Introduction to categorical logic

Tom Hirschowitz

Chambéry, May 2010

LAMA
Laboratoire de Mathématiques
Université de Savoie

UMR 5127

What is categorical logic?

- Many varieties of logics, here: mainly fragments of first-order logic, plus an incursion in (extensional) higher-order logic.
- Goal: uniform formulation of
  - their definitions,
  - the associated notions of models and maps between them.
- Tool: the informal notion of an internal language.
- Expository choice: lazy.

The three main approaches

$$\begin{tikzcd}
 & \textit{hyperdoctrines} \arrow[dl, leftrightarrow] \arrow[dr, leftrightarrow] & \\
\textit{categories} \arrow[rr, leftrightarrow] & & \textit{allegories}
\end{tikzcd}$$

- Allegories (Freyd): not covered here, but very effective.
- Variant: cartesian bicategories (Carboni and Walters).

Terms vs. formulas

First-order logic layers:

1. sorts, function symbols, equations,
2. first-order axioms.

For terms (and equations)

Categories with finite products, aka functorial semantics (Lawvere, 1963).

- With one sort t, terms $M(x_1, \ldots, x_n)$ are morphisms

$$t \times \ldots \times t \to t$$

  in a category.
- Tuples represented by (formal) products $t \times \ldots \times t$.

Hyperdoctrines

For formulas, naive idea:

- formulas are indexed over variables;
- $\rightsquigarrow$ hyperdoctrines, a kind of indexed categories.

Categories

A less naive idea:

- Start from terms, i.e., a category with finite products.
- Formulas add subobjects to terms:

$$\varphi(x) \hookrightarrow t.$$

Mainstream approach

All packed up into a category, logic done in terms of subobjects.

Allegories

A "converse" approach:

- instead of forcing formulas into terms,
- smoothly plunge operations into formulas:

$$f(x) \text{ viewed as a relation } y = f(x).$$

- Invent a calculus of relations: allegories.

Perhaps tighter:

- constructing an allegory from a theory is more direct,
- the logic is more primitive than with categories.

On the other hand: more *ad hoc*.

Warm-up: Equational theories

Definition

An equational signature is

- a set T of base types, or sorts,
- a set F of function symbols, or operations, with arities $t_1, \ldots, t_n \longrightarrow t$ in T^{n+1}.

Ex: magma, one sort t, one operation $t, t \xrightarrow{\cdot} t$: theory Σ.

- Any such signature $S = (T, F)$ freely generates a category with finite products.
- Let's define this: categories, and then finite products.

Categories

Definition

A category $\mathcal{C}$ is

- a (possibly large) graph $\mathcal{C}_1 \overset{s,t}{\rightrightarrows} \mathcal{C}_0$,
- with an associative composition of edges

$$(A \xrightarrow{f} B \xrightarrow{g} C) \mapsto (A \xrightarrow{g \circ f} B),$$

- with units $A \xrightarrow{\mathrm{id}_A} A$.

- Small: $\mathcal{C}_1$ and $\mathcal{C}_0$ are sets ($\neq$ classes).
- Locally small: for all $A, B \in \mathcal{C}_0$, $\mathcal{C}_1(A, B)$ is a set.

Boat examples of categories

- Locally small: sets, magmas, monoids, groups, ...
- Small:
  - any preordered set,
  - the paths of any graph,
  - the homotopy classes of paths of any topological space.

Our example: the theory of magmas

- Objects: natural numbers,
- Arrows $p \longrightarrow q$:
  - consider $p = \{0, \ldots, p-1\}$ as a set of variables,
  - consider terms with variables in p, as generated by the grammar

    $$M, N, \ldots ::= x \mid M \cdot N, \qquad\qquad x \in p,$$

    and call that $\mathcal{T}_\Sigma(p)$,
  - e.g., $(0 \cdot 0) \cdot 3 \in \mathcal{T}_\Sigma(4)$,
  - and let the set of arrows $p \longrightarrow q$ be $\mathcal{T}_\Sigma(p)^q$.
- Composition and identities?

The theory of magmas

The composition h of

$$p \xrightarrow{f} q \xrightarrow{g} r$$

is given by

$$h_k = g_k[f], \qquad\qquad k \in r,$$

where

- f is seen as the substitution $j \mapsto f_j$, for $j \in q$,
- and $g_k[f]$ replaces each j with f_j in g_k.

The theory of magmas

Example of composition:

$$3 \xrightarrow{(0\cdot 2),(2\cdot 1)} 2 \xrightarrow{0\cdot 1} 1$$

compose to

$$3 \xrightarrow{(0\cdot 2)\cdot(2\cdot 1)} 1.$$

Less cryptic notation:

$$x, y, z \xrightarrow{(x\cdot z),(z\cdot y)} u, v \xrightarrow{u\cdot v} 1$$

compose to

$$x, y, z \xrightarrow{(x\cdot z)\cdot(z\cdot y)} 1.$$

Identities

- The identity on p is the tuple $(0, \ldots, p-1)$.
- Seen as a substitution, replaces i with itself.

Summing up

- Objects: natural numbers,
- Arrows $p \longrightarrow q$: $\mathcal{T}_\Sigma(p)^q$.
- Composition by substitution.

Proposition

This yields a (small) category $\mathcal{C}_\Sigma$.

Associativity is a variant of the standard substitution lemma

$$f[g][h] = f[g[h]].$$

Hence:

- The theory Σ of magmas yields a category $\mathcal{C}_\Sigma$.
- Finite products?

Binary products

In any category $\mathcal{C}$, a product of two objects A, B is:

- an object C
- arrows π and π'

$$A \xleftarrow[\pi]{} C \xrightarrow[\pi']{} B$$

such that

Binary products

In any category $\mathcal{C}$, a product of two objects A, B is:

- an object C
- arrows π and π'

$$\begin{tikzcd}
 & D \arrow[dl, "f"'] \arrow[dr, "g"] & \\
A & C \arrow[l, "\pi"] \arrow[r, "\pi'"'] & B
\end{tikzcd}$$

such that for any D, f, g

Binary products

In any category $\mathcal{C}$, a product of two objects A, B is:

- an object C
- arrows π and π'

$$
\begin{tikzcd}
 & D \arrow[dl, "f"'] \arrow[d, dashed, "h"'] \arrow[dr, "g"] & \\
A & C \arrow[l, "\pi"] \arrow[r, "\pi'"'] & B
\end{tikzcd}
$$

such that for any D, f, g, there is a unique arrow h making both triangles commute.

