Discrete Mathematics and Theoretical Computer Science **1**, 1997, 129–138

Computations in finite-dimensional Lie algebras[†]

A. M. Cohen[1], W. A. de Graaf[1] and L. Rónyai[2]

[1] *Faculty of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands*
[2] *Computer and Automation Institute, Hungarian Academy of Sciences, 1111 Budapest, Lágymányosi u. 11, Hungary*

This paper describes progress made in context with the construction of a general library of Lie algebra algorithms, called ELIAS (Eindhoven LIe Algebra System), within the computer algebra package GAP. A first sketch of the package can be found in Cohen and de Graaf[1]. Since then, in a collaborative effort with G. Ivanyos, the authors have continued to develop algorithms which were implemented in ELIAS by the second author. 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. Cohen and Meertens [2]). This paper gives a global description of the main ways in which to present Lie algebras on a computer. We focus on the transition from a Lie algebra abstractly given by an array of structure constants to a Lie algebra presented as a subalgebra of the Lie algebra of $n \times n$ matrices. We describe an algorithm typical of the structure analysis of a finite-dimensional Lie algebra: finding a Levi subalgebra of a Lie algebra.

Keywords: Lie algebra algorithms, ELIAS

1 Presentations of Lie Algebras

The three most common ways to present a Lie algebra over a field F are

- FL by means of generators and relations,
- GL as a Lie subalgebra of the general linear Lie algebra $gl_n(F)$, or
- SC by means of an explicit multiplication table.

These three ways will be called the *basic presentations*. Together they suffice for most applications.

[†]Research supported in part by Hungarian National Foundation for Scientific Research grant T016503 and by the Dutch Technology Foundation STW.

1.1 Basic Presentations

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 can be presented by sums of terms, where a *term* is the product of a scalar (from F) with a (square bracketed) monomial in the variables from A. The Lie bracket is (formally) F-bilinear, whence determined by its value on a pair of monomials; the Lie bracket of two monomials s and t is just the bracketed monomial $[s, t]$. This presentation is not unique as one has to divide out by the anti-symmetry and the Jacobi identities. Normal forms algorithms exist for $L(A)$ – see below.

GL The linear Lie algebra $gl_n(F)$ of all $n \times n$ matrices over F. Here the Lie bracket of the matrices x and y is $[x, y] = x * y - y * x$, 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, consisting of the so-called *structure constants* c_{ij}^k $(i, j, k \in I)$ which are defined by the relations

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

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

Starting from an atomic presentation, we can obtain a presentation for a subalgebra or a quotient algebra. 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 semi-simple or nilpotent, other efficient presentations are known, depending on the structure of L, e.g. the Chevalley generators and Serre relations for semi-simple Lie algebras (cf. Humphreys [3]). Here we shall not go into those ramifications.

1.2 Example

We give a basic presentation of each kind for the *3-dimensional Heisenberg algebra*. This is a Lie algebra with basis $\{x, y, z\}$ whose structure constants are given by the following table:

H	x	y	z
x	0	z	0
y	$-z$	0	0
z	0	0	0

In matrix form, H can be taken to be generated by the following three matrices corresponding to x, y, z, respectively:

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Clearly, H is a quotient of the free Lie algebra on the alphabet $\{X, Y, \}$. The relations (corresponding to generators of the ideal that has to be divided out to get H) can be described by first expressing all other basis elements as products of x and y, and subsequently substituting these expressions in the multiplications given by the multiplication table. Thus, one easily finds that H is a quotient of the free Lie algebra on the alphabet $\{X, Y\}$ with relations

$$0 = [X, [X, Y]], \quad 0 = [Y, [X, Y]]$$

1.3 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. 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*.

The transitions

$$\text{GL} \to \text{SC} \to \text{FL}$$

are straightforward. The first transition can be achieved by using linear algebra for $gl_n(F)$. Find a basis $x_1, \ldots, x_n$ 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 elements x_k. For the second transition (SC $\to$ FL) we can take the generators to be the basis elements, and the relations to be all commutation relations.

The reverse arrows are significantly harder.

