Categories for the Lazy Functional Programmer

Thorsten Altenkirch

School of Computer Science
University of Nottingham

May 15, 2009
Saunders MacLane  
(1909 - 2005)  
Samuel Eilenberg  
(1913 - 1998)

- Originally: tool for algebraic topology.
- Relevance for Computer Science (Lambek’s obs)
  E.g. *Cartesian Closed Cats* $\approx$ *Simply Typed $\lambda$-calculus*
- Categorical concepts in Haskell: *Functor*, *Monad*, ...
- Is Category Theory *Abstract Nonsense*?
- Is Category Theory an alternative to Set Theory?
Books

MacLane

Pierce

Awodey
Overview

1. Intro
2. Categories
3. Functors and natural transformations
4. Adjunctions
5. Products and coproducts
6. Exponentials
7. Limits and Colimits
8. Initial algebras and terminal coalgebras
9. Monads and Comonads
The category **Set**

**Objects:** Sets

\[ |\text{Set}| = \text{Set} \]

**Morphisms:** Functions, given \( A, B \in |\text{Set}| \)

\[ \text{Set}(A, B) = A \rightarrow B \]

**Identity:** Given \( A \in \text{Set} \)

\[ \text{id}_A \in \text{Set}(A, A) \]

\[ \text{id}_A = \lambda a. a \]

**Composition:** Given \( f \in \text{Set}(B, C), g \in \text{Set}(A, B) \):

\[ f \circ g \in \text{Set}(A, C) \]

\[ f \circ g = \lambda a. f(ga) \]

**Laws:**

\[ f \circ \text{id} = f \]

\[ \text{id} \circ f = f \]

\[ (f \circ g) \circ h = f \circ (g \circ h) \]
Exercise 1

Derive the laws for \textbf{Set} using only the equations of the simply typed \(\lambda\)-calculus, i.e.

\begin{align*}
\beta & \quad (\lambda x. t)u = t[x := u] \\
\eta & \quad \lambda x. t \ x = t \text{ if } x \not\in \text{FV } t \\
\xi & \quad t = u \\
\xi & \quad \lambda x. t = \lambda x. u
\end{align*}
Definition: \( \mathbf{C} \) is a category
A (large) set of objects:

\[ |\mathbf{C}| \in \text{Set}_1 \]

Morphisms: For every \( A, B \in |\mathbf{C}| \) a homset

\[ \mathbf{C}(A, B) \in \text{Set} \]

Identity: For any \( A \in \mathbf{C} \):

\[ \text{id}_A \in \mathbf{C}(A, A) \]

Composition: For \( f \in \mathbf{C}(B, C), g \in \mathbf{C}(A, B) \):

\[ f \circ g \in \mathbf{C}(A, C) \]

Laws:

\[ f \circ \text{id} = f \]

\[ \text{id} \circ f = f \]

\[ (f \circ g) \circ h = f \circ (g \circ h) \]
Size matters

- I assume as given a predicative hierarchy of set-theoretic universes:

  \[ \text{Set} = \text{Set}_0 \in \text{Set}_1 \in \text{Set}_2 \in \ldots \]

  which is cumulative

  \[ \text{Set}_0 \subseteq \text{Set}_1 \subseteq \text{Set}_2 \subseteq \ldots \]

- To accommodate categories like \textbf{Set} we allow that the objects are a large set (\(|C| \in \text{Set}_1\)) but require the homsets to be proper sets \(C(A, B) \in \text{Set} = \text{Set}_0\).

- A category is \textit{small}, if the objects are a set \(|C| \in \text{Set}\)

- We can repeat this definition at higher levels, a category at level \(n\) has as objects \(|C| \in \text{Set}_{n+1}\) and homsets \(C(A, B) \in \text{Set}_n\)
Dual category

Given a category $\mathcal{C}$ there is a dual category $\mathcal{C}^{\text{op}}$ with

- **Objects** $|\mathcal{C}^{\text{op}}| = |\mathcal{C}|$
- **Homsets** $\mathcal{C}^{\text{op}}(A, B) = \mathcal{C}(B, A)$

and composition defined backwards.

Notation

For $n \in \mathbb{N}$ we define

$$\bar{n} = \{i < n\}$$

Question

How many elements are in $\text{Set}(\bar{2}, \bar{3})$ and in $\text{SET}^{\text{op}}(\bar{2}, \bar{3})$?
**Isomorphism**

An isomorphism between \( A, B \in |\mathbf{C}| \) is given by two morphisms \( f \in \mathbf{C}(A, B) \) and \( f^{-1} \in \mathbf{C}(B, A) \) such that \( f \circ f^{-1} = \text{id} \), \( f^{-1} \circ f = \text{id} \):

\[
\begin{array}{ccc}
\text{id} & \circlearrowleft & A \\
\circlearrowleft & f & \rightarrow \\
\circlearrowleft & f^{-1} & \circlearrowleft \\
B & \circlearrowright & \text{id}
\end{array}
\]

We say that \( A \) and \( B \) are isomorphic \( A \simeq B \).

