Lectures on Categorical Logic

Philip Scott
Department of Mathematics
University of Ottawa,
Ottawa, Ontario, K1N 6N5,
CANADA
scpsg@matrix.cc.uottawa.ca,
phil@site.uottawa.ca

1. Categories

A *category* $\mathcal{C}$ consists of

- 2 classes: *Objects* and *Arrows*
- 2 functions: $Arrows \underset{cod}{\overset{dom}{\rightrightarrows}} Objects$

Satisfying the following:

(Notation: we write $A \xrightarrow{f} B$ for: $f \in Arrows$, $dom(f) = A$ and $cod(f) = B$):

- There are *identity* arrows $A \xrightarrow{id_A} A$, for each object A,
- There is a partially-defined binary *composition* operation on arrows, denoted by $\circ$,

$$\frac{A \xrightarrow{f} B \quad B \xrightarrow{g} C}{A \xrightarrow{g \circ f} C}$$

(defined only when $dom(g) = cod(f)$) satisfying the following equations:

(i) $f \circ id_A = f = id_B \circ f$, where $A \xrightarrow{f} B$,

(ii) $h \circ (g \circ f) = (h \circ g) \circ f$, where $A \xrightarrow{f} B$, $B \xrightarrow{g} C$, $C \xrightarrow{h} D$

A category is called *large* or *small* depending upon whether its class of objects is respectively a proper class or a set, in the sense of Gödel-Bernays set theory. We denote by $\mathcal{C}(A, B)$ or $Hom_{\mathcal{C}}(A, B)$, called a *hom-set*, the collection of $\mathcal{C}$-arrows $A \to B$. A category is *locally small* if $\mathcal{C}(A, B)$ is a set, for all objects A, B.

Examples (Exercise: check the axioms !)

Set: This (large) category has the class of all sets as Objects, with all set-theoretic functions as Arrows. Similarly, the small category $\mathbf{Set}_{\text{fin}}$ of *finite* sets & functions.

Identity arrows and composition of arrows:

$id_A(x) = x$, for $x \in A$, $(g \circ f)(x) = g(f(x))$

Rel: This has the same objects as **Set**, but an arrow $A \xrightarrow{R} B$ is a binary relation $R \subseteq A \times B$. Here composition = relational product, i.e.

$$A \xrightarrow{R} B \xrightarrow{S} C = \{(a, c) \in A \times C \mid \exists b \in B (a, b) \in R \;\&\; (b, c) \in S$$

while the identity arrows $A \xrightarrow{id_A} A$ are given by the diagonal: $id_A = \Delta_A =_{def} \{(a, a) \mid a \in A\}$.

Universal Algebras: Objects = any equational class of algebras (e.g. semigroups, monoids, groups, rings, lattices, heyting or boolean algebras, $\cdots$). Arrows = homomorphisms, i.e. set-theoretic functions preserving the given structure. Composition and identities are lifted from **Set** (why?)

E.g. **Monoids:** structures $(M, \cdot, e_M)$ where M is a set, $M^2 \xrightarrow{\cdot} M$, $e \in M$ satisfying unit and associativity laws. Arrows = homs = functions preserving $\cdot$ and the unit e.

E.g. **Boole:** structures $(B, \wedge, \vee, \bar{\ }, 0, 1)$ where B is a set, $\vee, \wedge : B^2 \longrightarrow B$, $\bar{\ } : B \to B$, constants $0, 1 \in B$, satisfying equations of a boolean algebra. Arrows = functions preserving all the structure.

Vec$_\mathbf{k}$: Objects = vector spaces over field **k** and Arrows = linear maps. An important subcategory is **Vec**$_{fd}$ of finite dimensional **k**-vector spaces and linear maps. e.g. if **k** $= \mathbb{R}$, $\mathbb{R}^n \in$ **Vec**$_{fd}$.

Top: Here Objects = topological spaces and Arrows = continuous maps. (Exercise: identity maps are continuous. Composition of continuous maps is continuous).

Preord: A preordered set $(A, \leq)$ is a set A with a reflexive, transitive relation on it. **Preord** is the following large category:

Objects = pre-ordered sets

Arrows = monotone (= order-preserving) functions.

Similarly for **PO** = the category of posets.

ω-**CPO**: Important example arising in denotational semantics. Objects of ω-**CPO** are posets in which ascending countable chains $\cdots a_i \leq a_{i+1} \leq a_{i+2} \leq \cdots$ have suprema: i.e. we can form the l.u.b. $\bigvee\{a_i \mid i \in \mathbb{N}\}$ of an ascending chain $\{a_i\}$. Morphisms are poset maps preserving suprema of countable chains. Composition and identities are inherited from **PO**.

Many more sophisticated examples of categories arise in algebraic topology, algebraic geometry, homological algebra, and functional analysis.

ome very small categories:

One: The category with one object and one (identity) arrow.

Discrete Categories: (Essentially sets). Categories where the only arrows are identities. A set X becomes a discrete category, by letting the objects be the elements of X, and adding one identity arrow $x \xrightarrow{id_x} x$ for each $x \in X$. All (small) discrete categories arise in this way.

A monoid: A single monoid M gives a category with one object, call it $\mathcal{C}_M$, as follows: if the single object is $*$, we define $\mathcal{C}_M(*,*) = M$. Composition of maps is multiplication in the monoid. The monoid laws are exactly the category laws!

Conversely, note that every category $\mathcal{C}$ with one object corresponds to a monoid, namely $\mathcal{C}(*,*)$.

A preorder: A single preordered set $\mathbb{P} = (P, \leq)$ (where $\leq$ is reflexive & transitive) may be considered as a category:

Objects = the elements of $\mathbb{P}$. We define hom-sets by:

$$\mathbb{P}(a,b) = \begin{cases} \{*\} & \text{if } a \leq b \\ \emptyset & \text{if } a \not\leq b \end{cases}$$

Thus, given two objects $a, b \in \mathbb{P}$, there is at most one arrow from a to b; moreover, there is an arrow $a \to b \in \mathbb{P}$ exactly when $a \leq b$. In this case, the category laws are exactly the preorder conditions.

$a \xrightarrow{\leq} b$

Underlying multigraph of a category

A *graph* (= a directed multigraph with loops) is a pair of sets, together with two functions:

$$\textit{Arrows} \underset{cod}{\overset{dom}{\rightrightarrows}} \textit{Objects}$$

(more traditionally, $\textit{Edges} \underset{target}{\overset{source}{\rightrightarrows}} \textit{Vertices}$)

- Every category has an *underlying graph*, obtained by simply ignoring the other data beyond dom, cod.
- All vertices in the underlying graph of a category have loops (= identity maps).

$\therefore$ A category = Graph + composition law + identity edges + equations.

How do we form the free category generated by a graph? We use logical methods:

Deductive Systems and Freely Generated Categories (for logicians):

A *deductive system* $\mathcal{D}$ is a (labelled) graph (whose nodes are called *formulas* and whose edges are called *labelled sequents* or *labelled deductions*).

We are given:

(i) specified labelled edges (= "axioms")

(ii) operations on edges (= *"rules of inference"*) for generating new edges from old ones.

Logic Motivation: $A \xrightarrow{f} B$ means "f is a proof of the entailment $A \vdash B$"

Postulates:

Identity axioms For each formula A, an edge $A \xrightarrow{id_A} A$.

Cut Rule (for generating new edges from old ones):

$$\frac{A \xrightarrow{f} B \quad B \xrightarrow{g} C}{A \xrightarrow{g \circ f} C}\ cut$$

There may be additional axioms, operations on formulas, and/or additional rules of inference.

The above operations allow us to freely generate what logicians would call "labelled proof trees", or "proofs" for short.

