\ \text{ren}}{(a;b)^F \mapsto (a;b')^F} \Rightarrow E_1 \qquad \dfrac{\dfrac{(\alpha,\beta)}{\beta}\ \pi'}{B'}\ \mathbf{tgt} \qquad \dfrac{\dfrac{\dfrac{(\alpha,\beta)}{\alpha}\ \pi}{a \mapsto a'}\ \text{ren} \qquad \dfrac{G}{a \Rightarrow (b \Rightarrow (a;b)^F)}\ \text{ren}}{(b \Rightarrow (a;b)^F) \mapsto (b \Rightarrow (a';b)^F)} \Rightarrow E_1}{b \xrightarrow{\bullet} ((a;b)^F \mapsto (a';b)^F)}\ \text{ren}}{(a;b')^F \mapsto (a';b')^F} \xrightarrow{\bullet} E$$
$$;$$
$$(a;b)^F \mapsto (a';b')^F$$

$$\dfrac{\dfrac{(\alpha,\beta)}{\beta}\ \pi' \qquad \dfrac{\dfrac{(\alpha,\beta)}{\alpha}\ \pi}{A}\ \mathbf{src} \qquad \dfrac{G}{GA} \Rightarrow E_0}{GA\beta} \Rightarrow E_1 \qquad \dfrac{\dfrac{(\alpha,\beta)}{\beta}\ \pi'}{B'}\ \mathbf{tgt} \qquad \dfrac{\alpha \qquad \dfrac{(\alpha,\beta)}{G}\ \pi}{G\alpha} \Rightarrow E_1}{G\alpha B'} \xrightarrow{\bullet} E$$
$$;$$
$$G^\flat(\alpha,\beta) \equiv GA\beta; G\alpha B'$$

$$\dfrac{\dfrac{\dfrac{[(A,B)]^1}{B}\ \pi' \qquad \dfrac{\dfrac{[(A,B)]^1}{A}\ \pi \qquad G}{GA} \Rightarrow E_0}{G^\flat(A,B) \equiv GAB}}{G^\flat{}_0 \equiv \lambda(A,B).GAB}\ 1 \qquad \dfrac{\dfrac{\dfrac{[(\alpha,\beta)]^2}{\beta}\ \pi' \qquad \dfrac{\dfrac{[(\alpha,\beta)]^2}{\alpha}\ \pi}{A}\ \mathbf{src} \qquad \dfrac{G}{GA} \Rightarrow E_0}{GA\beta} \Rightarrow E_1 \qquad \dfrac{\dfrac{[(\alpha,\beta)]^2}{\beta}\ \pi'}{B'}\ \mathbf{tgt} \qquad \dfrac{\alpha \qquad \dfrac{[(\alpha,\beta)]^2}{G}\ \pi}{G\alpha} \Rightarrow E_1}{G\alpha B'} \xrightarrow{\bullet} E$$
$$;$$
$$\dfrac{G^\flat(\alpha,\beta) \equiv GA\beta; G\alpha B'}{G^\flat{}_1 \equiv \lambda(\alpha,\beta).GA\beta; G\alpha B'}\ 2$$
$$G^\flat \equiv (\lambda(A,B).GAB, \lambda(\alpha,\beta).GA\beta; G\alpha B')$$

3.9 Subsets

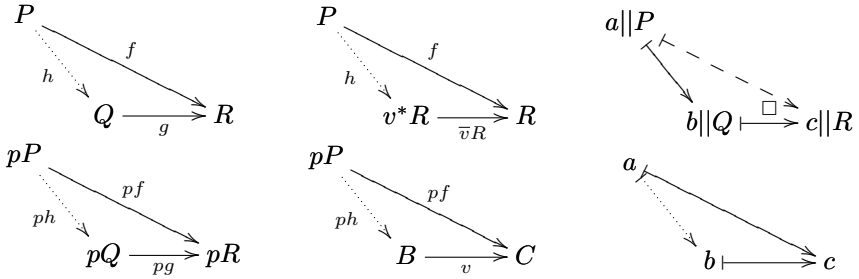
3.10 Subobjects

3.11 Projection Functors

A *prefibration* is just a functor $p : \mathbb{E} \to \mathbb{B}$ — a "projection functor" going from the "entire category" to the "base category". A *fibration* is a prefibration with enough "cartesian morphisms" in $\mathbb{E}$, as we will see.

We say that an object X of $\mathbb{E}$ is "above" its image pX, and the same for morphisms. Usually when we draw a morphism $X \xrightarrow{f} Y$ of $\mathbb{E}$ above a morphism $I \xrightarrow{u} J$ of $\mathbb{B}$ this means that $u = pf : pX \to pY$. For each object I of $\mathbb{B}$ we call the subcategory $p^{-1}(I)$ of $\mathbb{E}$ the *fiber above* (or "*of*") I, and we write $\mathbb{E}_I$ for $p^{-1}(I)$.

Each object Z of $\mathbb{E}$ induces a functor $p/Z : \mathbb{E}/Z \to \mathbb{B}/pZ$ between slice categories. We say that a morphism $g : Y \to Z$ of $\mathbb{E}$ is *cartesian* when for each object $f : X \to Y$ of $\mathbb{E}/Z$ the mapping $\mathrm{Hom}_{\mathbb{E}/Z}(f, g) \to \mathrm{Hom}_{\mathbb{B}/pZ}(pf, pg)$ — that takes each $h : X \to Y$ to $ph : pX \to pY$ — is a bijection.



This can be made much clearer. Our archetypical fibration is cod : Sub(**Set**) → **Set** [that we sometimes call just Sub(**Set**)]. An object of Sub(**Set**) is a monic $A' \rightarrowtail A$, and $\mathrm{cod}(A' \rightarrowtail A)$ is its codomain, A. Let's restrict our attention to subobjects where A' is a subset of A and '$\rightarrowtail$' is the inclusion; better yet, let's regard A' as the set of the 'a's in A that obey some property P. So our subobject $A' \rightarrowtail A$ becomes:

$$\{\, a \in A \mid P(a) \,\} \rightarrowtail A$$

Now let's contract it further, by the use of a double vertical bar in the '{ }'. This

$$\{\, a \mid\mid P(a) \,\}$$

will be our notation for the whole monic/subobject, and a morphism

$$\{\, a \mid\mid P(a) \,\} \to \{\, b \mid\mid Q(b) \,\}$$

in Sub(**Set**) is in fact a commuting square in **Set**:

$$\begin{array}{ccc} \{\, a \in A \mid P(a) \,\} & \longrightarrow & \{\, b \in B \mid Q(b) \,\} \\ \downarrow & & \downarrow \\ A & \longrightarrow & B \end{array}$$

The projection functor cod says that

$$\{\, a \mid\mid P(a) \,\} \to \{\, b \mid\mid Q(b) \,\}$$

is over $A \to B$ in the fibration $\mathrm{cod} : \mathrm{Sub}(\mathbf{Set}) \to \mathbf{Set}$:

$$\begin{array}{ccc} \{\, a \in A \mid\mid P(a) \,\} & \longrightarrow & \{\, b \in B \mid\mid Q(b) \,\} \\ & & \\ A & \longrightarrow & B \end{array} \qquad \begin{array}{c} \mathbb{E} = \mathrm{Sub}(\mathbf{Set}) \\ \downarrow{\scriptstyle p=\mathrm{cod}} \\ \mathbb{B} = \mathbf{Set} \end{array}$$

