

Finite time computation of A -infinity algebra structures on Ext algebras

Mikael Vejdemo-Johansson

April 4, 2009

Outline

Introduction

The Kadeishvili algorithm

Results

Setting the scene

Let

- ▶ k be a field
- ▶ R a k -algebra
- ▶ (P_*, d) a minimal projective resolution of k over R

Computation of Ext-algebras

Recall that

- ▶ $\text{Ext}_R(k, k)$ is chain homotopy classes of chain maps $P_* \rightarrow P_*$.
- ▶ $\text{End}_R(P_*)$ is the graded R -module of graded maps $P_* \rightarrow P_*$.
- ▶ $\text{End}_R(P_*)$ is a dg-algebra with $\partial f = df + (-1)^{|f|}fd$.
- ▶ Cycles in $\text{End}_R(P_*)$ are graded chain maps.
- ▶ Boundaries are null-homotopic maps.

Hence, $\text{Ext}_R(k, k) = H_* \text{End}_R(P_*)$.

A-infinity algebras

Recall that

Definition

An A_∞ -algebra A is a graded vector space equipped with a family of multilinear operations $\{m_k : A^{\otimes k} \rightarrow a\}_{k \geq 1}$ of degrees $|m_k| = k - 2$ fulfilling the Stasheff axioms:

$$\text{St}_k : \sum_i \pm m_{n-i+1} \circ_j m_i = 0$$

A-infinity algebra morphisms

Definition

A morphism $A \rightarrow B$ of A_∞ -algebras is a family of multilinear maps $\{f_k : A^{\otimes k} \rightarrow B\}_{k \geq 1}$ of degrees $|f_k| = k - 1$ fulfilling the Stasheff morphism axioms

$$\text{St}_k^m : \sum_i \pm f_{n-i+1} \circ_j m_i = \sum_{r, \sum i_k = n} \pm m_r \circ (f_{i_1} \otimes \cdots \otimes f_{i_r})$$

There are formulae for the signs in all these axioms. As the core of my arguments will end up showing that the signs are not perturbed, I will refrain from stating the signs explicitly.

Outline

Introduction

The Kadec algorithm

Results

The Minimality theorem

Theorem (Kadeishvili 1980, many many others)

*Suppose A is a dg-algebra. Then there is an A_∞ -structure on H_*A such that $m_1 = 0$, m_2 is induced by the multiplication of A , and there is a quasi-isomorphism of A_∞ -algebras $f_* : A \rightarrow H_*A$. Furthermore, if A has a unit 1 , then we can choose the A_∞ -structure $\{m_*\}$ on H_*A and the morphism f_* such that $m_k(a_1, \dots, 1, \dots, a_k) = 0$ and $f_k(a_1, \dots, 1, \dots, a_k) = 0$.*

The Kadeishvili algorithm

Reading Kadeishvili's original proof yields an algorithm for the computation of parts of an A_∞ -structure.

Input: elements $a_1, \dots, a_k \in A$.

Output: function values $m_k(a_1, \dots, a_k)$, $f_k(a_1, \dots, a_k)$.

1. Compute

$$\begin{aligned}
 U_k(a_1, \dots, a_k) = & \\
 & \sum_{1 < i < k-1} \pm f_{k-i+1}(a_1, \dots, m_i(a_s, \dots, a_{s+i}), \dots, a_k) - \\
 & \sum_{r \neq 1, k} \pm m_r(f_{i_1}(a_1, \dots, a_{i_1}), \dots, f_{i_r}(a_{k-i_r}, \dots, a_k))
 \end{aligned}$$

The Kadeishvili algorithm

Reading Kadeishvili's original proof yields an algorithm for the computation of parts of an A_∞ -structure.

Input: elements $a_1, \dots, a_k \in A$.

Output: function values $m_k(a_1, \dots, a_k), f_k(a_1, \dots, a_k)$.

1. Compute $U_k(a_1, \dots, a_k)$. This depends only on lower arity operations.
2. $U(a_1, \dots, a_k)$ is a cycle, so it belongs to a homology class. Call this class $m_k(a_1, \dots, a_k)$.

The Kadeishvili algorithm

Reading Kadeishvili's original proof yields an algorithm for the computation of parts of an A_∞ -structure.

Input: elements $a_1, \dots, a_k \in A$.

Output: function values $m_k(a_1, \dots, a_k)$, $f_k(a_1, \dots, a_k)$.

1. Compute $U_k(a_1, \dots, a_k)$. This depends only on lower arity operations.
2. $U(a_1, \dots, a_k)$ is a cycle, so it belongs to a homology class. Call this class $m_k(a_1, \dots, a_k)$.
3. $f_1(m_k(a_1, \dots, a_k))$ is homologous to $U(a_1, \dots, a_k)$, so there is some $w \in A$ such that
$$\partial w = U(a_1, \dots, a_k) - f_1(m_k(a_1, \dots, a_k)).$$
 Set
$$f_k(a_1, \dots, a_k) = w.$$
4. Return $m_k(a_1, \dots, a_k)$ and $f_k(a_1, \dots, a_k)$.