Examples: Ax : $\{A \xrightarrow{f} B,\ C \xrightarrow{g} A\}$

$$\cfrac{\cfrac{\overset{ax}{A \xrightarrow{f} B} \quad B \xrightarrow{id_B} B}{A \xrightarrow{id_B \circ f} B}\ cut \qquad A}{\xrightarrow{(id_B \cdot f) \cdot g} B}\ cut$$

(Exercise: Given a set Ax of axioms, describe the set of labelled proof trees generated by Ax)

A *deductive system* $\mathcal{D}$ forms a category ("the category freely generated by $\mathcal{D}$) as follows :

Objects = formulas

Arrows = equivalence classes of "proofs", i.e. we form the class of all possible "proof trees", then we impose certain equations between them: equality "$\equiv$" is the congruence relation generated by the following equations:

$$\frac{A \xrightarrow{f} B \quad B \xrightarrow{id_B} B}{A \xrightarrow{id_B \circ f} B} \equiv A \xrightarrow{f} B$$

$$\frac{A \xrightarrow{id_A} A \quad A \xrightarrow{f} B}{A \xrightarrow{f \circ id_A} B} \equiv A \xrightarrow{f} B$$

$$\dfrac{A \xrightarrow{f} B \qquad \dfrac{B \xrightarrow{g} C \quad C \xrightarrow{h} D}{B \xrightarrow{h\circ g} D}}{A \xrightarrow{(h\circ g)\circ f} D} \qquad \equiv$$

$$\dfrac{\dfrac{A \xrightarrow{f} B \quad B \xrightarrow{g} C}{A \xrightarrow{g\circ f} C} \qquad C \xrightarrow{h} D}{A \xrightarrow{h\circ(g\circ f)} D}$$

i.e. $\equiv$ is the smallest equivalence relation R between "proofs" which identifies the proofs shown above and such that if fRf' and gRg' then $(g\circ f)R(g'\circ f')$, for composable arrows.

Operations on categories

Dualization: If $\mathcal{C}$ is a category, so is its *dual* $\mathcal{C}^{op}$, with the same objects, but whose ar- rows are reversed (i.e. interchange *dom* and *cod*).

Products: If $\mathcal{C}$, $\mathcal{D}$ are categories, so is their cartesian product $\mathcal{C} \times \mathcal{D}$, with the obvious structure: objects are pairs of objects, ar- rows are pairs of arrows, composition and identities are defined componentwise.

$(A, D) \xrightarrow{(f,g)} (A', D')$

Finally, we end with a useful notion:

A *subcategory* $\mathcal{C}$ of $\mathcal{B}$ is a category consisting of:

1. $ob(\mathcal{C}) \subseteq ob(\mathcal{B})$
2. $\mathcal{C}(A, B) \subseteq \mathcal{B}(A, B)$, for all $A, B \in ob(\mathcal{C})$
3. The operations of $\mathcal{C}$ are the restriction of those of $\mathcal{B}$. This means:
   (a) the identity arrows of $\mathcal{C}$ are identity arrows of $\mathcal{B}$
   (b) composition in $\mathcal{C}$ is restricted from $\mathcal{B}$.

$\mathcal{C}$ is a *full* subcategory of $\mathcal{B}$ if for all objects $A, B \in \mathcal{C}$, $\mathcal{C}(A, B) = \mathcal{B}(A, B)$.

E.g. The full subcateogry of **Set** of finite sets and functions, versus finite sets and injective functions.

Functors

Let $\mathcal{C}, \mathcal{D}$ be categories. A *functor* $F : \mathcal{C} \to \mathcal{D}$ is a pair $F = (F_{ob}, F_{arr})$, where

$$F_{ob} : Objects(\mathcal{C}) \to Objects(\mathcal{D})$$

and similarly for arrows , satisfying:

$$\frac{A \xrightarrow{f} B}{FA \xrightarrow{Ff} FB}$$

with equations: $$F(g \circ f) = F(g) \circ F(f)$$ $$F(id_A) = id_{FA}$$

A functor $F : \mathcal{C}^{op} \to \mathcal{D}$ is sometimes called *contravariant*. A contravariant functor F reverses the order of composition:

$$F(g \circ f) = F(f) \circ F(g)$$
$$F(id_A) = id_{FA}$$

Examples

1. *Forgetful (= Underlying) Functors.*

   U : **Posets** $\to$ **Set**, U : **Top** $\to$ **Set**, U : **Alg** $\to$ **Set** (where **Alg** is any category of universal algebras and homomorphisms between them).

   U : Objects $\mapsto$ Underlying Set (omitting the other structure). & similarly on arrows.

   Sometimes, one only forgets part of the structure, e.g. the forgetful functor

   U : **TopGrp** $\to$ **Grp**

   topological group and cont. group homs. $\mapsto$ underlying groups (+ underlying group homs)

2. *Representable (or Hom) Functors.* If $A \in \mathcal{C}$, we have the dual co- and contravariant homs:

(a) *Covariant Hom* : $\mathcal{C}(A,-) : \mathcal{C} \to$ **Set** given by:

$$B \mapsto \mathcal{C}(A,B)$$
$$A \xrightarrow{g} B \xrightarrow{f} C \mapsto \mathcal{C}(A,f) : \mathcal{C}(A,B) \to \mathcal{C}(A,C)$$
$$\text{where } \mathcal{C}(A,f)(g) = f \circ g.$$

Usual notation: $h^A = \mathcal{C}(A,-)$

(b) *Contravariant hom*: $\mathcal{C}(-,A) : \mathcal{C}^{op} \to$ **Set** given by:

$$B \mapsto \mathcal{C}(B,A)$$
$$B \xrightarrow{f} C \mapsto \mathcal{C}(f,A) : \mathcal{C}(C,A) \to \mathcal{C}(B,A)$$
$$\text{where } \mathcal{C}(A,f)(g) = g \circ f.$$

Usual notation: $h_A = \mathcal{C}(-,A)$

3. *Co- and Contravariant Powerset Functors.* Let $\mathcal{P}(A) =$ the set of subsets of A. This is the object-part of two functors.

(a) *Covariant Powerset* $\mathcal{P}$: **Set** $\to$ **Set** given by: if $A \xrightarrow{f} B$, then $\mathcal{P}(A) \xrightarrow{\mathcal{P}(f)} \mathcal{P}(B)$
$$S \longmapsto f[S]$$
by direct image: $\mathcal{P}(f)(S) = f[S]$, for $S \subseteq A$.

(b) *Contravariant Powerset*, denoted $\mathcal{P}^*$: $\mathbf{Set}^{op} \longrightarrow$ **Set** given by: if $A \xrightarrow{f} B$, then $\mathcal{P}(B) \xrightarrow{\mathcal{P}^*(f)} \mathcal{P}(A)$ given by: $\mathcal{P}^*(f)(T) = f^{-1}(T)$, for $T \subseteq B$, where
$f^{-1}(T) = \{a \in A \mid f(a) \in T\}$

4. *Free Algebra Functors.* F : **Set** $\to$ **Alg**, where $F(X) =$ the free algebra generated by set X (e.g. **Alg** can be **Mon**, **Grp**, **Vec**)

5. *Identity and Inclusion Functors*: For example, Id : **Set** $\to$ **Set**, and the evident inclusion Inc : $\mathbf{Vec}_{fd} \hookrightarrow$ **Vec** of finite dimensional vector spaces among all vector spaces.

6. Exercise: If $\mathbb{P}, \mathbb{P}'$ are preorders, qua categories, a functor $F : \mathbb{P} \to \mathbb{P}'$ is the same as a monotone (= order-preserving) map.

7. *Dual Spaces*: Let $V \in \mathbf{Vec}$ and $V^{\perp} = Lin(V, \mathbf{k})$, the dual space of V. Exercise: show there are two functors: $(-)^{\perp} : \mathbf{Vec}^{op} \to \mathbf{Vec}$ and $(-)^{\perp\perp} : \mathbf{Vec} \to \mathbf{Vec}$.

