Lie algebraic computation

Citation for published version (APA):

Cohen, A. M., & Graaf, de, W. A. (1996). Lie algebraic computation. *Computer Physics Communications*, 97(1-2), 53-62. [https://doi.org/10.1016/0010-4655\(96\)00021-5](https://doi.org/10.1016/0010-4655(96)00021-5)

DOI:

[10.1016/0010-4655\(96\)00021-5](https://doi.org/10.1016/0010-4655(96)00021-5)

Document status and date:

Published: 01/01/1996

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

ELSEVIER

Computer Physics Communications 97 (1996) 53–62

Computer Physics
Communications

Lie algebraic computation

A.M. Cohen, W.A. de Graaf

Eindhoven University of Technology, Faculty of Math. & Comp. Science, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Abstract

In this paper we describe some basic algorithms for the structure determination of Lie algebras. They are implemented in a general library of Lie algebra algorithms, called ELIAS (for Eindhoven Lie Algebra System) which is built into the computer algebra package GAP. These activities are part of a bigger project, called ACELA and financed by STW, the Dutch Technology Foundation, which aims at an interactive book on Lie algebras.

1. Introduction

In this paper we describe some basic algorithms for the structure determination of Lie algebras. They are implemented in a general library of Lie algebra algorithms, called ELIAS (for Eindhoven Lie Algebra System) which is built into the computer algebra package GAP. These activities are part of a bigger project, called ACELA and financed by STW, the Dutch Technology Foundation, which aims at an interactive book on Lie algebras (cf. [2]).

Section 2 of this paper gives a global description of the main ways to present Lie algebras on computer. Section 3 briefly discusses the intended functionality offered in the library as well as four specific algorithms.

2. Presentations of Lie algebras

There are various ways to present a Lie algebra over a field F (sometimes one can even take F to be a ring). Here we will discuss three ways, which together seem to be sufficiently comprehensive for most applications. These are (1) by means of generators and relations, (2) as a Lie subalgebra of the general linear Lie algebra $gl_n(F)$, or (3) by means of an explicit multiplication table. These three ways will be called the *basic presentations*.

2.1. Atomic presentations

In order to obtain the basic presentations, we start with the following atomic presentations:

- FL The free Lie algebra $L(A)$ on the alphabet A . Its elements are given as sums of terms, where a *term* is the product of a scalar (from F) with a (bracketed) monomial in the variables from A . From a computational point of view, it is clear that we can restrict ourselves to finite alphabets.

In order to obtain a unique presentation for each element of $L(A)$, the monomials are usually chosen from a *Hall basis*, or, more specifically and more conveniently, to correspond with *Lyndon words* (cf. [9]).

GL The linear Lie algebra $gl_n(F)$ of all $n \times n$ matrices over F . Here the multiplication is $[x, y] = x * y - y * x$ for two matrices, where $*$ stands for the usual matrix multiplication.

SC The abstract Lie algebra L with basis $\{x_i \mid i \in I\}$, where the Lie bracket $[\cdot, \cdot]$ is determined by an explicitly given multiplication table. This multiplication table can, for instance, be represented by providing the so-called *structure constants* c_{ij}^k ($i, j, k \in I$) which are defined by the relation

$$[x_i, x_j] = \sum_{k \in I} c_{ij}^k x_k.$$

Using the bilinearity of the Lie bracket, these are sufficient to calculate the bracket of two arbitrary elements of L .

Remark. The free Lie algebra $L(A)$ (type FL) is infinite-dimensional, except when $|A| \leq 1$. In theory the other two types also admit infinite-dimensional Lie algebras, although in practice such presentations will not be very useful. In case of type GL, these are presented as infinite matrices given by means of functions. In case of type SC, we obtain an infinite-dimensional Lie algebra if I is infinite and c_{ij}^k is given as a function $I^3 \rightarrow k$. Below we shall only deal with presentation FL if infinite-dimensional Lie algebras are concerned; that is, we shall assume that $n < \infty$ and $|I| < \infty$.