Binary products

In any category $\mathcal{C}$, a product of two objects A, B is:

- an object C
- arrows π and π'

$$
\begin{tikzcd}
 & D \arrow[dl, "f"'] \arrow[d, dashed, "\langle f, g \rangle"'] \arrow[dr, "g"] & \\
A & A \times B \arrow[l, "\pi"] \arrow[r, "\pi'"'] & B
\end{tikzcd}
$$

such that for any D, f, g, there is a unique arrow h making both triangles commute.

Notation: $D = A \times B$, $h = \langle f, g \rangle$.

Binary products

Proposition

Binary products are unique up to unique commuting isomorphism.

Definition

An isomorphism in a category $\mathcal{C}$ is an arrow $A \xrightarrow{f} B$ which has a two-sided inverse, i.e., a g such that both

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, "\mathrm{id}_A"'] & B \arrow[d, "g"] \\ & A \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} B \arrow[r, "g"] \arrow[dr, "\mathrm{id}_B"'] & A \arrow[d, "f"] \\ & B \end{tikzcd}$$

commute.

Binary products

Proposition

Binary products are unique up to unique commuting isomorphism.

For any two products

$$\begin{tikzcd} & C \arrow[dl, "\pi"'] \arrow[dr, "\pi'"] & \\ A & & B \\ & D \arrow[ul, "\rho"] \arrow[ur, "\rho'"'] & \end{tikzcd}$$

Binary products

Proposition

Binary products are unique up to unique commuting isomorphism.

For any two products

$$\begin{tikzcd}
& C \arrow[dl, "\pi"'] \arrow[dr, "\pi'"] \arrow[dd, dashed, "i"'] & \\
A & & B \\
& D \arrow[ul, "\rho"] \arrow[ur, "\rho'"'] &
\end{tikzcd}$$

there is a unique isomorphism i making both triangles commute.

- Proof actually quite subtle, let's do it in detail.

First remark

Proposition

An iso has exactly one inverse.

Consider any two inverses j and j'. The diagram

$$\begin{tikzcd}
B \arrow[dr, "j"] \arrow[drr, bend left, "\mathrm{id}"] \arrow[ddr, bend right, "\mathrm{id}"'] & & \\
& A \arrow[r, "i"] \arrow[d, "i"'] \arrow[dr, "\mathrm{id}"] & B \arrow[d, "j"] \\
& B \arrow[r, "j'"'] & A
\end{tikzcd}$$

commutes, hence $j = j'$.

Second remark

Given a commuting iso i, i.e., one making

$$\begin{tikzcd} & C \arrow[dl, "\pi"'] \arrow[dr, "\pi'"] \arrow[d, "i"'] & \\ A & D \arrow[l, "\rho"] \arrow[r, "\rho'"'] & B \end{tikzcd}$$

commute, its inverse j is also commuting. E.g., the diagram

$$\begin{tikzcd} & C \arrow[dr, "\pi'"] \arrow[d, "i"'] & \\ D \arrow[ur, "j"] \arrow[r, "\mathrm{id}"'] & D \arrow[r, "\rho'"'] & B \end{tikzcd}$$

commutes, hence $\pi' \circ j = \rho'$. The rest symmetrically.

Third remark

$\langle \pi, \pi' \rangle = \mathrm{id}$. Indeed, $\langle \pi, \pi' \rangle$ is the unique arrow making

$$\begin{tikzcd} & C \arrow[dl, "\pi"'] \arrow[d, dashed, "{\langle \pi, \pi' \rangle}"'] \arrow[dr, "\pi'"] & \\ A & C \arrow[l, "\pi"] \arrow[r, "\pi'"'] & B \end{tikzcd}$$

commute. But id_C does, hence $\langle \pi, \pi' \rangle = \mathrm{id}$.

Proof of the proposition

Just for this proof, write:

- $\langle f, g\rangle$ for product w.r.t. C, π, π',
- $[f, g]$ for product w.r.t. D, ρ, ρ'.

Proof of the proposition: uniqueness

Any commuting inverses

$$\begin{tikzcd}
 & C \arrow[dl, bend right, "\pi"'] \arrow[dr, bend left, "\pi'"] \arrow[d, "i= {[\pi, \pi']}"] & \\
A & D \arrow[l, "\rho"'] \arrow[r, "\rho'"] \arrow[d, "j= \langle \rho, \rho' \rangle"] & B \\
 & C \arrow[ul, bend left, "\pi"] \arrow[ur, bend right, "\pi'"'] &
\end{tikzcd}$$

meet the conditions for being respectively $[\pi, \pi']$ and $\langle \rho, \rho' \rangle$.
By uniqueness, they have to.

Proof of the proposition: existence

Construct

$$\begin{tikzcd}
 & C \arrow[dl, "\pi"'] \arrow[dr, "\pi'"] \arrow[d, dashed, "{[\pi, \pi']}"'] & \\
A & D \arrow[l, "\rho"'] \arrow[r, "\rho'"] \arrow[d, dashed, "{\langle \rho, \rho' \rangle}"'] & B \\
 & C. \arrow[ul, "\pi"] \arrow[ur, "\pi'"'] &
\end{tikzcd}$$

- The dashed composite meets the condition for being $\langle \pi, \pi' \rangle$, i.e., id, hence has to.
- By a symmetric argument, the dashed arrows are two-sided, commuting inverses.

Summary

We have proved:

Proposition

Binary products are unique up to unique commuting isomorphism.

$\mathcal{C}_\Sigma$ has binary products

Slightly awkward: binary product in $\mathcal{C}_\Sigma$ is actually ... sum:

$$p \xleftarrow{(0,\ldots,p-1)} p+q \xrightarrow{(p,\ldots,p+q-1)} q,$$

e.g.,

$$x \xleftarrow{x} x,y,z \xrightarrow{(y,z)} y,z.$$

Proof by example

$$
\begin{tikzcd}
 & A \arrow[dl, "f_x"'] \arrow[dr, "(f_y, f_z)"] & \\
x & x, y, z \arrow[l, "x"'] \arrow[r, "(y, z)"] & y, z.
\end{tikzcd}
$$

E.g., the colored composite is

$$(y, z)[x \mapsto f_x, y \mapsto f_y, z \mapsto f_z],$$

i.e., (f_y, f_z), as expected.

