General Aims

- To teach basics of *category theory*.
- To study *programming language syntax with binding*.

- We only cover the category theory we need.
- Some categorical machinery is simplified – you read the abstract stuff *after* these lectures.
- We study syntax by examples – we omit the general theory of binding syntax.
- Syntax with binding is a hot research topic . . .
Basics of Algebraic and Binding Syntax
See OHP for Examples

- Algebraic syntax specified by constructor symbols $C_i$.
- Each symbol has an arity $a \in \mathbb{N}$.
- These generate (finite) expressions such as
  $$C_3 \ e_0 \ \ldots \ \ e_{a-1}$$

- \ldots from datatypes of the form
  $$\text{datatype } Exp = \ldots \ C_3 \overset{\text{length } a}{\underbrace{Exp \ \ldots \ Exp}} \ldots$$
- Binding syntax subsumes algebraic syntax.

- Binding syntax is specified by giving some constructor symbols $C_i$ where each symbol has an arity $a \in \mathbb{N}$ and a binding depth $(b_0, \ldots, b_{a-1}) \in \mathbb{N}^a$.

- These generate (finite) expressions such as
  \[
  C_3 \ldots (v^0, \ldots, v^{b_j-1}, e_j) \ldots
  \]
  \[
  \text{length } a
  \]

- \ldots from datatypes of the form
  \[
  \text{datatype } \text{Exp} = \ldots C_3 (\text{V}, \ldots, \text{V}, \text{Exp}) \ldots (\text{V}, \ldots, \text{V}, \text{Exp}) \ldots
  \]
  \[
  \text{length } b \quad \text{length } b \quad \text{length } a
  \]
Learning Outcomes: You Should

- know how examples of programming language syntax with binding can be specified inductively;
- be able to define basic categorical structures;
- know, by example, how to compute simple initial algebras;
- understand simple *abstract* models of syntax and know how to *manufacture* categorical models *from* syntax;
- be able to prove these models are essentially the same;
- understand current issues concerning variable binding and read the literature.
A category \( C \) is specified by:

- A collection \( ob \ C \) of objects; \( A, B, C \ldots \)
- A collection \( mor \ C \) of morphisms; \( f, g, h \ldots \)
- For each \( f \) a source \( src(f) \) in \( ob \ C \) and a target \( tar(f) \) in \( ob \ C \). Write

\[
f: src(f) \longrightarrow tar(f) \quad \text{or} \quad f:A \rightarrow B
\]
- $f$ and $g$ composable if $\text{tar}(f) = \text{src}(g)$.

- If $f:A \to B$ and $g:B \to C$ then there is $g \circ f:A \to C$, called the composition.

- For any object $A$ there is an identity morphism $id_A:A \to A$.

  For any $f$

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

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

- $\circ$ is associative: given $f:A \to B$, $g:B \to C$ and $h:C \to D$,

  $$(h \circ g) \circ f = h \circ (g \circ f)$$
Examples of Categories

- Consider $Exp ::= V V | S Exp | A Exp Exp$ with typical elements

\[
V v^0 \quad V v^{45} \quad A (S (V v^3)) (V v^2)
\]

- There is a category with typical morphisms

\[
6 \xrightarrow{[V v^4, V v^2, V v^1, S (V v^5) \ ]} 4
\]

\[
2 \xrightarrow{[A (A v^0 v^0) v^1, A v^1 v^0, A v^0 (S v^0) \ ]} 3
\]
If
\[
\begin{align*}
1 \quad [S \nu^0, A \nu^0 \nu^0] & \rightarrow 2 \quad [A (A \nu^0 \nu^1) \nu^1, A \nu^1 \nu^0, A \nu^0 (S \nu^1)] \\
& \rightarrow 3
\end{align*}
\]
the composition is
\[
[A (A (S \nu^0) (A \nu^0 \nu^0)) (A \nu^0 \nu^0), \\
A (A \nu^0 \nu^0) (S \nu^0), \\
A (S \nu^0) (S (A \nu^0 \nu^0))]
\]
\textbf{Set}

- The objects are sets.
- Morphisms are triples \((A, f, B)\) where \(f \subseteq A \times B\) is a \textit{graph} of a function:

\[
(\forall a \in A)(\exists! b \in B)((a, b) \in f)
\]

- Composition is given by

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

- \(id_A\) is \((A, id, A)\).
\((X, \leq)\)

- \((X, \leq)\) is a preordered set: \(\leq\) is reflexive and transitive.
- The collection of objects is the set \(X\).
- The collection of morphisms is the set \(\leq\). Typical morphism \((x, x')\).
- Composition is given by \((y, z) \circ (x, y) \overset{\text{def}}{=} (x, z)\).
- \(id_x \overset{\text{def}}{=} (x, x)\).
**Preset**

- The objects are the preordered sets.
- The morphisms are the **monotone** functions.

A morphism \((X, \leq_X) \rightarrow (Y, \leq_Y)\) is specified by a function \(f: X \rightarrow Y\) such that

\[
x \leq_X x' \implies f(x) \leq_Y f(x')
\]
The set of objects of $\mathbb{F}$ is $\mathbb{N}$.

- We regard $n \in \mathbb{N}$ as the set $\{0, \ldots, n-1\}$ for $n \geq 1$, and $0$ is the empty set $\emptyset$.

- A morphism $\rho: n \rightarrow n'$ is any set-theoretic function.
Isomorphisms and Equivalences

- A morphism \( f : A \to B \) is an isomorphism if there is some \( g : B \to A \) for which
  \[
  f \circ g = \text{id}_B \quad \land \quad g \circ f = \text{id}_A
  \]

- We say \( g \) is an inverse for \( f \) and vise versa.

- We say \( A \) is isomorphic to \( B \),
  \[
  f : A \cong B : g
  \]
  if such a mutually inverse pair of morphisms exists.

- \( f \) and \( g \) witness the isomorphism.
Examples of Isomorphisms

- Bijective functions in Set are isomorphisms.
- In \((X, \leq)\):
  - If \(\leq\) is a partial order, the only isomorphisms are the identities, *or*
  - If \(\leq\) is a preorder and \(x, y \in X\) we have \(x \cong y\) iff \(x \leq y\) and \(y \leq x\), with only one witness:

\[
(x, y) : x \cong y : (y, x)
\]
Definition of a Functor

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

- assigning an object $FA$ in $\mathcal{D}$ to any object $A$ in $\mathcal{C}$, and
- assigning a morphism $Ff : FA \to FB$ in $\mathcal{D}$, to any morphism $f : A \to B$ in $\mathcal{C}$,

for which

- $F(id_A) = id_{FA}$
- $F(g \circ f) = Fg \circ Ff$
An Example of a Functor

Define $F : set \rightarrow set$ by

- $FA \overset{\text{def}}{=} [A]$, the finite lists over $A$
- $Ff \overset{\text{def}}{=} \text{map}(f)$ where

