

Algorithms for finite-dimensional Lie algebras

Citation for published version (APA):

Graaf, de, W. A. (1997). *Algorithms for finite-dimensional Lie algebras*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mathematics and Computer Science]. Technische Universiteit Eindhoven.
<https://doi.org/10.6100/IR495936>

DOI:

[10.6100/IR495936](https://doi.org/10.6100/IR495936)

Document status and date:

Published: 01/01/1997

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.

Algorithms for Finite-Dimensional Lie Algebras

Willem A. de Graaf

Algorithms for Finite-Dimensional Lie Algebras

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van
de Rector Magnificus, prof.dr. M. Rem, voor een
commissie aangewezen door het College van
Dekanen in het openbaar te verdedigen op
dinsdag 10 juni 1997 om 16.00 uur

door

Willem A. de Graaf

geboren te Gorinchem

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr. A.M. Cohen

en

prof.dr. A.E. Brouwer

Contents

1	Introduction	1
1.1	Notation, definitions, and basic theory	2
1.2	Presentations of Lie algebras	7
1.3	Complexity	8
1.4	Earlier work	9
1.4.1	Product spaces	9
1.4.2	The centre	9
1.4.3	The centraliser	10
1.4.4	The normaliser	10
1.4.5	The solvable radical	10
1.4.6	The direct sum decomposition	11
1.4.7	The Levi decomposition	13
1.4.8	Zeros of polynomials	14
2	The nilradical	17
2.1	Known algorithms	17
2.2	The Downward Method	20
2.3	Evaluation	25
2.4	Acknowledgements	26
3	Cartan subalgebras	27
3.1	Known algorithms	27

3.2	Locally regular elements	28
3.3	Finding a non-nilpotent element in a Lie algebra	31
3.4	Cartan subalgebras	33
3.5	Evaluation	34
3.6	Acknowledgement	37
4	The decomposition of a semisimple Lie algebra	39
4.1	The generalised Cartan decomposition	39
4.2	Calculating a generalised Cartan decomposition	42
4.2.1	Splitting elements	42
4.2.2	Decomposable elements	45
4.3	Examples	48
4.4	Evaluation	49
4.5	Acknowledgements	50
5	The type of a semisimple Lie algebra	51
5.1	Identifying a semisimple Lie algebra	51
5.2	Isomorphism of semisimple Lie algebras	56
5.3	Examples	59
5.4	Acknowledgements	60
6	Constructive Ado	63
6.1	Calculating a series of extensions	64
6.2	The extension space	65
6.3	Extending a representation	66
6.4	An effective version of Ado's theorem	70
6.5	Examples, and practical experience	72
7	Practice	75
7.1	Isomorphism testing	75
7.2	Calculations in E_8	79

7.2.1 Preliminaries	79
7.2.2 Centralisers of nilpotent elements in E_8	81
A Manual of ELIAS	91
Index of Terminology	109
Index of Symbols	110
Acknowledgements	111
Samenvatting	112
Curriculum Vitae	113

Chapter 1

Introduction

Lie algebras arise naturally in various areas of mathematics and physics. Examples are: representation theory of Lie groups ([20]) and of algebraic groups ([30]) and the theory of Lie point symmetries of a differential equation ([9], [39]). The topic of the present work is the algorithmic treatment of finite dimensional Lie algebras. These Lie algebras live in an obscure world where they are only known by their multiplication table, that is by a faint shadow. Here we present algorithms for obtaining information about a Lie algebra. These allow us to shed rays of light in this world that make a Lie algebra cast more distinct shadows. In some cases, particularly when the Lie algebra is semisimple and of characteristic 0, this enables us to recognise it. In other cases we have to content ourselves with only partial information.

The work described here is implemented in a package called ELIAS (for Eindhoven Lie Algebra System), that will be a part of GAP4.

In this chapter we will first introduce some basic mathematical concepts. Then in Section 1.2 we will deal with the first step of the process of the algorithmic identification of a Lie algebra: representing it on a computer. In the next section we briefly discuss some complexity issues. Finally in Section 1.4, we will present a survey of algorithms known in the literature.

The chapters 2 to 6 each deal with a particular algorithmic problem. In Chapter 2, this is the calculation of the nilradical. In Chapter 3 we describe how a Cartan subalgebra can be found. This is used in Chapter 4, where algorithms for decomposing a semisimple Lie algebra are given. In Chapter 5 the problem of determining the isomorphism type of a semisimple Lie algebra is discussed. An effective version of Ado's theorem is given in Chapter 6. In Chapter 7, the system ELIAS is applied in two practical problems. Finally in Appendix A there is a manual of ELIAS.

In the chapters 2 to 7 running times of calculations are presented. All computations were performed on a SUN SPARC classic workstation.

1.1 Notation, definitions, and basic theory

In this section we describe the basic theoretical tools that we use. For the proofs we refer to the standard monographs ([29],[32]).

Definition 1.1 A Lie algebra is a vector space over a field F equipped with a bilinear map (multiplication)

$$[\cdot, \cdot] : L \times L \longrightarrow L$$

satisfying

$$(L_1) [x, x] = 0 \text{ for all } x \in L,$$

$$(L_2) [[x, y], z] + [[y, z], x] + [[z, x], y] = 0 \text{ for all } x, y, z \in L.$$

The second condition (L_2) is called the *Jacobi identity*. By applying (L_1) to the element $x + y$ we see that the first condition implies $[x, y] = -[y, x]$. If the characteristic of the field is not 2, then this in turn implies (L_1) .

Example 1.2 Let L be a 3-dimensional vector space over \mathbb{Q} with basis $\{x, y, h\}$ and Lie product described by

$$[x, y] = h, [h, x] = 2x, [h, y] = -2y.$$

By using bilinearity and anticommutativity this defines the Lie product for all elements of L .

Example 1.3 Let L be the 3-dimensional subspace of $M_3(\mathbb{Q})$ spanned by

$$A_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, A_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

For two elements $A, B \in L$ we set $[A, B] = A \cdot B - B \cdot A$ (where the \cdot stands for ordinary matrix multiplication). It is seen that the space L is closed under this operation. Furthermore the $[\cdot, \cdot]$ defined in this way satisfies the two requirements for being a Lie multiplication. It follows that L is a Lie algebra.

Definition 1.4 Let L be a Lie algebra over F and V a vector space over F . A representation of L on V is a linear map

$$\rho : L \longrightarrow \text{End}(V)$$

such that $\rho([x, y]) = \rho(x)\rho(y) - \rho(y)\rho(x)$.

Example 1.5 Let L be the Lie algebra of Example 1.2 and let A_1, A_2, A_3 be as in Example 1.3. Let ρ be the linear map from L into the space spanned by the A_i given by

$$\rho(x) = A_1, \quad \rho(y) = A_2, \quad \rho(h) = A_3.$$

Then it is seen that ρ is a representation of L . Furthermore we have that the kernel of ρ is 0, so that L is isomorphic to its image. Representations with this property are called *faithful*.

Example 1.6 Let L be a Lie algebra. Define a map

$$\text{ad} : L \longrightarrow \text{End}(L)$$

by $\text{ad}(x)(y) = [x, y]$. The fact that this is a Lie algebra representation is equivalent to the Jacobi identity. The map ad is called the *adjoint representation*.

To a representation ρ of L we associate a bilinear form f_ρ defined by $f_\rho(x, y) = \text{Tr}(\rho(x)\rho(y))$. In the case of the adjoint representation this form is called the *Killing form* and is denoted by κ .

Definition 1.7 Let L be a Lie algebra. A subspace K of L is called a *subalgebra* if $[x, y] \in K$ for all $x, y \in K$.

Definition 1.8 Let L be a Lie algebra. A subspace I of L is called an *ideal* if $[x, y] \in I$ for all $x \in L, y \in I$.

Let L be a Lie algebra and let I be an ideal of L such that there is a subalgebra K of L with the property that $L = K \oplus I$ (direct sum of vector spaces). Then L is called the *semidirect product* of K and I . It is denoted by $L = K \ltimes I$. A special case is the situation where K is also an ideal of L . Then L is called the *direct sum* of K and I . In this case we write $L = K \oplus I$.

Definition 1.9 Let L be a Lie algebra. Then the subspace

$$Z(L) = \{x \in L \mid [x, y] = 0 \text{ for all } y \in L\}$$

is called the *centre* of L .

As is easily seen, the centre of a Lie algebra L is an ideal in L . If L is equal to its centre, then L is called *Abelian* or *commutative*.

Definition 1.10 Let L be a Lie algebra and K a subspace of L . Then

$$Z_L(K) = \{x \in L \mid [x, y] = 0 \text{ for all } y \in K\}$$

is called the *centraliser* of K in L .

If K is an ideal of L , then also $Z_L(K)$ will be an ideal of L . This follows from the Jacobi identity.

Definition 1.11 Let K be a subspace of the Lie algebra L . Then

$$N_L(K) = \{x \in L \mid [x, y] \in K \text{ for all } y \in K\}$$

is called the normaliser of K in L .

If K_1 and K_2 are subspaces of L , then $[K_1, K_2]$ will denote the subspace spanned by all $[x_1, x_2]$ for $x_1 \in K_1$ and $x_2 \in K_2$.

Definition 1.12 Let L be a finite-dimensional Lie algebra. Set $L_1 = L$ and recursively $L_{k+1} = [L_k, L_k]$. Let s be the smallest integer such that $L_s = L_{s+1}$. The series

$$L_1 \supset L_2 \supset \cdots \supset L_s$$

is called the derived series of L .

A Lie algebra is called *solvable* if the final term of its derived series is 0. If I and J are solvable ideals of L , then it can be proved that $I + J$ is also a solvable ideal of L . It follows that if L is a finite-dimensional Lie algebra, then it has a maximal solvable ideal. This ideal is called the *solvable radical* of L . It is denoted by $R(L)$.

Definition 1.13 Let L be a finite-dimensional Lie algebra. Set $L^1 = L$ and $L^{k+1} = [L, L^k]$. Let t be the smallest integer such that $L^t = L^{t+1}$. The series

$$L^1 \supset L^2 \supset \cdots \supset L^t$$

is called the lower central series of L .

A finite-dimensional Lie algebra L is called *nilpotent* if $L^t = 0$. If I and J are nilpotent ideals of L , then it can be proved that so is $I + J$. It follows that a finite-dimensional Lie algebra L has a largest nilpotent ideal. It is called the *nilradical* and it is denoted by $NR(L)$.

Definition 1.14 Let L be a finite-dimensional Lie algebra. Set $Z_1 = Z(L)$ and define Z_{k+1} recursively by the relation $Z_{k+1}/Z_k = Z(L/Z_k)$. Let u be the smallest number such that $Z_u = Z_{u+1}$. Then the series

$$Z_1 \subset Z_2 \subset \cdots \subset Z_u$$

is called the upper central series of L .

The final term of the upper central series of L is called the *hypercentre* of L . It is denoted by $Z_\infty(L)$. It can be proved that L is nilpotent if and only if $L = Z_\infty(L)$.

Definition 1.15 Let L be a Lie algebra defined over a field of characteristic $p > 0$. Then L is called *restricted* if the set $\text{ad } L$ is closed under the operation of taking p -th powers.

Restricted Lie algebras admit a richer structure than Lie algebras that do not possess this property. We refer to [32] and [50] for the details.

Definition 1.16 A Lie algebra L is called *semisimple* if $R(L) = 0$.

The next lemma gives a useful criterion for a Lie algebra to be semisimple.

Lemma 1.17 Let L be a Lie algebra with basis $\{x_1, \dots, x_n\}$ and let d be the determinant of the matrix $(\kappa(x_i, x_j))$. If $d \neq 0$, then L is semisimple. If L is defined over a field of characteristic 0, then this in turn implies $d \neq 0$.

Definition 1.18 A Lie algebra L is called *simple* if $\dim L > 1$ and it has no ideals except 0 and L .

Let L be a simple Lie algebra. Then $R(L)$ can only be 0 or L . Suppose $R(L) = L$. Then L is solvable and hence $[L, L]$ is an ideal of L not equal to L . It follows that $[L, L] = 0$ so that L is Abelian. But then every subspace of L is an ideal contradicting the fact that L is simple. The conclusion is that $R(L) = 0$ and L is semisimple.

Semisimple Lie algebras play an important role in the structure theory of non-semisimple Lie algebras. This is due to the following theorem.

Theorem 1.19 (Levi) Let L be a Lie algebra over a field of characteristic 0. If L is not solvable then there exists a (necessarily semisimple) subalgebra S of L such that L is the semidirect product of S and $R(L)$.

The semisimple subalgebra S of the theorem is called a *Levi subalgebra* or *Levi factor* of L .

Now we define a class of subalgebras that is vital for the structure theory of semisimple Lie algebras.

Definition 1.20 A subalgebra H of L is called a *Cartan subalgebra* if H is nilpotent and $N_L(H) = H$.

If the size of the field over which L is defined is larger than its dimension, then L has a Cartan subalgebra ([29], Theorem 15.3).

Example 1.21 Let L be the Lie algebra of Example 1.2. Then the subalgebra spanned by the element h is a Cartan subalgebra.

Let L be a semisimple Lie algebra. Then it turns out to be a fruitful idea to analyse the adjoint action of a Cartan subalgebra on L . The next two results constitute a first step in that direction.

Lemma 1.22 (Fitting) *Let A be a linear transformation of a finite dimensional vector space V . Then V decomposes as*

$$V = V_0(A) \oplus V_1(A),$$

where $V_0(A) = \{v \in V \mid A^m v = 0 \text{ for some } m > 0\}$ and $V_1(A) = \bigcap_{i=1}^{\infty} A^i V$.

The decomposition in the lemma is called the *Fitting decomposition* of V with respect to A . The spaces $V_0(A)$ and $V_1(A)$ are called, respectively, the Fitting null and one component of V relative to A .

A similar decomposition exists with respect to a nilpotent Lie algebra of linear transformations. The next proposition is a transcription of Theorem II.4 of [32].

Proposition 1.23 *Let L be a nilpotent Lie algebra of linear transformations in a finite dimensional vector space V . Then V decomposes as a direct sum of L -invariant subspaces*

$$V = V_0(L) \oplus V_1(L),$$

where $V_0 = \bigcap_{A \in L} V_0(A)$ and $V_1 = \bigcap_{i=1}^{\infty} (L^*)^i V$ (where L^* is the associative subalgebra of $\text{End}(V)$ generated by L).

Let L be a semisimple Lie algebra over an algebraically closed field F of characteristic 0. Let H be a Cartan subalgebra of L . Then via the adjoint representation H acts as a nilpotent Lie algebra of linear transformations on L . Let $L = L_0(H) \oplus L_1(H)$ be the Fitting decomposition of L with respect to H . Now we have $L_0(H) = H$ (Proposition III.1 of [32]). Furthermore $L_1(H)$ decomposes as a direct sum of *simultaneous* eigenspaces relative to the action of H . This means that there are functionals $\alpha_i : H \rightarrow F$ such that

$$L = L_{\alpha_1} \oplus \cdots \oplus L_{\alpha_i} \oplus H \tag{1.1}$$

where $L_{\alpha_i} = \{x \in L \mid [h, x] = \alpha_i(h)x \text{ for all } h \in H\}$. The decomposition (1.1) is called the *Cartan decomposition* of L with respect to H .

It can be proved that the spaces L_{α_i} are all 1-dimensional. They are called *root spaces* and the α_i are called *roots*. Let R be the set of all roots. Then R is a root system (see Chapter 3 of [29]) in the dual space H^* of H . To a root system corresponds a Cartan matrix. Now all Cartan matrices have been classified. It follows that there is also a classification of all semisimple Lie algebras over algebraically closed fields of characteristic 0.

Example 1.24 Let L be the Lie algebra of Example 1.2. Then $H = \langle h \rangle$ is a Cartan subalgebra of L . It is seen that $\text{ad } h$ has all its eigenvalues in \mathbb{Q} so that L already has a Cartan decomposition over this field. This decomposition is

$$L = L_2 \oplus L_{-2} \oplus H.$$

There are two roots, $\alpha_1 = 2$ and $\alpha_2 = -2$ in the 1-dimensional space H^* . Furthermore, $L_2 = \langle x \rangle$ and $L_{-2} = \langle y \rangle$.

1.2 Presentations of Lie algebras

In this section we turn our attention to the problem of representing a Lie algebra on a computer. We refer to [10] for a more elaborate discussion of this topic.

In Examples 1.2 and 1.3 we already encountered two ways of handling the problem. First of all we can present the Lie algebra by a set of matrices that form a vector space basis of the Lie algebra. If A and B are two elements of this space, then their Lie product is defined as $[A, B] = A \cdot B - B \cdot A$.

The second approach is used in Example 1.2. Now the Lie algebra is viewed as a (abstract) vector space with basis $\{x_1, \dots, x_n\}$. The multiplication is described by a table of n^3 structure constants c_{ij}^k such that

$$[x_i, x_j] = \sum_{k=1}^n c_{ij}^k x_k \quad \text{for } 1 \leq i, j \leq n.$$

By bilinearity this defines the product for any two elements of L . In order that this be a Lie bracket, the structure constants have to satisfy the following relations:

$$c_{ii}^k = c_{ij}^k + c_{ji}^k = 0,$$

$$\sum_{s=1}^n c_{jk}^s c_{is}^m + c_{ki}^s c_{js}^m + c_{ij}^s c_{ks}^m = 0,$$

for $1 \leq i, j, k, m \leq n$.

The third way to present a Lie algebra on a computer is by generators and relations. Let A be a finite alphabet. Then $L(A)$ will denote the free Lie algebra on the alphabet A (see [42]). Let R be a finite subset of $L(A)$ generating an ideal I . Then $L = \langle A \mid R \rangle$ is the Lie algebra with generators A and relations R . It is defined as the quotient $L(A)/I$.

Example 1.25 Set $A = \{x, y\}$ and $R = \{[[x, y], x] - 2x, [[x, y], y] + 2y\}$. Then it is seen that the Lie algebra $L = \langle A \mid R \rangle$ is isomorphic to the Lie algebra of Example 1.2.

We can also represent subalgebras and ideals. In the first two cases this is done by specifying a basis of the subalgebra or ideal. In the case of generators and relations we can give a subset of $L(A)$ that generates the subalgebra or ideal.

We consider the possible transitions between the various representations. If L is given by matrices, then it is an easy task of linear algebra to calculate the structure constants and

obtain a presentation by means of a table. It is also not difficult to go from a table to a presentation by generators and relations. If $\{x_1, \dots, x_n\}$ is a basis of L , then the alphabet A will consist of the symbols x_1, \dots, x_n . The set R will simply consist of all relations

$$[x_i, x_j] - \sum_{k=1}^n c_{ij}^k x_k \text{ for } 1 \leq i < j \leq n.$$

The other transitions are more difficult. It is only possible to obtain a multiplication table from a presentation by generators and relations if the resulting quotient Lie algebra is finite-dimensional. In that case there are Todd-Coxeter techniques that find a basis and the multiplication table (see [35]). Also it is possible to use a kind of Gröbner basis to solve the problem. This direction was pursued in [21]. The remaining transition (from multiplication table to matrix presentation) is considered in Chapter 6.

Concerning the input to our algorithms, we shall always assume that the Lie algebras are given by a table of structure constants. The reason for this is that many algorithms have to know the structure constants anyway (see e.g., Section 1.4). So it is better to input them than to calculate them every time anew. Subalgebras and ideals will always be represented by a basis (that is by a set of coefficient vectors) of the corresponding subspace of the parent Lie algebra.