- Isomorphic sets are the same up to a *renaming* of elements.
- Concepts in category theory are usually defined *up to isomorphism*. 
Exercise 2

Which of the following isomorphisms hold in $\text{Set}$:

- $\overline{2} + \overline{2} \cong \overline{4}$
- $\overline{2} \times \overline{2} \cong \overline{4}$
- $\overline{2} \to \overline{2} \cong \overline{4}$
- $\mathbb{N} + \mathbb{N} \cong \mathbb{N}$
- $\mathbb{N} \times \mathbb{N} \cong \mathbb{N}$
- $\mathbb{N} \to \mathbb{N} \cong \mathbb{N}$

$A \times B$ is cartesian product

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

$A + B$ is disjoint union

$$A + B = \{\text{inl } a \mid a \in A\} \cup \{\text{inr } b \mid b \in B\}$$
Monomorphism

\( f \in C(B, C) \) is a monomorphism (short \textit{mono}), if for all \( g, h \in C(A, B) \)

\[
f \circ g = f \circ h \\
g = h
\]

- In \textbf{Set} monos are precisely the injective functions.
- We draw monos as \( A \rightarrow B \)
Epimorphism

\[ f \in C(A, B) \] is a epimorphism (short \textit{epi}), if for all \( g, h \in C(B, C) \)

\[
\begin{align*}
g \circ f &= h \circ f \\
g &= h
\end{align*}
\]

- In \textbf{Set} epis are precisely the surjective functions.
- We draw epis as \( A \longrightarrow B \)
Exercise 3

Show that every iso is both mono and epi.

- Assuming classical (non-constructive) logic, all bijections in \( \textbf{Set} \) are isos.

Exercise 4

Show that in \( \textbf{Set} \) every morphism \( f \in A \rightarrow B \) can be written as a composition of an epi and a mono:

\[ A \rightarrow C \rightarrow B \]

\[ f \leftarrow e \leftarrow m \]
Monoids

Definition: Monoid

A monoid \((M, e, \ast)\) is given by \(M \in \text{Set}\), \(e \in M\) and \((\ast) \in M \to M \to M\) such that:

\[
\begin{align*}
x \ast e &= x \\
e \ast x &= x \\
(x \ast y) \ast z &= x \ast (y \ast z)
\end{align*}
\]

Example

\((\mathbb{N}, 0, +)\) is a (commutative) monoid.

Question

Give an example of a non-commutative monoid.
Monoids correspond to categories with one object.

**Monoid as a category**

Every monoid \((M, e, \ast)\) gives rise to a category \(\textbf{M}\)

- **Objects:** \(|\textbf{M}| = \{(())\}\)
- **Morphisms:** \(\textbf{M}(((), ())) = M\)

\(e\) is the identity, \(\ast\) is composition.
Preorder

\((A, \sqsubseteq)\) with \(A \in \text{Set}\) and \(\sqsubseteq \in A \rightarrow A \rightarrow \text{Prop}\) is a preorder if \(R\) is

- reflexive \(\forall a \in A. a \sqsubseteq a\)
- transitive \(\forall a, b, c \in A. a \sqsubseteq b \rightarrow b \sqsubseteq c \rightarrow a \sqsubseteq c\)

Example

\((\mathbb{N}, \leq)\) is a preorder.

- \((\mathbb{N}, \leq)\) is a partial order, because it also satisfies

\[
\begin{align*}
  m \leq n & \quad n \leq m \\
  \hline
  m = n
\end{align*}
\]
Preorders correspond to categories where the homsets have at most one element.

**A preorder as a category**

A preorder \((A, \subseteq)\) can be viewed as a category \(\mathbf{A}\):

- **Objects** \(|\mathbf{A}| = A\)
- **Homsets** \(\mathbf{A}(a, b) = \begin{cases} \{()\} & \text{if } a \subseteq b \\ \{\} & \text{otherwise} \end{cases}\)

Monoids and preorders are degenerate categories.
Categories of sets with structure

The category of Monoids: \textbf{Mon}

**Objects:** Monoids \((M, e, *)\)

**Morphisms** \(\textbf{Mon}((M, e, *), (M', e', *))\) is given by \(f \in M \rightarrow M'\) such that \(f e = e'\) and \(f (x * y) = (f x) *' (f y)\).

Example

The embedding \(i \in \textbf{Mon}((\mathbb{N}, 0, +), (\mathbb{Z}, 0, +))\) with \(i n = n\)

Exercise 5

Show that \(i\) is a mono and an epi but not an iso in \textbf{Mon}.

Exercise 6

Define the category \textbf{Pre} of preorders and monotone functions.
Finite Sets

**FinSet**
- **Objects:** Finite Sets
- **Morphisms:** Functions

- **FinSet** is a full subcategory of **Set**.

**FinSetSkel**
- **Objects:** \( \mathbb{N} \)
- **Morphisms:** \( \text{FinSetSkel}(m, n) = \bar{m} \to \bar{n} \)

- **FinSetSkel** is skeletal, any isomorphic objects are equal.
- **FinSet** and **FinSetSkel** are equivalent (in the appropriate sense).
### Computational Effects

#### Error

Given a set of Errors $E \in \text{Set}$

**Objects:** Sets

**Morphisms:** $\text{Error}(A, B) = A \rightarrow B + E$

#### State

Given a set of states: $S \in \text{Set}$

**Objects:** Sets

**Morphisms:** $\text{State}(A, B) = A \times S \rightarrow B \times S$

### Exercise 7

Define identity and composition for both categories.
\textbf{\(\lambda\)-terms}

\textbf{Lam}

\textbf{Objects:} Finite sets of variables

\textbf{Morphisms:} \(\text{Lam}(X, Y) = Y \to \text{Lam} X\) where \(\text{Lam} X\) is the set of \(\lambda\)-terms whose free variables are in \(X\).

