Category Theory

Roy L. Crole

University of Leicester, UK

April 2015
Category Theory is a theory of abstraction (of algebraic structure).

It had its origins in Algebraic Topology with the work of Eilenberg and Mac Lane (1942-45).

It provides tools and techniques which allow the formulation and analysis of common features amongst apparently different mathematical/computational theories.

We can discover new relationships between things that are seemingly unconnected.

Category theory concentrates on how things behave and not on internal details.

As such, category theory can clarify and simplify our ideas—and indeed lead to new ideas and new results.
Connections with Computer Science were first made in the 1980s, and the subject has played a central role ever since.

Some contributions (chosen by me ...there are many many more) are

- Cartesian closed categories as models of pure functional languages.
- The use of strong monads to model notions of computation (well incorporated into Haskell).
- Precise correspondences between categorical structures and type theories (as internal languages).
- The categorical solution of domain equations as models of recursive types.
- Categories for concurrent computation.
- Nominal categories as models of variable binding.
Seminar Outline

Categories

Functors

Natural Transformations

Isomorphisms, Products, Coproducts

Algebras

Case Study: Modelling (Haskell) Algebraic Datatypes

Adjunctions

Colimits and Applications to Initial Algebras
Examples of Categories

- The collection of all sets and all functions
  - Each set has an identity function; functions compose; composition is associative.

- The collection of all elements of a preorder and all instances of the order relation (relationships) ≤
  - Each element has an identity relationship (reflexivity); relationships compose (transitivity); composition is associative.

- The collection of all elements of a singleton \( \{ * \} \) (!) and any collection of algebraic terms with just one variable \( x_0 \)
  - * has an identity term \( x_0 \); terms compose (substitution); composition is associative.
A category $\mathcal{C}$ is specified by the following data:

- A collection $\text{ob} \mathcal{C}$ of entities called objects. An object will often be denoted by a capital letter such as $A$, $B$, $C$...
- For any two objects $A$ and $B$, a collection $\mathcal{C}(A,B)$ of entities called morphisms. A morphism in $\mathcal{C}(A,B)$ will often be denoted by a small letter such as $f$, $g$, $h$...
- If $f \in \mathcal{C}(A,B)$ then $A$ is called the source of $f$, and $B$ is the target of $f$ and we write (equivalently) $f : A \rightarrow B$. 
Definition of A Category

A category \( C \) is specified by the following data (continued):

- There is an operation assigning to each object \( A \) of \( C \) an identity morphism \( id_A : A \to A \).
- There is an operation
  \[
  C(B,C) \times C(A,B) \to C(A,C)
  \]
  assigning to each pair of morphisms \( f : A \to B \) and \( g : B \to C \) their composition which is a morphism denoted by \( g \circ f : A \to C \) or just \( gf : A \to C \).
- Such morphisms \( f \) and \( g \), with a common source and target \( B \), are said to be composable.
A category \( \mathcal{C} \) is specified by the following data (continued):

- These operations are **unitary**
  
  \[
  id_B \circ f = f : A \to B \\
  f \circ id_A = f : A \to B
  \]

- and **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 we say “\( f \) is a morphism” we implicitly assume that the source and target are recoverable, that is, we can work out \( f \in \mathcal{C}(A, B) \) for some \( A \) and \( B \).
More Examples

- The category \textbf{Part} with \textit{ob Part} all \textit{sets} and morphisms \textbf{Part}(A, B) the \textit{partial functions} \( A \rightarrow B \).
  - The identity function \( id_A \) is a partial function!
  - Given \( f: A \rightarrow B \), \( g: B \rightarrow C \), then for each element \( a \) of \( A \), \((g \circ f)(a)\) is defined with value \( g(f(a)) \) if and only if both \( f(a) \) and \( g(f(a)) \) are defined.

- Given a category \( C \), the \textbf{opposite category} \( C^{\text{op}} \) has
  - \( \text{ob } C^{\text{op}} \overset{\text{def}}{=} \text{ob } C \) and \( C^{\text{op}}(A, B) \overset{\text{def}}{=} \{ f^{\text{op}} \mid f \in C(B, A) \} \).
  - The identity on an object \( A \) in \( C^{\text{op}} \) is defined to be \( \text{id}_A^{\text{op}} \).
  - If \( f^{\text{op}}: A \rightarrow B \) and \( g^{\text{op}}: B \rightarrow C \) are morphisms in \( C^{\text{op}} \), then \( f: B \rightarrow A \) and \( g: C \rightarrow B \) are composable morphisms in \( C \).

We define \( g^{\text{op}} \circ f^{\text{op}} \overset{\text{def}}{=} (f \circ g)^{\text{op}}: A \rightarrow C \).
More Examples

- A **discrete** category is one for which the only morphisms are identities.

- A **semigroup** \((S, b)\) is a set \(S\) together with an associative binary operation \(b: S \times S \to S, (s, s') \mapsto s \cdot s'\). An **identity element** for a semigroup \(S\) is some (necessarily unique) element \(e\) of \(S\) such that for all \(s \in S\) we have \(e \cdot s = s \cdot e = s\). A **monoid** \((M, b, e)\) is a semigroup \((M, b)\) with identity element \(e\). Any monoid is a **single object category** \(C\) with \(C(\ast, \ast) \overset{\text{def}}{=} M\). Concrete examples are
  - Addition on the natural numbers, \((\mathbb{N}, +, 0)\).
  - Concatenation of finite lists over a set \(A\), \((\text{list}(A), ++, [])\).
More Examples

- **Mon** has objects monoids and morphisms monoid homomorphisms: \( h: M \rightarrow M' \) is a homomorphism if \( h(e) = e \) and \( h(m_1 \cdot m_2) = h(m_1) \cdot h(m_2) \) for all \( m_i \in M \).

- **PreSet** has objects preorders and morphisms the monotone functions; and **ParSet** has objects partially ordered sets and morphisms the monotone functions.

- The category of relations **Rel** has objects sets and morphisms binary relations on sets; composition is relation-composition.

- The category of lattices **Lat** has objects lattices and morphisms the lattice homomorphisms.

- The category **CLat** has objects the complete lattices and morphisms the complete lattice homomorphisms.

- The category **Grp** of groups and homomorphisms.
Examples of Functors

- Let $C$ be a category. The **identity** functor $id_C : C \to C$ is defined by $id_C(A) \overset{\text{def}}{=} A$ on objects and $id_C(f) \overset{\text{def}}{=} f$ on morphisms; so $f : A \to B \implies id_C(f) : id_C(A) \to id_C(B)$.

- Let $(X, \leq_X)$ and $(Y, \leq_Y)$ be categories and $m : X \to Y$ a monotone function. Then $m$ gives rise to a functor $M : (X, \leq_X) \to (Y, \leq_Y)$ defined by $M(x) \overset{\text{def}}{=} m(x)$ on objects $x \in X$ and by $M(\leq_X) = \leq_Y$ on morphisms; since $m$ is monotone, $\leq_X : x \to x' \implies M(\leq_X) : M(x) \to M(x')$. 
Examples of Functors

- We may define a functor $F: \textbf{Set} \to \textbf{Mon}$ by $FA \overset{\text{def}}{=} \text{list}(A)$ and $Ff \overset{\text{def}}{=} \text{map}(f)$, where $\text{map}(f): \text{list}(A) \to \text{list}(B)$ is defined by

\[
\begin{align*}
\text{map}(f)([\]) & \overset{\text{def}}{=} [] \\
\text{map}(f)([a_1, \ldots, a_n]) & \overset{\text{def}}{=} [f(a_1), \ldots, f(a_n)]
\end{align*}
\]

It is easy to see that $\text{map}(f)$ is a homomorphism of monoids.