1.3 Complexity

Here we discuss some theoretical notions regarding the efficiency of algorithms. For a more thorough treatment and an extensive bibliography regarding this subject we refer to [49]. Polynomiality is a widely accepted theoretical model for efficiency. An algorithm is said to run in *polynomial time* if the number of basic steps taken by the algorithm (on any input) is bounded above by a polynomial in the size of the input. The size of the input is the length of the string needed to represent the input. So the size of a natural number is its number of digits. In general the size of composite objects (vectors, polynomials etc.) is the sum of the sizes of the components.

For the basic arithmetical operations (multiplying and adding numbers or elements of a finite field) there are polynomial time algorithms. Also we can solve systems of linear equations in polynomial time (Gaussian elimination).

We also consider randomisation. A randomised algorithm is an algorithm that uses random choices at certain points (i.e., it can flip a coin and the outcome determines the path taken in the rest of the algorithm). An important class of randomised algorithms are the so-called *Las Vegas* algorithms that never output a wrong result. An algorithm for computing a function $f(x)$ is called Las Vegas if on input a it either computes $f(a)$ correctly with probability $p > 0$, or stops without producing output. It is also required that calls to a Las Vegas algorithm produce independent results. Hence if a Las Vegas algorithm is repeated then it always produces a correct answer. The expected number of repetitions is $1/p$.

An important problem is the one of factoring polynomials. For a detailed exposé we refer to [36]. Let f be an element of $F[X]$ of degree n . First we suppose that $F = \mathbb{F}_q$ where $q = p^s$ for some $s > 0$ and a prime p . A deterministic method for factoring f was given by Berlekamp ([3]). However, the complexity of this method is a polynomial in p , s and n , whereas the input length is $O(n \log q)$ so that this is not a polynomial time algorithm. A polynomial time Las Vegas method was proposed in [7]. Via Hensel lifting the factorisation methods over finite fields can be used for polynomials over \mathbb{Q} as well. A polynomial time solution to the problem of factoring polynomials over \mathbb{Q} was given in [37].

So for factoring polynomials, more than one algorithm is available. For this reason an algorithm using an oracle to factor polynomials will be called an *f-algorithm* (following [44]). The cost of a call to the factoring oracle is the length of the input. The complexity of such f-algorithms then depends on the complexity of the particular oracle used.

1.4 Earlier work

In this section we give a survey of the algorithms for determining the structure of a Lie algebra that are known from the literature. We only describe those algorithms that will not be discussed in one of the other chapters. The main references are [2] and [41]. Throughout L will be a Lie algebra over the field F with basis $\{x_1, \dots, x_n\}$ and structure constants (c_{ij}^k) .

1.4.1 Product spaces

Let K_1 be a subspace of L spanned by $\{y_1, \dots, y_s\}$ and let K_2 be a subspace spanned by $\{z_1, \dots, z_t\}$. Now the product space will be spanned by the elements $[y_i, z_j]$. So in order to find a basis of $[K_1, K_2]$ we have to calculate a maximally linearly independent subset of the set of all $[y_i, z_j]$. This can be done by a Gaussian elimination procedure. Note that this also gives an algorithm for calculating the derived series and the lower central series of L .

1.4.2 The centre

Let

$$x = \sum_{i=1}^n \alpha_i x_i$$

be an element of L . Then x is an element of $Z(L)$ if and only if

$$\sum_{i=1}^n c_{ij}^k \alpha_i = 0 \quad \text{for all } 1 \leq j, k \leq n.$$

So we have n^2 equations for the n unknowns $\alpha_1, \dots, \alpha_n$, which can be solved by Gaussian elimination. This also provides an algorithm for calculating the upper central series of L .

1.4.3 The centraliser

Let $\{y_1, \dots, y_s\}$ be a basis of a subspace K of L , where

$$y_l = \sum_{j=1}^n \lambda_{lj} x_j.$$

Then $x = \sum_i \alpha_i x_i$ lies in $Z_L(K)$ if and only if

$$\sum_{i=1}^n \left(\sum_{j=1}^n \lambda_{lj} c_{ij}^k \right) \alpha_i = 0 \quad \text{for } 1 \leq k \leq n \text{ and } 1 \leq l \leq s.$$

It follows that we have ns equations for the n unknown $\alpha_1, \dots, \alpha_n$.

1.4.4 The normaliser

Let K and y_1, \dots, y_s be the same as in the previous section. Then $x = \sum_i \alpha_i x_i$ is an element of $N_L(K)$ if and only if there are β_{lm} for $1 \leq l, m \leq s$ such that

$$[x, y_l] = \beta_{l1} y_1 + \dots + \beta_{ls} y_s \quad \text{for } l = 1, \dots, s.$$

This amounts to the following linear equations in the variables α_i and β_{lm} :

$$\sum_{i=1}^n \left(\sum_{j=1}^n \lambda_{lj} c_{ij}^k \right) \alpha_i - \sum_{m=1}^s \lambda_{mk} \beta_{lm} = 0 \quad \text{for } 1 \leq k \leq n \text{ and } 1 \leq l \leq s.$$

1.4.5 The solvable radical

Suppose L is defined over a field of characteristic 0. Then there is a simple algorithm for finding $R(L)$. It relies on the following lemma:

Lemma 1.26 $R(L) = \{x \in L \mid \kappa(x, y) = 0 \text{ for all } y \in [L, L]\}$.

This is Theorem III.5 of [32]. If $\{y_1, \dots, y_s\}$ is a basis of $[L, L]$ then $x = \sum_i \alpha_i x_i$ is an element of $R(L)$ if and only if

$$\sum_{i=1}^n \text{Tr}(\text{ad } x_i \cdot \text{ad } y_j) \alpha_i = 0 \quad \text{for } 1 \leq j \leq s.$$

In the case where L is defined over a field of characteristic $p > 0$ the situation is much more difficult. In [44], L. Rónyai gives an algorithm for calculating the nilradical of a Lie algebra over a field of characteristic $p > 0$ (see also Chapter 2). Now we define a series R_k by

$$R_1 = \text{NR}(L), \quad R_{k+1}/R_k = \text{NR}(L/R_k).$$

Let u be the integer such that $R_u = R_{u+1}$. Then $R_u = \text{NR}(L)$.

1.4.6 The direct sum decomposition

A Lie algebra L may be the direct sum of two ideals I_1 and I_2 . In this section we consider the problem of deciding whether such a decomposition exists and if so, to find the change of basis that realises the decomposition. The following is a reformulation of Section 2 of [41]. For the proofs that we omit we refer to that paper.

Suppose that $L = I_1 \oplus I_2$ and I_1 is contained in the centre of L . Then I_1 is called a *central component* of L . First we give a method for finding such a central component if it exists.

Let J_1 be a complementary subspace in $Z(L)$ to $Z(L) \cap [L, L]$. Then J_1 is an ideal of L . Let J_2 be the complementary subspace in L to J_1 containing $[L, L]$. Then

$$[L, J_2] \subset [L, L] \subset J_2$$

so that J_2 is an ideal of L . Furthermore $L = J_1 \oplus J_2$ and J_1 is central and J_2 does not contain a central component. The conclusion is that J_1 is a maximal central component.

Now we suppose that $Z(L) \subset [L, L]$ and we try to decompose L as a direct sum of ideals.

By R_n we will denote the associative matrix algebra $M_n(F)$ where n is the dimension of L . An element $E \in R_n$ is called an *idempotent* if $E^2 = E$. Two idempotents E_1 and E_2 are called *orthogonal* if $E_1 E_2 = 0$. An idempotent is called *primitive* if it is not the sum of two other idempotents. Furthermore, if E is not equal to the zero or the identity matrix, then E is called a *nontrivial idempotent*.

Proposition 1.27 *The Lie algebra L is the direct sum of two ideals I_1 and I_2 if and only if the centraliser $Z_{R_n}(\text{ad } L)$ contains two nontrivial orthogonal idempotents E_1, E_2 such that $E_1 + E_2$ is the identity on L and $I_k = E_k I_k$ for $k = 1, 2$.*

If A is an associative algebra then its *radical* is defined as the set of all elements x such that xy is nilpotent for all $y \in A$. It is denoted by $\text{Rad}(A)$. In characteristic 0 it is easy to calculate the radical because in that case we have that the radical of an associative algebra A is given by

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

(see [13]). In the sequel A will denote the centraliser $Z_{R_n}(\text{ad } L)$ and $Q = A/\text{Rad}(A)$.

Theorem 1.28 *Suppose that $Z(L) \subset [L, L]$. Then Q is Abelian.*

By the following propositions it is sufficient to find idempotents in the factor algebra Q .

Proposition 1.29 *Every direct summand of L corresponds exactly to an idempotent of Q .*

Lemma 1.30 *Let e be an idempotent in Q . Then we can construct an idempotent $E \in A$ such that $E \equiv e \pmod{\text{Rad}(A)}$.*

Proof. (cf. Satz 1 of [51].) Let $E_0 \in A$ be such that $E_0 \equiv e \pmod{\text{Rad}(A)}$. Then $E_0^2 \equiv E_0 \pmod{\text{Rad}(A)}$ so that $N_0 = E_0^2 - E_0 \in \text{Rad}(A)$ and consequently it is nilpotent. Hence there is an integer q such that $N_0^{2^q} = 0$. Now recursively set

$$E_{i+1} = E_i + N_i - 2E_i N_i,$$

$$N_{i+1} = E_{i+1} E_{i+1} - E_{i+1}.$$

By induction on i it follows that $N_i \equiv 0$ and $E_{i+1} \equiv E_i$ modulo the module generated by the elements

$$N_0^{2^i}, N_0^{2^{i+1}}, \dots, E_0 N_0^{2^i}, E_0 N_0^{2^{i+1}}, \dots.$$

We conclude that $N_q = 0$ and $E_q E_q = E_q$ and the statement follows. \square

Proposition 1.31 *Let e_1, \dots, e_s be primitive orthogonal idempotents in Q . Then we can construct primitive orthogonal idempotents E_1, \dots, E_s in A such that*

$$E_i \equiv e_i \pmod{\text{Rad}(A)} \text{ for } 1 \leq i \leq s.$$

Proof. (cf. Satz 1 of [51].) For brevity we set $R = \text{Rad}(A)$. The proof is by induction on s . The case where $s = 1$ is covered by Lemma 1.30. Suppose that $s > 1$ and that we have constructed primitive orthogonal idempotents E_1, \dots, E_{s-1} in A satisfying the equivalences $E_i \equiv e_i \pmod{R}$ for $1 \leq i \leq s-1$. We describe how to construct E_s . Set

$$E = E_1 + \dots + E_{s-1},$$

and

$$e = e_s - E e_s - e_s E + E e_s E.$$

Then $E e_s \equiv e_s E \equiv 0 \pmod{R}$ and hence $e \equiv e_s \pmod{R}$ and $e^2 \equiv e \pmod{R}$. Now let E_s be the idempotent in A provided by the procedure in the proof of Lemma 1.30 (where we start with $E_0 = e$). So $E_s \equiv e_s \pmod{R}$. Since $E^2 = E$ we have that $E e = e E = 0$, and because E_s is a polynomial in e it follows that $E E_s = E_s E = 0$. By the induction

hypothesis we have $E_i E = E E_i = E_i$ for $1 \leq i \leq s-1$. Hence $E_s E_i = E_s E E_i = 0$ and similarly $E_i E_s = 0$. \square

We assume here that L does not have a central component which means that $Z(L) \subset [L, L]$. Hence by Theorem 1.28, we have that the factor algebra $Q = A/\text{Rad}(A)$ is commutative. So we can use algorithms described in [16] to find a set of primitive orthogonal idempotents in Q . Using the procedure in the proof of Proposition 1.31 we lift these idempotents to A . The direct summands of L are then obtained as in Proposition 1.27.

Remark. A practical evaluation of this algorithm is given in Section 4.4. It turns out that the calculation of the centraliser $Z_{R_n}(\text{ad } L)$ makes this procedure computationally difficult.

1.4.7 The Levi decomposition

Here we consider the problem of finding a Levi subalgebra of L . We follow [24]. In the sequel R will denote the solvable radical of L and by R^i we denote the ideal

$$R^i = [R, [R, \dots, [R, R] \dots]] \text{ (} m \text{ factors } R\text{)}.$$

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

Lemma 1.32 *Let S_1 be the inverse image in L of a Levi subalgebra of L/R^2 . Let S be a Levi subalgebra of S_1 , then S is a Levi subalgebra of L .*

Proof. (cf. [32] Section III.9) It is clear that S is a semisimple 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 subalgebra as required. \square

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 (the calculation of inverse images poses no problems). Now suppose that the solvable radical R of L is nilpotent. Let

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

be the lower central series of R .

Let $\{u_1, \dots, u_s\}$ be a maximal linearly independent set in the complement of R . Let (γ_{ij}^k) be the structure constants of the quotient L/R , i.e., $[\bar{u}_i, \bar{u}_j] = \sum_k \gamma_{ij}^k \bar{u}_k$, where \bar{u}_i is the

image of u_i in L/R for $1 \leq i \leq s$. Then we have that:

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

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

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

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

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

This is equivalent to

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

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

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

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 [24].

1.4.8 Zeros of polynomials

We finish this chapter with some observations about zeros of polynomials that will be useful later on.

For the proof of the following lemma we refer to [45].

Lemma 1.33 *Let $f \in F[x_1, \dots, x_n]$ be a polynomial of degree d . Let Ω be a subset of F of size N . Then the number of elements $v = (v_1, \dots, v_n)$ of Ω^n such that $f(v) = 0$ is at most dN^{n-1} .*

Corollary 1.34 *Let f and Ω be the same as in the preceding lemma. Let v be an element from Ω^n chosen randomly and uniformly. Then the probability that $f(v) = 0$ is at most d/N .*

Lemma 1.35 *Let $f \in F[x_1, \dots, x_n]$ be a polynomial of degree d . Let Ω be a subset of F of size at least $d + 1$. Suppose that we are given a vector $v = (\alpha_1, \dots, \alpha_n) \in F^n$ such that $f(v) \neq 0$. Then we can find a vector $w = (\xi_1, \dots, \xi_n) \in \Omega^n$ such that $f(w) \neq 0$ at the expense of at most $n(d + 1)$ tests whether $f(u)$ is zero for vectors $u \in (\Omega \cup \{\alpha_1, \dots, \alpha_n\})^n$.*

Proof. We construct a sequence w_0, w_1, \dots, w_n of vectors such that $f(w_i) \neq 0$ and $w_i = (\xi_1, \dots, \xi_i, \alpha_{i+1}, \dots, \alpha_n)$. For the initialisation we set $w_0 = v$. In the following way we obtain w_i from w_{i-1} . Consider the polynomial $g_i(x) = f(\xi_1, \dots, \xi_{i-1}, x, \alpha_{i+1}, \dots, \alpha_n)$. This polynomial is not identically zero, because $g_i(\alpha_i) = f(v_{i-1}) \neq 0$. Since $|\Omega| > d \geq \deg g_i$, we have that Ω contains an element ξ_i such that $g_i(\xi_i) \neq 0$. We can find such a ξ_i by trying at most $d + 1$ values. Since the procedure takes n steps the statement follows. \square

Chapter 2

The nilradical

The nilradical of a Lie algebra L reveals some of the properties of L . If the nilradical is 0 then so is the solvable radical and hence L is semisimple. On the other hand, if the solvable radical is not 0, then it contains the nilradical as an important invariant.

Throughout this chapter L will be a finite-dimensional Lie algebra of characteristic 0. However, we will also comment on the situation in characteristic $p > 0$.

In Section 2.1 we describe three previously published algorithms. Then in Section 2.2 we give a different algorithm. Finally in Section 2.3 the methods are put to some practical tests.

2.1 Known algorithms

In the past decades a few algorithms for calculating the nilradical have been given. Here we present a survey of these methods.

To the best of our knowledge, the first solution to the problem was described in [2]. It consists of the following steps:

1. Calculate the solvable radical R of the Lie algebra L .
2. Construct a series $0 = R_0 \subset R_1 \subset \cdots \subset R_m = R$ of ideals of R such that $\dim R_i = i$.
3. The nilradical is the set of all $x \in R$ such that $[x, R_i] \subset R_{i-1}$ for $1 \leq i \leq m$.

The main disadvantage of this algorithm lies in the fact that in general it requires the calculation of algebraic numbers that may not lie in the base field. (The series constructed in step 2. may not exist over the base field.)

In their paper [41], Rand, Winternitz, and Zassenhaus describe an algorithm that finds the nilradical by approximating it from below by smaller nilpotent ideals. Therefore we call it the *upward method*. Here we give a slightly simplified version of that algorithm. It relies on a series of lemmas. For the proof of the first four of them we refer to [41].

Lemma 2.1 *Let I be an ideal of L and let M be the ideal of L containing I and satisfying $\text{NR}(L/I) = M/I$. Then $\text{NR}(L) = \text{NR}(M)$.*

Lemma 2.2 *Set $I = [[L, L], [L, L]]$. Then*

$$\text{NR}(L)/I = \text{NR}(L/I).$$

Lemma 2.3 *We have*

$$\text{NR}(L)/Z(L) = \text{NR}(L/Z(L)).$$

Lemma 2.4 *Suppose $[[L, L], [L, L]] = 0$ and $Z(L) = 0$. Then*

$$Z_L([L, L]) = [L, L].$$

Lemma 2.5 *Suppose $[[L, L], [L, L]] = 0$ and $Z(L) = 0$. Then $[L, [L, L]] = [L, L]$.*

Proof. Let y_1, \dots, y_s be a basis of a complement of $[L, L]$ in L . Set $V = [L, L]$ and let U be the subspace of $\text{End}(V)$ spanned by $\text{ad}_V y_i$ for $1 \leq i \leq s$. Let $v \in V$; then by the Jacobi identity we have

$$\begin{aligned} \text{ad } y_i \text{ ad } y_j(v) = [y_i, [y_j, v]] &= -[y_j, [v, y_i]] - [v, [y_i, y_j]] \\ &= [y_j, [y_i, v]] \\ &= \text{ad } y_j \text{ ad } y_i(v). \end{aligned}$$

It follows that U is a commutative Lie algebra of linear transformations in V . Now let $V = V_0(U) \oplus V_1(U)$ be the Fitting decomposition of V with respect to U (Proposition 1.23). Suppose $V_0(U) \neq 0$, then by Theorem 3.3 in [29], p. 12, there is a nonzero $v \in V_0(U)$ killed by U . Since $[L, L]$ is commutative it follows that $v \in Z(L)$. Hence $V_0(U) = 0$ and $V = V_1(U)$. We recall that U^* is the associative algebra generated by U (inside $\text{End}(V)$) and that $V_1(U) = \bigcap_{i=1}^{\infty} (U^*)^i V$. So $U \cdot V_1 = V_1$, i.e., $[L, [L, L]] = [L, L]$. \square

Now the algorithm reads as follows:

Input: a finite-dimensional Lie algebra L of characteristic 0.

Output: $\text{NR}(L)$.

Step 1 Compute the solvable radical $R = \text{R}(L)$. If $R = 0$ then return R , otherwise continue with R in place of L .

- Step 2 Compute the ideal $I = [[L, L], [L, L]]$. If $I \neq 0$ then compute (by a recursive call) the nilradical \bar{N} of L/I and return the inverse image of N in L . Otherwise proceed to Step 3.
- Step 3 Compute the hypercentre $Z_\infty(L)$ of L and proceed as in Step 2 where $Z_\infty(L)$ plays the role of I .
- Step 4 Compute a basis of L of the form $\{x_1, \dots, x_s, y_1, \dots, y_t\}$ where $[L, L]$ is spanned by $\{x_1, \dots, x_s\}$. Set $i := 1$;
- Step 5 Let A be the matrix of the adjoint action of y_i on $[L, L]$. If $\text{rk}(A) < s$, then compute the ideal $J = A \cdot [L, L]$. Compute recursively the nilradical of L/J and let M be the ideal of L containing J such that $\text{NR}(L/J) = M/J$. Compute (by a recursive call) $\text{NR}(M)$ and return this ideal.
- Step 6 Let f be the minimum polynomial of A . If f is not squarefree then set $g = f/\text{gcd}(f, f')$. Compute the ideal $I = g(A) \cdot [L, L]$ and proceed as in Step 5. If f is squarefree proceed to Step 7.
- Step 7 If $i < t$ then set i equal to $i + 1$ and go to Step 5. Otherwise, $\text{NR}(L) = [L, L]$.