\textbf{Exercise 8}

1. Define identity and composition.
2. Extend the definition to typed \(\lambda\)-calculus.
### Product categories

Given categories $\mathbf{C}, \mathbf{D}$ we define $\mathbf{C} \times \mathbf{D}$:

- **Objects**: $\mathbf{C} \times \mathbf{D}$
- **Morphisms**: $\mathbf{C} \times \mathbf{D}((A, B), (C, D)) = \mathbf{C}(A, C) \times \mathbf{D}(B, D)$

We abbreviate $\mathbf{C}^2 = \mathbf{C} \times \mathbf{C}$

### Slice categories

Given a category $\mathbf{C}$ and an object $A \in |\mathbf{C}|$ we define $\mathbf{C}/A$ as:

- **Objects**: $|\mathbf{C}/A| = \Sigma B \in |\mathbf{C}|. \mathbf{C}(B, A)$
- **Morphisms**: $\mathbf{C}/A(((B, f), (C, g))$:

$$
\begin{array}{ccc}
B & \xrightarrow{h} & C \\
\downarrow f & & \downarrow g \\
A & \xrightarrow{g} & C
\end{array}
$$
Computable sets

$\omega$-Set

**Objects:** A Set $A$ and a relation $\models_A \subseteq \mathbb{N} \times A$ such that

$$\forall a \in A. \exists i \in \mathbb{N}. i \models_A a.$$  

**Morphisms:**

$$\omega-\text{Set}(((A, \models_A), (B, \models_B)))$$

$$= \{ f \in A \to B \mid \exists i \in \mathbb{N}. \forall j, a.j \models_A a$$

$$\rightarrow \exists k. \{i\}j \downarrow k \land k \models_B f a\}$$

where $\{i\}j \downarrow k$ means the $i$th Turing machine applied to input $j$ terminates and returns $k$. 
Partial computations

\(\omega\text{-CPO}\)

Objects: \((A, \sqsubseteq_A, \bigsqcup_A)\) such that \((A, \sqsubseteq_A)\) is a partial order, and

\[
\bigsqcup_A \in \{ f \in \mathbb{N} \to A \mid \forall i. f_i \sqsubseteq_A f(i + 1) \} \to A
\]

is the least upper bound of a chain, i.e. \(\forall i. f_i \sqsubseteq \bigsqcup_A f\) and \((\forall i. f_i \sqsubseteq a) \to \bigsqcup_A f \sqsubseteq a\).

Morphisms: \(\omega\text{-CPO}((A, \sqsubseteq_A, \bigsqcup_A), (B, \sqsubseteq_B, \bigsqcup_B))\) is given by functions \(f \in A \to B\) which are:

- monotone
  \[
  a \sqsubseteq_A b \quad \Rightarrow \quad f a \sqsubseteq f b
  \]

- continuous
  \[
  f(\bigsqcup_A h) = \bigsqcup_B (f \circ h)
  \]
Definition: Functor

Given categories \( C, D \) a functor \( F \in C \rightarrow D \) is given by

- a map on objects \( F \in |C| \rightarrow |D| \)
- maps on morphisms

Given \( f \in C(A, B) \), \( F f \in D(F A, F B) \) such that

\[
F \text{id}_A = \text{id}_{F A} \\
F(f \circ g) = (F f) \circ (F g)
\]

A functor \( F \in C \rightarrow C \) is called an endofunctor.

Example

List : \textbf{Set} \rightarrow \textbf{Set}, the list functor on morphisms is given by map

\[
\text{map } f \left[ \right] = \left[ \right] \\
\text{map } f \left( a : as \right) = f a : \text{map } f \ as
\]

We just write List \( f = \text{map } f \).
Exercise 9

Show that \texttt{List} satisfies the functor laws.

Question

We consider endofunctors on \texttt{Set}, given maps on objects:

1. Is $F_1 X = X \rightarrow \mathbb{N}$ a functor?
2. Is $F_2 X = X \rightarrow X$ a functor?
3. Is $F_3 X = (X \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ a functor?

- All type expressions with only positive occurrences of a set variable give rise to (covariant) functors in $\texttt{Set} \rightarrow \texttt{Set}$.
- All type expressions with only negative occurrences of a set variable give rise to (contravariant) functors in $\texttt{Set}^{\text{op}} \rightarrow \texttt{Set}$.

Exercise 10

Is there a type-expression which is not positive but still gives rise to a covariant endofunctor on \texttt{Set}?
Definition: natural transformation

Given functors $F, G \in \mathbf{C} \rightarrow \mathbf{D}$ a natural transformation $\alpha : F \rightarrow G$ is given by a family of maps

$$\alpha \in \prod_{A \in \mathbf{C}} \mathbf{D}(F A, G A)$$

such that for any $f \in \mathbf{C}(A, B)$

$$\begin{array}{ccc}
F A & \xrightarrow{\alpha_A} & G A \\
\downarrow F f & & \downarrow G f \\
F B & \xrightarrow{\alpha_B} & G B
\end{array}$$

Exercise 11

1. Show that $\text{reverse} \in \prod X \in \text{Set.} \text{List } X \rightarrow \text{List } X$ is a natural transformation.

