Overview: Categories, Proofs and Games

1. Introduction to Category Theory.
2. Curry-Howard isomorphism and Linear logic.
3. Introduction to Game Semantics.

Inter-twining of themes

Course web page:

`http://web.comlab.ox.ac.uk/oucl/work/samson.abramsky/gsem/index.html`

Useful text for background reading:

Categories, Types and Structures by Andrea Asperti and Giuseppe Longo.

Downloadable from:

`http://www.di.ens.fr/users/longo/download.html`

Relevant material:

Chapter	Sections
1	1.1–1.4
2	2.1–2.3
3	3.1–3.3, 3.5
4	4.3–4.4
8	8.1–8.7

Other Useful Reading on Category Theory

- **Category Theory for Computing Science, 3rd Edition** by M. Barr and C. Wells, Les Publications de CRM, Montreal. Contains many exercises with solutions.
- **Lecture Notes on Basic Category Theory** by Jaap van Oosten. Downloadable from: `http://www.math.uu.nl/people/jvoosten/onderwijs.html`

Preliminaries on Mathematical Structures

Monoids

A *monoid* is a structure $(M, \cdot, 1)$ where M is a set,

$$- \cdot - : M \times M \longrightarrow M$$

is a binary operation, and $1 \in M$, satisfying the following axioms:

$$(x \cdot y) \cdot z = x \cdot (y \cdot z) \qquad 1 \cdot x = x = x \cdot 1$$

Examples:

- Groups
- $(\mathbb{N}, +, 0)$
- Strings: Σ^*, $s \cdot t = st$, $1 = \varepsilon$.

Partial Orders

A *partial order* is a structure $(P, \leq)$ where P is a set, and $\leq$ is a binary relation on P satisfying:

- $x \leq x$ (Reflexivity)
- $x \leq y \ \wedge y \leq x \quad \Rightarrow \quad x = y$ (Antisymmetry)
- $x \leq y \ \wedge \ y \leq z \quad \Rightarrow \quad x \leq z$ (Transitivity).

Examples:

- $(\mathbb{R}, \leq)$
- $(\mathcal{P}(X), \subseteq)$
- Strings, the sub-string relation.

Sets and Maps: learning to think with arrows

Notation for maps (functions) between sets:

$$f : X \longrightarrow Y \qquad \text{Diagrammatic notation: } X \xrightarrow{f} Y$$

X is the *domain* of f. Y is the *codomain*.

Notation for composition:

$$g \circ f : X \longrightarrow Z \quad \text{or} \quad f ; g : X \longrightarrow Z \quad \textit{or} \quad X \xrightarrow{f} Y \xrightarrow{g} Z$$

Identity map:

$$1_X : X \longrightarrow X$$

Axioms relating these operations:

$$(f ; g) ; h = f ; (g ; h) \qquad 1_X ; f = f = f ; 1_Y$$

$f : X \longrightarrow Y$ is *injective* if

$$f(x) = f(y) \quad \Rightarrow \quad x = y.$$

$f : X \longrightarrow Y$ is *surjective* if

$$\forall y \in Y.\, \exists x \in X.\, f(x) = y.$$

$f : X \longrightarrow Y$ is *monic* if

$$f \circ g = f \circ h \quad \Rightarrow \quad g = h.$$

$f : X \longrightarrow Y$ is *epic* if

$$g \circ f = h \circ f \quad \Rightarrow \quad g = h.$$

Proposition

1. f injective iff f monic.
2. f surjective iff f epic.

Monoid Homomorphisms

If M_1, M_2 are monoids, a map $h : M_1 \longrightarrow M_2$ is a *monoid homomorphism* iff:

$$h(x \cdot y) = h(x) \cdot h(y) \qquad h(1) = 1.$$

Partial order Homomorphisms

If P, Q are partial orders,, a map $h : P \longrightarrow Q$ is a *partial order homomorphism* (or *monotone function*) if:

$$x \leq y \;\Rightarrow\; h(x) \leq h(y).$$

Note that homomorphisms are *closed under composition*, and that *identity maps* are homomorphisms.

Categories: basic definitions

Category $\mathcal{C}$:

Objects A, B, C, ...

Morphisms/arrows: for each pair of objects A, B, a set of morphisms $\mathcal{C}(A, B)$, with *domain* A and *codomain* B

Notation: $f : A \longrightarrow B$ for $f \in \mathcal{C}(A, B)$.