Natural Transformations

Given functors $F, G : \mathcal{C} \to \mathcal{D}$, a *natural transformation* is a family of arrows $\{\theta_C : FC \to GC \mid C \in \mathcal{C}\}$ satisfying: for every $f : C \to D$, the following diagram commutes:

$$\begin{tikzcd} FC \arrow[r, "\theta_C"] \arrow[d, "Ff"'] & GC \arrow[d, "Gf"] \\ FD \arrow[r, "\theta_D"] & GD \end{tikzcd}$$

Given n-ary functors $F, G : \mathcal{C}^n \to \mathcal{D}$, a family of arrows $\alpha_{A_1,\cdots,A_n} : F(A_1, \cdots, A_n) \to G(A_1, \cdots, A_n)$ is said to be *natural in A_i* if fixing all the other arguments $A_j, j \neq i$, the resultant family $\alpha_{\cdots,A_i,\cdots} : F(\cdots, A_i, \cdots) \to G(\cdots, A_i, \cdots)$ determines a natural transformation between functors $\mathcal{C} \to \mathcal{D}$ with respect to the ith argument as variable.

Examples

1. *Double Dual*: Define $\theta : Id \dot{\to} (-)^{\perp\perp} : \mathbf{Vec} \to \mathbf{Vec}$, where $\theta_V : V \to V^{\perp\perp}$ is given by:

$$\theta_V(x)(f) = f(x) \quad \text{for} \quad f \in V^{\perp}, x \in V.$$

Exercise:

(a) θ is well-defined and a n.t.

(b) (Hard) θ_V is an isomorphism if and only if V is finite dimensional

If V is indeed finite dimensional, there is no *natural* isomorphism $\eta : Id \to (-)^{\perp}$, even though for each V, $V \cong V^{\perp}$ in this case.

The reason is that this latter isomorphism depends on a choice of basis.

2. *Functor Categories*: Let $\mathcal{C}, \mathcal{D}$ be categories. Let $Funct(\mathcal{C}, \mathcal{D})$ be the category whose objects are functors from $\mathcal{C}$ to $\mathcal{D}$, and whose arrows are natural transformations between them, where we compose natural transformations as follows: given $F, G, H \in Funct(\mathcal{C}, \mathcal{D})$, define

$$FA \xrightarrow{(\psi\theta)_A} HA = FA \xrightarrow{\theta_A} GA \xrightarrow{\psi_A} HA$$

for each object $A \in \mathcal{C}$. In particular, if $\mathcal{C}$ is small, and $\mathcal{D} = \mathbf{Set}$, the category $Funct(\mathcal{C}^{op}, \mathcal{D}) = \mathbf{Set}^{\mathcal{C}^{op}}$ is called the category of *presheaves on* $\mathcal{C}$.

If $\mathcal{C}$ is the small category with two objects and two non-identity arrows, $\bullet \rightrightarrows \bullet$, we identify $\mathbf{Set}^{\mathcal{C}^{op}}$ with the category of *small graphs*, denoted **Grph**.

- $\mathrm{Set}^{\mathbb{P}}$ ($\mathbb{P}$ a poset)

= Sets $\{X_p\}_{p \in \mathbb{P}}$ s.t.

if $p \le q$, $\exists$ map $X_p \xrightarrow{f_{pq}} X_q$

(= Kripke models of shape $\mathbb{P}$)

- Set^M (M a monoid)

$F \in \mathrm{Set}^M$ = monoid hom $M \to (X^X, \circ)$

where $X = F(*)$, $*$ = the one object of M

and $(X^X, \circ)$ = the endomorphism monoid of X. Equivalently,

an <u>M-Set</u> = <u>M-action</u> on X

$\therefore$ $M \times X \xrightarrow{\bullet} X$ s.t. $(mn) \cdot x = m \cdot (n \cdot x)$
$e \cdot x = x$

<u>Exercise</u>: What is a hom of M-Sets? (cf. <u>natural transf^n</u>)

3. There is a category **Cat** of small categories and functors between them. There is a forgetful functor $U : \mathbf{Cat} \to \mathbf{Grph}$ which associates to every small category $\mathcal{C}$ its underlying graph.

Adjoints and Equivalences

An arrow in a category is an *iso* if it has a two-sided inverse. This corresponds to the usual mathematical notion of "isomorphism" in most familiar categories. In the case of functor categories, we obtain the following related notions:

- *Natural Isomorphisms*: A natural transformation $F \xrightarrow{\theta} G$ is a natural isomorphism if, for each A, $FA \xrightarrow{\theta_A} GA$ is an iso.

- *Natural Equivalence*: A pair of functors $\mathcal{C} \underset{G}{\overset{F}{\rightleftarrows}} \mathcal{D}$ is a natural equivalence of categories if there are natural isomorphisms $GF \cong Id_{\mathcal{C}}$ and $FG \cong Id_{\mathcal{D}}$. We shall see many examples of this notion below.

Most mathematical duality theories, as in the case of the famous representation theorems of Stone, Gelfand, and Pontrjagin, amount to "contravariant" natural equivalences $\mathcal{C} \cong \mathcal{D}^{op}$.

Barr's book on $*$-autonomous categories, which analyzes such duality theories, is an important source of concrete models for (fragments of) linear logic.

Adjoint Functors

One of the most important concepts in category theory. Given functors

$$\mathcal{D} \underset{U}{\overset{F}{\rightleftarrows}} \mathcal{C}$$

we say F *is left adjoint to* U (denoted $F \dashv U$) if there is a natural isomorphism

$$\mathcal{D}(FC, D) \cong \mathcal{C}(C, UD) \ .$$

i.e., there is a family of arrows

$$\alpha = \{\alpha_{C,D} : \mathcal{D}(FC, D) \to \mathcal{C}(C, UD)\}$$

which determines a natural isomorphism of functors (natural in C and D),

$$\alpha_{-,-} : \mathcal{D}(F-,-) \xrightarrow{\cong} \mathcal{C}(-,U-)$$

qua functors $\mathcal{C}^{op} \times \mathcal{D} \to$ **Set**. This isomorphism determines a natural bijection of arrows

$$\frac{FC \to \overset{FC}{\cancel{D}} \text{ in } \mathcal{D}}{C \to UD \text{ in } \mathcal{C}.}$$

Universal mapping property view.

$\exists!\ FC \xrightarrow{h} D$

$$\begin{tikzcd}
UFC \arrow[dr, "U(h)"] & \\
& UD \\
C \arrow[uu, "\eta_c"] \arrow[ur, "g"'] &
\end{tikzcd}$$

Fact: Notions defined by universal mapping properties are unique up to isomorphism.

Adjoint functors abound in mathematics. Lawvere has used this in an **attempted** axiomatic foundation for large parts of mathematics.

Adjoints

1. *Free Structures:* Typical examples are left adjoints to forgetful functors, which determine "free" structures: e.g. **Alg** $\underset{U}{\overset{F}{\leftrightarrows}}$ **Se** in which $F(X)$ is the free (universal) algebra generated by the set X.

Concrete Example: **Mon** $\underset{U}{\overset{F}{\leftrightarrows}}$ **Set**

Here **Mon** is the category of monoids and monoid homs. $F(X) = X^* =$ the free monoid on $X =$ all finite lists or words (including the empty list) of elements of X. X^* is a monoid, by letting multiplication = concatenating lists.

$\eta_X : X \to UFX$ is "inclusion of generators" : map $x \mapsto \langle x \rangle$, where $\langle x \rangle$ is the word of length one containing symbol x.