Comments:

- Step 1 Since $\text{NR}(R) = \text{NR}(L)$, (see [6], Corollaire 7, p. 67) we may replace L by R .
- Step 2 This step is justified by Lemma 2.2.
- Step 3 This step is justified by Lemma 2.3.
- Step 5 The rank of A is not 0 by Lemma 2.4 (the conditions of this lemma are fulfilled by Steps 2 and 3). If it is less than s , then $J = A \cdot [L, L]$ will be an ideal of L properly contained in $[L, L]$. Hence Lemma 2.5 (ensuring that L/J is not nilpotent, and hence $M \neq L$) and Lemma 2.1 justify the recursive calls.
- Step 6 For $z, a, b \in L$ we have

$$[z, (\text{ad } y_i)^m([a, b])] = (\text{ad } y_i)^m([z, [a, b]])$$

which is proved by induction on m . From this it follows that $h(A) \cdot [L, L]$ is an ideal of L for every polynomial h . In particular $g(A) \cdot [L, L]$ is an ideal of L and it is properly contained in $[L, L]$ because $g(A)$ is nilpotent.

- Step 7 If $i = t$ then all elements y_k act by a semisimple matrix on $[L, L]$. Furthermore these matrices commute. So any nilpotent element of L is contained in the span of x_1, \dots, x_s . It follows that $\text{NR}(L)$ is contained in $[L, L]$. By Step 1 we have that L is solvable so that $\text{NR}(L) = [L, L]$.

Remark. In [41], the following statement is used in place of Lemma 2.5.

Let L be a finite-dimensional Lie algebra with a solvable ideal A and an ideal B contained in $[A, A]$. Then L is nilpotent if and only if L/B is nilpotent.

Unfortunately this statement is false. Take the following counterexample. Let L be an $n + 2$ dimensional Lie algebra with basis $\{x_1, \dots, x_{n+2}\}$ and with commutation relations $[x_1, x_2] = x_{n+2}$, $[x_1, x_i] = x_i$ for $3 \leq i \leq n + 1$ (the other brackets of basis elements are 0). Set $A = L$ and let B be the ideal of L spanned by $\{x_3, \dots, x_{n+1}\}$. Then B is properly contained in $[A, A]$ and L/B is nilpotent, but L is not.

Finally, L. Rónyai ([44]) proposed an algorithm that uses the radical of an associative algebra. As was remarked in Section 1.4.6, the radical of a finite dimensional associative algebra over a field of characteristic 0 is easily calculated. By the following theorem this leads to an algorithm for determining the nilradical.

Theorem 2.6 *Let L be a finite-dimensional Lie algebra and let $(\text{ad } L)^*$ be the associative algebra generated by $\text{ad } L$ inside $\text{End}(L)$. Then an element $x \in L$ lies in $\text{NR}(L)$ if and only if $\text{ad } x$ is contained in $\text{Rad}((\text{ad } L)^*)$.*

For the proof we refer to [32], p. 36.

A major disadvantage of the last algorithm is the fact that the dimension of the associative algebra $(\text{ad } L)^*$ may be substantially bigger than the dimension of L . In the next section we will present an algorithm that does not suffer from this problem. The algorithm does not require the calculation of algebraic numbers. It also does not use recursion.

2.2 The Downward Method

Here we describe an algorithm that approximates the nilradical from above by other ideals. Therefore we call it the *Downward Method*. In the sequel there will appear many ideals. If we speak of a map $\text{ad } x$, we always mean the map $\text{ad}_L x : L \rightarrow L$.

Set $I_0 = \{x \in L \mid \text{Tr}(\text{ad } x) = 0\}$. Now for $k \geq 0$ we define subspaces I_k of L in the following way:

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

Then $L \supseteq I_0 \supseteq I_1 \supseteq \dots$. The next theorem states some useful properties of this series of subspaces.

Theorem 2.7 *For the sequence I_0, I_1, \dots defined above we have the following:*

1. I_k is an ideal of L for $k \geq 0$.

2. $NR(L)$ is contained in I_k for $k \geq 0$.
3. If $n = \dim L$ then $I_{n-2} = NR(L)$.

Proof.

1. Choose $x \in I_k$ and $y \in L$. We have to prove that $[x, y] \in I_k$.

First we derive a useful relation. For y_1, \dots, y_k in L and $1 \leq t \leq k$ we have that

$$\begin{aligned} \text{Tr}(\text{ad } y_1 \cdots \text{ad } y_{t-1} \text{ad } y \text{ad } y_t \cdots \text{ad } y_k \text{ad } x) &= \\ \text{Tr}(\text{ad } y_1 \cdots \text{ad } y_t \text{ad } y \text{ad } y_{t+1} \cdots \text{ad } y_k \text{ad } x) & \end{aligned} \quad (2.1)$$

This identity holds because the difference of these two traces equals

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

and the latter equals 0 as $x \in I_k$. Now we have

$$\begin{aligned} \text{Tr}(\text{ad } y_1 \cdots \text{ad } y_k \text{ad } x \text{ad } y) &= \text{Tr}(\text{ad } y \text{ad } y_1 \cdots \text{ad } y_k \text{ad } x) \\ &= \text{Tr}(\text{ad } y_1 \text{ad } y \text{ad } y_2 \cdots \text{ad } y_k \text{ad } x) \\ &= \dots = \text{Tr}(\text{ad } y_1 \cdots \text{ad } y_k \text{ad } y \text{ad } x). \end{aligned}$$

The first equality follows from $\text{Tr}(AB) = \text{Tr}(BA)$. The other equalities follow from (2.1). Subtracting the right hand side from the left hand side, we get

$$\text{Tr}(\text{ad } y_1 \cdots \text{ad } y_k \text{ad}[x, y]) = 0$$

for all y_1, \dots, y_k in L , i.e., $[x, y] \in I_k$.

2. 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 $NR(L)$. Then by Theorem 2.6 it follows that $\text{ad } x$ lies in $\text{Rad}((\text{ad } L)^*)$. Hence, by definition of the radical of an associative algebra 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 $k \geq 0$.
3. Because $NR(L)$ is contained in I_{n-2} , we only have to prove that I_{n-2} is nilpotent. By Engel's theorem ([32], p. 36), I_{n-2} is nilpotent if and only if $\text{ad}_{I_{n-2}} x$ is nilpotent for all $x \in I_{n-2}$.

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

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

Let f be the polynomial $\prod_i (X - \lambda_i)$ and write

$$f = X^{n-1} + a_1 X^{n-2} + \dots + a_{n-1}.$$

Now we recall Newton's identities (see [33], p. 287):

$$\begin{aligned} s_1 + a_1 &= 0 \\ s_2 + a_1 s_1 + 2a_2 &= 0 \\ &\vdots \\ s_{n-1} + a_1 s_{n-2} + \dots + a_{n-2} s_1 + (n-1)a_{n-1} &= 0. \end{aligned}$$

If $k = n - 2$, then $s_1 = s_2 = \dots = s_{n-1} = 0$. From this it easily follows that all λ_i must be 0 so that $\text{ad}_L x$ is nilpotent. From this it follows that also $\text{ad}_{J_{n-2}} x$ is nilpotent and we are done.

□

On the basis of Theorem 2.7 we can formulate an algorithm. Theorem 2.7, implies that it is correct and that it terminates after at most $n - 2$ steps.

Algorithm NilRadical

Input: A Lie algebra L of characteristic 0.

Output: $\text{NR}(L)$.

Step 1 $k := 0$; $I := I_0$;

Step 2 **if** I is nilpotent **then return** I ; **fi**;

Step 3 $k := k + 1$; $I := I_k$; go to Step 2;

We consider the calculation of a basis of I_k . First it is easily seen that a basis of I_0 can be computed by solving a system of linear equations. Let $k \geq 1$ and let $x \in I_{k-1}$. Then we have the following

$$\begin{aligned} \text{Tr}(\text{ad } y_1 \cdots \text{ad } y_i \text{ ad } y_{i+1} \cdots \text{ad } y_k \text{ ad } x) &= \\ \text{Tr}(\text{ad } y_1 \cdots \text{ad } y_{i+1} \text{ ad } y_i \cdots \text{ad } y_k \text{ ad } x) + \text{Tr}(\text{ad } y_1 \cdots \text{ad } [y_i, y_{i+1}] \cdots \text{ad } y_k \text{ ad } x) &= \\ \text{Tr}(\text{ad } y_1 \cdots \text{ad } y_{i+1} \text{ ad } y_i \cdots \text{ad } y_k \text{ ad } x). & \end{aligned}$$

Hence, by linearity of the trace, we have that

$$I_k = \{x \in I_{k-1} \mid \text{Tr}(\text{ad } x_{i_1} \cdots \text{ad } x_{i_k} \text{ ad } x) = 0 \text{ for } 1 \leq i_1 \leq i_2 \leq \cdots \leq i_k \leq n\},$$

where $\{x_1, \dots, x_n\}$ is a basis of L . Let $\{z_1, \dots, z_s\}$ be a basis of I_{k-1} . If $x \in I_{k-1}$, then $x = \sum \lambda_j z_j$. If x is also an element of I_k , then for every k -tuple $(x_{i_1}, \dots, x_{i_k})$ where $i_1 \leq i_2 \leq \cdots \leq i_k$, we must have that $\text{Tr}(\text{ad } x_{i_1} \cdots \text{ad } x_{i_k} \text{ ad } x) = 0$. This last condition amounts to the following linear equation in the variables λ_j

$$\sum_{j=1}^s \text{Tr}(\text{ad } x_{i_1} \cdots \text{ad } x_{i_k} \text{ ad } z_j) \lambda_j = 0.$$

It follows that a basis of I_k can be computed by solving a system of linear equations.

In the equation system for I_k , the number of “words” $\text{ad } x_{i_1} \cdots \text{ad } x_{i_k}$ that have to be taken into consideration is bounded by $(\dim L)^2$, because the dimension of $(\text{ad } L)^*$ is bounded by that number. So the number of steps taken by the algorithm is polynomial in the input size. Concerning the sizes of the intermediate results and of the output we use the following formula:

$$I_k = \{x \in L \mid \text{Tr}(\text{ad } x_{i_1} \cdots \text{ad } x_{i_m} \text{ ad } x) = 0 \text{ for } 0 \leq m \leq k\},$$

where x_1, \dots, x_n is a basis of L . For every basis element x_i , the entries of the matrix $\text{ad } x_i$ are elements from the multiplication table. So in each step we have to solve a homogeneous system of linear equations where the entries of the matrix of the equation system are polynomials in the constants from the multiplication table. Hence the sizes of the intermediate results and of the output are polynomial in the input size.

We would like to stress that in most cases the nilradical will be equal to I_k with $k \ll n - 2$. First we may suppose that the Lie algebra L is solvable, because $\text{NR}(L)$ is equal to the nilradical of the solvable radical of L and this solvable radical is easily calculated (see Section 1.4.5). Now, if L is a solvable Lie algebra over an algebraically closed field, then by Lie’s theorem (see [32], p. 50) there exists a basis of L such that the matrices of $\text{ad } x$ for all $x \in L$ are in upper triangular form. If such a basis already exists over the rational numbers, then by the next proposition, we have that the nilradical of L will be equal to I_1 .

Proposition 2.8 *Let L be a solvable Lie algebra over the field \mathbb{Q} of rational numbers. Suppose that L “splits” over \mathbb{Q} i.e., there is a basis of L such that the matrices of $\text{ad } x$ for all $x \in L$ are in upper triangular form. Then $\text{NR}(L) = I_1$.*

Proof. If x is an element of I_1 , then $\text{Tr}((\text{ad } x)^2) = 0$. But this number is the sum of the squares of the eigenvalues of $\text{ad } x$. So it can only be 0 if all eigenvalues of $\text{ad } x$ are 0. The conclusion is that I_1 is a nilpotent ideal. \square

However, there are Lie algebras for which this does not hold, as the next example shows.

Example 2.9 Set $\zeta = e^{2\pi i/q}$ and let L be the Lie algebra with basis $\{x_0, \dots, x_q\}$ and commutation relations

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

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

The next proposition expresses the fact that this is more or less the only example where we have to calculate the I_k up to $k = n - 2$.

Proposition 2.10 *Let L be a finite-dimensional solvable Lie algebra over an algebraically closed field of characteristic 0. Suppose $\dim L = n + 1$ and $\text{NR}(L) = I_{n-1}$ but $\text{NR}(L) \neq I_{n-2}$. Then we have that $\dim \text{NR}(L) = n$ and there is an element y of the complement such that the characteristic polynomial of the restriction of $\text{ad } y$ to $\text{NR}(L)$ is $X^n - 1$. Moreover, if 6 does not divide n , then this determines L upto isomorphism.*

Proof. Let x be a nonzero element of the complement of $\text{NR}(L)$ such that $x \in I_k$ for $0 \leq k \leq n - 2$. Let A be the restriction of $\text{ad } x$ to $\text{NR}(L)$. Suppose $\dim \text{NR}(L) = m$ and let $\lambda_1, \dots, \lambda_m$ be the eigenvalues of A . Then since $\text{ad } x$ maps L into $\text{NR}(L)$ (see [32], p. 51, Theorem 13), we have that, except for some extra occurrences of 0, these are also the eigenvalues of $\text{ad } x$. Hence

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

If $f = X^m + a_1X^{m-1} + \dots + a_m$ is the characteristic polynomial of A , then by Newton's identities we have that $a_1 = a_2 = \dots = a_{n-1} = 0$. Suppose that $n - 1 \geq m$, then $\lambda_1 = \dots = \lambda_m = 0$. From this it follows that $\text{ad } x$ is nilpotent and $x \in \text{NR}(L)$ ([32], p. 45, Corollary 2). But this is a contradiction and hence $m + 1 > n$, so that the only possibility for m is $m = n$ (as L is not nilpotent). Now it is also clear that a_m is the only nonzero coefficient of f . Hence if we set

$$y = \sqrt[n]{-\frac{1}{a_m}}x$$

then the characteristic polynomial of $\text{ad } y$ restricted to $\text{NR}(L)$ will be $X^n - 1$.

Now the eigenvalues of $\text{ad } y$ on $\text{NR}(L)$ are ζ^i for $0 \leq i \leq n - 1$, where $\zeta = e^{2\pi i/n}$. Hence there is a basis $\{x_1, \dots, x_n\}$ of $\text{NR}(L)$ such that x_i is an eigenvector of $\text{ad } y$ with the eigenvalue ζ^{i-1} , i.e., $[y, x_i] = \zeta^{i-1}x_i$. By the Jacobi identity we have that $[x_i, x_j]$ is an eigenvector of $\text{ad } y$ with eigenvalue $\zeta^{i-1} + \zeta^{j-1}$. But if n is not divisible by 6, then a sum of two roots of unity is not a root of unity so that $[x_i, x_j] = 0$. It follows that L is the Lie

algebra of Example 2.9. \square

The conclusion is that the case where we have to calculate the ideals I_a up to $a = \dim L - 2$ is quite exceptional.

Remark. If L is defined over a field of characteristic $p > 0$, then the proof of Theorem 2.7 only fails because the use of Newton's identities may not lead to the desired conclusion. However, if $p \geq \dim L$, then this problem does not occur. So the algorithm also works in that case.

To the best of our knowledge the only algorithm that also works over all finite fields was described in [44]. There the author describes a polynomial time method to find the radical of an associative algebra over \mathbb{F}_q (see also [11]). By Theorem 2.6 this also leads to an algorithm for calculating the nilradical.

2.3 Evaluation

Here we compare the algorithm described in Section 2.2 with the other methods mentioned in Section 2.1. We did not implement the method described in [2], because it requires the computation of algebraic numbers. An example of a Lie algebra where this is necessary is provided by the Lie algebra L_n below. Hence we are left with three algorithms:

1. The Downward Method described in Section 2.2.
2. The Upward Method described in Section 2.1.
3. The method proposed in [44] that calculates the radical of $(\text{ad } L)^*$. We call it the *Radical Method*.

We used two examples of solvable Lie algebras to put the algorithms to a practical test. If $n \geq 2$ is an integer, then K_n is the subalgebra of the full matrix algebra $M_n(\mathbb{Q})$ generated by all upper triangular matrices. It is a solvable Lie algebra of dimension $(n+1)n/2$. The second example L_n is an $n+1$ dimensional Lie algebra with basis $\{x_0, \dots, x_n\}$ and commutation relations

$$\begin{aligned} [x_0, x_1] &= x_n \\ [x_0, x_i] &= x_{i-1} \quad \text{for } i > 1 \\ [x_i, x_j] &= 0 \quad \text{for } i, j > 0. \end{aligned}$$

Notice that this Lie algebra is isomorphic (over an extension of \mathbb{Q}) to the Lie algebra in Example 2.9.

We let the methods calculate the nilradicals of K_n and L_n for some values of n . The results are displayed in Tables 2.1 and 2.2, respectively. The values in the last two columns are part of the output of the Downward Method and the Radical Method respectively.

n	Downward	Upward	Radical	$\dim K_n$	k	$\dim(\text{ad } K_n)^*$
4	1	11	6	10	1	25
5	4	21	38	15	1	55
6	10	36	233	21	1	105
7	24	63	1334	28	1	182

Table 2.1: Computation times (in seconds) of the calculation of the nilradicals of K_n (the 6th column contains the number k such that $I_k = \text{NR}(L)$).

n	Downward	Upward	Radical	$\dim L_n$	k	$\dim(\text{ad } L_n)^*$
13	17	13	11	14	12	26
14	21	17	13	15	13	28
15	27	18	16	16	14	30
16	35	20	19	17	15	32

Table 2.2: Computation times (in seconds) of the calculation of the nilradicals of L_n (the 6th column contains the number k such that $I_k = \text{NR}(L)$).

From this we see that the Radical Method becomes very slow if the dimension of $(\text{ad } L)^*$ gets big. Furthermore, the order of the Upward Method appears to be less than the order of the Downward Method, although we have no bounds on the running time of the first method. On the Lie algebras L_n the Downward Method will not have a very good performance because it is the “worst case” Lie algebra for this method (see Proposition 2.10), whereas the Upward Method is particularly well suited for this Lie algebra. Also the Radical Method has a good performance in this case, mainly because the dimension of the associative algebra is not growing rapidly.

2.4 Acknowledgements

The Lie algebra of Example 2.9 was found by L. Rónyai. The proof of the uniqueness part of Proposition 2.10 originated in a conversation with A. Caranti.

Chapter 3

Cartan subalgebras

Cartan subalgebras are the talebearers of the semisimple Lie algebras in characteristic 0; the adjoint action of a Cartan subalgebra reveals the isomorphism type of such a Lie algebra. This will be exploited in Chapters 4 and 5 where algorithms for calculating the structure of semisimple Lie algebras will be given.

In Section 3.1 we briefly describe two previously published algorithms. In Sections 3.2, 3.3, and 3.4 a third algorithm is described. It finds a locally regular element in the space $\text{ad } L$. The eigenspace corresponding to the eigenvalue 0 of such element is a Cartan subalgebra. The algorithm needs a subroutine that finds a non-nilpotent element in a Lie algebra. This is described in Section 3.3. A slightly different algorithm for calculating a Cartan subalgebra is given for the case where the Lie algebra is of characteristic $p > 0$ and restricted (Section 3.4). Finally in Section 3.5 we give a practical evaluation of the algorithms.