$\text{map}(f) : [A] \rightarrow [B]$ is defined by

\[\text{map}(f)(as) \overset{\text{def}}{=} \text{case } as \text{ of }\]
\[
\begin{array}{c}
\varepsilon \rightarrow \varepsilon \\
[a_0, \ldots, a_{l-1}] \rightarrow [f(a_0), \ldots, f(a_{l-1})]
\end{array}
\]
To see that $F(g \circ f) = Fg \circ Ff$ note that

$$F(g \circ f)([a_0, \ldots, a_{l-1}]) \overset{\text{def}}{=} map(g \circ f)([a_0, \ldots, a_{l-1}])$$

$$= [g(f(a_0)), \ldots, g(f(a_{l-1}))]$$

$$= map(g)([f(a_0), \ldots, f(a_{l-1})])$$

$$= map(g)(map(f)([a_0, \ldots, a_{l-1}])))$$

$$= Fg \circ Ff([a_0, \ldots, a_{l-1}]).$$
More Examples

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

- We can define a functor $\mathcal{P}: \text{Set} \rightarrow \text{Set}$ by setting

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

where the function $\mathcal{P}f$ is defined by

$$\mathcal{P} f(A') \overset{\text{def}}{=} \{ f(a) \in B \mid a \in A' \}$$

where $A' \in \mathcal{P}(A)$. 
Definition of a Natural Transformation

Let \( F, G : C \to D \) be functors. Then a natural transformation \( \alpha : F \to G \) is

\[
\alpha : F \to G \quad \text{is} \quad \left( \alpha_A : FA \to GA \mid A \text{ in } \text{ob } C \right)
\]

such that for any \( f : A \to B \) in \( C \),

\[
\begin{array}{ccc}
FA & \xrightarrow{\alpha_A} & GA \\
Ff \downarrow & & \downarrow Gf \\
FB & \xleftarrow{\alpha_B} & GB
\end{array}
\]
An Example of a Natural Transformation

- Recall $F : \text{Set} \rightarrow \text{Set}$ where $FA \overset{\text{def}}{=} [A]$ and $Ff \overset{\text{def}}{=} \text{map}(f)$.

- There is a natural transformation $\text{rev}: F \rightarrow F$ with components $\text{rev}_A : [A] \rightarrow [A]$ defined by

$$\text{rev}_A(as) \overset{\text{def}}{=} \begin{cases} \varepsilon \rightarrow \varepsilon \\ [a_0, \ldots, a_{l-1}] \rightarrow [a_{l-1}, \ldots, a_0] \end{cases}$$

- Naturality is

$$Ff \circ \text{rev}_A([a_0, \ldots, a_{l-1}]) = [f(a_{l-1}), \ldots, f(a_0)]$$

$$= \text{rev}_B \circ Ff([a_0, \ldots, a_{l-1}])$$
Another Example

- Define $F_X: \text{set} \rightarrow \text{set}$ by

  - $F_X(A) \overset{\text{def}}{=} (X \rightarrow A) \times X$
  
  - $F_X(f): (X \rightarrow A) \times X \longrightarrow (X \rightarrow B) \times X$ where
    
    $(g, x) \mapsto (f \circ g, x)$

- Then $ev: F_X \rightarrow \text{id}_{\text{set}}$ defined by $ev_A(g, x) \overset{\text{def}}{=} g(x)$ is natural

\[
(id_{\text{set}}(f) \circ ev_A)(g, x) = f(g(x)) = ev_B(f \circ g, x) = ev_B(F_X(f)(g, x)) = (ev_B \circ F_X(f))(g, x).
\]
Definition of Functor Category

- Let $F$, $G$, $H$ be functors $\mathcal{C} \to \mathcal{D}$ and $\alpha: F \to G$ and $\beta: G \to H$ be natural transformations.

- Define $\beta \circ \alpha: F \to H$ by

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

- Then $\mathcal{D}^\mathcal{C}$ is the functor category of $\mathcal{C}$ and $\mathcal{D}$, where
  - objects are functors $\mathcal{C} \to \mathcal{D}$,
  - morphisms are natural trans $\alpha: F \to G: \mathcal{C} \to \mathcal{D}$
An isomorphism in a functor category is referred to as a **natural isomorphism**.

If there is a natural isomorphism between the functors $F$ and $G$, then we say that $F$ and $G$ are **naturally isomorphic**, written

$$\phi: F \cong G: \psi$$

with witnesses the natural transformations $\phi$ and $\psi$. 
Motivating Binary Products

(Property $\Phi(P)$)

- Given any two sets $A$ and $B$,
- there are functions $\pi: P \rightarrow A$, $\pi': P \rightarrow B$ such that:

Given any $f: C \rightarrow A$, $g: C \rightarrow B$ there is a unique $h: C \rightarrow P$ s.t.

\[
\begin{array}{c}
C \\
\downarrow f \quad \exists! h \\
A & \xleftarrow{\pi} & P & \xrightarrow{\pi'} & B \\
\end{array}
\]
Suppose that $A \overset{\text{def}}{=} \{a, b\}$ and $B \overset{\text{def}}{=} \{c, d, e\}$.

- Let $P$ be $A \times B \overset{\text{def}}{=} \{(x, y) \mid x \in A, y \in B\}$ and
- $\pi$ and $\pi'$ be coordinate projections.

Let $f: C \to A$ and $g: C \to B$ be any two functions. Define

$$h: C \to P \quad z \mapsto (f(z), g(z))$$

We can check ($\text{Property } \Phi(P)$) …
Now define $P' \overset{\text{def}}{=} \{1, 2, 3, 4, 5, 6\}$ and

$p: P' \to A$ and $q: P' \to B$ where

\[
\begin{align*}
p(1), & \quad p(2), & \quad p(3) = a & \quad q(1), & \quad q(4) = c \\
p(4), & \quad p(5), & \quad p(6) = b & \quad q(2), & \quad q(5) = d \\
\end{align*}
\]

We can check ($\text{Property } \Phi(P')$) \ldots

\ldots the required function $h: C \to P'$ exists and is unique: for example, $x \in C$ and $f(x) = a$ and $g(x) = d$ forces $h(x) = 2$

Note $P' \cong \{(a, c), (a, d), (a, e), (b, c), (b, d), (b, e)\} = P$
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 $\pi_B \circ \langle f, g \rangle = g$. 

Diagrams are helpful

The unique morphism $\langle f, g \rangle : C \to A \times B$ is called the mediating morphism.
• A property involving existence of a unique morphism leading to a structure determined up to isomorphism is a universal property.

• Call \( \langle f, g \rangle \) the pair of \( f \) and \( g \).

• \( \mathcal{C} \) has binary products if there is \( A \times B \) for any \( A \) and \( B \).

• \( \mathcal{C} \) has specified binary products if there is a canonical choice.