The universal property of adjoint functors reduces to the familiar one for free algebras.

$$\begin{tikzcd} FX \arrow[r, dashed, "\exists! h \in \mathrm{Mon}"] & M \end{tikzcd}$$

$$\begin{tikzcd} UFX \arrow[dr, "Uh"] & \\ X \arrow[u, "\eta_X"] \arrow[r, "f"'] & UM \end{tikzcd}$$

Another such example: $\mathbf{Graph} \xrightarrow{F} \mathbf{Cat}$ giving *the free (small) category generated by a (small) graph.*

2. *Galois Correspondences:* Consider two pre-orders as categories, with a pair of adjoint functors (= monotone maps) between them : $(P,\leq)\underset{G}{\overset{F}{\rightleftarrows}}(Q,\leq)$. Then $F \dashv G$ means: $F(a) \leq b$ iff $a \leq G(b)$, for all $a \in P, b \in Q$. Let $j = GF : Q \to Q$. This gives a monotone *closure operator* satisfying: (i) $a \leq j(a)$ and (ii) $j^2(a) \leq j(a)$, for all $a \in P$

Lawvere's slogan: *many categorical notions arise as adjoints to previously defined functors.*

We now illustrate this viewpoint by introducing *products, coproducts, and function spaces.*

Let <u>One</u> be the category with one object and one arrow. $* \circlearrowleft \mathrm{id}_*$

$\exists!$ functor $\mathcal{C} \xrightarrow{\ !\ } \underline{\mathrm{One}}$

By Lawvere's principle, we examine adjoints to !

$$\begin{tikzcd} \mathcal{C} \arrow[r, "!"] & \underline{\mathrm{One}} \arrow[l, bend right, "L"'] \arrow[l, bend left, "R"] \end{tikzcd}$$

Postulate L, R exist

$\mathcal{C}(L(*), C) \cong \underline{\mathrm{One}}(*, !(C)) \cong \{\mathrm{id}_*\}$

$\therefore L(*) \in \mathrm{ob}(\mathcal{C})$ is an object $\bot$ s.t. $\exists!$ arrow $\bot \to C$ for any C, if it exists = <u>Initial</u> <u>Obj</u>

Dually, $R(*) \in \mathrm{ob}(\mathcal{C})$ is an object $\top$ such that $\exists!$ arrow $C \longrightarrow \top$ for any C

$\top$ = terminal object

<u>Examples</u>:

<u>Set</u> : $\bot = \emptyset$, $\top = \{*\}$

<u>Vec</u> : $\bot = \top = \{0\}$

<u>Mon</u>, <u>Group</u> : $\bot = \top = \{e\}$

<u>Exercise</u>: Find categories that don't have initial and/or terminal objects. Also check poset $\mathbb{P}$, qua ca

If $\mathcal{C}$ is a category, so is $\mathcal{C} \times \mathcal{C}$.

<u>Objects</u>: pairs (A,B) of objects

<u>Maps</u>: pairs $(A,B) \xrightarrow{(f,g)} (C,D)$ of maps.

Composition & identities <u>component-wise</u>.

<u>Diagonal Functor</u>: $\mathcal{C} \xrightarrow{\Delta} \mathcal{C} \times \mathcal{C}$

$$A \longmapsto (A,A)$$

$$A \xrightarrow{f} B \longmapsto \begin{array}{c}(A,A)\\ \downarrow{\scriptstyle (f,f)}\\ (B,B)\end{array}$$

By Lawvere's Principle, we postulate existence of adjoints

$$L \dashv \Delta \dashv R.$$

To say: $\exists R$:
$$\begin{tikzcd} \mathcal{C} \arrow[r, "\Delta"{name=D}] & \mathcal{C} \times \mathcal{C} \arrow[l, bend left, dotted, "R"{name=R}] \arrow[from=D, to=R, phantom, "\perp"] \end{tikzcd}$$

means: $\mathcal{C} \times \mathcal{C}(\Delta(C), (A,B)) \cong \mathcal{C}(C, R(A,B))$

Check: This is equivalent to: There's a natural iso

$$\mathcal{C}(C,A) \times \mathcal{C}(C,B) \cong \mathcal{C}(C, R(A,B))$$

natural in A, B, C.

We write $R(A,B) = A \times B$.

$\therefore$ Get natural bij^{n} $\dfrac{C \to A, \ C \to B}{C \to A \times B}$

To set up bijection

$$\mathcal{C}(C,A) \times \mathcal{C}(C,B) \cong \mathcal{C}(C, A\times B)$$

On RHS, set $C \equiv A \times B$

<u>Map</u>: $\mathrm{id}_{A\times B} \longmapsto (A\times B \xrightarrow{\pi_1} A,\ A\times B \xrightarrow{\pi_2} B)$

Bijection says,

$$\frac{\forall\ C \xrightarrow{f} A,\ C \xrightarrow{g} B}{\exists!\ C \xrightarrow{h} A\times B} \quad \text{s.t.} \quad \boxed{\begin{matrix}\pi_1 \circ h = f \\ \pi_2 \circ h = g\end{matrix}}$$

$$\begin{tikzcd}
 & C \arrow[dl, "f"'] \arrow[d, dotted, "\exists! h"] \arrow[dr, "g"] & \\
A \arrow[r, phantom, "="] & A\times B \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & B
\end{tikzcd}$$

<u>Example</u>

- <u>Set</u> : $A\times B$ = Cartesian product $= \{(a,b) \mid a\in A, b\in B\}$

$A\times B \xrightarrow{\pi_1} A$ is $\pi_1(a,b) = a$

similarly for π_2.

$$\frac{C \xrightarrow{f} A \quad C \xrightarrow{g} B}{\exists!\ C \xrightarrow{h} A\times B}$$

$h \equiv \langle f, g$ where $\langle f,g\rangle(c) = (f(c), g(c$ $\forall c \in C$.

<u>Exercise</u>

In any category with products, can form

$$\frac{A \xrightarrow{f} C \quad B \xrightarrow{g} D}{A\times B \xrightarrow{f\times g} C\times D}$$

where $A\times B \xrightarrow{f\times g} C\times D \xrightarrow{\pi_1} C = A\times B \xrightarrow{\pi_1} A \xrightarrow{f}$

$A\times B \xrightarrow{f\times g} C\times D \xrightarrow{\pi_2} D = A\times B \xrightarrow{\pi_2} B \xrightarrow{g}$

Prove $f\times g$ exists. In Sets = ?

- Any eq^{n}l class of algebras has products (e.g. Monoids, Groups, Rings, Boolean Algebras, Lattices, etc.) Given by taking the Cartesian Product of underlying sets & "ptwise" operations on tuples.
- Top, the category of topological spaces & cont. maps has products – using the product topology.
- Functor Categories : $Funct(\mathscr{C}^{op}, Set)$

pt-wise : $(F \times G)(A) = FA \times GA$

$$(F \times G)(f) = F(f) \times G(f)$$

f. $F, G \in Funct(\mathscr{C}^{o}, SET)$

- A poset $(\mathbb{P}, \leq)$ has products (as a category) if :
  - $\forall a, b \in \mathbb{P}, \ \exists \ a \wedge b \in \mathbb{P}$

s.t.
  - $a \wedge b \leq a$, $a \wedge b \leq b$
  - for all $c \in \mathbb{P}$,

$$\frac{c \leq a \quad c \leq b}{c \leq a \wedge b}$$

$\therefore \boxed{a \wedge b = g.l.b\{a, b\}}$

- Dually, $(\mathbb{P}, \leq)$ has coproducts

if: $\forall a, b \in \mathbb{P}, \ \exists \ a \vee b \in \mathbb{P}$.

s.t.
  - $a \leq a \vee b$, $b \leq a \vee b$
  - for all $c \in \mathbb{P}$

$$\frac{a \leq c, \ b \leq c}{a \vee b \leq c}$$

$\therefore \boxed{a \vee b = l.u.b.\{a, b\}}$

Dually, to say $\exists L$:
$$\begin{tikzcd} \mathcal{C} \arrow[r, "\Delta"'] & \mathcal{C}\times\mathcal{C} \arrow[l, bend right, dotted] \end{tikzcd}$$

Says $\mathcal{C}$ has binary coproducts

Writing $L(A,B) = A+B$,

$$\mathcal{C}(A+B, C) \cong \mathcal{C}(A,C) \times \mathcal{C}(B,C)$$

natural in A, B, C

$\therefore$ $\exists$ bijection $\dfrac{A \to C \quad B \to C}{A+B \longrightarrow C}$

As above, $\exists$ $A \xrightarrow{in_L} A+B$, $B \xrightarrow{in_R} A+B$

s.t. $\forall$ $A \xrightarrow{u} C$, $B \xrightarrow{v} C$,

$\exists!\, h$

$$\begin{tikzcd} & C & \\ A \arrow[ur, "u"] \arrow[r, "in_L"'] & A+B \arrow[u, dotted, "\exists! h"] & B \arrow[l, "in_R"] \arrow[ul, "v"'] \end{tikzcd}$$

<u>Coproducts</u> : In <u>Sets</u>

$A+B$ = disjoint union $= (A\times\{0\})\cup(B\times\{$
(up to $\cong$!)