- Note that $F(id_A) = id_{FA}$

\[
\begin{align*}
F(id_A)([a_1, \ldots, a_n]) & \overset{\text{def}}{=} \text{map}(id_A)([a_1, \ldots, a_n]) \\
& = id_{\text{list}(A)}([a_1, \ldots, a_n]) \\
& \overset{\text{def}}{=} id_{FA}([a_1, \ldots, a_n])
\end{align*}
\]
Examples of Functors

... and note that $F(g \circ f) = Fg \circ Ff$

\[
F(g \circ f)([a_1, \ldots, a_n]) \overset{\text{def}}{=} \text{map}(g \circ f)([a_1, \ldots, a_n]) = [\,(g \circ f)(a_1), \ldots, (g \circ f)(a_n)\,] = [\,g(f(a_1)), \ldots, g(f(a_n))\,] = \text{map}(g)([f(a_1), \ldots, f(a_n)]) = \text{map}(g)(\text{map}(f)([a_1, \ldots, a_n])) = (Fg \circ Ff)([a_1, \ldots, a_n]).
\]
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 $\mathcal{D}$, such that
  - $F(id_A) = id_{FA}$, and
  - $F(g \circ f) = Fg \circ Ff$ provided $g \circ f$ is defined.
More Functor Examples

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

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

where $f: A \to B$ is a function and $f_*$ is defined by

$$f_*(A') \overset{\text{def}}{=} \{ f(a') \mid a' \in A' \} \quad \text{where} \quad A' \in \mathcal{P}(A).$$

- $f_*$ is sometimes called the **direct image** of $f$.  

More Functor Examples

- We can define a **contravariant powerset** functor $\mathcal{P} : \text{Set}^{\text{op}} \rightarrow \text{Set}$ by setting

  $$f^{\text{op}} : B \rightarrow A \mapsto f^{-1} : \mathcal{P}(B) \rightarrow \mathcal{P}(A),$$

where $f : A \rightarrow B$ 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 $B' \in \mathcal{P}(B)$.

- $f^*$ is sometimes called the **inverse image** of $f$. 

$\text{MGS 2015, 7-11 April, University of Sheffield, UK}$
Definition of a Natural Transformation

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

\[
\begin{array}{ccc}
FA & \xrightarrow{\alpha_A} & GA \\
\downarrow{Ff} & & \downarrow{Gf} \\
FB & \xrightarrow{\alpha_B} & GB
\end{array}
\]

The \( \alpha_A \) are the components of the natural transformation.
Examples of Natural Transformations

- Recall $F: \mathbf{Set} \to \mathbf{Mon}$ where $FA \overset{\text{def}}{=} \text{list}(A)$ and $F(f: A \to B) \overset{\text{def}}{=} \text{map}(f): \text{list}(A) \to \text{list}(B)$. Define a natural transformation $\text{rev}: F \to F$, by specifying functions $\text{rev}_A: \text{list}(A) \to \text{list}(A)$ where

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

We check

\[
(Ff \circ \text{rev}_A)([a_1, \ldots, a_n]) = [f(a_n), \ldots, f(a_1)]
\]
\[
= (\text{rev}_B \circ Ff)([a_1, \ldots, a_n]).
\]
Examples of Natural Transformations

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 \( D^C \) with objects functors from \( C \) to \( D \), morphisms natural transformations between such functors, and composition as given above.
Examples of Natural Transformations

- Define a functor $F_X : \textbf{Set} \to \textbf{Set}$ by
  - (! Products) $F_X(A) \overset{\text{def}}{=} (X \Rightarrow A) \times X$ on objects
  - (! Products) $F_X(f) \overset{\text{def}}{=} (f \circ -) \times id_X$ on morphisms

Then define a natural transformation $ev : F_X \to id_{\textbf{Set}}$ with components $ev_A : (X \Rightarrow A) \times X \to A$ 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 let $f : A \to B$ and note that

$$
(id_{\textbf{Set}}(f) \circ ev_A)(g, x) = f(ev_A(g, x)) = \ldots (ev_B \circ F_X(f))(g, x).
$$
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 \( g \circ f = id_A \).
- \( g \) is an **inverse** for \( f \) and vise versa.
- \( A \) is **isomorphic** to \( B \), \( A \cong B \), if such a mutually inverse pair of morphisms exists.
- Bijectons in \( \textbf{Set} \) are isomorphisms. There are typically many isomorphisms witnessing that two sets are bijective.
- 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 \( x \cong y \).
Definition of Binary Products

A binary product of objects $A$ and $B$ in $C$ 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 exists a unique morphism $\langle f, g \rangle: C \to A \times B$ for which $\langle f, g \rangle: C \to A \times B$ is called the mediating morphism for $f$ and $g$. 

\[ \begin{array}{ccc}
A & \overset{\pi_A}{\leftarrow} & A \times B \\
\downarrow & & \downarrow \pi_B \\
C & \longrightarrow & B
\end{array} \]

$\exists! \langle f, g \rangle \colon C \to A \times B$
Examples of Binary Products

- Let \((X, \leq)\) be a preorder. \(l \in X\) is a lower bound of \(x, y \in X\) just in case \(l \leq x, y\). \(u \in X\) is a upper bound of \(x, y \in X\) just in case \(x, y \leq u\).

- \(x \in S \subseteq X\) is greatest in \(S\) if \((\forall s \in S)(s \leq x)\) and is least in \(S\) if \((\forall s \in S)(x \leq s)\).

- In a preorder a greatest lower bound \(x \land y\) of \(x\) and \(y\) (if it exists) is a binary product \(x \times y\) of the category determined by \((X, \leq)\) with projections \(x \land y \leq x\) and \(x \land y \leq y\). \(x \land y\) is also called the meet of \(x\) and \(y\).
Examples of Binary Products

The binary product of $A$ and $B$ in $\mathbf{Set}$ has

$$A \times B \overset{\text{def}}{=} \{ (a,b) \mid A \in A, b \in B \}$$

with projection functions $\pi_A(a,b) \overset{\text{def}}{=} a$ and $\pi_B(a,b) \overset{\text{def}}{=} b$. The mediating function for any $f : C \to A$ and $g : C \to B$ is

$$\langle f, g \rangle (c) \overset{\text{def}}{=} (f(c), g(c))$$.

In any $\mathbf{C}$, if $p_i : P \to A_i$ is any product of $A_1$ and $A_2$ then $A_1 \times A_2 \cong P$. All binary products are determined up to isomorphism: Existence yields mediating morphisms $\phi : A_1 \times A_2 \to P$ and $\psi : P \to A_1 \times A_2$; uniqueness means that $\phi$ and $\psi$ witness an isomorphism.
Definition of Finite Products

A **product** of a non-empty finite family of objects \((A_i \mid i \in I)\) in \(C\), where \(I \overset{\text{def}}{=} \{1, \ldots, n\}\), is specified by

- an object \(A_1 \times \ldots \times A_n\) (or \(\prod_{i \in I} A_i\)) in \(C\), and
- for every \(j \in I\), a morphism \(\pi_j: A_1 \times \ldots \times A_n \to A_j\) in \(C\) called the \(j\)th **product projection**

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

\[
\langle f_1, \ldots, f_n \rangle: C \to A_1 \times \ldots \times A_n
\]

for which given any \(j \in I\), we have \(\pi_j \circ \langle f_1, \ldots, f_n \rangle = f_j\).