Each map $f : A \to B$ in $\mathcal{B}$ induces a "change-of-base functor", $f^* : \mathbb{E}_B \to \mathbb{E}_A$:

$$\begin{array}{ccc} \mathbb{E}_A & \xleftarrow{\;f^*\;} & \mathbb{E}_B \end{array}$$

$$\begin{array}{ccc} \{\, a \mid\mid Q(f(a)) \,\} & \xleftarrow{f^*} & \{\, b \mid\mid Q(b) \,\} \\ {\scriptstyle \forall a \in A.\, Q(f(a)) \supset Q'(f(a)) \atop \text{or} \atop a;Q(f(a)) \vdash Q'(f(a))} \downarrow & \dashv & \downarrow {\scriptstyle ba \in B.\, Q(b) \supset Q'(b) \atop \text{or} \atop b;Q(b) \vdash Q'(b)} \\ \{\, a \mid\mid Q'(f(a)) \,\} & \xleftarrow[f^*]{} & \{\, a \mid\mid Q'(b) \,\} \\ A & \xrightarrow{\;f\;} & B \end{array} \qquad \begin{array}{c} \mathbb{E} \\ \downarrow{\scriptstyle p} \\ \mathbb{B} \end{array}$$

and also a natural transformation, that produces for each object $\{\, b \mid\mid Q(b) \,\}$ over B a "horizontal" map (we will define "horizontality" and "verticality" in $\mathbb{E}$ precisely very soon), $\{\, a \mid\mid Q(f(a)) \,\} \to \{\, b \mid\mid Q(b) \,\}$, in $\mathbb{E}$. If we expand this "horizontal" map into a square in **Set** we see that what we've got is a pullback:

$$\begin{array}{c} \mathbb{E} \\ \downarrow \\ \mathbb{B} \end{array} \qquad \begin{array}{ccc} \{\, a \mid\mid Q(f(a)) \,\} & \rightleftarrows & \{\, b \mid\mid Q(b) \,\} \\ & {\scriptstyle f^*}\dashv & \\ A & \longrightarrow & B \end{array} \qquad \begin{array}{ccc} \{\, a \in A \mid Q(f(a)) \,\} & \longrightarrow & \{\, b \in B \mid Q(b) \,\} \\ \downarrow & \lrcorner \quad \prec f^* \dashv & \downarrow \\ A & \longrightarrow & B \end{array} \qquad \begin{array}{ccc} A & \longrightarrow & B \end{array}$$

It is these "maps in Sub(**Set**) that correspond to pullbacks in **Set**" that we will call "cartesian" — but the actual formal definition of cartesianness is more

abstract; it is the one using a universal property, that we saw at the beginning of this section.

Before proceeding let's see some further shorthands, and their downcasings. Let's write $\{\, c \mid\mid R(c) \,\}$ as just $\{\, c \mid R \,\}$, and $\{\, b \mid\mid R(g(b)) \,\}$ as just $\{\, b \mid R \,\}$; we will say that in a diagram

$$\{\, b \mid\mid Q \,\} \longrightarrow \{\, c \mid\mid R \,\}$$

$$B \xrightarrow{\quad g \quad} C$$

the object $\{\, b \mid\mid Q \,\}$ *deserves the name* $\{\, b \mid\mid R \,\}$ when the arrow $\{\, b \mid\mid Q \,\} \to \{\, c \mid\mid R \,\}$ is cartesian. We will downcase $\{\, c \mid\mid R \,\}$ to $c||R$, and we will drop the '$c||$' in diagrams when we can deduce it from the projections below. Also, we will write $\overline{f}$ for the natural transformation that returns cartesian morphisms, we will shrink "$\{\, c \mid\mid R \,\}$" even more to just 'R' (now we've reached the "standard" level of abstraction: R is an object over C), and we will use a '$\square$' in the downcased notation to stress that a morphism is cartesian. So:

$$\{\, b \mid\mid R(g(b)) \,\} \underset{f^*}{\overset{\overline{f}R}{\rightleftarrows}} \{\, c \mid\mid R(c) \,\} \qquad \{\, b \mid\mid R \,\} \underset{f^*}{\overset{\overline{f}R}{\rightleftarrows}} \{\, c \mid\mid R \,\} \qquad f^*R \underset{f^*}{\overset{\overline{f}R}{\rightleftarrows}} R$$

$$B \xrightarrow{\quad g \quad} C \qquad B \xrightarrow{\quad g \quad} C \qquad B \xrightarrow{\quad g \quad} C$$

$$b \mid\mid R(g(b)) \underset{\Longrightarrow}{\overset{\square}{\Longleftarrow}} c \mid\mid R(c) \qquad b \mid\mid R \underset{\Longrightarrow}{\overset{\square}{\Longleftarrow}} c \mid\mid R \qquad R \underset{\Longrightarrow}{\overset{\square}{\Longleftarrow}} R$$

$$b \longmapsto c \qquad b \longmapsto c \qquad b \longmapsto c$$

[Cartesian liftings]
[The adjunction]
[Cleavage]
[Change-of-base functors]
[$\overline{v}$ as a natural transformation]
[Vertical and horizontal; mention "prone"]
[Every morphism in $\mathbb{E}$ factors as 'vertical;horizontal']
[Archetypical case 1: cod : Sub(**Set**) → **Set**. Subobjects]
[Archetypical case 2: cod : $\mathbf{Set}^{\to} \to \mathbf{Set}$. Display maps]
[Archetypical case 3: **Set**//**Set**. Simplified display maps]

3.12 Some Uppercasings

3.12.1 A Semi-Logical Notation

3.12.2 Lawvere

3.12.3 Seely

3.12.4 Jacobs

3.12.5 Maclane

3.12.6 Awodey

[Awo06]

3.13 Change-of-base

(Transcribed from the back of p.246 in "On Property-Like Structures").

In the archetypical hyperdoctrine, Sub(**Set**) — and thus also in the downcased notation — the adjoints to weakening (π^*) and contraction (δ^*) functors take particularly simple forms, while the adjoints to arbitrary substitution functors (f^*) look more complex, and indeed can be (re)constructed from the adjoints to weakening and contraction.

This makes more sense in the diagrams. If $\pi \equiv (a, b \mapsto a)$ is a projection, its associated change-of-base functor is a "weakening" rule because its weakens the context by introducing a new hypothesis:

$$\begin{array}{ccc} a,b||Pa & \Longleftarrow\!\!= & a||Pa \\ {\scriptstyle a,b||Pa\vdash Qa}\downarrow\top & \overset{\pi^*}{\underset{\text{weakening}}{\longleftarrow\!\!\dashv}} & \downarrow\top{\scriptstyle a||Pa\vdash Qa} \\ a,b||Qa & \Longleftarrow\!\!= & a||Qa \end{array}$$

$$a, b \vdash \xrightarrow[\text{projection}]{\pi} a$$

If $\delta \equiv (a, b \overset{b' :\!= b}{\mapsto} a, b, b')$ is a "duplication" (or "diagonal"), the associated change-of-base functor is a "contraction" rule, that collapses two hypotheses of the same type into one:

$$\begin{array}{ccc} a,b||Pabb & \Longleftarrow\!\!\!= & a,b,b'||Pabb' \\ {\scriptstyle a,b||Pabb\vdash Qabb}\downarrow & \underset{\text{contraction}}{\overset{\delta^*}{\longleftarrow\!\!\shortmid}} & \downarrow{\scriptstyle a,b,b'||Pabb'\vdash Qabb'} \\ a,b||Qabb & \Longleftarrow\!\!\!= & a,b,b'||Qabb' \end{array}$$

$$a,b \vdash \xrightarrow[\substack{\text{duplication/}\\ \text{diagonal}\\ (b':=b)}]{\delta} a,b,b'$$

