Journal of Software for Algebra and Geometry

Software for doing computations in graded Lie algebras

CLAS LÖFWALL AND SAMUEL LUNDQVIST

vol 11 2021

JSAG **11** (2021), 9–14
https://doi.org/10.2140/jsag.2021.11.9

The Journal of Software for
Algebra and Geometry

Software for doing computations in graded Lie algebras

CLAS LÖFWALL AND SAMUEL LUNDQVIST

ABSTRACT: We introduce the *Macaulay*2 package *GradedLieAlgebras* for doing computations in graded Lie algebras presented by generators and relations.

1. INTRODUCTION. In order to support computer based research on graded Lie (super-)algebras, we have developed the package *GradedLieAlgebras* as part of [Macaulay2].

The package has basic routines for computing Hilbert series, for doing operations on ideals, subalgebras, derivations, and maps. It also has support for constructing holonomy Lie algebras of arrangements, computing homology and constructing minimal models. For a full list of features, we refer to the documentation of the package [GradedLieAlgebras].

The algorithmic idea used in the package goes back to [Löfwall and Roos 1997, Theorem 5.3], which was used to identify a periodic structure in a certain 1,2-presented Lie algebra. The first author then developed an algorithm and implemented that algorithm in *Mathematica*, under the name [Liedim]. That implementation has been cited or referred to in a number of papers; see for instance [Fröberg and Löfwall 2002; Löfwall et al. 2015; Peeva 2003; Roos 2008].

The aim of this paper is to describe the *Macaulay*2 implementation, which is a major extension of the implementation in *Mathematica*.

In the next two sections, we discuss implementation details and present the algorithmic theory used in the package. In the last section, we give a brief introduction to using the package.

2. REPRESENTING LIE ALGEBRAS IN *Macaulay2*. In order to be able to use the built-in operations in *Macaulay*2, we decided to convert each computational step in the algorithm to a computation in a corresponding polynomial ring over the same field as the Lie algebra. That polynomial ring is referred to as `lieRing` in the code.

Let $\mathfrak{g}$ be a Lie algebra given by a finite set $\{x_i\}$ of generators and a set of relations. The generators have predefined degrees given by a function $\deg : \{x_i\} \to \mathbb{Z}_+$ and the relations are supposed to be homogeneous with respect to this degree function. This makes $\mathfrak{g}$ a *positively graded Lie algebra*. When $\mathfrak{g}$ is a Lie super-algebra, the generators have an additional $\mathbb{Z}/2\mathbb{Z}$-grading, and the relations are then supposed to be homogeneous also with respect to this grading.

MSC2010: primary 17-04, 17B70; secondary 13-04.
Keywords: graded Lie algebras, Hilbert series, holonomy Lie algebras, Koszul duality, differential graded Lie algebras.
GradedLieAlgebras version 3.0

An iterated Lie product in $\mathfrak{g}$ of the form $[x_{i_1}, [x_{i_2}, [x_{i_3}, \ldots, [x_{i_{m-1}}, x_{i_m}], \ldots]]]$ is called a *Lie monomial*. In the program, Lie monomials are identified with monomials in `lieRing`, and the Lie product of two elements is performed by a repeated series of normal form computations in `lieRing`, where the normal form computations are being performed with respect to a family of Gröbner bases in `lieRing`. It is important to understand that we use the Gröbner bases only as a way of doing Gaussian elimination, and that there is no connection to the Buchberger algorithm.

We now describe this correspondence in detail. To make the notation more easy to follow, we will use a slightly different way of naming the generators than in the program.

Each generator x_i in $\mathfrak{g}$ corresponds to n generators $x_{i1}, \ldots, x_{in}$ in `lieRing`, where n is an upper bound on the degrees handled during our computations. A Lie monomial $[x_{i_1}, [x_{i_2}, [x_{i_3}, \ldots, [x_{i_{m-1}}, x_{i_m}], \ldots]]]$ in $\mathfrak{g}$ is represented as a monomial in `lieRing` in the following way. A generator x_i is represented by x_{i1}, and if e is a Lie monomial of degree $d-1$ represented by m, then $[x_i, e]$ is represented by $x_{id} \cdot m$, which is also denoted $x_i.m$.

