Monads of expressions[1]

Vladimir Voevodsky[2,3]

November 2015

Abstract

1 Introduction

One defines, following Aczel [1], [2] and Hurschewitz-Maggesi [3, p.228] a binding arity as a finite sequence (or list) of natural numbers. Let $\mathbf{N}^*$ be the set of binding arities. A binding signature is a pair $\Sigma = (Op, Ar : Op \to \mathbf{N}^*)$ where Op is a set.

For any binding signature Σ and any Grothendieck universe U we will constructs a monad R_Σ on the category of U-sets such that $R_\Sigma(X)$ is the set of α-equivalence classes of expressions with free variables from X. First some preliminaries.

2 Preliminaries

Let U be a Grothendieck universe. Let C be a category in U.

Consider the following structure that is called a Kleisli triple in [4].

Definition 2.1 *[**205.11.14.def1**] A Kleisli triple on C is a collection of data of the form:*

1. *a mapping $R : Ob(C) \to Ob(C)$,*
2. *for each X in C a morphism $\eta(X) : X \to R(X)$,*
3. *for each X, Y in C and $f : X \to R(Y)$ a morphism $\rho(f) : R(X) \to R(Y)$,*

such that the following conditions hold:

1. *for any $X \in C$, $\rho(\eta(X)) = Id_{R(X)}$,*
2. *for any $f : X \to R(Y)$, $\eta(X) \circ \rho(f) = f$,*
3. *for any $f : X \to R(Y)$, $g \to R(Z)$,*

$$\rho(f) \circ \rho(g) = \rho(f \circ \rho(g))$$

It turns out that Kleisli triples are equivalent to monads see, e.g., [3, p.219]. We want to have a precise statement of this equivalence.

---

[1] *2000 Mathematical Subject Classification: 18D99, 08C99, 03B15 03F50,*

[2] School of Mathematics, Institute for Advanced Study, Princeton NJ, USA. e-mail: vladimir@ias.edu

[3] Work on this paper was supported by NSF grant 1100938.

Problem 2.2 *[***2015.11.14.prob1***] To construct a function MK from monads on C to Kleisli triples on C.*

Construction 2.3 *[***2015.11.14.constr1***]*Given a monad $\mathbf{R} = (R, \eta, \mu)$ we define the corresponding Kleisli triple as the triple $(R_{Ob}, \eta, \rho_{MK(\mathbf{R})})$ where

$$\rho_{MK(\mathbf{R})}(f) = R_{Mor}(f) \circ \mu(Y).$$

Verification of the equations is simple.

Problem 2.4 *[***2015.11.14.prob2***] To construct a function KM from Kleisli triples on C to monads on C.*

Construction 2.5 *[***2015.11.14.constr2***]*Let $\mathbf{R} = (R_{Ob}, \eta, \rho)$ be a Kleisli triple on C. To define the functor underlying the corresponding monad we take $R_{Ob} = R_{Ob}$ and define R_{Mor} by the rule

$$R_{Mor}(f) = \rho(f \circ \eta(Y))$$

verification of functor axioms is simple.

To define η of the monad we set it equal to the η of the Kleisli triple.

To define μ we set

$$\mu(X) = \rho(Id_{R_{Ob}(X)})$$

Verification of the equations that form the axioms of a monad is simple.

Let $Monads$ be the set of monads on a category C that lies in a Grothendieck universe U and let $KTriples$ be the set of Kleisli triples in the same category.

Lemma 2.6 *[***2015.11.14.l1***] One has*

$$MK \circ KM = Id_{Monads}$$

$$KM \circ MK = Id_{KTriples}$$

Proof: Given a monad $\mathbf{R} = (R_{Ob}, \eta, \mu)$ on C we have for $KM(MK(\mathbf{R}))$:

1. $KM(MK(\mathbf{R}))_{Ob} = MK(\mathbf{R})_{Ob} = R_{Ob}$,
2. for a morphism $f : X \to Y$ we have

   $$KM(MK(R))_{Mor}(f) = \rho_{MK(R)}(f \circ \eta_{MK(R)}(Y)) = R_{Mor}(f \circ \eta_R(Y)) \circ \mu_R(Y) = f \circ \eta_R(Y) \circ \mu_R(Y) = f \circ Id_Y = f$$

Since two monads are equal when the underlying functors are equal we conclude that

$$KM(MK(\mathbf{R})) = \mathbf{R}$$

Given a Kleisli triple $\mathbf{R} = (R_{Ob}, \eta, \rho)$ we have for $MK(KM(\mathbf{R}))$:

1. $KM(MK(R))_{Ob} = R_{Ob}$,

2. for a morphism $f : X \to R(Y)$ we have

$$\rho_{MK(KM(\mathbf{R}))}(f) = KM(\mathbf{R})_{Mor}(f) \circ \mu_{KM(\mathbf{R})}(Y) = \rho_{\mathbf{R}}(f \circ \eta_{\mathbf{R}}(Y)) \circ \rho(Id_{R_{Ob}(X)}) = \rho(f \circ \eta_{\mathbf{R}}(Y) \circ \rho(Id_{R_{Ob}(X)})) = \rho$$

Since two Kleisli triples are equal when the corresponding functions on objects and the ρ function are equal we conclude that $MK(KM(\mathbf{R})) = Id$.