Note: We get binary products when \(I \overset{\text{def}}{=} \{1, 2\}\)!
Examples of Finite Products

▸ A finite product of \((A_1, \ldots, A_n) \equiv (A_i \mid i \in I)\) in \(\mathbf{Set}\) is given by the cartesian product \(A_1 \times \ldots \times A_n\) with the obvious projection functions. Given functions \((f : C \to A_i \mid i \in I)\) then

\[
\langle f_1, \ldots, f_n \rangle (c) \overset{\text{def}}{=} (f_1(c), \ldots, f_n(c))
\]

▸ In a preorder \((X, \leq)\), a finite product \(x_1 \times \ldots \times x_n\), if it exists, is a meet (greatest lower bound) of \((x_1, \ldots, x_n)\).

▸ A **terminal** object 1 in a category \(\mathbf{C}\) has the property that there is a unique morphism \(!_A : A \to 1\) for every \(A \in \text{ob } \mathbf{C}\). It is the finite product of an empty family of morphisms (check this!). Such a 1 may not exist, but is unique up to isomorphism if it does.
Definition of Finite Coproducts

A coproduct of a non-empty family of objects \((A_i \mid i \in I)\) in \(C\), where \(I = \{1, \ldots, n\}\), is specified by

- an object \(A_1 + \ldots + A_n (\Sigma_{i \in I} A_i)\), together with
- insertion morphisms \(\iota_j: A_j \rightarrow A_1 + \ldots + A_n\),

such that for any \(C\) and any family of morphisms \((f_i: A_i \rightarrow C \mid i \in I)\) there is a unique morphism

\([f_1, \ldots, f_n]: A_1 + \ldots + A_n \rightarrow C\)

for which given any \(j \in I\), we have \([f_1, \ldots, f_n] \circ \iota_j = f_j\).
In the case that $I \overset{\text{def}}{=} \{1, 2\}$ we have

$A_1 \xrightarrow{\iota_2} A_1 + A_2 \xleftarrow{\iota_2} A_2$

$\exists! [f_1, f_2]$

$C$

(Compare to the diagrams for colimits later on.)
Examples of (Co)Products

- In $\textbf{Set}$ the binary coproduct of sets $A_1$ and $A_2$ is given by their disjoint union $A_1 \uplus A_2$, defined as the union $(A_1 \times \{1\}) \cup (A_2 \times \{2\})$ with the insertion functions

$$\iota_{A_1} : A_1 \to A_1 \uplus A_2 \leftarrow A_2 : \iota_{A_2}$$

where $\iota_{A_1}$ is defined by $a_1 \mapsto (a_1, 1)$ for all $a_1 \in A_1$, and $\iota_{A_2}$ by $a_2 \mapsto (a_2, 2)$ for all $a_2 \in A_2$.

- Let preorder $(X, \leq)$ have top and bottom elements and all finite meets and joins (least upper bounds). Then the top of $X$ is terminal, the bottom of $X$ initial, and finite meets and joins are finite products and coproducts respectively.
Examples of (Co)Products

- Given \((X, \leq)\) and \((Y, \leq)\) in \(\text{ParSet}\), the binary product is the cartesian product \(X \times Y\) in \(\text{Set}\), with the pointwise order \((x, y) \leq (x', y')\) iff \(x \leq x'\) and \(y \leq y'\), together with the (monotone) set-theoretic projection functions. The binary coproduct is \(X \uplus Y\), with \((z, \delta) \leq (z', \delta')\) iff \(\delta = \delta'\) \((\delta, \delta' \in \{1, 2\})\), and \(z \leq z'\) (either in \(X\) or in \(Y\)).

- An initial object \(0\) in a category \(\mathcal{C}\) has the property that there is a unique morphism \(!_A : 0 \rightarrow A\) for every \(A \in \text{ob}\ \mathcal{C}\). It is the finite coproduct of an empty family of morphisms (check this!). Such a \(0\) may not exist, but is unique if it does.
Useful “Fact” for (Co)Products

- Suppose that we have \((f_i: C \to A_i \mid i \in \{1, 2\})\) and \(\theta: C \to A_1 \times A_2\). In order to prove that \(\theta = \langle f_1, f_2 \rangle\) it is sufficient to show that \(\pi_{A_i} \circ \theta = f_i\) for each \(i\).

- Suppose that we have \((f_i: A_i \to C \mid i \in \{1, 2\})\) and \(\theta: A_1 + A_2 \to C\). In order to prove that \(\theta = [f_1, f_2]\) it is sufficient to show that \(\theta \circ \iota_{A_i} = f_i\) for each \(i\).

Note: this “fact” is simply a consequence of uniqueness of mediating morphisms. It is crucial to the proof that (co)products are unique up to isomorphism, where both \(\phi \circ \psi\) and \(\text{id}\) (from an earlier slide) are shown to be mediating, and hence equal.
Notation for Finite (Co)Products

Suppose that \( f_1: A_1 \rightarrow B_1 \) and \( f_2: A_2 \rightarrow B_2 \). Then

\[
f_1 \times f_2 \overset{\text{def}}{=} \langle f_1 \circ \pi_{A_1}, f_2 \circ \pi_{A_2} \rangle : A_1 \times A_2 \rightarrow B_1 \times B_2
\]

\[
f_1 + f_2 \overset{\text{def}}{=} [\iota_{B_1} \circ f_1, \iota_{B_2} \circ f_2] : A_1 + A_2 \rightarrow B_1 + B_2
\]

and hence one can prove that

\[
\pi_{B_i} \circ (f_1 \times f_2) = f_i \circ \pi_{A_i}
\]

\[
(f_1 + f_2) \circ \iota_{A_i} = \iota_{B_i} \circ f_i
\]

This notation is easily extended to finite families \((A_i \mid i \in \{1, \ldots, n\})\) and \((B_i \mid i \in \{1, \ldots, n\})\) \ldots or indeed infinite families \((A_i \mid i \in I)\) and \((B_i \mid i \in I)\) where \(I\) is any set.
More Examples of (Co)Products

▶ Suppose that \( C \) has binary (co)products. The functors \( B \times (-), B + (-): C \to C \) are defined by

\[
f: A \to A' \mapsto id_B \times f: B \times A \to B \times A'
\]

\[
f: A \to A' \mapsto id_B + f: B + A \to B + A'
\]

Note that it is common to write \( f \times B \) instead of \( f \times id_B \); ditto \( + \).
More Examples of (Co)Products

Suppose that $F_1$ and $F_2$ are objects (that is, functors) of $\mathcal{D}^C$ and that $\mathcal{D}$ has finite (co)products. Then both $F_1 \times F_2$ and $F_1 + F_2$ exist and are defined pointwise. For products this means

$$(F_1 \times F_2)(\xi) \overset{\text{def}}{=} F_1\xi \times F_2\xi$$

where $\xi$ is either an object or morphism of $\mathcal{C}$. The projections $\pi^i : F_1 \times F_2 \to F_i$ are defined with pointwise components $\pi^i_A : F_1A \times F_2A \to F_iA$. These projections $\pi^i$ are indeed natural transformations.
An algebra for $F : C \to C$ is a morphism $\sigma : FI \to I$ in $C$. The algebra is initial if given any $f : FX \to X$ there is a homomorphism $\bar{f} : I \to X$, meaning that ($f$ is a morphism and)

$$
\begin{array}{ccc}
   FI & \xrightarrow{\sigma} & I \\
   Ff & \downarrow & \downarrow \bar{f} \\
   FX & \xrightarrow{f} & X
\end{array}
$$

and such a $\bar{f}$ is unique.

