The Microservices Operad
–OR–
the free monad monad is a module over the cofree comonad comonad

David I. Spivak

2023/02/21 at Topos Berkeley Seminar

I. Introduction

  A. Polynomials as problems and solutions — 5 mins
    1. Polynomials
    2. Maps
    3. Maps to composites of $\otimes$ and $\triangleleft$, eg. $q \to (p_1 \otimes p_2) \triangleleft p_3$
  B. Math: $\mathfrak{c}_p \otimes \mathfrak{m}_q \to \mathfrak{m}_{p \otimes q}$. — 2 mins
  C. $\mathbb{O}$**rg** and updating behavior — 2 mins
  D. Plan for the talk — 1 min

time check: 14:10

II. Background on **Poly**, $\otimes, \vee, \triangleleft, [-,-]$

  A. $p \otimes q$ — 2 mins
  B. $p \vee q$ — 2 mins
  C. $p \triangleleft q$ and trees — 3 mins
  D. $[p, q]$ — 2 mins

time check: 14:19

III. Free monads and cofree comonads

A. The free monad $\mathfrak{m}_p$ on p (finitary) 6 mins

1. Define $p_0 := y, \quad p_{i+1} := y + p \triangleleft p_i$
2. Define $p_i \to p_{i+1}$ inductively
3. Then $\mathfrak{m}_p := \operatorname{colim}(\cdots \leftarrow p_1 \leftarrow p_0)$
4. Idea as syntax trees
5. $\mathfrak{m}_p \otimes \mathfrak{m}_q \to \mathfrak{m}_p \vee \mathfrak{m}_q \to \mathfrak{m}_{p \vee q}$
6. $p \to \mathfrak{m}_p$ and $\mathfrak{m}_{\mathfrak{m}_p} \to \mathfrak{m}_p$

B. The cofree comonad $\mathfrak{c}_p$ on p 8 mins

1. Define $p^0 := y, \quad p^{i+1} := y(p \triangleleft p^i)$
2. Define $p^{i+1} \to p^i$ inductively
3. Then $\mathfrak{c}_p := \lim(\cdots \to p^1 \to p^0)$
4. Idea as behavior trees
5. Comonads are categories
   a. $\mathfrak{c}_p$ as category
   b. p-**Coalg** $\cong \mathfrak{c}_p$-**Set**
   c. $\mathbf{Cat}^\sharp \to \mathbf{Cat}$ by $c \mapsto c$-**Set** is lax monoidal

$$c\text{-}\mathbf{Set} \times c'\text{-}\mathbf{Set} \to (c \otimes c')\text{-}\mathbf{Set}$$

6. $\mathfrak{c}_p \to p$ and $\mathfrak{c}_p \to \mathfrak{c}_{\mathfrak{c}_p}$

C. Free monad as module over cofree comonad 8 mins

1. There is a natural map $\lessdot\colon \mathfrak{c}_p \otimes \mathfrak{m}_q \to \mathfrak{m}_{p \otimes q}$
2. In fact, this extends to $\lessdot\colon \mathfrak{c}_p \otimes \mathfrak{m}_{q \vee q'} \to \mathfrak{m}_{(p \otimes q) \vee q'}$
3. It satisfies the action laws (and similar for extension)

$$\begin{array}{ccc} \mathfrak{m}_q & = & \mathfrak{m}_q \\ \downarrow & & \| \\ \mathfrak{c}_y \otimes \mathfrak{m}_q & \xrightarrow[\lessdot]{} & \mathfrak{m}_{y \otimes q} \end{array} \qquad \begin{array}{ccc} \mathfrak{c}_{p_1} \otimes \mathfrak{c}_{p_2} \otimes \mathfrak{m}_q & \xrightarrow{\lessdot} & \mathfrak{c}_{p_1} \otimes \mathfrak{m}_{p_2 \otimes q} \\ \downarrow & & \downarrow{\scriptstyle \lessdot} \\ \mathfrak{c}_{p_1 \otimes p_2} \otimes \mathfrak{m}_q & \xrightarrow[\lessdot]{} & \mathfrak{m}_{p_1 \otimes p_2 \otimes p_3} \end{array}$$