Proof by example

$$
\begin{tikzcd}
& A \arrow[dl, "f_x"'] \arrow[d, dashed, "{(f_x, f_y, f_z)}"'] \arrow[dr, "{(f_y, f_z)}"] & \\
x & x, y, z \arrow[l, "x"'] \arrow[r, "{(y, z)}"] & y, z.
\end{tikzcd}
$$

E.g., the colored composite is

$$(y, z)[x \mapsto f_x, y \mapsto f_y, z \mapsto f_z],$$

i.e., (f_y, f_z), as expected.

Summary

We have (almost) proved:

Proposition

$\mathcal{C}_\Sigma$ *has binary products.*

More explicitly: any two objects have a binary product.

The nullary product

- From binary products, n-ary products.
- Associativity: $A \times (B \times C) \cong (A \times B) \times C$.
- How about nullary product?

The nullary product

Mimicking the binary case with $2 \rightsquigarrow 0$:

- A nullary product for a 0-tuple of objects,
- is an object 1 (with void projections),
- such that for all object D (and void arrows to the 0-tuple),
- there is a unique arrow $D \longrightarrow 1$ (making the void diagram commute).
- But there is exactly one 0-tuple of objects.

Compiling:

- The nullary product, or terminal object is an object 1,
- such that for all object A, there is a unique arrow $A \longrightarrow 1$.

Ex: in sets?

The terminal object in $\mathcal{C}_\Sigma$

Proposition

In $\mathcal{C}_\Sigma$, 0 is terminal.

Indeed, the unique morphism $A \longrightarrow 0$ is the unique 0-tuple of terms in $\mathcal{T}_\Sigma(A)$. Hence:

Proposition

$\mathcal{C}_\Sigma$ has finite products, i.e., binary products and a terminal object.

Signatures to categories: morphisms

- We have constructed a function:

$$\mathcal{F}\colon \mathsf{Sig} \to \mathsf{FPCat}$$

- from signatures $\Sigma = (T, F)$
- to categories with finite products $\mathcal{C}_\Sigma$.

But:

- There are natural morphisms of signatures
- and morphisms of categories with finite products.
- The assignment $\mathcal{F}$ extends to morphisms, i.e., to a functor.

Functors

Consider any two categories $\mathcal{C}$ and $\mathcal{D}$.

Definition

A functor $F\colon \mathcal{C} \to \mathcal{D}$ is a morphism of graphs which preserves compositions and identities, i.e.,

$$F(g \circ f) = F(g) \circ F(f) \qquad \text{and} \qquad F(\mathrm{id}_A) = \mathrm{id}_{F(A)}$$

for all sensible A, f, g.

The category of signatures

Let Sig have

- objects: signatures $\Sigma = (T, F)$,
- arrows $\Sigma \to \Sigma'$ given by:
  - a function $f_0 \colon T \to T'$, and
  - a function $f_1 \colon F \to F'$ compatible with the arities.

Formalisation

As an exercice, let's formalise this categorically.

Definition

For any set X, let $\mathcal{M}(X)$ be the free monoid on X, i.e., the set of finite words, or sequences on X.

$\mathcal{M}$ extends to a functor Set $\to$ Set:

- Recall: Set is the category of sets and functions,
- For $f\colon X \to Y$, let

$$\begin{array}{rccc} \mathcal{M}(f)\colon & \mathcal{M}(X) & \to & \mathcal{M}(Y) \\ & (x_1, \ldots, x_n) & \mapsto & (f(x_1), \ldots, f(x_n)). \end{array}$$

It is actually a monad, as we'll see later on.

Signatures as spans

- A signature (T, F) is the same as a diagram

$$\mathcal{M}(T) \xleftarrow{\ s\ } F \xrightarrow{\ t\ } T.$$

- An arrow $\Sigma \to \Sigma'$ is a pair (f_0, f_1) making

$$\begin{tikzcd}
\mathcal{M}(T) \arrow[d, "\mathcal{M}(f_0)"'] & F \arrow[l, "s"'] \arrow[r, "t"] \arrow[d, "f_1"'] & T \arrow[d, "f_0"] \\
\mathcal{M}(T') & F' \arrow[l, "s"] \arrow[r, "t"'] & T'
\end{tikzcd}$$

commute.

- Composition of arrows is componentwise composition.

Proposition

This yields a category Sig *of signatures.*

The category of categories with finite products

What should a morphism $\mathcal{C} \to \mathcal{D}$ of categories with finite products be?

- A functor $\mathcal{C} \to \mathcal{D}$,
- preserving products, i.e.,

$$F(A \times B) = F(A) \times F(B) \qquad \text{and} \qquad F(1_{\mathcal{C}}) = 1_{\mathcal{D}}.$$

Definition

Call this a finite product functor.

Details:

- We assume finite products as chosen structure;
- Finite product functors preserve it strictly.

The category of categories with finite products

Proposition

The data:

- *objects: (small) categories with finite products,*
- *arrows: finite product functors,*
- *composition: composition of finite product functors,*

define a (locally small) category FPCat*.*

The functor

The assignment

$$\mathcal{F}\colon \mathsf{Sig} \to \mathsf{FPCat}$$

- from signatures $\Sigma = (T, F)$
- to categories with finite products $\mathcal{C}_\Sigma$

extends to a functor.
Let's do that by example.

The functor by example

- Recall the theory Σ of magmas, with operation $t, t \xrightarrow{\cdot} t$.
- Assume a morphism $f = (f_0, f_1) \colon \Sigma \to \Sigma'$:

$$\begin{tikzcd}
\mathcal{M}(T) \arrow[d, "\mathcal{M}(f_0)"'] & F = \{\cdot\} \arrow[l, "s"'] \arrow[r, "t"] \arrow[d, "f_1"'] & T = \{t\} \arrow[d, "f_0"] \\
\mathcal{M}(T') & F' \arrow[l, "s"] \arrow[r, "t"'] & T'.
\end{tikzcd}$$

- In particular, let $t' = f_0(t)$.
- And let $\star \colon t', t' \to t'$ be $f_1(\cdot)$.

The functor by example

Summary for f:

Theory	Magmas Σ	Σ'
Basic type	t	t'
Operation	$\cdot$	$\star$

Define $\mathcal{F}(f)$ to be the finite product functor $\mathcal{C}_\Sigma \to \mathcal{C}_{\Sigma'}$:

- on objects: $p \mapsto {t'}^p$, i.e., $t' \times \ldots \times t'$;
- on morphisms $p \xrightarrow{g} q$, define $\mathcal{F}(f)$
  - componentwise: $g = (g_1, \ldots, g_q)$,
  - and then by induction on terms:
    - $\mathcal{F}(f)(z) = z$,
    - $\mathcal{F}(f)(M \cdot N) = \mathcal{F}(f)(M) \star \mathcal{F}(f)(N)$.

