Category Theory and Functional Programming
Some Basic Constructions

Andrés Sicard-Ramírez

Universidad EAFIT

Semester 2022-2

Preliminaries

Convention

The number assigned to chapters, examples, exercises, figures, pages, sections, and theorems on these slides correspond to the numbers assigned in the textbook [Abramsky and Tzevelekos 2011].

Outline

Initial and Terminal Objects

Products

Coproducts

References

Initial and Terminal Objects

Initial and Terminal Objects

Introduction

We shall introduce abstract characterisations of the empty set and the one-element sets in set theory.

Initial and Terminal Objects

Definition

Let $\mathcal{C}$ be a category. An object 0 in $\mathcal{C}$ is **initial** iff for any object A there is a unique arrow (universal property)

$$0 \to A.$$

Initial and Terminal Objects

Definition

Let $\mathcal{C}$ be a category. An object 0 in $\mathcal{C}$ is **initial** iff for any object A there is a unique arrow (universal property)

$$0 \to A.$$

Definition

Let $\mathcal{C}$ be a category. An object 1 in $\mathcal{C}$ is **terminal** iff for any object A there is a unique arrow (universal property)

$$A \to 1.$$

Initial and Terminal Objects

Remark

Initial and terminal objects are dual notions.

Initial and Terminal Objects

Example

- In $\mathbf{Set}$, the empty set is an initial object and any one-element set is a terminal object.
- In $\mathbf{Pos}$, the poset $(\emptyset, \emptyset)$ is an initial object and the poset $(\{*\}, \{(*, *)\})$ is a terminal object.
- In $\mathbf{Top}$, the topological space $(\emptyset, \{\emptyset\})$ is an initial object and the topological space $(\{*\}, \{\emptyset, \{*\}\})$ is a terminal object.

Initial and Terminal Objects

Example

- In $\mathbf{Set}$, the empty set is an initial object and any one-element set is a terminal object.
- In $\mathbf{Pos}$, the poset $(\emptyset, \emptyset)$ is an initial object and the poset $(\{*\}, \{(*, *)\})$ is a terminal object.
- In $\mathbf{Top}$, the topological space $(\emptyset, \{\emptyset\})$ is an initial object and the topological space $(\{*\}, \{\emptyset, \{*\}\})$ is a terminal object.

Exercise 1

Verify the initial and terminal objects in the previous example. In each case, identify the canonical arrows (Exercise 18).

Initial and Terminal Objects

Exercise 2

For the category $\mathbf{Rel}$, identify the initial and terminal objects, and the canonical arrows (Exercise 19).

Exercise 3

Suppose that a monoid, viewed as a category, has either an initial or a terminal object. What must the monoid be? (Exercise 20).

Initial and Terminal Objects

Example

In a poset, seen as a category,

(i) an object is initial iff it is the least element,

(ii) an object is terminal iff it is the greatest element.

Initial and Terminal Objects

Example

In a poset, seen as a category,

(i) an object is initial iff it is the least element,

(ii) an object is terminal iff it is the greatest element.

Question

Does a category need to have either an initial object or a terminal object?

Initial and Terminal Objects

Example

In a poset, seen as a category,

(i) an object is initial iff it is the least element,

(ii) an object is terminal iff it is the greatest element.

Question

Does a category need to have either an initial object or a terminal object?

Answer: No. The poset $(\mathbb{Z}, \leq)$, seen as a category, has neither.

Initial and Terminal Objects

Example

For **Hask**, the `Void` data type[†] is an initial object.

```
data Void

absurd :: Void -> a
absurd a = case a of {}
```

[†] From the module `Data.Void` of the `base` library.

Initial and Terminal Objects

Example

For **Hask**, the `Unit` data type is a terminal object.

```
data Unit = MkUnit

t :: a -> Unit
t _ = MkUnit
```

Initial and Terminal Objects

Example

For **Hask**, the `Unit` data type is a terminal object.

```
data Unit = MkUnit

t :: a -> Unit
t _ = MkUnit
```

The terminal object is built-in as `()` whose unique term is `()`, that is, `()::()`.

Initial and Terminal Objects

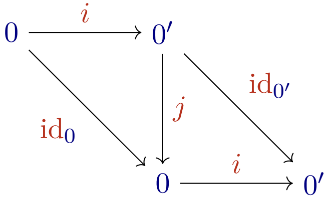
Theorem (Proposition 21)

Initial objects are unique up to isomorphism, that is, if 0 and $0'$ are initial objects in a category $\mathcal{C}$ then there exists a unique isomorphism $0 \xrightarrow{\cong} 0'$.