3.1 Known algorithms

In [2] an algorithm for the construction of a Cartan subalgebra in a Lie algebra L of characteristic 0 is described. It uses a reduction based on the following lemma.

Lemma 3.1 *Let I be an ideal of L and let K be a subalgebra of L containing I such that K/I is a Cartan subalgebra of L/I . Let H be a Cartan subalgebra of K . Then H is a Cartan subalgebra of L .*

For the proof we refer to [1], Lemma 4.

Now let R be the solvable radical of L . Then the algorithm first finds a Cartan subalgebra of the semisimple Lie algebra L/R . Let K be a subalgebra of L containing R such that K/R is a Cartan subalgebra of L/R . Then K is solvable and the procedure finds a Cartan subalgebra in K . By Lemma 3.1 this is also a Cartan subalgebra of L . It follows that the

task of finding a Cartan subalgebra in an arbitrary Lie algebra is reduced to the cases where L is semisimple or solvable. In the semisimple case a maximal torus is constructed, starting with the semisimple part of a non-nilpotent element. Subsequently more semisimple parts of elements of the centraliser of this torus are added. In the solvable case the Lie algebra is divided by well-chosen ideals, again using Lemma 3.1.

In [52], H. Zassenhaus described an algorithm for finding a Cartan subalgebra in a Lie algebra of characteristic zero. It is based on the following lemma. For the proof we refer to [50], Theorem 4.4.4.8.

Lemma 3.2 *Let L be an n -dimensional Lie algebra and assume that L is restricted if L is of characteristic p . Let N be a nilpotent subalgebra of L . Set*

$$L_0(N) = \{x \in L \mid (\text{ad } y)^n x = 0 \text{ for all } y \in N\}.$$

Then $L_0(N)$ is a subalgebra of L and every Cartan subalgebra of $L_0(N)$ is also a Cartan subalgebra of L .

The strategy of the algorithm consists of trying to find a nilpotent subalgebra K of L such that $L_0(K)$ is a proper subalgebra of L . When such a subalgebra is found, recursion is applied to find a Cartan subalgebra H of $L_0(K)$ and by Lemma 3.2, H is also a Cartan subalgebra of L . The algorithm starts with an arbitrary nilpotent subalgebra K . If $L_0(K)$ happens to be equal to L , then two strategies for replacing K are possible. First of all, if the centraliser of K in L is bigger than K we can add an element of the complement to K in the centraliser and produce a bigger nilpotent subalgebra. We can do the same with the normaliser. However in this case, in order to get a nilpotent subalgebra, we must make sure that the element x of the complement acts nilpotently on K . If this happens not to be the case then x is a non-nilpotent element and hence the nilpotent subalgebra K spanned by x will have the property that $L_0(K) \neq L$.

3.2 Locally regular elements

Throughout this section V will be an n -dimensional vector space over the field F .

Definition 3.3 *Let M be a linear subspace of $\text{End}(V)$. Let $A \in M$ and let $V_0(A)$ be the Fitting null component of V relative to A (see Lemma 1.22). Then $A \in M$ is called locally regular in M if every $B \in M$ that stabilises $V_0(A)$ acts nilpotently on $V_0(A)$.*

Our objective is to give an algorithm for finding a locally regular element in a given subspace M of $\text{End}(V)$. The algorithm starts with an element $A \in M$. If this element is not locally regular, then an element $B \in M$ is constructed such that $V_0(B)$ is properly contained in $V_0(A)$. We first derive a useful statement about the dimension of $V_0(B)$.

Let $M \subset \text{End}(V)$ be given by a basis $\{A_1, \dots, A_s\}$ and let x_1, \dots, x_s be s indeterminates. Consider a generic element $A = x_1 A_1 + \dots + x_s A_s$, which lives in $\text{End}(V \otimes_F F(x_1, \dots, x_s))$. Let $f(T) \in F(x_1, \dots, x_s)[T]$ be the characteristic polynomial of A . Then

$$f(T) = \det(TI_n - A) = T^n + f_1 T^{n-1} + \dots + f_{n-1} T + f_n,$$

where $f_i \in F[x_1, \dots, x_s]$ and $\deg f_i = i$ if $f_i \neq 0$. Furthermore, the characteristic polynomial of an element $B = \beta_1 A_1 + \dots + \beta_s A_s$ is obtained by substituting $x_i = \beta_i$ in f .

Lemma 3.4 *Let $B = \beta_1 A_1 + \dots + \beta_s A_s$ be an element from M . Then the following are equivalent:*

1. $\dim V_0(B) = d$,
2. $f_{n-d}(\beta_1, \dots, \beta_s) \neq 0$ and $f_j(\beta_1, \dots, \beta_s) = 0$ for $n - d < j \leq n$,
3. $d = n - \text{rank}(B^n)$.

Proof. Since $V_0(B)$ is the generalised eigenspace of B corresponding to the eigenvalue 0, we have that the dimension of this space is equal to the multiplicity of 0 as a root of the characteristic polynomial of B . Hence 1. and 2. are equivalent. The equivalence of 1. and 3. follows from $V_0(B) = \{v \in V \mid B^n v = 0\}$. \square

The next proposition allows us to control the coefficients of the elements $\sum \alpha_i A_i$ that we construct.

Proposition 3.5 *Let Ω be a subset of F of size $n + 1$. If F is of characteristic 0 then we take $\Omega = \{1, 2, \dots, n + 1\}$. Let M be a subspace of $\text{End}(V)$ with basis $\{A_1, \dots, A_s\}$. Let $A = \alpha_1 A_1 + \dots + \alpha_s A_s$ be an element of M . Then we can find an element $B = \beta_1 A_1 + \dots + \beta_s A_s \in M$ such that $\dim V_0(B) \leq \dim V_0(A)$ and $\beta_i \in \Omega$ for $1 \leq i \leq s$ in deterministic polynomial time.*

Proof. Let d be the dimension of $V_0(A)$. Then by Lemma 3.4 we have that $f_{n-d}(\alpha_1, \dots, \alpha_s)$ is nonzero. Also, by the same lemma, we must look for elements β_1, \dots, β_s in Ω such that $f_{n-d}(\beta_1, \dots, \beta_s) \neq 0$. By Lemma 1.35 this can be done at the expense of at most $s(n-d+1)$ tests whether $f_{n-d}(u) = 0$ on vectors $u \in (\Omega \cup \{\alpha_1, \dots, \alpha_s\})^s$. By Lemma 3.4 this can be done by inspecting the rank of B^n (where $B = \beta_1 A_1 + \dots + \beta_s A_s$). Hence these tests can be performed in time polynomial in the sizes of the matrices A_i and the numbers α_i . \square

In the sequel we let M be a linear subspace of $\text{End}(V)$ with basis $\{A_1, \dots, A_s\}$. If $A \in M$, then $N_M(V_0(A))$ will denote the set of all elements of M that stabilise $V_0(A)$ (in linear matrix action). We also fix a subset Ω of F of size $n + 1$. If F is of characteristic zero, then we take $\Omega = \{1, 2, \dots, n + 1\}$. The key to the algorithm will be the following proposition.

Proposition 3.6 *Let A be an element of M . Suppose $B \in N_M(V_0(A))$ does not act nilpotently on $V_0(A)$. Then there is an element c_0 from Ω such that $V_0(A + c_0(B - A))$ is properly contained in $V_0(A)$.*

Proof. (c.f., the proof of Lemma 15.2 A in [29].) Set $W = V_0(A)$ and $D_c = A + c(B - A)$ for $c \in F$. As $A, B \in N_M(V_0(A))$, we have that for all $c \in F$ the map D_c leaves W invariant. Hence D_c induces a transformation of V/W . It follows that the characteristic polynomial of D_c is the product of the characteristic polynomials of D_c on W and on V/W . Let $d = \dim W$ and let f_W be the characteristic polynomial of D_c on W and similarly for $f_{V/W}$. Then

$$f_W(T, c) = T^d + f_1(c)T^{d-1} + \cdots + f_d(c)$$

and

$$f_{V/W}(T, c) = T^{n-d} + g_1(c)T^{n-d-1} + \cdots + g_{n-d}(c),$$

where the f_i and g_i are polynomials in c of degree less than or equal to i .

By construction all eigenvectors of A belonging to the eigenvalue 0 lie in W . Hence $g_{n-d}(0) \neq 0$ so that g_{n-d} is not the zero polynomial. Furthermore, because B does not act nilpotently on W , there is an f_i such that $f_i(1) \neq 0$. In particular this f_i is not the zero polynomial. The degree of $f_i g_{n-d}$ is less than or equal to n , hence there is a $c_0 \in \Omega$ such that $f_i(c_0)g_{n-d}(c_0) \neq 0$. From $g_{n-d}(c_0) \neq 0$ it follows that $V_0(A + c_0(B - A))$ is contained in W . And $f_i(c_0) \neq 0$ implies that $V_0(A + c_0(B - A))$ is properly contained in W . \square

Remark. We can find an appropriate $c_0 \in \Omega$ by “trial and error”: after at most $n + 1$ computations of the dimension of a space of the form $V_0(A + c_0(B - A))$ we are done.

Now we are ready to formulate the algorithm for finding a locally regular element. We need an auxiliary procedure `NonNilpotentElement`(M, A) which returns an element $B \in M$ that does not act nilpotently on $V_0(A)$. If no such element exists (in particular when $V_0(A) = 0$) then it will return the zero element of M . Here we assume that such a procedure exists. In the next section we will construct such a procedure in the case where $M = \text{ad } L$, for a Lie algebra L .

Algorithm `LocallyRegularElement`

Input: A basis $\{A_1, \dots, A_s\}$ of a subspace M of $\text{End}(V)$.

Output: A locally regular element $A = \sum \alpha_i A_i$ of M such that $\alpha_i \in \Omega$.

Step 1 $A := 0$;

Step 2 $B := \text{NonNilpotentElement}(M, A)$;

Step 3 if $B = 0$ then return A ; fi;

Step 4 Select an element $c_0 \in \Omega$ such that $V_0(A + c_0(B - A))$ is properly contained in $V_0(A)$;

Step 5 $A := A + c_0(B - A)$;

Step 6 Replace A by an element $B = \sum \alpha_i A_i$ such that $\alpha_i \in \Omega$ and $\dim V_0(B) \leq \dim V_0(A)$;
Return to Step 2;

The computability of Steps 4 and 6 is ensured by the Propositions 3.6 and 3.5, respectively. Since the dimension of $V_0(A)$ decreases every round of the iteration, the procedure will finish. Furthermore, it is clear that upon termination A will be a locally regular element.

We now consider the complexity of the algorithm. The body of the loop will be executed at most $n = \dim V$ times. Step 4 requires the calculation of the Fitting null component of at most $n + 1$ linear transformations. Furthermore, Step 6 needs a polynomial number of operations by Proposition 3.5. Also in this step the size of the coefficients is kept under control. So the polynomiality of the method depends on the auxiliary procedure `NonNilpotentElement`. If this procedure runs in polynomial time, then the algorithm `LocallyRegularElement` will also run in polynomial time.

Proposition 3.7 *Suppose that M belongs to a class of subspaces of $\text{End}V$ for which the routine `NonNilpotentElement` runs in polynomial time. Then we can find a locally regular element of M in polynomial time.*

There is also a very efficient randomised algorithm available. It is of Las Vegas type provided that we have an efficient method for testing whether a given element is locally regular. It is based on the following proposition.

Proposition 3.8 *Let Δ be a subset of F of size at least n/ϵ for some $\epsilon > 0$. Let $\alpha_1, \dots, \alpha_s$ be elements chosen randomly and uniformly from Δ . Then the probability that the element $A = \alpha_1 A_1 + \dots + \alpha_s A_s$ is locally regular is at least $1 - \epsilon$.*

Proof. From Lemma 3.4 it follows that A is locally regular if and only if $g(\alpha_1, \dots, \alpha_s) \neq 0$ where g is a polynomial of degree at most n . By Corollary 1.34 we have that the probability that $g(\alpha_1, \dots, \alpha_s) = 0$ is at most $n/|\Delta| \leq n/(n/\epsilon) = \epsilon$. \square

3.3 Finding a non-nilpotent element in a Lie algebra

Throughout this section L will be a finite-dimensional Lie algebra over the field F . Here we realise the procedure `NonNilpotentElement`(M, A) where $M = \text{ad } L$. By the following lemma this task is reduced to the task of finding non-nilpotent elements in Lie algebras.

Lemma 3.9 *Set $M = \text{ad } L$ and let $A \in M$. Set $K = L_0(A)$. Then K is a subalgebra of L and $N_L(K) = K$. We have that $B \in N_M(K)$ does not act nilpotently on K if and only if there is an element $x \in K$ such that $\text{ad}_K x$ is not a nilpotent map and $B = \text{ad } x$.*

Proof. The first two statements are Lemma 15.1 and Lemma 15.2B from [29]. Since $N_L(K) = K$ we have that $N_M(K) = \text{ad}_L(K)$. So if B is an element from $N_M(K)$ not acting nilpotently on K , then $B = \text{ad}_L x$ for some $x \in K$ such that $\text{ad}_K x$ is not nilpotent. \square

By Engel's theorem ([32], p. 36) there exist elements x in L such that $\text{ad } x$ is not nilpotent if and only if L is not a nilpotent Lie algebra. Now we consider the problem of finding such an element.

If L is defined over a field of characteristic 0, then there is a particularly simple method available.

Proposition 3.10 *Let L be a non-nilpotent Lie algebra over a field of characteristic 0 with basis $\{x_1, \dots, x_n\}$, then the set*

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

contains a non-nilpotent element.

Proof. If L is solvable but not nilpotent then by Corollary 2 on p. 45 of [32], we have that the nilradical 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 Theorem 5 on p. 73 of [32] implies that the Killing form of L is not identically zero. It follows that there exist basis elements x_i and x_j for $1 \leq i < j \leq n$ 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.

Proposition 3.11 *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 generate a subalgebra of L of dimension $m + 1$, acting nilpotently on L .*

Proof. Let \overline{K} denote the subalgebra generated by K and x . The fact that \overline{K} is of dimension $m + 1$ is trivial. Suppose that $\text{ad } x$ is 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 Theorem 1 on p. 33 of [32] the associative algebra A generated by U is nilpotent. The conclusion is that $\text{ad}_L \overline{K} \subset A$ is also nilpotent. \square

If we start with $K = 0$ and repeatedly apply Proposition 3.11 then we either find a non-nilpotent element, or after n steps we have that $K = L$ whence L is nilpotent. An element $x \in N_L(K) \setminus K$ can be found by calculating $N_L(K)$. Alternatively, we can construct a sequence of elements in the following way. First we fix a basis of K . Let x be an element of L not lying in K . If for some basis element y of K we have that $[x, y] \notin K$, then replace x by $[x, y]$. Since K acts nilpotently on L we need no more than $\dim L - 1$ such replacement operations to obtain an element lying in $N_L(K) \setminus K$.

Now the procedure `NonNilpotentElement` can be implemented using the method for finding a non-nilpotent element. It clearly yields a polynomial time method if F is of characteristic 0, or if F is a finite field. In the other cases (i.e., when F is an infinite field of prime characteristic) we do not have a bound on the size of the element produced by Proposition 3.11.

It is also possible to use a randomised Las Vegas type method that finds its justification in the following proposition. The proof is similar to the proof of Proposition 3.8; therefore we do not formulate it.

Proposition 3.12 *Let L be a non-nilpotent Lie algebra with basis x_1, \dots, x_n . Let Δ be a subset of F of size at least n/ϵ . Let $\alpha_1, \dots, \alpha_n$ be elements chosen randomly and uniformly from Δ . Then the probability that $\alpha_1 x_1 + \dots + \alpha_n x_n$ is not nilpotent is at least $1 - \epsilon$.*

Remark. In [2] also an algorithm is described for finding a non-nilpotent element in a Lie algebra of characteristic zero. It is quite complicated so we do not reproduce it here.

3.4 Cartan subalgebras

Let L be a Lie algebra over the field F , where F is of characteristic zero, or a finite field. By the results of the preceding sections we can find a locally regular element in the space $M = \text{ad } L$ in polynomial time. By the next proposition this also yields a polynomial time algorithm for finding a Cartan subalgebra in L .

Proposition 3.13 *Let L be a finite-dimensional Lie algebra. Let $A \in \text{ad } L$ and set $H = L_0(A)$. Then H is a Cartan subalgebra of L if and only if A is locally regular in $\text{ad } L$.*

Proof. Suppose that A is locally regular. By Lemma 15.1 and Lemma 15.2B of [29] we know that H is a subalgebra of L and $N_L(H) = H$. Let x be an element of H . Then $\text{ad}_L x \in N_{\text{ad } L}(H)$ and hence $\text{ad}_L x$ acts nilpotently on H . By Engel's theorem ([32], p.

36), it now follows that H is nilpotent and hence it is a Cartan subalgebra. For the other direction let $x \in L$ be such that $\text{ad}_L x$ stabilises H . Then $x \in N_L(H) = H$ so that $\text{ad}_L x$ acts nilpotently on H . The conclusion is that A is locally regular in $\text{ad } L$. \square

The algorithm for finding a locally regular element is only guaranteed to work if the field of definition is of size at least $\dim L + 1$. However, for restricted Lie algebras over finite fields there a slightly different approach is possible.

Algorithm RestrictedCartan

Input: A restricted Lie algebra L .

Output: A basis of a Cartan subalgebra of L .

Step 1 $A := \emptyset$;

Step 2 $a := \text{NonNilpotentElement}(L_0(A), 0)$;

Step 3 **if** $a = 0$ (i.e., $L_0(A)$ is nilpotent) **then return** $L_0(A)$; **fi**;

Step 4 $A := A \cup \{a\}$; Go to Step 2;

Clearly every step is computable. Furthermore, the quantity $\dim L_0(A)$ decreases at every iteration, so that the algorithm finishes after at most n rounds. The correctness of the algorithm follows from Lemma 3.2. If L is nilpotent, then L is its own Cartan subalgebra. Otherwise, L contains a non-nilpotent element a . The subalgebra $K = L_0(\{a\})$ is properly contained in L . By Lemma 3.2 every Cartan subalgebra of K will be a Cartan subalgebra of L . Furthermore, the Lie algebra K will be restricted (see the proof of Corollary 4.4.4.9 of [50]). The conclusion is that we may continue with K in place of L .

By the results of Section 3.3 we can find non-nilpotent elements in Lie algebras over finite fields in polynomial time. Hence the algorithm `RestrictedCartan` is polynomial for restricted Lie algebras over finite fields.

Remark. By Lemma 3.2, it also follows that the algorithm `RestrictedCartan` will work for Lie algebras of characteristic zero. In that case however, we do not have satisfactory bounds on the sizes of the elements of the set A .

3.5 Evaluation

If L is a Lie algebra of characteristic 0, then we have the following algorithms for finding a Cartan subalgebra:

LRE The algorithm that calculates a locally regular element in $\text{ad } L$.

NSA The algorithm that recursively finds a Cartan subalgebra in $L_0(K)$ where K is a nilpotent subalgebra of L (see Section 3.1).

SeSo The method that reduces the task of finding a Cartan subalgebra of L to the cases where L is either semisimple or solvable (see Section 3.1).