In the case of an arbitrary morphism, say, $f \equiv (a \mapsto b)$, we will say that the associated change of base functor, f^*, is just a "substitution":

$$\begin{array}{ccc} a||P(fa) & \Longleftarrow\!\!\!= & b||P(b) \\ {\scriptstyle a||P(fa)\vdash Q(fa)}\downarrow & \underset{\text{substitution}}{\overset{f^*}{\longleftarrow\!\!\shortmid}} & \downarrow{\scriptstyle b||P(b)\vdash Q(b)} \\ a||Q(fa) & \Longleftarrow\!\!\!= & b||Q(b) \end{array}$$

$$a \vdash \xrightarrow[b:=fa]{f} b$$

Note that we are claiming that these change-of-base functors, π^*, δ^*, f^*, deserve the names "$a,b||Qa \Leftarrow a||Qa$", "$a,b||Qabb \Leftarrow a,b,b'||Qabb'$", "$a||Q(fa) \Leftarrow b||Q(b)$"; to check that these names are adequate (in the archetype, of course!) we need to check that the implied cartesian morphisms

$$a,b||Qa \overset{\square}{\mapsto} a||Qa$$

are really cartesian, which amounts to checking that these diagrams represent adjunction:

```
a||Pa |---------------------\
     | - >                   v
          b,c||Qb |-cart?-> b||Qb

 a |-----------------------\
   | - - - >               v
           b,c |-----------> b
```

We say that a morphism $b||Q \mapsto c||R$ in $\mathbb{E}$ is *pre-cartesian* when it is "on the way to becoming cartesian"; this means that we have not yet constructed the universal condition that proves that it is cartesian, but we are going towards that.

Note that this terminology — "on the way to becoming", "not yet", "towards that" — imply that there are underlying notions of *time* and *aim*; they refer to a mathematician (or a proof assistant) who is working on a proof.

We can picture the situation as this:

```
statement
+ proof   |- - ->
                    S+P^-
          <- - -|
statement
+ witness
```

Take a statement S; for example, "$b, c||Rc \mapsto c||Rc$ is cartesian". We may have a witness, W, for "σ is true"; in our everyday practice a "witness" may be a mathematician — in flesh and bones; or a book; or an article, an e-mail, etc — who says that σ is true; in a purely mathematical world, however, a witness for σ may be something as abstract as a point in the set

$$\{\, x \in \{*\} \mid \sigma \text{ is true} \,\},$$

which is a singleton if σ is true, or the empty set if σ is false.

Starting from S and W — or even from just S, if we are brave — we may try to work towards a situation where we have even more information: $S + P$, where P is a proof of the statement S. In our example, a proof is an inverse for a certain natural map.

If you, reader, are a strictly classical mathematician, I urge you to skip the rest of this paragraph; for those who are still reading, an inverse for the natural transformation

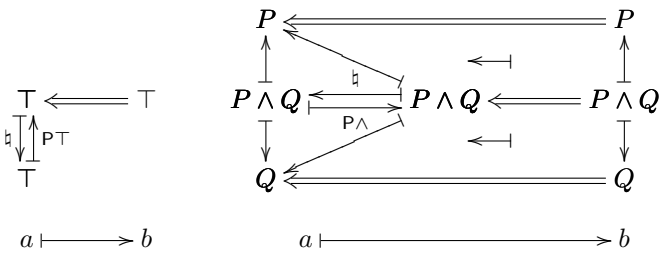
```
a||P |--------\
    |->        v
       b||Q |-> c||R

 a |----------\
   |->         v
       b |-----> c
```

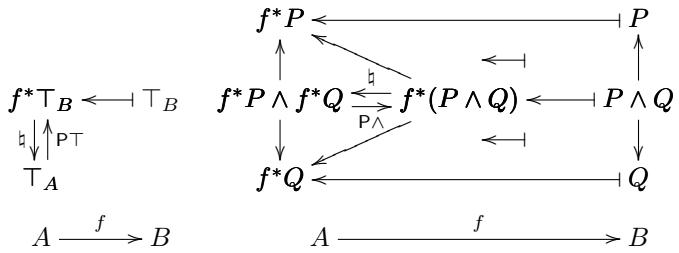
is a *construction* that receives triples $(\{\, a||P \,\}, (a||P \mapsto c||R), (a \mapsto b))$ obeying a certain commutativity condition and return the unique corresponding $a||P \mapsto b||Q$.

[And constructions are lambda-terms.]

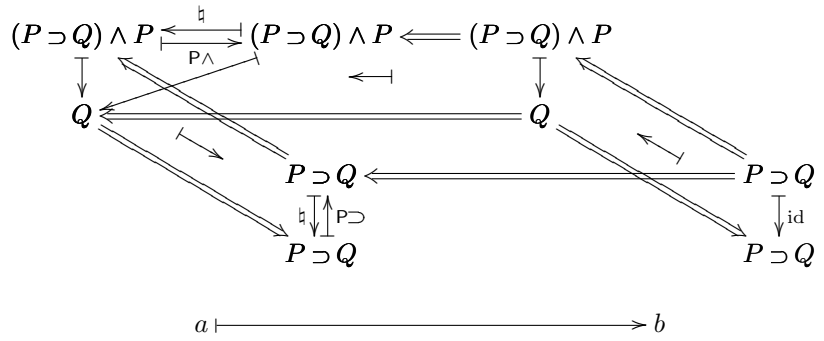
3.14 Preservations

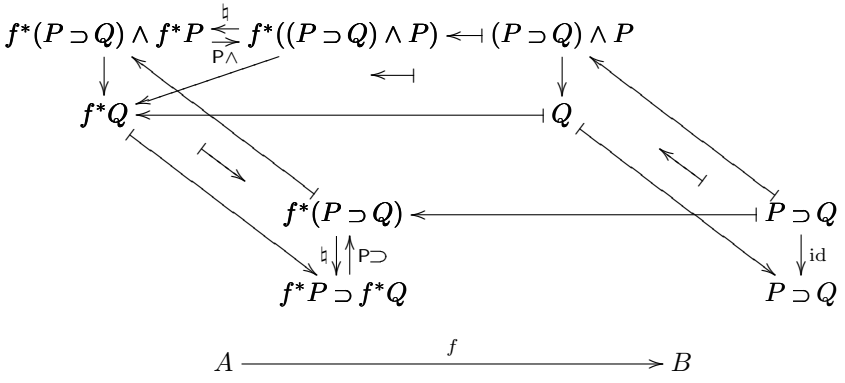
[Explain these diagrams:]





[Explain these diagrams:]