Initial and Terminal Objects

Proof.

Let 0 and $0'$ be initial objects in a category $\mathcal{C}$. Because 0 and $0'$ are initial objects we have that the following diagram commutes:



$$\begin{pmatrix} j \circ i = \mathrm{id}_0 \\ i \circ j = \mathrm{id}_{0'} \end{pmatrix}$$

That is, there is an unique isomorphism $i : 0 \xrightarrow{\cong} 0'$. ■

Initial and Terminal Objects

Theorem
Terminal objects are unique up to isomorphism.

Exercise 4
Prove the previous theorem.

Products

Products

Introduction

We shall introduce abstract characterisations of products (e.g. Cartesian products of sets and direct products of groups).

Binary Products

Example (Cartesian product in set theory)

(i) Let X and Y be sets. The **Cartesian product** of X and Y is defined by

$$X \times Y := \{ (x, y) \mid x \in X \land y \in Y \},$$

where the **ordered pair** (x, y) can be defined by

$$(x, y) := \{\{x, y\}, y\} \quad \text{(Kuratowski's definition)}$$

and it satisfies that

$$(x, y) = (x', y') \quad \text{iff} \quad x = x' \text{ and } y = y'.$$

(continued on next slide)

Binary Products

Example (Cartesian product in set theory (continuation))

(ii) Two **coordinate projections** on $X \times Y$ are defined by

$$\pi_1 : X \times Y \to X := (x, y) \mapsto x,$$
$$\pi_2 : X \times Y \to X := (x, y) \mapsto y,$$

where

$$c = (\pi_1\, c, \pi_2\, c), \quad \text{for all } c \in X \times Y.$$

(continued on next slide)

Binary Products

Example (Cartesian product in set theory (continuation))

(iii) Let $f : Z \to X$ and $g : Z \to Y$. The **pair $\boldsymbol{f}$ and $\boldsymbol{g}$** function is defined by

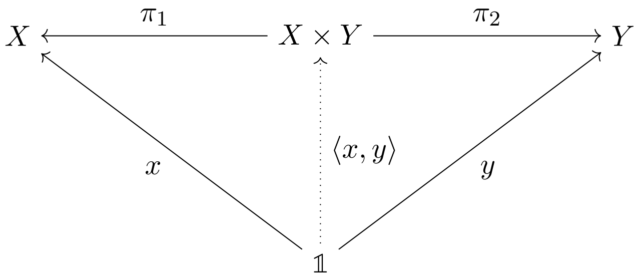
$$\langle f, g \rangle : Z \to X \times Y := z \mapsto (f\, z, g\, z).$$

(continued on next slide)

Binary Products

Example (Cartesian product in set theory (continuation))

(iv) We state the Cartesian product properties by saying that the following diagram commutes.



$$\begin{pmatrix} \pi_1 \circ \langle x, y \rangle = x \\ \pi_2 \circ \langle x, y \rangle = y \end{pmatrix}$$

Binary Products

Definition

Let A_1 and A_2 be objects in a category $\mathcal{C}$. A **binary product** of A_1 and A_2 is a triple (P, π_1, π_2), where P is an object in $\mathcal{C}$, denoted $A_1 \times A_2$, and π_1 and π_2 are two arrows

$$A_1 \xleftarrow{\pi_1} A_1 \times A_2 \xrightarrow{\pi_2} A_2,$$

such that for every object B and arrows

$$A_1 \xleftarrow{f_1} B \xrightarrow{f_2} A_2$$

there exists an unique arrow

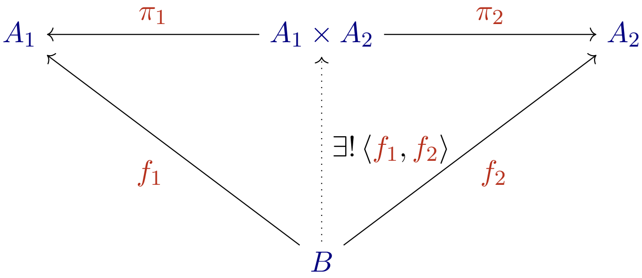
$$\langle f_1, f_2 \rangle : B \to A_1 \times A_2$$

such that the following diagram commutes (universal property):

(continued on next slide)

Binary Products

Definition (continuation)



$$\begin{pmatrix} \pi_1 \circ \langle f_1, f_2 \rangle = f_1 \\ \pi_2 \circ \langle f_1, f_2 \rangle = f_2 \end{pmatrix}$$

Binary Products

Example

- In **Set**, products are the Cartesian products.
- In **Pos**, products are Cartesian products with the product order.[†]
- In **Top**, products are Cartesian products with the product topology.