There is a category $C^F$ of algebras and algebra homomorphisms (details omitted) in which initial algebras are initial objects.
Algebras for $F: C \to C$

- $1 + (-): Set \to Set$ has an initial algebra

$$[z, s]: 1 + \mathbb{N} \to \mathbb{N}$$

where $z: 1 \to \mathbb{N}$ maps $\ast$ to 0 and $s: \mathbb{N} \to \mathbb{N}$ adds 1. If $f: 1 + X \to X$

the function $\overline{f}: \mathbb{N} \to X$ is uniquely defined by

$$\overline{f}(0) \overset{\text{def}}{=} \hat{x}(\ast) \overset{\text{def}}{=} x$$

$$\overline{f}(n + 1) \overset{\text{def}}{=} \phi^{n+1}(x) = \phi(\overline{f}(n))$$

where $\hat{x} \overset{\text{def}}{=} f \circ \iota_1: 1 \to 1 + X$ and

$\phi \overset{\text{def}}{=} f \circ \iota_X: X \to 1 + X$ (and hence $f = [\hat{x}, \phi]$).
EXAMPLE: the function \((+n): \mathbb{N} \rightarrow \mathbb{N}\) which adds \(n\), for any \(n \in \mathbb{N}\), is definable as \([\hat{n}, s]\) where

\[
1 + \mathbb{N} \xrightarrow{[\hat{n}, s]} \mathbb{N}
\]

and also \((\ast n) \overset{\text{def}}{=} [z, (+n)]: \mathbb{N} \rightarrow \mathbb{N}\).

A monoid \((M, b, e)\) is an algebra

\[
1 + (M \times M) \xrightarrow{[\hat{e}, b]} M
\]

plus the relevant equations.
Case Study: (Haskell) Algebraic Datatypes

We shall

- Define a Haskell (recursive) datatype grammar.
- Show that any datatype declaration \( D \) gives rise to a functor \( F \equiv F_D : \text{Set} \to \text{Set} \).
- Demonstrate that \( D \) can be modelled by an initial algebra \( \sigma : FI \to I \), where \( I \) is the set \( \text{Exp}_D \) of expressions of type \( D \) (up to isomorphism).

Later on we will

- Show that the functor \( F \) preserves colimits of diagrams of the form \( D : \omega \to \text{Set} \), and such colimits exist . . .
- and (hence) that \( F \) must have an initial algebra for purely categorical reasons.
A Recursive Datatype

- A set of **type patterns** $T$ is defined by
  $$T ::= D \mid \text{Unit} \mid \text{Int} \mid T \times T$$
- A **datatype** is specified by the statement
  $$D \equiv K_1 T_1 \mid \ldots \mid K_m T_m$$
- A collection of **type assignments** is defined inductively by the following rules

  $$\frac{}{(\_ :: \text{Unit})} \quad \frac{z \in \mathbb{Z}}{z :: \text{Int}} \quad \frac{E :: T_i}{K_i E :: D} \quad \frac{E_1 :: T_1 \quad E_2 :: T_2}{(E_1, E_2) :: T_1 \times T_2}$$

  and $Exp_T \overset{\text{def}}{=} \{ E \mid E :: T \}$. 
Defining $F$ from $D$

- The functor $F$ is defined (as a coproduct in $\text{Set}^{\text{Set}}$) by

$$F \overset{\text{def}}{=} F_{T_1} + \ldots + F_{T_m}$$

where each $F_{T_i} : \text{Set} \to \text{Set}$.

- Functors $F_T : \text{Set} \to \text{Set}$ are defined by recursion on the structure of $T$ by setting

  - $F_D \overset{\text{def}}{=} \text{id}_{\text{Set}}$
  - $F_{\text{Unit}}(g : U \to V) \overset{\text{def}}{=} \text{id}_1 : 1 \to 1$ where $1$ is terminal in $\text{Set}$
  - $F_{\text{Int}}(g : U \to V) \overset{\text{def}}{=} \text{id}_{\mathbb{Z}} : \mathbb{Z} \to \mathbb{Z}$
  - $F_{T_1 \times T_2} \overset{\text{def}}{=} F_{T_1} \times F_{T_2}$
Defining An Initial Algebra \( \sigma : F \mathcal{I} \rightarrow \mathcal{I} \)

- We set \( \mathcal{I} \defs \text{Exp}_D \) and we define

\[
\sigma \defs [\hat{K}_1 \circ \sigma_{T_1} \ldots \hat{K}_m \circ \sigma_{T_m}] : F \mathcal{I} \defs F_{T_1} \mathcal{I} + \ldots + F_{T_m} \mathcal{I} \rightarrow \mathcal{I}
\]

where the function \( \hat{K}_i : \text{Exp}_{T_i} \rightarrow \mathcal{I} \) applies the constructor and we define functions \( \sigma_{T} : F_{T} \mathcal{I} \rightarrow \text{Exp}_{T} \) by recursion over \( T \) as follows

- \( \sigma_{D}(E \in \mathcal{I}) \defs E \in \text{Exp}_D \)
- \( \sigma_{\text{Unit}}(*) \in 1 \defs () \in \text{Exp}_{\text{Unit}} \)
- \( \sigma_{\text{Int}}(z \in \mathbb{Z}) \defs z \in \text{Exp}_{\text{Int}} \)
- \( \sigma_{T_1 \times T_2}((e_1, e_2) \in F_{T_1} \mathcal{I} \times F_{T_2} \mathcal{I}) \defs (\sigma_{T_1}(e_1), \sigma_{T_2}(e_2)) \in \text{Exp}_{T_1 \times T_2} \)
- It may be useful to note that \( \sigma(\iota_i(e_i \in F_{T_i} \mathcal{I})) = K_i \sigma_{T_i}(e_i) \).

MGS 2015, 7-11 April, University of Sheffield, UK
Suppose that $f : FX \to X$ in $\textbf{Set}$. We have to prove that there is a unique $\bar{f}$ such that

$$FT_1 I + \ldots + FT_m I = F I \xrightarrow{\sigma} I$$

$$Ff$$

$$FT_1 X + \ldots + FT_m X = FX \xrightarrow{f} X$$
Verifying Initiality

- Note $\bar{f}: \text{Exp}_D \rightarrow F_D X$; we will define $\bar{f} \overset{\text{def}}{=} \theta_D$ and functions $\theta_T: \text{Exp}_T \rightarrow F_T X$

by recursion on $T$:

- $\theta_D(K_i E_i \in \text{Exp}_D) \overset{\text{def}}{=} f(\iota_i(\theta_{T_i}(E_i))) \in X$.
- $\theta_{\text{Unit}}(() \in \text{Exp}_{\text{Unit}}) \overset{\text{def}}{=} * \in 1$.
- $\theta_{\text{Int}}(z \in \text{Exp}_{\text{Int}}) \overset{\text{def}}{=} z \in \mathbb{Z}$.
- $\theta_{T_1 \times T_2}((E_1, E_2) \in \text{Exp}_{T_1 \times T_2}) \overset{\text{def}}{=} (\theta_{T_1}(E_1), \theta_{T_2}(E_2)) \in F_{T_1} 1 \times F_{T_2} 1$. 
Verifying Initiality

> Observe that for any $T$ we have $\theta_T \circ \sigma_T = F_T \theta_D$, which follows from an easy induction.