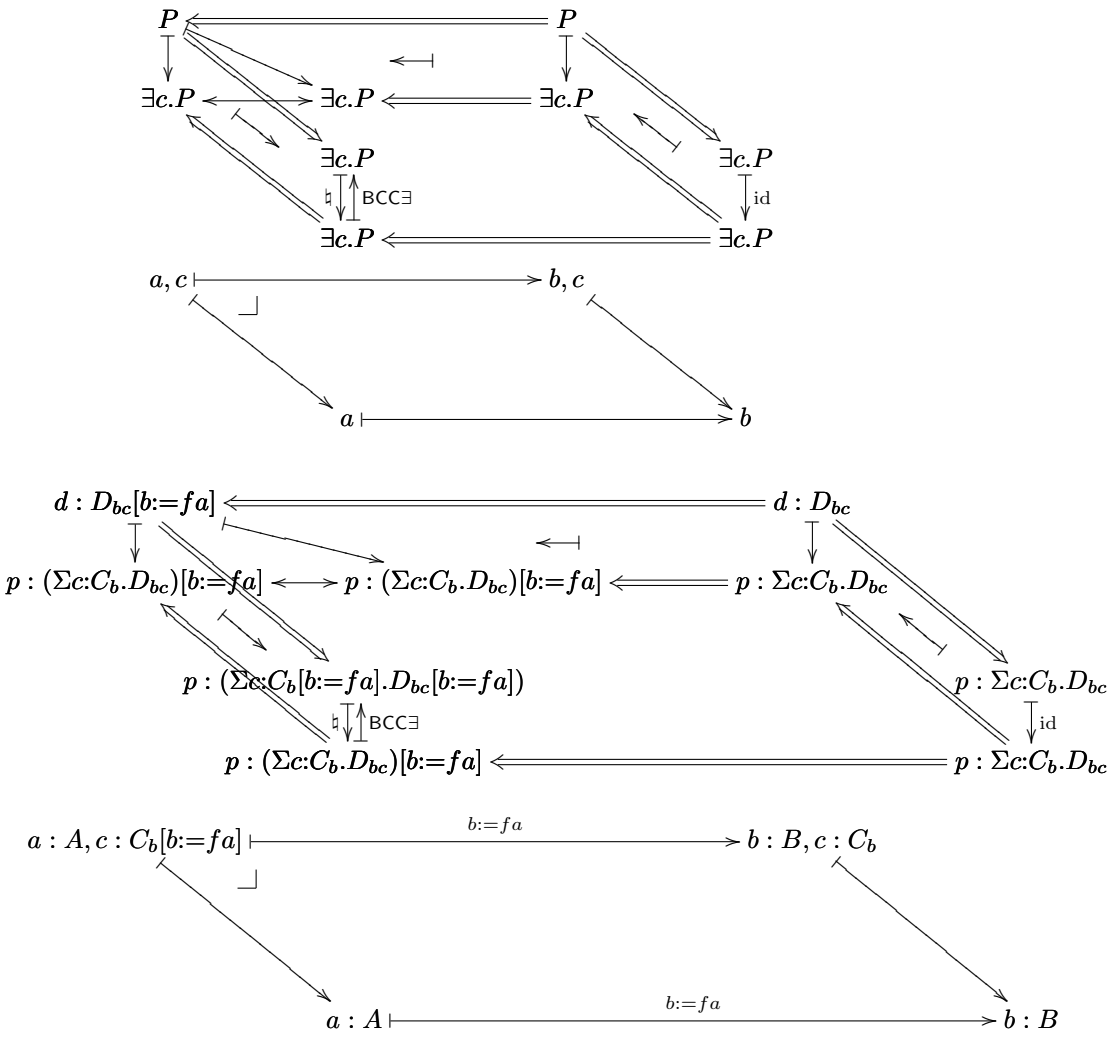


3.15 Display Maps

3.16 Quantifiers

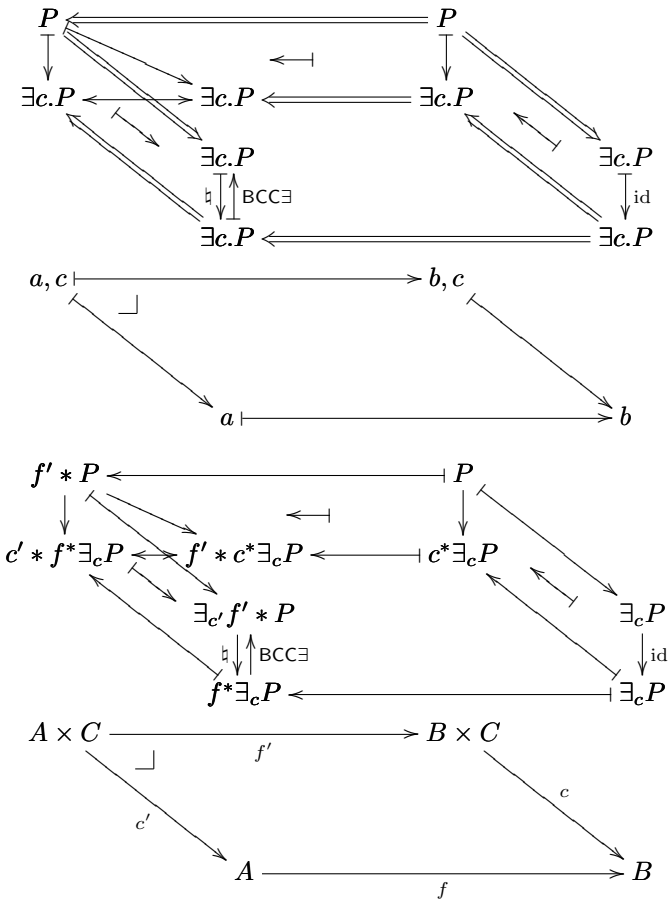
3.17 Equality

3.18 Beck-Chevalley for 'Exists'

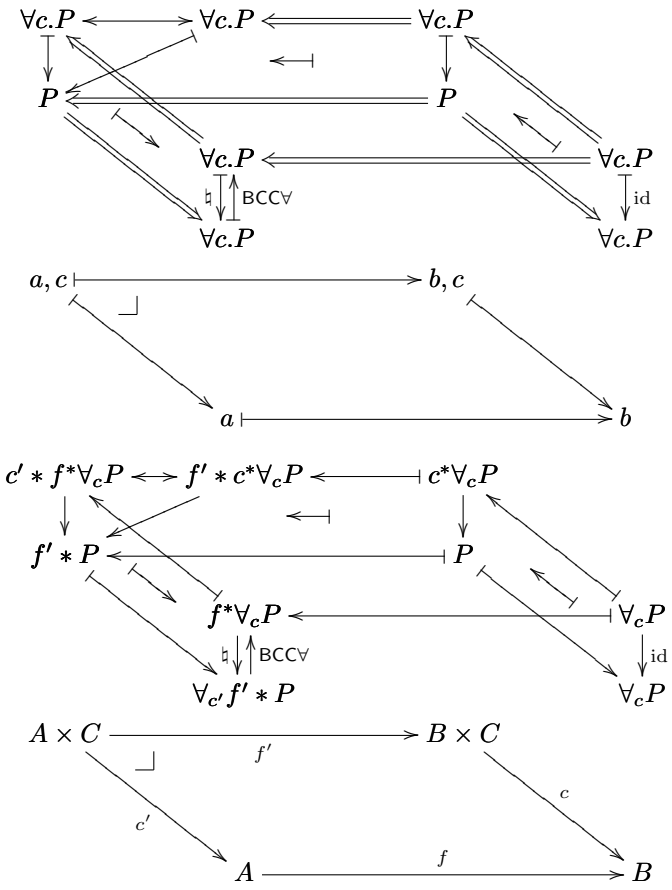


$$\dfrac{\dfrac{a : A \vdash fa : B \quad \dfrac{\dfrac{\dfrac{b : B, c : C \vdash D_b : \Theta}{b : B \vdash \Sigma c{:}C.D_b : \Theta}\,\Sigma}{b : B; p : \Sigma c{:}C_b.D_{bc} \vdash p : \Sigma c{:}C_b.D_{bc}}\,\mathrm{id}}{b : B, c : C_b; d : D_{bc} \vdash \langle c, d\rangle : \Sigma c{:}C_b.D_{bc}}\,\Sigma^\sharp}{a : A, c : C_b[b{:=}fa]; d : D_{bc}[b{:=}fa] \vdash \langle c, d\rangle[b{:=}fa] : (\Sigma c{:}C_b.D_{bc})[b{:=}fa]}\,\mathrm{cut} \quad a, c; p \vdash p}{a : A, c : C_b[b{:=}fa]; d : D_{bc}[b{:=}fa] \vdash \langle c, d\rangle[b{:=}fa] : (\Sigma c{:}C_b.D_{bc})[b{:=}fa]}\;;$$
$$\dfrac{}{a : A; p : \Sigma c{:}C_b[b{:=}fa].D_{bc}[b{:=}fa] \vdash \langle c, d\rangle[b{:=}fa][c, d := \mathrm{unp}\, p] : (\Sigma c{:}C_b.D_{bc})[b{:=}fa][c, d := \mathrm{unp}\, p]}\,\Sigma^\flat$$

