Overview

- We aim to study some of the connections between type theory and category theory.

- We shall assume knowledge of basic logic, and an appreciation of basic type theory from programming.

- We shall cover the category theory and type theory required for understanding the connections …

- … with certain topics being taught in more detail by José, so we include only notational summaries.
High Level Topics

- **Order Theory:** We shall study properties of orders, and show how these give rise to basic examples of categories and categorical structures.

- **Category Theory:** We shall study some simple category theory, enough to model simple type theories.
High Level Topics

- **Type Theory:** We will define a simple type system, derive a categorical semantics from first principles, and show how a theory in the type system “corresponds” to a special category.

- **Applications:** We apply the correspondence to obtain a result about type theory.

- If there is time we will also look at an application to logic.
Order Theory

- A (mathematical) order makes precise our intuitions about relations such as less than or less than or equal to.

- What’s the point? Order relations are used extensively in computing; and a particular kind of order gives rise to simple examples of categories.

- We review some formal definitions of order relations.

- We progress to basic mathematical structures that can be defined using order relations.
Order Theory

- **Why?** These structures are simple examples of the common structures found in categories!

- **We also define functions which preserve order structure.**

- **Why?** Such functions will give us examples of **functors**, which are mappings between categories, and are fundamental to category theory.
Basic Definitions

■ A binary relation $R$ on a set $X$ is any subset $R \subseteq X \times X$. If $x, y \in X$, then we will write $x R y$ for $(x, y) \in R$.

■ $R$ is reflexive if whenever $x \in X$ we have $x R x$;

■ transitive if whenever $x, y, z \in X$, 
$(x R y$ and $y R z)$ implies $x R z$;
■ **symmetric** if whenever \( x, y \in X \) then \( x R y \) implies \( y R x \);

■ **anti-symmetric** if whenever \( x, y \in X \),

\[ (x R y \text{ and } y R x) \text{ implies } x = y \]

■ \( R \) is an **equivalence relation** if it is reflexive, symmetric and transitive.

■ We will not make much use of the definitions on this slide, but they are (of course) used throughout computer science.
Preordered Sets

- **A preorder** on a set $X$ is a binary relation $\leq$ on $X$ which is reflexive and transitive.

- **A preordered set** $(X, \leq)$ is a set $X$, equipped with a preorder $\leq$ on the set $X$.

- **NOTE:** We often just refer to a “preorder $X$”.

- Every preorder is an example of a category! Elements $x$ in $X$ are objects and each relationship $x \leq x'$ is a morphism $x \rightarrow x'$.

- The axioms that make $X$ a preorder are exactly those required to make $X$ a category.
Examples of Preordered Sets

- The set $\mathcal{P}(X) = \{A \mid A \subseteq X\}$ is called the powerset of $X$. The powerset is a preorder with order given by inclusion of subsets, $A \subseteq A'$.

- Given preorders $X$ and $Y$, their cartesian product has underlying set

$$X \times Y \overset{\text{def}}{=} \{(x, y) \mid x \in X, y \in Y\}$$

with order given pointwise, that is $(x, y) \leq (x', y')$ iff $x \leq_X x'$ and $y \leq_Y y'$ (using the obvious notation).
Fix any set $X$. Consider the set $\text{Sub}(X)$ of all injective functions $i : A \to X$, and order $i : A \to X \leq j : A' \to X$ provided there is $f : A \to A'$ such that

$$
\begin{array}{c}
A' \\
| f \\
\downarrow
\end{array} 
\begin{array}{c}
A \\
i
\rightarrow
\end{array} 
\begin{array}{c}
X \\

\end{array}
$$

If $X$ is a preorder, then $X^{op}$ is a preorder given by changing the order to $\leq^{op}$ where for $x, x' \in X^{op} \overset{\text{def}}{=} X$, we define $x \leq^{op} x'$ if and only if $x' \leq x$. 
Monotone Functions between Preordered Sets

Let $f : X \rightarrow Y$ be a function, with $X$ and $Y$ equipped with preorders. $f$ is monotone if for all $x, x' \in X$ we have $x \leq x'$ implies $f(x) \leq f(x')$.

$f$ is also called a homomorphism of preorders, or sometimes simply a morphism.

If $X$ is regarded as a category then such a function is an example of a functor between categories.
Examples of Monotone Functions

■ The function $f : \mathbb{N} \rightarrow \mathbb{N}$ given by $f(n) \overset{\text{def}}{=} n + 3$ is monotone; there are many such functions on $\mathbb{N}$ (and of course $\mathbb{R}$).

■ The projection functions $\pi : X \times Y \rightarrow X$ and $\pi' : X \times Y \rightarrow Y$ are monotone if $X \times Y$ is cartesian product.

■ Given a set $U$, there is a function $S : \mathcal{P}(U) \rightarrow \mathcal{P}(\mathcal{P}(U))$ given by $S(X) \overset{\text{def}}{=} \mathcal{P}(X)$. This is monotone.

■ If $f : X \rightarrow Y$ is any set function, then $f^{-1} : \mathcal{P}(Y) \rightarrow \mathcal{P}(X)$ defined by $f^{-1}(B) \overset{\text{def}}{=} \{ x \in X \mid f(x) \in B \}$ is monotone.
Notions of Isomorphism

- The preorders $X$ and $Y$ are **isomorphic** if there are monotone functions $f : X \rightarrow Y$ and $g : Y \rightarrow X$ for which $g \circ f = id_X$ and $f \circ g = id_Y$.
- The monotone function $g$ is an **inverse** for $f$; and vice versa.
- The elements $x$ and $x'$ in $X$ are **isomorphic** if $x \leq x'$ and $x' \leq x$. We write $x \cong x'$.
- We can regard $\cong$ as a relation on $X$, which is in fact an equivalence relation.
Partially Preordered Sets

- A partial order on a set $X$ is a binary relation $\leq$ which is reflexive, transitive and anti-symmetric.

- A partially ordered set (poset) $(X, \leq)$ is a set $X$ equipped with a partial order $\leq$ on the set $X$.

- Trivially, any poset is of course a preorder in which isomorphic elements are always equal. Many of our examples of preorders will in fact be posets.
Properties in Ordered Sets

- Suppose that $X$ is a preorder and $A$ is a subset of $X$. An element $x \in X$ is an upper bound for $A$ if for every $a \in A$ we have $a \leq x$ (sometimes written $A \leq x$).

- An element $x \in X$ is a greatest element of $A$ if it is an upper bound of $A$ which belongs to $A$;

- Lower bounds and least elements are defined analogously.

- Greatest and least elements are unique up to isomorphism; so too for lower and upper bounds.
Binary Meets and Joins

- Meets and joins are very simple examples of universal constructions from category theory.

- Given a pair of elements \( x, x' \in X \) in a preorder \( X \), the set \( L \) of lower bounds for \( \{x, x'\} \) is

\[
L \overset{\text{def}}{=} \{ l \in X \mid l \leq x \text{ and } l \leq x' \}
\]

- A meet of a pair of elements \( x, x' \in X \), if it exists, is a greatest element in the set \( L \) of lower bounds for \( \{x, x'\} \).

- This is a simple example of a universal construction in category theory. A binary meet is an example of a binary product.
Writing $\rightarrow$ instead of $\leq$ we have
A join of a pair of elements $x, x' \in X$, if it exists, is a least element in the set $U$ of upper bounds for $\{x, x'\}$.

Writing $\rightarrow$ instead of $\leq$ we have
Top and Bottom Elements; Uniqueness Properties

- For a preorder $X$, a top element $\top \in X$ satisfies $x \leq \top$ for all $x \in X$.
- For a preorder $X$, a bottom element $\bot \in X$ satisfies $\bot \leq x$ for all $x \in X$.
- In a preorder, if a meet, join, top or bottom element exists, then it is unique up to isomorphism.
Some Examples

■ With the usual order on the natural numbers $\mathbb{N}$, binary meets and joins are given by simply taking the least and greatest elements (of the pair).

■ With the inclusion order on $\mathcal{P}(X)$, binary meets and joins are given by the operations of intersection and union. What are the top and bottom elements?

■ Define the order $d \mid n$ to mean that $(\exists k \in \mathbb{N})(n = k \times d)$. With this order, binary meets and joins are given simply by highest common factor and lowest common multiple respectively. Are there top and bottom elements?
Arbitrary Meets and Joins

- Let $X$ be a preordered set and $A \subseteq X$. A join of $A$ is a least element in the set of upper bounds for $A$. All joins are isomorphic.

- A meet of $A$ is a greatest element in the set of lower bounds for $A$. All meets are isomorphic.

- If $A$ has at least one join (it might not!) we write $\bigvee A$ for a choice of one of the joins of $A$. Write also $x \vee x'$ for $\bigvee \{x, x'\}$.

- $\bigwedge A$ is a choice of meet $A$. Write also $x \wedge x'$ for $\bigwedge \{x, x'\}$.

- If a preorder has all meets we say it is complete.

- If a preorder has all joins we say it is cocomplete.
Prelattices

- A prelattice is a preordered set which has binary meets and joins, plus top and bottom elements.
- In fact the examples \((\mathbb{N}, |)\) and \((\mathcal{P}(X), \subseteq)\) are prelattices.
- So too is \((\text{Sub}(X), \subseteq)\) but it requires a little more work to verify than the other three examples.
- \((\mathbb{N}, \leq)\) is not a prelattice.
Heyting Prelattices

A Heyting prelattice $X$ is a prelattice in which for each pair of elements $y,z \in X$ there is an element $y \Rightarrow z \in X$ such that

$$x \leq y \Rightarrow z \quad \text{iff} \quad x \land y \leq z.$$  

We call $y \Rightarrow z$ the Heyting implication of $y$ and $z$.

In a Heyting prelattice $X$, the Heyting implication of $y$ and $z$ is unique up to isomorphism.

Suppose that $a$ and $a'$ are two candidates for the element $y \Rightarrow z \in X$. Then $a \leq a$ implies $a \land y \leq z$ implies $a \leq a'$; the converse is similar.
Distributive Prelattices and Examples

Let $X$ be a prelattice. Then $X$ is distributive if it satisfies

$$x \land (y \lor z) \cong (x \land y) \lor (x \land z)$$

for all $x, y, z$ in $X$.

- $\mathcal{P}(X)$ is a Heyting prelattice where $A \Rightarrow A' \overset{\text{def}}{=} (X \setminus A) \cup A'$.

- Any finite distributive prelattice $X$ is a Heyting prelattice.

One may define

$$y \Rightarrow z \overset{\text{def}}{=} \bigvee \{ l \in X \mid l \land y \leq z \}$$
Homomorphisms of Prelattices and Heyting Prelattices

- A homomorphism of prelimaties is a function $f : X \to Y$ (with $X$ and $Y$ prelimaties) which preserves finite meets and joins, that is

$$f(\bigwedge\{x_1, \ldots, x_n\}) \cong \bigwedge\{f(x_1), \ldots, f(x_n)\}$$

and

$$f(\bigvee\{x_1, \ldots, x_n\}) \cong \bigvee\{f(x_1), \ldots, f(x_n)\}$$

and also $f(\top) \cong \top$ and $f(\bot) \cong \bot$.