Drawbacks with the Kadeishvili algorithm

- ▶ Will only compute the value of the structure maps for one single parameter set.
- ▶ Unless R has finite projective dimension, $\text{Ext}_R(k, k)$ has infinitely large basis.
- ▶ Kadeshvili's algorithm won't give a complete A_∞ -structure on its own.

Outline

Introduction

The Kadeishvili algorithm

Results

S-linearity

Definitions

- ▶ An A_n -algebra structure is an A_∞ -structure, truncated at m_n : only m_1, \dots, m_n are defined and only $\text{St}_1, \dots, \text{St}_n$ need to hold. Similarly, morphisms between A_n -algebras are defined.
- ▶ We call an A_n or A_∞ algebra A *S-linear*, for $S \subseteq A$ a commutative, associative subalgebra, if for any $a_1, \dots, a_k \in A$, $s \in S$:

$$m_k(a_1, \dots, sa_i, \dots, a_k) = sm_k(a_1, \dots, a_i, \dots, a_k)$$

- ▶ Similarly, we will call a morphism $f_* : A \rightarrow B$ of A_∞ -algebras *S-linear* if for any $a_1, \dots, a_k \in A$ and $s \in S$ (m_2 denoted by \cdot):

$$f_k(a_1, \dots, sa_i, \dots, a_k) = f_1(s) \cdot f_k(a_1, \dots, a_i, \dots, a_k)$$

S -linearity to the rescue

Theorem (V-J)

For R , k , P_* as given in the first slide, write $A = \text{End}_R(P_*)$ and suppose that

- ▶ There is a polynomial subalgebra $S = k[z_1, \dots, z_s] \subseteq H_*A$.
- ▶ H_*A is free as an S -module.
- ▶ We have an S -linear A_{n-1} -algebra structure on H_*A and an S -linear morphism of A_{n-1} -algebras $H_*A \rightarrow A$.
- ▶ All elements $f_1(s)$, for $s \in S$ commute with all elements on the form $f_k(a_1, \dots, a_k)$ for $k < n$.

Then we can choose m_n and f_n so that we get an S -linear A_n -algebra structure.

How does S -linearity help?

- ▶ If H_*A is a *finite* free S -module, then S -linearity helps us reduce the computational workload to compute an A_n -algebra structure with Kadeishvili's algorithm to a finite workload.
- ▶ Regardless of finiteness conditions, S -linearity allows us to excise parts of H_*A from consideration.
- ▶ We get a single recognition criteria while computing iteratively: as long as all $f_1(s)$ commute with all $f_k(a_1, \dots, a_k)$, this method works.

Proof sketch

By investigating the potential terms of $U_k(a_1, \dots, a_k)$, we show that the S -linearity and commutativity conditions globally admit

$$U_k(a_1, \dots, sa_i, \dots, a_k) = f_1(s)U_k(a_1, \dots, a_i, \dots, a_k)$$

Signs remain fixed since S is a *commutative* subalgebra and therefore necessarily concentrated in even degrees.

Finite time computation

Theorem (V-J)

*Suppose that A is a dg-algebra and that H_*A admits an A_{2q-2} -algebra structure with a corresponding quasiisomorphism $H_*A \rightarrow A$ such that all structure maps m_k and all quasiisomorphism component maps f_k vanish for $q \leq k \leq 2q - 2$. Then this A_{2q-2} -algebra structure extends to an A_∞ -algebra structure with no further non-trivial maps introduced.*

Proof.

follows by showing that for all terms in the higher Stasheff morphism axioms, at least one component falls within the interval $q \leq k \leq 2q - 2$, and hence vanishes. \square

These two theorems combined enable us – for good enough algebras A – to compute A_∞ -algebra structures inductively and in finite time.

Computations

- ▶ Let $R = k[x]/x^n$. The A_∞ -algebra structure on $\text{Ext}_R(k, k)$ may be computed in finite time with this approach. The resulting structure was first demonstrated by Madsen (2002).
- ▶ Suppose G is a finite p -group. Then k has a periodic resolution over kG if G is cyclic or generalized quaternionic. The cyclic case is the one above. Preliminary investigations indicate that the quaternionic unit group has too rich structure to be amenable to this approach.
- ▶ For more complex groups – such as for instance the dihedral groups – an approach with regular sequences might work. The current state of these results, however, does not seem to help.

Conclusions

- ▶ The naïve approach to computation does work. In special cases, it can be made to work in finite time.
- ▶ Whether or not the finite time computation simplifications work can be discovered while computing.
- ▶ Whether or not the entire structure is already computed can be discovered while computing.
- ▶ The classes of algebras for which this has been made to work are embarrassingly few.
- ▶ More examples and more work needed.

Any questions?