• In \( \text{Set} \) take \( A \times B \overset{\text{def}}{=} \{ (a, b) \mid a \in A, b \in B \} \) with standard projections.
Examples of Binary Products

\textbf{Preset} Given $A \overset{\text{def}}{=} (X, \leq_X)$ and $B \overset{\text{def}}{=} (Y, \leq_Y)$,

$$A \times B \overset{\text{def}}{=} (X \times Y, \leq_{X \times Y})$$

where $X \times Y$ is cartesian product, and

$$(x, y) \leq_{X \times Y} (x', y') \iff x \leq_X x' \land y \leq_Y y'$$

The projection

$$\pi_A : (X \times Y, \leq_{X \times Y}) \rightarrow (X, \leq_X)$$

is given by $(x, y) \mapsto x$, and is monotone
Part Given $A$ and $B$,

$$P \overset{\text{def}}{=} (A \times B) \cup (A \times \{*_A\}) \cup (B \times \{*_B\})$$

- $\pi_A : (A \times B) \cup (A \times \{*_A\}) \cup (B \times \{*_B\}) \longrightarrow A$
  is undefined on $B \times \{*_B\}$, $\pi_B$ on $A \times \{*_A\}$

- $\pi_A(a, *_A) = a$ for all $a \in A$, …

The product of $n$ and $m$ is written $n \times m$ and is given by $n \ast m$, that is, the set $\{0, \ldots, (n \ast m) - 1\}$.
Additional Notation

- Can define $A \times B \times C$ and $\langle f, g, h \rangle$

- Take $f: A \to B$ and $f': A' \to B'$. We write

$$f \times f' \overset{\text{def}}{=} \langle f \circ \pi, f' \circ \pi' \rangle : A \times A' \to B \times B'$$

- Universal property means

$$id_A \times id_{A'} = id_{A \times A'} \quad \text{and} \quad (g \times g') \circ (f \times f') = g \circ f \times g' \circ f'$$

where $g: B \to C$ and $g': B' \to C'$.

- Write $A^2$ or $f^2$ for $A \times A$ and $f \times f$
Another Example – Presheaves on $\mathbb{F}$

$\mathcal{F} \overset{\text{def}}{=} \text{Set}^\mathbb{F}$ If $F$ and $F'$ are presheaves, $F \times F' : \mathbb{F} \to \text{Set}$ defined by

$$(F \times F')(n) \overset{\text{def}}{=} (Fn) \times (F'n)$$

for $n$ in $\mathbb{F}$ and if $\rho : n \to n'$

$$(F \times F')(\rho) \overset{\text{def}}{=} (F\rho) \times (F'\rho)$$

Also

$$\pi_F : F \times F' \to F \quad (\pi_F)_n \overset{\text{def}}{=} \pi_{Fn}$$
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

$$A \xrightarrow{\iota_A} A + B \xrightarrow{\iota_B} B$$

$$\xrightarrow{[f, g]}$$

$$C$$

for all such $f$ and $g$
Example of Binary Coproducts

Set For sets $A$ and $B$ define

$$A + B \overset{\text{def}}{=} (A \times \{1\}) \cup (B \times \{2\})$$

and

$$\iota_A : A \rightarrow A + B \quad a \mapsto (a, 1)$$

Given $f : A \rightarrow C$ and $g : B \rightarrow C$, then $[f, g] : A + B \rightarrow C$ is defined by

$$[f, g](\xi) \overset{\text{def}}{=} \text{case } \xi \text{ of }$$

$$\iota_A(\xi_A) = (\xi_A, 1) \mapsto f(\xi_A)$$

$$\iota_B(\xi_B) = (\xi_B, 2) \mapsto f(\xi_B)$$
**Additional Notation**

- Can define $A + B + C$ with the cotupling $[f, g, h]$

- Take morphisms $f: A \to B$ and $f': A' \to B'$. We write

  $$f + f' \overset{\text{def}}{=} [\iota_B \circ f, \iota_{B'} \circ f'] : A + A' \to B + B'$$

- Universality means

  $$id_A + id_{A'} = id_{A + A'} \quad \text{and} \quad (g + g') \circ (f + f') = g \circ f + g' \circ f'$$

  where $g: B \to C$ and $g': B' \to C'$.

- If $l: C \to D$ then $l \circ [f, g] = [l \circ f, l \circ g]$
More Examples

The coproduct of $n$ and $m$ is $n + m$ where we interpret $+$ as addition on $\mathbb{N}$.

If $F$ and $F'$ are presheaves then $F + F'$ is defined by

$$(F + F')\xi \overset{\text{def}}{=} (F\xi) + (F'\xi)$$

for any object or morphism $\xi$ in $F$, and

$$\iota_F : F + F' \to F \quad (\iota_F)_n \overset{\text{def}}{=} \iota_{Fn} : (Fn) + (F'n) \to Fn$$

Sometimes say $+$ is defined pointwize.
Definition of Algebras

Let $F: C \to C$. An algebra for the functor $F$ is a pair $(A, \sigma_A)$ where $\sigma_A: FA \to A$.

An initial $F$-algebra $(I, \sigma_I)$ is an algebra for which given any other $(A, \sigma_A)$,

\[
\begin{array}{c}
Ff & \xrightarrow{\sigma_I} & I \\
\downarrow Ff & & \downarrow \exists! f \\
FA & \xrightarrow{\sigma_A} & A
\end{array}
\]
Motivation for Initial Algebras

■ (Some) Datatypes are initial algebras

■ The datatype

\[ \text{Exp ::= V V | S Exp | A Exp Exp} \]

is modeled by an object \( E \) such that

\[ E \cong V + E + (E \times E) \]

■ We show how to solve \( \dagger \) in Set.

■ If \( \Sigma : \text{Set} \to \text{Set} \) is \( \Sigma \xi \overset{\text{def}}{=} V + \xi + (\xi \times \xi) \), then the solution we construct is an initial algebra \( (\sigma_E, E) \).
An Initial Algebra for $1 + (-): \text{Set} \longrightarrow \text{Set}$

- $1: \text{Set} \rightarrow \text{Set}$ is defined by
  \[ f:A \rightarrow B \quad \mapsto \quad id_{\{\ast\}}: \{\ast\} \rightarrow \{\ast\} \]

- $1 + (-)$ is defined by
  \[ f:A \rightarrow B \quad \mapsto \quad id_1 + f: 1 + A \rightarrow 1 + B \]

- The initial algebra is $\mathbb{N}$ up to isomorphism.
We set $S_0 \overset{\text{def}}{=} \emptyset$ and $S_{r+1} \overset{\text{def}}{=} 1 + S_r$.

Note there is an insertion $\imath_{S_r}: S_r \rightarrow S_{r+1}$.

Note also that $i_r: S_r \hookrightarrow S_{r+1}$ where $i_0 \overset{\text{def}}{=} \emptyset: S_0 \rightarrow S_1$, and $i_{r+1} \overset{\text{def}}{=} id_1 + i_r$.

We also write $i'_r: S_r \hookrightarrow T$ where $T \overset{\text{def}}{=} \bigcup_r S_r$.

$T$ is the object part of an initial algebra for $1 + (-)$. 
As $\sigma_T : 1 + T \to T$ then $\sigma_T$ must be a copair.

We set $\sigma_T \overset{\text{def}}{=} [k,k']$ where $k : 1 \to T$ and $k' : T \to T$.

