The Haskell Programmer's Guide to the IO Monad – Don't Panic – 第4回

今井宣洋

平成 20 年 6 月 12 日

概 要

この文章は [] の 6 章を読んで、まとめたものである、圏論特有の用語に慣れるため、本名に加え、日本語訳として使われる言葉をカギカッコ「」で付記した。この文章達に、間違い、もしくは不適切な点があれば、教えていただけると幸いである。

6 章 Monads「モナド」

6.1 圏論における Monad「モナド」とは

定義 1

$\mathcal{C}$ をカテゴリ, $F : \mathcal{C} \to \mathcal{C}$ をファンクタ, $\eta : I_{\mathcal{C}} \Rightarrow F$, $\mu : F^2 \Rightarrow F$ を *natural transformations*「自然変換」とする。このとき三つ組み (F, η, μ) が *monad*「モナド」であるとは次が成り立つ事である。

$$\mu(F\mu) = \mu(\mu F)$$

$$\mu(F\eta) = id_F = \mu(\eta F)$$

この等式は何を表しているんだろう。わかんないね。今日はこの 2 つの等式についてじっくり考えていこう！

このカッコの中身は関数適用じゃない事に注意しよう。だって μ とかはナチュラルトランスフォーメイションだもんね。

二つの transformationη, μ はある意味、反対だね。よくみてごらん η は F の構造が 1 レベル付加されているけど、μ は 1 レベル下がってるもんね。

[set category の例は何を言っているのか謎なので略]

改めてもう一度定義１を見てみよう。一つ目の等式がどういう意味なのかは言ってなかったけど、実は次のような意味なんだ。

$$\forall A \in \mathcal{O},\quad \mu_A \circ F(\mu_A) = \mu_A \circ \mu_{F(A)}$$

そう、ここで　$F(\mu_A)$ と $\mu_{F(A)}$ ってなんだろう？図を書くとわかりやすいよ。［ホワイトボード参照］

μ_A は F のネストを１レベル下げる morphism だし、$F(\mu_A)$ も同様だね。でも μ_A は F の２重構造以上になっているものが適用できるのに対して、$F(\mu_A)$ は３重構造以上になってるものが適用できるってところがちょっとちがうね。

定義１の二つ目の等式も見てみよう。これも今ちゃんと説明するけど、実は次のような意味なんだ。

$$\forall A \in \mathcal{O},\quad \mu_A \circ F(\eta_A) = id_{F(A)} = \mu_A \circ \eta_{F(A)}$$

これも図を書いてみよう［ホワイトボード参照］

6.2 Haskell category における Monad とは

前節の説明でみんなモナドの定義はばっちりだね。でも具体例があったほうが理解が深まるから、念のため Haskell カテゴリでの例を見てみよう。

list 構造を考えるのが、Haskell カテゴリでの monad に一番わかりやすいよ。

Haskell カテゴリー $\mathcal{H}$ において、$L : \mathcal{H} \to \mathcal{H}$ のリストファンクターと仮定しよう。

定義 2
二つの *transformations*「変換」μ, η を次で定義する。

$$\mu := \text{型}\ \mathtt{A} \mapsto \mathtt{concat} \qquad (\mathtt{concat} :: [[\mathtt{A}]] \to [\mathtt{A}])$$

$$\eta := \text{型}\ \mathtt{A} \mapsto \mathtt{return} \qquad (\mathtt{return} :: \mathtt{A} \to [\mathtt{A}])$$

補題 1
μ も η も *natural transformation* であり、かつ三つ組み (L, η, μ) は *monad*「モナド」となる。

[証明デモ] ■