1.4 Todd–Coxeter Type Algorithms

In case L is finite dimensional, the transition FL $\to$ SC can be achieved by applying various kinds of Todd–Coxeter techniques.

The first and perhaps most practical approach is to start with a unique presentation for each element of $L(A)$. This can be done by allowing, in the presentation of a member of $L(A)$, only those monomials to occur that belong to a so-called *Hall basis* – cf. Reutenauer [4]. A Hall set is an ordered set of monomials H in the free magma $M(A)$ on A with the properties that A is contained in H, that $[g, h] \in H$ for $g, h \in M(A)$ implies $h \in H$ and $[g, h] < h$, and that, for any magma element $[g, h]$ in $M(A) \setminus A$, we have

$$[g, h] \in H \Leftrightarrow \{g, h \in H \text{ and } g < h \text{ and } (g = [x, y] \Rightarrow y \geq h)\}$$

A basis of $L(A)$ is obtained from H by interpreting its elements as members of $L(A)$; this is the corresponding Hall basis. Hall bases exist. Thus, $L(A)$ can be viewed as an infinite-dimensional Lie algebra (provided $|A| > 1$) with an ordered basis. Moreover, given an arbitrary monomial in $L(A)$, by use of the Jacobi identity it can easily be rewritten to a normal form: a linear combination of Hall monomials.

Given a finite subset R of $L(A)$, the Todd–Coxeter algorithms search for a (finite) set S of Hall monomials of $L(A)$ such that each element of the quotient Lie algebra $L(A)/I$, where I is the ideal generated by R, has an inverse image in the linear span of S. Initially, in the algorithm, the set S is taken to be A. The strategy is to extend S in such a manner that the multiplication of any two of its members gives an element of the linear span of S. To this end, combinations of the following two steps are taken:

1. Add to S monomials obtained from multiplication of two members of S.
2. Use linear relations of the form $[[\cdots[r,t],t_1]\cdots,t_n]$ for $r \in R$ and $t_i \in A$ to replace members of S by smaller Hall monomials.

Various strategies for applying Steps 1 and 2, etc. are possible, but clearly, such a procedure will only terminate if the Lie algebra $L = L(A)/I$ is finite dimensional. By a result in Ufnarovsky [5], the converse is true, i.e. such a procedure does indeed terminate if $\dim(L) < \infty$. This algorithm has been implemented by Gerdt and Kornyak in C code (see Gerdt and Kornyak [7]).

Another approach has been chosen by Roelofs in `Reduce` (cf. Roelofs [6] and Gragert [8]). Its theoretical foundation is explained in van Leeuwen and Roelofs [9]. Here, instead of starting with a Hall basis in $L(A)$, the authors work in the universal anti-commutative tensor algebra on the (formal) linear F vector space spanned by A, divide out the relations R, and subsequently impose instances of the Jacobi identity (instead of the consequences of the relations as in Step 2 above).

It is our intention to have both approaches built into ELIAS.

1.5 Ado's Theorem

The remaining transition SC $\to$ GL can be seen as an effective version of Ado's theorem (see Bourbaki [10], Chapt. VI). Up to now, no satisfactory complete solution to this problem is known. For instance, in Bourbaki [10], an effective solution is described, but a simple complexity analysis gives a clear signal not to attempt an implementation of the implicitly given algorithm.

In this section we try to find matrix representations of finite-dimensional Lie algebras defined over a field of characteristic 0, given by structure constants. We shall succeed here only for some special classes of Lie algebra. Throughout L will be a finite-dimensional Lie algebra with basis $\{x_1, \ldots, x_n\}$. The universal enveloping algebra of L will be denoted by $U(L)$. We first recall some results proved in [11], Chapter VI.