The functor $\mathcal{F}$, and back

We have constructed a functor

$$\mathcal{F} : \mathsf{Sig} \to \mathsf{FPCat},$$

sending Σ to $\mathcal{F}(\Sigma) = \mathcal{C}_\Sigma$ on objects.

Now, there is another functor

$$\mathcal{U} : \mathsf{FPCat} \to \mathsf{Sig},$$

sending any category with finite products $\mathcal{C}$ to the signature with

- types the objects of $\mathcal{C}$,
- operations $c_1, \ldots, c_n \to c$ all morphisms

$$c_1 \times \cdots \times c_n \to c.$$

The unit

The signature $\mathcal{U}(\mathcal{C})$ is big.
Example: for magmas, the signature $\mathcal{U}(\mathcal{F}(\Sigma))$ has

- types the natural numbers,
- and operations $p_1, \ldots, p_n \to p$ all morphisms $p_1 \times \ldots \times p_n \to p$ in $\mathcal{C}_\Sigma$, i.e.,
- all p-tuples of terms in $\mathcal{T}(p_1 + \ldots + p_n)$.

The unit

Observation: there is a morphism

$$\eta_\Sigma \colon \Sigma \to \mathcal{U}(\mathcal{F}(\Sigma)),$$

sending each operation to itself, seen as a morphism in $\mathcal{F}(\Sigma)$.

Example

- The operation $t, t \stackrel{\cdot}{\to} t$ is sent to
- the morphism $t \times t \stackrel{\cdot}{\to} t$,
- seen as an operation in $\mathcal{U}(\mathcal{F}(\Sigma))$.

Naturality of the unit

For all sensible F, the diagram

$$\begin{tikzcd} \Sigma \arrow[r, "\eta_\Sigma"] \arrow[d, "F"'] & \mathcal{U}(\mathcal{F}(\Sigma)) \arrow[d, "\mathcal{U}(\mathcal{F}(F))"] \\ \Sigma' \arrow[r, "\eta_{\Sigma'}"'] & \mathcal{U}(\mathcal{F}(\Sigma')) \end{tikzcd}$$

commutes.

Proof.

Easy check: $\mathcal{F}(F)$ is defined by induction from F. □

Interlude: natural transformations

Definition

A natural transformation
$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] \arrow[r, bend right, "G"'] \arrow[r, phantom, "\Downarrow\alpha"] & \mathcal{D} \end{tikzcd}$$
is a family of arrows $\alpha_c \colon F(c) \to G(c)$ making the diagram

$$\begin{tikzcd} F(c) \arrow[r, "\alpha_c"] \arrow[d, "F(f)"'] & G(c) \arrow[d, "G(f)"] \\ F(d) \arrow[r, "\alpha_d"'] & G(d) \end{tikzcd}$$

commute for all sensible f.

Naturality of the unit

We have proved:

Proposition

The units $\Sigma \to \mathcal{U}(\mathcal{F}(\Sigma))$ form a natural transformation

$$\begin{tikzcd} \mathsf{Sig} \arrow[r, bend left, "\mathrm{id}_{\mathsf{Sig}}"] \arrow[r, bend right, "\mathcal{U} \circ \mathcal{F}"'] \arrow[r, phantom, "\Downarrow\eta"] & \mathsf{Sig}. \end{tikzcd}$$

Universal property of the unit

For any F, there is a unique $\overline{F}$ making the following triangle commute:

$$\begin{tikzcd}
\Sigma \arrow[r, "\eta_\Sigma"] \arrow[dr, "F"'] & \mathcal{U}(\mathcal{F}(\Sigma)) \arrow[d, dashed, "\mathcal{U}(\overline{F})"] \\
& \mathcal{U}(\mathcal{C}),
\end{tikzcd}
\quad \text{where} \quad
\begin{tikzcd}
\mathcal{F}(\Sigma) \arrow[d, "\overline{F}"] \\
\mathcal{C}.
\end{tikzcd}$$

Idea of the proof

- Finite products are expressive enough to encode term formation.
- Ex:
  - if $F : t \mapsto t'$ and $\cdot \mapsto \star$,
  - given $M, N \in \mathcal{T}_\Sigma(p)$,
  - translated to $[\![M]\!]$ and $[\![N]\!]$,
  - send $M \cdot N$ to $t'^p \xrightarrow{\langle [\![M]\!], [\![N]\!] \rangle} t' \times t' \xrightarrow{\star} t'$.
- The constraint that $\overline{F}$ be functorial and preserve finite products forces it to be that way.

Informally

- The pair (Σ, F) is an internal language of $\mathcal{C}$.
- $\mathcal{U}(\mathcal{C})$ is the internal language of $\mathcal{C}$ (actually, the version with equations, see below).

Definition

An adjunction
$$\begin{tikzcd} \mathcal{C} \arrow[rr, bend left, "F"] \arrow[rr, phantom, "\perp"] & & \mathcal{D} \arrow[ll, bend left, "G"] \end{tikzcd}$$
is a natural transformation

$$\begin{tikzcd} \mathcal{C} \arrow[rr, bend left, "\mathrm{id}_{\mathcal{C}}"] \arrow[rr, bend right, "G \circ F"'] \arrow[rr, phantom, "\Downarrow \eta"] & & \mathcal{C}, \end{tikzcd}$$

such that for all f there is a unique $\overline{f}$ making the triangle commute

$$\begin{tikzcd} c \arrow[r, "\eta_c"] \arrow[dr, "f"'] & G(F(c)) \arrow[d, dashed, "G(\overline{f})"] \\ & G(d), \end{tikzcd}$$

where

$$\begin{tikzcd} F(c) \arrow[d, "\overline{f}"] \\ d. \end{tikzcd}$$

Adjunctions

Terminology

In $\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, bend left, "G"] \end{tikzcd}$, F is the left adjoint and G is the right adjoint.

Proposition

Saying that F has a right adjoint determines G up to iso, and conversely.

Your first adjunction!

We have proved:

Proposition

There is an adjunction $\begin{tikzcd} \mathrm{Sig} \arrow[r, bend left, "\mathcal{F}"] \arrow[r, phantom, "\perp"] & \mathrm{FPCat}. \arrow[l, bend left, "\mathcal{U}"] \end{tikzcd}$

- Discovered and studied by Lawvere under the name functorial semantics.
- Generalises usual semantics in Set: models may exist in any category with finite products.
- Nicely ties syntax and semantics together.