Composition of morphisms: for any triple of objects A, B, C a map

$$c_{A,B,C} : \mathcal{C}(A, B) \times \mathcal{C}(B, C) \longrightarrow \mathcal{C}(A, C)$$

Notation: $c_{A,B,C}(f, g) \;=\; f ; g \;=\; g \circ f$.

Diagrammatically: $A \xrightarrow{f} B \xrightarrow{g} C$.

Identities: for each object A, a morphism id_A.

Axioms

$$(f ; g); h = f ; (g; h) \qquad f ; \mathsf{id}_B = f = \mathsf{id}_A ; f$$

Examples

- Monoids are one-object categories
- A category in which for each pair of objects A, B there is at most one morphism from A to B is the same thing as a *preorder*, i.e. a reflexive and transitive relation.
- Any kind of mathematical structure, together with structure preserving functions, forms a category. E.g. **Set** (sets and functions), **Grp** (groups and group homomorphisms), **Mon** (monoids and monoid homomorphisms), $\mathbf{Vect}_k$ (vector spaces over a field k, and linear maps), **Top** (topological spaces and continuous functions), **Pos** (partially ordered sets and monotone functions), etc. etc.

Duality

The *opposite* of a category $\mathcal{C}$, written $\mathcal{C}^{\mathsf{op}}$, has the same objects as $\mathcal{C}$, and

$$\mathcal{C}^{\mathsf{op}}(A, B) = \mathcal{C}(B, A).$$

If we have

$$A \xrightarrow{f} B \xrightarrow{g} C$$

in $\mathcal{C}^{\mathsf{op}}$, this means

$$A \xleftarrow{f} B \xleftarrow{g} C$$

in $\mathcal{C}$, so composition $g \circ f$ in $\mathcal{C}^{\mathsf{op}}$ is defined as $f \circ g$ in $\mathcal{C}$!

This leads to a principle of duality: dualize a statement about $\mathcal{C}$ by making the same statement about $\mathcal{C}^{\mathsf{op}}$.

Example of Duality

A morphism f is *monic* in $\mathcal{C}^{\mathsf{op}}$ iff it is *epic* in $\mathcal{C}$; so monic and epic are dual notions.

$f : A \longrightarrow B$ in $\mathcal{C}$ iff $f : B \longrightarrow A$ in $\mathcal{C}^{\mathsf{op}}$. Thus f is monic in $\mathcal{C}^{\mathsf{op}}$ iff for all $g, h : C \longrightarrow B$ in $\mathcal{C}^{\mathsf{op}}$,

$$f \circ g = f \circ h \quad \Rightarrow \quad g = h,$$

iff for all $g, h : B \longrightarrow C$ in $\mathcal{C}$,

$$g \circ f = h \circ f \quad \Rightarrow \quad g = h,$$

iff f is epic in $\mathcal{C}$.

Many important mathematical notions can be expressed at the general level of categories.

Isomorphism

An isomorphism in a category $\mathcal{C}$ is an arrow

$$i : A \longrightarrow B$$

such that there exists an arrow $j : B \longrightarrow A$ satisfying

$$j \circ i = \mathsf{id}_A \qquad\qquad i \circ j = \mathsf{id}_B$$

Notation: $i : A \xrightarrow{\cong} B$

In **Set** this gives bijection, in **Grp**, group isomorphism, in **Top**, homeomorphism, in **Pos**, order isomorphism, etc. etc.

Initial and terminal objects

- An object I in a category $\mathcal{C}$ is *initial* if for every object A, there exists a unique arrow $\iota_A : I \longrightarrow A$.
- A terminal object in $\mathcal{C}$ is the dual notion (i.e. an initial object in $\mathcal{C}^{\mathsf{op}}$).

There is a unique isomorphism between any pair of initial objects; thus initial objects are 'unique up to (unique) isomorphism', and we can (and do) speak of *the* initial object (if any such exists).

Products

Let A, B be objects in a category $\mathcal{C}$. An A, B–pairing is a triple (P, p_1, p_2) where P is an object, $p_1 : P \longrightarrow A$, $p_2 : P \longrightarrow B$. A morphism of A, B–pairings

$$f : (P, p_1, p_2) \longrightarrow (Q, q_1, q_2)$$

is a morphism $f : P \longrightarrow Q$ in $\mathcal{C}$ such that

$$q_1 \circ f = p_1, \qquad\qquad q_2 \circ f = p_2.$$

The A, B–pairings form a category $\mathbf{Pair}(A, B)$.

