Cartesian Closed Categories

Ronald Huidrom
MS19167

A Term Paper

Cartesian Closed Categories

by

Ronald Huidrom
Mathematics Major, MS19167
IISER Mohali.

This is a term paper for the course MTH432: Category Theory completed under and submitted to

Dr. Kapil Hari Paranjape
Department of Mathematical Sciences
IISER Mohali,

during the fall semester of 2022.

Cartesian Closed Categories

Ronald Huidrom

Abstract

Function spaces in the category **Set** can be generalized to exponential objects in more general categories. This gives rise to cartesian closed categories in which any morphism defined on a product of two objects can be naturally identified with a morphism defined on one of the factors. These categories are particularly important in mathematical logic and the theory of programming, in that their internal language is the simply typed λ-calculus.

Introduction

Using the concept of finite products and exponential objects, one can define the notion of a cartesian closed category. As formal systems, these categories have the same expressive power as a typed λ-calculus. We shall assume the knowledge of some basic category theory, particularly the notion of finite products.

In section 1, we define the concept of exponential objects in categories. They are generalizations of function spaces from the category of sets. In section 2, we define cartesian closed categories and look at some interesting examples. In section 3, we look at the applications of cartesian closed categories in various areas, particularly in computer science.

1 Exponentials

In the category **Set** of sets, we use the notation C^B to denote the set of all functions from B to C. That is, $C^B := \mathbf{Set}(B, C)$, the set of all morphisms from B to C in **Set**. Let $f : A \times B \to C$ be a morphism in **Set**. If we hold some $a \in A$ fixed, we have a morphism $f_a : B \to C$ taking $y \in B$ to $f(a, y) \in C$ so that $f_a \in C^B$. Thus, we have a morphism $\tilde{f} : A \to C^B$ which takes $x \in A$ to $f_x \in C^B$. The morphism $\tilde{f}$ is uniquely determined by the equation

$$\tilde{f}(x)(y) = f_x(y) = f(x, y).$$

Conversely, for any $\tilde{f} : A \to C^B$, there is a unique morphism $f : A \times B \to C$ given by the equation

$$f(x, y) = f_x(y) = \tilde{f}(x)(y).$$

Thus, we have established an isomorphism of Hom-sets:

$$\mathbf{Set}(A \times B, C) \cong \mathbf{Set}(A, C^B).$$

The bijective correspondence between morphisms of the form $f : A \times B \to C$ and those of the form $\tilde{f} : A \to C^B$ is mediated by a certain operation, which we will call the *evaluation* map. In **Set**, we define the evaluation function

$$\text{eval } : C^B \times B \to C$$

which takes (f, y) to $f(y)$. This hardly says anything more than the fact that we are evaluating the map $f : B \to C$ at $y \in B$. However, written in this way, it is more convenient to see that the evaluation map has the universal mapping property: given any set A and any map $f : A \times B \to C$, there exists a unique map $\tilde{f} : A \to C^B$ such that $\text{eval } \circ (\tilde{f} \times 1_B) = f$. In other words, $\text{eval } (\tilde{f}(a), b) = f(a, b)$.

The idea may be extended to any category having binary products. In **Set**, the function space C^B is an object having the complete information of all morphisms from B to C. In a general category having binary products, we have an object C^B conveying information of morphisms from B to C in some way. Indeed, we may define

1.1 Definition. *Let $\mathcal{A}$ be a category having binary products. Let B and C be two objects of $\mathcal{A}$. The exponential of B and C is an object C^B and a morphism (called evaluation) $\epsilon : C^B \times B \to C$ such that, for any object A and a morphism $f : A \times B \to C$ there exists a unique morphism $\tilde{f} : A \to C^B$ such that $\epsilon \circ (\tilde{f} \times 1_B) = f$.*

We call $\tilde{f}$ the conjugate of f. If $g : A \to C^B$ is a morphism, we define $\bar{g} := \epsilon \circ (g \times 1_B) : A \times B \to C$. By the uniqueness clause of the definition, we have $\tilde{\bar{g}} = g$ and for any $f : A \times B \to C$, we have $\bar{\tilde{f}} = f$. Thus, there is a one-to-one correspondence between morphisms of the form $f : A \times B \to C$ and those of the form $\tilde{f} : A \to C^B$. Indeed, we have an isomorphism of Hom-sets:

$$\mathcal{A}(A \times B \to C) \cong \mathcal{A}(A, C^B).$$

2 Cartesian closed categories

2.1 Definition (cartesian closed category)**.** *A category is said to be cartesian closed if it has all finite products and exponentials.*

As we have seen above, **Set** is cartesian closed. Consider the category **FinSet** of finite sets. Clearly, it has all finite products, which are just the cartesian products. Also, it has all the exponentials, since given any two finite sets A and B, we have the finite function space B^A for $|B^A| = |B|^{|A|}$.