We tried these methods on the Lie algebras $\mathfrak{sl}_n(\mathbb{Q})$ for $n = 4, 5, 6, 7, 8$. The structure constants of these Lie algebras were taken relative to the standard basis of $\mathfrak{sl}_n(\mathbb{Q})$. By e_{ij}^n we denote the $n \times n$ matrix with a 1 on position (i, j) and zeros elsewhere. Then the standard basis of $\mathfrak{sl}_n(\mathbb{Q})$ is given by

$$\{e_{ij}^n \mid 1 \leq i, j \leq n \text{ and } i \neq j\} \cup \{e_{ii}^n - e_{i+1, i+1}^n \mid 1 \leq i < n\}.$$

The results are shown in Table 3.1 and graphically in Figure 3.1

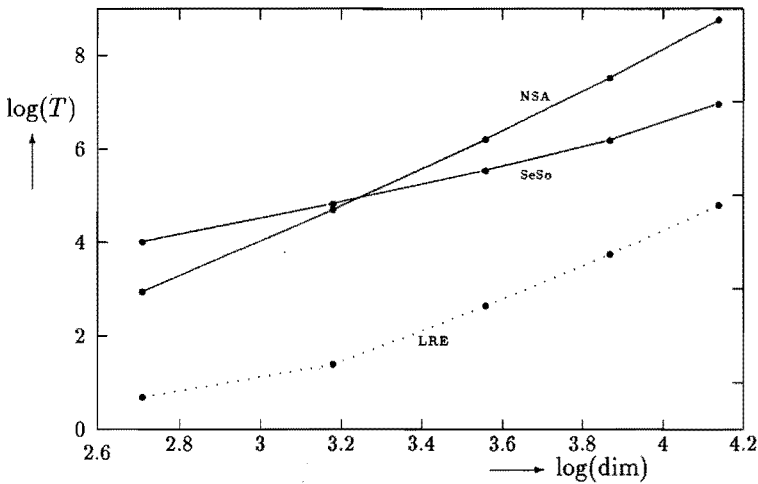


Figure 3.1: Running times (in seconds) of the methods LRE, NSA, and SeSo on the input $\mathfrak{sl}_n(\mathbb{Q})$ for $n = 4, 5, 6, 7, 8$.

There are also two deterministic methods for restricted Lie algebras over a field F such that $|F| > \dim L$, namely LRE, and RestrictedCartan. We also tried these methods on $\mathfrak{sl}_n(\mathbb{F}_{625})$. The results are collected in Table 3.2.

If L is defined over a big field (e.g., of size at least $2 \dim L$), then Propositions 3.8 and 3.13 give a straightforward randomised method to find a Cartan subalgebra in L . We also tested these methods. The results are displayed in Table 3.3.

On Figure 3.1 and Table 3.1, 3.2 and 3.3 we make the following comments:

n	$\dim \mathfrak{sl}_n$	LRE	NSA	SeSo
4	15	2	19	55
5	24	4	108	124
6	35	14	494	252
7	48	42	1835	480
8	63	119	6330	1030

Table 3.1: Running times (in seconds) of the methods LRE, NSA, and SeSo on the input $\mathfrak{sl}_n(\mathbb{Q})$.

n	Restricted Cartan	LRE
4	2	2
5	7	8
6	32	34
7	128	136

Table 3.2: Running times (in seconds) of the LRE method and RestrictedCartan on the input $\mathfrak{sl}_n(\mathbb{F}_{625})$.

- Of the three deterministic methods in characteristic zero, the LRE method is by far the fastest. The SeSo method suffers from the fact that it has to compute the Jordan decomposition of a matrix. For the NSA method the path towards a suitable nilpotent subalgebra is fairly long.
- The scale of Figure 3.1 is logarithmic so that the points should lie on a straight line with slope equal to the order of the method. It turns out that LRE has an order of approximately 2.86, NSA an order of 4.06 and SeSo an order of 2.05. Since the order of the SeSo method is smaller than the order of the LRE method, the first method will eventually be faster. However, the numbers in Table 3.1 indicate that the dimension of the Lie algebra for which this happens will be so high that this is of minor practical value.

n	LRE ($\mathfrak{sl}_n(\mathbb{Q})$)	Random ($\mathfrak{sl}_n(\mathbb{Q})$)	Res. Cart. ($\mathfrak{sl}_n(\mathbb{F}_{625})$)	Random ($\mathfrak{sl}_n(\mathbb{F}_{625})$)
4	2	25	2	0.3
5	4	985	7	1
6	14	16676	32	3

Table 3.3: Running times (in seconds) of the deterministic and the randomised methods for $\mathfrak{sl}_n(\mathbb{Q})$ and $\mathfrak{sl}_n(\mathbb{F}_{625})$.

- It is seen that the algorithm `RestrictedCartan` is somewhat faster than the LRE method. Also the first algorithm can be applied more generally. So for restricted Lie algebras this method is to be preferred.
- The randomised method for $\mathfrak{sl}_n(\mathbb{Q})$ explodes. This is caused by the following: in $\mathfrak{sl}_n(\mathbb{Q})$ the algorithm for the computation of a non-nilpotent element will return a basis element or the sum of two basis elements, whereas a random element will be a sum of all basis elements, most of them with nonzero coefficient. In the computation of the Fitting null component, the first case will lead to a set of sparse equations, whereas in the second case we will have a nonsparse set of equations, for which the Gaussian elimination is much more difficult.
- The randomised method for $\mathfrak{sl}_n(\mathbb{F}_{625})$ has a very good performance. This can be explained by the fact that in this case the Gaussian elimination is fast even for non sparse equations (there is no coefficient growth as in the characteristic zero case). Furthermore, usually after one or two steps a locally regular element is found.

Remark. We have used the LRE method without the replacement step (Step 6 of the algorithm `LocallyRegularElement`), because this step slows things down considerably. Furthermore, it is more of theoretical than of practical value (in practice after two or three steps a Cartan subalgebra is found).

3.6 Acknowledgement

The greater part of this section is a reformulation of [25] (joint work with G. Ivanyos and L. Rónyai).

Chapter 4

The decomposition of a semisimple Lie algebra

Let L be a semisimple Lie algebra over a field F of characteristic 0. Then L is a direct sum of simple ideals (see [32], p. 71). In this chapter we present methods to obtain this direct sum decomposition. The Cartan decomposition of L relative to a Cartan subalgebra gives the key to understanding its structure. Here we use a somewhat weaker instrument, called the generalised Cartan decomposition. In Section 4.1 it is shown how to obtain the direct sum decomposition from a generalised Cartan decomposition. Then in Section 4.2 algorithms are given for obtaining a generalised Cartan decomposition.

4.1 The generalised Cartan decomposition

First we transcribe two results from [32] on nilpotent Lie algebras of linear transformations. We recall that $V_0(A)$ is the Fitting null component of the vector space V relative to the linear transformation A (see Lemma 1.22).

Lemma 4.1 *Let A, B be linear transformations in a finite-dimensional vector space satisfying*

$$[A, [A, \dots, [A, B] \dots]] = 0 \quad (n \text{ factors } A)$$

for some n . Let p be a polynomial, then $V_0(p(A))$ is invariant under B .

Proof. See [32], p. 40. \square

Theorem 4.2 *Let N be a nilpotent Lie algebra of linear transformations acting on a finite-dimensional vector space V . Then there is a decomposition*

$$V = V_1 \oplus \dots \oplus V_s$$

where each V_i is invariant under N and the restriction of every element $A \in N$ to V_i has a minimum polynomial that is a prime power ($1 \leq i \leq s$).

Proof. (cf. [32] p. 41) If every element of N has a minimum polynomial that is a prime power, then we set $t = 1$ and $V_1 = V$. Otherwise, let A be an element of N such that its minimum polynomial f_A factors as $f_A = f_1^{m_1} \cdots f_t^{m_t}$ where the f_i are distinct irreducible polynomials. Then

$$V = V_0(f_1(A)) \oplus \cdots \oplus V_0(f_t(A)).$$

The minimum polynomial of the restriction of A to $V_0(f_i(A))$ is $f_i^{m_i}$. Furthermore, by Lemma 4.2 we have that each $V_0(f_i(A))$ is invariant under N , so that we can proceed by induction on the dimension. \square

Now we turn our attention to semisimple Lie algebras. Let L be a semisimple Lie algebra over a field of characteristic 0 with Cartan subalgebra H . Then via the adjoint representation H acts as a nilpotent Lie algebra of linear transformations on the vector space L . Hence, by Theorem 4.2, L decomposes as

$$L = L_0 \oplus L_1 \oplus \cdots \oplus L_s,$$

where each L_i is stable under $\text{ad } H$ and the minimum polynomial of the restriction of $\text{ad } h$ to L_i is a prime power for all $h \in H$. Now, because $\text{ad } h$ is a semisimple transformation (see Corollary 15.3 of [29]), we have that this minimum polynomial is irreducible. Furthermore, L_0 is the space corresponding to the polynomial X , i.e.,

$$L_0 = \{x \in L \mid (\text{ad } h)^m x = 0 \text{ for all } h \in H \text{ and some } m > 0\}.$$

Also, by Proposition 1 of Chapter III of [32], the space L_0 is equal to H . These considerations lead to the following definition.

Definition 4.3 *Let L be a semisimple Lie algebra with Cartan subalgebra H . A generalised Cartan decomposition of L with respect to H is a decomposition*

$$L = L_1 \oplus \cdots \oplus L_s \oplus H$$

such that L_i is stable under $\text{ad } H$ and the restriction of $\text{ad } h$ to L_i has an irreducible minimum polynomial for $h \in H$ and $1 \leq i \leq s$.

Remark. Let L be a semisimple Lie algebra with Cartan subalgebra H . If the minimum polynomials of the elements of H split into linear factors over the ground field, then the spaces L_i in the generalised Cartan decomposition will be the common eigenspaces of the Cartan subalgebra. It follows that in this case the generalised Cartan decomposition coincides with the Cartan decomposition as described in Section 1.1.

The next theorem states that the generalised Cartan decomposition of L with respect to a Cartan subalgebra is compatible with the direct sum decomposition of L .

Theorem 4.4 *Let L be a semisimple Lie algebra with Cartan subalgebra H and let*

$$L = L_1 \oplus \cdots \oplus L_s \oplus H$$

be a generalised Cartan decomposition of L with respect to H . Suppose that L decomposes as a direct sum of ideals, $L = I_1 \oplus I_2$. Then every L_i is contained in either I_1 or I_2 .

Proof. By Proposition 1.27 there are two orthogonal idempotents E_1, E_2 commuting with $\text{ad } L$ such that $E_1 + E_2$ is the identity on L and $I_l = E_l L$ for $l = 1, 2$. Hence

$$H = (E_1 + E_2)H \subset E_1 H \oplus E_2 H.$$

Let $g \in H$, then $(\text{ad } h)^m E_1 g = E_1 (\text{ad } h)^m g = 0$ for all $h \in H$ so $E_1 g \in H$ (because $L_0(H) = H$). Therefore we have that $E_1 H \subset H$ and similarly $E_2 H \subset H$. Set $H_l = E_l H$ for $l = 1, 2$, then it follows that H decomposes as $H = H_1 \oplus H_2$ and H_l is a Cartan subalgebra of I_l for $l = 1, 2$. By the definition of generalised Cartan decomposition, there is an element $h \in H_1 \cup H_2$ such that the restriction of $\text{ad } h$ to L_i is nonsingular. (Otherwise the minimum polynomial of the restriction of every element of a basis of H to L_i would be X . This implies that $[H, L_i] = 0$ and by definition of Cartan subalgebra we have $L_i \subset H$, a contradiction.)

First suppose that $h \in H_1$. Then also $h \in I_1$ so that $\text{ad } h(L) \subset I_1$ and in particular $\text{ad } h(L_i) \subset I_1$. Now the fact that $\text{ad } h$ is nonsingular on L_i implies that $L_i = [h, L_i] \subset I_1$. In the same way $h \in H_2$ implies that L_i is contained in I_2 . \square

This theorem implies that the following algorithm is correct.

Algorithm Decompose

Input: A semisimple Lie algebra L .

Output: A list of the direct summands of L .

Step 1 Compute a generalised Cartan decomposition $L = L_1 \oplus \cdots \oplus L_s \oplus H$.

Step 2 For $1 \leq i \leq s$ determine a basis of the ideal of L generated by L_i .

Step 3 Delete multiple instances from the list.

In Step 1 an oracle is called that computes a generalised Cartan decomposition of a semisimple Lie algebra. In the next section we will describe some methods to find such a decomposition.

The algorithm is clearly polynomial, except maybe for Step 1, where the oracle is called. The complexity of this oracle will be discussed in the next section.

4.2 Calculating a generalised Cartan decomposition

4.2.1 Splitting elements

Throughout this section L will be a semisimple Lie algebra over a field F of characteristic 0 with Cartan subalgebra H .

Definition 4.5 An element $h \in H$ is called a splitting element if the minimal polynomial of $\text{ad } h$ has degree $\dim L - \dim H + 1$.

Let Φ be the root system of L and set $\Phi^* = \Phi \cup \{0\}$.

Lemma 4.6 Let $L = \bigoplus_{\alpha \in \Phi^*} L_\alpha$ be a Cartan decomposition of L . Then $h \in H$ is a splitting element if and only if all numbers $\alpha(h)$ are different for $\alpha \in \Phi^*$.

Proof. Let Δ be a maximal subset of Φ^* such that for no pair $\alpha, \beta \in \Delta$ we have $\alpha(h) = \beta(h)$. Then the minimum polynomial of $\text{ad } h$ is

$$\prod_{\alpha \in \Delta} (X - \alpha(h)).$$

This polynomial is of degree $\dim L - \dim H + 1$ if and only if $\Delta = \Phi^*$. \square

Proposition 4.7 Let $h_0 \in H$ be a splitting element. Let

$$L = L_0 \oplus L_1 \oplus \cdots \oplus L_s \tag{4.1}$$

be the decomposition of L such that the restriction of $\text{ad } h_0$ to L_i has an irreducible minimum polynomial ($1 \leq i \leq s$). The subspace corresponding to the polynomial X is L_0 . Then $H = L_0$ and the decomposition (4.1) is a generalised Cartan decomposition.

Proof. Lemma 4.6 implies that $\dim L_0 = \dim H$, and hence $L_0 = H$. Suppose that there is an $h \in H$ such that the restriction of $\text{ad } h$ to L_j has a reducible minimum polynomial. Then $L_j = V_1 \oplus V_2$ and $\text{ad } h_0$ has the same minimum polynomial on V_1 as on V_2 . So $\text{ad } h_0$ has an eigenvalue of multiplicity at least 2. But this contradicts Lemma 4.6. \square

Proposition 4.8 Splitting elements exist in H .

Proof. Let $\{h_1, \dots, h_l\}$ be a basis of H and let $h = \sum \epsilon_i h_i$ be an element of H . Let α, β be two elements of Φ^* . Then the eigenvalue of h on L_α is $\sum \epsilon_i \alpha(h_i)$. Consider the function

$$f_{\alpha, \beta}(\lambda_1, \dots, \lambda_l) = \sum_{i=1}^l (\alpha(h_i) - \beta(h_i)) \lambda_i.$$

Then the set of zeros of $f_{\alpha,\beta}$ corresponds to the elements of H that have the same eigenvalue on L_α as on L_β . But this set is a hyperplane in H . So we must choose an element $h_0 \in H$ lying outside a finite set of hyperplanes. Since F is infinite, there are such elements. \square

We define two abbreviations: $N = (\dim L - \dim H)/2$ (i.e., the number of positive roots), $m = N(N + 1)/2$.

Proposition 4.9 *Let $0 < \epsilon < 1$ and let Ω be a subset of F of size at least m/ϵ . Let $\epsilon_1, \dots, \epsilon_l$ be elements chosen uniformly and independently from Ω . Then the probability that $h = \sum \epsilon_i h_i$ is a splitting element is at least $1 - \epsilon$.*

Proof. Let $\{\alpha_1, \dots, \alpha_N\}$ be the set of positive roots. Set $\alpha_0 = 0$ in H^* . Set

$$f_{ij}(x_1, \dots, x_l) = \sum_{k=1}^l (\alpha_i(h_k) - \alpha_j(h_k)) x_k,$$

and set

$$G(x_1, \dots, x_l) = \prod_{0 \leq i < j \leq N} f_{ij}(x_1, \dots, x_l).$$

Then $\sum \epsilon_i h_i$ is not a splitting element if and only if $G(\epsilon_1, \dots, \epsilon_l) = 0$. Now since $\deg G = m$, Corollary 1.34 implies that the probability that $G(\epsilon_1, \dots, \epsilon_l) = 0$ is less than $m/|\Omega| \leq \epsilon$. \square

Proposition 4.9 gives a powerful randomised (Las Vegas type) algorithm for finding a splitting element in H . We also consider deterministic algorithms for finding a splitting element. Let G be the polynomial in the proof of Proposition 4.9. Substitute y^{i-1} for x_i in G . This yields a polynomial in $F[y]$ of degree at most $m(\dim H - 1)$. Hence by trying at most $m(\dim H - 1) + 1$ values for y , we obtain a number ξ such that $G(1, \xi, \dots, \xi^{l-1}) \neq 0$. A disadvantage of this algorithm is the fact that the numbers y^{i-1} may be big even if y is small. However, if we have found a vector $\epsilon = (\epsilon_1, \dots, \epsilon_l)$ such that $G(\epsilon) \neq 0$ then we can apply Lemma 1.35. Let Ω be a subset of F of size at least $m + 1$. Then we can find a vector $\xi = (\xi_1, \dots, \xi_l) \in \Omega^l$ such that $G(\xi) \neq 0$ at the expense of at most $l(m + 1)$ tests whether $G(v) = 0$ for vectors $v \in (\Omega \cup \{\epsilon_1, \dots, \epsilon_l\})^l$.

There is a second deterministic method relying on a second characterisation of splitting elements, expressed in the following lemma. Here $\langle 1, \text{ad } H \rangle$ is the associative subalgebra of $\text{End}(L)$ generated by 1 and $\text{ad } H$. It is of dimension $\dim L - \dim H + 1$.

Lemma 4.10 *An element $h \in H$ is a splitting element if and only if 1 and $\text{ad } h$ generate $\langle 1, \text{ad } H \rangle$.*

Proof. The dimension of the subalgebra of $\langle 1, \text{ad } H \rangle$ generated by 1 and $\text{ad } h$ is equal to the degree of the minimum polynomial of $\text{ad } h$. This degree equals $\dim L - \dim H + 1$ if

and only if h is a splitting element. \square

The following algorithm is analogous to the method of finding primitive elements in field extensions.

Algorithm PrimitiveSplittingElement

Input: A semisimple Lie algebra L with Cartan subalgebra H with basis h_1, \dots, h_l .

Output: A splitting element $h \in H$.

$h := h_1;$

$k := 2;$

while $k \leq l$ **do**

$c :=$ a constant such that $\langle 1, \text{ad}(h + ch_k) \rangle$ contains $\text{ad } h$ and $\text{ad } h_k$

$h := h + ch_k$

$k := k + 1;$

od;

Due to the following lemma, we can find an appropriate constant c in the first step of the loop.