Note that by universality of coproducts $\bar{f} \circ \sigma = f \circ F \bar{f}$ iff

$$\bar{f} \circ \sigma \circ \iota_i = f \circ F \bar{f} \circ \iota_i$$

Then for any $e_i \in F_{T_i} I$

$$\left( \theta_D \circ \sigma \circ \iota_i \right)(e_i) \quad = \quad \theta_D(\epsilon_i(\sigma_{T_i}(e_i)))$$

$$\overset{\text{def}}{=} \theta_D \quad f(\iota_i(\theta_{T_i}(\sigma_{T_i}(e_i))))$$

$$\overset{\text{def}}{=} \quad f(\iota_i((F_{T_i} \theta_D)(e_i)))$$

$$\overset{\text{def}}{=} \quad f((F_{T_i} \theta_D + \ldots + F_{T_m} \theta_D)(\iota_i(e_i)))$$

$$\overset{\text{def}}{=} F \quad (f \circ F \theta_D \circ \iota_i)(e_i)$$

The steps follow by: definition of $\sigma$; definition of $\theta_D$; the observation; properties of $+$; the definition of $F$. 
Verifying Initiality

Observe that for any $T$ we have $\theta_T \circ \sigma_T = F_T \theta_D$, which follows from an easy induction.

Note that by universality of coproducts $\bar{f} \circ \sigma = f \circ F\bar{f}$ iff

$$\bar{f} \circ \sigma \circ \iota_i = f \circ F\bar{f} \circ \iota_i$$

Then for any $e_i \in F_{T_i}1$

$$(\theta_D \circ \sigma \circ \iota_i)(e_i) \equiv \theta_D(K_i \sigma_{T_i}(e_i))$$

$\overset{\text{def}}{=} \theta_D f(\iota_i(\theta_{T_i}(\sigma_{T_i}(e_i))))$

$\overset{\text{def}}{=} f(\iota_i((F_{T_i}\theta_D)(e_i)))$

$\overset{\text{def}}{=} f(((F_{T_1}\theta_D + \ldots + F_{T_m}\theta_D)(\iota_i(e_i))))$

$\overset{\text{def}}{=} \theta_D f(\iota_i(F \theta_D)(e_i))$

The steps follow by: definition of $\sigma$; definition of $\theta_D$; the observation; properties of $\tau$; the definition of $F$. 

MGS 2015, 7-11 April, University of Sheffield, UK
Verifying Initiality

- **Observe** that for any $T$ we have $\theta_T \circ \sigma_T = F_T \theta_D$, which follows from an easy induction.

Note that by universality of coproducts $\bar{f} \circ \sigma = f \circ F\bar{f}$ iff

$$\bar{f} \circ \sigma \circ \iota_i = f \circ F\bar{f} \circ \iota_i$$

Then for any $e_i \in F_{T_i} l$

$$(\theta_D \circ \sigma \circ \iota_i)(e_i) = \theta_D(K_i \sigma_{T_i}(e_i))$$
$$\overset{\text{def}}{=} \theta_D f(\iota_i(\theta_T(\sigma_{T_i}(e_i))))$$
$$= f(\iota_i((F_{T_i} \theta_D)(e_i)))$$
$$= f((F_{T_i} \theta_D + \ldots + F_{T_m} \theta_D)(\iota_i(e_i)))$$
$$\overset{\text{def}}{=} F(f \circ F\theta_D \circ \iota_i)(e_i)$$

The steps follow by: definition of $\sigma$; definition of $\theta_D$; the observation; properties of $\oplus$; the definition of $F$. 

MGS 2015, 7-11 April, University of Sheffield, UK
Verifying Initiality

- Observe that for any $T$ we have $\theta_T \circ \sigma_T = F_T \theta_D$, which follows from an easy induction.

Note that by universality of coproducts $\bar{f} \circ \sigma = f \circ \bar{F} \bar{f}$ iff

$$\bar{f} \circ \sigma \circ \iota_i = f \circ \bar{F} \bar{f} \circ \iota_i$$

Then for any $e_i \in F_T i$

$$(\theta_D \circ \sigma \circ \iota_i)(e_i) = \theta_D(K_i \sigma_T (e_i))$$

$\overset{\text{def}}{=} \theta_D f(\iota_i(\theta_T (\sigma_T (e_i))))$

$$= f(\iota_i((F_T \theta_D)(e_i)))$$

$$= f((F_{T_1} \theta_D + \ldots + F_{T_m} \theta_D)(\iota_i(e_i)))$$

$\overset{\text{def}}{=} F (f \circ F \theta_D \circ \iota_i)(e_i)$

The steps follow by: definition of $\sigma$; definition of $\theta_D$; the observation; properties of $+$; the definition of $F$. 
Observe that for any $T$ we have $\theta_T \circ \sigma_T = F_T \theta_D$, which follows from an easy induction.

Note that by universality of coproducts $\bar{f} \circ \sigma = f \circ F \bar{f}$ iff

$$\bar{f} \circ \sigma \circ \iota_i = f \circ F \bar{f} \circ \iota_i$$

Then for any $e_i \in F_{T_i} I$

$$(\theta_D \circ \sigma \circ \iota_i)(e_i) \overset{\text{def}}{=} \theta_D(K_i \sigma_{T_i}(e_i)) \overset{\theta_D}{=} f(\iota_i(\theta_{T_i}(\sigma_{T_i}(e_i)))) = f(\iota_i((F_{T_i} \theta_D)(e_i))) = f(((F_{T_1} \theta_D + \ldots + F_{T_m} \theta_D)(\iota_i(e_i)))) \overset{\text{def}}{=} (f \circ F \theta_D \circ \iota_i)(e_i)$$

The steps follow by: definition of $\sigma$; definition of $\theta_D$; the observation; properties of $+$; the definition of $F$. 

MGS 2015, 7-11 April, University of Sheffield, UK
Adjunctions between Preorders

- A pair of monotone functions

\[
X \leftrightarrow \frac{l}{r} \rightarrow Y
\]

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

\[
\begin{array}{ccc}
X & \overset{\Delta}{\leftrightarrow} & X \times X \\
\lor & \quad & \land
\end{array}
\]

just in case $X$ has all binary meets and joins: for any $l \in X$,

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

- This structure corresponds to $X$ having binary products and coproducts.
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 $C$, that is, between $C(LA, B)$ and $D(A, RB)$,

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

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

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

$$\psi \circ f \circ L\phi = R\psi \circ \bar{f} \circ \phi \quad \text{and/or} \quad (R\psi \circ g \circ \phi)^\wedge = \psi \circ \hat{g} \circ L\phi.$$
Examples of Adjunctions

The **forgetful** functor $U : \text{Mon} \to \text{Set}$ taking a monoid to its underlying set, and the functor $\text{list}(\_) : \text{Set} \to \text{Mon}$ taking a set to finite lists over the set, are adjoints:

$$\text{list}(\_ \ ) \dashv U$$

So there is a natural bijection between $\text{Mon}(\text{list}(A), M)$ and $\text{Set}(A, UM)$

$$f : \text{list}(A) \to M$$
$$\bar{f} : A \to UM$$

$$g : A \to UM$$
$$\hat{g} : \text{list}(A) \to M$$
Examples of Adjunctions

This is given by

\[ g : A \to UM \implies \]
\[ \hat{g} : \text{list}(A) \quad [a_1, \ldots, a_n] \mapsto g(a_1) \ldots g(a_n) \quad \to M, \]
\[
\begin{array}{c}
[] \mapsto e
\end{array}
\]

and

\[ f : \text{list}(A) \to M \implies \bar{f} : A \quad a \mapsto f([a]) \quad \to UM. \]