2. Give a family of maps with the same type, which is not natural.
Functor categories

Given categories $\mathbf{C}, \mathbf{D}$ the functor category $\mathbf{C} \to \mathbf{D}$ is given by:

**Objects:** Functors $F \in \mathbf{C} \to \mathbf{D}$

**Morphisms** Given $F, G \in \mathbf{C} \to \mathbf{D}$, a morphism is a natural transformation $\alpha \in F \to G$

- If $\mathbf{C}$ is small, the functor category $\text{PSh} \mathbf{C} = \mathbf{C}^{\text{op}} \to \text{Set}$ is called *the category of presheaves over $\mathbf{C}$*.

**Exercise 12**

Spell out the details of the objects and morphisms of $\text{PSh} (\mathbb{N}, \leq)$. 
We define a functor $Y$, the Yoneda embedding:

$$Y ∈ C → PSh C$$

$$Y A = \lambda X. C(X, A)$$

**Exercise 13**

Show that $Y$ is a functor.

**The Yoneda Lemma**

Given $F ∈ PSh C$ the following are naturally isomorphic in $A ∈ |C|$

$$PSh C(Y A, F) ≃ F A$$

**Exercise 14**

Prove the Yoneda Lemma.
The category of categories

**CAT**

The category of categories is given by:

**Objects:** Categories  
**Morphisms:** Functors

- This is a category on level 1, $|\text{CAT}| \in \text{Set}_2$.
- **CAT** is a 2-category because its homsets are categories themselves (+ Godemont rules).
Free Monoids

- The forgetful functor:

\[ U \in \text{Mon} \rightarrow \text{Set} \]
\[ U (M, e, \ast) = M \]

- Can we go the other way?

- The free functor:

\[ F \in \text{Set} \rightarrow \text{Mon} \]
\[ F A = (\text{List} A, [], (++)) \]

- How to specify that \( F \) is \textit{free}?
We construct two natural families of maps:

\[
\begin{align*}
\text{Mon}(F A, (M, e, *)) & \xrightarrow{\phi} \text{Set}(A, U (M, e, *)) \\
\phi^{-1} & \xleftarrow{\phi}
\end{align*}
\]

\[
\phi \in (\text{List } A \rightarrow M) \rightarrow A \rightarrow M
\]
\[
\phi f a = f [a]
\]
\[
\phi^{-1} \in (A \rightarrow M) \rightarrow (\text{List } A \rightarrow M)
\]
\[
\phi^{-1} g [] = e
\]
\[
\phi^{-1} g (a :: as) = (g a) \ast (\phi^{-1} g as)
\]

**Exercise 15**

Show:

1. \(\phi \circ \phi^{-1} = \text{id}\)
2. \(\phi^{-1} \circ \phi = \text{id}\)
**Definition: Adjunction**

Given functors:

\[
\begin{array}{ccc}
C & \xrightarrow{U} & D \\
\xleftarrow{F} & & \xleftarrow{}
\end{array}
\]

we say that \( F \) is left adjoint to \( U \) (\( F \dashv U \))

or \( U \) is right adjoint to \( F \)

if there is a natural isomorphism (in \( A \in |D|, B \in |C| \))

\[
\begin{array}{ccc}
D(F A, B) & \xrightarrow{\phi} & C(A, U B) \\
\xleftarrow{\phi^{-1}} & & \xleftarrow{}
\end{array}
\]
A semilattice (with zero) is a monoid \((M, e, \ast)\) such that:

- **commutative**, if for all \(x, y \in M\):
  \[x \ast y = y \ast x\]

- **idempotent**, if for all \(x \in M\):
  \[x \ast x = x\]

- We define \(\text{SLat}\) as the category of semilattices with zero.
- Morphisms and forgetful functors are defined as for \(\text{Mon}\).

**Exercise 16**

Construct the free functor \(F \in \text{Set} \rightarrow \text{SLat}\) and show that \(F\) is left adjoint to \(U \in \text{SLat} \rightarrow \text{Set}\).
Products in $\text{Set}$

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

$\pi_0(a, b) = a$

$\pi_1(a, b) = b$

$\langle f, g \rangle \circ c = (f \circ c, f \circ c)$

Laws:

$\pi_0 \circ \langle f, g \rangle = f$

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

$\pi_0 \circ h = f \quad \pi_1 \circ h = g$

$h = \langle f, g \rangle$
Given objects \( A, B \in \mathbf{C} \) we say that \( A \times B \) is their product if the morphisms \( \pi_0, \pi_1 \) exists and for every \( f, g \) there is a morphism \( \langle f, g \rangle \) so that the following diagram commutes:

\[
\begin{array}{ccc}
\pi_0 & \pi_1 \\
A \times B & \rightarrow & B \\
\downarrow & & \downarrow \\
A & \leftarrow & \langle f, g \rangle
\end{array}
\]

Moreover, the morphism \( \langle f, g \rangle \) is the unique morphism which makes this diagram commute, i.e.

\[
\pi_0 \circ h = f \quad \pi_1 \circ h = g \\
\frac{\pi_0 \circ h = f \quad \pi_1 \circ h = g}{h = \langle f, g \rangle}
\]
Exercise 17
Show that products in $C$ give rise to a functor $(\times) \in C^2 \to C$.

