Operad and regular languages

Introduction

→ (topological) operad : introduced in the 70's to describe loop spaces [May, Boardman–Vogt]

↳ encodes type of monoids algebras, it corresponds to the universal operations which acts on an algebra of a given type.

→ Regular languages : simplest class of languages ; computed by some finite deterministic automaton.

→ question : how to describe regular languages by means of an operad and why?

Plan: 1_ Operad in topology

2_ Representation of an operad

3_ Regular languages

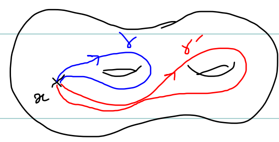
1_ Operad in topology

The notion of operad comes from the mix between algebraic structure and topology.

a_ Loop spaces

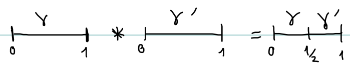
M topological space, $x \in M$

$\Omega M := \{ [0,1] \xrightarrow{\gamma} M$ continuous map s.t. $\gamma(0) = \gamma(1) = x \}$.



Product on ΩM

$$\gamma * \gamma' : [0,1] \longrightarrow M, \quad t \longmapsto \begin{cases} \gamma(2t) & \text{if } 0 \leq t \leq \frac{1}{2}, \\ \gamma'(2t-1) & \text{if } \frac{1}{2} \leq t \leq 1. \end{cases}$$

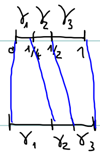


(speed is multiplied by 2).

Associativity ? No!

$(\overset{\gamma_1}{\longmapsto} * \overset{\gamma_2}{\longmapsto}) * \overset{\gamma_3}{\longmapsto} =$

$\overset{\gamma_1}{\longmapsto} * (\overset{\gamma_2}{\longmapsto} * \overset{\gamma_3}{\longmapsto}) =$



$\exists H : [0,1]\times[0,1] \longrightarrow M$, $(s,t) \longmapsto H(s,t)$ s.t. $\begin{cases} H(0,-) = (\gamma_1 * \gamma_2) * \gamma_3 \\ H(1,-) = \gamma_1 * (\gamma_2 * \gamma_3) \end{cases}$ and $H(-,0) = H(-,1) = x$.

$\hookrightarrow$ Get $\mu_2 = * : \Omega M^{\times 2} \longrightarrow \Omega M$ and $\underset{\underset{H}{\Updownarrow}}{\mu_3} : [0,1] \times \Omega M^{\times 3} \longrightarrow \Omega M$.

Is that all? What is the algebraic structure on ΩM?

Recognition principle [May, Boardman–Vogt]

Let Y be a grouplike ($\pi_0 Y$ group) top. space. Then

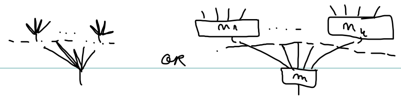
$\exists M$ s.t. $Y \underset{\text{homotopic}}{\sim} \Omega M$ iff Y is an algebra over the operad A_∞.

b_ Operad and algebras

A collection $M = \{M_n\}_{n \geq 0}$ is a collection of topological spaces M_n.

We define a monoidal product $\circ$ on collections ($\circ$ satisfies pentagon diagram + unit)

$(M \circ N)_n := \coprod_{k \geq 0} M_k \times \left(\coprod_{i_1 + \dots + i_k = n} N_{i_1} \times \dots \times N_{i_k} \right)$ [Picture

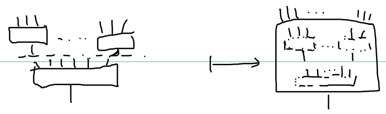


]

(Rk: We can add an action of the symm. group Σ_n on M_n.)

Def: An operad is a monoid in the monoidal category (collections, $\circ$): (P, γ, η), $P = \{P_n\}_{n \geq 0}$

$\gamma : P \circ P \longrightarrow P$, $\eta : I := (\phi, pt, \phi, \dots) \longrightarrow P$ s.t. γ is associative and η is a unit for γ.

[]

Ex: A_∞ is an operad $((A_\infty)_0 = \phi$, $(A_\infty)_1 = pt$, $(A_\infty)_2 = \underset{0 \quad 1}{\longmapsto}$, $(A_\infty)_3 = ?$). Question: What is the monoid struct?