- A homomorphism of Heyting prelimaties is as above but also preserves Heyting implications.
Examples of Homomorphisms of Prelattices

- Consider the inverse image function $f^{-1} : P(Y) \to P(X)$. This is a homomorphism of Heyting prelattices.

- The function that multiplies by a natural number $k$ preserves lowest common multiples, that is, preserves binary joins in $(\mathbb{N}, |)$ where recall $d \mid m$ means that $d$ divides exactly into $m$. 
Definition of a Category

A category $\mathcal{C}$ is specified by the following data:

- A collection $\text{ob} \mathcal{C}$ of entities called objects, written $A, B, C \ldots$

- A collection $\text{mor} \mathcal{C}$ of entities called morphisms written $f, g, h \ldots$

- For each morphism $f$ a source $\text{src}(f)$ which is an object of $\mathcal{C}$ and a target $\text{tar}(f)$ also an object of $\mathcal{C}$. We shall write $f : \text{src}(f) \rightarrow \text{tar}(f)$ or perhaps $f : A \rightarrow B$. 
Morphisms $f$ and $g$ are composable if $\text{tar}(f) = \text{src}(g)$. If $f : A \to B$ and $g : B \to C$, then there is a morphism $g \circ f : A \to C$.

For each object $A$ of $C$ there is an identity morphism $id_A : A \to A$, where

$$id_{\text{tar}(f)} \circ f = f$$

$$f \circ id_{\text{src}(f)} = f$$

Composition is associative, that is given morphisms $f : A \to B$, $g : B \to C$ and $h : C \to D$ then

$$(h \circ g) \circ f = h \circ (g \circ f).$$

If the collection of morphisms from $A$ to $B$ forms a set, then we write $C(A,B)$ for this set. Such categories are called locally small.
■ Sets and total functions, $\mathbf{Set}$. The objects are sets and morphisms are $(A, f, B)$ where $f \subseteq A \times B$ is a function. Composition is given by

$$(B, g, C) \circ (A, f, B) = (A, g \circ f, C)$$

Finally, if $A$ is any set, the identity is $(A, id_A, A)$.

■ Any preordered set $(X, \leq)$ is a category. The objects are elements of $X$. The collection of morphisms is the set of pairs $(x, y)$ where $x \leq y$. Composition is $$(y, z) \circ (x, y) \overset{\text{def}}{=} (x, z)$$
(because $\leq$ is transitive). The identities are the pairs $(x, x)$
(because $\leq$ is reflexive).
Definition of a Functor

A functor $F : \mathcal{C} \to \mathcal{D}$ is specified by

- an operation taking objects $A$ in $\mathcal{C}$ to objects $FA$ in $\mathcal{D}$, and
- an operation sending morphisms $f : A \to B$ in $\mathcal{C}$ to morphisms $Ff : FA \to FB$ in $\mathcal{D}$,

for which $F(id_A) = id_{FA}$, and whenever the composition of morphisms $g \circ f$ is defined in $\mathcal{C}$ we have $F(g \circ f) = Fg \circ Ff$. 
Examples of Functors

- The functors between two preorders $A$ and $B$ are precisely the monotone functions from $A$ to $B$.

- The set $[A]$ of finite lists over a set $A$ gives a monoid via list concatenation.

Hence we may define $F : \text{Set} \to \text{Mon}$ by $FA \overset{\text{def}}{=} [A]$ and $Ff \overset{\text{def}}{=} \text{map}(f)$, where $\text{map}(f) : [A] \to [B]$ is defined by

$$\text{map}(f)([a_1, \ldots, a_n]) = [f(a_1), \ldots, f(a_n)],$$

with $[a_1, \ldots, a_n]$ any element of $[A]$. 
To see that $F(g \circ f) = Fg \circ Ff$ where $A \xrightarrow{f} B \xrightarrow{g} C$ note that

$$F(g \circ f)([a_1, \ldots, a_n]) \overset{\text{def}}{=} map(g \circ f)([a_1, \ldots, a_n])$$

$$= [(g \circ f)(a_1), \ldots, (g \circ f)(a_n)]$$

$$= [g(f(a_1)), \ldots, g(f(a_n))]$$

$$= map(g)([f(a_1), \ldots, f(a_n)])$$

$$= map(g)(map(f)([a_1, \ldots, a_n]))$$

$$= (Fg \circ Ff)([a_1, \ldots, a_n]).$$
Let $\mathcal{C}$ be a category. The identity functor $id_{\mathcal{C}}$ is defined by $id_{\mathcal{C}}(A) \overset{\text{def}}{=} A$ where $A$ is an object of $\mathcal{C}$ and $id_{\mathcal{C}}(f) \overset{\text{def}}{=} f$ where $f$ is a morphism of $\mathcal{C}$.

Given a set $A$, recall that the powerset $\mathcal{P}(A)$ is the set of subsets of $A$. We can define a functor $\mathcal{P} : \text{Set} \rightarrow \text{Set}$ which is given by

$$f : A \rightarrow B \quad \mapsto \quad f_* : \mathcal{P}(A) \rightarrow \mathcal{P}(B),$$

where $f : A \rightarrow B$ is a function and $f_*$ is defined by $f_*(A') \overset{\text{def}}{=} \{ f(a') \mid a' \in A' \}$ where $A' \in \mathcal{P}(A)$. We call $P : \text{Set} \rightarrow \text{Set}$ the covariant powerset functor.
Given a category \( C \), the category \( C^{\text{op}} \) has objects those of \( C \), and morphisms \( f^{\text{op}} : A \to B \) in \( C^{\text{op}} \) are morphisms \( f : B \to A \) in \( C \). Composition is \( g^{\text{op}} \circ f^{\text{op}} \overset{\text{def}}{=} (f \circ g)^{\text{op}} \).

We can define a functor \( \mathcal{P} : \text{Set}^{\text{op}} \to \text{Set} \) by setting

\[
f : B \to A \quad \mapsto \quad f^{-1} : \mathcal{P}(B) \to \mathcal{P}(A),
\]

where \( f : A \to B \) is a function in \( \text{Set} \), and the function \( f^{-1} \) is defined by \( f^{-1}(B') \overset{\text{def}}{=} \{ a \in A \mid f(a) \in B' \} \) where \( B' \in \mathcal{P}(B) \).

Note that the source of the functor is an opposite category. We refer to \( \mathcal{P} \) as the contravariant powerset functor.
Let $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{C}' \to \mathcal{D}$ be functors. The comma category $(F \downarrow G)$ has objects triples $(A, f, A')$ where $A$ and $A'$ are objects of $\mathcal{C}$ and $\mathcal{C}'$ respectively and $f : FA \to GA'$ is a morphism of $\mathcal{D}$. A morphism $(A, f, A') \to (B, f', B')$ is a pair $(g, h)$ where $g : A \to B$ in $\mathcal{C}$ and $h : A' \to B'$ in $\mathcal{C}'$ for which the following diagram commutes:

\[
\begin{array}{ccc}
FA & \xrightarrow{Fg} & FB \\
\downarrow{f} & & \downarrow{f'} \\
GA' & \xrightarrow{Gh} & GB'
\end{array}
\]
Definition of a Natural Transformation

Let \( F, G : \mathcal{C} \rightarrow \mathcal{D} \) be functors. Then a natural transformation \( \alpha \) from \( F \) to \( G \), written \( \alpha : F \rightarrow G \), is specified by giving a morphism \( \alpha_A : FA \rightarrow GA \) in \( \mathcal{D} \) for each object \( A \) in \( \mathcal{C} \), such that for any \( f : A \rightarrow B \) in \( \mathcal{C} \), we have

\[
\begin{align*}
FA & \xrightarrow{\alpha_A} GA \\
\downarrow Ff & \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \downarrow Gf \\
FB & \xrightarrow{\alpha_B} GB
\end{align*}
\]
Examples of Natural Transformations

- Recall $F : \text{Set} \to \text{Mon}$ where $FA \overset{\text{def}}{=} [A]$ and $Ff \overset{\text{def}}{=} \text{map}(f)$ and $\text{map}(f) : [A] \to [B]$. We can define a natural transformation $\text{rev} : F \to F$ by

$$\text{rev}_A([a_1, \ldots, a_n]) \overset{\text{def}}{=} [a_n, \ldots, a_1]$$

We check

$$(Ff \circ \text{rev}_A)([a_1, \ldots, a_n]) = [f(a_n), \ldots, f(a_1)] = (\text{rev}_B \circ Ff)([a_1, \ldots, a_n]).$$
Let \( C \) and \( D \) be categories and let \( F, G, H \) be functors from \( C \) to \( D \). Also let \( \alpha : F \to G \) and \( \beta : G \to H \) be natural transformations. We can define a natural transformation \( \beta \circ \alpha : F \to H \) by setting the components to be

\[
(\beta \circ \alpha)_A \overset{\text{def}}{=} \beta_A \circ \alpha_A.
\]

This yields a category \([C, D]\) with objects functors from \( C \) to \( D \), morphisms natural transformations between such functors, and composition as given above.
Define a functor $F_X : \text{Set} \to \text{Set}$ by

- $F_X(A) \overset{\text{def}}{=} (X \Rightarrow A) \times X$ on objects
- $F_X(f) \overset{\text{def}}{=} (f \circ -) \times id_X$ on morphisms

Then define a natural transformation $ev : F_X \to id_{\text{Set}}$ by

$ev_A(g,x) \overset{\text{def}}{=} g(x)$ where $(g,x) \in (X \Rightarrow A) \times X$. To see that we have defined a natural transformation $ev_A : (X \Rightarrow A) \times X \to A$

let $f : A \to B$ and $(g,x) \in (X \Rightarrow A) \times X$ and note that

$$(id_{\text{Set}}(f) \circ ev_A)(g,x) = f(ev_A(g,x))$$

$$= \ldots (ev_B \circ F_X(f))(g,x).$$
Isomorphisms and Equivalences

- A morphism $f : A \rightarrow B$ is an isomorphism if there is some $g : B \rightarrow A$ for which $f \circ g = id_B$ and $g \circ f = id_A$.

- We shall say $g$ is an inverse for $f$ and vise versa.

- We say that $A$ is isomorphic to $B$, $A \cong B$, if such a mutually inverse pair of morphisms exists.
An isomorphism $\alpha : F \cong G$ in a functor category is referred to as a natural isomorphism. This is the same as having a collection of isomorphisms $FA \cong GA$ for each object $A$ which are “natural in $A\ldots$ ”

Two categories $C$ and $D$ are equivalent if there are functors $F : C \to D$ and $G : D \to C$ together with natural isomorphisms $\varepsilon : FG \cong \text{id}_D$ and $\eta : \text{id}_C \cong GF$. We say that $F$ is an equivalence with an inverse equivalence $G$ and denote the equivalence by $F : C \simeq D : G$. 
Examples

- **Bijections in** $\text{Set}$ **are isomorphisms.**

- In the category determined by a partially ordered set, the only isomorphisms are the identities, and in a preorder $X$ with $x, y \in X$ we have $x \cong y$ iff $x \leq y$ and $y \leq x$. Note that in this case there can be only one pair of mutually inverse morphisms witnessing the fact that $x \cong y$.