Note that

$$1 \xrightarrow{\iota_1} 1 + \emptyset = S_1 \xrightarrow{i'_1} T$$

and we set $k \overset{\text{def}}{=} i'_1 \circ \iota_1$. 
Note that

\[ S_r \xrightarrow{\iota_{S_r}} 1 + S_r = S_{r+1} \xrightarrow{i'_{r+1}} T \]

and we set \( k'_r \overset{\text{def}}{=} i'_{r+1} \circ \iota_{S_r} \).

- In fact \( k'_{r+1} \circ i_r = k'_r \) by induction on \( r \).
- Hence can legitimately define \( k': T \to T \) by setting
  \[ k'(\xi) \overset{\text{def}}{=} k'_r(\xi) \text{ for any } r \text{ such that } \xi \in S_r. \]
We check initiality

\[
\begin{array}{c}
1 + T \xrightarrow{\sigma_T} T \\
id_1 + \overline{f} \quad \overline{f} \text{ needs defining} \\
1 + A \xrightarrow{f} A
\end{array}
\]

We define a family of functions \( \overline{f}_r : S_r \to A \)

\[
\overline{f}_0 \overset{\text{def}}{=} \emptyset : S_0 \to A \quad \land \quad \overline{f}_{r+1} \overset{\text{def}}{=} [f \circ \iota_1, f \circ \iota_A \circ \overline{f}_r]
\]

- In fact \( \overline{f}_{r+1} \circ i_r = \overline{f}_r \).
- Hence we can legitimately define \( \overline{f} : T \to A \) by \( \overline{f}(\xi) \overset{\text{def}}{=} \overline{f}_r(\xi) \) for any \( r \) where \( \xi \in S_r \).
To check that the diagram commutes, we have to prove that

$$
\overline{f} \circ [k, k'] = f \circ (id_1 + \overline{f})
$$

By the universal property of coproducts, this is equivalent to showing

$$
[\overline{f} \circ k, \overline{f} \circ k'] = [f \circ \iota_1, f \circ \iota_A \circ \overline{f}]
$$

which we can do by checking that the respective components are equal.

We give details for $\overline{f} \circ k' = f \circ \iota_A \circ \overline{f}$. 
\[ \overline{f} \circ k' = f \circ \iota_A \circ \overline{f} \]. Take any element \( \xi \in T \). Then we have

\[
\begin{align*}
\overline{f}(k'(\xi)) &= \overline{f}(\iota_{S_r}(\xi)) \\
&= \overline{f}_{r+1}(\iota_{S_r}(\xi)) \\
&= [f \circ \iota_1, f \circ \iota_A \circ \overline{f}_r](\iota_{S_r}(\xi)) \\
&= f(\iota_A(\overline{f}_r(\xi))) \\
&= f(\iota_A(\overline{f}(\xi)))
\end{align*}
\]

The first equality is by definition of \( k' \) and \( k'_r \); the second by definition of \( \overline{f} \); the third by definition of \( \overline{f}_{r+1} \).

You check that \( T \cong N \).
Some Results for Use in Modelling Syntax

Let $F$ and $F'$ be two presheaves in $\mathcal{F}$. Suppose for any $n$ in $\mathcal{F}$, $F'n \subseteq Fn$, and

\[
\begin{array}{c}
F'n \\ \leftarrow \downarrow F'\rho \\
F'n' \\
\end{array}
\quad \begin{array}{c}
\subseteq \\
\downarrow F\rho \\
\subseteq \\
\end{array}
\begin{array}{c}
Fn \\
\end{array}
\quad \begin{array}{c}
Fn' \\
\end{array}
\]

commutes for any $\rho: n \rightarrow n'$.

There is a natural transformation

\[i: F' \rightarrow F\]
We define $\delta: \mathcal{F} \to \mathcal{F}$

Suppose that $F$ is an object in $\mathcal{F}$. Then $\delta F$ is defined by

$$\rho: n \to n' \quad \mapsto \quad F(\rho + id_1): F(n + 1) \to F(n' + 1)$$

If $\alpha: F \to F'$ in $\mathcal{F}$, then the components of $\delta \alpha$ are given by

$$\delta \alpha)_n \overset{\text{def}}{=} \alpha_{n+1}$$
(\mathcal{S}_r \mid r \geq 0) is a family of presheaves in \mathcal{F}, with \ i_r : \mathcal{S}_r \hookrightarrow \mathcal{S}_{r+1}. Then there is a union presheaf \mathcal{T} in \mathcal{F}, such that \ i'_r : \mathcal{S}_r \hookrightarrow \mathcal{T}. We sometimes write \bigcup_r \mathcal{S}_r for \mathcal{T}.

Let \rho : n \to n'. Then

\[ Tn \stackrel{\text{def}}{=} \bigcup_r \mathcal{S}_r n \]

and \( T\rho : Tn \to Tn' \) is defined by

\[ (T\rho)(\xi) \stackrel{\text{def}}{=} (\mathcal{S}_r \rho)(\xi) \]

where \( \xi \in Tn \), and \( \xi \in \mathcal{S}_r(n) \) for some \( r \).
Let \((\phi_r : S_r \rightarrow A \mid r \geq 0)\) be natural transformations in \(\mathcal{F}\), the \(S_r\) as before, and such that \(\phi_{r+1} \circ i_r = \phi_r\). Then there is a unique natural transformation

\[
\phi : T \rightarrow A
\]

such that \(\phi \circ i'_r = \phi_r\).

The functions \(\phi_n : Tn \rightarrow An\) defined by

\[
\phi_n(\xi) \overset{\text{def}}{=} (\phi_r)_n(\xi) \quad \xi \in S_r n
\]

yield the required natural transformation.
Syntax with Distinguished Variables and without Binding

- The set of expressions $Exp$ is inductively defined by

$$Exp ::= V \lor V \mid S \ Exp \mid A \ Exp \ Exp$$

- $v^i$ occurs in $e$ is written $v^i \in e$.

- The set of (free) variables of any $e$ is denoted by $fv(e)$.

- We will want to consider expressions $e$ for which

$$fv(e) \subset \{v^0, \ldots, v^{n-1}\}$$

and we give an inductive definition of such expressions.
First we define inductively a set of judgements $\Gamma^n \vdash^{d\overline{b}} e$ where $n \geq 1$, $\Gamma^n \overset{\text{def}}{=} v^0, \ldots, v^{n-1}$ is a list, and of course $e$ is an expression.