2.2. Lie algebra constructors

We can obtain a Lie algebra in one of the basic presentations, by applying the *constructors* **Sub** and **Quo** one or more times to an atomic presentation. These can be used to construct a new Lie algebra L from a given Lie algebra $\tilde{L}$ as follows:

Sub Constructs a Lie subalgebra. This can be done by specifying

- a subset of $\tilde{L}$ consisting of generators of L , or
- a basis of the linear subspace of $\tilde{L}$ underlying L .

Quo Forms the quotient Lie algebra $L = \tilde{L}/I$ with respect to an ideal I of $\tilde{L}$. In the same way as a subalgebra, an ideal I can be specified by giving

- a subset of $\tilde{L}$ consisting of generators of I , or
- a basis of the linear subspace of $\tilde{L}$ underlying I .

From a computational point of view, it is clear that we have to restrict ourselves to finitely generated subalgebras and ideals. This is a real restriction when we are starting from presentation FL, since not every subalgebra of a free Lie algebra (even on a finite alphabet) is finitely generated.

2.3. Basic presentations and transitions

Starting from an atomic presentation and applying one or more of the constructors **Sub** and **Quo** will generally lead to a Lie algebra with quite a complicated description. The simplest and most frequently employed constructions of a Lie algebra L make use of one of the following three *basic presentations*:

FL $L = L(A)/I$ for an ideal I of $L(A)$;

GL $L = \langle X \rangle$ for a subset X of $gl_n(F)$;

SC L given by a basis $\{x_i \mid i \in I\}$ and structure constants c_{ij}^k ($i, j, k \in I$).

If L is known to be semisimple or nilpotent, other efficient presentations are known, depending on the structure of L , e.g., consider the Chevalley generators and Serre relations for semi-simple Lie algebras (cf. [4]). Here we shall not go into those ramifications.

Table 1
Structure constants of the Poincaré algebra

	B_1	B_2	B_3	B_4	B_5	B_6	X_1	X_2	X_3	X_4
B_1	0	0	$2B_4$	$-2B_3$	$-2B_6$	$2B_4$	0	$2X_3$	$-2X_2$	0
B_2	0	0	$2B_3$	$2B_4$	$-2B_5$	$-2B_6$	$2X_1$	0	0	$-2X_4$
B_3	$-2B_4$	$-2B_3$	0	0	B_2	B_1	0	0	$2X_1$	X_3
B_4	$2B_3$	$-2B_4$	0	0	B_1	$-B_2$	0	$-2X_1$	0	$-X_2$
B_5	$2B_6$	$2B_5$	$-B_2$	$-B_1$	0	0	X_3	0	$2X_4$	0
B_6	$-2B_5$	$2B_6$	$-B_1$	B_2	0	0	X_2	$2X_4$	0	0

2.4. Changing presentations

From a theoretical point of view it is known that every finite-dimensional Lie algebra can be presented in any of the three basic presentations. However, for performing computations on finite-dimensional Lie algebras the presentation by means of a commutator table giving the structure constants (SC) seems to be the most suitable one. In the implementation of the algorithms we will therefore assume that every finite-dimensional Lie algebra is presented in this way. For this, and other reasons, we must be able to compute the transition from the first two presentations to a commutator table (SC) presentation and vice versa.

Suppose we start with a presentation of a Lie algebra L of type i and we want to transform it to type j . Then, under item $i \rightarrow j$, we list what we know about the possibility to perform the transition. In view of the above, we restrict ourselves to cases where $SC \in \{i, j\}$.

GL $\rightarrow$ SC This is straightforward using linear algebra for $gl_n(F)$: e.g., find a basis $x_1, \dots, x_n$ for L , and determine the structure constants c_{ij}^k by computing $[x_i, x_j]$ and expressing it as a linear combination of the basis $(x_k)_k$.

SC $\rightarrow$ GL This is possible: it boils down to an effective version of Ado's Theorem (or, in case F has positive characteristic, Iwasawa's Theorem). But no efficient algorithm to this effect has been implemented (yet).