[†]The textbook uses 'pointwise order' instead of 'product order'.

Binary Products

Example

- In $\mathbf{Set}$, products are the Cartesian products.
- In $\mathbf{Pos}$, products are Cartesian products with the product order.†
- In $\mathbf{Top}$, products are Cartesian products with the product topology.

Exercise 5

Verify the previous claims (Exercise 19).

† The textbook uses 'pointwise order' instead of 'product order'.

Binary Products

Definition

A category $\mathcal{C}$ **has binary products** iff each pair of objects of $\mathcal{C}$ have a binary product.

Binary Products

Example

Since it possible to define the Cartesian product between any pair of sets, the category $\mathbf{Set}$ has binary products.

Binary Products

Example

Since it possible to define the Cartesian product between any pair of sets, the category $\mathbf{Set}$ has binary products.

Example

In a poset, seen as a category, products are (binary) greatest lower bounds (meets). This category has not binary products.

Binary Products

Exercise 6

Prove Proposition 27.

Exercise 7

Prove Proposition 28.

Ternary Products

Definition

Let A_1, A_2 and A_3 be objects in a category $\mathcal{C}$. A **ternary product** of A_1, A_2 and A_3 is a quadruple

$$(P, \pi_1, \pi_2, \pi_3),$$

where P is an object in $\mathcal{C}$, denoted $A_1 \times A_2 \times A_3$, and π_1, π_2, π_3 are arrows from $A_1 \times A_2 \times A_3$ to A_1, A_2, A_3, respectively, such that for every object B and arrows f_1, f_2, f_3 from B to A_1, A_2, A_3, respectively, there exists an unique arrow

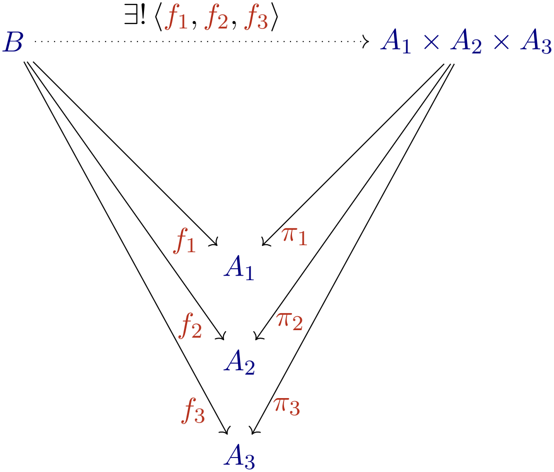
$$\langle f_1, f_2, f_3 \rangle : B \to A_1 \times A_2 \times A_3$$

such that the following diagram commutes (universal property):

(continued on next slide)

Ternary Products

Definition (continuation)



$$\begin{pmatrix} \pi_1 \circ \langle f_1, f_2, f_3 \rangle = f_1 \\ \pi_2 \circ \langle f_1, f_2, f_3 \rangle = f_2 \\ \pi_3 \circ \langle f_1, f_2, f_3 \rangle = f_3 \end{pmatrix}$$

Nullary Products

Remark

By removing the objects A_i (which also remove the projections π_i and the equations $\pi_i \circ \langle f_i \rangle = f_i$) from the binary (or ternary) products, we get the nullary products.

Nullary Products

Remark

By removing the objects A_i (which also remove the projections π_i and the equations $\pi_i \circ \langle f_i \rangle = f_i$) from the binary (or ternary) products, we get the nullary products.

Definition

A **nullary product** in a category $\mathcal{C}$ is an object P, such that for any object B, there is a unique arrow $B \to P$ (universal property).

Nullary Products

Remark

By removing the objects A_i (which also remove the projections π_i and the equations $\pi_i \circ \langle f_i \rangle = f_i$) from the binary (or ternary) products, we get the nullary products.

Definition

A **nullary product** in a category $\mathcal{C}$ is an object P, such that for any object B, there is a unique arrow $B \to P$ (universal property).

Remark

Note that the above object P is just a terminal object of $\mathcal{C}$.

