Lambda-Calculus & Categories 12+1

Monday 11 January 2021

Monads

II

(A) the category Alg_T of T-algebras and T-homomorphisms

start with the example of

$$T : Set \longrightarrow Set$$
$$A \longmapsto A^*$$

"free monoid" monad.

how could we recover the usual notion of monoid?

the "usual" definition:

a monoid is a set A equipped with a binary operation

$$m : A \times A \longrightarrow A$$
$$e : 1 \longrightarrow A$$

$$1 = \{*\} = A^0$$

$$m : A^2 \longrightarrow A$$

$$e : A^0 \longrightarrow A$$

the constant e can be seen as a "nullary" operation.

which is associative:

$$\forall_{xyz} \quad m(m(x,y),z) = m(x, m(y,z))$$

and with e as neutral element

$$\forall_x \quad m(x,e) = x = m(e,x).$$

We are going to give an alternative definition.

<u>A first observation</u>. A monoid for all $n \in \mathbb{N}$, there is a function

$$A^n \xrightarrow{\text{mult}_n} A$$

$$(a_1, \dots, a_n) \longmapsto a_1 \cdot \text{---} \cdot a_n$$

where $a_1 \cdot \ldots \cdot a_n = m(a_1, m(a_2, \ldots))$

is obtained by multiplying the elements

$a_1, \ldots, a_n$ together.

In particular

$$A^0 \xrightarrow{\text{mult}_0} A$$

is just given by the neutral element.

So, we obtain in this way

a function:

$$\text{mult} : \coprod_{n \in \mathbb{N}} A^n \longrightarrow A$$

$$(n, a_1, \ldots, a_n) \longmapsto \text{mult}_n(a_1, \ldots, a_n)$$

$n \in \mathbb{N}$

In other words, every monoid A comes equipped with a function

$$TA \xrightarrow{\text{alg}} A$$

(alg is another name for mult)

where $TA = A^* = \coprod_{n \in \mathbb{N}} A^n$.

If we think of $\text{alg} : TA \longrightarrow A$

$$\text{alg}([a_1, \dots, a_n]) = \underbrace{a_1 \cdot \_\_\_ \cdot a_n}_{\substack{\text{multiplication} \\ \text{in the monoid.}}}$$

<u>The question is</u>: what should we require of <u>alg</u> to be sure that it defines a monoid?

$$\mathrm{alg} : A^* \longrightarrow A$$

<u>first observation</u>: for every monoid A

$$\mathrm{alg}(\underbrace{[a]}_{\substack{\text{word} \\ \text{with} \\ \text{one letter}}}) = a \qquad \text{for all letters } a \in A$$

in other words, the diagram commutes:

$$\begin{tikzcd}
& TA \arrow[dr, "\mathrm{alg}"] \arrow[d, phantom, "\circledast"] & \\
A \arrow[ur, "\eta_A"] \arrow[rr, "\mathrm{id}_A"'] & {} & A
\end{tikzcd}$$

$$\begin{tikzcd}
& {[a]} \arrow[dr, mapsto, bend left] & \\
a \arrow[ur, mapsto] \arrow[rr, mapsto, bend right] & & \mathrm{alg}([a]) = a
\end{tikzcd}$$

