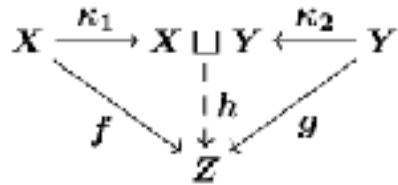


圏論は数学をするための高級言語

圏論 category theory は、集合の代わりに、**対象**と**射**を使う数学のコトバ



X、Y、Z、 $X \sqcup Y$ というのが対象 objects で、その間に描いてある矢印 arrows が射 morphism.

圏論ではこの図を、

- ・ X、Y が与えられたとき、
- ・ 特別な $X \sqcup Y$ と、 κ_1 、 κ_2 がとれる。
- ・ どう特別かという、ほかに Z と f、g というものが現れたとき、
- ・ 図を可換にする ($h \circ \kappa_1 = f$ 、 $h \circ \kappa_2 = g$) ような h がただ1 つ定まる。

と読む。

これは集合の排他的和 $X \sqcup Y$ の、集合と関数の圏 Sets における圏論的特徴付けだ。

排他的和とは、集合の合併をただし重複がないようにとるという意味で、例えばラベル1、2 を付けて、

$$X \sqcup Y = \{(1, x) \mid x \in X\} \cup \{(2, y) \mid y \in Y\}$$

というようにすると、 κ_1 、 κ_2 は X、Y を $X \sqcup Y$ に**埋め込む**関数で、h は f と g から**場合分け** ($X \sqcup Y$ の元が X 由来だったら f を適用、Y 由来だったら g を適用で)つくられる。

圏論の基本的アイデアは：

“あるモノについて調べるとき、そのモノの**成り立ち**を考えるのではなく、そのモノと他のモノの間の**作用**や**関係性**を考える”

排他的和 $X \sqcup Y$ の例でいうと、その集合としての成り立ち (元 (1, x) や (2, y) があって.....) は無視して、「f と g が h を引き起こして.....」と考える。

ここでは圏の対象 X、Y.....が集合、射 f、g、h.....が関数になっている。前者がモノ、後者が作用である。

関数型プログラムを扱う圏 Types がある。

プログラムが扱う int や bool などのデータの型が対象で、その間のプログラムが射(矢印)だ。

このとき、Sets の $X \sqcup Y$ に対応するのが、c言語のような関数型プログラミングでいうところの直和型である。

この他にも、考えたい対象や射を換えて、

- ・ ブール代数とその準同型写像の圏 Bool、
- ・ 位相空間と連続写像の圏 Top、
- ・ 圏とその間の写像(関手functorと呼ぶ)の圏 Cat など、

いろんな圏がある。

対象と射が変わっても、圏論のコトバは共通して使える。

あたかも「 n を自然数とする。このとき……」と書くように、「 \mathcal{C} を圏とする。このとき……」と書く。

圏論は、数学のなかでも異なる分野をつなぐ抽象言語の役割をする。

“対象、射としてとる概念の抽象度をいろいろ変えることによって、

その局面局面でフォーカスしたい抽象度にぴったりの数学的コトバが提供される”

集合のコトバでは、要素ベースでいちばん下のレベルからすべてのディテールを積み上げていかなければいけないところを、圏論のコトバを使えば、適切な圏を選ぶことで「いままさに気になっているレベルの構造」だけをササッと書けるのだ。

東大情報科学科案内ページの解説から