- Let an object of $1/\text{Set}$ be a pair $(A, a)$ where $a \in A$ and a morphism $g : (A, a) \rightarrow (B, b)$ be a function $g : A \rightarrow B$ for which $b = g(a)$. Let $\text{Part}$ be the category of sets and partial functions. Then $\text{Part} \simeq 1/\text{Set}$. 
Definition of Binary Products

A binary product of objects $A$ and $B$ in a category $C$ is specified by

- an object $A \times B$ of $C$, together with
- two projection morphisms $\pi_A : A \times B \to A$ and $\pi_B : A \times B \to B$,

for which given any object $C$ and morphisms $f : C \to A$, $g : C \to B$, there is a unique morphism $\langle f, g \rangle : C \to A \times B$ for which $\pi_A \circ \langle f, g \rangle = f$ and $\pi_B \circ \langle f, g \rangle = g$. 
The data for a binary product is more readily understood as a commutative diagram,

\[
\begin{array}{ccc}
A & \xrightarrow{\pi_A} & A \times B \\
\downarrow & & \downarrow \pi_B \\
C & \xrightarrow{\exists ! \langle f, g \rangle} & B
\end{array}
\]

The unique morphism \( \langle f, g \rangle : C \rightarrow A \times B \) is called the mediating morphism for \( f \) and \( g \).

The definition can be extended to families of objects \( (A_i \mid i \in I) \).
Definition of Finite Products

Given a family of objects in $C$, a product is specified by

- an object $\Pi_{i \in I} A_i$ in $C$, and
- for every $j \in I$, a morphism $\pi_j : \Pi_{i \in I} A_i \to A_j$ in $C$ called the $j$th product projection,

such that for any $C$ and $(f_i : C \to A_i \mid i \in I)$ there is a unique morphism

$$\langle f_i \mid i \in I \rangle : C \to \Pi_{i \in I} A_i$$

for which given any $j \in I$, we have $\pi_j \circ \langle f_i \mid i \in I \rangle = f_j$. 
Examples

- A binary product of $x$ and $y$ in a preordered set $X$ is given by $x \land y$ with projections $x \land y \leq x$ and $x \land y \leq y$.

- A (non-empty) finite product of $(A_i \mid i \in I)$ in $Set$ is given by the cartesian product $\prod_{i \in I} A_i$. The product of the empty family is a terminal object $1$, with the property that there is a unique morphism $!_A : A \to 1$ for every $A$. 
Definition of Binary Coproducts

A binary coproduct of $A$ and $B$ is specified by

- an object $A + B$, together with
- two insertion morphisms $\iota_A : A \to A + B$ and $\iota_B : B \to A + B$,

such that there is a unique $[f, g]$ for which

$$
\begin{array}{ccc}
C & \xleftarrow{\exists! [f, g]} & A + B \\
\downarrow & \nearrow & \downarrow \iota_A \swarrow \\
A & & B
\end{array}
$$
Definition of Cartesian Closed Categories

- \( C \) is cartesian closed if it has finite products, and for any \( B \) and \( C \) there is \( B \Rightarrow C \) and morphism
  
  \[ ev : (B \Rightarrow C) \times B \rightarrow C \]

  such that for any \( f : A \times B \rightarrow C \) there is a unique morphism \( \lambda(f) : A \rightarrow (B \Rightarrow C) \) such that \( f = ev \circ (\lambda(f) \times id_B) \).

- \( B \Rightarrow C \) is called the exponential of \( B \) and \( C \)

- \( \lambda(f) \) is the exponential mate of \( f \).
Examples

- The category $\textbf{Set}$.
  - The terminal object is $\{\emptyset\}$ and binary products are given by cartesian product.
  - $B \rightarrow C$ is the set of functions from $B$ to $C$.
  - The function $ev : (B \rightarrow C) \times B \rightarrow C$ is given by $ev(h,b) = h(b)$, where $b \in B$ and $h : B \rightarrow C$ is a function.
  - Given $f : A \times B \rightarrow C$ we define $\lambda(f) : A \rightarrow (B \rightarrow C)$ by $\lambda(f)(a)(b) = f(a,b)$.

- A Heyting prelattice viewed as a category is indeed cartesian closed, with Heyting implications as exponentials. In fact such a prelattice also has finite coproducts.
Definition of Distributive and Bicartesian Closed Categories

- A category with finite products and coproducts is said to be distributive if the mediating morphisms

\[ [id_A \times i, id_A \times j] : (A \times B) + (A \times C) \xrightarrow{\cong} A \times (B + C) \]

and \( !_{A \times 0} : 0 \xrightarrow{\cong} A \times 0 \) are isomorphisms.

- A category \( C \) is a bicartesian closed category if it is a cartesian closed category which has finite coproducts.
Examples

■ The category $\text{Set}$.

■ Any category $[C, \text{Set}]$. Categorical structure is defined pointwise meaning, for example, that $(F \times G)(A) \overset{\text{def}}{=} FA \times GA$ and so on.

■ Any Heyting prelattice which is regarded as a category.

■ In fact any bicartesian closed category is automatically distributive—we will see why this is so later on.
Functors Preserving Products

- The functor $F : C \rightarrow D$ preserves finite products if for any finite family of objects $(A_1, \ldots, A_n)$ in $C$ the morphism

$$m \overset{\text{def}}{=} \langle F\pi_i \mid i \in I \rangle : F(A_1 \times \ldots \times A_n) \rightarrow FA_1 \times \ldots \times FA_n$$

is an isomorphism.

- We refer to $m$ as the canonical isomorphism.

- $F$ is strict if the above isomorphisms are identities.

- The functor $\mathcal{C}(C, -)$ preserves finite products.
Functors Preserving Coproducts and Exponentials

A functor $F : \mathcal{C} \rightarrow \mathcal{D}$ is said to be bicartesian closed if it preserves finite products, coproducts and exponentials.

We shall also call such a functor a morphism of bicartesian closed categories.
Adjunctions between Preorders

- A pair of monotone functions

\[
\begin{align*}
X & \leftrightarrow Y \\
\downarrow & \quad \downarrow \\
l & \quad r
\end{align*}
\]

is said to be an adjunction if for all \( x \in X \) and \( y \in Y \),

\[
l(x) \leq y \iff x \leq r(y)
\]

- We say that \( l \) is left adjoint to \( r \) and that \( r \) is right adjoint to \( l \). We write \( l \dashv r \).
Examples

■ Let $1 \overset{\text{def}}{=} \{ \ast \}$ be the one element preorder. Then there are adjunctions $(\bot \dashv ! \dashv \top)$

$$
\begin{array}{ccc}
X & \overset{!}{\longrightarrow} & 1 \\
\downarrow & & \downarrow \\
\bot & \overset{!}{\longleftarrow} & \top
\end{array}
$$

provided that $X$ has both top and bottom elements. For example, for any $x \in X$,

$$
!(x) \overset{\text{def}}{=} \ast \leq \ast \iff x \leq \top(\ast) \overset{\text{def}}{=} \top
$$
Examples

- Define $\Delta : X \to X \times X$ by $\Delta(x) \overset{\text{def}}{=} (x, x)$. Then there are adjoints $(\lor \dashv \Delta \dashv \land)$

\[
\begin{array}{c}
\Delta \\
X & \xrightarrow{\lor} & X \times X & \xleftarrow{\Delta} & X & \xleftarrow{\land} & X \times X \\
\end{array}
\]

provided that $X$ has both binary meets and joins. For example, for any $l \in X$,

\[
\Delta(l) \overset{\text{def}}{=} (l, l) \leq (x, x') \iff l \leq \land(x, x') \overset{\text{def}}{=} x \land x'
\]
Adjunctions between Categories

Let $L : C \to D$ and $R : D \to C$ be functors. $L$ is left adjoint to $R$, written $L \dashv R$, if given any objects $A$ of $C$ and $B$ of $D$ we have

- a bijection between morphisms $LA \to B$ in $D$ and $A \to RB$ in $C$,

  $f : LA \to B$ \quad $g : A \to RB$

  $\overline{f} : A \to RB$ \quad $\hat{g} : LA \to B$

- this bijection is natural in $A$ and $B$: given morphisms $a : A' \to A$ in $C$ and $b : B \to B'$ in $D$ we have

  $b \circ f \circ La = Rb \circ \overline{f} \circ a$ and $(Rb \circ g \circ a)^\wedge = b \circ \hat{g} \circ La$. 

Notation for Adjunctions

Let $A$ and $B$ be objects of a locally small category $\mathcal{C}$. We define a functor

$$\mathcal{C}(-, +) : \mathcal{C}^{op} \times \mathcal{C} \to \text{Set}$$

by taking any morphism $(f, g) : (A, B) \to (A', B')$ in $\mathcal{C}^{op} \times \mathcal{C}$ to the set-theoretic function

$$\mathcal{C}(f, g) : \mathcal{C}(A, B) \to \mathcal{C}(A', B')$$

where $\mathcal{C}(f, g)(h) = g \circ h \circ f$ for each morphism $h : A \to B$. (Note that $f$ is a morphism $A' \to A$ in $\mathcal{C}$).
We can also define

\[ C(A, +) : C \rightarrow \text{Set}. \]

This takes \( B \) of \( C \) to the set \( C(A, B) \), and if \( g : B \rightarrow B' \) is a morphism of \( C \) then the functor \( C(A, +) \) takes \( g : B \rightarrow B' \) to the function

\[ C(A, g) : C(A, B) \rightarrow C(A, B') \]

defined by setting

\[ C(A, g)(h) \overset{\text{def}}{=} g \circ h, \]

where \( h : A \rightarrow B \).

Similarly, we can define a functor \( C( -, B) : C^{op} \rightarrow \text{Set} \).
If a categories $\mathcal{C}$ and $\mathcal{D}$ are locally small, then $L \dashv R$ provided that there is an isomorphism

$$\mathcal{D}(-, +) \circ (L^{op} \times id) \overset{\text{def}}{=} \mathcal{D}(L-, +) \cong \mathcal{C}(-, R+) \overset{\text{def}}{=} \mathcal{C}(-, +) \circ (id \times R)$$

in the functor category $[\mathcal{C}^{op} \times \mathcal{D}, Set]$ where $L^{op} : \mathcal{C}^{op} \rightarrow \mathcal{D}^{op}$ is defined by

$$L^{op}(f : A' \rightarrow A) \overset{\text{def}}{=} Lf : LA' \rightarrow LA$$

We also say that $R$ is right adjoint to $L$. 
Examples

The *forgetful* functor $U : \text{Mon} \to \text{Set}$ taking a monoid to its underlying set, and the functor $[-] : \text{Set} \to \text{Mon}$ taking a set to lists over the set, are adjoints: $[-] \dashv U$.