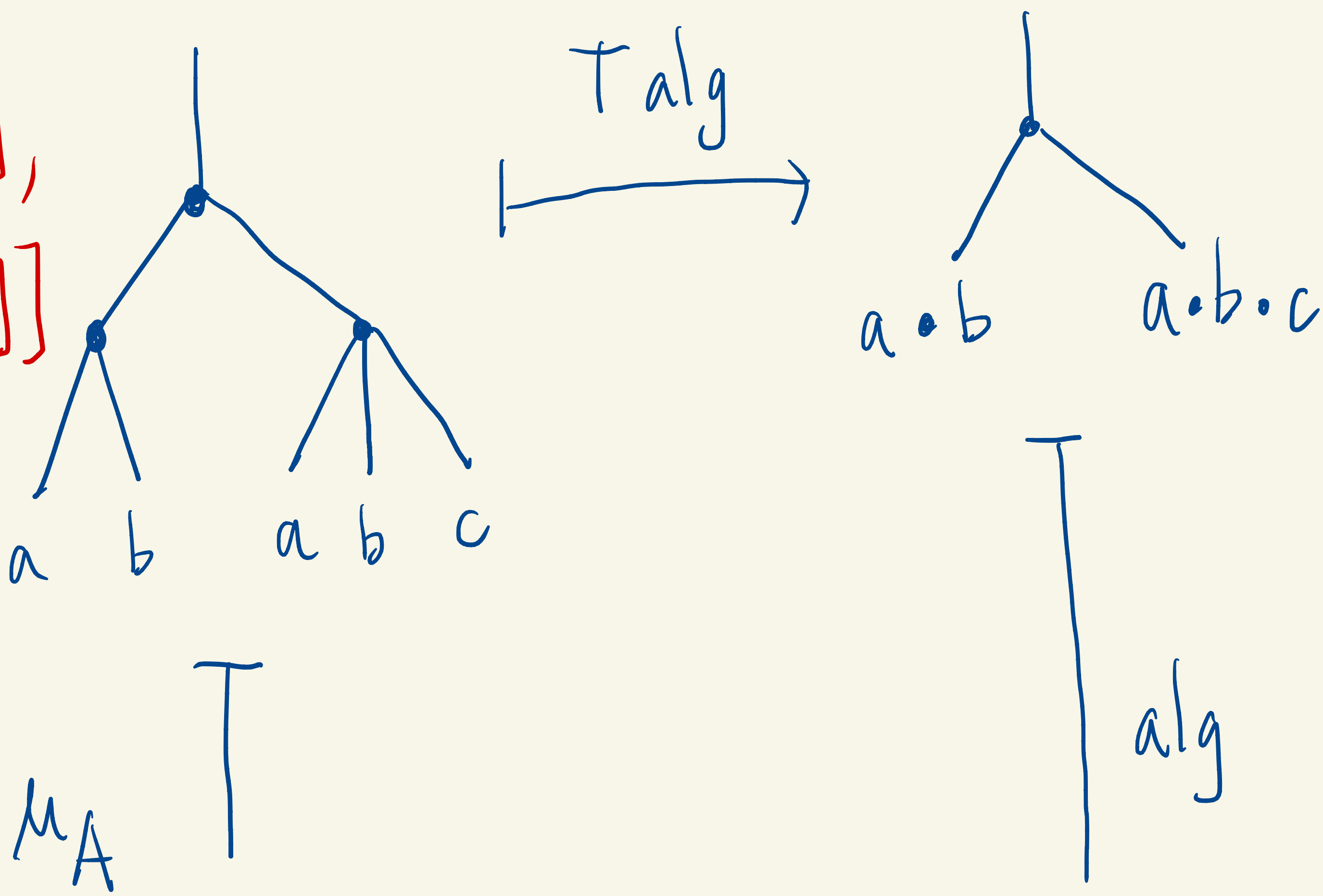
second observation for a monoid A

the diagram should commute:

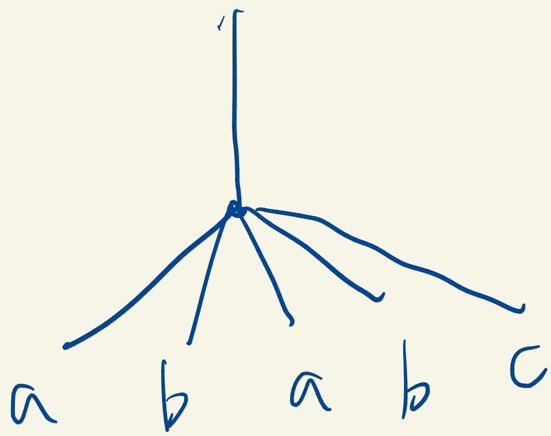
$$\begin{tikzcd} TTA \arrow[r, "T\mathrm{alg}"] \arrow[d, "\mu_A"'] \arrow[dr, phantom, "**"] & TA \arrow[d, "\mathrm{alg}"] \\ TA \arrow[r, "\mathrm{alg}"] & A \end{tikzcd}$$

given an element of A^{**}: $[a \cdot b, a \cdot b \cdot c]$

$[[a, b], [a, b, c]]$



Flatten $\downarrow$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\downarrow$



$\xmapsto{\text{alg}} \quad a \cdot b \cdot a \cdot b \cdot c$

[a, b, a, b, c]

<u>Thm</u>. a monoid (in the usual sense)

is the same thing

as a set A

equipped with a function

$$\text{alg} : TA \longrightarrow A$$

making the two diagrams

(*) and (**)

commute.