$(A \times B, \pi_1, \pi_2)$ is a *product* of A and B if it is *terminal* in $\mathbf{Pair}(A, B)$.

Thus products are unique up to isomorphism (if they exist).

Unpacking the definition of product,

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

is a product if for every A, B-pairing

$$A \xleftarrow{f} C \xrightarrow{g} B$$

there exists a unique morphism

$$\langle f, g \rangle : C \longrightarrow A \times B$$

such that the following diagram commutes:

$$\begin{tikzcd}
& C \arrow[dl, "f"'] \arrow[d, dashed, "\langle f, g \rangle"] \arrow[dr, "g"] & \\
A & A \times B \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & B
\end{tikzcd}$$

Examples

- **Set**?
- $\mathbf{Vect}_k$?
- In a poset?

General Products

A product for a family of objects $\{A_i\}_{i \in I}$ in a category $\mathcal{C}$ is an object P and morphisms

$$p_i : P \longrightarrow A_i \quad (i \in I)$$

such that, for all objects B and arrows

$$f_i : B \longrightarrow A_i \quad (i \in I)$$

there is a *unique* arrow

$$g : B \longrightarrow P$$

such that, for all $i \in I$,

$$p_i \circ g = f_i.$$

As before, if such a product exists, it is unique up to (unique) isomorphism.

General Products continued

Notation We write $P = \prod_{i \in I} A_i$ for the product object, and $g = \langle f_i \mid i \in I \rangle$ for the unique morphism in the definition.

What is the product of the empty family?

Fact If a category has binary and nullary products, then it has all finite products.

Pullbacks

Given a pair of morphisms

$$A \xrightarrow{f} C \xleftarrow{g} B$$

with common codomain, we define an (f, g)-**pairing** (or (f, g)-**cone**) to be

$$A \xleftarrow{p} D \xrightarrow{q} B$$

such that the following diagram commutes:

$$\begin{tikzcd} D \arrow[r, "q"] \arrow[d, "p"] & B \arrow[d, "g"] \\ A \arrow[r, "f"] & C \end{tikzcd}$$

A morphism of (f, g)-cones $h : (D_1, p_1, q_1) \longrightarrow (D_2, p_2, q_2)$ is a morphism $h : D_1 \longrightarrow D_2$ such that

$$\begin{tikzcd} & D_1 \arrow[dl, "p_1"'] \arrow[d, "h"] \arrow[dr, "q_1"] & \\ A & D_2 \arrow[l, "p_2"] \arrow[r, "q_2"'] & B \end{tikzcd}$$

We can thus form a category $\mathbf{Cone}(f, g)$. The *pull-back of f along g (or "fibred product of A and B over C", written $A \times_C B$)* is the terminal object of $\mathbf{Cone}(f, g)$ (if it has one).

Examples

- In **Set** the pullback is defined as a *subset of the cartesian product*:

$$A \times_C B \;=\; \{(a,b) \in A \times B \mid f(a) = g(b)\}.$$

  Examples: the unit circle, composable morphisms ...

- In **Set** again, subsets (*i.e.* inclusion maps) pull back to subsets:

$$\begin{tikzcd} f^{-1}(U) \arrow[r] \arrow[d, hook] & U \arrow[d, hook] \\ X \arrow[r, "f"'] & Y \end{tikzcd}$$

Limits and Colimits

The notions we have introduced so far are all special cases of a general notion of *limits* in categories, and the dual notion of *colimits*:

Limits	Colimits
Monics	Epics
Terminal Objects	Initial Objects
Products	Coproducts
Pullbacks	Pushouts

An important aspect of studying any kind of mathematical structure is to see what limits and colimits the category of such structures has.

For lack of time, we will not develop these notions in full generality.

Functors

Part of the ‘categorical philosophy’ is

Don’t just look at the objects; take the morphisms into account too

We can apply this to categories too! A ‘morphism of categories’ is a *functor*.

A functor $F : \mathcal{C} \longrightarrow \mathcal{D}$ is an assignment of:

- An object FA in $\mathcal{D}$ to every object A in $\mathcal{C}$.
- A map $F_{A,B} : \mathcal{C}(A, B) \longrightarrow \mathcal{D}(FA, FB)$ for every pair of objects A, B of $\mathcal{C}$.
  (In practice, we write $Ff : FA \longrightarrow FB$).