Proposition 1.1 *The Lie algebra L has a faithful finite-dimensional representation if and only if there is an ideal I of $U(L)$ of finite codimension such that $L \cap I = 0$.*

Proof. We only prove the if part. For the proof of the other direction we refer to Jacobson [11]. Suppose that I is an ideal of $U(L)$ of finite codimension such that $L \cap I = 0$. Set $A = U(L)/I$, then A is a finite-dimensional associative algebra containing L. This algebra contains an identity element. From this it follows that the regular representation (sending an element $x \in A$ to the matrix of the right multiplication by x in A) is faithful. By restricting this representation to the subspace $L \subset A$ we obtain a faithful representation of L in $gl(A)$. □

An element a of an associative algebra A over the field F is called *algebraic* if there is a nonzero polynomial $f \in F[z]$ such that $f(a) = 0$. If I is an ideal of A then a is called *algebraic modulo I* if $f(a) \in I$ for some nonzero polynomial $f \in F[z]$.

Lemma 1.2 *Let $\{x_1, \ldots, x_n\}$ be a basis of L. An ideal I of $U(L)$ is of finite codimension if and only if every x_i is algebraic modulo I.*

Proof. (cf. Jacobson [11]) Let $f_i \in F[z]$ be a nonzero polynomial such that $f_i(x_i) \in I$. Let n_i be the degree of f_i. Then any power of x_i is congruent modulo I to a linear combination of the elements $1, x_i, x_i^2, \ldots, x_i^{n_i-1}$. Hence a monomial

$$x_1^{k_1} \cdots x_n^{k_n}$$

is congruent modulo I to a sum of monomials of the form $x_1^{m_1} \cdots x_n^{m_n}$ where $0 \leq m_i < n_i$.

Now by the Poincaré–Birkhoff–Witt theorem ([11], p. 156), we have that $U(L)/I$ is finite dimensional. This proves the 'if' part. The other implication is trivial. □

Let N be a nilpotent Lie algebra, and let q be the smallest integer such that all $q+1$-fold brackets of elements of N are zero. Then q is called the *nilpotency class* of N. The following generalization of Ado's theorem was proved in Block [12].

Theorem 1.3 *Let L be a finite-dimensional Lie algebra. Suppose K is a nilpotent ideal of L of nilpotency class q. Then there exists a faithful finite-dimensional representation ϕ of L such that $\phi(x_1 \cdots x_{q+1}) = 0$ for all $x_1, \ldots, x_{q+1} \in K$.*

Proof. See Block [12]. □

We now describe two cases where we can find ideals I of $U(L)$ of finite codimension such that $L \cap I = 0$.

Proposition 1.4 *Let L be a nilpotent Lie algebra of nilpotency class q. Let I_k be the ideal of $U(L)$ generated by $\{x_1^k, \ldots, x_n^k\}$. Then I_k has finite codimension and there is an m such that $1 < m \leq q+1$ and $L \cap I_m = 0$.*

Proof. The fact that I_k has finite codimension follows from Lemma 1.2. By Theorem 1.3 there exists a faithful finite-dimensional representation of L such that all elements of L are mapped to nilpotent linear transformations. Let I be the ideal corresponding to this representation. It follows that there is an $m > 1$ such that $x_i^m \in I$ for $1 \leq i \leq n$. Now $I_m \subset I$ and $I \cap L = 0$, so that $I_m \cap L = 0$. Take m minimal such that $I_m \cap L = 0$; clearl $m > 1$. By Theorem 1.3 we infer that there exists a faithful finite-dimensional representation ϕ of L such that $\phi(x^{q+1}) = 0$ for all $x \in L$. Hence, $I_{q+1} \cap L = 0$. It follows that $1 < m \leq q+1$. □

Lemma 1.5 *Let $f_1, \ldots, f_n \in F[z]$ be polynomials such that $f_i(\operatorname{ad} x_i) = 0$. Let I be the ideal of $U(L)$ generated by $\{f_1(x_1), \ldots, f_n(x_n)\}$. If $x \in L \cap I$ then $x \in Z(L)$.*

Proof. Let

$$x = \sum_{i=1}^{n} g_i f_i(x_i) h_i \tag{1}$$

be an element of $L \cap I$, where g_i and h_i are elements of $U(L)$. Let ϕ denote the extension of the map $\operatorname{ad} : L \to \operatorname{End}_F(L)$ to $U(L)$. We have

$$\operatorname{ad} x = \phi(x) = \sum_{i=1}^{n} \phi(g_i)\phi(f_i(x_i))\phi(h_i) = \sum_{i=1}^{n} \phi(g_i) f_i(\operatorname{ad} x_i)\phi(h_i) = 0 \tag{2}$$

implying that $x \in Z(L)$. $\square$

The next statement concerns Lie algebras L for which $Z(L) \cap [L, L] = 0$. Such a Lie algebra has a basis $\{x_1, \dots, x_n\}$ over F such that

1. $\{x_1, \dots, x_s\}$ is a basis of $Z(L)$ (where $0 \le s \le n$).

