

# The Haskell Programmer's Guide to the IO Monad

— Don't Panic —

## 第 1 回

下村翔

平成 20 年 4 月 18 日

### 3 Categories

#### 3.1 Categories in theory

カテゴリを、集合とその集合上の全称関数 (いわゆる射) と一緒になったものとする。

- 関数  $f$  が、集合  $A$  から集合  $B$  にマップを行うとき、 $f: A \rightarrow B$  と書く
- $f: A \rightarrow B$  かつ  $g: B \rightarrow C$  ならば、2つの関数を合成することができ、 $f \circ g$  と書く
- 集合  $A$  に対し、恒等関数  $id_A: A \rightarrow A$  が存在する

ただし一般の圏論では、対象が集合や全称関数に制限されない。

##### 3.1.1 Definition

カテゴリ  $C = (O_C, M_C, T_C, \circ_C, Id_C)$  は、射  $M_C$ 、オブジェクト  $O_C$ 、合成  $\circ_C$ 、型情報  $T_C$  からなる。これらは次の制約に従う。

1. オブジェクトの集まりを集合と仮定する。オブジェクト自身も集合であることがよくあるが、圏論の世界では特に気にしない
2. 本書では、射の集まりも集合と仮定する
3. 3項関係  $T \subseteq (M \times O \times O)$  は  $C$  の型情報と呼ばれる。すべての射に型を付けたいので、カテゴリは以下を要求する。

$$\forall f \subseteq M \ \exists A, B \subseteq O \ (f, A, B) \subseteq T \quad (1)$$

$(f, A, B) \in T_C$  を  $f: A \xrightarrow{C} B$  と書く。型の一意性を確保するために、以下を要求する。

$$f: A \rightarrow B \wedge f: A' \rightarrow B' \Rightarrow A = A' \wedge B = B' \quad (2)$$

4. 部分関数  $\circ : M \times M \rightarrow M$  は、 $C$  での合成と呼ばれる。1 つめの関数のターゲットと 2 つめの関数のソースが等しければ、常に合成を行うことができる。

$$f : A \rightarrow B \wedge g : B \rightarrow C \Rightarrow g \circ f : A \rightarrow C \quad (3)$$

関数合成は結合律を満たす

5. すべてのオブジェクトは一意的恒等射  $id_A$  と関連づけられる。

$$Id : O \rightarrow M \quad (4)$$

$$A \mapsto id_A \quad (5)$$

恒等射の型は

$$\forall A \in O \quad id_A : A \rightarrow A \quad (6)$$

恒等射は名前の通り、左から合成しても右から合成しても中立である。

$$f \circ id_{src\ f} = f = id_{tgt\ f} \circ f \quad (7)$$

**Another example** 半順序な集合  $A$  はカテゴリを作る。これを  $Q$  と呼ぶ。

$$O_Q := A \quad (8)$$

$$M_Q := (x, y) | x, y \in A \wedge x \leq y \quad (9)$$

$Q$  がカテゴリとなるように、型情報  $T_Q$ 、合成  $\circ_Q$ 、恒等射  $Id_Q$  を定義せよ。

### 3.2 Spot a category in Haskell

本書では、Haskell でのカテゴリを考える。これを  $H$  と呼ぶ。カテゴリ  $H$  のオブジェクト  $O_H$  は Haskell の型であり、射  $M_H$  は Haskell の 1 引数関数、関数のシグニチャは  $H$  の型情報  $T_H$  となる。Haskell での関数合成は、 $\circ_H$  となる。Haskell での恒等関数は

$$id :: forall a. a \rightarrow a \quad (10)$$

だが、これは

$$\forall A \in O_H \quad id_A : A \rightarrow A \quad (11)$$

に対応する。

ここで、任意の引数の射に対して話をしているわけではないことに注意する必要がある。Haskell ではよくカーリー化が行われるが、本書ではカーリー化された関数を射とはみなさず、オブジェクトとみなす。