[Explain these diagrams:]

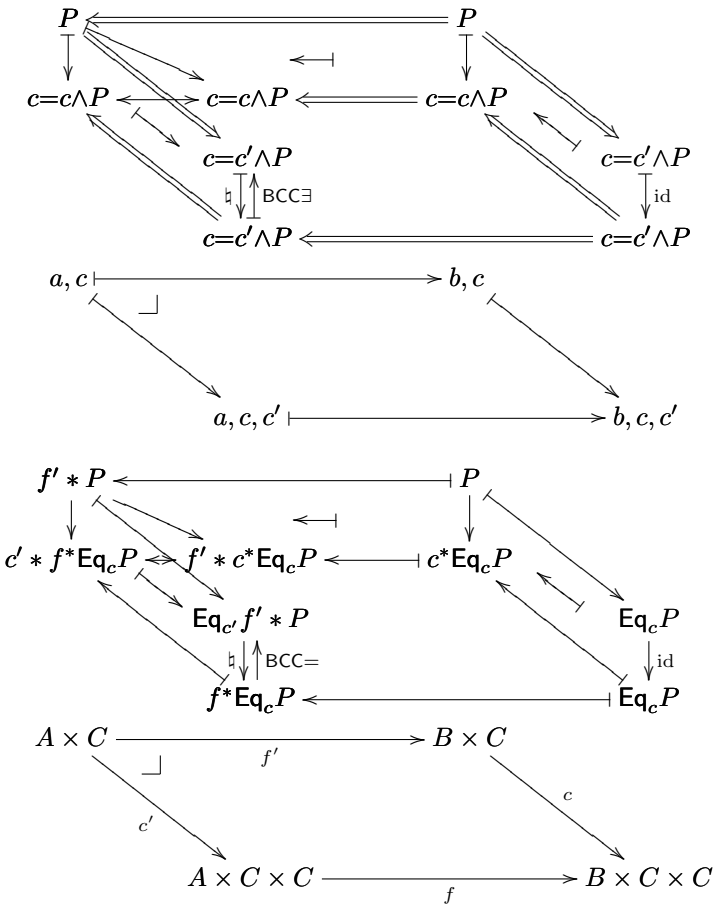


3.19 Beck-Chevalley for 'For All'



3.20 Beck-Chevalley for Equality

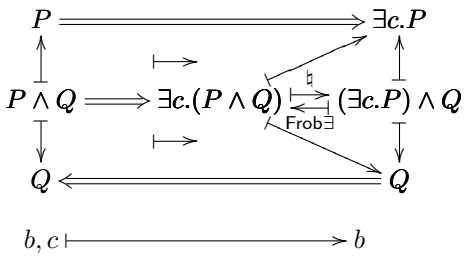
[BCC for equality is the same, categorically; point that in [Jacobs] and in [Seely-Hyp] they are kept separate, because they don't require a left adjoint to all 'f's in the base — just for projections and diagonals. The downcased diagram is different...]

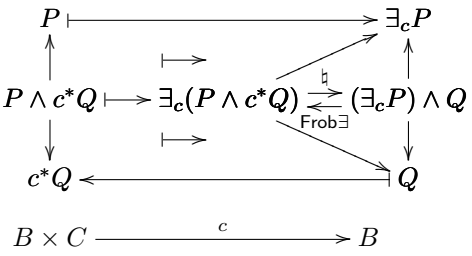


3.21 Frobenius for 'Exists'

[Explain the diagrams below, and include the diagrams that show that in the presence of a left adjoint to change-of-base we have that "Frobenius is equivalent to preservation of 'implies' ".]

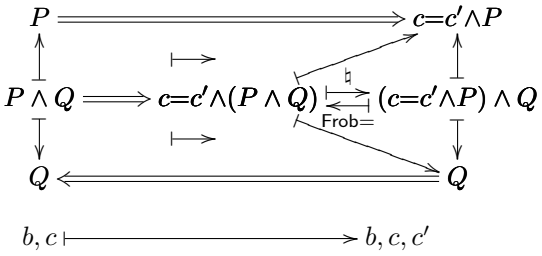
[Frobenius for exists:]

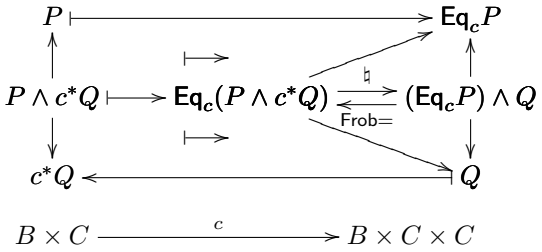




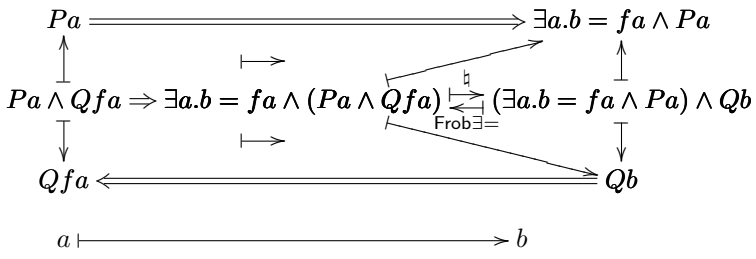
3.22 Frobenius for Equality

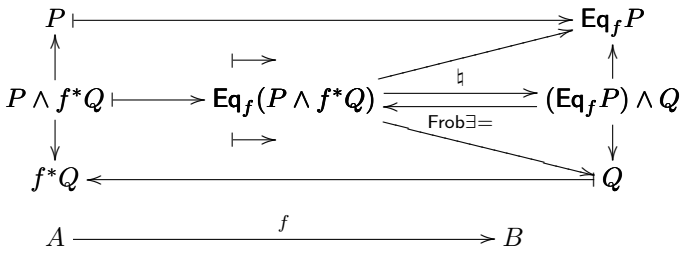
[Frobenius for equality:]





[Frobenius for an arbitrary map:]