SC $\rightarrow$ FL This is again straightforward. Take the generators to be the basis elements, and the relations to be all commutation relations.

FL $\rightarrow$ SC This can be achieved by applying various kinds of Todd–Coxeter techniques that work in case L is finite-dimensional. The result of applying the **Quo** constructor to a Lie algebra in presentation FL may, but need not be a finite-dimensional Lie algebra. Answering the question whether or not such a quotient algebra is finite-dimensional boils down to solving the word problem, which is known to be unsolvable in general. Nevertheless, in practice there are a number of cases where Todd–Coxeter like techniques can be used to solve it effectively. For instance, this is the case when L admits some kind of positive grading. Such facilities are implemented by M. Roelofs in *Reduce* (cf. [10]), and, recently, by V.P. Gerdt and V.V. Kornyak in *C* code. It is our intention to have these built into ELIAS as well.

By way of example consider the *Poincaré algebra*. This is a 10-dimensional Lie algebra with basis $\{B_1, \dots, B_6, X_1, \dots, X_4\}$ whose structure constants are given by Table 1 (see [7]), with the understanding that the X_i commute: $[X_i, X_j] = 0$ for $i, j = 1, \dots, 4$.

In matrix form, the Lie algebra can be taken to be generated by the following three matrices corresponding to B_1 , B_5 , and X_1 , respectively:

$$\begin{pmatrix} i & 0 & 0 & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & -i & 0 \\ 0 & 0 & 0 & i \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 & i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

In particular, the Poincaré Lie algebra is a quotient of the free Lie algebra on the alphabet $\{B_1, B_5, X_1\}$.

The relations (corresponding to generators of the ideal that has to be divided out to get the Poincaré Lie algebra) can be described by first expressing all other basis elements as products of B_1 , B_5 , and X_1 , and subsequently, substituting these expressions in the multiplications given by the multiplication table. For instance, $B_6 = -\frac{1}{2}[B_1, B_5]$, $B_4 = -\frac{1}{4}[B_1, [B_1, B_5]]$, $B_3 = \frac{1}{8}[B_1, [B_1, [B_1, B_5]]]$, $B_2 = [B_6, B_4]$ express the B -part of the basis in terms of B_1 and B_5 (after expanding the last equation using those for B_4 and B_6 , of course). As an example of the second stage, we substitute the expressions for B_3 and B_4 in the multiplication $[B_1, B_3] = 2B_4$, in order to obtain the relation

$$[B_1, [B_1, [B_1, [B_1, B_5]]]] = -4[B_1, [B_1, B_5]].$$

3. Structure determination

The problem considered here is that of identifying the structure of a Lie algebra given by a table of structure constants. To the best of our knowledge, this problem was first addressed by Beck and Kolman (see, e.g., [1]), and later by Rand, Winternitz, and Zassenhaus, cf. [8]. Recently, Rónyai, Ivanyos, and the authors have taken up the subject again. In this report we attempt to survey some of these developments.

For the duration of this section, L will be a Lie algebra of dimension n over the field F with basis $\{x_1, \dots, x_n\}$ and the Lie multiplication described by

$$[x_i, x_j] = \sum_{k=1}^n c_{ij}^k x_k,$$

where $c_{ij}^k \in F$. This set of structure constants $\{c_{ij}^k\}$ will be the input of our algorithms. Thus, we suppose that L is given in the SC presentation.

It is then straightforward to check whether the multiplication table defines a Lie algebra indeed: by linearity, the Jacobi identity and, if not already taken care of at storage level, antisymmetry, both need only be checked for (triples, respectively, pairs of) standard basis elements.

3.1. Linear algebra

Several substructures of a Lie algebra L given in the SC presentation are readily computed by solving an appropriate set of linear equations over F or applying standard linear algebra: a basis for $[U, V]$ where U and V are subspaces of L , the centraliser and normaliser of a subspace of L ; the derived series and the lower and upper central series of L ; the matrix of the Killing form of L , and the radical of the Killing form; and a test whether L is nilpotent, or solvable. Also computable by linear algebra, but maybe less obviously so, are

- (i) the Jacobson–Morozov construction of an sl_2 containing a given nilpotent element;
- (ii) the Levi–Mal’cev decomposition;
- (iii) if L is semisimple, the Jordan decomposition of elements of L .