Remark 1. We were informed by Jörgen Backelin that similar approaches to coding noncommutative monomials as commutative monomials have been considered independently in [Gerasimov 1976; La Scala and Levandovskyy 2009].

From the algorithm described in Section 3, it follows that the basis elements of degree d are of the form $[x_i, e_j]$, with e_j a basis element of degree $< d$.

If e_j is a basis element of degree $r-1$ represented by the monomial m in `lieRing`, and $[x_i, e_j]$ is of degree d but not a basis element, then $x_i.m = x_{ir} \cdot m$ will be the leading monomial of a polynomial in a reduced Gröbner basis associated to degree d. This polynomial then has the form $x_{ir} \cdot m - \sum c_i m_i$, where each m_i corresponds to a Lie monomial in $\mathfrak{g}$ that is a basis element in degree d, and where each c_i is an element in the underlying field.

The reduced Gröbner basis associated to degree d is the degree d part of the reduced Gröbner basis of the ideal generated by the set of polynomials of degree d corresponding to the elements that come from the expressions (2), (3), (5) and (6) in the next section.

3. COMPUTING A VECTOR SPACE BASIS OF A GRADED LIE ALGEBRA IN A GIVEN DEGREE. A Lie (super-)algebra $\mathfrak{g}$ over a field k may be specified by giving a positively graded finite set X of generators and a finite set Y of homogenous relations,

$$y = \sum_{x \in X} \lambda_{x,y} x + \sum_i [g_{i,y}, h_{i,y}],$$

where $\lambda_{x,y} \in k$ and $g_{i,y}$ and $h_{i,y}$ are in $\mathcal{F}(X)$, the free Lie algebra on X over k. Throughout this section we will use the following example to illustrate the steps in the algorithm.

Let $k = \mathbb{Z}/3\mathbb{Z}$. The set of generators is $X = \{a, b, c\}$ where a is odd, and b, c are even, and a, b have degree 1, and c has degree 2. The set of relations is $Y = \{[a, a], [b, [b, a]] - [a, c]\}$. We want

to compute a basis in degree 3, and we will use `lieRing`=$\mathbb{Z}/3\mathbb{Z}[a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3, c_3]$ with GRevLex, where a_i, b_i have degree 1, and c_i has degree 2 for $i = 1, 2, 3$.

The program constructs, degree by degree, starting with degree 1, a graded $\mathfrak{g}$-module M that is a subspace of `lieRing`. The $\mathfrak{g}$-module operation on M is denoted $g.m$, where $g \in \mathfrak{g}$ and $m \in M$. There is also a $\mathfrak{g}$-module map def : $M \to \mathfrak{g}$. Assume $n \geq 1$ and everything is done in degree $< n$. This means that def : $M_{<n} \to \mathfrak{g}/\mathfrak{g}_{\geq n}$ is an isomorphism of $\mathfrak{g}$-modules, with an inverse to be denoted by fed. If $\deg(g) + \deg(m) \geq n$, $g.m$ is defined to be zero.

In the example $\{a_1, b_1\}$ is a basis for M in degree 1 and $\{b_2a_1, c_1\}$ is a basis for M in degree 2. We also have the reduction rules $a_2a_1 \to 0$, $a_2b_1 \to -b_2a_1$, $b_2b_1 \to 0$. Moreover, $\mathrm{def}(a_1) = a$, $\mathrm{def}(b_1) = b$, $\mathrm{def}(b_2a_1) = [b, a]$, $\mathrm{def}(c_1) = c$.