$A \xrightarrow{in_\ell} A+B \quad : \quad a \mapsto (a,0)$

$B \xrightarrow{in_r} AB \quad : \quad b \mapsto (b,1)$

$$\dfrac{A \xrightarrow{f} C \qquad B \xrightarrow{g} C}{A+B \xrightarrow{h=[f,g]} C}$$

$$[f,g](x) = \begin{cases} f(a) & \text{if } x=(a,0) \\ g(b) & \text{if } x=(b,1) \end{cases}$$

In <u>Vec</u> : $V \times W \cong V + W$

(products and coproducts coincide

(Exercise)

Def^{n}: A cartesian category is a category with terminal object and binary ($\therefore$ all finite) products.

$\therefore$ $\mathcal{C}$ cartesian means:

- There's a terminal object T
- Given objects A, B, we can form $A \times B$ s.t.

$$\mathcal{C}(A, T) \cong \{*\}$$

$$\begin{array}{ccc} \mathcal{C}(C,A) \times \mathcal{C}(C,B) & \cong & \mathcal{C}(C, A\times B) \\ (f, \quad g) & \longmapsto & \langle f, g\rangle \end{array}$$

We now give an equational presentation of such cats:

- $A \xrightarrow{!^{A}} T$ $\quad$ { (unique) map to 1

$$\left.\begin{array}{l} \bullet\ A\times B \xrightarrow{\pi_1^{A,B}} A \\ \quad A\times B \xrightarrow{\pi_2^{A,B}} B \end{array}\right\} \text{Proj}^{\underline{n}}\text{s}$$

- $$\frac{C \xrightarrow{f} A \qquad C \xrightarrow{g} B}{C \xrightarrow{\langle f, g\rangle} A \times B}$$

Eq$^{\underline{n}}$s

- $f = !^{A}$, any $f : A \to T$
- $\pi_1 \circ \langle f, g\rangle = f \quad : C \to A$
  $\pi_2 \circ \langle f, g\rangle = g \quad : C \to B$
- $\langle \pi_1 \circ h, \pi_2 \circ h\rangle = h \ : C \to A\times B$

Exercise: This sets up required natural isos of cartesian cats.

Let $\mathcal{C}$ be a cartesian category. There is a functor

$$\mathcal{C} \times \mathcal{C} \xrightarrow{\times} \mathcal{C}$$

$$\begin{tikzcd}
(A,B) \arrow[r, mapsto] \arrow[d, "{(f,g)}"] & A \times B \arrow[d, "{\langle f\pi_1, g\pi_2 \rangle \overset{=}{\text{def}} f \times g}"] \\
(A',B') \arrow[r, mapsto] & A' \times B'
\end{tikzcd}$$

Fix $A \in \mathcal{C}$. $\therefore \; - \times A : \mathcal{C} \to \mathcal{C}$

A cartesian category $\mathcal{C}$ is <u>cartesian closed</u> if $- \times A$ has a right adjoint $- \times A \dashv (-)^A$

$\therefore \exists$ bijection $\dfrac{C \times A \longrightarrow B}{C \longrightarrow B^A}$

$\therefore \exists$ distinguished arrow $B^A \times A \xrightarrow{ev} B$ s.t.

$$\frac{\forall f : C \times A \longrightarrow B}{\exists!\, f^* : C \longrightarrow B^A} \quad \text{s.t.}$$

$$\begin{tikzcd}
 & C \times A \arrow[dl, "f^* \times id_A"'] \arrow[dr, "f"] & \\
B^A \times A \arrow[rr, "ev"] & & B
\end{tikzcd}$$

<u>Exercise</u>: We may equationally specify $\mathcal{C}(C \times A, B) \cong \mathcal{C}(C, B^A)$ as follows:

$$C \times A \xrightarrow{\langle f^* \pi_1, \pi_2 \rangle} B^A \times A \xrightarrow{ev^{A,B}} B$$

$$= \quad C \times A \xrightarrow{f} B \qquad \underline{\text{BETA}}$$

$$\left(C \times A \xrightarrow{\langle g\pi_1, \pi_2 \rangle} B^A \times A \xrightarrow{ev} B \right)^*$$

$$= \quad C \xrightarrow{g} B^A \qquad \underline{\text{ETA}}$$

Exercise: (BETA), (ETA) guarantee a natural bijection

$$\mathcal{C}(C \times A, B) \cong \mathcal{C}(C, B^A)$$

$$f \longmapsto f^*$$

$$ev \langle g\pi_1, \pi_2 \rangle \longleftarrow g : C \to B^A$$

Examples of CCC's

- **Sets**: Let B^A = the set of all functions $A \to B$.

$$\frac{C \times A \xrightarrow{f} B}{C \xrightarrow{f^*} B^A}\ \text{curry} \qquad f^*(c)(a) = f(c,a)$$

and $B^A \times A \xrightarrow{ev} B$

$$ev(f, a) = f(a)$$

-

2) ω-CPO (Fund. Example in CS)

Objects: posets $(P, \leq)$ in which countable ascending chains $a_0 \leq a_1 \leq a_2 \leq \cdots\cdots$ $(a_i \in P)$ have suprema, $\bigvee_{i \in \mathbb{N}} a_i$ (= l.u.b.'s).

arrows: Order-preserving maps $P \xrightarrow{f} Q$ s.t. $f(\bigvee_{i \in \mathbb{N}} a_i) = \bigvee_{i \in \mathbb{N}} f(a_i)$ for chains $\{a_i\}$ in P.

Products: $A \times B$ with pointwise structure (& sups defined ptwise $\bigvee_n (a_n, b_n) = (\bigvee_n a_n, \bigvee_n b_n)$.)

Function Spaces: $B^A = \mathrm{Hom}(A, B)$

$f \leq g$ iff $\forall a \in A\ (f(a) \leq g(a))$

$f_1 \leq f_2 \leq f_3 \leq \cdots$ & $(\bigvee_n f_n)(a) = \bigvee_n \{f_n(a)\}$

A **preorder** $(P, \leq)$ forms a category $\mathbb{P}$, with $|\mathbb{P}|$ = The set P, and

$$\mathbb{P}(a,b) = \begin{cases} \{*\} & \text{if } a \leq b \\ \emptyset & \text{else} \end{cases}$$

$\therefore$ $\exists$ arrow $a \to b$ in $\mathbb{P}$ iff a in

$\mathbb{P}$ has **products** means exac $\mathbb{P}$ is a **$\wedge$-semilattice with**:

- $a \leq \top$
- $a \wedge b \leq a$, $a \wedge b \leq b$.
- $$\frac{c \leq a \quad c \leq b}{c \leq a \wedge b}$$

$\mathbb{P}$ is **CCC** means: $\mathbb{P}$ is a **Heyting semilattice** i.e. $\exists$ an operation $\Rightarrow : P \times P \to P$ satisfying:

- $(a \Rightarrow b) \wedge a \leq b$
- $$\frac{c \wedge a \leq b}{c \leq a \Rightarrow b}$$

($\therefore$ $a \Rightarrow b =$ the largest element $c \in P$ such that $c \wedge a \leq b$)

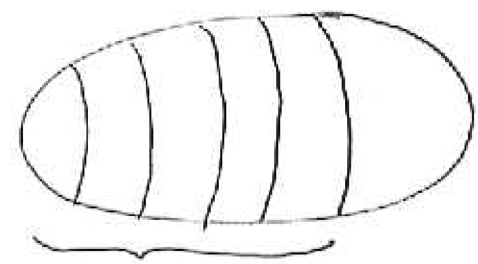
__Ex__: $P = \mathcal{O}(X) =$ the open sets of a top. space X.