And the previous ones

Actually, you've already seen three adjunctions:

- between sets and monoids $\begin{tikzcd} \mathrm{Set} \arrow[r, bend left, "\mathcal{M}"] \arrow[r, phantom, "\perp"] & \mathrm{Mon} \arrow[l, bend left] \end{tikzcd}$,
- between the diagonal and binary product

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "\Delta"] \arrow[r, phantom, "\perp"] & \mathcal{C} \times \mathcal{C} \arrow[l, bend left, "\times"] \end{tikzcd},$$

- between ! and the terminal object

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "!"] \arrow[r, phantom, "\perp"] & 1 \arrow[l, bend left, "1"] \end{tikzcd}.$$

Slogan

An original slogan of category theory was:

Slogan

Adjoint functors are everywhere.

- Very abstract and scary.
- Very powerful:
  - limits (generalising finite products),
  - colimits (idem for coproducts),
  - free constructions (e.g., algebraic),
  - a few more well-known "types" of adunctions,
  - maybe a lot more to be discovered.

A bit more on adjunctions

Equivalent definition (among others):

Definition

An adjunction $\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, bend left, "G"] \end{tikzcd}$ is a natural isomorphism

$$\frac{\mathcal{D}(F(c), d)}{\mathcal{C}(c, G(d))}$$

of functors $\mathcal{C}^{op} \times \mathcal{D} \to \mathrm{Set}$.

Equational theories

- Up to now, signatures: sorts and operations.
- Routine, but bureaucratic generalisation:

Definition

An equational theory $\tau = (T, F, E)$ is

- a signature $\Sigma = (T, F)$, plus
- a set of equations, i.e., elements of $\coprod_{p \in \mathbb{N}} \mathcal{T}_\Sigma(p)^2$.

- A morphism of equational theories is
  - a morphism f of signatures,
  - such that $\mathcal{F}(f)$ respects the equations.

Proposition

This yields a category ETh *of equational theories.*

Extending the adjunction

Too briefly:

- $\mathcal{F}$ extends to equational theories by quotienting $\mathcal{C}_\Sigma$ by the equations.
- $\mathcal{U}$ refines into a functor FPCat $\longrightarrow$ ETh:
  - $\mathcal{U}(\mathcal{C})$ comes with a morphism

    $$\mathcal{T}(\mathcal{U}(\mathcal{C})) \xrightarrow{h_\mathcal{C}} \mathcal{U}(\mathcal{C})$$

    interpreting terms in $\mathcal{C}$;
  - take as equations all terms identified by this $h_\mathcal{C}$.

Extending the adjunction

Proposition

This still yields an adjunction

$$\begin{tikzcd} \mathrm{ETh} \arrow[r, bend left, "\mathcal{F}"] \arrow[r, phantom, "\perp"] & \mathrm{FPCat}. \arrow[l, bend left, "\mathcal{U}"] \end{tikzcd}$$

This is both a soundness and completeness theorem:

- $\mathcal{U}(\mathcal{F}(\tau))$ contains everything derivable from τ.
- Existence of $\overline{F}$ is soundness: derivable implies true in all models.
- Completeness: $\mathcal{F}(\tau)$ is the generic model where exactly what's derivable is true.

Summary of functorial semantics

Equational theories specify categories with finite products.

Otherwise said:

The semantics for equational theories is in categories with finite products.

Remark

Completely eluded here: the importance of monads in the picture, both

- as a tool for presenting the framework (e.g., $\mathcal{T}$ is a monad),
- as an alternative semantics.

Hyperdoctrines

Now, how do formulas enter the picture?

- Consider an equational theory $\tau = (T, F, E)$,
- plus a set of formulas A on the generated terms.

Observation 1

- Formulas make sense in a context, i.e., an object of $\mathcal{C}_\tau$.
- Example: $0 = 1$ makes sense in 2.
- More readable: $x = y$ makes sense in x, y.
- Above each object, actually a partially ordered set (poset).
- So we have an indexed poset:

$$\mathcal{H} \colon \operatorname{ob} \mathcal{C}_\tau \to \mathrm{PoSet}.$$

Observation 2

- Morphisms in $\mathcal{C}_\tau$ act on formulas:

$$\varphi(1, \ldots, q)$$

$$p \xrightarrow{f} q$$

- Contravariantly.
- Functorially: $\varphi(f(g(x))) = \varphi(f \circ g(x))$, i.e.,

$$\varphi \cdot f \cdot g = \varphi \cdot (f \circ g).$$

- So we have a functor:

$$\mathcal{H} \colon \mathcal{C}_\tau^{op} \to \mathsf{PoSet}.$$

Observation 2

- Morphisms in $\mathcal{C}_\tau$ act on formulas:

$$\varphi(f_1, \ldots, f_q) \dashleftarrow\!\!\!\shortmid \varphi(1, \ldots, q)$$

$$p \xrightarrow{\quad f \quad} q$$

- Contravariantly.
- Functorially: $\varphi(f(g(x))) = \varphi(f \circ g(x))$, i.e.,

$$\varphi \cdot f \cdot g = \varphi \cdot (f \circ g).$$

- So we have a functor:

$$\mathcal{H} \colon \mathcal{C}_\tau^{op} \to \mathrm{PoSet}.$$

Generalisations

Two directions:

- proof theory: replace PoSet with Cat,
- replace 'functor' with 'pseudofunctor' (or 'fibration').

Not pursued here.

Logic in a hyperdoctrine

So:

- We interpreted equational theories τ in categories with finite products $\mathcal{C}_\tau$.
- The proposal is to interpret logic over τ as a functor

$$\mathcal{H} \colon \mathcal{C}_\tau^{op} \to \mathrm{PoSet}.$$

What to interpret?

- Equational theories: substitution as composition.
- Logic: implication as ordering; other connectives?

Propositional

Definition

A Heyting algebra is a which is bicartesian closed as a category. A morphism between such is a structure-preserving, monotone map.

I.e., we may interpret logic with $\top$, $\bot$, $\wedge$, $\vee$, $\Rightarrow$ in functors

$$\mathcal{C}_T^{op} \to \mathrm{HA},$$

where HA is the category of Heyting algebras.