4. And it satisfies additional coherence with respect to $\mathfrak{c}_p \to p, \mathfrak{c}_p \to \mathfrak{c}_{\mathfrak{c}_p}, q \to \mathfrak{m}_q, \mathfrak{m}_{\mathfrak{m}_q} \to \mathfrak{m}_q$:

$$\begin{array}{ccc} \mathfrak{c}_p \otimes q & \longrightarrow & \mathfrak{c}_p \otimes \mathfrak{m}_q \\ \downarrow & & \downarrow \\ p \otimes q & \longrightarrow & \mathfrak{m}_{p \otimes q} \end{array}$$

$$\begin{array}{ccccccc} \mathfrak{c}_p \otimes \mathfrak{m}_{\mathfrak{m}_q} & \longrightarrow & \mathfrak{c}_{\mathfrak{c}_p} \otimes \mathfrak{m}_{\mathfrak{m}_q} & \longrightarrow & \mathfrak{m}_{\mathfrak{c}_p \otimes \mathfrak{m}_q} & \longrightarrow & \mathfrak{m}_{\mathfrak{m}_{p \otimes q}} \\ \downarrow & & & & & & \downarrow \\ \mathfrak{c}_p \otimes \mathfrak{m}_q & & & \longrightarrow & & & \mathfrak{m}_{p \otimes q} \end{array}$$

time check: 14:41

IV. Implementing the microservices operad

A. Variants of $\mathbb{O}\mathbf{rg}$ — 6 mins

1. $\mathfrak{c}$ lifts to a comonad on $\mathbb{O}\mathbf{rg}$
   a. This uses lax monoidality of $\mathfrak{c}$.
2. $\mathfrak{m}$ lifts to a monad on $\mathbb{O}\mathbf{rg}$
   a. This uses module structure $\mathfrak{c} \otimes \mathfrak{m} \to \mathfrak{m}$.
3. Define $\mathbb{O}\mathbf{rg}_{\mathfrak{m}}$, "Microservices operad", to be $\mathfrak{m}$-Kleisli
   a. $\mathrm{Ob}(\mathbb{O}\mathbf{rg}_{\mathfrak{m}}) := \mathrm{Ob}(\mathbf{Poly})$
   b. $\mathbb{O}\mathbf{rg}_{\mathfrak{m}}(p_1, \ldots, p_k; q) := [q, \mathfrak{m}_{p_1 \vee \cdots \vee p_k}]$-**Coalg**
   c. Explain composition
4. Define $\mathbb{O}\mathbf{rg}^{\mathfrak{c}}$, "machines operad", to be $\mathfrak{c}$-coKleisli
   a. $\mathrm{Ob}(\mathbb{O}\mathbf{rg}^{\mathfrak{c}}) := \mathrm{Ob}(\mathbf{Poly})$
   b. $\mathbb{O}\mathbf{rg}^{\mathfrak{c}}(p_1, \ldots, p_k; q) := [\mathfrak{c}_{p_1} \otimes \cdots \otimes \mathfrak{c}_{p_k}, q]$-**Coalg**
   c. These are exactly "linear effects handlers" as with Owen.
   d. There's a locally fully faithful 2-functor $\mathbb{O}\mathbf{rg}^{\mathfrak{c}} \to \mathbb{C}\mathbf{at}^{\sharp}$
      (1) On objects it sends $p \mapsto \mathfrak{c}_p$.

(2) On hom-categories it sends $Sy^S \to [\mathfrak{c}_p, q]$ to a bicomodule of the form $\mathfrak{c}_p \xleftarrow{Sy \triangleleft \mathfrak{c}_q} \mathfrak{c}_q$

B. 2-functors $(\mathbb{O}\mathbf{rg}_{\mathfrak{m}})^{\text{op}} \to \mathbb{O}\mathbf{rg}^{\mathfrak{c}}$. 3 mins

1. For any monad t we have one, given by $p \mapsto [p, t]$.
   a. Examples: $t = y$, $t = y + 1$, $t = \text{dist}$
   b. Note that $[Ay^B, y] \cong [Ay, By]$ and $[Ay^B, y + 1] \cong [Ay, By + 1]$
   c. This relies heavily on the module structure $\mathfrak{c} \otimes \mathfrak{m} \to \mathfrak{m}$.
2. This says that the machine operad can implement microservices operad
3. The machine $[p, y]$ associated to p outputs solution-sets $p \to y$ and inputs problems $I \in p(1)$.

time check: 14:50