Def: A topological space M is an algebra over the operad $P = \{P_n\}_{n \geq 0}$ if there are maps

$\mu_n : P_n \times M^{\times n} \longrightarrow M$ $\forall n$ satisfying associativity and unit relations. $\left[\begin{array}{ccc} P \circ P(M) & \xrightarrow{\gamma_P} & P(M) \\ {\scriptstyle \{\mu_n\}_n} \downarrow & \circlearrowleft & \downarrow {\scriptstyle \{\mu_n\}_n} \\ P(M) & \xrightarrow{\{\mu_n\}_n} & M \end{array} \right]$

c_ Examples and applications in topology

× A_∞ : $\bullet \times \Omega M^{\times 2} \longrightarrow \Omega M$, $\longmapsto \times \Omega M^{\times 3} \longrightarrow \Omega M$, $\times \Omega M^{\times 4} \longrightarrow \Omega M$, ...

× As : $As_n = \{\cdot\}$ $\forall n \geq 1$; $\gamma : As_k \times As_{i_1} \times \dots \times As_{i_k} \longrightarrow As_{i_1+\dots+i_k}$, $(pt, pt, \dots) \mapsto pt$.

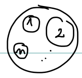
$[(\ast_k ; \ast_{i_1}, \dots, \ast_{i_k}) \longmapsto \ast_{i_1+\dots+i_k}]$

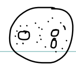
Alg. over As = associative magma (= monoid without unit)

↳ associative alg. in (vector spaces, $\otimes$).

× Little discs/cubes operad : $\mathbb{D}^2 = \{(x,y) \text{ s.t. } x^2+y^2 \leq 1\} \subset \mathbb{R}^2$.

$(\mathcal{D}_2)_n = \{ \underline{c} = (c_1, \dots, c_n) ;\ c_i : \mathbb{D}^2 \hookrightarrow \mathbb{D}^2$ affine embedding $\}$ $c_i(x,y) = (a_1,a_2) + r\cdot(x,y)$ $\forall (x,y) \in \mathbb{D}^2$

$= \{$  $\}$

(Composition : $\gamma($  ,  ,  $)$ =  $\in \mathcal{D}_2(3)$)

Rks : . We have $(\mathcal{D}_2)_n \sim Conf_n(\mathbb{R}^2)$

. We can replace 2 by any number $k \geq 1$.

Results : . recognition principle for iterated loop spaces $\Omega^k M$: $\mathcal{D}_1 = A_\infty \leftrightarrow \Omega M$, $\mathcal{D}_2 \leftrightarrow \Omega^2 M$, $\mathcal{D}_3 \leftrightarrow \Omega^3 M$, ...

. Long knots : [Dwyer – Hess] $\overline{Emb}_c(\mathbb{R}^m, \mathbb{R}^n) \simeq \Omega^{m+1} Map^h_{Op}(\mathcal{D}_m, \mathcal{D}_n)$

↳ if $m=1$, get that long knots forms a double loop space

[Lambrechts – Turchin – Volić] $H_*(Emb(\mathbb{R}, \mathbb{R}^d); \mathbb{Q}) \otimes \underbrace{H_*(\Omega^2 S^{d-1}; \mathbb{Q})}_{\text{known (easy)}} \cong H_*(\overline{Emb}(\mathbb{R}, \mathbb{R}^d)) \overset{[\text{Turchin}]}{\cong} HH_*(Poiss_{d-1})$

↳ computation of the rational homology of long knots

. Grothendieck – Teichmüller group : [Fresse, Horel] $GT_{\mathbb{Q}} \xrightarrow{\simeq} Aut_{Ho(TopOp)}(E_2)^{\wedge}_{\mathbb{Q}}$

2_ Regular languages [Girardo – Luque – Mignot – Nicart, Caron]

[Ehrenfeucht – Zeiger] $\exists (L_n)_n$ finite languages s.t. L_n is recognized by an automaton with n states whereas the smallest regular expression corresponding to L_n is exponential in n.

Therefore, the idea is to increase the number of operations on regular expressions in order to get smaller presentation of the regular expression.

a _ Multitilde

Let Σ be an alphabet and ε be the empty word.

A regular expression E over Σ is inductively defined by the following possibilities

$E = \emptyset$, $E = \varepsilon$, $E = a$, $E = F.G$, $E = F + G$, $E = F^*$ ($a \in \Sigma$, F, G regular expression).