2. The space spanned by $\{x_{s+1}, \dots, x_n\}$ is a subalgebra of L.

A basis with these properties is called a *good basis* of L. Starting from an arbitrary basis of L, it is easy to obtain a good basis.

Proposition 1.6 *Suppose that $Z(L) \cap [L, L] = 0$ and let $\{x_1, \dots, x_n\}$ be a good basis basis of L over F. We define the polynomials $f_i \in F[z]$ as follows: if $1 \le i \le s$, then $f_i = z^2$. Otherwise, if $s < i \le n$, then f_i is the minimal polynomial of $\operatorname{ad} x_i$ over F. Let I be the ideal of $U(L)$ generated by $\{f_1(x_1), \dots, f_n(x_n)\}$. Then I is of finite codimension and $L \cap I = 0$.*

Proof. By Lemma 1.2, I has finite codimension. Let x be an element of $U(Z(L)) \cap I$. We can write

$$x = \sum_{i=1}^{s} e_i x_i^2 + \sum_{i=1}^{s} g_i x_i^2 + \sum_{i=s+1}^{n} p_i f_i(x_i) q_i$$

where the e_i are elements of $U(Z(L))$, the g_i are sums of monomials, each monomial containing at least one x_i such that $i > s$ and p_i, q_i are arbitrary elements of $U(L)$. In the process of straightening monomials in $x_1, \dots, x_n$ to express them as linear combinations of standard monomials we make substitutions of the form

$$x_i x_j = x_j x_i + \sum_{k=1}^{n} c_{ij}^k x_k$$

By condition (2) above, we have that $c_{ij}^k = 0$ for $k = 1, \dots, s$. This, together with the fact that the constant term of f_i is 0, implies that no monomial of $\sum p_i f_i(x_i) q_i$ lies in $U(Z(L))$. The same is valid for $\sum g_i x_i^2$. It follows that $\sum g_i x_i^2 + \sum p_i f_i(x_i) q_i = 0$. The conclusion is that $U(Z(L)) \cap I$ is the ideal in $U(Z(L))$ generated by $\{x_1^2, \dots, x_s^2\}$. Now let $x \in I \cap L$. From Lemma 1.5, we infer that $x \in Z(L)$. Hence $x \in U(Z(L)) \cap I$. But $U(Z(L))$ is the commutative polynomial ring over the variables $x_1, \dots, x_s$. From this we see that the ideal $I \cap U(Z(L))$ of $U(Z(L))$ does not contain linear elements. It follows that $x = 0$. $\square$

Remarks.

1. If we have an ideal I of $U(L)$ of finite codimension, then by the algorithm described in Linton [13], [14] (which is implemented in GAP) we can calculate a basis and a multiplication table of $U(L)/I$. Hence, we can check whether $L \cap I = 0$ and we can calculate the matrices corresponding to the representation determined by I.

2. In the case where $Z(L) \cap [L, L] = 0$ (Proposition 1.6), we can calculate a good basis of L. The first s basis elements form a basis of $Z(L)$. The next basis elements will form a basis of $[L, L]$. Finally we complete the basis.

The representation obtained by Proposition 1.6 is closely related to the following one. Using a good basis we see that we can write

$$L = Z(L) \oplus K$$

where K is an ideal without centre. A faithful representation is given by the direct sum of the adjoint representation of K and a faithful representation of $Z(L)$ (which is easy to construct).