Note that

\[ \hat{f}[a_1, \ldots, a_n] = \bar{f}(a_1) \ldots \bar{f}(a_n) = f([a_1]) \ldots f([a_n]) = f([a_1]++\ldots++[a_n]) \]

It is an exercise to verify that \( \hat{g} = g \) and that this bijection is natural.
Examples of Adjunctions

For a fixed set $A$, the functor $(-) \times B : \textbf{Set} \to \textbf{Set}$ has a right adjoint $B \Rightarrow (-) : \textbf{Set} \to \textbf{Set}$. If $c : C \to C'$ then

$B \Rightarrow c : B \Rightarrow C \longrightarrow B \Rightarrow C'$ where $(B \Rightarrow c)(\theta) \overset{\text{def}}{=} c \circ \theta$

$$f : A \times B \to C$$

$$\bar{f} \overset{\text{def}}{=} \lambda a.\lambda b. f(a,b) : A \to B \Rightarrow C$$

$$g : A \to B \Rightarrow C$$

$$\hat{g} \overset{\text{def}}{=} \lambda(a,b).g(a)(b) : A \times B \to C$$

It is immediate that we have a bijection; naturality is an exercise. Having products and such a “function structure” is known as cartesian closure.
Examples of Adjunctions

- The **diagonal functor** \( \Delta : \text{Set} \rightarrow \text{Set} \times \text{Set} \) taking a function \( f : A \rightarrow B \) to \((f, f) : (A, A) \rightarrow (B, B)\) has right and left adjoints \( \Pi \) and \( \Sigma \) taking any morphism \((f_1, f_2) : (A_1, A_2) \rightarrow (B_1, B_2)\) of \( \text{Set} \times \text{Set} \) to
  
  \[
  f_1 \times f_2 \overset{\text{def}}{=} \langle f_1 \circ \pi_{A_1}, f_2 \circ \pi_{A_2} \rangle : A_1 \times A_2 \rightarrow B_1 \times B_2 \quad \text{and} \quad f_1 + f_2 \overset{\text{def}}{=} [\iota_{B_1} \circ f_1, \iota_{B_2} \circ f_2] : A_1 + A_2 \rightarrow B_1 + B_2
  \]
  respectively, where the bijection for \( \Pi \) is

  \[
  (f, g) \quad \overset{\text{def}}{=} \quad \langle \pi_A \circ m, \pi_B \circ m \rangle : \Delta C \longrightarrow (A, B)
  \]

  \[
  \hat{m} \overset{\text{def}}{=} (\pi_A \circ m, \pi_B \circ m) : \Delta C \longrightarrow (A, B)
  \]

- If we replace \( \text{Set} \) by any category \( \mathcal{C} \) with (co)products, defining \( \Delta : \mathcal{C} \rightarrow \mathcal{C} \times \mathcal{C} \) analogously, everything still works.
Let $\mathcal{C}$ be a category with finite products. Existence of a right adjoint $R$ to the functor $(−) \times B : \mathcal{C} \to \mathcal{C}$ for each object $B$ of $\mathcal{C}$, is equivalent to $\mathcal{C}$ being cartesian closed.
Let $\mathcal{C}$ be a category with finite products. Existence of a right adjoint $R$ to the functor $(-) \times B: \mathcal{C} \to \mathcal{C}$ for each object $B$ of $\mathcal{C}$, is equivalent to $\mathcal{C}$ being cartesian closed.

$(\Rightarrow)$ Given an object $B$ of $\mathcal{C}$ set $B \Rightarrow C \overset{\text{def}}{=} R(C)$ for any object $C$ of $\mathcal{C}$. 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 $(\widehat{id}_{B \Rightarrow C})$ of the identity $id_{B \Rightarrow C}: (B \Rightarrow C) \to (B \Rightarrow C)$. 

\[\]
CCC via Adjunctions

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.

Next, we need to show that $ev \circ (\lambda(f) \times id_B) = f$. This follows directly from the naturality of the adjunction; we consider naturality in $A$ and $C$ at the morphisms $\lambda(f) : A \to (B \Rightarrow C)$ and $id_C : C \to C$:

\[
\begin{array}{ccc}
\text{id}_B \Rightarrow C & \xrightarrow{\lambda(f)} & ev \\
\downarrow & & \downarrow \\
R(id_C) \circ \text{id}_B \Rightarrow C \circ \lambda(f) & \xrightarrow{\lambda(f)} & \text{id}_C \circ ev \circ (\lambda(f) \times id_B)
\end{array}
\]

We let the reader show that $\lambda(f)$ is the unique morphism satisfying the latter equation.
Conversely, let $B$ be an object of $C$. We define a right adjoint to $(-) \times B$ denoted by $B \Rightarrow (-)$, by setting

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

for each morphism $c : C \to C'$ of $C$ (this matches our earlier definition – check). We define a bijection by declaring the mate of $f : A \times B \to C$ to be $\lambda(f) : A \to (B \Rightarrow C)$ and the mate of $g : A \to (B \Rightarrow C)$ to be

$$\hat{g} \overset{\text{def}}{=} ev \circ (g \times id_B) : A \times B \to C.$$
It remains to verify that we have defined a bijection which is natural in the required sense. We only check one part of naturality. Let $a: A' \to A$ and $c: C \to C'$ be morphisms of $C$. Then

$$ev \circ ((\lambda(c \circ ev) \circ \lambda(f) \circ a) \times id) =$$

$$ev \circ (\lambda(c \circ ev) \times id) \circ (\lambda(f) \times id) \circ (a \times id) =$$

$$c \circ ev \circ (\lambda(f) \times id) \circ (a \times id) =$$

$$c \circ f \circ (a \times id)$$

implying that $\lambda(c \circ f \circ (a \times id)) = (B \Rightarrow c) \circ \lambda(f) \circ a$ since $C$ is a CCC.

The steps above are: *categorical properties of $\times$*; cartesian closure of $C$; cartesian closure again.
It remains to verify that we have defined a bijection which is natural in the required sense. We only check one part of naturality. Let \( a: A' \to A \) and \( c: C \to C' \) be morphisms of \( C \). Then

\[
ev \circ \left( (\lambda (c \circ ev) \circ \lambda (f) \circ a) \times id \right) =
\]

\[
ev \circ (\lambda (c \circ ev) \times id) \circ (\lambda (f) \times id) \circ (a \times id) =
\]

\[
c \circ ev \circ (\lambda (f) \times id) \circ (a \times id) =
\]

\[
c \circ f \circ (a \times id)
\]

implying that \( \lambda (c \circ f \circ (a \times id)) = (B \Rightarrow c) \circ \lambda (f) \circ a \) since \( C \) is a CCC.