Given a monoid $M$ and a set $A$ any function $g : A \to UM$ corresponds to a unique monoid morphism $\hat{g} : [A] \to M$. 
Indeed, there is a bijection for each $A$ and $M$

$$(\overline{(-)}): \text{Mon}([A], M) \cong \text{Set}(A, UM): (\overline{(-)})$$

given by

$$g: A \to UM \quad \mapsto \quad \hat{g}: [A] \quad [a_1, \ldots, a_n] \mapsto g(a_1) \cdots g(a_n) \quad M,$$

$$f: [A] \to M \quad \mapsto \quad \overline{f}: A \quad a \mapsto f([a]) \quad UM.$$
and a natural isomorphism

$$\Psi : \text{Mon}([-], +) \cong \text{Set}(-, U(+)) : \text{Set}^{\text{op}} \times \text{Mon} \to \text{Set}$$

where at any object \((A, M)\) of \(\text{Set}^{\text{op}} \times \text{Mon}\) we have

$$(\Psi)_{(A, M)} \overset{\text{def}}{=} (-)$$

and (say)

$$\Psi^{-1}_{(A, M)} \overset{\text{def}}{=} (\hat{-}).$$
There are other examples of forgetful functors. The functor $U : C \to Graph$ taking a category to its underlying graph has a left adjoint taking a graph to the free category over the graph.

The functor $U : Group \to Set$ taking a group to its underlying set has a left adjoint taking a set to the free group over the set.
\( F : \text{Set} \rightarrow \text{Vec}_K \) is the functor taking set \( X \) to the vector space \( FX \) with vectors linear combinations \( \Sigma_{i \in n} k_i x_i \) where \( x_i \in X \) and \( k_i \in K \). Given function \( f : X \rightarrow Y \), the linear map \( Ff : FX \rightarrow FY \) is

\[
Ff(\Sigma_{i \in n} k_i x_i) \overset{\text{def}}{=} \Sigma_{i \in n} k_i f(x_i).
\]

The functor \( U : \text{Vec}_K \rightarrow \text{Set} \) is forgetful. Any function \( g : X \rightarrow UV \) has a unique extension to a linear map \( \hat{g} : FX \rightarrow V \). The assignment \( g \mapsto \hat{g} \) has an inverse: any linear \( f : FX \rightarrow V \) restricts to function \( \overline{f} : X \rightarrow UV \). Thus we have a natural bijection

\[
\overline{(-)} : \text{Set}(X, UV) \leftrightarrow \text{Vec}_K(FX, U) : \widehat{(-)}
\]
The diagonal functor \( \Delta : \text{Set} \to \text{Set} \times \text{Set} \) taking a function \( f : A \to B \) to \( (f, f) : (A, A) \to (B, B) \) has right and left adjoints \( \Pi \) and \( \Sigma \) taking any morphism \( (f, g) : (A, A') \to (B, B') \) of \( \text{Set} \times \text{Set} \) to \( f \times g \overset{\text{def}}{=} \langle f \circ \pi_A, g \circ \pi_B \rangle : A \times A' \to B \times B' \) and \( f + g \overset{\text{def}}{=} [\iota_B f, \iota_B' g] : A + A' \to B + B' \) respectively, where

\[
\begin{array}{ccc}
A & \xrightarrow{\pi_A} & A \times A' & \xrightarrow{\pi_{A'}} & A' \\
\downarrow{\pi_A} & & \downarrow{\pi_{A'}} & & \\
A & \xrightarrow{\iota_B} & B + B' & \xleftarrow{\iota_{B'}} & B'
\end{array}
\]

This example remains valid if we replace \( \text{Set} \) by any category \( C \), where we leave the reader to define the diagonal functor \( \Delta : C \to C \times C \).
Let \( C \) be a category with finite products. Existence of a right adjoint \( R \) to the functor \((-) \times B : C \to C\) for each object \( B \) of \( C \), is equivalent to \( C \) being cartesian closed.

\((\Rightarrow)\) Given an object \( B \) of \( C \) set \( B \Rightarrow C \overset{\text{def}}{=} R(C) \) for any object \( C \) of \( C \). Given a morphism \( f : A \times B \to C \) we define \( \lambda(f) : A \to (B \Rightarrow C) \) to be the mate of \( f \) across the given adjunction. The morphism

\[
ev : (B \Rightarrow C) \times B \to C
\]

is the mate \( (id_{B \Rightarrow C})^* \) of the identity \( id_{B \Rightarrow C} : (B \Rightarrow C) \to (B \Rightarrow C) \).
Conversely, let $B$ be an object of $\mathcal{C}$. We define a right adjoint to $(-) \times B$ denoted by $B \Rightarrow (-)$, by setting

$$c : C \rightarrow C' \mapsto B \Rightarrow c \overset{\text{def}}{=} \lambda(c \circ ev) : (B \Rightarrow C) \rightarrow (B \Rightarrow C')$$

for each morphism $c : C \rightarrow C'$ of $\mathcal{C}$. We define a bijection by declaring the mate of $f : A \times B \rightarrow C$ to be $\lambda(f) : A \rightarrow (B \Rightarrow C)$ and the mate of $g : A \rightarrow (B \Rightarrow C)$ to be

$$g^* \overset{\text{def}}{=} ev \circ (g \times id_B) : A \times B \rightarrow C.$$
The Yoneda Lemma — Preliminaries

Let $C$ be a locally small category, $F : C \to \text{Set}$ a functor and $A$ an object of $C$. Then the collection $Nat(C(A,+),F)$ of natural transformations $C(A,+) \to F$ is a set and so we can define a functor

$$Nat(C(-,+),+): C \times [C, \text{Set}] \to \text{Set}$$

The morphism $(g,\mu):(A,F) \to (A',F')$ in $C \times [C, \text{Set}]$ is taken to the function

$$Nat(C(g,+),\mu): Nat(C(A,+),F) \to Nat(C(A',+),F')$$

$$Nat(C(g,+),\mu)(\alpha) \overset{\text{def}}{=} \mu \circ \alpha \circ C(g,+) \text{ where } \alpha : C(A,+) \to F \text{ is a natural transformation.}$$
The Yoneda Lemma

There is an “evaluation” functor

$$Ev : C \times [C, Set] \rightarrow Set.$$ 

Then there is a natural isomorphism

$$\Phi : Nat(C(-, +), +) \cong Ev : \Psi$$

If $A$ is an object of $C$, this amounts to saying that there is an isomorphism (set-theoretic bijection)

$$\Phi_{(A,F)} : Nat(C(A, +), F) \cong FA : \Psi_{(A,F)}$$

and this isomorphism is natural in $(A,F)$. 

Categorical Type Theory

- We shall define an equational type theory with products, sums, and functions.

- Working from first principles, we shall derive a semantics.
  - First we examine the rules for deriving type assignments, and show that basic properties lead naturally to categorical models.
  - Second, we examine each of the rules for deriving equations, and extract constraints on our models which guarantee soundness.
Categorical Type Theory

- We show how structure preserving functors can transform one model into another …
- and use this to show how theories correspond to categories with a universal property.
Signatures

A \( \lambda \times + \)-signature, \( Sg \), is given by:

- A collection of ground types. The collection of types is inductively defined:

  \[
  \gamma \quad \text{unit} \quad \text{null} \quad \sigma \times \tau \quad \sigma + \tau \quad \sigma \Rightarrow \tau
  \]

- A collection of function symbols \( f : \sigma_1 \ldots \sigma_a \to \sigma \) which may be constants \( k : \sigma \) when \( a = 0 \).
We define the raw terms generated by a $\lambda \times +$-signature:

- $\text{f}(M_1, \ldots, M_a)$
- $\langle \rangle$
- $\langle M, N \rangle$
- $\text{Fst}(P)$
- $\text{Snd}(P)$

- $\text{Emp}_\sigma(S)$
- $\text{Inr}_\tau(M)$
- $\text{Inl}_\tau(M)$
- $\text{Case}(S, x.E \mid y.F)$

- $\lambda x : \sigma.M$
- $FA$

We will use simultaneous substitution of raw terms for free variables, $T[\vec{U} / \vec{v}]$. For example, $\langle x, y \rangle[\text{Inl}(y), x/x, y] = \langle \text{Inl}(y), x \rangle$. 
Proved Terms

- A context is a finite list of (variable, type) pairs, usually written as $\Gamma = [x_1 : \sigma_1, \ldots, x_n : \sigma_n]$, where the variables are required to be distinct.

- A term-in-context is a judgement of the form $\Gamma \vdash M : \sigma$

- Given a signature $Sg$, the proved terms are those terms-in-context which are inductively generated by the following rules.
\[
\begin{array}{c}
\Gamma, x : \sigma, \Gamma' \vdash x : \sigma \\
\Gamma \vdash k : \sigma
\end{array}
\quad
\begin{array}{c}
\Gamma \vdash M_1 : \sigma_1 \\
\ldots
\Gamma \vdash M_a : \sigma_a
\end{array}
\quad
\begin{array}{c}
\Gamma \vdash f(M_1, \ldots, M_a) : \tau
\end{array}
\]
\[\Gamma \vdash S : \text{null}\]
\[\Gamma \vdash M : \sigma\]
\[\Gamma \vdash N : \tau\]

\[\Gamma, x : \sigma \vdash E : \delta\]
\[\Gamma, y : \tau \vdash F : \delta\]

\[\Gamma \vdash \text{Case}(S, x.E \mid y.F) : \delta\]

\[\Gamma \vdash \lambda x : \sigma. M : \sigma \Rightarrow \tau\]
\[\Gamma \vdash F : \sigma \Rightarrow \tau\]
\[\Gamma \vdash A : \sigma\]
\[\Gamma \vdash FA : \tau\]
Admissible Properties

Whenever $Sg \triangleright \Gamma \vdash M : \sigma$, we have $Sg \triangleright \pi \Gamma \vdash M : \sigma$.

We use rule induction. More precisely we prove

$$\forall Sg \triangleright \Gamma \vdash M : \sigma. \quad Sg \triangleright \pi \Gamma \vdash M : \sigma$$

We give some examples of property closure.
\[
\Gamma \vdash M_1 : \sigma_1 \quad \ldots \quad \Gamma \vdash M_a : \sigma_a \\
\Gamma \vdash f(M_1, \ldots, M_a) : \sigma
\]

(Property Closure for the inductive rule for function symbols): The inductive hypotheses are 
\[ Sg \triangleright \pi \Gamma \vdash M_i : \sigma_i \]
for each \( i \), that is, there is a derivation for each term-in-context. But now we can just apply an instance of the rule to these derivations to deduce that 
\[ Sg \triangleright \pi \Gamma \vdash f(M_1, \ldots, M_a) : \sigma, \] as required.
Theories

- A \( \lambda \times + \)-theory, \( Th \), is a pair \((Sg, Ax)\) where \( Ax \) is a collection of equations-in-context for \( Sg \).

- An equation-in-context is a judgement \( \Gamma \vdash M = M' : \sigma \) where \( \Gamma \vdash M : \sigma \) and \( \Gamma \vdash M' : \sigma \) are proved terms.