$(P, \subseteq)$ forms a ccc, with

- $U \wedge V = U \cap V$
- $U \Rightarrow V = \mathrm{int}((X - U) \cup V)$

Per (IN)

A per (partial equiv. rel$^{\underline{n}}$) is a symm., transitive rel$^{\underline{n}}$ on $\mathbb{N}$. Picture: $R \subseteq \mathbb{N} \times \mathbb{N}$ s.t. $R \restriction \{x \mid (x,x) \in R\}$ is an e.r.



$\mathrm{dom}_R = \{x \mid (x,x) \in R\}$

Consider (Turing Machine) computable partial f$^{\underline{n}}$s $\varphi_e : \mathbb{N} \to \mathbb{N}$. Identify e with φ_e.

Category $Per(\mathbb{N})$

<u>Objects</u>: pers on $\mathbb{N}$

<u>arrows</u>: Equivalence classes of partial computable $f^{\underline{n}}s$

e represents an arrow $R \to S$ iff $\forall m,n \,(m R n \Rightarrow \varphi_e(m)\downarrow,\ \varphi_e(n)\downarrow \ \&\ \varphi_e(m)\, S\, \varphi_e(n))$

Consider $e \sim e'$ if

$\forall m,n \,[\, mRn \Rightarrow \varphi_e(m), \varphi_{e'}(n)\downarrow$ and $\varphi_e(m)\, S\, \varphi_{e'}(n)\,]$

Using the recursive bij$^{\underline{n}}$ $\mathbb{N}\times\mathbb{N} \cong \mathbb{N}$, Define $R\times S$ by: $\langle m,n\rangle\, R\times S\, \langle m',n'\rangle$ iff mRm' <u>and</u> nSn'.

S^R = the set of per maps $R \to S$, under $\sim$ of per maps.

The fact that this forms a CCC arises from some elementary recursive $f^{\underline{n}}$ theory ("S-m-n Theorem")

Pers are interesting:

- Models of inheritance, subtypin[g] higher-order λ-calculi, etc.
- arbitrary intersect$^{\underline{n}}$s of Pers are Pers: models notions of $\forall$-quantifiers via $\cap$'s.

Propositional $\{\wedge, \top\}$-Logic as a deductive System

Formulas: freely generated from atoms

$$A ::= \text{atom} \mid \top \mid A_1 \wedge A_2$$

Deductions:

$$A \overset{id_A}{\vdash} A$$

$$\frac{A \overset{f}{\vdash} B \quad B \overset{g}{\vdash} C}{A \overset{g \cdot f}{\vdash} C}\ \text{cut}$$

$$A \wedge B \overset{\pi_1}{\vdash} A \ , \quad A \wedge B \overset{\pi_2}{\vdash} B$$

$$A \overset{!A}{\vdash} \top \ , \quad \frac{C \overset{f}{\vdash} A \quad C \overset{g}{\vdash} B}{C \overset{\langle f, g \rangle}{\vdash} A \wedge B}\ \wedge$$

Free Cartesian Cat. gen by a set $\mathcal{X}$ of atoms:

impose equations of cartesian categories on proofs

- Category
- products & terminal object

∴ An arrow is $A \xrightarrow{[f]_=} B$, where $A \overset{f}{\vdash} B$.

$$\begin{tikzcd} \mathrm{Cart} \arrow[r, "U"] \arrow[r, phantom, shift right=2, "\top"] & \mathrm{Set} \arrow[l, bend left, "F"] \end{tikzcd}$$

$F(\mathcal{X})$ = free cart. category gen. by $\mathcal{X}$

Intuitionistic Prop^{n}l Calc.
of $\{\wedge, \Rightarrow, T\}$

ormulas : freely generated

$$A ::= \text{atom} \mid T \mid A_1 \wedge A_2 \mid A_1 \Rightarrow A_2$$

roofs : labelled entailments

Add to $\{\wedge, T\}$-fragment :

$$(A \Rightarrow B) \wedge A \vdash^{ev^{A,B}} B$$

$$\frac{C \wedge A \vdash^{h} B}{C \vdash^{h^*} A \Rightarrow B}$$

$\Bigg\}$ Schem. for all formulae A, B, C

q^{n}s between proofs = CCC eq^{n}s.

Free CCC gen by a set $\mathcal{X}$ of atomic types : $\mathcal{F}_{\mathcal{X}}$

Objects : formulas in $\{\wedge, \Rightarrow, T\}$ generated from atoms $\mathcal{X}$.

Arrows : An arrow $A \xrightarrow{[f]} B$ is an equivalence class of proofs $A \vdash^{f} B$ (where $\equiv$ is the finest equiv. rel^{n} satisfying the CCC eq^{n}s)

Additional Data Types

Suppose we wish to add coproducts & $\perp$ to CCC's :

Biccc's $\mathbb{B}$: Add to objects & arrows new formation rules:

Objects $\perp \in |\mathbb{B}|$, $\dfrac{A, B \in |\mathbb{B}|}{A+B \in |\mathbb{B}|}$

Arrows : (Dual to products)

Add :
- $\perp \xrightarrow{\Box_A} A$
- $A \xrightarrow{in_1} A+B$, $B \xrightarrow{in_2} A+B$
- $\dfrac{A \xrightarrow{f} C \quad B \xrightarrow{g} C}{A+B \xrightarrow{[f,g]} C}$

Eq^{n}s : Dual to products and T .

Logic Level : Biccc's correspond to the logic of $\{\wedge, \Rightarrow, T, \vee, \perp\}$

Formulas : add to formation rules : $\perp$, $A \vee B$.

Proofs : Add to axioms

Axioms $\left\{ \begin{array}{l} \perp \vdash^{\Box_A} A \\ A \vdash^{in_1} A+B \qquad B \vdash^{in_2} A+B \end{array} \right.$

Rule $\left\{ \dfrac{A \vdash^{f} C \qquad B \vdash^{g} C}{A \vee B \vdash^{[f,g]} C} \right.$

Eq^{n}s : Biccc eq^{n}s .

Propositions-as-Types

Ded. (Int.) System	F^{n}al Lang.	Cat/Alg.
Prop^{n} Logic $\wedge, \supset, \top$	Typed λ-Calculus	C.C.C.'s
First Order Logic	Martin-Löf Type theories	Locally C.C.C.'s.
2^{nd} Order Prop^{n} Logic $\forall, \wedge, \supset, \top$	Polymorphic λ-Calculi [Girard 71, Reynolds 74]	Indexed Cats [Seely 87]
H.O.L. (= Topos theory)	F_ω [Girard 72]	"

$$
\begin{tikzcd}
 & \boxed{\text{CARTESIAN CLOSED CATEGORIES}} \arrow[dl, shift left, "1"'] \arrow[dr, shift left, "2"] & \\
\boxed{\text{TYPED } \lambda\text{-CALCULI}} \arrow[ur, shift left] \arrow[rr, shift right, "3"'] & & \boxed{\text{DEDUCTIVE SYSTEMS } \wedge, \supset, \top, \text{ EQUATION}} \arrow[ul, shift left] \arrow[ll, shift right]
\end{tikzcd}
$$

1, 2 Lambek [1969]

3 { Curry [1958],
Howard, de Bruijn [1969]

Typed Lambda Calculi

A term language for describing arrows in ccc's ("internal language of ccc's")

- A basic f^{n}l language – programming paradigm
- A term calculus for proofs in intuitionistic calculi.

Basic IDEA: If $\varphi(x)$ is some expression, the function $x \mapsto \varphi(x)$ is written $\lambda x.\, \varphi(x)$. If we apply this f^{n} to argument "a":

$$(\lambda x \varphi(x))(a) = \varphi[a/x]$$

E.g. $(\lambda x.\, x^3)(a) = a^3$

Typed λ-Calculi

Two views common in CS:

- Church View: all vbls & terms have an explicit type associated with them

  e.g. $x_i^A : A$, $f : B^A$, ...