Nullary Products

Exercise 8

What is the product of the empty family? (Exercise 29)

Finite Products

Definition

A category **has finite products** iff the category has products for all $n \in \mathbb{N}$.

Finite Products

Exercise 9

Show that if a category has binary and nullary products then it has finite products (Exercise 30).

General Products

Introduction

We shall generalise finite products to products of arbitrary objects.

General Products

Example (Cartesian product of a family of sets)

(i) Let $\{X_i\}_{i \in I}$ be a family of sets indexed by I. The **Cartesian product of the family of sets $\{X_i\}_{i \in I}$** is defined by

$$\prod_{i \in I} X_i = \left\{ f : I \to \bigcup_{i \in I} X_i \;\middle|\; \text{for all } i \in I, f\, i \in X_i \right\}.$$

General Products

Example (Cartesian product of a family of sets)

(i) Let $\{X_i\}_{i\in I}$ be a family of sets indexed by I. The **Cartesian product of the family of sets $\{X_i\}_{i\in I}$** is defined by

$$\prod_{i\in I} X_i = \left\{ f : I \to \bigcup_{i\in I} X_i \;\middle|\; \text{for all } i \in I, f\,i \in X_i \right\}.$$

(ii) For $i \in I$, the **ith-coordinate projection** map is defined by

$$\pi_i : \left(\prod_{j\in J} X_j\right) \to X_i := f \mapsto f\,i.$$

General Products

Definition

Let $\{A_i\}_{i\in I}$ be a family of objects in a category $\mathcal{C}$. A product for the family $\{A_i\}_{i\in I}$ is an object $\prod_{i\in I} A_i$ and arrows

$$\pi_i : \left(\prod_{i\in I} A_i\right) \to A_i$$

such that for every object B and arrows

$$f_i : B \to A_i$$

there exists an unique arrow

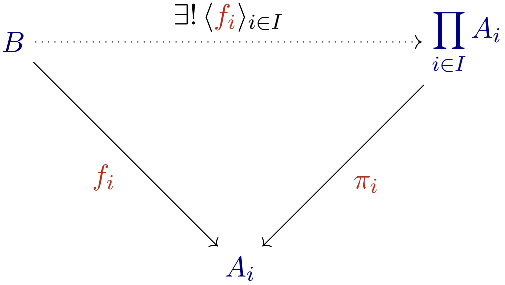
$$\langle f_i \rangle_{i\in I} : B \to \prod_{i\in I} A_i$$

(continued on next slide)

General Products

Definition (continuation)

such that, for $i \in I$, the following diagram commutes (universal property):



$$\left(\pi_i \circ \langle f_i \rangle_{i \in I} = f_i \right)$$

Coproducts

Coproducts

Introduction

We shall introduce abstract characterisations of disjoint unions (also called disjoint sums).

Binary Coproducts

Example (Disjoint union in set theory)

(i) Let X and Y be sets. The **disjoint union** of X and Y is defined by

$$\begin{aligned} X + Y &:= (\{1\} \times X) \cup (\{2\} \times Y) \\ &= \{\, (1, x) \mid x \in X \,\} \cup \{\, (2, y) \mid b \in Y \,\}. \end{aligned}$$

(continued on next slide)

Binary Coproducts

Example (Disjoint union in set theory (continuation))

(ii) Two **injections** for $X + Y$ are defined by

$$\mathsf{in}_1 : X \to X + Y := x \mapsto (1, x),$$
$$\mathsf{in}_2 : Y \to X + Y := y \mapsto (2, y).$$

(continued on next slide)

Binary Coproducts

Example (Disjoint union in set theory (continuation))

(iii) Let $f : X \to Z$ and $g : Y \to Z$. The **case $\boldsymbol{f}$ or $\boldsymbol{g}$** function is defined by

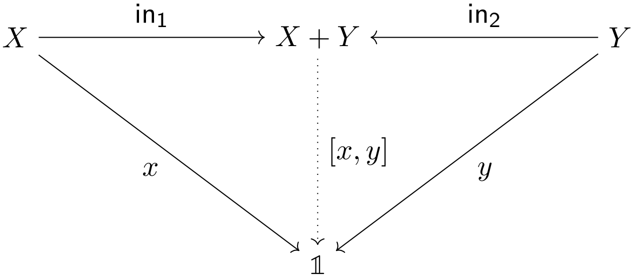
$$
\begin{aligned}
[f, g] &: X + Y \to Z \\
[f, g]\,(1, x) &:= f\,x, \\
[f, g]\,(2, y) &:= g\,x.
\end{aligned}
$$

