The Haskell Programmer's Guide to the IO Monad
— Don't Panic —
第 3 回

下村翔

平成 20 年 5 月 2 日

5 Natural Transformations

今日のテーマは Natural Transformation(自然変換) です。難しいです。

5.1 Natural transformations in theory

Natural transformation(自然変換) とは、そのカテゴリのオブジェクトに影響を与えたり与えられたりせずに、ある構造から別の構造への変換を行うことである。

2 つのカテゴリ $\mathcal{A}$ と $\mathcal{B}$ の間の 2 つのファンクタ F と G を考える。自然変換は、2 つのファンクタのカテゴリ $\mathcal{B}$ での射の集まりであり、オブジェクト FA をオブジェクト GA にマップする。

5.1.1 Definition

$\mathcal{A}$ と $\mathcal{B}$ をカテゴリとし、F と G を $\mathcal{A} \to \mathcal{B}$ のファンクタとする。このとき関数

$$\begin{aligned} \eta : \mathcal{O}_{\mathcal{A}} &\to \mathcal{M}_{\mathcal{B}} \\ A &\mapsto \eta_{\mathcal{A}} \end{aligned}$$

を

$$\forall A \in \mathcal{O}_{\mathcal{A}} \;\; \eta_{\mathcal{A}} : FA \underset{\mathcal{B}}{\to} GA$$

でありかつそのときに限り、F から G への transformation(変換) と呼ぶ。また、以下の条件を満たす時は特に自然変換と呼ばれ、$\eta : F \stackrel{.}{\to} G$ と書く。

$$\forall f : A \underset{\mathcal{A}}{\to} B \;\; \eta_B \circ_{\mathcal{B}} Ff = Gf \circ_{\mathcal{B}} \eta_A$$

この定義は、自然変換はオブジェクト上の射の動作を変えない構造 F から構造 G への変換であるとしている。そのため、射 f が他の構造の上の射に持ち上げられていたとしても問題ない。

5.2 Natural transformations in Haskell

まず最初に、ソースとターゲットの型が同じでしかもポリモーフィックであるような 1 引数関数は変換であることに注意する。例えば Haskell の Just は

$$\text{Just} \ :: \ \text{forall } a.\ a \rightarrow \text{Maybe } a$$

Just : $\mathcal{O}_{\mathcal{H}} \rightarrow \mathcal{M}_{\mathcal{H}}$ と考えれば、オブジェクト $A \in \mathcal{O}_{\mathcal{H}}$ への Just の適用は、型変数 a にある Haskell の型 A を束縛するという意味になる。つまり、Just はオブジェクト A を $A \rightarrow$ Maybe A という射へとマップするということである。

変換について考えるためには、Just のソース側のファンクタが必要になる。そこで恒等ファンクタ $I_{\mathcal{H}}$ を考える。そうすると

$$\text{Just} : I_{\mathcal{H}} \rightarrow \text{Maybe}$$

となる。

5.2.1 Lemma

Just : $I_{\mathcal{H}} \stackrel{.}{\rightarrow}$ Maybe

5.2.2 Prrof

$f :: A \rightarrow B$ を任意の Haskell の関数とする。そのときに以下の式が証明できればよい。

$$\text{Just} \ . \ I_{\mathcal{H}} \ f \ \ == \ \ \text{fmap} \ f \ . \ \text{Just}$$

ここで、左側の Just は η_B を指し、右側のものは η_A を指す。証明は、Maybe の fmap の定義から明らかである。

他の例として maybeToList と listToMaybe を考える。これらの定義は以下の通り。

```
maybeToList :: forall a. Maybe a -> [a]
maybeToList Nothing = []
maybeToList (Just x) = [x]

listToMaybe :: forall a. [a] -> Maybe a
listToMaybe [] = Nothing
listToMaybe (x:xs) = Just x
```

listToMaybe を適用すると情報が減るが、このことは naturality とは関係がないので注意する。

5.2.3 練習問題

maybeToList と listToMaybe の naturality を証明してみよう。
maybeToList の場合は maybeToList . fmap f == fmap f . maybeToList を言えばよい。左辺の fmap は Maybe の、右辺の fmap は List のファンクタである。

Nothing の場合

$$
\begin{aligned}
& (\text{maybeToList . fmap } f) \text{ Nothing} \\
\Rightarrow\ & \text{maybeToList Nothing} \\
\Rightarrow\ & []
\end{aligned}
$$

$$
\begin{aligned}
& (\text{fmap } f \text{ . maybeToList}) \text{ Nothing} \\
\Rightarrow\ & \text{fmap } f\ [] \\
\Rightarrow\ & []
\end{aligned}
$$