- The theorems of \( Th \) consist of the judgements of the form \( \Gamma \vdash M = M' : \sigma \) inductively generated by the rules on the following slides—it is a consequence of the rules that \( Sg \triangleright \Gamma \vdash M : \sigma \) and \( Sg \triangleright \Gamma \vdash M' : \sigma \).
\[
\begin{align*}
Ax & \quad \Gamma \vdash M = M' : \sigma \\
\hline
\Gamma \vdash M = M' : \sigma
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash M = M' : \sigma \\
\pi \Gamma \vdash M = M' : \sigma \quad \text{(where } \pi \text{ is a permutation)}
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash M = M' : \sigma \\
\Gamma' \vdash M = M' : \sigma \quad \text{(where } \Gamma \subseteq \Gamma' \text{)}
\end{align*}
\]

\[
\begin{align*}
\Gamma, x : \sigma \vdash N = N' : \tau & \quad \Gamma \vdash M = M' : \sigma \\
\Gamma \vdash N[M/x] = N'[M'/x] : \tau
\end{align*}
\]

plus rules to ensure that \( = \) is an equivalence relation.
\[
\begin{align*}
Sg & \triangleright \Gamma \vdash M : \text{unit} \\
\Gamma & \vdash M = \langle \rangle : \text{unit} \\
Sg & \triangleright \Gamma \vdash M : \sigma \quad Sg & \triangleright \Gamma \vdash N : \tau \\
\Gamma & \vdash \text{Fst}(\langle M, N \rangle) = M : \sigma \\
Sg & \triangleright \Gamma \vdash N : \tau \\
\Gamma & \vdash \text{Snd}(\langle M, N \rangle) = N : \tau \\
Sg & \triangleright \Gamma \vdash P : \sigma \times \tau \\
\Gamma & \vdash \langle \text{Fst}(P), \text{Snd}(P) \rangle = P : \sigma \times \tau
\end{align*}
\]
\[
\begin{align*}
Sg & \triangleright \Gamma \vdash S : \text{null} & Sg & \triangleright \Gamma, x : \text{null} \vdash M : \sigma \\
\hline
\Gamma \vdash \text{Emp}_\sigma(S) = M[S/x] : \sigma \\
Sg & \triangleright \Gamma \vdash M : \sigma & Sg & \triangleright \Gamma, x : \sigma \vdash E : \delta & Sg & \triangleright \Gamma, y : \tau \vdash F : \delta \\
\hline
\Gamma \vdash \text{Case}(\text{Inl}_\tau(M), x.E \mid y.F) = E[M/x] : \delta \\
Sg & \triangleright \Gamma \vdash N : \tau & Sg & \triangleright \Gamma, x : \sigma \vdash E : \delta & Sg & \triangleright \Gamma, y : \tau \vdash F : \delta \\
\hline
\Gamma \vdash \text{Case}(\text{Inr}_\sigma(N), x.E \mid y.F) = F[N/x] : \delta
\end{align*}
\]
\[
Sg \triangleright \Gamma \vdash S : \sigma + \tau \quad Sg \triangleright \Gamma, z : \sigma + \tau \vdash L : \delta \\
\Gamma \vdash \text{Case}(S, x.L[\text{lnl}_\tau(x)/z] \mid y.L[\text{lnr}_\sigma(y)/z]) = L[S/z] : \delta
\]

(provided \(x, y \not\in \text{fv}(L)\))

\[
\begin{align*}
\Gamma \vdash S &= S' : \sigma + \tau & \Gamma, x : \sigma \vdash E &= E' : \delta \\
\Gamma, y : \tau \vdash F &= F' : \delta
\end{align*}
\]

\[
\Gamma \vdash \text{Case}(S, x.E \mid y.F) = \Gamma \vdash \text{Case}(S', x.E' \mid y.F') : \delta
\]
\[
\begin{align*}
Sg \uprel{} \Gamma, x : \sigma \vdash M : \tau \\
\frac{Sg \uprel{} \Gamma \vdash A : \sigma}{\Gamma \vdash (\lambda x : \sigma. M) A = M[A/x] : \tau}
\end{align*}
\]

\[
\begin{align*}
Sg \uprel{} \Gamma \vdash F : \sigma \Rightarrow \tau \\
\frac{Sg \uprel{} \Gamma \vdash \lambda x : \sigma.(Fx) = F : \sigma \Rightarrow \tau}{\text{(provided } x \notin \text{fv}(F))}
\end{align*}
\]

\[
\begin{align*}
\Gamma, x : \sigma \vdash M = M' : \tau \\
\frac{\Gamma \vdash \lambda x : \sigma.M = \lambda x : \sigma. M' : \sigma \Rightarrow \tau}{\Gamma \vdash \lambda x : \sigma.M = \lambda x : \sigma. M' : \sigma \Rightarrow \tau}
\end{align*}
\]
Deriving a Semantics For Proved Terms

■ Suppose we model (or interpret) $\sigma$ and $\tau$ by “objects” $A$ and $B$. Let us model $x : \sigma \vdash M : \tau$ as a “relationship” $A \xrightarrow{m} B$.

■ We first think about the process of substitution. Let

$$\begin{align*}
[x : \sigma \vdash M : \tau] &= A \xrightarrow{m} B \\
[y : \tau \vdash N : \gamma] &= B \xrightarrow{n} C
\end{align*}$$

Then

$$[x : \sigma \vdash N[M/y] : \gamma] = A \xrightarrow{\diamond(n,m)} C$$
Let $z : \gamma \vdash L : \delta$ be a further proved term. Note that we shall identify the semantics of the proved terms

$$x : \sigma \vdash (L[N/z])[M/y] : \delta \quad \text{and} \quad x : \sigma \vdash L[N[M/y]/z] : \delta$$

Thus

$$\Box(\Box(l,n),m) = \Box(l,\Box(n,m))$$

We will have to model $x : \sigma \vdash x : \sigma$ as a relationship $A \xrightarrow{\star A} A$. We can deduce that if $E \xrightarrow{e} A$, then $\Box(\star A, e) = e$ because $x[E/x] = E$. 
We summarise our deductions, writing $n \circ m$ for $\square(n, m)$ and $id_A$ for $\star_A$, which amount to the definition of a category:

- Types are interpreted by “objects,” say $A$, $B$… and proved terms are interpreted by “relationships,” say $A \xrightarrow{m} B$…
- For each object $A$ there is a relationship $id_A$.
- Given relationships $A \xrightarrow{m} B$ and $B \xrightarrow{n} C$, there is a relationship $A \xrightarrow{n \circ m} C$.
- Given relationships $E \xrightarrow{e} A$ and $A \xrightarrow{m} B$, then we have $id_A \circ e = e$ and $m \circ id_A = m$.
- For any $A \xrightarrow{m} B$, $B \xrightarrow{n} C$ and $C \xrightarrow{l} D$, we have $l \circ (n \circ m) = (l \circ n) \circ m$. 
Summary

■ We will model a proved term $x_1 : \sigma_1, \ldots, x_n : \sigma_n \vdash M : \tau$ in a category with finite products as a morphism of the form

$$[\Gamma \vdash M : \tau] : [\Gamma] \to [\tau]$$

where $\Gamma \overset{\text{def}}{=} x_1 : \sigma_1, \ldots, x_n : \sigma_n$ and $[\Gamma]$ stands for $[\sigma_1] \times \ldots \times [\sigma_n]$.

■ Substitution of terms will be modelled by categorical composition …
Deriving a Semantics for Theories

- First we consider the types of $Sg$. We have to give an object $\lbrack \gamma \rbrack$ of $C$ to interpret each of the ground types $\gamma$, $\lbrack \text{unit} \rbrack$ to interpret unit, and $\lbrack \text{null} \rbrack$ to interpret null.

- We define $\lbrack \sigma \times \tau \rbrack \overset{\text{def}}{=} \lbrack \sigma \rbrack \Box \lbrack \tau \rbrack$, etc.

- We choose a morphism $\lbrack f \rbrack : \lbrack \sigma_1 \rbrack \times \ldots \times \lbrack \sigma_n \rbrack \to \lbrack \sigma \rbrack$ in $C$ for each function symbol.

- Recall that the interpretation of $\Gamma \vdash M : \sigma$ is given by $\lbrack \Gamma \vdash M : \sigma \rbrack : \lbrack \Gamma \rbrack \to \lbrack \sigma \rbrack$. By looking at how to soundly interpret the theorems of $Th$ we will deduce what the interpretation must be.
A typical rule looks like

\[
\frac{\Gamma \vdash M : \sigma}{\Gamma \vdash R(M) : \tau} \quad (R)
\]

Now suppose that \( m \Def [\Gamma \vdash M : \sigma] \) which is an element of \( C([\Gamma], [\sigma]) \). How do we model \( [\Gamma \vdash R(M) : \tau] \in C([\Gamma], [\tau]) \)? All we can say at the moment is that this will depend on \( m \), and we can model this idea by having a function

\[
\Phi : C([\Gamma], [\sigma]) \rightarrow C([\Gamma], [\tau])
\]

and setting \( [\Gamma \vdash R(M) : \tau] \Def \Phi(m) \).
Suppose that $x : \gamma \vdash M : \sigma$ and $y : \gamma' \vdash N : \gamma$ are any two given proved terms. If $m \overset{\text{def}}{=} \llbracket x : \gamma \vdash M : \sigma \rrbracket$ and $n \overset{\text{def}}{=} \llbracket y : \gamma' \vdash N : \gamma \rrbracket$ then
\[
\llbracket y : \gamma' \vdash M[N/x] : \sigma \rrbracket = m \circ n.
\]
Note that there are (definitionally) equal proved terms
\[
y : \gamma' \vdash R(M)[N/x] : \tau \quad \text{and} \quad y : \gamma' \vdash R(M[N/x]) : \tau.
\]
and so
\[
\Phi(m) \circ n = \Phi(m \circ n). \quad (\ast)
\]
(\ast) will hold if there are natural transformations
\[
\Phi : C(-,A) \longrightarrow C(-,B) : C^{op} \longrightarrow \text{Set}.
\]
Recall that the rule for introducing product terms is

\[ \frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash N : \sigma}{\Gamma \vdash \langle M, N \rangle : \sigma \times \tau} \]

In order to soundly interpret this rule we shall need a natural transformation

\[ \Phi : C(-, A) \times C(-, B) \rightarrow C(-, \mathsf{A} \square B) \]

for all objects \( A \) and \( B \) of \( C \).
Now let \( m : C \rightarrow A \) and \( n : C \rightarrow B \) be morphisms of \( C \). Applying naturality in \( C \) at the morphism \( \langle m, n \rangle : C \rightarrow A \times B \) we deduce

\[
\Phi_C(\pi_A \circ \langle m, n \rangle, \pi_B \circ \langle m, n \rangle) = \Phi_{A \times B}(\pi_A, \pi_B) \circ \langle m, n \rangle,
\]

that is \( \Phi_C(m, n) = \Phi_{A \times B}(\pi_A, \pi_B) \circ \langle m, n \rangle \). Now let us define the morphism \( q_{A,B} : A \times B \rightarrow A \Box B \) to be \( \Phi_{A \times B}(\pi_A, \pi_B) \). Then we can make the definition

\[
[\Gamma \vdash \langle M, N \rangle : A \times B] \overset{\text{def}}{=} [\Gamma] \frac{[\Gamma \vdash M : \sigma], [\Gamma \vdash N : \sigma]}{[\sigma] \times [\tau]} \overset{q[\sigma],[\tau]}{\rightarrow} [\sigma] \Box [\tau].
\]
\[ \Gamma \vdash H : \sigma \times \tau \]
\[ \Gamma \vdash \text{Fst}(H) : \sigma \]