All these algorithms have been implemented. Besides these we also have implemented algorithms for finding

- (i) a nonnilpotent element in L ;
- (ii) a Cartan subalgebra and the decomposition of L w.r.t. this algebra;
- (iii) the solvable and the nilpotent radical of L .

We shall discuss the algorithms here, following [12] for (i) and (ii).

3.2. A nonnilpotent element

Suppose that L is not nilpotent. Then, by Engel’s theorem ([5], p 36) there exists an element x in L such that $\text{ad } x$ is not nilpotent. Here we consider the problem of finding such an element.

If F is a field of characteristic 0, then the problem is particularly simple.

Proposition 3.1. Let L be a nonnilpotent Lie algebra over a field of characteristic 0 with basis $\{x_1, \dots, x_n\}$, then the set

$$\{x_1, \dots, x_n\} \cup \{x_i + x_j; 1 \leq i < j \leq n\}$$

contains a nonnilpotent element.

Proof. If L is solvable but not nilpotent then by ([5], Corollary 2, p. 45), we have that the nil radical of L is the set of all nilpotent elements of L . Hence there must be a basis element x_i such that x_i is not nilpotent. On the other hand, if L is not solvable, then ([5], Theorem 5, p. 73) implies that the Killing form of L is not identically zero. It follows that there exist basis elements x_i and x_j for $1 \leq i \leq j \leq n$ such that $\text{Tr}(\text{ad } x_i \cdot \text{ad } x_j) \neq 0$. From

$$\begin{aligned} & \text{Tr}((\text{ad } x_i + \text{ad } x_j)^2) - \text{Tr}((\text{ad } x_i)^2) - \text{Tr}((\text{ad } x_j)^2) \\ &= \text{Tr}(\text{ad } x_i \cdot \text{ad } x_j) + \text{Tr}(\text{ad } x_j \cdot \text{ad } x_i) = 2\text{Tr}(\text{ad } x_i \cdot \text{ad } x_j) \neq 0, \end{aligned}$$

we infer that the elements x_i , x_j and $x_i + x_j$ cannot be all nilpotent. $\square$

If L is of characteristic $p > 0$, then we have to follow another course. This is the general pattern for the algorithms in this section. Since for physicists they are not the most relevant ones, we shall refrain from discussing them all, but we cannot resist giving an idea of how this particular one is dealt with.

Proposition 3.2. Suppose K is a proper subalgebra of L of dimension m such that K acts nilpotently on L . Suppose that x is an element of $N_L(K) \setminus K$. Then either $\text{ad } x$ is not nilpotent or K together with x generates a subalgebra of L of dimension $m + 1$, acting nilpotently on L .

Proof. Let $\bar{K}$ denote the subalgebra generated by K and x . The fact that $\bar{K}$ is of dimension $m + 1$ is trivial. Suppose that $\text{ad } x$ is not nilpotent and set $U = \text{ad}_L(\{x\} \cup K)$. Then U is closed under the bracket operation because $x \in N_L(K)$. Furthermore all elements of U are nilpotent maps. Now by ([5], Theorem 1, p 33) the associative algebra A generated by U is nilpotent. The conclusion is that $\text{ad}_L \bar{K} \subset A$ is also nilpotent. $\square$

If we start with $K = 0$ and repeatedly apply Proposition 3.2 then we either find a nonnilpotent element, or after n steps we have that $K = L$ whence L is nilpotent. Note that $N_L(K) = K$ does not occur for K a proper nilpotent Lie subalgebra of L due to the following result, which can be found in [4].

Lemma 3.3. Let L be a Lie subalgebra of $gl(V)$ with V a finite-dimensional vector space. If every $x \in L$ is a nilpotent endomorphism of V , and $V \neq 0$, then there exists a nonzero $v \in V$ with $L \cdot v = 0$.