These maps must preserve composition and identities:

$$F(g \circ f) = Fg \circ Ff \qquad\qquad F\mathsf{id}_A = \mathsf{id}_{FA}.$$

Variance

A *contravariant* functor $F : \mathcal{C} \longrightarrow \mathcal{D}$ is a functor $F : \mathcal{C}^{\mathsf{op}} \longrightarrow \mathcal{D}$. (We sometimes refer to an ordinary functor as *covariant* for emphasis).

Products

The product $\mathcal{C} \times \mathcal{D}$ of categories $\mathcal{C}$, $\mathcal{D}$ is defined in the obvious way (an object is a pair of objects ...)

Mixed Variance

Functors 'of several variables' are simply functors whose domain is a product category. Such functors can be covariant in some variables and contravariant in others, e.g.

$$F : \mathcal{C}^{\mathsf{op}} \times \mathcal{D} \longrightarrow \mathcal{E}.$$

Examples of Functors

- A functor between monoids is just a monoid homomorphism.
- A functor between preorders is just a monotone map.
- $U : \mathbf{Mon} \longrightarrow \mathbf{Set}$ is the 'forgetful' or 'underlying' functor which sends a monoid to its set of elements, 'forgetting' the algebraic structure, and sends a homomorphism to the corresponding function between sets. There are similar forgetful functors for other categories of structured sets. Why are these trivial-looking functors useful— we'll see!

Set-valued Functors

Many important constructions arise as functors $F : \mathcal{C} \longrightarrow \mathbf{Set}$.

Examples

- If G is a group, a functor $F : G \longrightarrow \mathbf{Set}$ is an *action of* G *on a set.*
- If P is a poset representing time, a functor $F : P \longrightarrow \mathbf{Set}$ is a notion of *sets varying through time*. This is related to *Kripke semantics*, and to *forcing arguments* in set theory.
- Let $\mathcal{C}$ be the (finite) category

$$\bullet \rightrightarrows \bullet$$

  Functors $F : \mathcal{C} \longrightarrow \mathbf{Set}$ correspond to *directed graphs*.

Example: Lists

Data-type constructors are functors. As a basic example, we consider lists. There is a functor

$$\mathsf{List} : \mathbf{Set} \longrightarrow \mathbf{Set}$$

which takes a set X to the set of all finite lists (sequences) of elements of X. List is functorial: its action on morphisms (*i.e.* functions, *i.e.* (functional) programs) is given by *maplist*:

$$\frac{f : X \longrightarrow Y}{\mathsf{List}(f) : \mathsf{List}(X) \longrightarrow \mathsf{List}(Y)}$$

$$\mathsf{List}(f)[x_1, \ldots, x_n] = [f(x_1), \ldots, f(x_n)]$$

Products as functors

If a category C has binary products, then there is automatically a functor

$$- \times - : \mathcal{C} \times \mathcal{C} \longrightarrow \mathcal{C}$$

If $(f, g) : (A, B) \longrightarrow (C, D)$ in $\mathcal{C} \times \mathcal{C}$ — which just means

$$f : A \longrightarrow C \quad \text{and} \quad g : B \longrightarrow D$$

in $\mathcal{C}$, then we define

$$f \times g : A \times B \longrightarrow C \times D$$

by

$$f \times g = \langle f \circ \pi_1, g \circ \pi_2 \rangle$$

One can use the equational properties of pairing and projections to show *functoriality* (*i.e.* that composition and identities are preserved).

Hom-functors

- For each object A of $\mathcal{C}$, there is a functor

$$\mathcal{C}(A,-) : \mathcal{C} \longrightarrow \mathbf{Set}$$

(the covariant Hom-functor at A), where

$$\mathcal{C}(A,-)(B) = \mathcal{C}(A,B), \qquad \mathcal{C}(A,-)(f : B \to C) : g \mapsto f \circ g.$$

- There is also a contravariant Hom-functor

$$\mathcal{C}(-,A) : \mathcal{C}^{\mathsf{op}} \longrightarrow \mathbf{Set}$$

$$\mathcal{C}(-,A)(B) = \mathcal{C}(B,A), \qquad \mathcal{C}(-,A)(h : C \to B) : g \mapsto g \circ h.$$

- Generalizing both of these, there is a bivariant Hom-functor

$$\mathcal{C}(-,-) : \mathcal{C}^{\mathsf{op}} \times \mathcal{C} \longrightarrow \mathbf{Set}.$$

Natural transformations

> ‘Categories were only introduced to allow functors to be defined; functors were only introduced to allow natural transformations to be defined.’