<u>Left as exercise</u>

<u>Hint</u>:

$$TA \xrightarrow{alg} A$$

may be seen as a function

$$\coprod_{n \in \mathbb{N}} A^n \xrightarrow{alg} A$$

and thus as a family of functions

$$A^n \xrightarrow{\text{mult}_n} A$$

$$[a_1, \dots, a_n] \longmapsto \text{mult}_n [a_1, \dots, a_n]$$

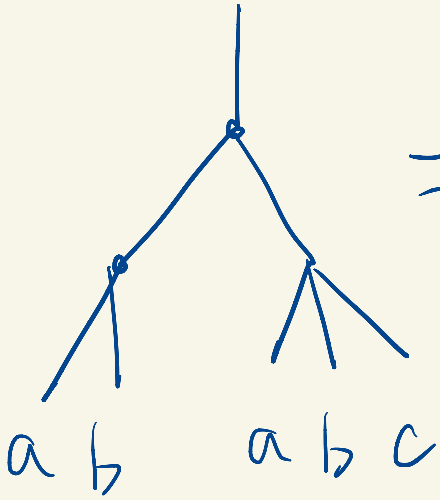
which could be noted

$$\underbrace{(a_1 *_n \text{———} *_n a_n)}_{\text{a } n\text{-ary multiplication}}$$

the diagram ($*$) means that

$$*_1([a]) = a$$

the diagram ($**$) means that



$$= [[a,b],[a,b,c]] \in A^{**}$$

$$(a *_2 b) *_2 (a *_3 b *_3 c)$$

$$= (a *_5 b *_5 a *_5 b *_5 c)$$

for example.

Purpose of the exercise is to show that $*_n$ can be derived from $*_2, *_0$

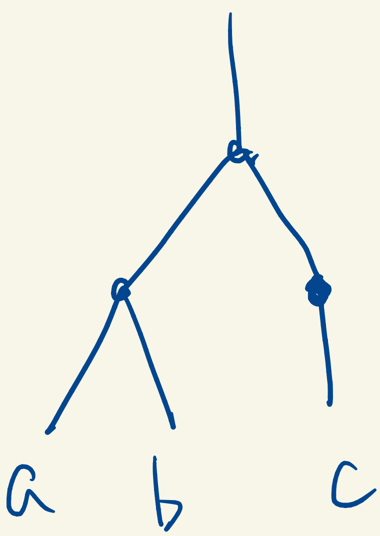
and that moreover

$*_2$ is associative

$*_0$ is a neutral element for $*_3$.

$(*)$

<u>associativity</u> follows from $(**)$

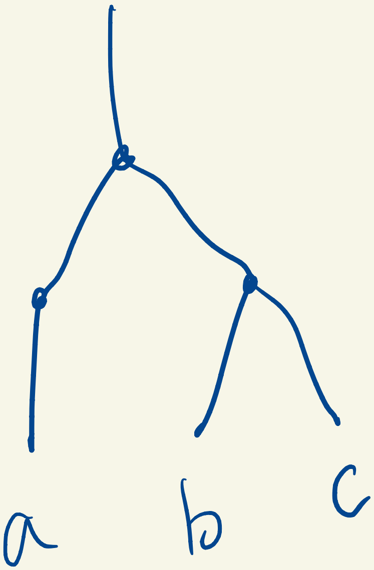


$$= [[a,b],[c]] \in A^{**}$$

$$(a *_2 b) *_2 (*_1 c) = (a *_3 b *_3 c)$$

$$\|$$

$$(a *_2 b) *_2 c$$



$$= [[a],[b,c]] \in A^{**}$$

$$(*_1 a) *_2 (b *_2 c) = (a *_3 b *_3 c)$$

$$\parallel$$

$$a *_2 (b *_2 c)$$

hence associativity of the monoid comes from the equation:

$$(a *_2 b) *_2 c = (a *_3 b *_3 c) = a *_2 (b *_2 c)$$

similarly for neutrality.

