Category Theory Monads Monads and Adjunctions Universal Algebra Comonads

## Monads and universal algebra

Anthony Voutas

Australian National University

June, 2012



Category Theory

- Category Theory
- 2 Monads

- Category Theory
- 2 Monads
- Monads and Adjunctions

- Category Theory
- 2 Monads
- Monads and Adjunctions
- 4 Universal Algebra

- Category Theory
- 2 Monads
- Monads and Adjunctions
- 4 Universal Algebra
- Comonads

Algebraic theory of functions

- Algebraic theory of functions
- Alternative language for all mathematics

- Algebraic theory of functions
- Alternative language for all mathematics
- Relatively abstract

- Algebraic theory of functions
- Alternative language for all mathematics
- Relatively abstract
- Highlights relationships missed by other fields

Functional programming

- Functional programming
- Different view of adjunctions

- Functional programming
- Different view of adjunctions
- Dual notion: comonad

- Functional programming
- Different view of adjunctions
- Dual notion: comonad
- Universal algebra

## An example with lists

The list monad  $\mathbb{T} = \langle T, \mu, \eta \rangle$ :

For set X, T(X) set of finite lists of X elements, for functions f and lists  $[x_1, x_2, ..., x_n] \in TX$ :

$$T(f)([x_1, x_2, ..., x_n]) = [f(x_1), f(x_2), ..., f(x_n)]$$

 $\mu_X: T^2(X) \to T(X)$  is a flattening in which:

$$\begin{split} & \left[ \left[ x_{1,1}, x_{1,2}, ..., x_{1,n} \right], \left[ x_{2,1}, x_{2,2}, ..., x_{2,n} \right], ..., \left[ x_{m,1}, x_{m,2}, ..., x_{m,n} \right] \right] \\ & \mapsto \left[ x_{1,1}, x_{1,2}, ..., x_{1,n}, x_{2,1}, x_{2,2}, ..., x_{2,n}, ..., x_{m,1}, x_{m,2}, ..., x_{m,n} \right] \end{split}$$

and  $\eta_X: X \to T(X)$  is an injection  $x_1 \mapsto [x_1]$ 

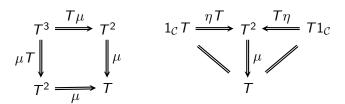


#### Monads

A Monad is a triple  $\mathbb{T}=\langle T,\mu,\eta\rangle$  such that  $T:\mathcal{C}\to\mathcal{C}$  is an endofunctor,  $\mu$  and  $\eta$  are natural transformations

$$\mu: T^2 \stackrel{\cdot}{\rightarrow} T, \qquad \eta: 1_{\mathcal{C}} \stackrel{\cdot}{\rightarrow} T$$

and the following are commutative diagrams



# Monad properties (Informally)

T is an algebra type.

 $\mu$  is a recipe for taking terms of terms to terms (flattening). Terms of terms of terms flatten to terms unambiguously.

 $\eta$  is a recipe for making singleton terms (embedding). Embedding and then flattening returns the original.

# Free-forgetful adjunction for monoids

Free functor  $F : \mathbf{Set} \to \mathbf{Mon}$ . F(X) monoid of strings of X elements with concatenation.  $Ff([x_1, ..., x_n]) = [f(x_1), ..., f(x_n)]$ .

Forgetful functor  $U: \mathbf{Mon} \to \mathbf{Set}$ . U(M) underlying set of M, monoid homomorphisms sent to underlying functions.

Natural transformations  $\eta_X(x_1) = [x_1]$ ,  $\epsilon_M([m_1, ..., m_n]) = m_1 \circ_M ... \circ_M m_n$ .

## Monads and Adjunctions

A natural construction for a monad from an adjunction

#### **Theorem**

Given an Adjunction  $(F, G, \eta, \epsilon)$  with  $F \dashv G$ , we can always construct a monad in the category C = dom(F) as follows:

$$T := GF : \mathcal{C} \to \mathcal{C}$$

$$\eta := \eta : 1_{\mathcal{C}} \xrightarrow{\cdot} GF$$

$$\mu := G \epsilon F : GFGF \xrightarrow{\cdot} GF$$

#### Monad for monoids

Monad  $\langle UF, \eta, U\epsilon F \rangle$ . UF(X) the set of lists over X,  $\eta_X(x_1) = [x_1]$  and  $U\epsilon F_X$  acts as follows:

$$\begin{split} & \left[ \left[ x_{1,1}, x_{1,2}, ..., x_{1,n} \right], \left[ x_{2,1}, x_{2,2}, ..., x_{2,n} \right], ..., \left[ x_{m,1}, x_{m,2}, ..., x_{m,n} \right] \right] \\ & \mapsto \left[ x_{1,1}, x_{1,2}, ..., x_{1,n}, x_{2,1}, x_{2,2}, ..., x_{2,n}, ..., x_{m,1}, x_{m,2}, ..., x_{m,n} \right] \end{split}$$