Just as categories have morphisms between them, namely functors, so functors have morphisms between them too — *natural transformations*.

Let $F, G : \mathcal{C} \longrightarrow \mathcal{D}$ be functors. A natural transformation $t : F \Longrightarrow G$ is a family of $\mathcal{D}$-morphisms

$$t_A : FA \longrightarrow GA$$

indexed by objects A of $\mathcal{C}$, such that, for all $f : A \longrightarrow B$,

$$\begin{tikzcd}
FA \arrow[r, "Ff"] \arrow[d, "t_A"'] & FB \arrow[d, "t_B"] \\
GA \arrow[r, "Gf"'] & GB
\end{tikzcd}$$

If each t_A is an isomorphism, we say that t is a *natural isomorphism.*

Examples

- If V is a finite dimensional vector space, then V is isomorphic to both its first dual V^* and to its second dual V^{**}. However, while it is naturally isomorphic to its second dual, there is no natural isomorphism to the first dual.

- Let Id be the identity functor on **Set**. Then there is a natural transformation

$$\Delta : \mathsf{Id} \Longrightarrow \mathsf{Id} \times \mathsf{Id}$$

$$\Delta_X : x \mapsto (x, x).$$

 (This is in fact the *only* natural transformation between these functors).

Natural transformations on lists

$$\mathsf{reverse}_X : \mathsf{List}(X) \longrightarrow \mathsf{List}(X) \qquad [x_1, \ldots, x_n] \mapsto [x_n, \ldots, x_1]$$

$$\mathsf{unit}_X : \mathsf{List}(X) \longrightarrow \mathsf{List}(X) \qquad x \mapsto [x]$$

$$\mathsf{flatten}_X : \mathsf{List}(\mathsf{List}(X)) \longrightarrow \mathsf{List}(X)$$

$$[[x_1^1, \ldots, x_{n_1}^1], \ldots, [x_1^k, \ldots, x_{n_k}^k]] \mapsto [x_1^1, \ldots\ldots, x_{n_k}^k]$$

Natural isomorphisms for products

If a category $\mathcal{C}$ has binary products and a terminal object, then there are natural isomorphisms

$$a_{A,B,C} : A \times (B \times C) \xrightarrow{\cong} (A \times B) \times C$$

$$l_A : \mathbf{1} \times A \xrightarrow{\cong} A \qquad\qquad r_A : A \times \mathbf{1} \xrightarrow{\cong} A$$

Since natural isomorphisms are a *self-dual* notion, the same holds if a category has binary coproducts and an initial object.

Universal Constructions

The categorical triad: *Functoriality, Naturality, Universality.*

Canonical solutions to problems.

In posets: *extremal* solutions. Thus sup and inf are extremal solutions to the problems of giving an *upper bound* or *lower bound* respectively of a set of reals.

Products in posets A product of A, B is an element P such that

$$P \leq A \quad \text{and} \quad P \leq B$$

and for any other other solution Q such that $Q \leq A$ and $Q \leq B$, we have $Q \leq P$. (Greatest lower bound).

(Co)Universal Arrows

Let $G : \mathcal{D} \longrightarrow \mathcal{C}$ be a functor, and C an object of $\mathcal{C}$. A *couniversal arrow from G to C* is an object D of $\mathcal{D}$ and a morphism

$$f : G(D) \longrightarrow C$$

such that, for any object D' of $\mathcal{D}$ and morphism $g : G(D') \longrightarrow C$ there exists a unique morphism $h : D' \longrightarrow D$ in $\mathcal{D}$ such that:

$$\begin{tikzcd}
D & & GD \arrow[r, "f"] & C \\
D' \arrow[u, dashed, "h"] & & GD' \arrow[u, dashed, "Gh"] \arrow[ur, "g"'] &
\end{tikzcd}$$

Examples

1. Terminal objects Let $\mathbf{1}$ be the one-object one-morphism category. A terminal object in a category $\mathcal{C}$ is exactly a couniversal arrow from the unique functor $\mathcal{C} \longrightarrow \mathbf{1}$ to the unique object in $\mathbf{1}$.

2. Products Let A, B be objects of $\mathcal{C}$. A product of A and B is exactly a couniversal arrow from the *diagonal functor*

$$\Delta : \mathcal{C} \longrightarrow \mathcal{C} \times \mathcal{C}$$

to (A, B).

Note that $\mathcal{C} \times \mathcal{C} = \mathcal{C}^{\mathbf{2}}$, where $\mathcal{C}^{\mathbf{2}}$ is the *functor category*; $\mathbf{2}$ is the discrete category (only identity morphisms) with two objects.