Exercise 18
Show that the following equation holds

$$\langle f, g \rangle \circ h = \langle f \circ h, g \circ h \rangle$$

Exercise 19
Show that the following isomorphism exist in all categories with products:

$$A \times B \simeq B \times A$$

and that the assignment is natural in $A, B$. 
Coproducts in \textbf{Set}

$$A + B = \{\text{inl } a \mid a \in A\} \cup \{\text{inr } b \mid b \in B\}$$

$$[f, g] (\text{inl } a) = f a$$

$$[f, g] (\text{inr } b) = g b$$

Laws:

$$[f, g] \circ \text{inl} = f$$

$$[f, g] \circ \text{inr} = g$$

$$h \circ \text{inl} = f \quad h \circ \text{inr} = g$$

$$h = [f, g]$$
Coproducts

Given objects $A, B \in \mathbf{C}$ we say that $A + B$ is their coproduct if the morphisms $\text{inl}, \text{inr}$ exists and for every $f, g$ there is a morphism $[f, g]$ so that the following diagram commutes:

\[
\begin{array}{ccc}
A & \xrightarrow{\text{inl}} & A + B \\
\downarrow & & \downarrow \\
C & \xrightarrow{f} & [f, g] & \xleftarrow{g} & B \\
\downarrow & & \downarrow & \downarrow & \downarrow \\
B & \xleftarrow{\text{inr}} & A + B & \xrightarrow{\text{inl}} & A \\
\end{array}
\]

Moreover, the morphism $[f, g]$ is the unique morphism which makes this diagram commute, i.e.

\[
h \circ \text{inl} = f \quad h \circ \text{inr} = g
\]

\[
h = [f, g]
\]
Products and coproducts are dual concepts:
Products in $|\mathbf{C}|$ are coproducts in $|\mathbf{C}^{\text{op}}|$ and vice versa.

Products and coproducts are left and right adjoints of the diagonal functor:

$$\Delta \in \mathbf{C} \to \mathbf{C}^2$$

$$\Delta A = (A, A)$$
Terminal objects

1 \in |C| is a terminal object, if for any object \( A \in C \) there is exactly one arrow \( !_A : A \rightarrow 1 \):

Initial objects

0 \in |C| is an initial object, if for any object \( A \in C \) there is exactly one arrow \( ?_A : 0 \rightarrow A \):

Question

What are initial and terminal objects in \textbf{Set}?

Exercise 20

Show that any two terminal objects are isomorphic.
Global elements

- In \textbf{Set} we have that
  \[ \text{Set}(1, A) \simeq A \]

- Hence the elements of \( \mathbf{C}(1, A) \) are called the \textbf{global elements} of \( A \).

- A category \( \mathbf{C} \) is \textit{well pointed}, if for \( f, g \in \mathbf{C}(A, B) \) we have
  \[
  \forall a \in \mathbf{C}(1, A). f \circ a = g \circ a \\
  \quad \Rightarrow \quad f = g
  \]

- \textbf{Set} is well pointed.

Exercise 21

Consider \( \mathbf{PSh}(\mathbb{N}, \leq) \) again. What is the terminal object and what are global elements? Show that \( \mathbf{PSh}(\mathbb{N}, \leq) \) is not well pointed.
Exercise 22

Construct the following isomorphism in \textbf{Set}:

\[ A \times (B + C) \cong A \times B + A \times C \]

Exercise 23

Show that \textbf{CMon} (the category of commutative monoids) has products and coproducts.

Exercise 24

Give a counterexample for the isomorphism:

\[ A \times (B + C) \nleq A \times B + A \times C \]

in \textbf{CMon}.
Exponentials in \textbf{Set}

- In \textbf{Set} we have the curry/uncurry isomorphism:
  \[ A \times B \to C \simeq A \to (B \to C) \]

- Indeed this is an adjunction \( F \dashv G \) for
  \[
  \begin{align*}
  F, G &\in \textbf{Set} \to \textbf{Set} \\
  FX &= X \times B \\
  GX &= B \to X
  \end{align*}
  \]
  \[
  \textbf{Set}(FA, C) \simeq \textbf{Set}(A, GC)
  \]
Exponentials

Given a category $\mathbf{C}$ with products. We say that the object $B \in \mathbf{C}$ is exponentiable, if the functor $F X = X \times B$ has a right adjoint $F \dashv G$, which we write as $G X = B \rightarrow X$.

A category with products where all objects are exponentiable is called cartesian closed.

$B \rightarrow C$ is often written as $C^B$.

**Question**

What are the exponentials in $\text{FinSetSkel}$?
Exercise 25
Show that the category of typed $\lambda$-terms is cartesian closed.

Indeed, this is the initial cartesian closed category (or the classifying category).

Exercise 26
Show that in a cartesian closed category with coproducts we have that

$$A \times (B + C) \simeq (A \times B) + (A \times C)$$

Corollary
\textbf{CMon} is not cartesian closed.
Exercise 27
Show that the presheaf categories \((\text{PSh} \mathbf{C})\) are cartesian closed.

Exercise 28
Is there a cartesian closed category whose dual is also cartesian closed?
Pullbacks

Given arrows $f \in \mathbf{C}(A, C)$ and $g \in \mathbf{C}(B, C)$, $(f \times_C g, \pi_0, \pi_1)$ is their pullback, if the diagram below commutes and for every $(D, p_0, p_1)$ there is a unique arrow $<p_0, p_1>$ such that the diagram commutes:

```
D
  |  p_0
  |  <p_0, p_1>
    |  \\
    v \\
  f \times_C g \xrightarrow{\pi_0} A
  |  \pi_1
  V \\
B \xrightarrow{g} C
  V \\
  V \\
  V \\
  \downarrow f
```

Pullbacks in $\textbf{Set}$:

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

Given arrows $f \in C(A, B)$ and $g \in C(A, C)$, $(f +^A g, \text{inl, inr})$ is their pushout, if the diagram below commutes and for every $(D, i_0, i_1)$ there is a unique arrow $[p_0, p_1]$ such that the diagram commutes:

Exercise 29
What are pushouts in $\textbf{Set}$?
Limits and colimits

Given a small category of diagrams $\mathbf{D}$, a $\mathbf{D}$-diagram in $\mathbf{C}$ is given by a functor $F \in \mathbf{D} \to \mathbf{C}$. A cone of a diagram is given by an object $D \in \mathbf{C}$ and a natural transformation $\alpha \in K_D \to F$ where $K_D X = D$ is a constant functor.