To model this rule we shall need a natural transformation \( \Phi : C(\dashv, A \square B) \to C(\dashv, A) \). Using the Yoneda lemma (see notes), the components of \( \Phi \) are given by \( \theta \mapsto p \circ \theta \) for some \( p : A \square B \to A \). So now we can define

\[
[\Gamma \vdash \text{Fst}(H) : \sigma] \overset{\text{def}}{=} [\Gamma] \xrightarrow{[\Gamma \vdash H : \sigma \times \tau]} [\sigma] \square [\tau] \xrightarrow{P[\sigma],[\tau]} [\sigma].
\]
Now we think about the equations

\[
\begin{align*}
\Gamma \vdash M : \sigma & \quad \Gamma \vdash N : \tau \\
\Gamma \vdash \text{Fst}(\langle M, N \rangle) = M : \sigma
\end{align*}
\]  

(1)

\[
\begin{align*}
\Gamma \vdash M : \sigma & \quad \Gamma \vdash N : \tau \\
\Gamma \vdash \text{Snd}(\langle M, N \rangle) = N : \sigma
\end{align*}
\]  

(2)

\[
\begin{align*}
\Gamma \vdash H : \sigma \times \tau \\
\Gamma \vdash \langle \text{Fst}(H), \text{Snd}(H) \rangle = H : \sigma \times \tau
\end{align*}
\]  

(3)
If we put $h \overset{\text{def}}{=} [\Gamma \vdash H : \sigma \times \tau] : C \to A \sqcap B$, $m \overset{\text{def}}{=} [\Gamma \vdash M : \sigma] : C \to A$
and $n \overset{\text{def}}{=} [\Gamma \vdash N : \tau] : C \to B$, and our categorical interpretation satisfies the equations-in-context, this forces

\begin{align*}
p_{A,B} \circ q_{A,B} \circ \langle m, n \rangle &= m & (1) \\
p'_{A,B} \circ q_{A,B} \circ \langle m, n \rangle &= n & (2) \\
q_{A,B} \circ \langle p_{A,B} \circ h, p'_{A,B} \circ h \rangle &= h & (3)
\end{align*}

These equations imply that, up to isomorphism, $A \sqcap B$ and $A \times B$ are the same. Thus we may soundly interpret binary product types by binary categorical product.
To soundly interpret the rule

\[
\Gamma \vdash S : \text{null} \\
\overline{\quad} \\
\Gamma \vdash \text{Emp}_\sigma(S) : \sigma
\]

we shall need a natural transformation \( \Phi : C(\text{−}, N) \rightarrow C(\text{−}, A) \), where \( N = [\text{null}] \). The Yoneda Lemma tells us that the components of \( \Phi \) are given by \( \theta \mapsto n_A \circ \theta \) where \( n_A : N \rightarrow A \) is a morphism, one for each \( A \). So now we can define

\[
[\Gamma \vdash \text{Emp}_\sigma(S) : \sigma] \overset{\text{def}}{=} [\Gamma] \xrightarrow{\quad} N \xrightarrow{n_{[\sigma]}} [\sigma].
\]
If we write $s \overset{\text{def}}{=} [\Gamma \vdash S : \text{null}] : C \to N$, and
$m \overset{\text{def}}{=} [\Gamma, x : \text{null} \vdash M : \sigma] : C \times N \to A$ then

$$\Gamma \vdash \text{Emp}_\sigma(S) = M[S/x] : \sigma$$

will be soundly modelled providing that

$$n_A \circ s = m \circ \langle \text{id}_C, s \rangle \quad \text{(†)}$$

holds for any such morphisms. Suppose that $t : N \to A$. Taking $s$ to be $\text{id}_N$ and $m$ to be $t \circ \pi_N$, then

$$n_A = t \circ \pi_N \circ \langle \text{id}_N, \text{id}_N \rangle = t$$

Thus $N$ is an initial object in the category $C$. (In fact (†) forces $N$ to be distributive, that is $\pi_N : C \times N \to N$ is an isomorphism for every $C$.)
Formal Semantics of Proved Terms

Let $\mathcal{C}$ be a BCC. Then a structure, $M$, for some $Sg$ in $\mathcal{C}$ is specified by:

- For every ground type $\gamma$ an object $[\gamma]$ of $\mathcal{C}$,
- for every function symbol $f : \sigma_1 \ldots \sigma_n \rightarrow \tau$ a morphism $[f] : [\sigma_1] \times \ldots \times [\sigma_n] \rightarrow [\tau]$, where we define $[\sigma]$ by recursion, setting $[\text{unit}] \overset{\text{def}}{=} 1$, $[\sigma \times \tau] \overset{\text{def}}{=} [\sigma] \times [\tau]$ etc.

Then for every proved term $\Gamma \vdash M : \sigma$ we specify a morphism $[\Gamma \vdash M : \sigma] : [\Gamma] \rightarrow [\sigma]$ by recursion.
\[
\begin{align*}
[\Gamma, x : \sigma, \Gamma' \vdash x : \sigma] & \overset{\text{def}}{=} \pi : [\Gamma] \times [\sigma] \times [\Gamma'] \to [\sigma] \\
[\Gamma \vdash k : \sigma] & \overset{\text{def}}{=} [k] \circ ! : [\Gamma] \to 1 \to [\sigma] \\
[\Gamma \vdash M_1 : \sigma_1] & = m_1 : [\Gamma] \to [\sigma_1] \\
[\Gamma \vdash f(\vec{M}) : \tau] & = [f] \circ \langle m_1, \ldots, m_n \rangle : [\Gamma] \to ([\sigma_1] \times \ldots \times [\sigma_n]) \to [\tau]
\end{align*}
\]
(where 1 is the terminal object of $C$)

$$[\Gamma \vdash \langle \rangle : \text{unit}] \overset{\text{def}}{=} \! : [\Gamma] \rightarrow 1$$

$$[\Gamma \vdash P : \sigma \times \tau] = p : [\Gamma] \rightarrow ([\sigma] \times [\tau])$$

$$[\Gamma \vdash \text{Fst}(P) : \sigma] = \pi_1 \circ p : [\Gamma] \rightarrow ([\sigma] \times [\tau]) \rightarrow [\sigma]$$

$$[\Gamma \vdash \text{Snd}(P) : \tau] = \pi_2 \circ p : [\Gamma] \rightarrow ([\sigma] \times [\tau]) \rightarrow [\tau]$$

$$[\Gamma \vdash M : \sigma] = m : [\Gamma] \rightarrow [\sigma] \quad [\Gamma \vdash N : \tau] = n : [\Gamma] \rightarrow [\tau]$$

$$[\Gamma \vdash \langle M, N \rangle : \sigma \times \tau] = \langle m, n \rangle : [\Gamma] \rightarrow ([\sigma] \times [\tau])$$
\[
\Gamma ⊢ S : \text{null} = s : \Gamma \rightarrow 0
\]

\[
\Gamma ⊢ \text{Emp}_\sigma(S) : \sigma = ! \circ \cong \circ (id_\Gamma, s) : \Gamma \rightarrow \Gamma \times 0 \cong 0 \rightarrow [\sigma]
\]

(where 0 is the initial object of C)

\[
\begin{align*}
\Gamma ⊢ M : \sigma &= m : \Gamma \rightarrow [\sigma] \\
\Gamma ⊢ \text{Inl}_\tau(M) : \sigma + \tau &= i \circ m : \Gamma \rightarrow [\sigma] \rightarrow [\sigma] + [\tau] \\
\Gamma ⊢ N : \tau &= n : \Gamma \rightarrow [\tau] \\
\Gamma ⊢ \text{Inr}_\sigma(N) : \sigma + \tau &= j \circ n : \Gamma \rightarrow [\tau] \rightarrow [\sigma] + [\tau]
\end{align*}
\]
\[ \begin{align*}
[\Gamma \vdash S : \sigma + \tau] &= s : [\Gamma] \to [\sigma] + [\tau] \\
[\Gamma, x : \sigma \vdash E : \delta] &= e : [\Gamma] \times [\sigma] \to [\delta] \\
[\Gamma, y : \sigma \vdash F : \delta] &= f : [\Gamma] \times [\tau] \to [\delta]
\end{align*} \]

\[\begin{align*}
[\Gamma \vdash \text{Case}(S, x.E \mid y.F) : \delta] &= \\
[e, f] \circ \cong \circ \langle \text{id}_{[\Gamma]}, s \rangle : [\Gamma] \to [\Gamma] \times ([\sigma] + [\tau]) \\
&\cong ([\Gamma] \times [\sigma]) + ([\Gamma] \times [\tau]) \to [\delta]
\end{align*}\]
\[
\left[ \Gamma, x : \sigma \vdash M : \tau \right] = m : \left[ \Gamma \right] \times \left[ \sigma \right] \to \left[ \tau \right]
\]

\[
\left[ \Gamma \vdash \lambda x : \sigma. M : \sigma \Rightarrow \tau \right] = \lambda (m) : \left[ \Gamma \right] \to \left[ \sigma \right] \Rightarrow \left[ \tau \right]
\]

\[
\left[ \Gamma \vdash F : \sigma \Rightarrow \tau \right] = f : \left[ \Gamma \right] \to \left( \left[ \sigma \right] \Rightarrow \left[ \tau \right] \right) \quad \left[ \Gamma \vdash A : \sigma \right] = a : \left[ \Gamma \right] \to \left[ \sigma \right]
\]

\[
\left[ \Gamma \vdash FA : \tau \right] \overset{\text{def}}{=} ev \circ \langle f, a \rangle : \left[ \Gamma \right] \to \left( \left[ \sigma \right] \Rightarrow \left[ \tau \right] \right) \times \left[ \sigma \right] \to \left[ \tau \right]
\]
Modelling Composition

Let $\Gamma' \vdash N : \tau$ be a proved term where $\Gamma' = [x_1 : \sigma_1, x_2 : \sigma_2]$ and let $\Gamma \vdash M_i : \sigma_i$ be proved terms for $i = 1, 2$. Then one can show that $\Gamma \vdash N[M_1, M_2/x_1, x_2] : \tau$ and

$$[[\Gamma \vdash N[M_1, M_2/x_1, x_2] : \tau]] = [[\Gamma' \vdash N : \tau]] \circ \langle[[\Gamma \vdash M_1 : \sigma_1]], [[\Gamma \vdash M_2 : \sigma_2]]\rangle$$

**Proof:** By rule induction on the derivation of the judgement $\Gamma' \vdash N : \tau$. 
Soundness

Let $M$ be a structure for a $\lambda \times +$-signature in a bicartesian closed category $C$. $M$ satisfies the equation-in-context $\Gamma \vdash M = M'$ if $[\Gamma \vdash M : \sigma]$ and $[\Gamma \vdash M' : \sigma]$ are equal. We say that $M$ is a model of a $\lambda \times -$-theory $Th = (Sg, Ax)$ if $M$ satisfies the axioms.

Then $M$ satisfies any equation-in-context which is a theorem of $Th$.