We refer to $\Gamma^n$ as an environment of variables.

\[
\begin{align*}
0 \leq i < n & \quad \Gamma^n \vdash^{d\overline{b}} e & \quad \Gamma^n \vdash^{d\overline{b}} e & \quad \Gamma^n \vdash^{d\overline{b}} e' \quad \Gamma^n \vdash^{d\overline{b}} e' \\
\Gamma^n \vdash^{d\overline{b}} v^i & \quad \Gamma^n \vdash^{d\overline{b}} S e & \quad \Gamma^n \vdash^{d\overline{b}} A e e' \\
\end{align*}
\]

One can then prove by rule induction that if $\Gamma^n \vdash^{d\overline{b}} e$ then $fv(e) \subseteq \Gamma^n$. We prove by Rule Induction

\[
(\forall (\Gamma^n, e) \in \vdash^{d\overline{b}}) [(fv(e) \subseteq \Gamma^n)]
\]
Syntax with Distinguished Variables and Binding

Consider

\[ \text{Exp ::= V V | L V Exp | E Exp Exp} \]

We inductively define a set of judgements \( \Gamma^n \vdash_{db} e \) where \( n \geq 1 \).

\[
\begin{align*}
0 \leq i < n & \quad \Gamma^{n+1} \vdash_{db} e \\
\Gamma^n \vdash_{db} v^i & \quad \Gamma^n \vdash_{db} L v^n e \\
\Gamma^n \vdash_{db} E e e' & \quad \Gamma^n \vdash_{db} E e e'
\end{align*}
\]

One can then prove by rule induction that if \( \Gamma^n \vdash_{db} e \) then \( fv(e) \subset \Gamma^n \).
Notice that the rule for introducing abstractions $\mathsf{L} \, \mathsf{v}^n \, \mathsf{e}$ forces a distinguished choice of binding variable.

The advantage of distinguished binding is that the expressions correspond exactly to the terms of the $\lambda$-calculus, without the need to define $\alpha$-equivalence.

In essence, we are forced to pick a representative of each $\alpha$-equivalence class.
Syntax with Arbitrary Variables and Binding

- Expressions are still defined by

\[ Exp ::= \forall \forall | \lambda \forall Exp | \exists Exp \ Exp \]

- Now let \( \Delta \) range over all non-empty finite lists of variables which have distinct elements. Thus a typical non-empty \( \Delta \) is \( v^1, v^8, v^{100}, v^2 \in [\forall] \). Let \( x, y, \ldots \) range over \( \forall \).

- Define \( \Delta \vdash^{ab} e \) by

\[
\begin{align*}
& x \in \Delta \\
\hline
\Delta \vdash^{ab} x \\

& \Delta \vdash^{ab} e \\
\hline
\Delta \vdash^{ab} \lambda x \ e \\

& \Delta \vdash^{ab} e \quad \Delta \vdash^{ab} e' \\
\hline
\Delta \vdash^{ab} \exists \ e \ e' 
\end{align*}
\]
We define simultaneous substitution – used to define \( \alpha \)-equivalence, and to construct mathematical models.

\( \text{el}_p(\Delta) \) is the \( p \)th element of \( \Delta \), with position 0 the “first” element.

We will define by recursion over expressions \( e \), new expressions \( e\{\epsilon/\epsilon\} \) and \( e\{\Delta'/\Delta\} \), where \( \text{len}(\Delta) = \text{len}(\Delta') \).

For example,

\[
(L \ v^8 \ (A \ v^{10} \ v^2))\{v^3, v^8/v^8, v^2\} = L \ v^{11} \ (A \ v^{10} \ v^8)
\]
\[
(\forall x)\{\Delta'/\Delta\} \overset{\text{def}}{=} \begin{cases} 
  x & \text{if } (\forall p)(\text{el}_p(\Delta) \neq x) \\
  \text{el}_p(\Delta') & \text{if } (\exists p)(\text{el}_p(\Delta) = x)
\end{cases}
\]

\[
(\text{L } x \ e)\{\Delta'/\Delta\} \overset{\text{def}}{=} \begin{cases} 
  \text{L } x \ e\{\overline{\Delta}'/\overline{\Delta}\} & \\
  & \text{if } (\forall p)(\text{el}_p(\Delta') \neq x \lor \text{el}_p(\Delta) \not\in \text{fv}(e)) \\
  \text{L } x' \ e\{\overline{\Delta}',x'/\overline{\Delta},x\} & \\
  & \text{if } (\exists p)(\text{el}_p(\Delta') = x \land \text{el}_p(\Delta) \in \text{fv}(e))
\end{cases}
\]

\[
(\text{E } e \ e')\{\Delta'/\Delta\} \overset{\text{def}}{=} \text{E } e\{\Delta'/\Delta\} \ e'\{\Delta'/\Delta\}
\]
where

- $\overline{\Delta}$ is $\Delta$ with $x$ deleted (from position $p$, if it occurs) and, if $x$ does occur, $\overline{\Delta'}$ is $\Delta'$ with the element in position $p$ deleted, and is otherwise $\Delta'$; and

- $x'$ is the variable $\nu^w$ where $w$ is 1 plus the maximum of the indices appearing in $\overline{\Delta'}$ and $fv(e)$. 
- We inductively define the relation $\sim_\alpha$ of $\alpha$-equivalence
  - Single axiom (schema) $\text{L } x \ e \sim_\alpha \text{L } x' \ e\{x'/x\}$ with $x' \not\in \text{fv}(e)$
  - Rules such as
    \[
    \frac{e \sim_\alpha e' \quad e' \sim_\alpha e''}{e \sim_\alpha e''}
    \]
    \[
    e \sim_\alpha e' \quad \text{L } x \ e \sim_\alpha \text{L } x \ e'
    \]
- Note that the terms of the $\lambda$-calculus are given by the
  \[
  [e]_\alpha \overset{\text{def}}{=} \{e' | e' \sim_\alpha e\}\\
  \]
A Programme for Modelling Syntax

Step 1 define an abstract endofunctor $\Sigma_V$ on $\mathcal{F} \overset{\text{def}}{=} \text{Set}^{\mathcal{F}}$
(similar to the datatype in question);

Step 2 construct an initial algebra $T$ for $\Sigma_V$;

Step 3 show that the syntax yields a functor $\text{Exp}: \mathcal{F} \rightarrow \text{set}$;

Step 4 show that $T \cong \text{Exp}$
Modelling \( \textit{Exp} ::= \bigvee \bigvee | S \textit{Exp} | A \textit{Exp} \textit{Exp} \)

**Step 1**

- First, we define the functor \( \mathbb{V} : \mathbb{F} \rightarrow \textit{Set} \). Let \( \rho : m \rightarrow n \) in \( \mathbb{F} \). Then we set

\[
\forall m \overset{\text{def}}{=} \{ v^0, \ldots, v^{m-1} \} \quad \land \quad \forall \rho(v^i) \overset{\text{def}}{=} v^{\rho i}
\]

- Define a functor \( \Sigma_{\mathbb{V}} : \textit{Set}^\mathbb{F} \rightarrow \textit{Set}^\mathbb{F} \) by setting

\[
\Sigma_{\mathbb{V}} \xi \overset{\text{def}}{=} \mathbb{V} + \xi + \xi^2
\]
Step 2