Morphisms between cones $(D, \alpha)$ and $(E, \beta)$ are given by $f \in D \to E$ such that $\alpha \circ f = \beta$.

The limit of $F$ is the terminal object in the category of cones.

Dually, a cocone is given by a natural transformation $\alpha \in F \to K_D$, and a morphism of cocones $(D, \alpha)$ and $(E, \beta)$ are given by $f \in D \to E$ such that $f \circ \alpha = \beta$.

The colimit of $F$ is the initial object in the category of cocones.
Examples

- Products are given by limits of
  \[
  \bullet \quad \bullet
  \]
  Note that we are leaving out identity arrows.
- Dually, coproducts are given by colimits of the same diagram.
- Pullbacks are limits of
  \[
  \bullet \quad \bullet
  \downarrow \quad \downarrow
  \bullet \quad \bullet
  \rightarrow \quad \rightarrow
  \]
- Pushouts are colimits of the dual diagram:
  \[
  \bullet \quad \bullet
  \downarrow
  \bullet
  \rightarrow \quad \rightarrow
  \]
Equalizers are limits of $\bullet \to \bullet \to \bullet \to \bullet$.

Dually, coequalizers are colimits of the same diagram.

Exercise 30
What are equalizers and coequalizers in $\mathbf{Set}$?

Exercise 31
Show that pullbacks can be constructed from equalizers and products.

Actually, all finite limits can be constructed from equalizers and finite products (i.e. binary products and terminal objects).
Diagrams of \((\mathbb{N}, \leq)\) are called \(\omega\)-chains:

\[
\begin{array}{c}
A_0 \xrightarrow{a_0} A_1 \xrightarrow{a_1} A_2 \xrightarrow{a_2} \ldots \\
\end{array}
\]

Note that we are leaving out the composites of arrows.

An \(\omega\)-chain in \(\textbf{Set}\) is given by

\[
\begin{align*}
A & \in \mathbb{N} \rightarrow \text{Set} \\
a & \in \prod n \in \mathbb{N}. A_n \rightarrow A(n + 1)
\end{align*}
\]

We write \(\text{colim} (A, a)\) for the colimit of an \(\omega\)-chain.

**Exercise 32**

What is the colimit of the following chain?

\[
\begin{align*}
A_n &= \bar{n} \\
a_n i &= i
\end{align*}
\]
Dually, Diagrams of \((\mathbb{N}, \geq)\) are called \(\omega\)-cochains:

\[
A \xleftarrow{a_0} A_1 \xleftarrow{a_1} A_2 \xleftarrow{a_2} \ldots
\]

An \(\omega\)-cochain in \textbf{Set} is given by

\[
A \in \mathbb{N} \to \text{Set} \\
a \in \prod n \in \mathbb{N}. A(n + 1) \to A_n
\]

We write \(\lim (A, a)\) for the limit of an \(\omega\)-cochain.

**Exercise 33**

Given a set \(X \in \text{Set}\). What is the limit of the following chain?

\[
A_n = \bar{n} \to X \\
anf = \lambda i. f i
\]
• Natural numbers $\mathbb{N} \in \text{Set}$ are given by:

\[
0 \in \mathbb{N} \\
\simeq 1 \to \mathbb{N} \\
S \in \mathbb{N} \to \mathbb{N}
\]

• We can combine the two constructors in one morphism:

\[
[0, S] \in 1 + \mathbb{N} \to \mathbb{N}
\]

• The functor $T X = 1 + X$ is called the signature functor.

• A pair $(A \in \text{Set}, f \in 1 + A \to A)$ is a $1+$-algebra.
For any \(1^+\)-algebra \((A, f)\) there is a unique morphism \(\text{fold}(A, f)\) such that the following diagram commutes:

\[
\begin{align*}
1 + \mathbb{N} & \xrightarrow{[0,S]} \mathbb{N} \\
1 + (\text{fold}(A, f)) & \downarrow \quad \downarrow \\
1 + A & \xrightarrow{f} A
\end{align*}
\]

with

\[
\begin{align*}
\text{fold}(A, f) 0 &= f(\text{inl}()) \\
\text{fold}(A, f)(S\ n) &= f(\text{inr}(\text{fold}(A, f)n))
\end{align*}
\]

**Exercise 34**

Define addition \((+) \in \mathbb{N} \rightarrow \mathbb{N} \rightarrow \mathbb{N}\) using \(\text{fold}\).
$T$-algebras

Given an endofunctor $T \in \mathbf{C} \rightarrow \mathbf{C}$ the category of $T$-algebras is given by

**Objects** $T$-algebras $(A, f)$ with

\[ T A \xrightarrow{f} A \]