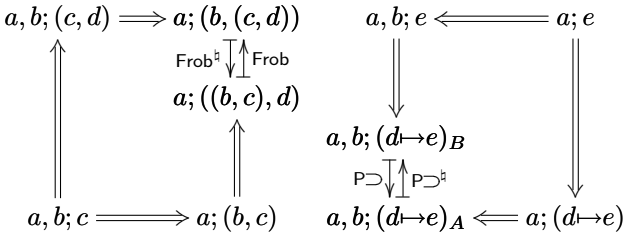
3.23 Hyperdoctrines

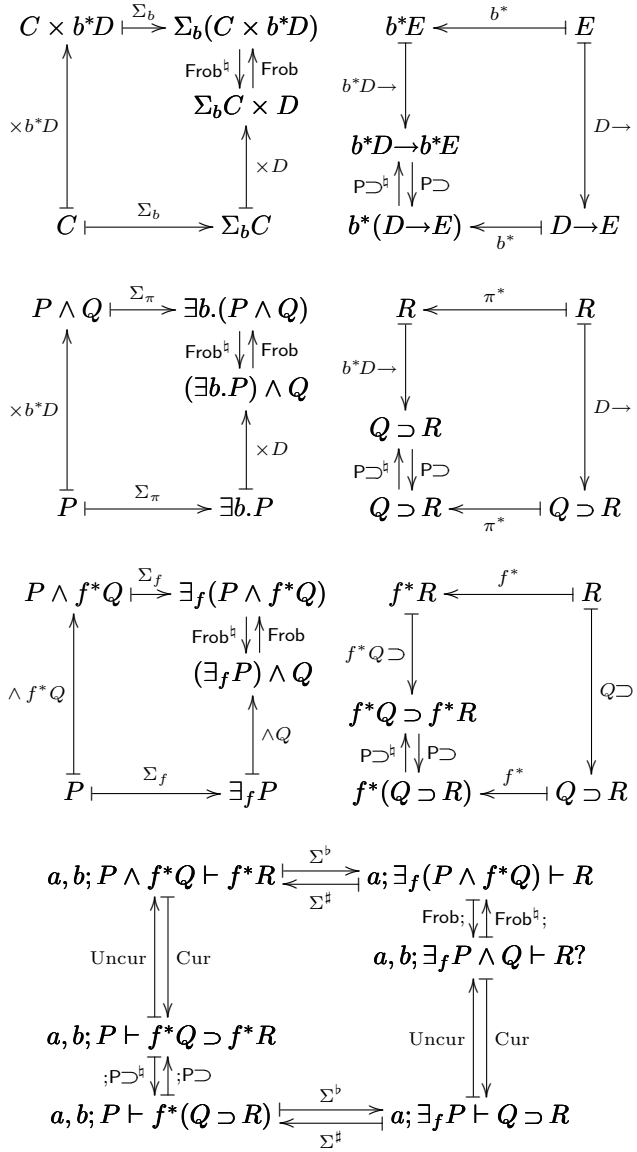
We say that a cloven fibration $p : \mathbb{E} \to \mathbb{B}$ is a *hyperdoctrine* when:
(1) the base category $\mathbb{B}$ is a CCC,
(2) each fiber $\mathbb{E}_B$ is a CCC,
and for each $f : A \to B$ in $\mathbb{B}$ the change-of-base functor f^* has:
(3) a left adjoint, $\Sigma_f \dashv f*$,
(4) a right adjoint, $f^* \dashv \Pi_f$,
and each change-of base functor f^* preserves, modulo iso,
(5) the terminal,
(6) binary products,
(7) exponentials,
and furthermore the following three "technical conditions" hold:
(8) Beck-Chevalley for "exists",
(9) Beck-Chevalley for "forall",
(10) Frobenius.

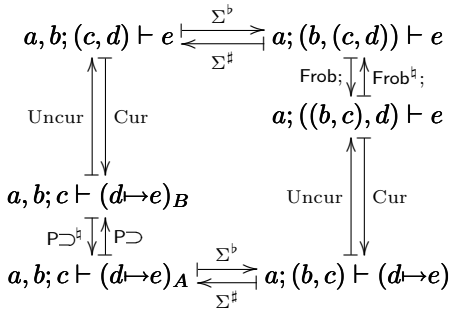
Our archetypical fibration cod : Sub(**Set**) → **Set** is a hyperdoctrine — it was the motivation for the definition of hyperdoctrine, by the way — and we can use a notation "lifted" from cod : Sub(**Set**) → **Set** to define each of these properties diagramatically, an to prove a few theorems about hyperdoctrines.

3.24 Three Theorems from Lawvere's "Hyperdoctrines" Paper

3.24.1 Frob and Pimp



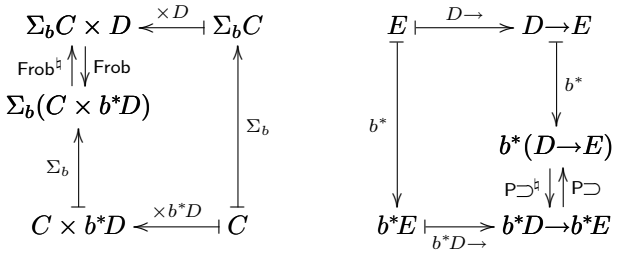

$C \times b^*D$
Σ_b
$\Sigma_b(C \times b^*D)$
$\mathsf{Frob}^\natural$
Frob
$\Sigma_b C \times D$
$\times b^*D$
$\times D$
C
Σ_b
$\Sigma_b C$
b^*E
b^*
E
$b^*D\to$
$D\to$
$b^*D \to b^*E$
$\mathsf{P}\supset^\natural$
$\mathsf{P}\supset$
$b^*(D\to E)$
b^*
$D\to E$
$P \wedge Q$
Σ_π
$\exists b.(P \wedge Q)$
$(\exists b.P) \wedge Q$
P
$\exists b.P$
R
π^*
$Q \supset R$
$Q \supset R$
$P \wedge f^*Q$
Σ_f
$\exists_f(P \wedge f^*Q)$
$(\exists_f P) \wedge Q$
$\wedge f^*Q$
$\wedge Q$
$\exists_f P$
f^*R
f^*
$f^*Q \supset f^*R$
$f^*(Q \supset R)$
$Q\supset$
$a, b; P \wedge f^*Q \vdash f^*R$
$\Sigma^\flat$
$\Sigma^\sharp$
$a; \exists_f(P \wedge f^*Q) \vdash R$
$\mathsf{Frob};$
$\mathsf{Frob}^\natural;$
$a, b; \exists_f P \wedge Q \vdash R?$
Uncur
Cur
$a, b; P \vdash f^*Q \supset f^*R$
$;\mathsf{P}\supset^\natural$
$;\mathsf{P}\supset$
$a, b; P \vdash f^*(Q \supset R)$
$a; \exists_f P \vdash Q \supset R$

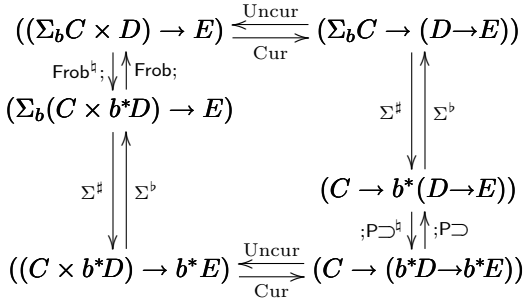


Let

$$\begin{array}{rcl} A & \equiv & \mathbf{E}[a], \\ B & \equiv & \mathbf{E}[a,b], \\ b & \equiv & a, b{\mapsto}b, \\ C & \equiv & \mathbf{O}[a,b;c], \\ D & \equiv & \mathbf{O}[a;d], \\ E & \equiv & \mathbf{O}[a;e]. \end{array}$$

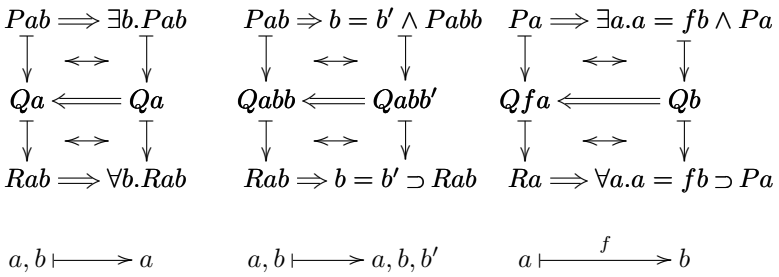
Then:





3.24.2 Adjoints to Arbitrary Changes of Base

[Explain these diagrams:]



3.24.3 Equality

```
(find-lawvere70page  6 "Substitutivity of equality")
```

PROPOSITION (SUBSTITUTIVITY OF EQUALITY). *In any eed in which, for every term $f : X \to Y$ and any two attributes α, ψ of type Y, the canonical deduction*

$$f \cdot (\alpha \Rightarrow \psi) \to f \cdot \alpha \Rightarrow f \cdot \psi$$

is an isomorphism, one also has, for any attribute ψ of type X, a canonical deduction

$$\Theta_X \to \pi_1 \cdot \psi \Rightarrow \pi_2 \cdot \psi$$

over $X \times X$.