To apply it, take V of the lemma to be the quotient vector space L/K and L to be the Lie subalgebra K of L ; then the nonzero vector $v \in L/K$ of the lemma corresponds to a nontrivial extension $Fx + K$ of K , normalizing K .

Algorithm Nonnilpotentelt(L):

$K := 0$;

forever do

if $K = L$ **then return** “ L is nilpotent” **fi**;

$x :=$ an element of $N_L(K)$;

if $\text{ad } x$ is nilpotent **then** $K := K + Fx$ **else return** x **fi**.
od;

3.3. A Cartan subalgebra

If the cardinality of F is big enough, then by [4], Theorem 15.3, L contains a Cartan subalgebra. Here we give an algorithm to find a Cartan subalgebra, again following [12].

First suppose that the cardinality of the field F is at least $n + 1$. For a subset $B = \{b_1, \dots, b_s\}$ of L set $L_0(B) = \{x \in L \mid (\text{ad } b_i)^n x = 0 \text{ for } 1 \leq i \leq s\}$. By $L_0(x)$ we denote $L_0(\{x\})$. The proof of the following lemma is a reformulation of the proof of Lemma 15.2 A in [4].

Lemma 3.4. Suppose that L is not nilpotent. Let Ω be a subset of F of size $n + 1$ and let x be a nonnilpotent element of L . Suppose that $L_0(x)$ is not a nilpotent subalgebra and let y be a nonnilpotent element of $L_0(x)$. Then there exists $c_0 \in \Omega$ such that $L_0(x + c_0(y - x))$ is properly contained in $L_0(x)$.

Proof. Set $K = L_0(x)$ and $D_c = \text{ad}_L(x + c(y - x))$ for $c \in F$. Then by ([4], Lemma 15.1, p 78) K is a subalgebra of L , so that all derivations D_c map K into itself. Hence D_c induces a linear map of the linear space L/K . It follows that the characteristic polynomial of D_c is the product of the characteristic polynomials of D_c on K and on L/K . Let $r = \dim K$ and let f_K be the characteristic polynomial of D_c on K and similarly for $f_{L/K}$. Then

$$f_K(T, c) = T^r + f_1(c)T^{r-1} + \dots + f_r(c)$$

and

$$f_{L/K} = T^{n-r} + g_1(c)T^{n-r-1} + \dots + g_{n-r}(c),$$

where the f_i and the g_i are polynomials in c of degree at most i .

By construction all eigenvectors of $\text{ad } x$ belonging to the eigenvalue 0, lie in K . Hence $g_{n-r}(0) \neq 0$ so that g_{n-r} is not the zero polynomial. Furthermore, because y is a nonnilpotent element of K , there is an f_i such that $f_i(1) \neq 0$. In particular, this f_i is not the zero polynomial. Thus, $f_i g_{n-r}$ is a nonzero polynomial in c of degree at most n . As $|\Omega| = n + 1$, there is a $c_0 \in \Omega$ such that $f_i(c_0)$ and $g_{n-r}(c_0)$ are both nonzero. From $g_{n-r}(c_0) \neq 0$ it follows that $L_0(x + c_0(y - x))$ is contained in K . Finally, $f_i(c_0) \neq 0$ implies that $L_0(x + c_0(y - x))$ is properly contained in K . $\square$

Now this lemma implies that the following algorithm terminates:

Algorithm Cartan(L):

$x := 0$;

forever do

$K := L_0(x)$;

if K is nilpotent **then return** K **fi**;

$y :=$ a nonnilpotent element of K ;

$c :=$ an element of Ω such that $L_0(x + c(y - x))$ is properly contained in K .

$x := x + c(y - x)$;

od;

It is clear that upon termination K is a nilpotent subalgebra of L . Now by [4], Lemma 15.2 B, it follows that K is a Cartan subalgebra.

If F is a finite field with $|F| \leq \dim L$ then we have no guarantee that this algorithm produces a Cartan subalgebra.