Given a term $t : A$, we may interpret it as saying

- t is a term of type A
- t is an element of "set" A
- t is a proof of formula A
- t is a functional program meeting specification A
- t is an arrow with codomain A

A bit closer to CS practice is

<u>Curry View</u>: vbls, terms, etc. are untyped (e.g. $\lambda x . x$), but there is a typing procedure (hopefully, an explicit algorithm) to assign types to terms.

Once initial typing (in Curry style) is assigned to vbls, usually Curry & Church rules are identical.

$\therefore$ We stick to Church-style.

(Q: Is there a nice categorical story of Church vs Curry typing?)

<u>Typed</u> <u>Lambda</u> <u>Calculus</u>

<u>Types</u>: Freely generated from a set of <u>atoms</u> by the c.c.c. operations: 1, $(-)\times(-)$, $(-)^{(-)}$

i.e. $A ::= 1 \mid \text{Atom} \mid A_1 \times A_2 \mid A_1^{A_2}$

<u>Terms</u>: Freely generated from vbls & constants by the following rules (write $t : A$ for "t is a term of type A"

- $\forall$ types A, ∞ly many vbls $x_i^A : A$
- $\forall$ types A, B, constants

$$\pi_1^{A,B} : A^{(A\times B)}, \quad \pi_2^{A,B} : B^{(A\times B)}$$

$$ev^{A,B} : B^{(B^A \times A)}, \quad * : 1$$

- closure under rules

$$\frac{t : B^A \quad a : A}{ev(\langle t, a\rangle)} \qquad \frac{a : A \quad b : B}{\langle a, b\rangle : A \times B}$$

$$\frac{\varphi(x) : B \quad x : A \ (\text{a vbl.})}{\lambda x{:}_A\, \varphi(x) : B^A}$$

$\lambda x{:}_A\, \varphi(x)$ is a quantifier, binding x. Define as usual: $FV(t)$, $BV(t)$ of set of free & bnd vbls in term t.

Write $t'a$ or just $t\,a$ for $ev(\langle t, a\rangle)$ as above.

- identify terms up to change of bnd vbls. Define substitution as usual, so that $t[a/x]$ is only well-defined if $FV(a) \cap BV(t) = \emptyset$, perhaps after change of bnd vbls.

λ-Calculus Eq^{n}s-in-Context

Let $t_1, t_2 : A$ be terms, with $FV(t_i) \subseteq \underbrace{\{x_1 : A_1, \ldots, x_n : A_n}_{\text{"Context" } \Gamma}$

Define $\Gamma \vdash t_1 = t_2 : A$ as the eq^{n}l theory generated as follows:

(i) $\Gamma \vdash t_1 = t_2 : A$ and $\Gamma \subseteq \Delta$ then $\Delta \vdash t_1 = t_2 : A$

.) Equational Theory: The relation $R(t_1, t_2) \Leftrightarrow \Gamma \vdash t_1 = t_2 : A$ is reflexive, symm., and transitive

i) Substition / Congruence Rules

$$\frac{\Gamma \vdash t_1 = t_2 : A}{\Gamma \vdash f`t_1 = f`t_2 : B}$$ where $f : B^A$ s.t. $FV(f) \subseteq \Gamma$

$$\frac{\Gamma, x : A \vdash \varphi_1(x) = \varphi_2(x) : B}{\Gamma \vdash \lambda x{:}A\, \varphi_1(x) = \lambda x{:}A\, \varphi_2(x) : B^A}$$

v) Products

$\Gamma \vdash t = * : 1$ for all terms $t : 1$

$\Gamma \vdash \pi_i`\langle a_1, a_2\rangle = a_i : A_i \quad i = 1, 2$

$\Gamma \vdash \langle \pi_1`c, \pi_2`c\rangle = c : A_1 \times A_2$

(v) Lambda - Calculus Eq^{n}s

(β) $\Gamma \vdash (\lambda x{:}_A\, \varphi(x))`a = \varphi[a/x]$

(for all $a : A$, substitutable for x)

(η) $\Gamma \vdash \lambda x{:}_A (f`x) = f : B^A$

for all terms $f : B^A$ with $FV(f) \subseteq \Gamma$, such that $x \notin FV(f)$.

Applied λ-theories: May have additional types, terms, eq^{n}s.

It is convenient to consider contexts as ordered: $\vec{\Gamma} = (x_1 : A_1, \dots, x_n : A_n)$

Summary of Typed Lambda Calculus

Types

$A, B ::= \text{atom} \mid 1 \mid A \times B \mid B^A$

Terms

$s, t ::= \text{vbl} \mid * \mid \pi_1 \mid \pi_2 \mid \langle s, t \rangle \mid \lambda x. \varphi \mid$
$ev \langle s, t \rangle$ or abbreviated $s`t$

Equations

- Equality / Congruence eq^{n}s
- Products & Terminal Object
- (β) $(\lambda x \varphi)`a = \varphi[a/x]$

  (η) $(\lambda x. f`x) = f$ if $x \notin FV(f)$

Let $\mathcal{L}$ = simply typed λ-calc.
$\mathcal{C}$ = a ccc

An interpretation function
$[\![-]\!] : \mathcal{L} \rightsquigarrow \mathcal{C}$

- $[\![-]\!]$: Types $\longrightarrow ob(\mathcal{C})$

  $[\![\text{atom}]\!] \in ob(\mathcal{C})$ [arb.]

  $[\![A \times B]\!] = [\![A]\!] \times [\![B]\!]$

  $[\![B^A]\!] = [\![B]\!]^{[\![A]\!]}$

- $[\![-]\!]$: Terms $\longrightarrow$ arrows of $\mathcal{C}$ inductively : let $\{x_1^{A_1}, \ldots, x_n^{A_n}\} = \Gamma$ be a context (considered ordered). Write $\Gamma \rhd t : B$ for

"t is a term of type B with free vbls$(t) \subseteq \Gamma$"

Terms are defined inductively.
So define the meaning of t

$$= \llbracket \Gamma \rhd t : B \rrbracket : \prod_{i=1}^{n} \llbracket A_i \rrbracket \longrightarrow \llbracket B \rrbracket$$

by induction on t. Write $\llbracket t \rrbracket$ for short:

- $t = x_i^{A_i}$ $\quad \llbracket t \rrbracket = \prod_{j=1}^{n} \llbracket A_j \rrbracket \xrightarrow{\pi_i} \llbracket A_i \rrbracket \in \mathcal{C}$

- $t = c : B$ (a constant)

  $\llbracket t \rrbracket : \mathbb{1} \longrightarrow \llbracket B \rrbracket \in \mathcal{C}$

  (an arrow with same 'constant' value in $\mathcal{C}$) - see discussion in class.

- $t = \langle t_1, t_2 \rangle : B_1 \times B_2$. By I.H.

  $\therefore \llbracket t_i \rrbracket : \prod_{j=1}^{n} \llbracket A_j \rrbracket \longrightarrow \llbracket B_i \rrbracket$

  then $\llbracket t \rrbracket = \prod_{j} \llbracket A_j \rrbracket \xrightarrow{\langle \llbracket t_1 \rrbracket, \llbracket t_2 \rrbracket \rangle} \llbracket B_1 \times B_2 \rrbracket$

  $\therefore \llbracket \langle t_1, t_2 \rangle \rrbracket = \langle \llbracket t_1 \rrbracket, \llbracket t_2 \rrbracket \rangle$

- $t = u`a =_{def} ev\langle u, a \rangle$, where $u : B^A$, $a : A$

  By I.H., say

  $$\Gamma \rhd u : B^A \quad , \quad \Gamma \rhd a : A$$

  and suppose

  $$\prod_{i=1}^{n} \llbracket A_i \rrbracket \xrightarrow{\llbracket u \rrbracket} \llbracket B \rrbracket^{\llbracket A \rrbracket}$$

  $$\prod_{i=1}^{n} \llbracket A_i \rrbracket \xrightarrow{\llbracket a \rrbracket} \llbracket A \rrbracket$$