(continued on next slide)

Binary Coproducts

Example (Disjoint union set theory (continuation))

(iv) We state the disjoint union properties by saying that the following diagram commutes.



$$\begin{pmatrix} [x, y] \circ \mathsf{in}_1 = x \\ [x, y] \circ \mathsf{in}_2 = y \end{pmatrix}$$

Binary Coproducts

Definition

Let A_1 and A_2 be objects in a category $\mathcal{C}$. A **binary coproduct** of A_1 and A_2 is a triple $(P, \mathsf{in}_1, \mathsf{in}_2)$, where P is an object in $\mathcal{C}$, denoted $A_1 + A_2$, and in_1 and in_1 are two arrows

$$A_1 \xrightarrow{\mathsf{in}_1} A_1 + A_2 \xleftarrow{\mathsf{in}_2} A_2,$$

such that for every object B and arrows

$$A_1 \xrightarrow{f_1} B \xleftarrow{f_2} A_2$$

there exists an unique arrow

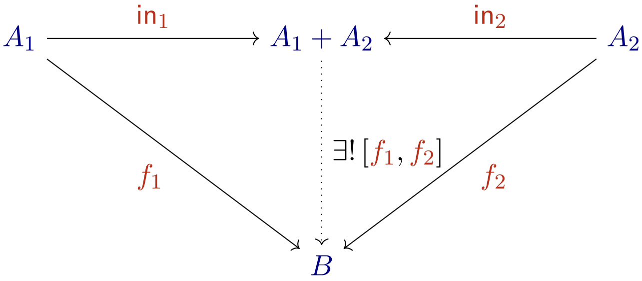
$$[f_1, f_2] : A_1 + A_2 \rightarrow B$$

such that the following diagram commutes (universal property):

(continued on next slide)

Binary Coproducts

Definition (continuation)



$$\begin{pmatrix} [f_1, f_2] \circ \mathsf{in}_1 = f_1 \\ [f_1, f_2] \circ \mathsf{in}_2 = f_2 \end{pmatrix}$$

Binary Coproducts

Example

- In $\mathbf{Set}$, disjoint unions are binary coproducts.
- In $\mathbf{Pos}$, disjoint unions are binary coproducts.
- In $\mathbf{Top}$, topological disjoint unions are binary coproducts.

Binary Coproducts

Example

- In $\mathbf{Set}$, disjoint unions are binary coproducts.
- In $\mathbf{Pos}$, disjoint unions are binary coproducts.
- In $\mathbf{Top}$, topological disjoint unions are binary coproducts.

Exercise 10

Verify the previous claims (Exercise 33).

Binary Coproducts

Example

In a poset, seen as a category, binary coproducts are (binary) least upper bounds (joins).

Binary Coproducts

Example

In a poset, seen as a category, binary coproducts are (binary) least upper bounds (joins).

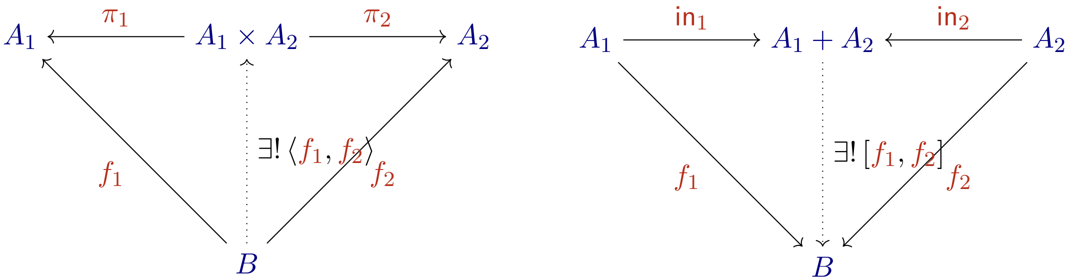
Remark

The previous example show that, a difference of the disjoint union in set theory, the binary coproduct between any pair of objects of a category may not exist.

Binary Coproducts

Duality

Binary products and binary co-products are dual notions.



References

References

Abramsky, S. and Tzevelekos, N. (2011). Introduction to Categories and Categorical Logic. In: New Structures for Physics. Ed. by Coecke, B. Vol. 813. Lecture Notes in Physics. Springer, pp. 3–94. DOI: 10.1007/978-3-642-12821-9_1 (cit. on p. 2).