The steps above are: categorical properties of \( \times \); cartesian closure of \( C \); cartesian closure again.
It remains to verify that we have defined a bijection which is natural in the required sense. We only check one part of naturality. Let \( a: A' \to A \) and \( c: C \to C' \) be morphisms of \( C \). Then

\[
e v \circ ((\lambda (c \circ e v) \circ \lambda (f) \circ a) \times id) = \]

\[
e v \circ (\lambda (c \circ e v) \times id) \circ (\lambda (f) \times id) \circ (a \times id) = \]

\[
c \circ e v \circ (\lambda (f) \times id) \circ (a \times id) = \]

\[
c \circ f \circ (a \times id) \]

implying that \( \lambda (c \circ f \circ (a \times id)) = (B \Rightarrow c) \circ \lambda (f) \circ a \) since \( C \) is a CCC.

The steps above are: categorical properties of \( \times \); cartesian closure of \( C \); cartesian closure again.
It remains to verify that we have defined a bijection which is natural in the required sense. We only check one part of naturality. Let \( a: A' \to A \) and \( c: C \to C' \) be morphisms of \( C \). Then

\[
ev \circ ( (\lambda (c \circ ev) \circ \lambda(f) \circ a) \times id) =
\]

\[
ev \circ (\lambda(c \circ ev) \times id) \circ (\lambda(f) \times id) \circ (a \times id) =
\]

\[
c \circ ev \circ (\lambda(f) \times id) \circ (a \times id) =
\]

\[
c \circ f \circ (a \times id)
\]

implying that \( \lambda(c \circ f \circ (a \times id)) = (B \Rightarrow c) \circ \lambda(f) \circ a \) since \( C \) is a CCC.

The steps above are: categorical properties of \( \times \); cartesian closure of \( C \); cartesian closure again.
Colimits

Given a diagram $D: \mathbb{I} \rightarrow \mathcal{C}$, a colimit for $D$ is given by an object $\text{col}_I DI$ of $\mathcal{C}$ together with a family of morphisms $(\iota_I: DI \rightarrow \text{col}_I DI | I \in \mathbb{I})$ such that for any $\alpha: I \rightarrow J$ in $\mathbb{I}$ we have $\iota_J \circ D\alpha = \iota_I$. This data satisfies: given any family $(h_I: DI \rightarrow C | I \in \mathbb{I})$ such that $h_J \circ D\alpha = h_I$, there is a unique morphism $\phi: \text{col}_I DI \rightarrow C$ satisfying $\phi \circ \iota_I = h_I$ for each object $I$ of $\mathbb{I}$ (and hence $\phi = [h_I | I \in \mathbb{I}]$)

Binary coproducts arise from the discrete category $\mathbb{I} \overset{\text{def}}{=} \{1, 2\}$.
Let $D : \omega \to C$; suppose that $i \leq i + 1$ is a typical morphism in $\omega$. Then a colimit diagram, if it exists, can be taken as

$$\cdots D(i) \xrightarrow{D(\leq i)} D(i + 1) \cdots$$

where for any given functions $h_i : D(i) \to C$ commuting with the functions $D(\leq i)$, a unique such $\phi$ exists. This fact follows, since $h_j \circ D(\leq i) = h_i$ for a general morphism $\leq^i_j$ (where $i \leq j$ in $\omega$) is immediate.
It is a fact that the category of sets, \( \textbf{Set} \), has all (small) colimits.

It is a fact that a colimit for \( \Delta : \omega \times \omega \to \mathcal{C} \) exists if and only if a colimit for \( \Delta' : \omega \to \mathcal{C} \) where \( \Delta'(i \in \omega) \overset{\text{def}}{=} \Delta(i, i) \) exists, and when they (both) exist they are isomorphic, that is

\[
\text{col}_k \Delta'(k) \cong \text{col}_{(i,j)} \Delta(i, j)
\]

Further (exercise: define the diagrams that give rise to the colimits below . . .)

\[
\text{col}_i(\text{col}_j \Delta(i, j)) \cong \text{col}_j(\text{col}_i \Delta(j, i))
\]

and all of the above colimits are isomorphic.
Left Adjointes Preserve Colimits

Let \( D : \mathbb{I} \rightarrow \mathcal{C} \), and \( L : \mathcal{C} \rightarrow \mathcal{D} \) and \( L \dashv R \) for some \( R \). Then

\[
L(\text{col}_I DI) \cong \text{col}_I LDI
\]

and is witnessed by \([L(\iota_{DI}) | I \in \mathbb{I}] : \text{col}_I LDI \rightarrow L(\text{col}_I DI)\). It suffices to show that \( L(\text{col}_I DI) \) is a colimit for \( LD : \mathbb{I} \rightarrow \mathcal{D} \).
Suppose that $h_I = h_J \circ LD\alpha$. We need to show there is a unique $\phi$ as above.
Left Adjointts Preserve Colimits

But

\[ h_I = h_J \circ LD\alpha \implies \overline{h_I} = \overline{h_J} \circ LD\alpha = \overline{h_J} \circ D\alpha \]

where the final equality follows by naturality.
Therefore there is $\rho$ with $\rho \circ \iota_{DI} = \overline{h}_I$. Define

$$\phi \overset{\text{def}}{=} \hat{\rho} : L(\text{col}_1 DI) \to X$$
Left Adjoinst Preserve Colimits

\[
L D I \xrightarrow{LD\alpha} L D J
\]

\[
\begin{array}{ccc}
L D I & \xrightarrow{L D I} & L D J \\
\downarrow h_I & & \downarrow h_J \\
L(\iota DI) & \xrightarrow{L(col_1 DI)} & \check{\rho} \circ L(\iota DI) = \rho \circ \iota DI = \hat{h}_I = h_I
\end{array}
\]

Hence, again using naturality,

\[
\phi \circ L(\iota DI) \overset{\text{def}}{=} \check{\rho} \circ L(\iota DI) = \rho \circ \iota DI = \hat{h}_I = h_I
\]
Existence of Initial Algebras

Suppose that $F$ preserves colimits of the form $D : \omega \to C$ and that $C$ has an initial object $0$. Define

$D(i \leq i + 1) \stackrel{\text{def}}{=} F^i !_X : F^i 0 \to F^{i+1} 0$ for $i \in \omega$. Then

$I \stackrel{\text{def}}{=} \text{col } i D i$ (if it exists) is an initial algebra for $F$.

Since $F$ preserves colimits and $I \stackrel{\text{def}}{=} \text{col } i D i$ we can define $\sigma : F I \to I$

\[
\begin{align*}
FF^i 0 & \xrightarrow{FF^i !_X} FF^{i+1} 0 \\
I & \xrightarrow{\iota_{i+1}} F I & F I & \xrightarrow{F \iota_{i+1}} I
\end{align*}
\]

where $\sigma \circ F \iota_i = \iota_{i+1}$.
Existence of Initial Algebras

Let $f : FX \to X$. Define $f_0 \overset{\text{def}}{=} !_X : 0 \to X$ and $f_{i+1} \overset{\text{def}}{=} f \circ Ff_i$. Certainly $f_1 \circ F^0!_X \equiv f_1 \circ !_X = f_0$ and for $i \geq 1$ we have inductively

$$f_{i+1} \circ F^i!_X \overset{\text{def}}{=} f \circ Ff_i \circ F^i!_X = f \circ F(f_i \circ F^{i-1}!_X) = f \circ Ff_{i-1} \overset{\text{def}}{=} f_i$$

and hence $\overline{f}$ exists where $\overline{f} \circ \iota_i = f_i$.