Example. Consider the Heisenberg algebra H of 1.2. A vector space basis of H is $\{x, y, z\}$ and the Lie bracket is specified by $[x, y] = z,\ [x, z] = [y, z] = 0$. Following Proposition 1.4, we try the ideal I_2 of $U(L)$ generated by $\{x^2, y^2, z^2\}$. Using the vector enumeration package of GAP, (which uses the algorithm described in Linton [13], [14]) we find that $\{1, x, y, z, xy\}$ is a basis of $U(L)/I_2$. The matrices of the corresponding representation are

$$x \mapsto \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix},\ y \mapsto \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix},\ z \mapsto \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

which indeed gives a GL presentation of H (different from that given in 1.2).

2 Levi Decomposition

The algorithm to be discussed in this section is part of a suite of routines built to identify the structure of a Lie algebra given by a table of structure constants. See Cohen and de Graaf [1] for more details on other routines, and Rand *et al.* [17] for earlier versions of the specific algorithm under discussion. Although we have no direct evidence, the fact that the Levi decomposition is used in various proofs of Ado's Theorem might indicate its use in constructing matrix representations.

For the duration of this section, L will be a Lie algebra of dimension n over the field F of characteristic 0, given in the SC presentation. Thus, L has a basis $\{x_1, \ldots, x_n\}$ and its Lie multiplication is 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.

Theorem 2.1 (Levi) *If L is not solvable, then there exists a (necessarily semi-simple) subalgebra S of L such that L is the semidirect product of S and the solvable radical R of L.*

Proof. See Jacobson [11], Section III.9. □

The subalgebra S in the conclusion of this theorem is called a *Levi subalgebra* of L. For solvable Lie algebras not much structure theory is known. So the computational analysis of the structure of the Lie algebra is not very promising in this case. On the other hand, for semi-simple Lie algebras a very elaborate theory is available. So the structure of the Levi subalgebra may be analysed in detail as well as the action of this subalgebra on the solvable radical R. Hence the importance of being able to calculate a Levi subalgebra. For the algorithm computing a Levi subalgebra we follow Rónyai *et al.* [15].

In the sequel R^k will denote the ideal

$$[R,[R,[\cdots,[R,R]\cdots]]] \ (k \text{ copies of } R)$$

By the following lemma we can reduce the problem of calculating a Levi subalgebra to the case where the solvable radical is nilpotent.

Lemma 2.2 *Let S_1 be the inverse image in L of a Levi subalgebra of L/R^2. If S is a Levi subalgebra of S_1, then S is a Levi subalgebra of L.*

Proof. (cf.Jacobson [11], Sect. III.9) It is clear that S is a semi-simple subalgebra of L. Furthermore, R^2 is the solvable radical of S_1. Hence

$$L = R + S_1 = R + R^2 + S = R + S$$

It follows that S is a Levi subalgebra of L. □

Since the radical of S_1 (which is R^2) and the radical of L/R^2 (which is abelian) are nilpotent, we can reduce to the case where the solvable radical is nilpotent. Now suppose that the solvable radical R of L is nilpotent. Let

$$R = R^1 \supset R^2 \supset \cdots \supset R^m = 0$$

be the lower central series of R. We note that this series can be computed efficiently (see Beck em et al. [16]).

Let $\{u_1, \ldots, u_s\}$ be a maximal linearly independent set in the complement of R. Then we have the following commutation relations:

$$[u_i, u_j] = \sum_{k=1}^{s} \gamma_{ij}^k u_k \mod R^1$$

and the u_i span a Levi subalgebra modulo R^1. We are looking for elements $y_1, \ldots, y_s$ of L that span a Levi subalgebra modulo $R^m = 0$. To this end, we construct a series y_i^t for $1 \le i \le s$ and $1 \le t \le m$ such that $\{y_1^t, \ldots, y_s^t\}$ spans a Levi subalgebra modulo R^t, i.e.