The associated language is denoted by $L(E)$ (and inductively defined by $L(\emptyset) = \emptyset$, $L(\varepsilon) = \{\varepsilon\}$, $L(a) = \{a\}$, $L(F+G) = L(F) \cup L(G)$, $L(F.G) = L(F).L(G)$, $L(F^*) = L(F)^*$).

Example: $E = a + ba^*$, $L(E) = \{a, b, ba, ba^2, \dots, ba^n, \dots\}$.

Tilde: Add the empty word ε: $L(\tilde{E}) = L(E) \cup \{\varepsilon\}$.

Let $[\![1,n]\!]^2_{\leq} := \{(k,k') \; ; \; k, k' \in [\![1,n]\!] \text{ and } k \leq k'\}$ and $S_n := \mathcal{P}([\![1,n]\!]^2_{\leq})$ subsets of $[\![1,n]\!]^2_{\leq}$.

Let $T \subset S_n$, $\Psi_T(E_1, \dots, E_n) = \sim_T(E_1, \dots, E_n) = \underset{\underbrace{S \in F(T)}_{\text{"free subsets"}}}{+} E_1^S \cdot \dots \cdot E_n^S$, where $E_i^S := \begin{cases} \varepsilon & \text{if } i \in \bigcup_{(k,k') \in S} [\![k,k']\!] \\ E_i & \text{otherwise} \end{cases}$

Idea/ex.: $\underset{\sim^{(1,2),(2,3)}}{\overset{\parallel}{\Psi_{(1,2),(2,3)}}} (E_1, E_2, E_3) = \widetilde{\widetilde{E_1 \cdot E_2} \cdot E_3}$ ... $= E_1 \cdot E_2 \cdot E_3 + E_1\,\varepsilon + \varepsilon\,.E_3$.

$\hookrightarrow \Psi_{(1,2),(2,3)}(E_1, \emptyset, E_2) = E_1 \cdot \emptyset \cdot E_2 + E_1 + E_2 = E_1 + E_2$.

Exo: $E_1 . E_2 = ?$

$\hookrightarrow$ Get an operad $\mathcal{T} := \{\mathcal{T}_n\}_{n \geq 1} = (\{ {}_n\Psi_S, S \in S_n\})_{n \geq 1}$ of multilde.

b. Operad and regular languages

Proposition [LMN] The operad $\mathcal{T}$ acts on (finite) languages.

Problem: by the action of $\mathcal{T}$ on $\Sigma \cup \{\emptyset\}$, we cannot obtain a^* (the $*$ operation).

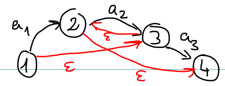
Solutions: . add $|^*$ a new generator to the operad,

- <u>double</u> $\mathcal{D}\mathcal{T} := \mathcal{T} \otimes_{H} \mathcal{T}$, $\mathcal{D}\mathcal{T}_n := \mathcal{T}_n \otimes \mathcal{T}_n$.

<u>Theorem</u> [GLMN] $\mathcal{D}\mathcal{T}/_\sim$ acts faithfully on regular languages on Σ.

Moreover, {reg. lang.} forms a $\mathcal{D}\mathcal{T}/_\sim$-alg. generated by $\Sigma \cup \{\emptyset\}$.

<u>Idea</u> : connection with automaton

$\Psi_{(1,3),(2,4)} \otimes \Psi_{(3,2)}$ applied to (a_1, a_2, a_3) gives  $\left[(a_1(a_2+\varepsilon)+\varepsilon)\, a_2^{*}(a_3+\varepsilon))\right]$

(or $\Psi_{(1,2),(2,3)} \otimes \Psi_{(3,2)}$)

c- <u>Complexity</u>

Two notions of complexity of a regular language :

- $rk_w(L) := \min\{k : \exists\ \Psi_T \in \mathcal{D}\mathcal{T}_k,\ \alpha_1, \dots, \alpha_k \in \Sigma \cup \{\emptyset\} \text{ s.t. } L = \Psi_T(\alpha_1, -, \alpha_k)\}$
- $rk_h(L) := \min\{h : \exists k \geq 1,\ \Psi_T \in \mathcal{D}\mathcal{T}_k,\ \alpha_1, -, \alpha_k \in \Sigma \cup \{\emptyset\} \text{ s.t. } \Psi_T(\alpha_1, -, \alpha_k) = L \text{ and } \#T = k\}$.

Because of the connection with automaton, the complexity might be related to the size of a minimal automaton in terms of states (rk_w) or transitions (rk_h).

↳ <u>Not done yet</u>.