

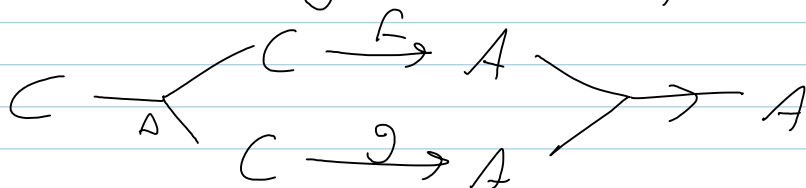
Bar and cobar constructions:

(A, μ) Associative algebra.

(C, Δ) co-associative co-algebra.

consider $\text{Hom}(C, A) \ni f, g$

make $f * g \in \text{Hom}(C, A)$ by



claim The convolution product is associative.

Two examples in video.

(A, μ, d_A) a dga (diff. graded alg.)

$$|d_A| = -1$$

(C, Δ, d_C) a dg-coalg $|d_C| = -1$.

Get $(\text{Hom}(C, A), *, \partial)$, where

$$\partial f = d_A \circ f - (-1)^{|f|} f \circ d_C$$

MC equation (Maurer-Cartan)

in degree -1 : $\partial \alpha + \alpha \neq \alpha$

The space of solns is $\text{Tw}(C, A)$,
 "twisting morphisms $C \rightarrow A$ "
 it's a bifunctor.

We want to represent Tw :

$$\text{Hom}(\underset{\substack{\uparrow \\ \text{in dga's}}}{?}, A) = \text{Tw}(C, A) = \text{Hom}_{\text{dga}}(C, \underset{?}{?})$$

First step:

$$\begin{aligned} \text{Hom}_{\text{alg}}(T_{S^{-1}C}, A) &\stackrel{\substack{\text{homological shift} \\ \text{of deg } -1}}{\cong} \text{Hom}_{-1}(C, A) \\ &\stackrel{\substack{\uparrow \\ \text{Free alg.}}}{\cong} \text{Hom}(C, \underset{\substack{\uparrow \\ \text{free co-alg}}}{T^C S A}) \end{aligned}$$

Need a differential on $T(S^{-1}C)$; define

d_2 to be the unique derivation which extends

$$\begin{aligned} S^{-1}C &\xrightarrow{\Delta} S^{-1}C \otimes C \xrightarrow{S^{-1}} S^{-1} \otimes S^{-1} \otimes C \otimes C \\ &\rightarrow (S^{-1}C) \otimes (S^{-1}C) \hookrightarrow T(S^{-1}C) \end{aligned}$$

Also, let d_1 be the unique derivation that extends d_C

Prop $(d_1 + d_2)^2 = 0$, making

$(T(S^{-1}C), d_1 + d_2)$ a dga.

This is the co-bar construction
of C , denoted $\mathcal{U}C$

claim

$$\text{Hom}_{\text{dga}}(\mathcal{U}C, A) = \text{Tw}(C, A)$$

details in video.

on the right hand side: co-everything

$$BA = (T^c(SA), d_1 + d_2)$$

↓
diff.

↓
use μ on sequences.

details in
video

$$\text{Hom}_{\text{dga}}(C, BA) \cong \text{Tw}(C, A)$$

Remark

Assoc alg \cong augmented unital algs

more in video.

$$\mathbb{K} \begin{array}{c} \xrightarrow{\quad} A \\ \xleftarrow{\quad} \\ \epsilon \end{array}$$