Just x の場合

$$
\begin{aligned}
& (\text{maybeToList . fmap } f)\ (\text{Just } x) \\
\Rightarrow\ & \text{maybeToList (Just } (f\ x)) \\
\Rightarrow\ & [f\ x]
\end{aligned}
$$

$$
\begin{aligned}
& (\text{fmap } f \text{ . maybeToList})\ (\text{Just } x) \\
\Rightarrow\ & \text{fmap } f\ [x] \\
\Rightarrow\ & [fx]
\end{aligned}
$$

5.3 Composing transformations and funcctors

モナドについて説明する前に、自然変換とファンクタを組み合わせて新しい別の自然変換を作れるようにする必要がある。

定義 4.14 でファンクタを合成したように、自然変換とファンクタを合成することができる。オブジェクトははじめにファンクタにより持ち上げられ、変換によって射にマップされる、あるいは先に射にマップされてから持ち上げられる。

5.3.1 Definition

$\mathcal{A}$,$\mathcal{B}$,$\mathcal{C}$,$\mathcal{D}$ をカテゴリとし、$E : \mathcal{A} \to \mathcal{B}$, $F, G : \mathcal{B} \to \mathcal{C}$, $H : \mathcal{C} \to \mathcal{D}$ をファンクタとする。ここで自然変換 $\eta : F \stackrel{.}{\to} G$ を考えると、変換 ηE と $H\eta$ を定義することができ、

$$
\begin{aligned}
(\eta E)A &:= \eta_{EA} \\
(H\eta)B &:= H(\eta_B)
\end{aligned}
$$

である。これにより、あいまいさなしで ηEA や $H\eta B$ と記述することができる。

5.3.2 Lemma

定義 5.3.1 のもとで、

$$
\begin{aligned}
H\eta : HF &\stackrel{.}{\to} HG \\
\eta E : FE &\stackrel{.}{\to} GE
\end{aligned}
$$

が保たれる。つまり、$H\eta$ と ηE はどちらも自然変換になっている。

5.3.3 Proof

Part I

$$\begin{aligned}
& \eta : F \dot{\to} G, \\
\Rightarrow\ & \forall f : A \underset{\mathcal{B}}{\to} B \ \ \eta B \circ Ff = Gf \circ \eta A \\
\Rightarrow\ & \forall f : A \underset{\mathcal{B}}{\to} B \ \ H(\eta B \circ Ff) = H(Gf \circ \eta A) \\
\Rightarrow\ & \forall f : A \underset{\mathcal{B}}{\to} B \ \ H(\eta B) \circ H(Ff) = H(Gf) \circ H(\eta A) \\
\Rightarrow\ & \forall f : A \underset{\mathcal{B}}{\to} B \ \ (H\eta)B \circ (HF)f = (HG)f \circ (H\eta)A \\
\Rightarrow\ & H\eta : HF \dot{\to} HG
\end{aligned}$$

Part II

$$\begin{aligned}
& \eta : F \dot{\to} G \\
\Rightarrow\ & \forall f : A \underset{\mathcal{B}}{\to} B \ \ \eta B \circ Ff = GF \circ \eta A \\
\Rightarrow\ & \forall f : A \underset{\mathcal{A}}{\to} B \ \ \eta(EB) \circ F(Ef) = G(Ef) \circ \eta(EA) \\
\Rightarrow\ & \forall f : A \underset{\mathcal{A}}{\to} B \ \ (\eta E)B \circ (FE)f = (GE)f \circ (\eta E)A \\
\Rightarrow\ & \eta E : FE \dot{\to} GE
\end{aligned}$$

自然変換をお互いに組み合わせられるからといって、単に次々に適用することはできない。組み合わせはコンポーネント順 (component wise) に行われ、vertical composition とも呼ばれる。

5.3.4 Definition

$\mathcal{A}$,$\mathcal{B}$ をカテゴリとし、$F, G, H : \mathcal{A} \to \mathcal{B}$、$\eta : F \dot{\to} G, \mu : G \dot{\to} H$ として変換を以下のように定義する。

$$\begin{aligned}
id_F : \mathcal{O}_{\mathcal{A}} &\to \mathcal{M}_{\mathcal{B}} \\
A &\mapsto id_{FA}
\end{aligned}$$

$$\begin{aligned}
\mu\eta : \mathcal{O}_{\mathcal{A}} &\to \mathcal{M}_{\mathcal{B}} \\
A &\mapsto \mu A \circ \eta A
\end{aligned}$$

5.3.5 Lemma

定義 5.3.4 において以下が保たれる。

$$id_F : F \dot{\to} F,$$

$$\mu\eta : F \dot{\to} H,$$

$$id_G\eta = \eta = \eta id_F$$

5.3.6 Proof

omitted :)