3.4. The nil radical

Let A be an associative algebra. An element x in A is called *nilpotent* if $x^m = 0$ for some positive integer m . Furthermore, x is called *strongly nilpotent* if xy is nilpotent for all $y \in A$. Now we define the *radical* of A to be the set

$$\text{Rad}(A) = \{x \in A \mid x \text{ is strongly nilpotent}\}.$$

Here we consider the associative matrix algebra $(\text{ad } L)^*$ generated by $\text{ad } L$. Furthermore, $\text{NR}(L)$ will denote the nil radical of L .

Theorem 3.5. (cf. [5], Theorem 3, p. 36) Let L be a Lie algebra. Then

$$\text{NR}(L) = \text{ad}^{-1}(\text{ad}(L) \cap \text{Rad}((\text{ad } L)^*)).$$

By this theorem, the problem of calculating the nil radical of a Lie algebra is reduced to that of calculating the radical of an associative algebra.

Theorem 3.6. (cf. [3]) Let A be a finite-dimensional (associative) matrix algebra over a field of characteristic 0. Then

$$\text{Rad}(A) = \{x \in A \mid \text{Tr}(xy) = 0 \text{ for all } y \in A\}.$$

Proof. Let R be the set of all $x \in A$ such that $\text{Tr}(xy) = 0$ for all $y \in A$. Then it is clear that $\text{Rad}(A) \subset R$. Let x be an element of R . Then we must show that xy is nilpotent for all $y \in A$. Because of the definition of R , we have that $\text{Tr}((xy)^i) = 0$ for $i \geq 1$. By Newton's identities it now follows that xy is nilpotent and we are done. $\square$

A problem with the algorithm just described is that the associative algebra $(\text{ad } L)^*$ may have a significantly higher dimension than the Lie algebra L itself. Therefore we will describe another method that does not increase the dimension. It works because of the proposition below.

We often abbreviate $\text{ad } L$ by ad . For $k \geq 0$ define subspaces I_k of L in the following way:

$$I_0 = \{x \in L \mid \text{Tr}(\text{ad } x) = 0\},$$

$$I_{k+1} = \{x \in I_k \mid \text{Tr}(\text{ad } y_1 \cdots \text{ad } y_{k+1} \cdot \text{ad } x) = 0 \text{ for all } y_1, \dots, y_{k+1} \in I_k\}.$$

Then $L \supseteq I_0 \supseteq I_1 \supseteq \cdots$. Furthermore, all the I_k can be computed by solving a system of linear equations, so that the following proposition gives a polynomial time algorithm to compute the nil radical.

Proposition 3.7. For the sequence $I_0, I_1, \dots$ defined above we have the following:

- (i) I_k is an ideal of L for $k \geq 0$.
- (ii) $\text{NR}(L)$ is contained in I_k for $k \geq 0$.
- (iii) If $n = \dim L$ then $I_{n-2} = \text{NR}(L)$.

Proof.

(i) For this we use induction on k . Since the trace of a commutator is always 0, we have that $\text{Tr}(\text{ad}[x, y]) = 0$ for all $x \in I_0$ and $y \in L$. Hence I_0 is an ideal of L . For the induction step suppose that I_k is an ideal of L . Choose $x \in I_{k+1}$ and $y \in L$. We need to prove that $[x, y] \in I_{k+1}$.

First note that, for $1 \leq t \leq k+1$ and $y_1, \dots, y_{k+1}$ in I_k , we have the identity

$$\text{Tr}(\text{ad } y_1 \cdots \text{ad } y_{t-1} \text{ad } y_t \text{ad } y_t \cdots \text{ad } y_{k+1} \text{ad } x) = \text{Tr}(\text{ad } y_1 \cdots \text{ad } y_t \text{ad } y_t \text{ad } y_{t+1} \cdots \text{ad } y_{k+1} \text{ad } x).$$

For the difference between these two traces equals

$$\begin{aligned} & \text{Tr}(\text{ad } y_1 \cdots \text{ad } y_{t-1} (\text{ad } y_t \text{ad } y_t - \text{ad } y_t \text{ad } y_t) \text{ad } y_{t+1} \text{ad } y_{k+1} \text{ad } x) \\ &= \text{Tr}(\text{ad } y_1 \cdots \text{ad } y_{t-1} (\text{ad } [y, y_t]) \text{ad } y_{t+1} \cdots \text{ad } y_{k+1} \text{ad } x) = 0, \end{aligned}$$

as, by the induction hypothesis, $[y, y_t]$ is an element of I_k .