Adjunctions

This defines connectives in terms of adjunctions

$$\frac{\varphi \leq \psi \qquad \varphi \leq \theta}{\varphi \leq \psi \wedge \theta} \qquad\qquad \frac{}{\varphi \leq \top}$$

$$\frac{\varphi \leq \theta \qquad \psi \leq \theta}{\varphi \vee \psi \leq \theta} \qquad\qquad \frac{}{\bot \leq \varphi}$$

$$\frac{\varphi \wedge \psi \leq \theta}{\psi \leq (\varphi \Rightarrow \theta)}$$

Quantifiers

- Define $\forall$ by adjunction:

$$\frac{\varphi(x) \leq \psi(x,y)}{\varphi(x) \leq \forall y.\psi(x,y).}$$

- But $\varphi(x)$ lives over the object x, while
- the inequality $\varphi(x) \leq \psi(x,y)$ really lives over x, y.
- So we should write

$$(\varphi \cdot \pi)(x,y) \leq \psi(x,y),$$

where

  - $\pi \colon x, y \longrightarrow x$ is the projection, and
  - $(\varphi \cdot \pi)(x,y) = \mathcal{H}(\pi)(\varphi)(x,y) = \varphi(x)$.

Quantifiers

Rephrasing:

$$\frac{\varphi \cdot \pi(x,y) \leq \psi(x,y)}{\varphi(x) \leq \forall y.\psi(x,y).}$$

Hence:

Observation 1

Universal quantification is a map *of posets*

$$\mathcal{H}(p+1) \to \mathcal{H}(p)$$

right adjoint to $\mathcal{H}(\pi)$.

Remark: not in HA, since in general

$$\forall x.(\varphi \Rightarrow \psi) \neq (\forall x.\varphi) \Rightarrow (\forall x.\psi).$$

Quantification and substitution

- Assuming no variable capture, substitution interacts with $\forall$ via:
$$(\forall x.\varphi)[f] = \forall x.(\varphi[f]).$$
- The square
$$\begin{tikzcd} p+1 \arrow[r, "\pi"] \arrow[d, "f+1"'] & p \arrow[d, "f"] \\ q+1 \arrow[r, "\pi"'] & q \end{tikzcd}$$
is a pullback.

Interlude: pullbacks

As with products, but over a fixed object D:

$$\begin{tikzcd} A \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d] \\ C \arrow[r] & D. \end{tikzcd}$$

Interlude: pullbacks

As with products, but over a fixed object D:

$$\begin{tikzcd}
X \arrow[drr, bend left] \arrow[ddr, bend right] & & \\
& A \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d] \\
& C \arrow[r] & D.
\end{tikzcd}$$

Interlude: pullbacks

As with products, but over a fixed object D:

$$\begin{tikzcd}
X \arrow[dr, dashed] \arrow[drr, bend left] \arrow[ddr, bend right] & & \\
& A \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d] \\
& C \arrow[r] & D.
\end{tikzcd}$$

Quantification and substitution

Rephrasing $(\forall x.\varphi)[f] = \forall x.(\varphi[f])$:

Observation 2

For any pullback square as above, the square

$$\begin{tikzcd}
\mathcal{H}(p+1) \arrow[r, "\forall"] & \mathcal{H}(p) \\
\mathcal{H}(q+1) \arrow[u, "- \cdot (f+1)"] \arrow[r, "\forall"'] & \mathcal{H}(q) \arrow[u, "- \cdot f"']
\end{tikzcd}$$

commutes in PoSet.

Generalised quantifiers

To get hyperdoctrines,

- require such right adjoints
  - not only to $- \cdot \pi$,
  - but to arbitrary $- \cdot f$,

  satisfying a similar condition, called a Beck-Chevalley condition,
- and require also left adjoints to encode $\exists$.

Hyperdoctrines

Definition

A (posetal, strict) hyperdoctrine is a functor $\mathcal{H}\colon \mathcal{C}_\tau^{op} \to \mathrm{HA}$ with left and right adjoints to all $\mathcal{H}(f)$:

$$\exists_f \dashv (- \cdot f) \dashv \forall_f$$

making

- for every pullback square as on the left
- the right-hand diagram commutes serially in PoSet:

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "h"'] \arrow[dr, phantom, "\lrcorner", very near start] & C \arrow[d, "k"] \\ B \arrow[r, "g"'] & D \end{tikzcd} \qquad \begin{tikzcd} \mathcal{H}(A) \arrow[r, shift left, "{\forall_f, \exists_f}"] \arrow[r, shift right] & \mathcal{H}(C) \\ \mathcal{H}(B) \arrow[u, "- \cdot h"] \arrow[r, shift left, "{\forall_g, \exists_g}"] \arrow[r, shift right] & \mathcal{H}(D). \arrow[u, "- \cdot k"'] \end{tikzcd}$$

Morphisms of hyperdoctrines

A morphism between hyperdoctrines $\mathcal{H}$ and $\mathcal{H}'$ is a diagram

$$\begin{tikzcd}
\mathcal{C}^{op} \arrow[rr, "F^{op}"] \arrow[dr, "\mathcal{H}"'] & \arrow[d, phantom, "\overset{\alpha}{\Longrightarrow}"] & \mathcal{D}^{op} \arrow[dl, "\mathcal{H}'"] \\
& \mathrm{HA}, &
\end{tikzcd}$$

with

- F preserving finite products, and
- α preserving $\forall$ and $\exists$.

The category of hyperdoctrines

Proposition

Hyperdoctrines and their morphisms form a category Hyp.

First-order signatures

Definition

A 1st-order signature $\Sigma = (T, F, R)$ consists of:

- (T, F) is an equational signature,
- R is a set of relations, equipped with a function $R \to \mathcal{M}(T)$.

Categorically, a diagram:

$$R \xrightarrow{a} \mathcal{M}(T) \xleftarrow{s} F \xrightarrow{t} T.$$

Proposition

The obvious morphisms yield a category Sig_1.

Theories

- Consider a signature $\Sigma = (T, F, R)$.
- Let $\mathrm{Form}(\Sigma)$ be the set of formulas generated by R, $=$, $\wedge$, ...

Definition

A 1st-order theory τ consists of

- a 1st-order signature $\Sigma = (T, F, R)$, plus
- a set E of equations in $\mathcal{T}(T, F)^2$, plus
- a set A of axioms in $\mathrm{Form}(\Sigma)$.

Theories

A morphism between theories is

- a morphism between the underlying signatures,
- sending equations to equations,
- and axioms to axioms.

Proposition

This yields a category Th_1 *of first-order theories.*

The free hyperdoctrine

Given a theory $\tau = (T, F, R, E, A)$, construct a hyperdoctrine $\mathcal{H}_\tau$ with:

- base category $\mathcal{C}_{(T,F,E)}$, terms modulo equations,
- over each object c, formulas in (T, F, R) with variables in c, modulo provable equivalence.