Proof: This can be shown by rule induction using the rules for deriving theorems.
Let
\[ m \overset{\text{def}}{=} \left[ \Gamma, x : \sigma \vdash M : \tau \right] : [\Gamma] \times [\sigma] \to [\tau] \]
and \( a \overset{\text{def}}{=} [\Gamma \vdash A : \sigma] : [\Gamma] \to [\sigma] \). Then we have

(Property Closure for the (base) rule):

\[ \text{Sg} \triangleright \Gamma, x : \sigma \vdash M : \tau \quad \text{Sg} \triangleright \Gamma \vdash A : \sigma \]

\[ \Gamma \vdash (\lambda x : \sigma.M)A = M[A/x] : \tau \]

\[ \left[ \Gamma \vdash (\lambda x : \sigma.M)A : \tau \right] = ev \circ \langle [\Gamma \vdash \lambda x : \sigma.M : \tau], [\Gamma \vdash A : \sigma] \rangle \]

\[ = ev \circ \langle \lambda(m), a \rangle \]

\[ = ev \circ (\lambda(m) \times id) \circ \langle id, a \rangle \]

\[ = m \circ \langle id, a \rangle \]

\[ = [\Gamma \vdash M[A/x] : \tau] \]
Transporting Models

Suppose that we are given a morphism of bicartesian closed categories $F : C \to D$. Let $M$ be a model of $Th$ in $C$. We shall show how to define a new model, of $Th$ in $D$, denoted by $F_*M$. We shall need a lemma, that may be proved by induction over types:

If we set $\llbracket \gamma \rrbracket_{F_*M} \overset{\text{def}}{=} F \llbracket \gamma \rrbracket_M$ where $\gamma$ is a ground type of $Th$, then it follows from this that there is a canonical isomorphism $\llbracket \sigma \rrbracket_{F_*M} \cong F \llbracket \sigma \rrbracket_M$ where $\sigma$ is any type of $Th$. 
A structure $F_*M$ is given by $[[\gamma]]_{F_*M} \overset{\text{def}}{=} F[[\gamma]]_M$ on ground types and $[[f]]_{F_*M}$ is given by the composition

$$\left[[\sigma_1]\right]_{F_*M} \times \cdots \times \left[[\sigma_n]\right]_{F_*M} \cong F\left[[\sigma_1]\right]_M \times \cdots \times F\left[[\sigma_n]\right]_M \cong'$$

$$F\left(\left[[\sigma_1]\right]_M \times \cdots \times \left[[\sigma_n]\right]_M\right) \xrightarrow{F[[f]]_M} F\left[[\tau]\right]_M \cong \left[[\tau]\right]_{F_*M}$$

where $f : \sigma_1, \ldots, \sigma_n \rightarrow \tau$ is a function symbol of $Th$, the isomorphisms $\cong$ exist because of the lemma, and $\cong'$ arises from $F$ preserving finite products.
In fact $F_*M$ is a model of $Th$.

Given a proved term $\Gamma \vdash M : \sigma$ one can show by induction that
the morphism $[[\Gamma \vdash M : \sigma]]_{F_*M}$ is given by the composition

$$[[\sigma_1]_{F_*M} \times \ldots \times [\sigma_n]_{F_*M}] \cong F([\sigma_1]_M \times \ldots \times [\sigma_n]_M)^{F[[\Gamma \vdash M : \sigma]]_M}_F[[\sigma]]_M.$$

If we are given proved terms $\Gamma \vdash M : \sigma$ and $\Gamma \vdash N : \sigma$ for which
$[[\Gamma \vdash M : \sigma]]_M = [[\Gamma \vdash N : \sigma]]_M$ then certainly
$[[\Gamma \vdash M : \sigma]]_{F_*M} = [[\Gamma \vdash N : \sigma]]_{F_*M}$. Thus if $M$ is a model of $Th$ in $C$
then $F_*M$ is a model of $Th$ in $D$. 
Classifying Categories

Let $Th$ be a $\lambda \times +$-theory. A bicartesian closed category $Cl(Th)$ is called the\textit{ classifying} category of $Th$ if there is a model $G$ of $Th$ in $Cl(Th)$ for which given any category $D$ with finite products, and a model $M$ of $Th$ in $D$, then there is a functor $\mu : Cl(Th) \rightarrow D$ such that

\[ \begin{array}{ccc}
Th & \xrightarrow{M} & D \\
\downarrow & & \downarrow \\
Cl(Th) & \xrightarrow{\mu} & G
\end{array} \]

where $\mu_*G = M$. 
Constructing Classifiers

Every $\lambda \times +$-theory $Th$ has a classifying category $Cl(Th)$. We can construct a canonical classifying category using the syntax of $Th$.

Proof:

- The objects of $Cl(Th)$ are the types of $Th$.

- A morphism $\sigma \rightarrow \tau$ is an equivalence class $(x : \sigma | M)$ of pairs $(x : \sigma, M)$ where $Sg \triangleright x : \sigma \vdash M : \tau$, with equivalence relation

\[
(x : \sigma, M) \sim (x' : \sigma, M') \quad \text{iff} \quad Th \triangleright x : \sigma \vdash M = M'[x/x'] : \tau.
\]
Given \( \sigma \) and \( \tau \), the binary product is \( \sigma \times \tau \) with projection \( \pi_\sigma : \sigma \times \tau \to \sigma \) given by \( (z : \sigma \times \tau \mid \text{Fst}(z)) \). If \( (x : \gamma \mid M) : \gamma \to \sigma \) and \( (y : \gamma \mid N) : \gamma \to \tau \), then the mediating morphism is

\[
(z : \gamma \mid \langle M[z/x], N[z/y] \rangle) : \gamma \to \sigma \times \tau.
\]

\( (x : \sigma \mid \langle \rangle) \) is the unique morphism \( \sigma \to \text{unit} \) so that \( \text{unit} \) is a terminal object for \( \text{Cl}(\text{Th}) \).

\( (x : \text{null} \mid \text{Emp}_\sigma(x)) \) is the unique morphism \( \text{null} \to \sigma \) so that \( \text{null} \) is a terminal object for \( \text{Cl}(\text{Th}) \).
We define a structure $G$ for $Sg$ in $Cl(Th)$. $\llbracket \gamma \rrbracket_G \overset{\text{def}}{=} \gamma$ (and hence it follows that $\llbracket \sigma \rrbracket_G = \sigma$ for any type $\sigma$).

Also define for $f : \sigma_1, \sigma_2 \rightarrow \tau$

$$\llbracket f \rrbracket_G \overset{\text{def}}{=} (z : \sigma_1 \times \sigma_2 | f(Fst(z), Snd(z)))$$

Certainly we have

$$Sg \triangleright z : \sigma_1 \times \sigma_2 \vdash f(Fst(z), Snd(z)) : \tau$$

If $k : \sigma$ then $\llbracket k \rrbracket_G \overset{\text{def}}{=} (x : \text{unit} | k)$. 
We check that $G$ is indeed a model of $Th = (Sg, Ax)$. Suppose that $Sg \vdash x : \sigma, y : \tau \vdash M : \rho$. Then we can prove by induction that

$$[[x : \sigma, y : \tau \vdash M : \rho]]_G = (z : \sigma \times \tau \mid M[Fst(z)/x, Snd(z)/y])$$

Now, if we have $Th \vdash x : \sigma, y : \tau \vdash M = M' : \rho$, then

$$Th \vdash z : \sigma \times \tau \vdash M[Fst(z)/x, Snd(z)/y] = M'[Fst(z)/x, Snd(z)/y] : \rho$$

and hence that $[[x : \sigma, y : \tau \vdash M : \rho]]_G = [[x : \sigma, y : \tau \vdash M' : \rho]]_G$. 
Now let $M$ be a model of $Th$ in $D$. We define $\mu : Cl(Th) \to D$ by

$$
(x : \sigma \mid M) : \sigma \to \tau \quad \mapsto \quad [x : \sigma \vdash M : \tau]_M : [\sigma]_M \to [\tau]_M
$$

The soundness theorem says that the definition makes sense. It is easy to see that $\mu$ is a bicartesian closed functor.
It is routine to verify that $\mu_* G = M$. For example, consider a function symbol $f : \sigma_1, \sigma_2 \rightarrow \tau$. Then

$$\llbracket f \rrbracket_{\mu_* G} = \mu(z : \sigma_1 \times \sigma_2 \mid f(\text{Proj}_1(z), \text{Proj}_2(z)))$$

$$= \llbracket z : \sigma_1 \times \sigma_2 \vdash f(\text{Proj}_1(z), \text{Proj}_2(z)) : \tau \rrbracket_M$$

$$= \llbracket f \rrbracket_M \circ \langle \pi, \pi' \rangle$$

$$= \llbracket f \rrbracket_M.$$
Suppose that there is another bicartesian closed functor $\mu' : Cl(Th) \to \mathcal{D}$ for which $\mu'_* G = M$. If $\sigma$ is an object of $Cl(Th)$ then

$$\mu \sigma \overset{\text{def}}{=} \left[ \sigma \right]_M = \left[ \sigma \right]_{\mu'_* G} \cong \mu' \left[ \sigma \right]_G = \mu' \sigma$$

using a previous lemma that establishes the isomorphism, and this gives rise to a natural isomorphism $\mu \cong \mu'$. 
Some Applications

We show that by starting with a very simple type theory, the expressive power (in a sense to be made precise) is not increased by adding products, sums and functions. This is proved by establishing an equivalent categorical problem, and solving it using categorical methods.
An algebraic theory is a $\lambda \times +$-theory in which there are no product, sum, and function types. More precisely, an algebraic theory $Th = (Sg, Ax)$ consists of

- a collection of types and function symbols;
- raw terms generated from these data, using only the rules

$$
\begin{array}{c}
\vdash \\
x & k & M_1 & \ldots & M_a \\
\hline
\end{array} \\
f(M_1, \ldots, M_a)
$$

- proved terms, generated as expected; and
- theorems, generated by the rules of equality.
Classifiers for Algebraic Theories

Every algebraic theory $Th$ has a classifying theory $Cl(Th)$.

- The objects of $Cl(Th)$ are finite lists of types from the algebraic signature $Sg$ of $Th$, for example $\vec{\sigma} \overset{\text{def}}{=} [\sigma_1, \ldots, \sigma_n]$.