this leads us to the definition

<u>def</u>. given a monad T on a category $\mathcal{C}$

a T-algebra is a pair

$$(A, alg)$$

consisting of an object A of $\mathcal{C}$ and a morphism

$$alg : TA \longrightarrow A$$

making the two diagrams below commute:

$$\begin{tikzcd} & TA \arrow[dr, "alg"] & \\ A \arrow[ur, "\eta_A"] \arrow[rr, "id_A"'] & & A \end{tikzcd} \qquad (*)$$

$$\begin{tikzcd} TTA \arrow[r, "Talg"] \arrow[d, "\mu_A"'] & TA \arrow[d, "alg"] \\ TA \arrow[r, "alg"'] & A \end{tikzcd} \qquad (**)$$

A T-homormorphism between two T-algebras

$$(A, alg_A) \xrightarrow{f} (B, alg_B)$$

is a morphism

$$f : A \longrightarrow B$$

of the category $\mathcal{C}$ making the diagram below commute

$$\begin{tikzcd}
TA \arrow[r, "Tf"] \arrow[d, "alg_A"'] & TB \arrow[d, "alg_B"] \\
A \arrow[r, "f"] & B
\end{tikzcd}$$

We obtain in this way a category

$$\mathrm{Alg}_T$$

with T-algebras as objects

T-homomorphisms as maps.

<u>Thm</u>. there is an adjunction

$$\begin{tikzcd} \mathcal{C} \arrow[rr, bend left, "\mathrm{Free}"] \arrow[rr, phantom, "\perp"] & & \mathrm{Alg}_T \arrow[ll, bend left, "\mathrm{Forget}"] \end{tikzcd}$$

$$\mathrm{Free} : A \longmapsto (TA, \mu_A)$$

$$\mathrm{Forget} : (A, alg) \longmapsto A$$

<u>Remark</u>: (TA, μ_A) is indeed a T-algebra, with algebra structure given by the map

$$\mu_A : TTA \longrightarrow TA$$

for instance, in the case of the free monoid monad:

$$\mu_A : A^{**} \longrightarrow A^*$$
$$[[a,b],[a,b,c]] \longmapsto [a,b,a,b,c]$$

this is indeed the multiplication (by concatenation) of the free monoid.

Moreover, the monad

$$\text{Forget} \circ \text{Free} : \mathcal{C} \longrightarrow \mathcal{C}$$

associated to the adjunction coincides with the original monad

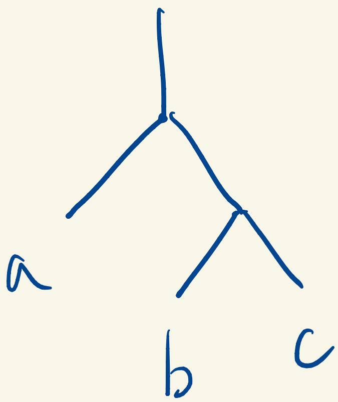
$$(T, \mu, \eta)$$

<u>Example</u>:

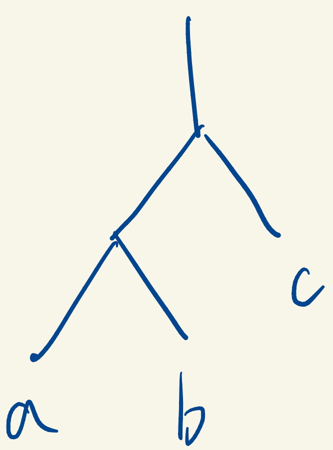
the category of Mag-algebras for the monad $\text{Mag} : \text{Set} \to \text{Set}$ coincides with the category of magmas and homomorphisms.