The first canonical deduction that he mentions is my derived rule $\mathsf{P} \supset^\natural$:

$$\frac{f \quad \alpha \quad \psi}{f \cdot (\alpha \Rightarrow \psi) \to f \cdot \alpha \Rightarrow f \cdot \psi}\ \mathsf{P} \supset^\natural$$

which expands to:

$$\dfrac{\dfrac{f \quad \alpha \quad \dfrac{\alpha \quad \psi}{\alpha \Rightarrow \psi}}{f \cdot \alpha \wedge f \cdot (\alpha \Rightarrow \psi) \to f \cdot (\alpha \wedge \alpha \Rightarrow \psi)}\ \mathsf{P}\wedge \quad \dfrac{\dfrac{\dfrac{\alpha \quad \psi}{\alpha \Rightarrow \psi}}{\alpha \Rightarrow \psi \to \alpha \Rightarrow \psi}\ \text{id}}{\alpha \wedge \alpha \Rightarrow \psi \to \psi}\ \text{Uncur}}{f \cdot (\alpha \wedge \alpha \Rightarrow \psi) \to f \cdot \psi}\ (f\cdot)_1}{f \cdot \alpha \wedge f \cdot (\alpha \Rightarrow \psi) \to f \cdot \psi}\ \text{Cur}}{f \cdot (\alpha \Rightarrow \psi) \to f \cdot \alpha \Rightarrow f \cdot \psi}\ \text{Cur}$$

The second deduction is this:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\varphi}{\varphi \wedge 1_X \to \varphi}\,\mathrm{Cur}}{1_X \to \varphi \Rightarrow \varphi}\,\mathrm{iso}}{1_X \to \mathrm{id}\cdot\varphi \Rightarrow \mathrm{id}\cdot\varphi}\,\mathrm{ren}}{1_X \to (\delta\pi_1)\cdot\varphi \Rightarrow (\delta\pi_2)\cdot\varphi}\,\mathrm{iso}}{1_X \to \delta\cdot(\pi_1\cdot\psi) \Rightarrow \delta\cdot(\pi_2\cdot\psi)}\,\mathrm{iso} \quad \dfrac{\delta \quad \pi_1\cdot\psi \quad \pi_2\cdot\psi}{\delta\cdot(\pi_1\cdot\psi) \Rightarrow \delta\cdot(\pi_2\cdot\psi) \to \delta\cdot(\pi_1\cdot\psi \Rightarrow \pi_2\cdot\psi)}\,\mathsf{P}\supset}{\dfrac{\dfrac{1_X \to \delta\cdot(\pi_1\cdot\psi \Rightarrow \pi_2\cdot\psi)}{1_X\Sigma(X\delta) \to \pi_1\cdot\psi \Rightarrow \pi_2\cdot\psi}\,\mathsf{Eq}^\flat}{\Theta_X \to \pi_1\cdot\psi \Rightarrow \pi_2\cdot\psi}\,\mathrm{ren}}\;;$$

I just realized that I have never defined the "iso" rules... an example:

$$\dfrac{\alpha \leftrightarrow \alpha' \quad \varphi \leftrightarrow \varphi' \quad \psi \leftrightarrow \psi' \quad \dfrac{\alpha \quad \varphi \quad \psi}{\alpha \wedge \varphi \Rightarrow \psi}}{(\alpha \wedge \varphi \Rightarrow \psi) \leftrightarrow (\alpha' \wedge \varphi' \Rightarrow \psi')}\,\mathrm{iso}$$

Chapter 4

Notes and Further Reading

I've learned hyperdoctrines from [SeelyPLC], [SeelyHyp], [Jacobs], and [StreicherBenabou], and I found the subject very hard. The ideas from this paper can be used — I hope! — to make it a bit simpler: we've seen how to split the theory in two parts, and how to start by the syntactical part; and we can draw diagrams with the same structure as the ones in these notes but using the notations of other texts, and then we can compare the diagrams in different notations and use them as dictionaries between the different languages.

[To do: dictionaries for the papers and books above, and for [LawvereAdjFound] and [LawvereEqHyp]. The most important parts are the diagrams for the structure in $\mathbb{B}$ and $\mathbb{E}$ and for the change-of-base functors and their adjoints — the diagrams for the preservations, BCCs and Frobenius don't really need to be drawn, as they can be reconstructed from the others; we only need the show the notation for the arrows in them that are required to be isos.]

[I have the impression that there are very good accounts of fibrations in [TaylorPhDThesis] and [JohnstoneElephant], but I haven't read them with enough attention yet.]

[I've learned monads from [CWM], [TTT], [SchalkMonads], [BeckPhDThesis] and [Awodey]; compare their notations. To do, also: create a comparison diagram for comonads (easy), discuss the case of building $R[x]$ from R when R is a ring (good references: [LambekScott], [Awodey]). Try to find a proto-proof for Lambek's theorem (that says that an initial algebra in an Eilenberg-Moore category is an iso).]

[A medium-term goal: finish the downcasing of differential categories, as presented in [BluteCockettSellyDiffCats]. I have crappy ascii-art renderizations of some of the downcased diagrams here: `http://angg.twu.net/2007diffcats.html` (but it may contain errors). I don't understand enough of [BluteCockettSellyCartDiffCats] yet.]

[I have a very nice downcasing of the best part of this: Kock, Anders: A simple axiomatics for differentiation (1977) — I presented it in seminars at UFF, but I'd look to clean it up (I have better notations now), and I lost the source

code for my diagrams... (what is the URL of the PDF with the slides?)]

[This looks extremely interesting: [KellyLack], "On Property-Like Structures" — but its ideas are far above my head now. It will be better to try to read it when I have more contact with grown-up categorists.]

[This looks extremely interesting too: section D1.4, "Syntactic categories", of [JohnstoneElephant2].]

[Point to several parts of [KromerToolAndObject] where he discusses having vs. not having elements; my '$a \mapsto b$' notation is like the "internal view" of morphisms that appears in [LawvereSchanuelCM], p.13 (the "internal diagram of a set"), and it's like a mix between the usual '$x \mapsto f(x)$' and the '$y = y(x)$' notations.]