  ---

  then we define $\llbracket ev\langle u, a \rangle \rrbracket$

  $= \llbracket u`a \rrbracket = ev \langle \llbracket u \rrbracket, \llbracket a \rrbracket \rangle$, i.e.

$$\begin{tikzcd}
\prod_{i=1}^{n} \llbracket A_i \rrbracket \arrow[r, "\langle \llbracket u \rrbracket, \llbracket a \rrbracket \rangle"] & \llbracket B \rrbracket^{\llbracket A \rrbracket} \times \llbracket A \rrbracket \arrow[d, "ev"] \\
& \llbracket B \rrbracket
\end{tikzcd}$$

- $t = \lambda x{:}A\, \varphi(x) : B^A$

By I.H., suppose $\vec{\Gamma}, x{:}A \rhd \varphi(x) : B$ with meaning:

$$\prod_{i=1}^{n} [\![A_i]\!] \times [\![A]\!] \xrightarrow{[\![\varphi]\!]} [\![B]\!] \in \mathcal{C}$$

Define $[\![t]\!] = [\![\varphi]\!]^* =$ curry of $[\![\varphi]\!]$

so, $$\prod_{i=1}^{n} [\![A_i]\!] \xrightarrow{[\![\varphi]\!]^*} [\![B^A]\!] \in \mathcal{C}$$

<u>Thm</u> (Soundness) In typed λ-calculus, if $\Gamma \vdash t_1 = t_2 : B$ then $[\![\Gamma \rhd t_1]\!] = [\![\Gamma \rhd t_2]\!]$, as arrows $\prod_{i=1}^{n} [\![A_i]\!] \longrightarrow [\![B]\!] \in \mathcal{C}$.

<u>Pf</u>: By induction on equations (2 rules) of λ-calculus. We need:

<u>Properties of Substitution</u>

- The following Substitution rule holds

$$\frac{\Gamma \rhd a : A \qquad \Gamma, x{:}A \rhd \varphi : B}{\Gamma \rhd \varphi[a/x] : B}$$

- $[\![\varphi[a/x]]\!]$ is the composite

$$\begin{tikzcd} \prod_i [\![A_i]\!] \times [\![A]\!] \arrow[r, "{[\![\varphi]\!]}"] & {[\![B]\!] \in \mathcal{C}} \\ \prod_i [\![A_i]\!] \arrow[u, "{\langle id, [\![a]\!] \rangle}"] & \end{tikzcd}$$

"Substitution = composition"

<u>Pf</u>: By structural induction

We also need:

<u>Fact</u> : In any CCC,

$$\langle a,b\rangle \circ c = \langle a\circ c, b\circ c\rangle$$

<u>Proof</u>: Let $h =_{\text{def}} \langle a,b\rangle \circ c$.

Then, in any CCC,

$$\begin{aligned} h &= \langle \pi_1 \circ h, \pi_2 \circ h\rangle \\ &= \langle \pi_1 \circ (\langle a,b\rangle \circ c), \pi_2 \circ (\langle a,b\rangle \circ c)\rangle \\ &= \langle (\pi_1 \circ \langle a,b\rangle)\circ c, (\pi_2 \circ \langle a,b\rangle)\circ c\rangle \\ &= \langle a\circ c, b\circ c\rangle \end{aligned}$$

<u>Dually for coproducts</u> :

$$u \circ [f,g] = [u\circ f, u \circ g]$$

Proof of <u>Beta Rule</u> :

$$\begin{aligned} &[\![\lambda x{:}A\, \varphi \,`\, a]\!] \\ &= ev \langle [\![\lambda x{:}A\, \varphi]\!], [\![a]\!] \rangle \\ &= ev \langle [\![\varphi]\!]^*, [\![a]\!] \rangle \\ &= \underbrace{ev \langle [\![\varphi]\!]^* \circ \pi_1, \pi_2 \rangle}_{} \circ \langle id, [\![a]\!] \rangle \\ &= [\![\varphi]\!] \circ \langle id, [\![a]\!] \rangle \quad \text{by } (\beta \\ &= [\![\varphi [a/x]]\!] \quad \text{by substit}^{\text{n}} \text{ Lemma.} \end{aligned}$$

In fact, there is a tight connection:

$$\boxed{\text{Cartesian closed categories}} \rightleftarrows \boxed{\text{Typed } \lambda\text{-theories}}$$

- Every CCC has an associated "typed λ-calculus" (internal language)
- Every typed λ-calculus $\mathcal{L}$ syntactically generates a CCC (term model construction) $C(\mathcal{L})$

Pf sketch:

- objects of $C(\mathcal{L})$ = types of $\mathcal{L}$
- arrows of $C(\mathcal{L})$ = equivalence classes (modulo provable equality) of terms.

Formally, an arrow, denoted $A \xrightarrow{t(x)} B$, is really

a pair $(x:A,\ t(x):B)$ where $t(x)$ is a term with free vbls $\subseteq \{x\}$, modulo equality

i.e., $t_1(x) = t_2(x)$ means
$x:A \vdash_{\mathcal{L}} t_1(x) = t_2(x)$

- Composition: $A \xrightarrow{t(x)} B \xrightarrow{s(y)} C$, composite $s[t(x)/y]$

$$\begin{tikzcd} A \arrow[r, "t(x)"] \arrow[rr, bend right, "{s[t(x)/y]}"'] & B \arrow[r, "s(y)"] & C \end{tikzcd}$$

- identity $A \xrightarrow{x} A$
- pairing $$\frac{C \xrightarrow{t(x)} A \qquad C \xrightarrow{r(x)} B}{C \xrightarrow{\langle t(x), r(x)\rangle} A \times B}$$
- Proj^{n}: $A \times B \xrightarrow{\pi_1 z} A$, $A \times B \xrightarrow{\pi_2 z} B$

$$\frac{C \times A \xrightarrow{t(z)} B}{C \xrightarrow{t^*(y)} B^A}$$

where $t^*(y) = \lambda x{:}A\; t(\langle y, x\rangle)$

<u>Exercise</u>: Check the equations,

e.g. (β) $Ev \circ \langle \varphi^* \circ \pi_1, \pi_2 \rangle = \varphi : B$

etc.

<u>Conclusion</u>: $C(\mathcal{L})$ forms a CCC

Coproducts

In a CCC with coproducts, we can form $1 + 1 =_{def} 2$.

Think of arrows $1 \longrightarrow 2$ as "propositions" or "truth values".

<u>Define</u>: $\underline{True} : 1 \longrightarrow 2 = in_L$

$\underline{False} : 1 \longrightarrow 2 = in_R$

$\neg : 2 \longrightarrow 2$

by: $\neg = [\,False, True\,]$.

$\underline{p \wedge q} = [p, \underline{False}]\, q$

where $p, q : 1 \longrightarrow 2$.

<u>Exercise</u>: $\left.\begin{array}{l} \cdot\ \neg \cdot \neg \\ \cdot\ p \wedge q \end{array}\right\}$ on truth-values have correct truth-table ppties.

"<u>Double Duals</u>" <u>in CCC's</u>:

Let B be a fixed object of a CCC. There is a canonical arrow ("deduction")

$$A \longrightarrow B^{(B^A)}$$

as follows:

$$A \times B^A \underset{\tau}{\xrightarrow{\sim}} B^A \times A \xrightarrow{ev} B$$

where $\tau = \langle \pi_2, \pi_1 \rangle$ is the "twist" map.

$$\therefore \quad A \xrightarrow{(ev \circ \tau)^*} B^{(B^A)}$$

(like in $\underline{Vec}_K$, $V \longrightarrow V^{\perp\perp}$)

In boolean logic, we define

$$\neg A = A \Rightarrow \bot$$

where $\bot$ = false.

Similarly, we can define this in any Biccc, letting $\bot$ = the initial object.

<u>Question</u>: for which biccc's do we have $\neg\neg A \cong A$?

<u>Thm</u> In a biccc with zero object $\bot$, there is <u>at most</u> one arrow $A \longrightarrow \bot$, i.e. there's at most one proof $T \vdash \neg A$, where T = terminal object.