- \( T \overset{\text{def}}{=} \bigcup (S_r | r \geq 0) \).
- \( S_0 \overset{\text{def}}{=} \emptyset \), the empty presheaf, and
  \[
  S_{r+1} \overset{\text{def}}{=} \sum V S_r = V + S_r + S_r^2
  \]
- Need to check \( i_r : S_r \hookrightarrow S_{r+1} \) for all \( r \geq 0 \). We use induction over \( r \).
- It is immediate that \( i_0 : S_0 \hookrightarrow S_1 \).
Now suppose that \( i_r : S_r \hookrightarrow S_{r+1} \). We are required to show that \( i_{r+1} : S_{r+1} \hookrightarrow S_{r+2} \), that is,

\[
\forall n + S_r n + (S_r n)^2 \subset \forall n + S_{r+1} n + (S_{r+1} n)^2
\]

\[
\forall \rho + S_r \rho + (S_r \rho)^2 \subset \forall \rho + S_{r+1} \rho + (S_{r+1} \rho)^2
\]

\[
\forall n' + S_r n' + (S_r n')^2 \subset \forall n' + S_{r+1} n' + (S_{r+1} n')^2
\]

\[\Sigma \forall i_r = \text{id}_{\forall} + i_r + i_r^2.\] Thus we have \( i_{r+1} = \Sigma \forall i_r \).
We define the structure map $\sigma_T \overset{\text{def}}{=} [\kappa, \kappa', \kappa''] : V + T + T^2 \to T$

$\forall \cong S_1$, so that $\kappa : \forall \cong S_1 \hookrightarrow T$. $S_1 = \forall + \emptyset + \emptyset^2$, and so $S_1 n = \forall n \times \{1\}$.

We define $\kappa'$ by

$$\kappa'_r : S_r \xrightarrow{1_{S_r}} \forall + S_r + S_r^2 = S_{r+1} \hookrightarrow T$$

check that $\kappa'_{r+1} \circ i_r = \kappa'_r$, ie

$$i_r \quad i_{r+1} \quad i_{r+1} = id_{\forall} + i_r + i_r^2$$
Write $S'_r \overset{\text{def}}{=} S^2_r$. Consider the family of morphisms

$$\kappa''': S'_r = S^2_r \xrightarrow{1_{S^2_r}} \mathcal{V} + S_r + S^2_r = S_{r+1} \hookrightarrow T$$

$\kappa''_r$ satisfy the union conditions . . .

Hence they define $\kappa'': U \rightarrow T$ where $U \overset{\text{def}}{=} \bigcup_r S'_r$. But note that

$$Un = \bigcup_r S'_r n = \bigcup_r (S_r n)^2 = (\bigcup_r S_r n)^2 = (T n)^2 = T^2 n$$

and also $U \rho = T^2 \rho$. Hence $U = T^2$. 
We check initiality

\[
\begin{align*}
\forall + T + T^2 & \xrightarrow{\sigma_T} T \\
\forall + \overline{\alpha} + \overline{\alpha}^2 & \xrightarrow{(*)} \overline{\alpha} \\
\forall + A + A^2 & \xrightarrow{\alpha} A
\end{align*}
\]

To define \(\overline{\alpha} : T \rightarrow A\) we specify a family \(\overline{\alpha}_r : S_r \rightarrow A\).

Note that \(\overline{\alpha}_0 : \emptyset \rightarrow A\) and thus we define

\[
(\overline{\alpha}_0)_n \overset{\text{def}}{=} \emptyset : \emptyset \rightarrow An
\]

Note that \(\overline{\alpha}_{r+1} : S_{r+1} = \forall + S_r + S_r^2 \rightarrow A\) and hence

\[
\overline{\alpha}_{r+1} \overset{\text{def}}{=} [\alpha \circ \iota_{\forall}, \alpha \circ \iota_A \circ \overline{\alpha}_r, \alpha \circ \iota_{A^2} \circ \overline{\alpha}_r^2]
\]

Need to verify \(\overline{\alpha}_{r+1} \circ i_r = \overline{\alpha}_r\) for all \(r \geq 0\).
Proving that the diagram (*) commutes is equivalent to proving

\[ [\overline{\alpha} \circ \kappa, \overline{\alpha} \circ \kappa', \overline{\alpha} \circ \kappa'] = [\alpha \circ \iota_{V}, \alpha \circ \iota_{A} \circ \overline{\alpha}, \alpha \circ \iota_{A2} \circ \overline{\alpha}^{2}] \]

We prove that \( \overline{\alpha}_{n} \circ \kappa'_{n} = \alpha_{n} \circ \iota_{An} \circ \overline{\alpha}_{n} : Tn \rightarrow An. \)

Suppose that \( \xi \) is an arbitrary element of \( Tn \), where \( \xi \in S_{r}n. \)

\[
\begin{align*}
\overline{\alpha}_{n}(\kappa'_{n}(\xi)) & = \overline{\alpha}_{n}(\iota_{S_{r}n}(\xi)) \\
& = (\overline{\alpha}_{r+1})_{n}(\iota_{S_{r}n}(\xi)) \\
& = [\alpha_{n} \circ \iota_{Vn}, \alpha_{n} \circ \iota_{An} \circ (\overline{\alpha}_{r})_{n}, \alpha_{n} \circ \iota_{(An)^{2}} \circ (\overline{\alpha}_{r})_{n}^{2}] \iota_{S_{r}n}(\xi) \\
& = \alpha_{n}(\iota_{An}(\overline{\alpha}_{r})_{n}(\xi))) \\
& = \alpha_{n}(\iota_{An}(\overline{\alpha}_{n}(\xi)))
\end{align*}
\]
Step 3

Suppose that $\rho : n \to n'$ is any function. We define

$$\textit{Exp}_{d\bar{B}} n \overset{\text{def}}{=} \{ e \mid \Gamma^n \vdash_{d\bar{B}} e \}$$

We can define $(\textit{Exp}_{d\bar{B}} \rho)e$ by recursion over $e$, by setting

- $(\textit{Exp}_{d\bar{B}} \rho)(V v^i) \overset{\text{def}}{=} V \rho i$
- $(\textit{Exp}_{d\bar{B}} \rho)(S e) \overset{\text{def}}{=} S (\textit{Exp}_{d\bar{B}} \rho)e$
- $(\textit{Exp}_{d\bar{B}} \rho)(A e e') \overset{\text{def}}{=} A (\textit{Exp}_{d\bar{B}} \rho)e (\textit{Exp}_{d\bar{B}} \rho)e'$
\[ \ldots \text{and then showing that if } e \in \text{Exp}_{\overline{dB}} n, \text{ then} \]
\[(\text{Exp}_{\overline{dB}} \rho) e \in \text{Exp}_{\overline{dB}} n'.\]

\[\text{Thus we have a function}\]

\[\text{Exp}_{\overline{dB}} \rho : \text{Exp}_{\overline{dB}} n \rightarrow \text{Exp}_{\overline{dB}} n'\]

\[\text{for any } \rho : n \rightarrow n'.\]

\[\text{Note that there are natural transformations}\]

\[S : \text{Exp}_{\overline{dB}} \rightarrow \text{Exp}_{\overline{dB}} \quad \land \quad A : \text{Exp}_{\overline{dB}}^2 \rightarrow \text{Exp}_{\overline{dB}}\]
Step 4