To construct M_n, the first step is to construct a subspace $\hat{M}_n$ of `lieRing`, with basis

$$\{x.m \mid x \in X, m \in M, \deg(x) + \deg(m) = n\} \cup \{x_{i1} \mid x_i \in X, \deg(x_i) = n\}.$$

In a natural way we get an $\mathcal{F}(X)$-module $\hat{M} = M_{<n} \oplus \hat{M}_n$ by first defining the action of X, and then extending this action by the derivation rule. Also def is defined on $\hat{M}_n$ by $\mathrm{def}(x.m) = [x, \mathrm{def}(m)]$ and $\mathrm{def}(x_{i1}) = x_i$. It follows that def is surjective in degree n.

In the example, a basis for $\hat{M}_3$ is

$$\{a_3b_2a_1, a_3c_1, b_3b_2a_1, b_3c_1, c_2a_1, c_2b_1\}. \tag{1}$$

The next step is to divide out by a subspace R_n of $\hat{M}_n$ to obtain a $\mathfrak{g}$-module. For this reason we compute

$$R = Y.\hat{M}. \tag{2}$$

It is a subspace of $\hat{M}_n$, since $M_{<n}$ is a $\mathfrak{g}$-module. A generating set for R_n is obtained by computing $y.m$ for each relation y and basis element $m \in M$ such that $\deg(y) + \deg(m) = n$.

In the example, the space R_3 is spanned by $[a, a].a_1 = 2a.(a_2a_1) = 0$ and $[a, a].b_1 = 2a.(a_2b_1) = -2a_3b_2a_1$, yielding the reduction rule $a_3b_2a_1 \to 0$.

We apply [Löfwall and Roos 1997, Theorem 5.3] to obtain M_n as $\hat{M}_n$ modulo R_n and the expressions

$$\sum_{x \in X} \lambda_{x,y} m_x + \sum_i g_{i,y}.\mathrm{fed}(h_{i,y}) \quad \text{for all } y \in Y_n, \tag{3}$$

$$x.m + \epsilon(m, x)\mathrm{def}(m).m_x, \tag{4}$$

where m_x is the element in `lieRing` corresponding to $x \in X$, and where the last expression is computed for all basis elements $m \in M$ and all $x \in X$ such that $\deg(x) + \deg(m) = n$. Here $\epsilon(m, x)$ is the sign of interchanging the super-elements m and x.

In fact, using a linearization idea described in [Löfwall and Roos 1997], the "commutative" law (4) — the computationally most heavy part — does not need to be checked for all elements. Indeed, in characteristic zero, a basis is computed for the quotient space $\tilde{M}_n$ of $\hat{M}_n$ with respect to (2), (3), and the extra expressions

$$g_{i,y}.\mathrm{fed}(h_{i,y}) + \epsilon(g_{i,y}, h_{i,y})h_{i,y}.\mathrm{fed}(g_{i,y}) \quad \text{for all } y \in Y_n \text{ and for all } i. \tag{5}$$

If $\mathrm{char}(k) > 0$ then a basis for the quotient space $\tilde{M}_n$ of $\hat{M}_n$ is computed with respect to (2), (3), (5), and also the expressions

$$x.m + \epsilon(m, x)\mathrm{def}(m).m_x \tag{6}$$

coming from (4), for which $\deg x$ is a multiple of the characteristic.

Finally, M_n is obtained from $\tilde{M}_n$ by factoring out (4) applied to the basis elements $x.m$ of $\tilde{M}_n$.

In the example, (6) gives nothing, while (3) and (5) give

$$b.b_2a_1 - a.c_1 = b_3b_2a_1 - a_3c_1 \implies a_3c_1 \to b_3b_2a_1,$$
$$b.b_2a_1 + [b, a].b_1 = b_3b_2a_1 - b.(b_2a_1) - a.(b_2b_1) = b_3b_2a_1 - b_3b_2a_1 - 0 = 0,$$
$$a.c_1 + c.a_1 = a_3c_1 + c_2a_1 \implies c_2a_1 \to -a_3c_1 \to -b_3b_2a_1.$$