**Morphisms** Given $T$-algebras $(A, f), (B, g)$ a $T$-algebra morphism is a morphism $h \in \mathbf{C}(A, B)$ such that

\[ \begin{align*}
T A & \xrightarrow{f} A \\
T h & \downarrow h \\
T B & \xrightarrow{g} B
\end{align*} \]

commutes.
Initial \( T \)-algebras

The initial object (if it exists) in the category of \( T \)-algebras is denoted as \( (\mu T, \text{in}_T) \). For every \( T \)-algebra \((A, f)\) there is a unique morphism \( \text{fold}_T (A, f) \) such that

\[
\begin{array}{c}
T (\mu T) \xrightarrow{\text{in}_T} \mathbb{N} \\
\downarrow \\
T (\text{fold} (A, f)) \\
\downarrow \\
T A \xrightarrow{f} A
\end{array}
\]

commutes.
Given $A \in \text{Set}$ the set of streams over $A$: $A^\omega$ comes with two destructors

$$\text{hd} \in A^\omega \to A$$
$$\text{tl} \in A^\omega \to A^\omega$$

We can combine the two destructors in one morphism:

$$<\text{hd}, \text{tl}> \in A^\omega \to A \times A^\omega$$

A pair $(X \in \text{Set}, f \in X \to A \times X)$ is a $A \times$-coalgebra.
For any $A \times$-algebra $(X, f)$ there is a unique morphism $\text{unfold}(X, f)$ such that the following diagram commutes:

\[
\begin{array}{ccc}
X & \xrightarrow{f} & A \times X \\
\downarrow{\text{unfold}(X,f)} & & \downarrow{A \times \text{unfold}(X,f)} \\
A^\omega & \xrightarrow{<\text{hd},\text{tl}>} & A \times A^\omega
\end{array}
\]

with

\[
\text{hd}(\text{unfold}(X, f) x) = \pi_0(f x) \\
\text{tl}(\text{unfold}(X, f) x) = \text{unfold}(X, f)(\pi_1(f x))
\]

**Exercise 35**

Define the function $\text{from } \mathbb{N} \rightarrow \mathbb{N}^\omega$, which produces the stream of natural numbers starting with a given number, using $\text{unfold}$. 
Initial algebras and terminal coalgebras

Terminal coalgebras

$T$-coalgebras

Dually, given an endofunctor $T \in \mathbf{C} \rightarrow \mathbf{C}$ the category of $T$-coalgebras is given by

**Objects** $T$-coalgebras $(A, f)$ with

$$
\begin{array}{c}
A \\
\downarrow f
\end{array} \\
\begin{array}{c}
T A \\
\downarrow
\end{array}
$$

**Morphisms** Given $T$-coalgebras $(A, f), (B, g)$ a $T$-coalgebra morphism is a morphism $h \in \mathbf{C}(A, B)$ such that

$$
\begin{array}{c}
A \\
\downarrow h
\end{array} \\
\begin{array}{c}
T A \\
\downarrow T h
\end{array} \\
\begin{array}{c}
B \\
\downarrow g
\end{array}
$$

commutes.
Terminal $T$-coalgebras

The terminal object (if it exists) in the category of $T$-coalgebras is denoted as $(\nu T, \text{out}_T)$. For every $T$-coalgebra $(A, f)$ there is a unique morphism $\text{unfold}_T (A, f)$ such that

\[
\begin{align*}
A & \xrightarrow{f} TA \\
\nu T & \xrightarrow{\text{out}_T} T (\nu T)
\end{align*}
\]
Lambek’s lemma

- Initial algebras and terminal coalgebras are always isomorphisms.
- We construct the inverse of $\text{in}_T \in \mathbf{C}(T(\mu T), \mu T)$ as

$$\text{in}_T^{-1} \in \mathbf{C}(\mu T, T(\mu T))$$

$$\text{in}_T^{-1} = \text{fold}_T(T(\mu T), T\text{in}_T)$$

- Dually, we construct an inverse to $\text{out}_T$.

Exercise 36

Construct explicitly the inverses to the $[0, S]$ and $<\text{hd}, \text{tl}>$.

Exercise 37

Prove Lambek’s lemma, i.e. show that $\text{in}_T^{-1}$ is inverse to $\text{in}_T$. 
A functor $T$ is called $\omega$-cocontinuous if it preserves colimits of $\omega$-chains, that is

$$T(\text{colim} (A, a)) \simeq \text{colim} (\lambda n. T(A n), \lambda n. T(a n))$$

We can construct the initial $T$-algebra of an $\omega$-cocontinuous functor $T$ by constructing the colimit of the following chain:

$$0 \xrightarrow{T} T0 \xrightarrow{T} T^20 \xrightarrow{T^2} \ldots$$

**Exercise 38**

Complete the construction, and show that the colimit is indeed an initial $T$-algebra.
Exercise 39

Dualize the previous slide. What is an $\omega$-continous functor? How can we construct its terminal coalgebra?

Exercise 40

Which of the following endofunctors on $\text{Set}$ are $\omega$-cocontinous, and which are $\omega$-continous:

\[
\begin{align*}
T_1 X &= X \times X \\
T_2 X &= \mathbb{N} \to X \\
T_3 X &= (X \to \mathbb{N}) \to \mathbb{N}
\end{align*}
\]
We define the functor of binary trees with labelled leaves:

\[ \begin{align*}
BT & \in \mathbf{Set} \rightarrow \mathbf{Set} \\
BT \, X & = \mu Y . X + Y \times Y
\end{align*} \]

We write \( L = \text{in} \circ \text{inl} \) and \( N = \text{in} \circ \text{inr} \) for the constructors.