Lemma 4.11 *Let h_1, h_2 be elements of H . Then there exists a $c \in \{0, \dots, m\}$ such that $\langle 1, \text{ad}(h_1 + ch_2) \rangle = \langle 1, \text{ad } h_1, \text{ad } h_2 \rangle$.*

Proof. The dimensions of the associative algebras do not change if we tensor with the algebraic closure of F , hence we may suppose that the ground field is algebraically closed. Let

$$L = L_1 \oplus \dots \oplus L_t$$

be the decomposition of L such that $\text{ad } h_1$ and $\text{ad } h_2$ have irreducible minimum polynomials on L_i for $1 \leq i \leq t$. So the restrictions of $\text{ad } h_1$ and $\text{ad } h_2$ to L_i are scalar multiplications ($1 \leq i \leq t$). It follows that $\langle 1, \text{ad } h_1, \text{ad } h_2 \rangle = A_1 \oplus \dots \oplus A_t$, where $A_i = \langle 1_{L_i} \rangle$ for $1 \leq i \leq t$. Hence the dimension of $\langle 1, \text{ad } h_1, \text{ad } h_2 \rangle$ is t . We have to prove that there is a $c \in \{0, \dots, m\}$ such that the dimension of $\langle 1, \text{ad}(h_1 + ch_2) \rangle$ is also t . This is equivalent to the decomposition of L relative to $\text{ad}(h_1 + ch_2)$ also having t components. Let $\{\alpha_1, \dots, \alpha_N\}$ be the set of positive roots and set $\alpha_0 = 0$ in H^* . For $0 \leq i < j \leq N$ we consider the function

$$g_{ij}(x) = \alpha_i(h_1 + xh_2) - \alpha_j(h_1 + xh_2) = \alpha_i(h_1) - \alpha_j(h_1) + x(\alpha_i(h_2) - \alpha_j(h_2)).$$

We have that $g_{ij}(x) \equiv 0$ if and only if L_{α_i} and L_{α_j} are both subspaces of L_k for some $k \in \{1, \dots, t\}$. Also we have that $g_{ij}(c_0) \neq 0$ if and only if $\text{ad}(h_1 + c_0h_2)$ separates L_{α_i} and L_{α_j} (i.e., the eigenvalues of $\text{ad}(h_1 + c_0h_2)$ on L_{α_i} and L_{α_j} differ). Now let $G(x)$ be

the product of all $g_{ij}(x)$ that are not identically zero. If $G(c_0) \neq 0$, then the decomposition of L relative to $\text{ad}(h_1 + c_0 h_2)$ has also t components. Since $\deg G \leq m$, there is a $c_0 \in \{0, \dots, m\}$ satisfying this requirement. \square

The deterministic methods for finding a splitting element are clearly polynomial. In order to find the Cartan decomposition we need to factor one polynomial so that the resulting algorithms are polynomial time algorithms.

Remark. If L is defined over a field of characteristic $p \neq 2, 3$, then most statements in this chapter hold for L , provided that the Killing form of L is nondegenerate. In this case L behaves like a semisimple Lie algebra of characteristic 0 (see [48]). The only difference is that splitting elements need not exist if the size of the ground field is small. By the proof of Proposition 4.9 it is seen that if $|F| > m$, then splitting elements exist in H . In that case the algorithm `PrimitiveSplittingElement` also works. The randomised method is only guaranteed to work if $|F| \gg m$.

Remark. If $h \in H$ is a splitting element, then $L_0(\text{ad } h) = H$, so that every element of $\text{ad } H$ acts trivially on $L_0(\text{ad } h)$. It follows that $\text{ad } h$ is a locally regular element in the space $\text{ad } H$ (see Definition 3.3). The converse is not true however. See Example 4.14 below. There $\text{ad } h_1$ is a locally regular element in $\text{ad } H$, but h_1 is not a splitting element.

4.2.2 Decomposable elements

Here we show a method for calculating a generalised Cartan decomposition of a Lie algebra L defined over a small finite field.

Definition 4.12 *Let V be a subspace of L stable under $\text{ad } H$. Let T_V be the associative algebra generated by 1 and $\text{ad } h|_V$ for $h \in H$. Let $x \in T_V$ and let f be the minimum polynomial of x . Then x is called decomposable (on V) if f is reducible. And x is called good (with respect to V) if f is irreducible and $\deg f = \dim T_V$.*

Algorithm GeneralisedCartanOverSmallFields

Input: A Lie algebra L with nondegenerate Killing form, defined over a field F with q elements.

Output: A generalised Cartan decomposition of L .

$H := \text{CartanSubalgebra}(L);$

$dec := \{L_1(H)\}; k := 1;$

while $k \leq \#dec$ **do**

$V :=$ the k -th element of dec ;

$T_V :=$ the associative algebra generated by 1 and $\text{ad}_V H$;

```

x := a random element from  $T_V$ ;
f := the minimum polynomial of x;
{f1, ..., fs} := the factorisation of f;
if f is irreducible then
  if  $\deg(f) = \dim T_V$  then k := k + 1; fi;
else
   $V_i := V_0(f_i(x)); (1 \leq i \leq s)$ 
   $dec := dec \cup \{V_1, \dots, V_s\}$ ;
  erase V from dec;
fi;
od;

```

Proof. First we prove the correctness of the algorithm. At termination we have that for every element V of the set dec there exists an $x_V \in T_V$ such that x_V is good with respect to V . Then x_V generates T_V and x_V has an irreducible minimum polynomial. This implies that T_V is a field and every element of T_V has an irreducible minimum polynomial. Hence the minimum polynomial of $\text{ad } h|_V$ is irreducible for every $h \in H$. The conclusion is that the elements of dec form a generalised Cartan decomposition.

To prove termination we must show that the random element x chosen in the algorithm is either decomposable or good with sufficiently high probability. Let V be a subspace of L that is stable under H . Let T_V be the associative algebra generated by $\text{ad } h|_V$ for $h \in H$. We have that T_V is a semisimple commutative associative algebra (see [47], Theorem II.1.2) so that T_V splits into a direct sum

$$T_V = F_1 \oplus \cdots \oplus F_s$$

where the F_i are finite extensions of the ground field F . This follows from Wedderburn's structure theorem (see [40]).

If $s = 1$ then $T_V = \mathbb{F}_q^m$. We estimate the probability that x is good. First of all, if $m = 1$ then all elements of T_V are good. Now suppose that $m > 1$. Let E be the subset of T_V consisting of all elements x of T_V that do not lie in a proper subfield of T_V . Now to every irreducible polynomial f over \mathbb{F}_q corresponds a subset of m elements of E (namely the set of the roots of f ; see [38], Theorem 2.14). Also the sets corresponding to different irreducible polynomials do not intersect. Let $N_q(m)$ be the number of irreducible polynomials over \mathbb{F}_q . Write $m = ab$ where b is the largest proper divisor of m . Then

$$\begin{aligned}
 |E| = mN_q(m) &= \sum_{d|m} \mu(d)q^{m/d} \\
 &\geq q^m - q^b - \cdots - q \\
 &= q^m - q \frac{q^b - 1}{q - 1}.
 \end{aligned}$$

(See [38], Theorem 3.25). An element $x \in T_V$ is good if and only if $x \in E$. And the probability that a randomly chosen element $x \in T_V$ lies in E is

$$\frac{|E|}{q^m} \geq (q^m - q \frac{q^b - 1}{q - 1}) / q^m = 1 - \frac{1}{q^{a-1}} \frac{1 - 1/q^b}{q - 1} \geq 1 - \frac{1}{q^{a-1}(q - 1)} \geq \frac{1}{2}.$$

Now let $s > 1$. We now estimate the probability that x is decomposable. First we have that $x = x_1 + \dots + x_s$ where $x_i \in F_i$ are randomly and independently chosen elements. The minimum polynomial of x is the least common multiple of the minimum polynomials of the x_i . So if x is *not* decomposable then all x_i have the same minimum polynomial. It follows that the subfields of the F_i generated by the x_i are all isomorphic. Let this subfield be \mathbb{F}_{q^m} . We may suppose that all F_i are equal to \mathbb{F}_{q^m} ; otherwise the probability that $x_i \in \mathbb{F}_{q^m}$ is less. Now we assume that we have chosen an element $x_1 \in F_1$ with an irreducible minimum polynomial f of degree m . Because there are exactly m elements in F_2 with minimum polynomial equal to f , we have that the probability that a randomly chosen element $x_2 \in F_2$ also has minimum polynomial f is equal to

$$\frac{m}{q^m} \leq \frac{1}{2}.$$

It follows that the probability that x is decomposable is $\geq 1/2$. The conclusion is that the probability that a randomly chosen element is either good or decomposable is $\geq 1/2$. Hence we expect to find such an element in at most two steps. \square

Corollary 4.13 *Let L be a Lie algebra over a finite field with a nondegenerate Killing form. Then there is a Las Vegas f -algorithm for calculating a generalised Cartan decomposition of L .*

Remark. The algorithm described in this section also works over big fields; then the random element from T will be a splitting element with high probability.

Remark. The method given in this section provides a simplicity test for Lie algebras that have a nondegenerate Killing form. But if L is a semisimple Lie algebra of characteristic p , then L may have a degenerate Killing form (examples are the Lie algebras of type W , S , H , K , see [19]). In this case the algorithm above may fail because the associative algebra generated by $\text{ad } H$ may not be a torus. However, a convenient way of solving this problem is provided by the Meataxe algorithm (see [28]). Let A be the associative algebra generated by $\text{ad } L$. Then L is a finite dimensional module for A and a proper submodule will correspond to an ideal of L . So L is simple if and only if L is irreducible as A -module. Now the MeatAxe package of GAP provides a method for testing irreducibility. It has the disadvantage that the dimension of A is substantially bigger than the dimension of L . For instance if L is simple of dimension n , then the dimension of A is n^2 (see the proof of

Theorem X.3 in [32]). However, for Lie algebras of small dimension the **MeatAxe** provides an efficient simplicity test. We tested the simplicity of the Lie algebra $A_2 \oplus B_2 \oplus G_2$ defined over \mathbb{F}_5 using the **MeatAxe**. It needed 27 seconds to decide that the module was not irreducible. Since in this case the Killing form is not degenerate, we also tried the algorithm described in this section; it needed 803 seconds to calculate the direct sum decomposition.

4.3 Examples

Example 4.14 Let L be a 6 dimensional Lie algebra with basis

$$\{h_1, x_1, y_1, h_2, x_2, y_2\}.$$

The structure constants of L are specified in Table 4.1.

$[h_1, x_1]$	=	$2x_1$	$[h_2, x_1]$	=	$2x_1$
$[h_1, y_1]$	=	$-2y_1$	$[h_2, y_1]$	=	$-2y_1$
$[h_1, x_2]$	=	$2x_2$	$[h_2, x_2]$	=	$-2x_2$
$[h_1, y_2]$	=	$-2y_2$	$[h_2, y_2]$	=	$2y_2$
$[x_1, y_1]$	=	$\frac{1}{2}h_1 + \frac{1}{2}h_2$	$[x_2, y_2]$	=	$\frac{1}{2}h_1 - \frac{1}{2}h_2$

Table 4.1: Nonzero products of the basis elements of a 6 dimensional Lie algebra.

Brackets of pairs of basis elements that are not present are assumed to be 0. The determinant of the matrix of the Killing form is 2^{16} , hence L is semisimple. As is easily verified, $H = \langle h_1, h_2 \rangle$ is a Cartan subalgebra.

First we take the ground field to be equal to \mathbb{Q} . Then the minimum polynomial of $\text{ad}(h_1 + 2h_2)$ is $X(X + 6)(X - 6)(X + 2)(X - 2)$ so that $h_1 + 2h_2$ is a splitting element. The decomposition of L relative to $\text{ad}(h_1 + 2h_2)$ is

$$L = \langle x_1 \rangle \oplus \langle x_2 \rangle \oplus \langle y_1 \rangle \oplus \langle y_2 \rangle \oplus \langle h_1, h_2 \rangle. \quad (4.2)$$

Now the ideal generated by x_1 is spanned by $\{x_1, y_1, (h_1 + h_2)/2\}$. Similarly, the ideal generated by x_2 is spanned by $\{x_2, y_2, (h_1 - h_2)/2\}$. It follows that we have found the decomposition of L into simple ideals.

The structure constants of L can also be viewed as elements of \mathbb{F}_3 . So now we take \mathbb{F}_3 as the ground field. Then the Killing form is nondegenerate so that we can apply the algorithm described in Section 4.2.2. The Fitting-one component $L_1(H)$ is spanned by $\{x_1, y_1, x_2, y_2\}$. The minimum polynomial of $\text{ad}(h_1 + 2h_2)$ is $X(X - 1)(X - 2)$ (denoting the elements of \mathbb{F}_3

by $\{0, 1, 2\}$. So $h_1 + 2h_2$ is a decomposable element and the corresponding decomposition of $L_1(H)$ is

$$L_1(H) = \langle x_1 \rangle \oplus \langle y_1, x_2 \rangle \oplus \langle y_2 \rangle.$$

Now we turn our attention to the space $V = \langle y_1, x_2 \rangle$ (the other two spaces are 1-dimensional and hence irreducible). The minimum polynomial of the restriction of $\text{ad}(h_1 + h_2)$ to V is $X(X - 2)$ so that $h_1 + h_2$ is a decomposable element. We again find the generalised Cartan decomposition (4.2).

Example 4.15 Let L be a Lie algebra with basis $\{x_1, \dots, x_6\}$ and multiplication table as shown in Table 4.2.

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	0	$2x_4$	$-2x_3$	$-2x_6$	$2x_5$
x_2	0	0	$2x_3$	$2x_4$	$-2x_5$	$-2x_6$
x_3	$-2x_4$	$-2x_3$	0	0	x_2	x_1
x_4	$2x_3$	$-2x_4$	0	0	x_1	$-x_2$
x_5	$2x_6$	$2x_5$	$-x_2$	$-x_1$	0	0
x_6	$-2x_5$	$2x_6$	$-x_1$	x_2	0	0

Table 4.2: Multiplication table of a 6 dimensional Lie algebra.

The determinant of the matrix of the Killing form is -2^{20} so that L is semisimple. A Cartan subalgebra of L is spanned by $\{x_1, x_2\}$. The minimum polynomial of $\text{ad}(x_1 + x_2)$ is $X(X^2 - 4X + 8)(x^2 + 4X + 8)$. Hence $x_1 + x_2$ is a splitting element. The corresponding generalised Cartan decomposition is

$$L = L_{3,4} \oplus L_{5,6} \oplus L_{1,2},$$

where $L_{i,j}$ is the subspace spanned by $\{x_i, x_j\}$. From the multiplication table it follows that the ideals generated by $L_{3,4}$ and $L_{5,6}$ are both equal to L . Hence, by Theorem 4.4 we have that L is a simple Lie algebra.

4.4 Evaluation

In Section 1.4.6 a more general method for decomposing a Lie algebra as a direct sum of ideals is described. Here we compare this general method with the special methods for semisimple Lie algebras that we propose. The general method has of course the advantage of being more general. However, a disadvantage of this method is the fact that it computes the centraliser of $\text{ad } L$ in the matrix algebra $M_{\dim L}(F)$. Since the dimension of this centraliser can be significantly bigger than the dimension of L , this may be a difficult task.

So now we have the following methods:

- (General method) The algorithm described in Section 1.4.6.
- (Random) The method that uses the randomised method to find a splitting element.
- (Subs.) The method that substitutes y^{k+1} for x_k in the polynomial G in order to find a splitting element.
- (Prim. elt.) The algorithm that finds a “primitive element” in the associative algebra generated by 1 and $\text{ad } H$.
- (Dec. elt.) The algorithm given in Section 4.2.2 that uses decomposable elements.

We have tested these methods on some direct sums of \mathfrak{sl}_2 , \mathfrak{sl}_3 and \mathfrak{sl}_4 over the field \mathbb{Q} . We took the standard Chevalley basis to produce the structure constants of L . The results are shown in Table 4.3. The randomised method used a set of $2m$ elements, from which it selected the coefficients of a splitting element. From this table we see that the running times of the general method increase rapidly if the dimension increases. This is caused by the computation of the centraliser of $\text{ad } L$ in $M_{\dim L}(F)$. The primitive element method has the disadvantage that it has to calculate the dimension of an associative algebra many times. It is seen that the randomised method is the fastest.

Lie algebra	dimension	General method	Random	Subs.	Prim. elt.	Dec. elt.
$\mathfrak{sl}_2 \oplus \mathfrak{sl}_2$	6	12	13	13	13	18
$\mathfrak{sl}_2 \oplus \mathfrak{sl}_3$	11	35	24	39	31	33
$\mathfrak{sl}_3 \oplus \mathfrak{sl}_3$	16	127	38	64	113	71
$\mathfrak{sl}_2 \oplus \mathfrak{sl}_4$	18	205	48	69	150	109

Table 4.3: Running times (in seconds) of the algorithms for decomposing a semisimple Lie algebra.

4.5 Acknowledgements

The deterministic algorithms for finding a splitting element saw the light in conversations with G. Ivanyos. The algorithm `PrimitiveSplittingElement` is a variant of the algorithm for finding a splitting element in a maximal torus, given in [31], Section 5.5.

Chapter 5

The type of a semisimple Lie algebra

The human mind has proved to be able to solve the problem of the identity of simple Lie algebras. Let L be a simple Lie algebra over an algebraically closed field F of characteristic 0. Then L is known: it is isomorphic either to an element of one of the “great” classes of simple Lie algebras (A_l, B_l, C_l, D_l) or to one of the exceptional Lie algebras (E_6, E_7, E_8, F_4, G_2).

The semisimple Lie algebras are direct sums of simple ones. Hence also the semisimple Lie algebras are classified. If a given Lie algebra L is isomorphic to e.g. $A_2 + D_5 + G_2$, then we call $A_2 + D_5 + G_2$ the *type* of L .

In Section 5.1 we give a method for obtaining the type of a semisimple Lie algebra with structure constants in \mathbb{Q} . Thereby we solve the isomorphism problem for semisimple Lie algebras over \mathbb{Q} having structure constants in \mathbb{Q} . In Section 5.2 we give a different method for solving this problem. Finally in Section 5.3 examples are given and the two methods are evaluated.

5.1 Identifying a semisimple Lie algebra

For a semisimple Lie algebra we would like to be able to obtain its type. In general however, to calculate the root system and the corresponding Cartan matrix, we need arbitrary number fields. We have an example illustrating this.

Example 5.1 Let L be the 6-dimensional Lie algebra of Example 4.15. Then, as shown in Example 4.15, L is simple. However, over \mathbb{Q} there is only one 6-dimensional semisimple Lie algebra, namely $A_1 + A_1$. In this case, to obtain a splitting of the Cartan subalgebra, we need the field $\mathbb{Q}(\sqrt{-1})$, a field of degree two.

Let H be a Cartan subalgebra of L with splitting element h (see Section 4.2.1). Then the degree of the field extension needed to split the action of $\text{ad } h$ can be very large (if the minimum polynomial of $\text{ad } h$ is irreducible and of degree d , then “generically” the degree of the field extension needed is $d!$). The idea we pursue here is to avoid working over large number fields by reducing the Lie algebra modulo a prime number p . (Note that if we multiply all basis elements by a scalar λ , then the structure constants relative to this new basis are also multiplied by λ , so that we can get all structure constants to be integers.) The algebraic extensions of \mathbb{F}_p are much easier to handle. If p does not divide the determinant of the matrix of the Killing form, then the reduced Lie algebra fits into a similar classification (see [48]). The only thing we have to prove is that both Lie algebras produce equivalent Cartan matrices.

We start with a semisimple Lie algebra K defined over \mathbb{Q} with a basis chosen in such a way that the structure constants relative to this basis lie in \mathbb{Z} . We may assume that this basis contains a basis $\{h_1, \dots, h_l\}$ of a Cartan subalgebra. Let f_i be the characteristic polynomial of $\text{ad } h_i$ for $1 \leq i \leq l$. Write $f_i = X^{m_i} g_i$ where $g_i(0) \neq 0$. If $p \geq 7$ is a prime number not dividing the determinant of the matrix of the Killing form of K and not dividing the numbers $g_i(0)$ for $1 \leq i \leq l$, then p is called *pleasant*. In the sequel we use a fixed pleasant prime number p .