$$[y_i^t, y_j^t] = \sum_{k=1}^{s} \gamma_{ij}^k y_k^t \mod R^t$$

For the initialization we set $y_i^1 = u_i$ for $1 \le i \le s$. We now describe the iteration step. We define a vector space V_t by the formula $R^t = R^{t+1} \oplus V_t$. We set $y_i^{t+1} = y_i^t + v_i^t$ where $v_i^t \in V_t$ for $1 \le i \le s$ and require that

$$[y_i^{t+1}, y_j^{t+1}] = \sum_{k=1}^{s} \gamma_{ij}^k y_k^{t+1} \mod R^{t+1}$$

This is equivalent to

$$[y_i^t, v_j^t] + [v_i^t, y_j^t] + [v_i^t, v_j^t] = \sum_{k=1}^{s} \gamma_{ij}^k y_k^t + \sum_{k=1}^{s} \gamma_{ij}^k v_k^t - [y_i^t, y_j^t] \mod R^{t+1}$$

Since $[v_i^t, v_j^t] \in R^{t+1}$ and $[y_i^t, v_j^t] = [u_i, v_j^t] \bmod R^{t+1}$ we have that this is equivalent to

$$[u_i, v_j^t] + [v_i^t, u_j] - \sum_{k=1}^{s} \gamma_{ij}^k v_k^t = \sum_{k=1}^{s} \gamma_{ij}^k y_k^t - [y_i^t, y_j^t] \bmod R^{t+1}$$

This is a system of equations for the v_i^t. Since the equations are modulo R^{t+1}, the left-hand side as well as the right-hand side can be viewed as elements of V_t. By Levi's theorem applied to the Lie algebra L/R^{t+1} this system has a solution. The conclusion is that after $m - 1$ iteration steps we have found a Levi subalgebra of L.

Remark. The method described here runs in polynomial time. This fact is proved in [15].

References

[1] Cohen, A. M. and de Graaf, W. A. (1996). Lie algebraic computation. *Computer Physics Communications* **989** 1–10.

[2] Cohen, A. M. and Meertens, L. (1995). The ACELA project: Aims and Plans. In: N. Kajler, editor, *Human Interaction for Symbolic Computation*. Texts and Monographs in Symbolic Computation. Springer-Verlag.

[3] Humphreys, J. E. (1972). *Introduction to Lie Algebras and Representation Theory*. Springer-Verlag.

[4] Reutenauer, C. (1993). *Free Lie algebras*. LMS Monographs, New series, 7, Oxford University Press.

[5] Ufnarovsky, V. A. (1995). Combinatorial and asymptotic methods in algebra. In: A. I. Kostrikin and I. R. Shafarevich, editors, *Algebra VI*, EMS 57. Springer-Verlag.

[6] Roelofs, G. H. M. (1991). The LIESUPER package for REDUCE. *Memorandum 943*, University of Twente, The Netherlands.

[7] Gerdt, V. P. and Kornyak, V. V. (1995). Construction of finitely presented Lie algebras and superalgebras. *J. Symb. Comp.* **11**.

[8] Gragert, P. K. H. (1989). Lie algebra computations. *Acta Applicandae Mathematicae* **16** 231–242.

[9] van Leeuwen, M. A. A. and Roelofs, G. H. M. (1997). Algorithms for Constructing algebras by generators and Relations. *J. Pure & Applied Algebra* **117–118** 431–445.

[10] Bourbaki, N. (1971). *Groupes et Algèbres de Lie (2nd ed.)*. Hermann, Paris.

[11] Jacobson, N. (1979). *Lie Algebras*. Dover.

[12] Block, R. E. (1986). On nilpotency in the Ado–Harish–Chandra Theorem on Lie algebra representations. *Proc. Amer. Math. Soc.* **98** 406–410.

[13] Linton, S. A. (1991). Constructing Matrix Representations of Finitely Presented Groups. *J. Symbolic Computation* **12** 427–438.

[14] Linton, S. A. (1993). On Vector Enumeration. *Linear Alg. Appl.* **192** 235–248.

[15] Rónyai, L., Ivanyos, G., Küronya, A. and de Graaf, W. A. (1995). Computing Levi decompositions. Preprint.

[16] Beck, R. E., Kolman, B. and Stewart, I. N. (1977). Computing the structure of a Lie algebra. In: *Computers in Non-associative Rings and Algebras*. Academic Press.

[17] Rand, D., Winternitz, P. and Zassenhaus, H. (1988). 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** 197–246.