Repeated application of the identity gives

$$\text{Tr}(\text{ad } y_1 \cdots \text{ad } y_{k+1} \text{ad } x \text{ad } y) = \text{Tr}(\text{ad } y \text{ad } y_1 \cdots \text{ad } y_{k+1} \text{ad } x) = \text{Tr}(\text{ad } y_1 \cdots \text{ad } y_{k+1} \text{ad } y \text{ad } x),$$

whence

$$\text{Tr}(\text{ad } y_1 \cdots \text{ad } y_{k+1} \text{ad } [x, y]) = 0$$

for all $y_1, \dots, y_{k+1}$ in I_k . But this means $[x, y] \in I_{k+1}$, as required.

(ii) Let $(\text{ad } L)^*$ be the subalgebra of the associative algebra $M_n(F)$ generated by $\text{ad } L$. Let x be an element of $\text{NR}(L)$, then by Theorem 3.5 it follows that $\text{ad } x$ lies in $\text{Rad}((\text{ad } L)^*)$. Hence, by definition of the radical we have that $\text{ad } x \cdot a$ is nilpotent for all a in $(\text{ad } L)^*$. So $\text{Tr}(a \cdot \text{ad } x) = \text{Tr}(\text{ad } x \cdot a) = 0$ for all $a \in (\text{ad } L)^*$. The conclusion is that x lies in I_k for all $k \geq 0$.

(iii) Because $\text{NR}(L)$ is contained in I_{n-2} , we only have to prove that I_{n-2} is nilpotent. Let $x \in I_k$, then by extending a basis of I_k to a basis of L we see that the matrix of $\text{ad } x$ has the form

$$\begin{pmatrix} \text{ad }_{I_k} x & * \\ 0 & 0 \end{pmatrix}.$$

By Engel's theorem ([5], p. 36) an ideal I_k is nilpotent if and only if for all $x \in I_k$ the eigenvalues of the matrix $\text{ad }_{I_k} x$ are all 0. But because $\text{ad }_L x$ has the above mentioned form, this is also the case if all eigenvalues of $\text{ad }_L x$ are 0.

Let x be an element of I_k ($k \geq 1$) and suppose that the eigenvalues of $\text{ad } x$ are $0, \lambda_1, \dots, \lambda_{n-1}$. From the fact that

$$\text{Tr}((\text{ad } x)^l) = 0 \quad \text{for } l = 1, \dots, k+1,$$

it follows that

$$\sum_{i=1}^{n-1} \lambda_i^l = 0 \quad \text{for } l = 1, \dots, k+1.$$

So, if $k = n-2$, then by Newton's identities (see [6], p 287) we have that all λ_i must be 0 and we are done. $\square$

Remarks.

(i) From $\text{Tr}(\text{ad } z_1 \cdots \text{ad } z_k \text{ad } x) = 0$ for all $x \in I_k$ and $z_1, \dots, z_k \in I_{k-1}$ and

$$\text{ad } y_1 \cdots \text{ad } y_i \text{ad } y_j \cdots \text{ad } y_{k+1} = \text{ad } y_1 \cdots \text{ad } y_j \text{ad } y_i \cdots \text{ad } y_{k+1} + \text{ad } y_1 \cdots \text{ad } [y_i, y_j] \cdots \text{ad } y_{k+1}$$

it is obvious that, in the calculation of I_{k+1} , we only need consider sequences of the form

$$\text{ad } y_{i_1} \text{ad } y_{i_2} \cdots \text{ad } y_{i_{k+1}},$$

where $1 \leq i_1 \leq i_2 \leq \cdots \leq i_{k+1}$.