Hence, we have the reduction rules $a_3b_2a_1 \to 0$ (from (2)), $a_3c_1 \to -b_3b_2a_1$, $c_2a_1 \to b_3b_2a_1$ and hence $\tilde{M}_3$ has the basis $\{b_3b_2a_1, b_3c_1, c_2b_1\}$. Finally, (4) gives the reduction rule $c_2b_1 \to -b_3c_1$ yielding the basis $\{b_3b_2a_1, b_3c_1\}$ for M_3 and $\mathrm{def}(b_3b_2a_1) = [b, [b, a]]$, $\mathrm{def}(b_3c_1) = [b, c]$.

4. USING THE PACKAGE. The main introduction to using the package is by means of the tutorials that are part of the documentation [GradedLieAlgebras]. Here we give three small examples of possible computations.

The most common way to construct a Lie algebra is by means of the constructor `lieAlgebra`. In our first example, we construct the free Lie algebra on three even generators, all of degree 1:

```
i2 : L1=lieAlgebra({a,b,c})
o2 : LieAlgebra
i3 : dims(1,6,L1)
o3 = {3, 3, 8, 18, 48, 116}
i4 : basis(2,L1)
o4 =  {(b a), (c a), (c b)}
```

Here is the example from Section 3:

```
i5 : L2=lieAlgebra({a,b,c}, Field=>ZZ/3, Signs=>{1,0,0}, Weights=>{1,1,2})/{a a, b b a - a c}
o5 : LieAlgebra
i6 : dims(1,5,L2)
o6 = {2, 2, 2, 3, 5}
i7 :  b c c a
o7 = (b a b b c) + (b b a b c) + (b b b b b a)
o7 : L2
i8 :  basis(3,L2)
o8 = {(b b a), (b c)}
```

Let us now give a short example of computing the homology of a Lie algebra. The generators are odd, a and b have degree 1, and homological degree 0, c has degree 2, and homological degree 1. The differential is defined by $a, b \mapsto 0$, $c \mapsto [a, b]$. The homology can now be obtained using `lieHomology`,

`basis` and `dims` (the columns refer to the first degree, and the rows refer to the homological degree). The Lie subalgebras consisting of the cycles and boundaries of the Lie algebra are obtained using `cycles` and `boundaries`. The underlying field is $\mathbb{Q}$ by default.

```
i9  : F3=lieAlgebra({a,b,c},Signs=>1,
         Weights=>{{1,0},{1,0},{2,1}},LastWeightHomological=>true)

o9  : LieAlgebra

i10 : L3 = differentialLieAlgebra{0_F3,0_F3,a b}/{a a, b b}

o10 : L3

o10 : LieAlgebra

i11 : H = lieHomology L3

o11 : H

o11 : VectorSpace

i12 : dims(4,H)

o12 = | 2 0 0 0 |
      | 0 0 2 1 |
      | 0 0 0 0 |
      | 0 0 0 0 |

i13 : basis(4,1,H)

o13 : {(b a c)}

i14 : B = boundaries L3

o14 : B

o14 : LieSubAlgebra

i15 : basis(4,1,B)

o15 : {(a b c) + b a c)}
```

ACKNOWLEDGEMENT. The authors want to thank Dan Grayson for his help with many of the implementation issues we faced during the development of the package.

SUPPLEMENT. The online supplement contains version 3.0 of *GradedLieAlgebras*.

REFERENCES.

[Fröberg and Löfwall 2002] R. Fröberg and C. Löfwall, "Koszul homology and Lie algebras with application to generic forms and points", *Homology Homotopy Appl.* **4**:2, part 2 (2002), 227–258. MR

[Gerasimov 1976] V. N. Gerasimov, "Distributive lattices of subspaces and the word problem for one-relator algebras", *Algebra i Logika* **15**:4 (1976), 384–435, 487. In Russian. MR

[GradedLieAlgebras] C. Löfwall and S. Lundqvist, "GradedLieAlgebras", Macaulay2 package, available at https://github.com/Macaulay2/M2/tree/master/M2/Macaulay2/packages.