- The morphisms with source $\vec{\sigma}$ and target $\vec{\tau}$, where $\tau \overset{\text{def}}{=} [\tau_1, \ldots, \tau_m]$ and both $\vec{\sigma}$ and $\vec{\tau}$ are non-empty lists, are given by finite lists of the form

\[
[(\Gamma \mid M_1), \ldots, (\Gamma \mid M_m)] : \vec{\sigma} \to \vec{\tau}
\]

where the types $\vec{\sigma}$ appear in $\Gamma$ and we have $Sg \vdash \Gamma \vdash M_j : \tau_j$ for $1 \leq j \leq m$. 
A Conservative Extension

Let $Th = (Sg, Ax)$ be an algebraic theory. Let $Th' = (Sg', Ax')$ be the $\lambda \times +$-theory with ground types and function symbols those of $Sg$, and $Ax' \overset{\text{def}}{=} Ax$. Let $\Gamma \overset{\text{def}}{=} [x_1 : \gamma_1, \ldots, x_n : \gamma_n]$. Suppose that

$$Sg' \triangleright [x_1 : \gamma_1, \ldots, x_n : \gamma_n] \vdash E : \gamma$$

Then there exists $M$ for which

$$Sg \triangleright \Gamma \vdash M : \gamma \quad \text{and} \quad Th' \triangleright \Gamma \vdash E = M : \gamma.$$

Moreover, if there is $M'$ for which $Sg \triangleright \Gamma \vdash M' : \gamma$ and also $Th' \triangleright \Gamma \vdash E = M' : \gamma$ then we have $Th \triangleright \Gamma \vdash M = M' : \gamma$. 
Free Bicartesian Closed Categories

Let $\mathcal{C}$ be a category with finite products. Then $\mathcal{F}\mathcal{C}$ is the relatively free BCCC generated by $\mathcal{C}$ if there is a finite product preserving functor $I: \mathcal{C} \to \mathcal{F}\mathcal{C}$ such that if $F: \mathcal{C} \to \mathcal{D}$ is finite product preserving and $\mathcal{D}$ is a BCCC then there is a BCCC functor $\overline{F}: \mathcal{F}\mathcal{C} \to \mathcal{D}$ for which $\phi: \overline{FI} \cong F$ and $\overline{F}$ is unique up to isomorphism.
Relating $Th$ and $Th'$ Categorically

We define a functor $I : Cl(Th) \to Cl(Th')$. Very roughly, if

$$(x : \gamma \mid M)_Th : \gamma \to \gamma'$$

then we set

$I(x : \gamma \mid M)_Th \overset{\text{def}}{=} (x : \gamma \mid M)_{Th'}$

Warning: the objects of $Cl(Th)$ are in fact lists of types (in the example above the source $\gamma$ and target $\gamma'$ are lists of length one) and the precise definition of $I$ is rather messy . . . .
On an object $\vec{\gamma}$ of $Cl(Th)$ set

$$I(\vec{\gamma}) \overset{\text{def}}{=} (\ldots (\gamma_1 \times \gamma_2) \times \ldots) \times \gamma_n$$

and given a morphism $(\Gamma | \vec{M})_{Th} : \vec{\gamma} \to \vec{\gamma}'$ (where the subscript $Th$ denotes equivalence up to provable equality in $Th$), then we set

$$I(\Gamma | \vec{M})_{Th} \overset{\text{def}}{=} (z : \Pi \gamma_i | \langle \ldots \langle \widehat{M}_1, \widehat{M}_2 \rangle, \ldots, \widehat{M}_m \rangle)_Th'$$

in which we have written $\Pi \gamma_i$ for $(\ldots (\gamma_1 \times \gamma_2) \times \ldots) \times \gamma_n$ and also

$$\widehat{M}_j \overset{\text{def}}{=} M_j[\text{Proj}_1(z)/x_1, \ldots, \text{Proj}_j(z)/x_j, \ldots, \text{Proj}_n(z)/x_n]$$

where $\text{Proj}_j(z)$ is a term for $j$-th projection.
Full and Faithful Functors

- $F : C \to D$ is **faithful** if given a parallel pair of morphisms $f, g : A \to B$ in $C$ for which $Ff = Fg$, then $f = g$. Thus

  $C(A, B) \to D(FA, FB)$

  is 1-1.

- $F$ is **full** if given objects $A$ and $B$ in $C$ and a morphism $g : FA \to FB$ in $D$, then there is some $f : A \to B$ in $C$ for which $Ff = g$. Thus

  $C(A, B) \to D(FA, FB)$

  is onto.
Outlining a Proof of the Con. Extension

1. Show that $I : Cl(Th) \rightarrow Cl(Th')$ yields a free BCCC.

2. Prove a purely categorical result called the “logical relations” gluing lemma.

3. Apply the gluing lemma and the free BCCC property, to show that $I$ is full and faithful . . .

$$Cl(Th)(\gamma, \gamma') \overset{\cong}{\longrightarrow} Cl(Th')(I\gamma, I\gamma')$$
**Existence:** Suppose that $Sg' \triangleright x : \gamma \vdash E : \gamma'$. Then we certainly have

$$e \overset{\text{def}}{=} (x : \gamma \mid E)_{Th'} : I\gamma \to I\gamma'$$

in $Cl(Th')$. Using the fullness of $I$, there is a morphism $(x : \gamma \mid M)_{Th} : \gamma \to \gamma'$ which is taken to $e$ by $I$. But this implies

$$Th' \triangleright x : \gamma \vdash M = E : \gamma'$$

as required.
A Free BCCC

The functor $I : Cl(Th) \rightarrow Cl(Th')$ presents $Cl(Th')$ as the relatively free BCC generated by $Cl(Th)$.

**Proof:** Let $F : Cl(Th) \rightarrow C$ preserve finite products where $C$ is a BCCC. We shall define a functor $\overline{F} : Cl(Th') \rightarrow C$ by recursion over the syntactic structure of $Cl(Th')$. For example

- $\overline{F}\gamma \overset{\text{def}}{=} F[\gamma]$ where $\gamma$ is a ground type of $Sg'$,
- $\overline{F}(\sigma \times \tau) \overset{\text{def}}{=} \overline{F}\sigma \times \overline{F}\tau$,
- $\overline{F}(z : \delta | \langle \rangle) \overset{\text{def}}{=} ! : \overline{F}\delta \rightarrow 1_C$,
- $\overline{F}(z : \delta | \text{Fst}(P)) \overset{\text{def}}{=} \pi_1 \overline{F}(z : \delta | P)$ where $\pi_1 : \overline{F}\sigma \times \overline{F}\tau \rightarrow \overline{F}\sigma$, 
**Gluing Lemma by Logical Relations**

Let $\mathcal{D}$ be a BCC and let $I : \mathcal{C} \to \mathcal{D}$ preserve finite products. We define a category $\mathcal{Gl}$ as follows:

- **Objects of $\mathcal{Gl}$** are $(F, \sqsubset, D)$ where $F : \mathcal{C}^{\text{op}} \to \text{Set}$ is a functor, $D$ is an object of $\mathcal{D}$, and for each object $C$ of $\mathcal{C}$, $\sqsubset_C \subseteq FC \times \mathcal{D}(IC, D)$.

- **A morphism** $(\alpha, d) : (F, \sqsubset, D) \to (F', \sqsubset', D')$ is given by a natural transformation $\alpha : F \to F'$ and a morphism $d : D \to D'$ in $\mathcal{D}$ for which if $x \sqsubset_C u$ then $\alpha_C(x) \sqsubset'_C d \circ u$, where of course $x \in FC$ and $u \in \mathcal{D}(IC, D)$.

Then $\mathcal{Gl}$ is a bicartesian closed category and the obvious functor $\pi_2 : \mathcal{Gl} \to \mathcal{D}$ is a morphism of BCCCCs.
Proof The structure of $Gl$ is specified by a “logical relations” procedure on the subset $\triangleleft_C$.

*(Binary Products)*: We set

$$(F, \triangleleft, D) \times (F', \triangleleft', D') \overset{\text{def}}{=} (F \times F', \triangleleft \times \triangleleft', D \times D')$$

where $(x, x')(\triangleleft \times \triangleleft')_C u$ just in case $x \triangleleft_C \pi u$ and $x' \triangleleft'_C \pi' u$ where of course $\pi : D \times D' \to D$ and $\pi' : D \times D' \to D'$ in $D$. The projections in $Gl$ are given by pairing of projections in $[C^{op}, Set]$ and $D$, such as:

$$\pi_{(F, \triangleleft, D)} \overset{\text{def}}{=} (\pi_F, \pi_D) : (F \times F', \triangleleft \times \triangleleft', D \times D') \to (F, \triangleleft, D).$$
Freeness Implies Full and Faithful

Let $\mathcal{C}$ be a locally small category, and $\mathcal{FC}$ the freely generated bicartesian closed category. Then the canonical functor $I: \mathcal{C} \to \mathcal{FC}$ is full and faithful.

**Proof** We apply the gluing lemma to $I$ (so $\mathcal{D} \overset{\text{def}}{=} \mathcal{FC}$). We define a functor $J: \mathcal{C} \to \mathcal{Gl}$: on objects $C$ of $\mathcal{C}$ define $JC$ by $(\mathcal{C}(\_, C), \triangleleft^C, IC)$ where the subset

$$\triangleleft^C_{C'} \subseteq \mathcal{C}(C', C) \times \mathcal{FC}(IC', IC)$$

is defined by just requiring $c \triangleleft^C_{C'} Ic$ for each morphism $c: C' \to C$ in $\mathcal{C}$. On morphisms $c$ of $\mathcal{C}$ we set $Jc \overset{\text{def}}{=} (\mathcal{C}(\_, c), Ic)$. 
The Yoneda functor $\mathcal{C}(-,+) : \mathcal{C} \to [\mathcal{C}^{op}, \text{Set}]$ is full and faithful, where $c : \mathcal{C} \to \mathcal{C}' \mapsto \mathcal{C}(-,c) : \mathcal{C}(-,\mathcal{C}) \to \mathcal{C}(-,\mathcal{C}')$.

$J$ is faithful for $\mathcal{C}(-,+)$ is faithful. For fullness, let $(\alpha, d) : JC \to JC'$. Hence $\alpha : \mathcal{C}(-,\mathcal{C}) \to \mathcal{C}(-,\mathcal{C}')$ and so $\alpha = \mathcal{C}(-,c)$ for some $c : \mathcal{C} \to \mathcal{C}'$ in $\mathcal{C}$. Now certainly $id_{\mathcal{C}} \sqcup_{\mathcal{C}} id_{IC}$ and so

$$\alpha_{\mathcal{C}}(id_{\mathcal{C}}) = \mathcal{C}(\mathcal{C},c)(id_{\mathcal{C}}) = c \sqcup_{\mathcal{C}'} d \circ id_{IC} = d$$

implying $d = Id_{\mathcal{C}}$; therefore $Jc = (\alpha, d)$, that is $J$ is full.
Consider the following diagram

\[
\begin{array}{ccc}
\text{Cl}(Th') & \xrightarrow{=} & \text{Cl}(Th') \\
\downarrow & & \downarrow \\
\text{Gl}(\Gamma) & \xrightarrow{\ \ J \ \} & \text{id}_{\text{Cl}(Th')} \\
\downarrow & \downarrow & \\
\text{Cl}(Th) & \xrightarrow{\ P_2 \ } & \text{Cl}(Th') \\
\end{array}
\]
By freeness, the functor $\overline{J}$ exists and $\overline{J} \circ I \cong J$ naturally. By definition, $P_2 \circ J = I$. It follows that $P_2 \circ \overline{J} \circ I \cong I$ naturally, that is $(P_2 \circ \overline{J}) \circ I \cong I$, and as $id_{cl(Th')} \circ I \cong I$ (trivially!) it follows from the universal property of relatively free bicartesian closed categories that $id_{cl(Th')} \cong P_2 \circ \overline{J}$ naturally. This latter isomorphism implies that $\overline{J}$ is faithful. This fact, together with $J$ full and faithful proved above, and $\overline{J} \circ I \cong J$ implies that $I$ is full and faithful.