Proposition

The assignment

$$\mathcal{F} \colon \mathrm{Th}_1 \to \mathrm{Hyp}$$

- *from theories τ*
- *to hyperdoctrines $\mathcal{H}_\tau$*

extends to a functor.

The internal language of $\mathcal{H}$

Definition

Define $\mathcal{U}_0(\mathcal{H})$ to be the 1st-order signature with

- Types: the objects of $\mathcal{C}$.
- Operations $t_1, \ldots, t_n \to t$: morphisms $t_1 \times \ldots \times t_n \to t$.
- Relation symbols $R \colon t_1, \ldots, t_n \to \mathit{prop}$: objects of $\mathcal{H}(t_1 \times \ldots \times t_n)$.

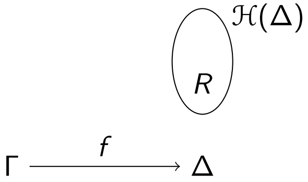
How to deal with axioms?

Interpreting formulas

Interpet formulas over $\mathcal{U}_0(\mathcal{H})$ in context $\Gamma = (x_1 \colon t_1, \ldots, x_p \colon t_p)$ by induction:

$$\begin{array}{rcl} \varphi & \mapsto & [\![\varphi]\!] \\ \hline R(f_1, \ldots, f_q) & \mapsto & R \cdot \langle f_1, \ldots, f_q \rangle \end{array}$$

where



Interpreting formulas

Interpet formulas over $\mathcal{U}_0(\mathcal{H})$ in context $\Gamma = (x_1 \colon t_1, \ldots, x_p \colon t_p)$ by induction:

$$\begin{array}{rcl} \varphi & \mapsto & [\![\varphi]\!] \\ \hline R(f_1, \ldots, f_q) & \mapsto & R \cdot \langle f_1, \ldots, f_q \rangle \\ f = g & \mapsto & Eq(\top) \cdot \langle f, g \rangle \end{array}$$

where $f, g \colon \Gamma \to \Delta$.

Interpreting formulas

Interpet formulas over $\mathcal{U}_0(\mathcal{H})$ in context $\Gamma = (x_1 \colon t_1, \ldots, x_p \colon t_p)$ by induction:

$$
\begin{array}{rcl}
\varphi & \mapsto & [\![\varphi]\!] \\
\hline
R(f_1, \ldots, f_q) & \mapsto & R \cdot \langle f_1, \ldots, f_q \rangle \\
f = g & \mapsto & Eq(\top) \cdot \langle f, g \rangle \\
\top & \mapsto & \top \\
\bot & \mapsto & \bot \\
\varphi \wedge \psi & \mapsto & [\![\varphi]\!] \wedge [\![\psi]\!] \\
\varphi \vee \psi & \mapsto & [\![\varphi]\!] \vee [\![\psi]\!] \\
\varphi \Rightarrow \psi & \mapsto & [\![\varphi]\!] \Rightarrow [\![\psi]\!] \\
\forall x \colon t.\varphi & \mapsto & \forall_\pi [\![\varphi]\!] \\
\exists x \colon t.\varphi & \mapsto & \exists_\pi [\![\varphi]\!]
\end{array}
$$

where $\pi \colon \Gamma, t \to \Gamma$ is the projection.

The internal language of $\mathcal{H}$

Definition

Define $\mathcal{U}(\mathcal{H})$ to be the 1st-order theory with

- Types: the objects of $\mathcal{C}$.
- Operations $t_1, \ldots, t_n \to t$: morphisms $t_1 \times \ldots \times t_n \to t$.
- Equations: those validated by $\mathcal{C}$.
- Relation symbols $R \colon t_1, \ldots, t_n \to prop$: objects of $\mathcal{H}(t_1 \times \ldots \times t_n)$.
- Axioms the formulas φ in $\mathrm{Form}(\mathcal{H})$ such that $\top \leq [\![\varphi]\!]$.

Proposition

This assignment extends to a functor $\mathcal{U} \colon \mathrm{Hyp} \to \mathrm{Th}_1$.

The adjunction, at last

Theorem

These functors define an adjunction

$$\begin{tikzcd} \mathsf{Th}_1 \arrow[r, bend left, "\mathcal{F}"] \arrow[r, phantom, "\perp"] & \mathsf{Hyp} \arrow[l, bend left, "\mathcal{U}"] \end{tikzcd}$$

between first-order theories and hyperdoctrines.

- Soundness: any derivable sequent holds in any model.
- Completeness: $\mathcal{F}(\tau)$ validates exactly the provable.

A remark: internal equality

- Morphisms in $\mathcal{C}_\tau$ are sometimes equal.
- That is external equality.
- A notion of equality internal to the logic may be specified by adjunction.

A remark: internal equality

- Consider duplication

$$\delta \colon x \colon A, y \colon B \xrightarrow{(x,x,y)} u \colon A, u' \colon A, v \colon B.$$

We may define $Eq(\varphi)(u, u', v) = \varphi(u, v) \wedge (u = u')$ by adjunction:

$$\frac{\varphi(x, y) \leq \psi(x, x, y)}{\varphi(u, v) \wedge u = u' \leq \psi(u, u', v)} \qquad \frac{\varphi \leq \psi \cdot \delta}{Eq(\varphi) \leq \psi}$$

A remark: internal equality

- Then define equality of $f, g \colon A \to B$ as
$$Eq(\top) \cdot \langle f, g \rangle,$$
i.e., $\top \wedge (f = g)$.
- This (bidirectional) rule is interderivable with more usual rules for equality.

Constructing hyperdoctrines

An important construction of hyperdoctrines

- Start from a small category $\mathcal{C}$ with finite limits.
- Let $\begin{array}{rccc} \mathcal{H}_{\mathcal{C}}\colon & \mathcal{C}^{op} & \longrightarrow & \mathrm{PoSet} \\ & c & \longmapsto & Sub(c), \end{array}$

  where $Sub(c)$ is the set of equivalence classes of monics into c.

Interlude: monic arrows

Definition

An arrow $f : c \to d$ in $\mathcal{C}$ is monic when for all g, h as in

$$e \overset{g,h}{\rightrightarrows} c \xrightarrow{f} d$$

such that $fg = fh$, also $g = h$.

- In Set: injective.
- Generally written $c \rightarrowtail d$.

Interlude: monic arrows

Proposition

For any commuting triangle

$$\begin{tikzcd} c \arrow[rr, "f"] \arrow[dr, tail, "u"'] & & d \arrow[dl, tail, "v"] \\ & e, & \end{tikzcd}$$

f is monic.