![Diagram](attachment:image.png)
Existence of Initial Algebras

We now have $\sigma \circ F \iota_i = \iota_{i+1}$; and $f_{i+1} \overset{\text{def}}{=} f \circ F f_i$ (which implied $f_{i+1} = f_{i+2} \circ F^{i+1}!_X$) yielding $\bar{f} \circ \iota_i = f_i$

The equality follows since

$$\bar{f} \circ \sigma \circ F \iota_i = f_{i+1} \quad f \circ F \bar{f} \circ F \iota_i = f \circ F (\bar{f} \circ \iota_i) = f \circ F f_i = f_{i+1}$$
Suppose that a functor \( F : \mathbf{Set} \to \mathbf{Set} \) is defined by a grammar
\[ F ::= P \mid F \times F \mid F + F \]
where \( P \) preserves colimits of diagrams \( D : \omega \to \mathbf{Set} \). Then so too does \( F \). This follows by induction.
Suppose that \( F, G \) preserve such colimits.

\[
(F \times G)(\text{col}_i D_i) \overset{\text{def}}{=} (\text{col}_i F D_i) \times (\text{col}_i G D_i) \\
(\text{col}_j F D_j) \times (\text{col}_i G D_i) \\
\text{col}_i ((\text{col}_j F D_j) \times D G i) \\
\text{col}_i (\text{col}_j (D F i \times D G j)) \\
\text{col}_k (D F k \times D G k)
\]

The steps follow by: induction on \( F \) and \( G \); \( (\text{col}_j F D_j) \times (\text{col}_i G D_i) \) has a right adjoint so preserves colimits; \( (\text{col}_i G D_i) \times (\text{col}_j F D_j) \) also has a right adjoint; the earlier fact that a colimit for \( \Delta : \omega \times \omega \to \mathbf{C} \) and \( \Delta' : \omega \to \mathbf{C} \) where \( \Delta'(k) \overset{\text{def}}{=} \Delta(k, k) \) are isomorphic.
Suppose that a functor $F : \textbf{Set} \rightarrow \textbf{Set}$ is defined by a grammar $F ::= P \mid F \times F \mid F + F$ where $P$ preserves colimits of diagrams $D : \omega \rightarrow \textbf{Set}$. Then so too does $F$. This follows by induction. Suppose that $F, G$ preserve such colimits.

$$(F \times G)(\text{col}_iDi) \overset{\text{def}}{=} (F\text{col}_iDi) \times (G\text{col}_iDi)$$

$$\overset{\text{IR IR IR IR}}{=} (\text{col}_jFDj) \times (\text{col}_iGDj)$$

$$\overset{\text{col}_i((\text{col}_jDFj) \times DGi)}{=} \text{col}_i(\text{col}_j(DFi \times DGj))$$

$$\overset{\text{IR IR}}{=} \text{col}_k(DFk \times DGk)$$

The steps follow by: induction on $F$ and $G$; $(\text{col}_jFDj) \times (-)$ has a right adjoint so preserves colimits; $(-) \times DGi$ also has a right adjoint; the earlier fact that a colimit for $\Delta : \omega \times \omega \rightarrow \mathcal{C}$ and $\Delta' : \omega \rightarrow \mathcal{C}$ where $\Delta'(k) \overset{\text{def}}{=} \Delta(k, k)$ are isomorphic.
Suppose that a functor \( F : \mathbf{Set} \to \mathbf{Set} \) is defined by a grammar 
\[
F ::= P \mid F \times F \mid F + F
\]
where \( P \) preserves colimits of diagrams \( D : \omega \to \mathbf{Set} \). Then so too does \( F \). This follows by induction.

Suppose that \( F, G \) preserve such colimits.

\[
(F \times G)(\text{col}_iD_i) \overset{\text{def}}{=} (F\text{col}_iD_i) \times (G\text{col}_iD_i)
\]
\[
(\text{col}_jFD_j) \times (\text{col}_iGD_i)
\]
\[
\text{col}_i((\text{col}_jDF_j) \times DG_i)
\]
\[
\text{col}_i(\text{col}_j(DF_i \times DG_j))
\]
\[
\text{col}_k(DF_k \times DGk)
\]

The steps follow by: induction on \( F \) and \( G \); \( (\text{col}_jFD_j) \times (\vdash) \) has a right adjoint so preserves colimits; \( (\vdash) \times DG_i \) also has a right adjoint; the earlier fact that a colimit for \( \Delta : \omega \times \omega \to \mathbf{C} \) and \( \Delta' : \omega \to \mathbf{C} \) where \( \Delta'(k) \overset{\text{def}}{=} \Delta(k, k) \) are isomorphic.
Suppose that a functor $F : \textbf{Set} \rightarrow \textbf{Set}$ is defined by a grammar $F ::= P \mid F \times F \mid F + F$ where $P$ preserves colimits of diagrams $D : \omega \rightarrow \textbf{Set}$. Then so too does $F$. This follows by induction. Suppose that $F, G$ preserve such colimits.

\[(F \times G)(\text{col}_iD_i) \overset{\text{def}}{=} (F\text{col}_iD_i) \times (G\text{col}_iD_i)\]
\[= (\text{col}_jFD_j) \times (\text{col}_iGD_i)\]
\[= \text{col}_i((\text{col}_jDFj) \times DG_i)\]
\[= \text{col}_i(\text{col}_j(DFi \times DGj))\]
\[= \text{col}_k(DFk \times DGk)\]

The steps follow by: induction on $F$ and $G$; $(\text{col}_jFD_j) \times (\_)$ has a right adjoint so preserves colimits; $(\_ \times DGi)$ also has a right adjoint; the earlier fact that a colimit for $\Delta : \omega \times \omega \rightarrow C$ and $\Delta' : \omega \rightarrow C$ where $\Delta'(k) \overset{\text{def}}{=} \Delta(k, k)$ are isomorphic.
Suppose that a functor \( F : \text{Set} \to \text{Set} \) is defined by a grammar
\[
F ::= P \mid F \times F \mid F + F
\]
where \( P \) preserves colimits of diagrams \( D : \omega \to \text{Set} \). Then so too does \( F \). This follows by induction.

Suppose that \( F, G \) preserve such colimits.

\[
(F + G)(\text{col}_i Di) \overset{\text{def}}{=} (F\text{col}_i Di) + (G\text{col}_i Di) \\
\overset{\supset}{=} (\text{col}_i FDi) + (\text{col}_i GDi) \\
\overset{\subseteq}{=} \text{col}_i (DFi + DGi)
\]

The first step follows by induction on \( F \) and \( G \); the second step can be proven directly from the definition of a colimit (coproduct).

Hence any such \( F \) preserves \( D : \omega \to \text{Set} \) colimits.
Datatype Initial Algebra, Categorically

It follows from this, plus the fact that identity functors and constant functors preserve colimits of diagrams $D: \omega \rightarrow C$ for any $C$, that the datatype functor

$$F \overset{\text{def}}{=} F_{T_1} + \ldots + F_{T_m}: \text{Set} \longrightarrow \text{Set}$$

preserves colimits of shape $D: \omega \longrightarrow \text{Set}$. Since in fact $\text{Set}$ has all colimits, by purely categorical reasoning it has an initial algebra $\sigma: F1 \longrightarrow 1$. 
Find out what nominal sets are, and learn the basic properties of the category $\text{Nom}$ (of nominal sets and finitely supported functions) such as finite products and coproducts. Follow this up by learning what a nominal algebraic datatype is. Then see if you can construct an initial algebra model of expressions for such a datatype, proving the relevant properties, and further show that initial algebras exist for purely categorical reasons, much as we did in these slides for (ordinary) algebraic datatypes.
References