[La Scala and Levandovskyy 2009] R. La Scala and V. Levandovskyy, "Letterplace ideals and non-commutative Gröbner bases", *J. Symbolic Comput.* **44**:10 (2009), 1374–1393. MR

[Liedim] C. Löfwall, "Liedim: a Mathematica program for Lie-calculations", available at http://www2.math.su.se/liedim/.

[Löfwall and Roos 1997] C. Löfwall and J.-E. Roos, "A nonnilpotent 1-2-presented graded Hopf algebra whose Hilbert series converges in the unit circle", *Adv. Math.* **130**:2 (1997), 161–200. MR

[Löfwall et al. 2015] C. Löfwall, S. Lundqvist, and J.-E. Roos, "A Gorenstein numerical semi-group ring having a transcendental series of Betti numbers", *J. Pure Appl. Algebra* **219**:3 (2015), 591–621. MR

[Macaulay2] D. R. Grayson and M. E. Stillman, "Macaulay2: a software system for research in algebraic geometry", available at http://www.math.uiuc.edu/Macaulay2.

[Peeva 2003] I. Peeva, "Hyperplane arrangements and linear strands in resolutions", *Trans. Amer. Math. Soc.* **355**:2 (2003), 609–618. MR

[Roos 2008] J.-E. Roos, "The homotopy Lie algebra of a complex hyperplane arrangement is not necessarily finitely presented", *Experiment. Math.* **17**:2 (2008), 129–143. MR

RECEIVED: 21 Aug 2017 REVISED: 22 Jun 2020 ACCEPTED: 28 Aug 2020

CLAS LÖFWALL:

clas.lofwall@gmail.com
Department of Mathematics, Stockholm University, Stockholm, Sweden

SAMUEL LUNDQVIST:

samuel@math.su.se
Department of Mathematics, Stockholm University, Stockholm, Sweden

Journal of Software for Algebra and Geometry vol 11, no 1, 2021

Phylogenetic trees 1
Hector Baños, Nathaniel Bushek, Ruth Davidson, Elizabeth Gross, Pamela E. Harris, Robert Krone, Colby Long, Allen Stewart and Robert Walker

Software for doing computations in graded Lie algebras 9
Clas Löfwall and Samuel Lundqvist

The relative canonical resolution: Macaulay2-package, experiments and conjectures 15
Christian Bopp and Michael Hoff

The FrobeniusThresholds package for Macaulay2 25
Daniel J. Hernández, Karl Schwede, Pedro Teixeira and Emily E. Witt

Computing theta functions with Julia 41
Daniele Agostini and Lynn Chua

Decomposable sparse polynomial systems 53
Taylor Brysiewicz, Jose Israel Rodriguez, Frank Sottile and Thomas Yahl

A package for computations with sparse resultants 61
Giovanni Staglianò

ExteriorModules: a package for computing monomial modules over an exterior algebra 71
Luca Amata and Marilena Crupi

The Schur–Veronese package in Macaulay2 83
Juliette Bruce, Daniel Erman, Steve Goldstein and Jay Yang

admcycles - a Sage package for calculations in the tautological ring of the moduli space of stable curves 89
Vincent Delecroix, Johannes Schmitt and Jason van Zelm

Coding theory package for Macaulay2 113
Taylor Ball, Eduardo Camps, Henry Chimal-Dzul, Delio Jaramillo-Velez, Hiram López, Nathan Nichols, Matthew Perkins, Ivan Soprunov, German Vera-Martínez and Gwyn Whieldon

Threaded Gröbner bases: a Macaulay2 package 123
Sonja Petrović and Shahrzad Zelenberg

Standard pairs of monomial ideals over nonnormal affine semigroups in SageMath 129
Byeongsu Yu

Computations with rational maps between multi-projective varieties 143
Giovanni Staglianò