Let F be the smallest number field containing all eigenvalues of the $\text{ad } h_i$ for $1 \leq i \leq l$ and let \mathcal{O}^F be the ring of algebraic integers of F . There exists a prime ideal P of \mathcal{O}^F such that $P \cap \mathbb{Z} = (p)$ (see [34], p. 9). Let

$$\mathcal{O}_P^F = \left\{ \frac{x}{y} \mid x \in \mathcal{O}^F, y \in \mathcal{O}^F \setminus P \right\}$$

be the localisation of \mathcal{O}^F at P . Let M_p be the unique maximal ideal of \mathcal{O}_P^F . It follows that there is an $m > 0$ such that $\mathcal{O}_P^F/M_p = \mathbb{F}_{p^m}$, the finite field of p^m elements.

Now let L be a Lie algebra over \mathcal{O}_P^F with the same multiplication table as K . Set

$$L_p = L \otimes_{\mathcal{O}_P^F} \mathbb{F}_{p^m}.$$

Let $\phi : \mathcal{O}_P^F \rightarrow \mathbb{F}_{p^m}$ be the projection map. In the obvious way ϕ carries over to a map from L to L_p . Let $\{x_1, \dots, x_n\}$ be a basis of L and set $\bar{x}_i = x_i \otimes 1 \in L_p$ for $1 \leq i \leq n$. Then

$$\phi\left(\sum_{i=1}^n a_i x_i\right) = \sum_{i=1}^n \phi(a_i) \bar{x}_i,$$

where $\{\bar{x}_1, \dots, \bar{x}_n\}$ is a basis of L_p . Let κ be the Killing form of L and let κ_p be the Killing form of L_p . The structure constants of L_p are the images under ϕ of the structure constants of L , and hence they lie in the prime field \mathbb{F}_p . From this it follows that

$$\kappa_p(\phi(x), \phi(y)) = \phi(\kappa(x, y)) \text{ for } x, y \in L.$$

Because p is pleasant, we have that κ_p is nondegenerate.

Since the structure constants of L are the same as those of K , also the basis of L corresponding to these structure constants contains a basis of a Cartan subalgebra H . Let H_p be the image under ϕ of H . Furthermore H^* will be the dual space of H (defined over \mathcal{O}_p^F) and H_p^* will be the dual of H_p . The map ϕ induces a map

$$\tilde{\phi} : H^* \longrightarrow H_p^*.$$

For $\lambda : H \rightarrow \mathcal{O}_p^F$ an element of H^* we set $\tilde{\phi}(\lambda)(\phi(h)) = \phi(\lambda(h))$. The image of λ is denoted by $\bar{\lambda}$.

Since \mathcal{O}_p^F contains all eigenvalues of $\text{ad } h_i$ for $1 \leq i \leq l$, the roots exist over \mathcal{O}_p^F . Let $R \subset H^*$ be the set of roots and denote the image of R under $\tilde{\phi}$ in H_p^* by \bar{R} . Then the elements of \bar{R} are the roots of L_p .

Lemma 5.2 *The subalgebra H_p of L_p is a Cartan subalgebra of L_p .*

Proof. Let α be a root of L . The fact that p is pleasant implies that the multiplicity of 0 as a root of f_i is the same as the multiplicity of 0 as a root of \bar{f}_i and hence $\bar{\alpha}$ is nonzero. So if $x_{\bar{\alpha}}$ is a nonzero element of the root space of L_p belonging to $\bar{\alpha}$, then there is an index i such that $[\bar{h}_i, x_{\bar{\alpha}}] = \bar{\alpha}(\bar{h}_i)x_{\bar{\alpha}}$ is nonzero. Hence if $[\bar{h}, \bar{x}] \in H_p$ for an $\bar{x} \in L_p$ and all $\bar{h} \in H_p$, then $\bar{x} \in H_p$, i.e., the normaliser of H_p in L_p is H_p itself. The fact that H_p is nilpotent is a consequence of the nilpotency of H . \square

Lemma 5.3 *The restriction of κ_p to H_p is nondegenerate.*

Proof. Let h be an element of H and \bar{h} its image in H_p . Let $L = H \oplus L_1(H)$ be the Fitting decomposition of L relative to H (see Proposition 1.23). Then from [32], p. 108, it is seen that $\kappa(h, x) = 0$ for all $x \in L_1(H)$. Hence also $\kappa_p(\bar{h}, \bar{x}) = 0$. Because κ_p is nondegenerate there must be a $g \in H$ such that $\kappa_p(\bar{h}, \bar{g})$ is nonzero. \square

It is well known that we can identify H and H^* because the Killing form is nondegenerate. Let λ be an element of H^* . Then the corresponding element $\theta(\lambda)$ of H is required to satisfy $\kappa(\theta(\lambda), h) = \lambda(h)$ for all $h \in H$. If $\{h_1, \dots, h_l\}$ is a basis of H and $\theta(\lambda) = a_1 h_1 + \dots + a_l h_l$, then we have the system of equations

$$(\kappa(h_i, h_j)) \begin{pmatrix} a_1 \\ \vdots \\ a_l \end{pmatrix} = \begin{pmatrix} \lambda(h_1) \\ \vdots \\ \lambda(h_l) \end{pmatrix}. \quad (5.1)$$

By Lemma 5.3 the determinant of the matrix of this system is an integer not divisible by p . Hence by Cramer's rule there exists a unique solution over \mathcal{O}_p^F . Also by Lemma 5.3 we have that in the case of L_p there is a similar map θ_p .

Lemma 5.4 *We have the following identity:*

$$\phi \circ \theta = \theta_p \circ \tilde{\phi}.$$

Proof. Choose $\lambda \in H^*$ and suppose that $\theta_p(\tilde{\phi}(\lambda)) = b_1\bar{h}_1 + \cdots + b_l\bar{h}_l$, where $b_i \in \mathbb{F}_{p^m}$. Then because $\tilde{\phi}(\lambda)(\bar{h}_i) = \phi(\lambda(h_i))$ we have that the system of equations for the b_i is just the image under ϕ of the system of equations (5.1). Hence $b_i = \phi(a_i)$ and we are done. \square

Using the map θ , a bilinear form $(,)$ is defined on H^* by

$$(\lambda, \mu) = \kappa(\theta(\lambda), \theta(\mu)).$$

In the same way there is a bilinear form $(,)_p$ on H_p^* .

Lemma 5.5 *For $\lambda, \mu \in H^*$ we have $\phi((\lambda, \mu)) = (\bar{\lambda}, \bar{\mu})_p$.*

Proof. The proof is by straightforward calculation:

$$\begin{aligned} \phi((\lambda, \mu)) &= \phi(\kappa(\theta(\lambda), \theta(\mu))) \\ &= \kappa_p(\phi(\theta(\lambda)), \phi(\theta(\mu))) \\ &= \kappa_p(\theta_p(\tilde{\phi}(\lambda)), \theta_p(\tilde{\phi}(\mu))) \text{ by Lemma 5.4} \\ &= (\tilde{\phi}(\lambda), \tilde{\phi}(\mu))_p. \end{aligned}$$

\square

Let α and β be two roots; then we set

$$\frac{2(\alpha, \beta)}{(\beta, \beta)} = \langle \alpha, \beta \rangle.$$

For the modular case we have a similar formula

$$\frac{2(\bar{\alpha}, \bar{\beta})_p}{(\bar{\beta}, \bar{\beta})_p} = \langle \bar{\alpha}, \bar{\beta} \rangle_p.$$

We remark that since $p \geq 5$, Theorem 5.6 of [47] implies that $(\bar{\beta}, \bar{\beta})_p$ is nonzero.

We call a set of roots $\{\alpha_1, \dots, \alpha_l\}$ a fundamental system if for any root α one of the following holds:

1. α is a member of a sequence of the form $\alpha_{i_1}, \alpha_{i_1} + \alpha_{i_2}, \alpha_{i_1} + \alpha_{i_2} + \alpha_{i_3} \dots$
2. $-\alpha$ is a member of such a sequence.

We note that every root system has a fundamental system (see [29], [48]). If $\{\alpha_1, \dots, \alpha_l\}$ is a fundamental system, then $C = \langle \alpha_i, \alpha_j \rangle$ (respectively $\bar{C} = \langle \bar{\alpha}_i, \bar{\alpha}_j \rangle_p$ in the modular case) is the corresponding Cartan matrix.

Proposition 5.6 *Let C be a Cartan matrix of R , then $\phi(C)$ is a Cartan matrix of \bar{R} .*

Proof. Let $\{\alpha_1, \dots, \alpha_l\}$ be a fundamental system of roots in H^* . Then it is immediate that $\{\bar{\alpha}_1, \dots, \bar{\alpha}_l\}$ is a fundamental system of roots in H_p^* . Hence Lemma 5.5 implies that

$$\langle \bar{\alpha}_i, \bar{\alpha}_j \rangle_p = \phi(\langle \alpha_i, \alpha_j \rangle).$$

□

Corollary 5.7 *From a Cartan matrix of \bar{R} we can recover a Cartan matrix of R .*

Proof. The numbers $\langle \alpha_i, \alpha_j \rangle$ are known to be 2, 0, -1, -2 or -3 ([32], p. 121). Because $p \geq 7$, we can recover those numbers from their images in \mathbb{F}_p . Now Proposition 5.6 finishes the proof. □

The above results lead to the following algorithm:

Algorithm Type

Input: A semisimple Lie algebra L over \mathbb{Q}

Output: The type of L .

- Step 1 Calculate a Cartan subalgebra H of L (Chapter 3).
- Step 2 Extend a basis of H to a basis of L and multiply by an integer in order to ensure that all structure constants relative to this basis are integers.
- Step 3 Select a pleasant prime p .
- Step 4 Let S be the table of structure constants obtained from the table of structure constants of L by reducing every constant modulo p . Let L_p be the Lie algebra with structure constants table S , defined over \mathbb{F}_{p^m} where m is large enough to ensure that the characteristic polynomials of $\text{ad } h_i$ for h_i in a basis of H_p split into linear factors.
- Step 5 Decompose L_p into a direct sum of simple ideals (Chapter 4).
- Step 6 For each component of L_p , determine a fundamental system inside the root system. Calculate the Cartan matrices which determine the type of L .

Remark. The integer m in Step 4 will be the least common multiple of the degrees of the irreducible factors of the minimum polynomials of $\text{ad } h_i$, where h_i runs over a basis of H_p . Also, if L_p has a splitting element, then m will be the least common multiple of the degrees of the irreducible factors of the minimum polynomial of a splitting element. We have no proof that this number is polynomial in the dimension of L , so that we do not know the complexity of this algorithm.

Remark. The number -3 will occur in the Cartan matrix only if there is a simple factor isomorphic to G_2 . So if the semisimple Lie algebra L has no ideals of type G_2 , then we can take $p \geq 5$. On the other hand, if a simple ideal I of L is of type G_2 , then we can recognise it by inspection of the dimension and the rank. The conclusion is that we can always take $p \geq 5$.

Remark. The only cases where we need to calculate the root system of a simple component K of L is where K is isomorphic to B_l , C_l or E_6 . The Lie algebras B_l and C_l have the same dimension and rank and the dimension and rank of E_6 are equal to those of B_6 and C_6 . The other simple Lie algebras reveal their type by their dimension and rank.

Remark. For the numbers $\langle \alpha, \beta \rangle$ appearing in the Cartan matrix there is a well-known formula. Let s and t be the largest non-negative integers such that

$$\alpha - s\beta, \alpha - (s-1)\beta, \dots, \alpha + (t-1)\beta, \alpha + t\beta$$

are all roots. Then $\langle \alpha, \beta \rangle = s - t$ ([29], p. 45, see Theorem 5.6 of [47] for the modular case). This gives an easy and fast algorithm to calculate these numbers.

5.2 Isomorphism of semisimple Lie algebras

Here we present an algorithmic method to decide whether two semisimple Lie algebras are isomorphic. Let L be a semisimple Lie algebra of dimension n with Cartan subalgebra H . Suppose $\{h_1, \dots, h_l\}$ is a basis of H . Let x_1, \dots, x_l be indeterminates and set $h = \sum x_i h_i$ which is an element of $L \otimes F(x_1, \dots, x_l)$. Let

$$f(T) = T^n + p_1(x_1, \dots, x_l)T^{n-1} + \dots + p_0(x_1, \dots, x_l)$$

be the characteristic polynomial of $\text{ad } h$. Then we call f the *characteristic polynomial of the action of H on L* .

Theorem 5.8 *Let L_1 and L_2 be semisimple Lie algebras over an algebraically closed field F of characteristic 0. Let H_1 and H_2 be Cartan subalgebras of L_1 and L_2 , respectively.*

Suppose $\dim H_1 = \dim H_2 = l$ and $\dim L_1 = \dim L_2 = n$. Let

$$f_1(T) = T^n + p_1(x_1, \dots, x_l)T^{n-1} + \dots + p_0(x_1, \dots, x_l)$$

and

$$f_2(T) = T^n + q_1(y_1, \dots, y_l)T^{n-1} + \dots + q_0(y_1, \dots, y_l)$$

be the characteristic polynomials of the action of H_1 (on L_1) and H_2 (on L_2), respectively. Then L_1 and L_2 are isomorphic if and only if there is a transformation

$$\begin{cases} \bar{y}_1 &= a_{11}x_1 + a_{12}x_2 + \dots + a_{1l}x_l \\ \vdots & \\ \bar{y}_l &= a_{l1}x_1 + a_{l2}x_2 + \dots + a_{ll}x_l \end{cases} \quad (5.2)$$

such that $\det(a_{ij}) \neq 0$ and $p_i(x_1, \dots, x_l) = q_i(\bar{y}_1, \dots, \bar{y}_l)$ for $1 \leq i \leq l$.

Proof. Write $L_1 = V_1 \oplus H_1$ and $L_2 = V_2 \oplus H_2$ (direct sums of vector spaces), where the subspaces V_1 and V_2 are the sums of the root spaces of L_1 and L_2 , respectively. Suppose that L_1 and L_2 are isomorphic. Because all Cartan subalgebras are conjugate (Theorem IX.3 of [32]), we may assume that an isomorphism $L_1 \rightarrow L_2$ maps H_1 to H_2 . Let $\alpha_1, \dots, \alpha_r$ be the roots of L_1 and let $\{h_1, \dots, h_l\}$ be a basis of H_1 . Then

$$f_1(T) = T^l \prod_{i=1}^r (T - \alpha_i(h_1)x_1 - \dots - \alpha_i(h_l)x_l).$$

A base change of V_1 does not affect the characteristic polynomial of the action of H_1 . Hence we consider the effect of a base change in H_1 on the polynomial f_1 . Suppose $\{\bar{h}_1, \dots, \bar{h}_l\}$ is a second basis of H_1 , where $\bar{h}_i = \sum a_{ki}h_k$. Then

$$\begin{aligned} T - \alpha_i(\bar{h}_1)x_1 - \dots - \alpha_i(\bar{h}_l)x_l &= T - \alpha_i\left(\sum_{k=1}^l a_{k1}h_k\right)x_1 - \dots - \alpha_i\left(\sum_{k=1}^l a_{kl}h_k\right)x_l \\ &= T - \alpha_i(h_1) \sum_{k=1}^l a_{1k}x_k - \dots - \alpha_i(h_l) \sum_{k=1}^l a_{lk}x_k \\ &= T - \alpha_i(h_1)\bar{y}_1 - \dots - \alpha_i(h_l)\bar{y}_l. \end{aligned}$$

The conclusion is that a base change of H_1 corresponds exactly to a change of variables in the polynomials p_i . So $L_1 \cong L_2$ implies that there is a transformation of the form (5.2).

Now suppose that there is a transformation of the form (5.2). Let $\{\bar{h}_1, \dots, \bar{h}_l\}$ be a basis of H_2 . We define a linear map $\phi: H_1 \rightarrow H_2$ by

$$\phi(h_i) = \sum_{j=1}^l a_{ji}\bar{h}_j.$$

We extend this map to the dual spaces by $\phi(\alpha)(\phi(h)) = \alpha(h)$ for $\alpha \in H_2^*$ and $h \in H_1$. We claim that if α is a root of L_1 , then $\phi(\alpha)$ is a root of L_2 . To see this, note that a root β of L_2 corresponds to a factor

$$T - \beta(\bar{h}_1)y_1 - \cdots - \beta(\bar{h}_l)y_l$$

in f_2 . Now choose β such that by the transformation (5.2) this factor is mapped to

$$T - \alpha(h_1)x_1 - \cdots - \alpha(h_l)x_l.$$

Then we calculate

$$\begin{aligned} T - \beta(\bar{h}_1)\bar{y}_1 - \cdots - \beta(\bar{h}_l)\bar{y}_l &= T - \beta(\bar{h}_1) \sum_{j=1}^l a_{1j}x_j - \cdots - \beta(\bar{h}_l) \sum_{j=1}^l a_{lj}x_j \\ &= T - \left(\sum_{j=1}^l a_{j1}\beta(\bar{h}_j) \right) x_1 - \cdots - \left(\sum_{j=1}^l a_{jl}\beta(\bar{h}_j) \right) x_l. \end{aligned}$$

It follows that

$$\alpha(h_i) = \sum_{j=1}^l a_{ji}\beta(\bar{h}_j) = \beta\left(\sum_{j=1}^l a_{ji}\bar{h}_j\right) = \beta(\phi(h_i)).$$

So on a basis of H_2 the functions $\phi(\alpha)$ and β have the same values, forcing $\phi(\alpha) = \beta$. The conclusion is that L_1 and L_2 have isomorphic root systems and hence $L_1 \cong L_2$ (see [29], Theorem 14.2). \square

The algorithm resulting from this is the following:

Algorithm: IsIsomorphic

Input: Two semisimple Lie algebras L_1 and L_2 .

Output: The boolean $L_1 \cong L_2$.

Step 1 If $\dim L_1 \neq \dim L_2$ then return false. Otherwise set $n = \dim L_1$.

Step 1 Calculate Cartan subalgebras H_1 and H_2 of L_1 and L_2 (Chapter 3). If $\dim H_1 \neq \dim H_2$ then return false. Otherwise set $l = \dim H_1$.

Step 2 Calculate the polynomials p_i and q_i (for $0 \leq i \leq n-1$).

Step 3 Introduce the variables a_{jk} for $1 \leq j, k \leq l$ and substitute $\bar{y}_j = \sum a_{jk}x_k$ in the p_i . Require that the resulting polynomials are equal to the q_i . This yields a system of polynomial equations in the variables a_{jk} .

Step 4 Now by a Gröbner basis computation we can check whether there is a solution to the system of equations we obtained in the preceding step.