Consider the category **Set**. Let $\beta : B \to C$ be a set map. We define $\beta^A : B^A \to C^A$ sending $f : A \to B$ to $\beta \circ f : A \to C$. It is easy to see that this assignment gives a functor $F : \textbf{Set} \to \textbf{Set}$. Indeed, we only need to check if F preserves composition and identity morphisms: for any $\alpha : C \to D$

$$\begin{aligned}(\alpha \circ \beta)^A(f) &= \alpha \circ \beta \circ f \\ &= \alpha \circ \beta^A(f) \\ &= \alpha^A \circ \beta^A(f),\end{aligned}$$

so that $(\alpha \circ \beta)^A = \alpha^A \circ \beta^A$. Also,

$$(1_B)^A(f) = 1_B \circ f = f = 1_{B^A}(f).$$

so that $(1_B)^A = 1_{B^A}$. This shows that F is indeed a functor. In fact, this is true in a more general setting as stated (without proof) in the following proposition.

2.2 Proposition. *Let $\mathcal{A}$ be a cartesian closed category. Then the exponentiation by a fixed object A is an endofunctor $(-)^A : \mathcal{A} \to \mathcal{A}$.*

2.3 Proposition. *The category* **Pos** *of posets is cartesian closed.*

Proof. In the category **Pos** of posets, the morphisms are the monotone functions $f : P \to Q$. That is, $p \leq p'$ implies $fp \leq fp'$. Let P and Q be two posets. The product poset $P \times Q$ has pairs (p, q) as elements and is partially ordered by

$$(p, q) \leq (p', q') \text{ iff } p \leq p' \text{ and } q \leq q'.$$

It is easy to see that the projections $\pi_1 : P \times Q \to P$ and $\pi_2 : P \times Q \to Q$ are monotone. Also, if $f : X \to P$ and $g : X \to Q$ are monotone, the pairing $(f, g) : X \to P \times Q$ is also monotone.

We define the exponential Q^P as follows

$$Q^P = \{f : P \to Q \mid f \text{ monotone}\}.$$

It is easy to see that Q^P is again a poset ordered pointwise. That is,

$$f \leq g \text{ iff } fp \leq gp \text{ for all } p \in P.$$

The evaluation map $\epsilon : Q^P \times P \to Q$ and the transposition $\tilde{f} : X \to Q^P$ of some given morphism $f : X \times P \to Q$ are the usual ones of the underlying functions. It suffices to show that these are monotone.

Suppose $(f, p) \leq (f', p')$ in $Q^P \times P$. We have

$$\epsilon(f, p) = f(p) \leq f(p') \leq f'(p') = \epsilon(f', p'),$$

so that ϵ is monotone. Suppose $f : X \times P \to Q$ is monotone and let $x \leq x'$. It suffices to show that $\tilde{f}(x) \leq \tilde{f}(x')$ in Q^P, that is, $\tilde{f}(x) \leq \tilde{f}(x')(p)$ for all $p \in P$. But then $\tilde{f}(x)(p) = f(x, p) \leq f(x', p) = \tilde{f}(x')(p)$. □

The following are some other examples of cartesian closed categories.

1. The category **Grph** of graphs whose morphisms are homomorphisms of graphs is cartesian closed.
2. Let $\mathcal{A}$ be a small category. The category $\mathbf{Func}(\mathcal{A}, \mathbf{Set})$ whose objects are functors and morphsims are natural transformations is cartesian closed.
3. The category **Cat** of small categories and functors is cartesian closed. Given two small categories $\mathcal{A}$ and $\mathcal{B}$, the exponential $\mathcal{B}^{\mathcal{A}}$ is the category whose objects are functors from $\mathcal{A}$ to $\mathcal{B}$ and whose morphisms are natural transformations between them.
4. A functional programming language can be regarded as a category. If such a language is cartesian closed, then for any types A and B, there is a type B^A of functions from A to B. Since that is a type, one can apply programs to data of that type. This is the same as saying that functions are on the same level as other data, which is often described by saying "functions are first class objects".
5. A Heyting algebra is a poset with all finite infimums and supremums that is cartesian closed as a category. A Heyting algebra is a generalization of a Boolean algebra. Heyting algebras correspond to intuitionistic logic in the way that Boolean algebras correspond to classical logic.

3 Applications

Lambda calculus (also λ-calculus) is a formal system in mathematical logic for expressing computation based on function abstraction and application using variable binding and substitution. It can be used to simulate any Turing machine. Lambda calculus may be untyped or typed. In typed lambda calculus, functions can be applied only if they are capable of accepting the given input's "type" of data. A typed λ-calculus is a typed formalism that uses the λ symbol to denote anonymous function abstraction. Typed λ-calculus is equivalent to cartesian closed categories. The Curry–Howard–Lambek correspondence provides a deep isomorphism between intuitionistic logic, simply-typed lambda calculus and cartesian closed categories.

A topos (plural topoi) is a category that behaves like the category of sheaves of sets on a topological space. It behaves much like the category of sets and possess a notion of localization; they are a direct generalization of point-set topology. Topoi are cartesian closed categories; they have been proposed as a general setting for mathematics, instead of traditional set theory.

Function-level programming is a variable-free programming paradigm advocated by the computer scientist John Backus. It bears some similarity to the internal language of cartesian closed categories.

CAML (orginally an acronym for Categorical Abstract Machine Learning) is a multi-paradigm, general-purpose programming language which is a dialect of the ML programming language family. It is more consciously modelled on cartesian closed categories.

References

[1] Barr, M,; Wells, C. *Category theory for computing science*; Prentice Hall: London, 1995.

[2] Awodey, S. *Introduction to category theory*; Oxford University Press: Oxford, 2010.