- We now show that $T \cong \mathit{Exp}_{\mathit{db}}$ in $\mathcal{F}$

- We define $\phi : T \to \mathit{Exp}_{\mathit{db}}$ and $\psi : \mathit{Exp}_{\mathit{db}} \to T$, such that
  \[ \phi_n : T_n \cong \mathit{Exp}_{\mathit{db}} n : \psi_n \]

- To specify $\phi : T \to \mathit{Exp}_{\mathit{db}}$ define a family $\phi_r : S_r \to \mathit{Exp}_{\mathit{db}}$.
  - $\phi_0 : S_0 = \emptyset \to \mathit{Exp}_{\mathit{db}}$ has components $(\phi_0)_n : \emptyset \to \mathit{Exp}_{\mathit{db}} n$
  - Recursively we define
    \[ \phi_{r+1} \overset{\text{def}}{=} [V, S \circ \phi_r, A \circ \phi_r^2] : S_{r+1} = V + S_r + S_r^2 \to \mathit{Exp}_{\mathit{db}} \]
To specify $\psi: \text{Exp}_{d\bar{b}} \rightarrow T$, for any $n$ in $F$ we define functions

$$\psi_n: \text{Exp}_{d\bar{b}} n \rightarrow Tn$$

as follows.

1. $\psi_n(V \, v^i) \overset{\text{def}}{=} (v^i, 1) \in S_1n$
2. $\psi_n(\text{S} \, e) \overset{\text{def}}{=} \iota_{S,n}(\psi_n(e))$ where $r \geq 1$ is the height of the deduction of $\text{S} \, e$
3. $\psi_n(\text{A} \, e \, e') \overset{\text{def}}{=} \iota_{(S,n)^2}((\psi_n(e), \psi_n(e')))$ where $r \geq 1$ is the height of the deduction of $\text{A} \, e \, e'$. 
We next check that for any $n$ in $\mathbb{F}$,

\[
\begin{array}{c}
\phi_n \\
\cong \\
\psi_n \\
\end{array}
\]

\[
\begin{array}{c}
Tn \\
\cong \\
\text{Exp}_{db} n \\
\end{array}
\]

Suppose $\xi \in S_r n \subset Tn$ for some $r$. Then by definition,

\[
\psi_n(\phi_n(\xi)) = \psi_n((\phi_r)_n(\xi))
\]

By induction, for all $r \geq 0$, if $\xi \in S_r n$ and $n$ any object of $\mathbb{F}$, then

\[
\psi_n((\phi_r)_n(\xi)) = \xi
\]
Let $\xi \in S_{r+1}n = \nabla n + S_r n + S_r n^2$. Then we have

$$\psi_n((\phi_{r+1})_n(\xi)) = \psi_n([\nabla_n, S_n \circ (\phi_r)_n, A_n \circ (\phi_r)_n^2](\xi))$$

Consider the case when $\xi = \iota_{S_r n}(\xi')$ for some $\xi' \in S_r n$. We have

$$\psi_n((\phi_{r+1})_n(\xi)) = \psi_n((S_n \circ (\phi_r)_n)(\xi'))$$

$$= \psi_n(S(\phi_r)_n(\xi'))$$

$$= \iota_{S_r n}(\psi_n((\phi_r)_n(\xi')))$$

$$= \iota_{S_r n}(\xi')$$

$$= \xi$$
**Modelling**

\[ \text{Exp ::= } \mathcal{V} \mathcal{V} \mid \mathcal{L} \mathcal{V} \text{Exp} \mid \mathcal{E} \text{Exp Exp} \]

**Case** \( \Gamma^n \vdash^{\text{db}} e \) with *Distinguished Binding*

- **Step 1** The abstract endofunctor \( \Sigma_{\mathcal{V}} : \mathcal{F} \rightarrow \mathcal{F} \) is

\[
\Sigma_{\mathcal{V}} \xi \overset{\text{def}}{=} \mathcal{V} + \delta \xi + \xi^2
\]

*Motto:* Any constructor with 1 argument and which binds \( b \) variables is modelled by \( \delta^b \xi \). Thus

Split \( P \) as \( \langle x, y \rangle \) in \( E \)

would be modelled by \( \xi \mapsto \xi \times \delta \delta \xi \)
Step 2 We can show that the functor $\Sigma_V$ has an initial algebra $\sigma_T: \Sigma_V T \rightarrow T$, by adapting the previous methods.

Have to define

$$\sigma_T \overset{\text{def}}{=} \left[ \kappa, \kappa', \kappa'' \right] \overset{\text{def}}{=} V + \delta T + T \times T \rightarrow T$$

via

$$\kappa'_r: \delta S_r \overset{\iota_{S_r}}{\rightarrow} V + \delta S_r + S_r^2 = S_{r+1} \leftarrow T$$

as

$$(\delta T)_n \overset{\text{def}}{=} T(n + 1) = \bigcup_{r} S_r(n + 1) = \bigcup_{r}(\delta S_r)n = (\bigcup_{r} \delta S_r)n$$
Step 3 Suppose $\rho : n \rightarrow n'$. Define

$$Exp_{db} n \overset{\text{def}}{=} \{ e \mid \Gamma^n \vdash_{db} e \}$$

Let $\rho \{ n'/n \} : n + 1 \rightarrow n' + 1$ be

$$\rho \{ n'/n \}(j) \overset{\text{def}}{=} \begin{cases} j & \text{if } 0 \leq j \leq n - 1 \\ n' & \text{if } j = n \end{cases}$$

Consider

- $(Exp_{db} \rho)(L \nu^n e) \overset{\text{def}}{=} L \nu^{n'} (Exp_{db} \rho \{ n'/n \})(e)$ and
- $(Exp_{db} \rho)(E e e') \overset{\text{def}}{=} E ((Exp_{db} \rho)e) ((Exp_{db} \rho)e')$

If $\Gamma^n \vdash_{db} e$ and $\rho : n \rightarrow n'$, then $\Gamma^n \vdash_{db} (Exp_{db} \rho)e$ yielding a functor $Exp_{db}$ in $\mathcal{F}$. 
There are natural transformations

\[ L : \delta \, \text{Exp}_{db} \rightarrow \text{Exp} \quad \land \quad E : \text{Exp}^2 \rightarrow \text{Exp} \]

The components are functions

\[ L_n : \text{Exp}_{db} (n + 1) \rightarrow \text{Exp}_{db} n \quad \leftrightarrow \quad e \mapsto L \, \nu^n \, e \]

Naturality is

\[ (\delta \, \text{Exp}_{db}) n = \text{Exp}_{db} (n + 1) \xrightarrow{L_n} \text{Exp}_{db} n \]

\[ (\delta \, \text{Exp}_{db}) \rho = \text{Exp}_{db} (\rho + \text{id}_1) \]

\[ (\delta \, \text{Exp}_{db}) n' = \text{Exp}_{db} (n' + 1) \xrightarrow{L_{n'}} \text{Exp}_{db} n' \]
Note that at the element $e$, this requires that

$$\mathbb{L} \nu^{n'} (\text{Exp}_{db} \rho\{n'/n\})e = \mathbb{L} \nu^{n'} ((\text{Exp}_{db} (\rho + id_1))e)$$