(ii) In the characteristic p case the proof only fails because the use of Newton's identities may not lead to the desired conclusion. However, if $p > \dim L$, then this problem does not occur. So the algorithm also works in that case.

(iii) An advantage of this algorithm lies in the fact that in most cases the nil radical will be equal to I_k with $k \ll n - 2$. However, there are Lie algebras for which this does not hold, as the following example shows. Set $\zeta = e^{2\pi i/n}$ and let L be the Lie algebra with basis $x_0, \dots, x_n$ and commutation relations

$$\begin{aligned} [x_0, x_i] &= \zeta^{i-1} x_i & \text{for } 1 \leq i \leq n, \\ [x_i, x_j] &= 0 & \text{for } 1 \leq i, j \leq n. \end{aligned}$$

Then it is easily seen that $\text{NR}(L) = \langle x_1, \dots, x_n \rangle$, but $x_0 \in I_k$ for $0 \leq k < n - 1$.

3.5. The solvable radical

Recall that the solvable radical, denoted by $\text{Rad}(L)$, is the biggest solvable ideal in L . In the characteristic 0 case, it can be characterized as the linear subspace of L perpendicular to $[L, L]$, which reduces it to a linear computation.

In the arbitrary characteristic, it can be found by first computing the nil radical $\text{NR}(L)$ (see [11]), then the nil radical of $L/\text{NR}(L)$, and so on, until the sequence stabilizes, and next taking the inverse image of the stabilized one in L .

3.6. Further work

Also implemented is an algorithm for the decomposition of L if it is semisimple. It mainly follows the ideas described in [8].

Acknowledgements

We are greatly indebted to Nico van den Hijligenberg and Marcel Roelofs for their help in setting up the package and preparing earlier versions of the first part of this text. Inspiring conversations with Lajos Rónyai are also gratefully acknowledged.

References

- [1] R.E. Beck, B. Kolman and I.N. Stewart, Computing the structure of a Lie algebra, in: Computers in Non-associative Rings and Algebras (Academic Press, New York, 1977).
- [2] A.M. Cohen and L. Meertens, The ACELA project: Aims and Plans, to appear in: Human Interaction for Symbolic Computation, Texts and Monographs in Symbolic Computation, N. Kajler, ed. (Springer-Verlag, Vienna, 1995).
- [3] L.E. Dickson, Algebras and Their Arithmetics (University of Chicago Press, Chicago, 1923).
- [4] J.E. Humphreys, Introduction to Lie Algebras and Representation Theory (Springer-Verlag, New York, Heidelberg, Berlin, 1972).
- [5] N. Jacobson, Lie Algebras (Dover, New York, 1979).
- [6] A.I. Kostrikin, Introduction to Algebra (Springer-Verlag, New York, Heidelberg, Berlin, 1982).

- [7] J. Patera, P. Winternitz and H. Zassenhaus, Continuous subgroups of the fundamental groups of physics. I-General method and the Poincaré group, *J. Math. Phys.* 16 (1975) 1597–1641.
- [8] D. Rand, P. Winternitz and H. Zassenhaus, On the Identification of a Lie algebra given by its structure constants. I. Direct decompositions, Levi decompositions, and nilradicals, *Linear Algebra and its Applications* 109 (1988) 197–246.
- [9] C. Reutenauer, Free Lie algebras, LMS Monographs, New series, Vol. 7 (Oxford University Press, Oxford, 1993).
- [10] G.H.M. Roelofs, The LIESUPER package for REDUCE, Memorandum 943 (University of Twente, The Netherlands, 1991).
- [11] L. Rónyai, Computing the Structure of Finite Algebras, *J. Symbolic Computation* 9 (1990) 355–373.
- [12] L. Rónyai, G. Ivanyos and W. de Graaf, Computing Cartan subalgebras of Lie algebras, preprint (1994); AAECC, to appear.