The natural transformation \( \eta \) constructs a leaf:

\[ \eta_A \in A \rightarrow BT \, A \]
\[ \eta_A = \lambda a . L \, a \]

We define a natural transformation \( \text{bind} \), which replaces each leaf by a tree.

\[ \begin{align*}
\text{bind}_{A,B} & \in (A \rightarrow BT \, B) \rightarrow BT \, A \rightarrow BT \, B \\
\text{bind}_{A,B} \, f \, (L \, a) & = f \, a \\
\text{bind}_{A,B} \, f \, (N \,(l, r)) & = N \,(\text{bind}_{A,B} \, f \, l, \text{bind}_{A,B} \, f \, r)
\end{align*} \]

Haskell’s \( >>= \) can be defined as \( a >>= f = \text{bind} \, f \, a \).
Monads (Kleisli triple)

A monad on $\mathsf{C}$ is a triple $(T, \eta, \text{bind})$ with

$$
T \in \mathsf{C} \rightarrow \mathsf{C}
$$

$$
\eta \in \mathsf{C}(A, TA)
$$

$$
\text{bind} \in \mathsf{C}(A, TB) \rightarrow \mathsf{C}(TA, TB)
$$

such that

$$(\text{bind } f) \circ \eta = f$$

$$
\text{bind} (\eta \circ f) = f
$$

$$(\text{bind } f) \circ (\text{bind } g) = \text{bind} ((\text{bind } f) \circ g)$$

Exercise 41

Show that the operations on binary trees satisfy the laws of a monad.
Show that the following functors over \textbf{Set} give rise to monads (assuming \( E, S \in \text{Set} \)):

\[
\begin{align*}
T_{\text{Error}} X &= E + X \\
T_{\text{State}} X &= S \rightarrow (X \times S)
\end{align*}
\]
**Monad**

A monad on $\mathbf{C}$ is a triple $(T, \eta, \mu)$ with

- $T \in \mathbf{C} \to \mathbf{C}$
- $\eta \in I \to T$
- $\mu \in T^2 \to T$

(where $T^2 = T \circ T$) such that the following diagrams commute.

**Exercise 43**

Show that the two definitions are equivalent.
We define infinite, labelled binary trees:

\[ BT^\infty \in \text{Set} \rightarrow \text{Set} \]

\[ BT^\infty \ X = \nu Y. X \times (Y \times Y) \]

The operation \( \epsilon \) extracts the top label:

\[ \epsilon \in BT^\infty A \rightarrow A \]

\[ \epsilon (a, (l, r)) = a \]

\text{cobind} relabels a tree recursively:

\[ \text{cobind} \in (BT^\infty A \rightarrow B) \rightarrow (BT^\infty A \rightarrow BT^\infty B) \]

\[ \text{cobind} f \ t = (f \ t, \text{cobind} f (\pi_2 t), \text{cobind} f (\pi_3 t)) \]

Exercise 44

Show that \((BT^\infty, \epsilon, \text{cobind})\) is a comonad, i.e. a monad in \(\text{Set}^{\text{op}}\).
Kleisli category

Given a monad \((T, \eta, \text{bind})\) on \(\mathbf{C}\) we define the Kleisli category \(\mathbf{C}_T\) as:

**Objects:** \(|\mathbf{C}|\)

**Morphisms:** \(\mathbf{C}_T A B = \mathbf{C}(A, T B)\)

**Identity:** \(\eta \in \mathbf{C}_T A A\)

**Composition:** Given \(f \in \mathbf{C}_T B C,\ g \in \mathbf{C}_T A B\) we define

\[
f \circ_T g = (\text{bind } f) \circ g
\]

Exercise 45

Verify that that \(\mathbf{C}_T\) is indeed a category.

Exercise 46

Explicitely construct the Kleisli-categories of \(T_{\text{Error}}\) and \(T_{\text{State}}\)
Given an adjunction $F \dashv U$

$$\mathcal{D}(F A, B) \xleftrightarrow{\phi} \mathcal{C}(A, U B) \xleftrightarrow{\phi^{-1}} \mathcal{D}(F A, B)$$

we define:

$$\eta \in \mathcal{C}(A, U (F A))$$
$$\eta = \phi (\text{id}_{F A})$$
$$\epsilon \in \mathcal{D}(F, U B)B$$
$$\epsilon = \phi^{-1} (\text{id}_{U B})$$

this gives rise to a monad $(T, \epsilon, \mu)$ on $\mathcal{C}$

$$T = U F$$
$$\mu = U \epsilon F$$
Exercise 47

Spell out the constructed monad in the case where $F \in \text{Set} \rightarrow \text{Mon}$ is the free monad functor and $U \in \text{Mon} \rightarrow \text{Set}$ the forgetful functor.

Exercise 48

Verify the monad laws of the construction of a monad from an adjunction.
Using $C_T$ we can also go the other way: $C_T$ gives rise to an adjunction $F_T \dashv U_T$ such that $T = U_T \circ F_T$:

\[
F_T \in C \to C_T \\
F_T A = A \\
F_T f = \eta \circ f \\
U_T \in C_T \to C \\
U_T A = T A \\
U_T f = \mu \circ T f
\]

**Exercise 49**

Verify that $F_T \dashv U_T$.

This is not the only way to factor a monad into an adjunction. Another construction is the Eilenberg-Moore category $C^T$, indeed the two are initial and terminal objects in the category of factorisations.