Interlude: monic arrows

Proposition

For any commuting triangle

$$\begin{tikzcd} a \arrow[r, shift left, "{g, h}"] \arrow[r, shift right] & c \arrow[rr, "f"] \arrow[dr, tail, "u"'] & & d \arrow[dl, tail, "v"] \\ & & e, & \end{tikzcd}$$

f is monic.

Proof: Assume g and h such that $fg = fh$.

- Then also $vfg = vfh$ by composition with v, i.e.,
- $ug = uh$.
- Hence $g = h$ since u is monic.

While we are at it

The dual is:

Definition

An arrow $f : c \longrightarrow d$ in $\mathcal{C}$ is epic when for all g, h as in

$$c \xrightarrow{\quad f \quad} d \overset{g,h}{\rightrightarrows} e$$

such that $gf = hf$, also $g = h$.

- In Set: surjective (trap: not in monoids).
- Generally written $c \twoheadrightarrow d$.
- Mnemonic: f should cover d to detect differences.

Constructing hyperdoctrines

An important construction of hyperdoctrines

- Start from a small category $\mathcal{C}$ with finite limits.
- Let $$\begin{array}{rccc} \mathcal{H}_{\mathcal{C}}: & \mathcal{C}^{op} & \to & \mathrm{PoSet} \\ & c & \mapsto & Sub(c), \end{array}$$
  where $Sub(c)$ is the set of equivalence classes of monics into c.

Important point, $Sub(c)$ is a poset:

- between u and v, at most one arrow since v monic,
- no cycle since we have quotiented under isomorphism.

What does $\mathcal{H}$ on morphisms?

Constructing hyperdoctrines

$$\begin{tikzcd} & b \arrow[d, tail, "u"] \\ c \arrow[r, "f"'] & d \end{tikzcd}$$

- Why is $u \cdot f$ monic?
- What does $- \cdot f$ do on morphisms?
- Why is $\mathcal{H}$ functorial?

Constructing hyperdoctrines

$$\begin{tikzcd} a \arrow[r] \arrow[d, "u \cdot f"'] \arrow[dr, phantom, "\lrcorner", very near start] & b \arrow[d, tail, "u"] \\ c \arrow[r, "f"'] & d \end{tikzcd}$$

- Why is $u \cdot f$ monic?
- What does $- \cdot f$ do on morphisms?
- Why is $\mathcal{H}$ functorial?

$\mathcal{H}$ is functorial

Lemma (The pullback lemma)

In a diagram

$$\begin{tikzcd} a \arrow[r] \arrow[d] & b \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & c \arrow[d] \\ x \arrow[r] & y \arrow[r] & z \end{tikzcd}$$

the left-hand square is a pullback iff the outer rectangle is.

Hence $u \cdot f \cdot g = u \cdot fg$, i.e., functoriality of $\mathcal{H}$.

When is $\mathcal{H}_{\mathcal{C}}$ a hyperdoctrine?

Definition

A category with finite limits is:

1. regular if it has stable images,
2. coherent if regular with stable unions,
3. effective if it has stable quotients of equivalence relations,
4. positive if coherent with disjoint finite coproducts,
5. Heyting if the pullback functors have right adjoints.

A category with all that is a Heyting pretopos.

- This yields enough to interpret 1st-order logic in $\mathcal{H}_{\mathcal{C}}$.
- Examples: conjunction, $\exists_f$, implication.

Conjunction

Conjunction is just pullback, i.e., intersection:

$$\begin{tikzcd} a \cap b \arrow[r, tail] \arrow[d, tail] \arrow[dr, phantom, "\lrcorner", very near start] & b \arrow[d, tail] \\ a \arrow[r, tail] & c. \end{tikzcd}$$

Proof.

The subobject $a \cap b$ is a product in the poset $Sub(c)$. □

Images and $\exists_f$

Definition

A category has images when every morphism has a an initial epi-mono factorisation.

- The epi in such a factorisation has to be a cover, i.e., the only subobjects through which it factors are isomorphisms.
- Requiring images to be stable under pullback = requiring covers to be.

Images and $\exists_f$

This allows interpreting $\exists_f$:

$$\begin{tikzcd}
a \cap b \arrow[r, two heads] \arrow[d, tail, "u"'] & b \arrow[d, tail, "\exists_f(u)"] \\
a \arrow[r, "f"'] & c.
\end{tikzcd}$$

Proof.

There is an isomorphism $Sub(c)(\exists_f, v) \cong Sub(a)(u, v \cdot f)$, for any v. □

Implication

Using the right adjoint $\forall_u$:

$$\begin{tikzcd}
u \cap v \arrow[rr, tail] \arrow[dd, tail] & & b \arrow[dd, tail, "v"] \\
& x \arrow[dr, tail, "\forall_u(u \cap v)"'] & \\
a \arrow[rr, tail, "u"'] & & c,
\end{tikzcd}$$

let $(u \Rightarrow v) = \forall_u(u \cap v)$.

Explanation:

- $(\forall_u(u \cap v))(x) = \forall y \colon a.(u(y) = x) \Rightarrow (u \cap v)(y)$.
- But there is either zero or one such y.
- If zero, then $x \notin u$ and $(u \Rightarrow v)$ holds.
- If one, then $x \in u$ and $(u \Rightarrow v)(x) = (u \cap v)(x)$.

Ad hoc?

- The conditions above are modular, but somewhat *ad hoc*.
- There is a particular case that implies them all, and more:

Definition

A topos is a category $\mathcal{C}$

- with finite limits,
- equipped with an object Ω
- and a function $P \colon \mathcal{C}_0 \to \mathcal{C}_0$,
- with for each object c two isomorphisms

$$Sub(c) \cong \mathcal{C}(c, \Omega) \qquad \mathcal{C}(d \times c, \Omega) \cong \mathcal{C}(c, P(d))$$

natural in A.

Equivalent, elementary definition.

Toposes

- The logic of $\mathcal{H}_{\mathcal{C}}$ for a topos $\mathcal{C}$ is higher-order.
- Main examples of toposes: logic and sheaves.
- There is a characterisation of hyperdoctrines of the form $\mathcal{H}_{\mathcal{C}}$.
- There is a slightly weaker notion of hyperdoctrines, triposes, which canonically generate toposes.
- They are important for Boolean- or Heyting-valued sets.
- They are important for realisability.

Let's spare the definition for M. Hyland's lecture.