This suggests an important generalization.

Generalization: Limits

Let $\mathcal{I}$ be an 'index category'. A *diagram of shape* $\mathcal{I}$ in a category $\mathcal{C}$ is just a functor $F : \mathcal{I} \longrightarrow \mathcal{C}$. We can form the functor category $\mathcal{C}^{\mathcal{I}}$ with objects the functors from $\mathcal{I}$ to $\mathcal{C}$, and natural transformations as morphisms.

There is a diagonal functor

$$\Delta : \mathcal{C} \longrightarrow \mathcal{C}^{\mathcal{I}}.$$

A *limit for the diagram* F ia a couniversal arrow from Δ to F.

This concept of limits subsumes products (including infinite products), pullbacks, inverse limits, etc. etc.

For example, we get pullbacks by taking

$$\mathcal{I} \quad = \quad \bullet \longrightarrow \bullet \longleftarrow \bullet$$

Exponentials

In **Set**, given sets A, B, we can form the set of functions $B^A = \mathbf{Set}(A, B)$, *which is again a set.* This closure of **Set** under forming 'function spaces' is one of its most important properties.

How can we axiomatize this situation? Once again, rather than asking what the elements of a function space are, we ask rather what can we *do* with it operationally?

Answer: apply functions to their arguments. That is, there is a map

$$\mathsf{ev}_{A,B} : B^A \times A \longrightarrow B \qquad \mathsf{ev}_{A,B}(f, a) = f(a)$$

Think of the function as a 'black box': we can feed it inputs and observe the outputs.

Couniversal property of evaluation

For any $g : C \times A \longrightarrow B$, there is a unique map $\Lambda(g) : C \longrightarrow B^A$ such that:

$$
\begin{tikzcd}
B^A & & B^A \times A \arrow[r, "\mathsf{ev}_{A,B}"] & B \\
C \arrow[u, dashed, "\Lambda(g)"] & & C \times A \arrow[u, dashed, "\Lambda(g) \times \mathsf{id}_A"] \arrow[ur, "g"'] &
\end{tikzcd}
$$

In **Set**, this is defined by

$$\Lambda(g)(c) = k : A \longrightarrow B \ \text{ where } \ k(a) = g(c, a).$$

This process of transforming a function of two arguments into a function-valued function of one argument is known as *Currying* after H. B. Curry. It is an algebraic form of λ-*abstraction*.

General definition of exponentials

Let $\mathcal{C}$ be a category with a terminal object and binary products. For each object A of $\mathcal{C}$, we can define a functor

$$- \times A : \mathcal{C} \longrightarrow \mathcal{C}$$

We say that $\mathcal{C}$ *has exponentials* if for all objects A and B of $\mathcal{C}$ there is a couniversal arrow from $- \times A$ to B, *i.e.* an object B^A of $\mathcal{C}$ and a morphism

$$\mathsf{ev}_{A,B} : B^A \times A \longrightarrow B$$

with the couniversal property: for every $g : C \times A \longrightarrow B$, there is a unique morphism $\Lambda(g) : C \longrightarrow B^A$ such that

$$\mathsf{ev}_{A,B} \circ (\Lambda(g) \times \mathsf{id}_A) = g.$$

(Same as diagram on previous slide).

Cartesian Closed Categories

A category with a terminal object, products and exponentials is called a *Cartesian Closed Category* (CCC).

This notion is fundamental in understanding functional types, models of λ-calculus, and the structure of proofs.

Notation The notation of B^A for exponential objects, and $\mathsf{ev}_{A,B}$ for evaluation, is standard in the category theory literature. However, for our purposes, the following notation will be more convenient: $A \Rightarrow B$ for exponential objects, and

$$\mathsf{Ap}_{A,B} : (A \Rightarrow B) \times A \longrightarrow B$$

for *application* (*i.e.* evaluation).

Example: Boolean Algebras

A Boolean algebra (*e.g.* a powerset $\mathcal{P}(X)$) is a CCC.

Products are given by conjunctions $A \wedge B$. We define exponentials as *implications*:

$$A \Rightarrow B \quad = \quad \neg A \vee B$$

Evaluation is just Modus Ponens:

$$(A \Rightarrow B) \wedge A \leq B$$

Couniversality is the 'Deduction Theorem':

$$C \wedge A \leq B \iff C \leq A \Rightarrow B.$$