This equality holds if and only if

$$\rho\{n'/n\} = \rho + id_1$$

...which is true if and only if in $\mathbb{F}$

$$\nu_1: 1 \to m + 1 \quad \ast \mapsto m \quad \nu_m: m \to m + 1 \quad i \mapsto \rho i$$

Step 4 A routine calculation that $T \simeq \text{Exp}_{db}$
Modelling \( Exp ::= \forall \forall \mid L \forall Exp \mid E Exp Exp \)

Case \( \Delta \vdash^{ab} e \) with Arbitrary Binding

- **Step 1** The abstract endofunctor \( \Sigma_\forall: \mathcal{F} \rightarrow \mathcal{F} \) is

  \[
  \Sigma_\forall \xi \overset{\text{def}}{=} \forall + \delta \xi + \xi^2
  \]

  **Note:** The functor is the SAME as before

- **Step 2** Thus solving for the initial algebra is the same as before!
Step 3 We define $Exp_{ab}$. For $n$ in $\mathbb{F}$ we set

\[ Exp_{ab} \ n \overset{\text{def}}{=} \{ [e]_\alpha \mid \Gamma^n \vdash_{ab} e \} \]

Now let $\rho : n \rightarrow n'$. We define

\[ (Exp_{ab} \ \rho)([e]_\alpha) \overset{\text{def}}{=} [e\{v^0, \ldots, v^{\rho(n-1)} / v^0, \ldots, v^{n-1}\}]_\alpha \]

One has to check that this is well defined ... see the notes.
Step 4  Note that current Step 2 was same as before. Rather than prove $Exp_{ab} \simeq T$ as a final step, we could in fact make use of the previous work, which proved that $Exp_{db} \simeq T$. Thus we omit Step 2, and instead show

$$\phi: Exp_{ab} \simeq Exp_{db} : \psi$$
The components of $\psi$ are functions $\psi_n : \text{Exp}_{db} n \to \text{Exp}_{ab} n$ given by $\psi_n(e) \overset{\text{def}}{=} [e]_\alpha$.

We consider the naturality of $\psi$ at a morphism $\rho : n \to n'$, computed at an element $\xi$ of $\text{Exp}_{db} n$. We show naturality for the case $\xi = L v^n e$.

\[
(\text{Exp}_{ab} \rho) \circ \psi_n(\xi) = (\text{Exp}_{ab} \rho)[L v^n e]_\alpha
\]
\[
= [(L v^n e)\{v^{\rho_0}, \ldots, v^{\rho(n-1)}/v^0, \ldots, v^{n-1}\}]_\alpha
\]
\[
\overset{\text{def}}{=} \square
\]

Let us consider the case when renaming takes place.

Suppose that there is a $j$ for which $\rho(j) = n$ and $v^j \in f\nu(e)$. 

Then

\[(L \ v^n \ e)\{v^{\rho(0)}, \ldots, v^{\rho(n-1)}/v^0, \ldots, v^{n-1}\} =
\]

\[L \ v^w \ e\{v^{\rho(0)}, \ldots v^{\rho(n-1)}, v^w/v^0, \ldots, v^{n-1}, v^n\}\]

- \[w = 1 + MaxIndex(e; \rho(0), \ldots, \rho(n-1))\] thus \(\rho(i) < w\) for all \(0 \leq i \leq n - 1\).
- But \(fv(e) \subset v^0, \ldots, v^n\) and \(n = \rho(j) \in \rho(0), \ldots, \rho(n-1)\).
- Also \(\rho(i) < n',\) and so we must have \(w \leq n'\).
- If \(w < n'\), then \(v^{n'}\) is not free in \(e\{v^{\rho(0)}, \ldots v^{\rho(n-1)}, v^w/v^0, \ldots, v^{n-1}, v^n\}\) and otherwise \(w = n'\).
Either way (why!?)

\[ \text{L } v^w e\{v^{\rho(0)}, \ldots, v^{\rho(n-1)}, v^w/v^0, \ldots, v^{n-1}, v^n\} \]

\[ \sim_\alpha \text{L } v'^{n'} e\{v^{\rho(0)}, \ldots, v^{\rho(n-1)}, v'^{n'}/v^0, \ldots, v^{n-1}, v^n\} \]

and so

\[ \square = \left[ \text{L } v'^{n'} e\{v^{\rho(0)}, \ldots, v^{\rho(n-1)}, v'^{n'}/v^0, \ldots, v^{n-1}, v^n\} \right]_\alpha \]

\[ = \left[ \text{L } v'^{n'} (Exp_{db} \rho\{n'/n\}) e \right]_\alpha \]

\[ = \psi_{n'} \circ (Exp_{db} \rho)(\xi) \]
Next we define $\phi_n : \text{Exp}_{ab} n \rightarrow \text{Exp}_{db} n$ by setting

$$\phi_n([e]_{\alpha}) \overset{\text{def}}{=} R^n(e)$$

where

- $R^m(\forall x) \overset{\text{def}}{=} \forall x$
- $R^m(\exists x e) \overset{\text{def}}{=} \exists v^m R^{m+1}(e[v^m/x])$
- $R^m(\exists e e') \overset{\text{def}}{=} \exists R^m(e) R^m(e')$

This is best understood by a simple example . . .

The verification that

$$\phi : \text{Exp}_{ab} \simeq \text{Exp}_{db} : \psi$$

is omitted from the lectures. See the notes.
\[
R^3(L v^7 (L v^3 (E v^7 (E v^0 (L v^6 (E v^2 v^3))))))) \\
= L v^3 R^4(L v^3 (E v^7 (E v^0 (L v^6 (E v^2 v^3)))))) \{v^3/v^7\} \\
= L v^3 R^4(L v^4 (E v^3 (E v^0 (L v^6 (E v^2 v^4)))))) \\
= L v^3 (L v^4 R^5(E v^3 (E v^0 (L v^6 (E v^2 v^4)))))) \{v^4/v^4\} \\
= L v^3 (L v^4 (E v^3 (E v^0 (R^5(L v^6 (E v^2 v^4))))))) \\
= L v^3 (L v^4 (E v^3 (E v^0 (L v^5 R^5(E v^2 v^4)\{v^5/v^6\})))))) \\
= L v^3 (L v^4 (E v^3 (E v^0 (L v^5 (R^5(E v^2 v^4))))))) \\
= L v^3 (L v^4 (E v^3 (E v^0 (L v^5 (E v^2 v^4))))) \\
= L v^3 (L v^4 (E v^3 (E v^0 (L v^5 (E v^2 v^4)))))
\]
Where to Now? You might

- learn more *Category Theory*;
- learn more *Type Theory*;
- learn more *Categorical Type Theory*;
- spend some time trying to understand the key problems and issues concerning modelling and reasoning about binding syntax; and
- read the current research literature on modelling and implementing binding syntax.