$\boxed{\text{binary tree}}$

$\longmapsto$ an element of the magma



$$\longmapsto \quad a*(b*c)$$



$$\longmapsto \quad (a*b)*c$$

this captures the essence of a "magma"

(B) another way to associate to any monad

$$(T, \mu, \eta) : \mathscr{C} \longrightarrow \mathscr{C}$$

an adjunction:

construct the Kleisli category

$$\mathscr{K}\ell_T$$

- whose objects are the objects of the category $\mathscr{C}$
- whose morphisms from A to B

are the morphisms

from A to TB

in the original category $\mathcal{C}$

<u>Notation</u>:

$A \dashrightarrow B$ in Kl_T

$A \longrightarrow B$ in $\mathcal{C}$

<u>Question</u>: how do we compose two morphisms in Kl_T?

$$\begin{tikzcd} A \arrow[r, dashed, "f"] \arrow[rr, dashed, bend right, "g \circ_K f"'] & B \arrow[r, dashed, "g"] & C \end{tikzcd} \quad \text{in } Kl_T$$

$$
\begin{tikzcd}
 & TB & & TC \\
A \arrow[ur, "f"] \arrow[urrr, "g \circ_K f"] & & B \arrow[ur, "g"'] &
\end{tikzcd}
\qquad \text{in } \mathcal{C}
$$

$$
\begin{tikzcd}
 & & & TTC \arrow[d, "\mu_C"] \\
 & TB \arrow[urr, "Tg"] & & TC \\
A \arrow[ur, "f"] & & B \arrow[ur, "g"] &
\end{tikzcd}
$$

we thus define

$$g \circ_K f = \mu_C \circ Tg \circ f$$

__Question__: how do we define the identity

$$A \dashrightarrow A \quad \text{in } \mathcal{K}\ell_T$$

well, just as:

$$A \xrightarrow{\eta_A} TA \quad \text{in } \mathcal{C}$$

__Left as exercise__:

Show that $\mathcal{K}\ell_T$ defines a category.

this means:

1. check that $\circ_K$ is associative

2. check that η_A defines an identity map in Kl_T.

<u>Thm</u>: for every monad (T, μ, η) on $\mathcal{C}$

we have an adjunction

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "L"] \arrow[r, phantom, "\perp"] & Kl_T \arrow[l, bend left, "R"] \end{tikzcd}$$

L transports an object A to itself in Kl_T

R transports an object A in Kl_T to TA.

Moreover, the monad (T, μ, η) coincides with the monad

$(R \circ L, \eta, R \circ \varepsilon \circ L)$

associated to the adjunction.

<u>Remark</u>: each adjunction

$$\begin{tikzcd} \mathcal{A} \arrow[r, bend left, "L"] \arrow[r, phantom, "\perp"] & \mathcal{B} \arrow[l, bend left, "R"] \end{tikzcd}$$

such that $T = R \circ L$

$\mu = R \circ \varepsilon \circ L \quad \eta = \eta$

can be seen as a way

to factor T as $R \circ L$

From that point of view,
the two factorisations
described in this lesson

- using the category Alg_T
- using the Kleisli category

can be seen as "extremal"
solutions where

$$\left.\begin{array}{l} L = T \text{ in } Alg_T \\ R = T \text{ in } Kl_T \end{array}\right\} \text{ on the objects}$$

on the objects

(1)
$$\mathcal{C} \underset{\substack{\text{identity} \\ \text{(forgetful)}}}{\overset{T}{\rightleftarrows}} \mathrm{Alg}_T$$

(2)
$$\mathcal{C} \underset{T}{\overset{\text{identity}}{\rightleftarrows}} \mathrm{Kl}_T$$

in fact this can be elaborated by constructing a category of adjunctions factoring T

and showing that
the adjunction 1
is terminal

the adjunction 2
is initial.

<u>Solution</u> in MacLane's
book categories for
the working mathematicians.

Example.

the monad T on Set

$$A \longmapsto A_* = A \uplus \{*\}$$

the Kleisli category of the monad has

- sets as objects
- functions $A \longrightarrow B_*$ as morphisms

hence partial function

from A to B.

For that reason, Kl_T coincides with the category of sets and partial functions.

In the case of the free monoid monad, Kl_T has sets as objects, and functions $A \xrightarrow{f} B^*$ as morphisms.

Such a function associates
to every letter $a \in A$
a finite word of letters in B

Such a function f
can be seen as
a homomorphism

$$A^* \longrightarrow B^*$$

$$[a_1 \dots a_n] \longmapsto \underbrace{fa_1 \cdot_B \dots \cdot_B fa_n}_{\text{concatenated in } B}$$

For that reason Kl_T can be seen as the category of free (and presented) monoids

More generally, for any monad (T, μ, η) Kl_T can be understood as the category of free (and presented) T-algebras.