Lemma 2.6 shows that (KM, MK) defines a pair of mutually inverse bijections between the sets of monads on C and Kleisli triples on C for any Grothendieck universe U and any category C in U. It also has its version in the UniMath language where it establishes the same fact for any type-theoretic universe U and monads and Kleisli triples on any pre-category C.

3 Monad defined by a binding signature

One construct a monad corresponding to a binding signature without any mention of syntactic expressions. To do it one proceeds as follows.

Problem 3.1 [**2015.11.22.prob1**] *Let $\mathbf{R} = (R, \eta, bind)$ be a monad on a category $\mathcal{C}$ and A an object of C. Assume in addition that a coproduct $(X \coprod A, ii_1 : X \to X \coprod A, ii_2 : A \to X \coprod A)$ is specified from each X. To construct a Kleisli triple structure $(\eta'_A, bind'_A)$ on the function R'_A from $Ob(C)$ to itself of the form $X \mapsto X \coprod A$.*

Construction 3.2 [**2015.11.22.constr1**] Define $\eta'_A(X) : X \to R(X \coprod A)$ as the composition $\eta(X) \circ R(ii_1)$. For $f : X \to R(Y \coprod A)$ define $bind'_A : R(X \coprod A) \to R(Y \coprod A)$ as $bind(coprod(f, ii_2 \circ \eta(Y \coprod A)))$.

4 Expressions over a binding signature

Given a binding signature $\Sigma = (Op, Ar)$ and a set Var one defines the set of linear expressions over Σ with the names of variables from Var as the set of strings of symbols exp generated by the grammar in the BN-normal form of the form

$$[\mathbf{2015.11.15.eq1}]exp ::= Var|(|_{op \in Op} op(Var. \ldots .Var.exp, \ldots, Var. \ldots .Var.exp)) \qquad (1)$$

where the number of occurrences of Var at the i-th argument of op is the number at the i-th position of $Ar(op)$. For example, for the signature $\Lambda_{sig} = (\{\lambda, ap\}, \{\lambda \mapsto (1), ap \mapsto (0,0))$ the grammar will consists of the line

$$exp ::= Var|\lambda(Var.exp)|ap(exp, exp)$$

and for the signature $\Pi_{sig} = (\{\Pi\}, \{\Pi \mapsto (0,1)\})$ of the line

$$exp ::= Var|\Pi(exp, Var.exp)$$

The set exp of expressions (strings) generated by the grammar (1) is the smallest set of strings that contains elements of Var as well as strings obtained by substituting elements of Var instead of Var and the elements of exp instead of exp in the strings $op(Var. \ldots .Var.exp, \ldots, Var. \ldots .Var.exp)$ for each $op \in Op$ and the numbers of Var's in front of the i-th exp equals to the i-th number in

$Ar(op)$. Let us denote the set of strings corresponding to the signature Σ and the set of names of variables Var by $LExp(\Sigma, Var)$. Then, for example, $LExp(\Lambda, \mathbf{N})$ contains strings such as 1, $\lambda(1.\lambda(1.1))$ and $ap(1,2)$ and does not contain the string $\lambda(1,2)$.

Consider another set defined by a pair (Var, Σ). Elements of this set are planar rooted labelled trees

Problem 4.1 *[***2015.11.15.prob1***] To construct a bijection from the set $LExp(\Sigma, Var)$ to the set*

References

[1] Peter Aczel. A general Church-Rosser theorem. *http://www.ens-lyon.fr/LIP/REWRITING/MISC/AGeneralChurch-RosserTheorem.pdf*, 1978.

[2] Marcelo Fiore, Gordon Plotkin, and Daniele Turi. Abstract syntax and variable binding (extended abstract). In *14th Symposium on Logic in Computer Science (Trento, 1999)*, pages 193–202. IEEE Computer Soc., Los Alamitos, CA, 1999.

[3] André Hirschowitz and Marco Maggesi. Modules over monads and linearity. In *Logic, language, information and computation*, volume 4576 of *Lecture Notes in Comput. Sci.*, pages 218–237. Springer, Berlin, 2007.

[4] Eugenio Moggi. Notions of computation and monads. *Inform. and Comput.*, 93(1):55–92, 1991. Selections from the 1989 IEEE Symposium on Logic in Computer Science.