#### Monad for monoids

Monad  $\langle UF, \eta, U\epsilon F \rangle$ . UF(X) the set of lists over X,  $\eta_X(x_1) = [x_1]$  and  $U\epsilon F_X$  acts as follows:

$$\begin{split} & \left[ \left[ x_{1,1}, x_{1,2}, ..., x_{1,n} \right], \left[ x_{2,1}, x_{2,2}, ..., x_{2,n} \right], ..., \left[ x_{m,1}, x_{m,2}, ..., x_{m,n} \right] \right] \\ & \mapsto \left[ x_{1,1}, x_{1,2}, ..., x_{1,n}, x_{2,1}, x_{2,2}, ..., x_{2,n}, ..., x_{m,1}, x_{m,2}, ..., x_{m,n} \right] \end{split}$$

Same as list monad seen earlier.

## Monads to adjunctions?

This raises the question: given a monad, can one split it into two adjoint functors T = UF?

# Monads to adjunctions?

This raises the question: given a monad, can one split it into two adjoint functors T = UF?

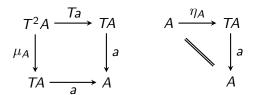
Two answers were independently given in 1965. The Kleisli adjunction, and the Eilenberg-Moore adjunction. We will focus on the Eilenberg-Moore adjunction, because it has relevance to universal algebra.

## Algebras for a Monad

Objects of The Eilenberg-Moore category  $\mathcal{C}^{\mathbb{T}}$  (the  $\mathbb{T}$ -algebras) will be the pairs (A,a) such that  $A\in\mathcal{C}$  and  $a:TA\to A$ 

$$a \circ Ta = a \circ \mu_A$$
 and  $a \circ \eta_A = 1_A$ 

That is, the following diagrams commute:



## Algebras for the list monad

TA is the set of lists over elements of A, the evaluation a is a function which picks a value from A for any given list.

- The evaluation a is associative
- Singleton lists must be evaluated as the value they contain.

These are essentially the requirements of a monoid.

## Eilenberg-Moore adjunction

#### Theorem

Given a monad  $\mathbb{T} = \langle T, \mu, \eta \rangle$  on  $\mathcal{C}$  we can construct an adjunction between  $\mathcal{C}$  and  $\mathcal{C}^{\mathbb{T}}$ .  $U : \mathcal{C}^{\mathbb{T}} \to \mathcal{C}$  (forgetful)

$$U(A, a) = A$$

$$U(f : (A, a) \rightarrow (B, b)) = f : A \rightarrow B$$

and  $F: \mathcal{C} \to \mathcal{C}^{\mathbb{T}}$  (free)

$$F(A) = (TA, \mu_A)$$
  
 $F(f : A \rightarrow B) = T(f) : (TA, \mu_A) \rightarrow (TB, \mu_B)$ 

Note: T = UF.



# EM adjunction for the list monad

 $U: \mathbf{Mon} \to \mathbf{Set}$  (forgetful):

$$U(M) = |M|$$

$$U(h: M \to N) = h: |M| \to |N|$$

and  $F : \mathbf{Set} \to \mathbf{Mon}$  (free)

$$F(A) = (TA, \mu_A)$$

$$F(f)([x_1, ..., x_n]) = T(f)([x_1, ..., x_n])$$

$$= [f(x_1), ..., f(x_n)]$$

The free-forgetful adjunction seen earlier.



# Comparison Functor

#### Theorem

Given an adjunction  $\mathcal{X} \leftrightarrow \mathcal{A}$ , there is a functor  $K : \mathcal{A} \rightarrow \mathcal{X}^{\mathbb{T}}$ .

This functor can always be defined. When is K an isomorphism?

## Universal Algebra

Universal algebra defines and explores a general concept of algebra. It has produced the notion of an algebraic variety, which is a category of algebras of the same type (groups, rings, monoids, etc).

It is a general result that there are always a free-forgetful adjunction between a variety and **Set**.

## Monadicity

An adjunction is called monadic if the comparison functor is an isomorphism.

Beck's Theorem can be used to show that the comparison functor for the free-forgetful adjunction of a variety is an isomorphism.

So a variety is isomorphic to an EM category of algebras.

# Duality and comonads

All of these results dualise to the case of comonads. This gives us a coherent notion of coalgebra, which is useful many applications including (but not limited to)

- modelling state transition systems;
- reasoning generically about modal logic; and
- representing unbounded data types (eg streams).

## Questions

Questions?