$$\frac{a \vdash D \quad a, b; c \vdash d}{a; (b, c) \vdash d}\ \Sigma^\flat \qquad \frac{a; (b, c) \vdash d}{a, b; c \vdash d}\ \Sigma^\sharp$$

$$\frac{a \vdash D_a \quad a, b; c \vdash d_{abc}}{a; p \vdash d_{abc}[b, c := \mathrm{unp}\, p]}\ \Sigma^\flat \qquad \frac{a; p \vdash d_{ap}}{a, b; c \vdash d_{ap}[p := \langle b, c \rangle]}\ \Sigma^\sharp$$

$$\frac{\dfrac{\dfrac{a \vdash D_a \quad a, b; c \vdash d_{abc}}{a; p \vdash d_{abc}[b, c := \mathrm{unp}\, p]}\ \Sigma^\flat}{a, b; c \vdash d_{abc}[b, c := \mathrm{unp}\, p][p := \langle b, c \rangle]}\ \Sigma^\sharp}{a, b; c \vdash d_{abc}[b, c := \mathrm{unp}\, \langle b, c \rangle]}\ \mathrm{ren} \qquad \frac{\dfrac{\dfrac{a; p \vdash d_{ap}}{a, b; c \vdash d_{ap}[p := \langle b, c \rangle]}\ \Sigma^\sharp}{a; p \vdash d_{ap}[p := \langle b, c \rangle][b, c := \mathrm{unp}\, p]}\ \Sigma^\flat}{a; p \vdash d_{ap}[p := \langle \mathrm{unp}\, p \rangle]}\ \mathrm{ren}$$

$$\frac{a \vdash D \quad a, b; d \vdash e}{a; d \vdash (b \mapsto e)}\ \Pi^\sharp \qquad \frac{a; d \vdash (b \mapsto e)}{a, b; d \vdash e}\ \Pi^\flat$$

$$\frac{a \vdash D_a \quad a, b; d \vdash e_{abd}}{a; d \vdash \lambda b.e_{abd}}\ \Pi^\sharp \qquad \frac{a; d \vdash f_{ad}}{a, b; d \vdash f_{ad} b}\ \Pi^\flat$$

$$\frac{\dfrac{a \vdash D_a \quad a, b; d \vdash e_{abd}}{a; d \vdash \lambda b.e_{abd}}\ \Pi^\sharp}{a, b; d \vdash (\lambda b.e_{abd}) b}\ \Pi^\flat \qquad \frac{\dfrac{a; d \vdash f_{ad}}{a, b; d \vdash f_{ad} b}\ \Pi^\flat}{a; d \vdash \lambda b.(f_{ad} b)}\ \Pi^\sharp$$

$$\begin{array}{ccc} Pab & \Longrightarrow & \exists b.Pab \\ \downarrow & \overset{\flat}{\underset{\sharp}{\dashv\!\rightrightarrows\!\vdash}} & \downarrow \\ Qa & \Longleftarrow & Qa \\ \downarrow & \overset{\flat}{\underset{\sharp}{\vdash\!\rightrightarrows\!\dashv}} & \downarrow \\ Rab & \Longrightarrow & \forall b.Rab \end{array}$$

$$a, b \longmapsto a$$

(Transcribed from the blue notebook).

••

Over the next few sections we are going to understand several *pure type systems*, then follow the formal definition of PTSs in [GeuvThesis]. The reader is encouraged to think that the PTSs that we will (begin to) define in sections _, _, _, _, namely, PTS_Θ, $\mathsf{PTS}_{\Theta\Box}$, PTS, PTS, PTS, are "fragments" of a bigger type system, $\mathsf{PTS}_{\Omega\Theta\Box}$, that will be formally defined in section _; they are "fragments" in the sense that they allow only a few of the rules of the bigger type system in their derivations (see section _), and so their sets of derivable judgments, $\mathrm{DJ}(\mathsf{PTS}_\Theta)$, ..., $\mathrm{DJ}(\mathsf{PTS}_{\Theta\Box})$, are subsets of $\mathrm{DJ}(\mathsf{PTS}_{\Omega\Theta\Box})$.

In a *pure* type system there are no constants besides the *sorts*, and the only "axioms" — i.e., the judgments that can appear in the leaves of the derivations — are only '$\vdash \Theta{:}\Box$' and '$\vdash \Omega{:}\Theta$'. Buth both the rules and the models become much easier to understand if we allow a few "impurities" of certain kinds; for example, if $\mathbb{N}$ is a set, $0{:}\mathbb{N}$ its element zero, and $s : \mathbb{N} \to \mathbb{N}$ is the successor function, then $s(s(0))$ is an element of $\mathbb{N}$, and we can derive $\vdash s(s(0)) : \mathbb{N}$ from these hypotheses using only the rules of PTS_Θ. We express this as:

$$\frac{\vdash \mathbb{N} : \Theta \quad \vdash 0 : \mathbb{N} \quad \vdash s : \mathbb{N} \to \mathbb{N}}{\vdash s(s(0)) : \mathbb{N}}\ \mathsf{PTS}_\Theta$$

where the double bar represents several derivation rules in PTS_Θ; the expansion of that derivation tree is:

$$\dfrac{\dfrac{\vdash 0 : \mathbb{N} \quad \vdash s : \mathbb{N} \to \mathbb{N}}{\vdash s(0) : \mathbb{N}}\ \mathrm{app}_{\Theta\Theta} \qquad \vdash s : \mathbb{N} \to \mathbb{N}}{\vdash s(s(0)) : \mathbb{N}}\ \mathrm{app}_{\Theta\Theta}$$

••

The intuitive way to recognize that a judgment like

$$A : \Theta,\ B : A \to \Theta,\ f : \Pi a{:}A.Ba,\ a : A \vdash fa : Ba$$

makes sense is to "read it aloud" — i.e., translate it to English — and check that its translated version "makes sense" — i.e., that all the variables that it mentions have been declared and that all the typings are correct. Let's see.

$A : \Theta$,	If we know the value of A, and it is an element of Θ (i.e., A is a set),
$B : A \to \Theta$,	and if we know the value of B, and it is an operation that maps each element of A to a set (i.e., B is a family of sets, indexed by A),
$f : \Pi a{:}A.Ba$,	and if we know the value of f, and it is a function that takes each element a of A to an element of the set Ba,

$a : A$	and if we know the value of a, and it is an element of the set A,
$\vdash fa : Ba$	then we know the value of fa, and it is an element of the set Ba.

The Θ is one of our few constants; it is a "sort" (more on "sorts" later), and it will play the role, *very* roughly, of the class of all sets. Mnemonic: "Θ" is for "*Th*ets".

••

$$\begin{array}{rl}
A : \Theta,\ B : \Theta,\ f : A \to B,\ a : A \vdash fa : B & \mathsf{PTS}_{\Theta} \\
A : \Theta,\ B : A \to \Theta,\ f : \Pi a{:}A.Ba,\ a : A \vdash fa : Ba & \mathsf{PTS}_{\Theta\Box} \\
A : \Theta \vdash \lambda a{:}A.a : A \to A & \mathsf{PTS}_{\Theta} \\
P : \Theta \vdash \lambda p{:}P.p : P \to P & \mathsf{PTS}_{\Omega} \\
A : \Theta \vdash (\lambda A{:}\Theta.\lambda a{:}A.a)A : A \to A & \mathsf{PTS}_{\Box\Theta} \\
P : \Omega \vdash (\lambda P{:}\Omega.\lambda p{:}P.p)P : P \to P & \mathsf{PTS}_{\Theta\Omega}
\end{array}$$

$$\begin{array}{ll}
A : \Theta \vdash \lambda a{:}A.a : A \to A & \mathsf{PTS}_{\Theta} \\
P : \Theta \vdash \lambda p{:}P.p : P \to P & \mathsf{PTS}_{\Omega} \\
A : \Theta \vdash (\lambda A{:}\Theta.\lambda a{:}A.a)A : A \to A & \mathsf{PTS}_{\Box\Theta} \\
P : \Omega \vdash (\lambda P{:}\Omega.\lambda p{:}P.p)P : P \to P & \mathsf{PTS}_{\Theta\Omega}
\end{array}$$

[How to interpret a judgment like $a{:}A, b{:}B_a, c{:}C_{abc} \vdash d_{abc}{:}D_{abc}$, where d_{abc} is a term, as a series of morphisms; actually to interpret these judgments categorically we will need LCCCs, that can only be explained after hyperdoctrines...]

(Transcribed from the back of p.247 in "On Property-Like Structures").
($A \to B$ is a hom-set, $A \xrightarrow{f} B$ is a morphism)
(Conventions for downcasing arrows)

Bibliography

[Awo06] S. Awodey. *Category Theory*. Oxford University Press, 2006.