5.3 Examples

Example 5.9 Let L_1 be the Lie algebra of Example 4.14. Then the matrix of the restriction of $\zeta_1 \text{ ad } h_1 + \zeta_2 \text{ ad } h_2$ to the space spanned by $\{x_1, y_1, x_2, y_2\}$ is

$$\begin{pmatrix} 2\zeta_1 + 2\zeta_2 & 0 & 0 & 0 \\ 0 & -2\zeta_1 - 2\zeta_2 & 0 & 0 \\ 0 & 0 & 2\zeta_1 - 2\zeta_2 & 0 \\ 0 & 0 & 0 & -2\zeta_1 + 2\zeta_2 \end{pmatrix}.$$

It follows that the characteristic polynomial of the action of H_1 on L_1 is

$$\begin{aligned} f_1(T) &= T^2(T - 2\zeta_1 - 2\zeta_2)(T + 2\zeta_1 + 2\zeta_2)(T - 2\zeta_1 + 2\zeta_2)(T + 2\zeta_1 - 2\zeta_2) \\ &= T^6 + (-8\zeta_1^2 - 8\zeta_2^2)T^4 + (16\zeta_1^4 - 32\zeta_1^2\zeta_2^2 + 16\zeta_2^4)T^2. \end{aligned}$$

Let L_2 be the Lie algebra of Example 4.15. Then the matrix of the restriction of $\xi_1 \text{ ad } x_1 + \xi_2 \text{ ad } x_2$ to the span of $\{x_3, x_4, x_5, x_6\}$ is

$$\begin{pmatrix} 2\xi_2 & -2\xi_1 & 0 & 0 \\ 2\xi_1 & 2\xi_2 & 0 & 0 \\ 0 & 0 & -2\xi_2 & 2\xi_1 \\ 0 & 0 & -2\xi_1 & -2\xi_2 \end{pmatrix}.$$

Hence the characteristic polynomial of the action of the Cartan subalgebra H_2 of L_2 is

$$\begin{aligned} f_2(T) &= T^2(T^2 - 4\xi_2T + 4\xi_1^2 + 4\xi_2^2)(T^2 + 4\xi_2T + 4\xi_1^2 + 4\xi_2^2) \\ &= T^6 + (8\xi_1^2 - 8\xi_2^2)T^4 + (16\xi_1^4 + 32\xi_1^2\xi_2^2 + 16\xi_2^4)T^2 \end{aligned}$$

It is easily seen that the transformation $\zeta_1 = i\xi_1, \zeta_2 = \xi_2$ transports f_1 to f_2 . The conclusion is that L_1 and L_2 are isomorphic.

Now we try to calculate the type of L_2 . The characteristic polynomials of $\text{ad } x_1$ and $\text{ad } x_2$ split over \mathbb{F}_5 . Over this field L_2 splits as a direct sum of two ideals I_1 and I_2 , where

$$I_1 = \langle x_5 + 2x_6, x_3 + 2x_4, x_1 + 3x_2 \rangle,$$

and

$$I_2 = \langle x_3 + 3x_4, x_5 + 3x_6, x_1 + 2x_2 \rangle.$$

(The elements of \mathbb{F}_5 are denoted by $\{0, 1, 2, 3, 4\}$.) Using the multiplication table of Example 4.15, we can calculate the multiplication tables of I_1 and I_2 . Then these ideals turn out to be isomorphic to \mathfrak{sl}_2 . The Lie algebra L_1 splits over any field. Also this Lie algebra is isomorphic to a direct sum of two copies of \mathfrak{sl}_2 . The details are left to the reader.

In order to test the algorithm Type, we first construct semisimple Lie algebras over \mathbb{Q} that do not split over \mathbb{Q} (i.e., they are simple over \mathbb{Q}). Let L be an absolutely simple Lie

algebra (i.e., L does not split over any algebraic extension of \mathbb{Q}), with basis $\{x_1, \dots, x_n\}$ and structure constants (c_{ij}^k) . Let $f \in \mathbb{Q}[x]$ be an irreducible monic polynomial of degree d and let α be a root of f . Then $L(f)$ will be the Lie algebra over \mathbb{Q} with basis

$$x_1, \dots, x_n, \alpha x_1, \dots, \alpha x_n, \alpha^2 x_1, \dots, \alpha^2 x_n, \dots, \alpha^{d-1} x_1, \dots, \alpha^{d-1} x_n.$$

The multiplication of $L(f)$ is specified by

$$[\alpha^k x_i, \alpha^l x_j] = \sum_{m=1}^n c_{ij}^k \alpha^{k+l} x_m.$$

Over an algebraic extension of \mathbb{Q} this Lie algebra will split as a direct sum of d ideals, all of the same type as L . However, $L(f)$ is simple over \mathbb{Q} . We calculated the type of $L(f)$ for various polynomials f and two simple Lie algebras $L = A_1$ and $L = B_2$. The results are displayed in Table 5.1.

f	splitting field	$\dim A_1(f)$	Type $(A_1(f))$	$\dim B_2(f)$	Type $(B_2(f))$
$x^3 - x + 1$	\mathbb{F}_5	9	23	30	339
$x^4 + 3x + 1$	\mathbb{F}_3	12	38	40	616
$x^5 - 4x + 2$	\mathbb{F}_2	15	66	50	1388
$x^6 + 4x + 1$	\mathbb{F}_3	18	94	60	2391
$x^7 + 8x + 1$	\mathbb{F}_7	21	140	70	3874
$x^8 + 5x + 1$	\mathbb{F}_4	24	194	80	5549

Table 5.1: Running times (in seconds) of the algorithm `Type`, with $A_1(f)$ (fourth column) and $B_2(f)$ (sixth column) as input. The second column displays the field that was used to split the Cartan subalgebra.

We consider the system of polynomial equations that arises in Step 3 of the algorithm `IsIsomorphic`. The number of variables is $(\dim H)^2$. Hence the number of variables increases rapidly if $\dim H$ increases. When testing the isomorphism of $\mathfrak{sl}_2 \oplus \mathfrak{sl}_2 \oplus \mathfrak{sl}_2$ and $A_1(x^3 - x + 1)$, we got 49 polynomial equations in 9 variables. And already the Gröbner basis computation became problematic. Also the calculation of the characteristic polynomial of the action of a Cartan subalgebra is problematic. Maple used 92.6 seconds for this computation in the case of $A_1(x^5 - 4x + 2)$ and it was not able to do the computation in the higher dimensional cases. The conclusion is that this isomorphism test is not well suited for practical problems.

5.4 Acknowledgements

Theorem 5.8 finds its origin in an idea by H. Kraft (personal communication). The method of constructing nonsplit semisimple Lie algebras (described at the end of the last section)

was communicated to the author by G. Ivanyos. The algorithm **Type** was also given in [23].

Chapter 6

Constructive Ado

In this chapter we consider the problem of constructing a faithful matrix representation of a Lie algebra given by a table of structure constants. According to Ado's theorem this is always possible. However, the standard proofs of this theorem (see [6], [32]) do not provide effective constructions.

Throughout we suppose that L is a finite-dimensional Lie algebra of characteristic 0. A first idea is to look at the adjoint representation of L . The kernel of ad is the centre $Z(L)$ of L . So for Lie algebras with a trivial centre the problem is solved by the adjoint representation. The rest of this chapter will be concerned with Lie algebras that have a nontrivial centre.

The method that we describe here follows roughly the lines of the proof of Ado's theorem given in [6]. In this proof a tower (with certain properties) of Lie algebra extensions (where every term is an ideal in the next one) is constructed with final term L . An algorithm for computing such a tower is given in Section 6.1. A representation of the first element of the tower is easily constructed. Then this representation is successively extended to representations of the higher terms of the tower and finally to L itself. Sections 6.2 and 6.3 focus on a single extension step. In Section 6.2 the vector space underlying the extension is described. We take a significantly smaller space than is done in the proof in [6]. Then in Section 6.3 it is proved that it is sufficient to work with this smaller space. An algorithm for calculating the extension is given. In Section 6.4 an algorithm for the construction of a faithful finite-dimensional representation of L is given and Ado's theorem is obtained as a corollary. Also an upper bound on the degree of the resulting representation is given in the case where L is nilpotent. Finally, in Section 6.5 some practical examples are discussed.

6.1 Calculating a series of extensions

Here we describe how a series of subalgebras $K_1 \subset K_2 \subset \dots \subset K_m = L$ can be constructed such that $K_{i+1} = K_i \rtimes H_i$, where H_i is a subalgebra of K_{i+1} . The following algorithm calculates such a series.

Algorithm ExtensionSeries

Input: A Lie algebra L over a field of characteristic 0.

Output: Series $K_1, K_2, \dots, K_r = L$ and H_1, \dots, H_{r-1} such that

1. K_1 is commutative,
2. $K_{i+1} = K_i \rtimes H_i$,
3. $[H_i, K_i] \subset \text{NR}(K_i)$ for $1 \leq i \leq r-1$,
4. $\dim H_i = 1$ for $1 \leq i < r-1$.

$R := \text{SolvableRadical}(L)$;

$K_1 :=$ the final term of the derived series of R ;

$i := 1$;

while $K_i \neq R$ **do**

$I :=$ the unique term of the derived series of R such that $[I, I] \subset K_i$,
but I is not contained in K_i ;

$y :=$ an element from $I \setminus K_i$;

$K_{i+1} := K_i \rtimes \langle y \rangle$;

$H_i := \langle y \rangle$;

$i := i + 1$;

od;

$r := i + 1$;

$K_r := L$; $H_{r-1} := \text{LeviSubalgebra}(L)$;

Proof. First we consider the computability of all the steps. Algorithms for the computation of the solvable radical, of the derived series and of a Levi subalgebra are described in Chapter 1. So it is easily seen that the algorithm terminates.

Now we prove that the output satisfies the properties listed above. In the first part of the algorithm a series of subalgebras

$$0 \subset K_1 \subset K_2 \subset \dots \subset K_{r-1} = R$$

is constructed such that $K_{i+1} = K_i \rtimes \langle y_i \rangle$. From the choice of y_i it is seen that K_i is indeed an ideal in K_{i+1} . For $1 \leq i < r-1$ we let H_i be the 1-dimensional subalgebra spanned by y_i . At the end we let H_{r-1} be a Levi subalgebra and we set $K_r = L$.

The first two properties of the output are immediate. We have that $[L, R] \subset \text{NR}(L)$ (Theorem II.13 of [32]) and $K_i \subset R$ for $1 \leq i \leq r-1$. Hence $[H_i, K_i] \subset \text{NR}(L) \cap K_i \subset \text{NR}(K_i)$. The last inclusion follows from the fact that $\text{ad}_{K_i} x$ is nilpotent for all $x \in \text{NR}(L) \cap K_i$.

Finally from the construction above it is seen that $\dim H_i = 1$ for $1 \leq i < r-1$. \square

6.2 The extension space

Here we consider the situation where $L = K \rtimes H$. Starting with a finite-dimensional representation $\rho : K \rightarrow \mathfrak{gl}(V)$ of K we try to find a finite dimensional representation σ of L . Under some conditions we succeed in doing this.

First we describe the space on which L is to be represented. By $U(K)$ we will denote the universal enveloping algebra of K . If $\{x_1, \dots, x_t\}$ is a basis of K , then by the Poincaré-Birkhoff-Witt (PBW) theorem ([32], Theorem 5.3) a basis of $U(K)$ (called PBW-basis) is formed by the *standard monomials* $x_1^{k_1} \cdots x_t^{k_t}$.

The representation space of L will be a finite-dimensional subspace of the dual space $U(K)^*$. First we describe how L acts on $U(K)^*$. Let f be an element of $U(K)^*$ and let $x \in K$ and $y \in H$. Then for $a \in U(K)$ we set

$$\begin{aligned}(x \cdot f)(a) &= f(ax) \\ (y \cdot f)(a) &= -f(ya - ay).\end{aligned}$$

Note that for $a \in U(K)$ we have that $ya - ay$ (which is an element of $U(L)$) also lies in $U(K)$. By some simple calculations it can be shown that this is indeed a Lie algebra action ([6], §7.2).

We extend the representation ρ of K to a representation of the universal enveloping algebra $U(K)$, by

$$\rho(x_1^{k_1} \cdots x_t^{k_t}) = \rho(x_1)^{k_1} \cdots \rho(x_t)^{k_t}.$$

Consider the map

$$\theta : V \times V^* \longrightarrow U(K)^*$$

defined by $\theta(v, w^*)(a) = w^*(\rho(a)v)$. An element $\theta(v, w^*)$ is called a *coefficient* of the representation ρ . By C_ρ we denote the image of θ in $U(K)^*$. For the proof of the following lemma we refer to [6], §7.1.

Lemma 6.1 C_ρ is a K -submodule in $U(K)^*$.

Let $S_\rho \subset U(K)^*$ be the L -submodule of $U(K)^*$ generated by C_ρ . Let $\sigma : L \rightarrow \mathfrak{gl}(S_\rho)$ be the corresponding representation. In [6] the direct sum of n copies of σ is taken as extension

of ρ , where $n = \dim V$. Then it is proved that this representation contains a copy of ρ . This is not guaranteed to hold for σ . However, by a slight abuse of language we will call σ the *extension* of ρ to L .

The next proposition states some conditions under which S_ρ is finite-dimensional.

Proposition 6.2 *Let $L = K \rtimes H$ be such that $[H, K] \subset \text{NR}(K)$ and let $\rho : K \rightarrow \mathfrak{gl}(V)$ be a finite-dimensional representation of K such that $\rho(x)$ is nilpotent for all $x \in \text{NR}(K)$. Let $\sigma : L \rightarrow \mathfrak{gl}(S_\rho)$ be the extension of ρ to L . Then we have that S_ρ is finite-dimensional and $\sigma(x)$ is nilpotent for all $x \in \text{NR}(L)$.*

Proof. The proof of these facts can be found in the proof of Theorem 1 of [6] §7.2. \square

6.3 Extending a representation

Here we show how a faithful finite-dimensional representation of a Lie algebra L can be constructed using the extension described in the previous section.

Throughout this section $L = K \rtimes H$ and $\rho : K \rightarrow \mathfrak{gl}(V)$ is a finite-dimensional representation of K . Furthermore, $\sigma : L \rightarrow \mathfrak{gl}(S_\rho)$ will be the extension of ρ to L .

The key to the algorithm will be the following proposition.

Proposition 6.3 *Suppose that ρ is a faithful representation of K . Then σ is faithful on K . Furthermore, if H is 1-dimensional, then σ is a faithful representation of L or there is an element $\tilde{y} \in K$ such that $y - \tilde{y} \in Z(L)$, where y is an element spanning H .*

Proof. Let x be a nonzero element of K . Then $\rho(x) \neq 0$ and hence there are $v \in V$ and $w^* \in V^*$ such that

$$0 \neq w^*(\rho(x)v) = \sigma(x) \cdot \theta(v, w^*)(1).$$

Hence $\sigma(x) \neq 0$ for all $x \in K$ so that σ is faithful on K .

Suppose that $H = \langle y \rangle$. Suppose further that σ is not faithful on L . This means that there is a nontrivial relation

$$\lambda\sigma(y) - \sigma(\tilde{y}) = 0,$$

where $\tilde{y} \in K$. Because σ is faithful on K , we may assume that $\lambda = 1$. It follows that $\sigma(y) = \sigma(\tilde{y})$. Then for all $x \in K$ we have

$$\sigma([\tilde{y}, x]) = [\sigma(\tilde{y}), \sigma(x)] = [\sigma(y), \sigma(x)] = \sigma([y, x]).$$

Since σ is faithful on K , this implies that $[\tilde{y}, x] = [y, x]$. Also $\sigma([y, y - \tilde{y}]) = 0$ and because $[y, y - \tilde{y}] \in K$ we have that it is 0. The conclusion is that $y - \tilde{y} \in Z(L)$. \square

Now we continue with some observations about the space S_ρ . In the sequel $\{y_1, \dots, y_s\}$ will be a basis of H , and $\{v_1, \dots, v_n\}$ will be a basis of V . By e_{ij}^n we denote the $n \times n$ matrix with a 1 on position (i, j) and zeros elsewhere.

Lemma 6.4 *Suppose that $\rho(x)v_1 = 0$ for all elements $x \in K$. Then there is a basis $\{w_1, \dots, w_m\}$ of S_ρ such that $\sigma(x)w_1 = 0$ for all $x \in L$.*

Proof. We work with the customary dual basis $\{v_1^*, \dots, v_n^*\}$ of V^* (i.e., $v_i^*(v_j) = \delta_{ij}$). Set $w_1 = \theta(v_1, v_1^*)$. Let a be a monomial in $U(K)$. We calculate $w_1(a) = \theta(v_1, v_1^*)(a) = v_1^*(\rho(a)v_1)$; it is 0 if $a \neq 1$ and 1 if $a = 1$. It follows that w_1 takes the value 1 on the element 1 of $U(K)$ and 0 on all other monomials. In particular w_1 is nonzero. Now we extend w_1 to a basis w_1, \dots, w_m of S_ρ . If x is an element of K then $\sigma(x)w_1(a) = w_1(ax)$. The support of ax does not contain a constant term, hence $w_1(ax) = 0$. Now let $x \in H$. Then $\sigma(x)w_1(a) = -w_1(xa - ax)$. Since the support of $xa - ax$ also does not contain a constant term, we have that $w_1(xa - ax) = 0$. It follows that $\sigma(x)w_1 = 0$ for all $x \in L$. \square

Lemma 6.5 *The space S_ρ is spanned by the elements*

$$y_1^{k_1} \cdots y_s^{k_s} \cdot \theta(v_i, v_j^*)$$

where $k_q \geq 0$ ($1 \leq q \leq s$) and $1 \leq i, j \leq n$.

Proof. Let $\{x_1, \dots, x_t\}$ be a basis of K . Then by the PBW theorem S_ρ is spanned by all elements of the form

$$y_1^{k_1} \cdots y_s^{k_s} \cdot x_1^{l_1} \cdots x_t^{l_t} \cdot \theta(v_i, v_j^*).$$

But since C_ρ is a $U(K)$ -module (Lemma 6.1), we have that such an element is a linear combination of elements of the form

$$y_1^{k_1} \cdots y_s^{k_s} \cdot \theta(v_k, v_l^*).$$

\square

Let a be an element of $U(K)$. By $\text{Orb}_H(a)$ we denote the orbit of a under the action of the elements of H , i.e.,

$$\text{Orb}_H(a) = \langle y_1^{k_1} \cdots y_s^{k_s} \cdot a \mid k_1, \dots, k_s \geq 0 \rangle,$$

where $y_i \cdot a = y_i a - a y_i$.

Lemma 6.6 *Let $f \in S_\rho$. If a is an element of $U(K)$ such that $\rho(\text{Orb}_H(a)) = 0$, then $f(a) = 0$.*

Proof. Set $g = y_1^{k_1} \cdots y_s^{k_s} \cdot \theta(v_i, v_j^*)$. Then

$$g(a) = \pm v_j^*(\rho(y_s^{k_s} \cdots y_1^{k_1} \cdot a)v_i) = 0.$$

Now Lemma 6.5 implies that f is a linear combination of elements of this form (note that since H is a subalgebra we have that a monomial $y_s^{k_s} \cdots y_1^{k_1}$ is a linear combination of monomials of the form $y_1^{m_1} \cdots y_s^{m_s}$). It follows that $f(a) = 0$. \square

Remark. Let a be an element of $U(K)$ of degree d . Let W be the span of all monomials in $U(K)$ of degree $\leq d$. Then $\text{Orb}_H(a) \subset W$. The conclusion is that $\text{Orb}_H(a)$ is finite dimensional. By viewing it as a subspace of W , we can calculate a basis of $\text{Orb}_H(a)$.

Now we formulate an algorithm for extending the representation ρ to L . There are two cases to be considered; the general case and the case where $H = \langle y \rangle$ and there is a $\tilde{y} \in K$ such that $y - \tilde{y} \in Z(L)$. In the second case we can easily construct a faithful representation of L . Then by Proposition 6.3 we always obtain a faithful representation of L in the case where H is 1-dimensional. For greater clarity we formulate the algorithm using a subroutine that treats the general case. We first state the subroutine.

We consider the problem of representing a function in $U(K)^*$ by a vector. In the algorithm this is done by taking a finite set of monomials (called V_d) and giving the values of the function on that set. This enables us to represent every element of S_ρ by a vector of finite length so that we can use linear algebra to calculate a basis and the coefficients of an element with respect to that basis.

Algorithm GeneralExtension

Input: $L = K \rtimes H$ and $\rho : K \rightarrow \mathfrak{gl}(V)$.

Output: The extension $\sigma : L \rightarrow \mathfrak{gl}(S_\rho)$.

- Step 1 Calculate a set of standard monomials $\{m_1, \dots, m_r\}$ that form a basis of a complement to $\ker \rho$ in $U(K)$.
- Step 2 Calculate a basis of C_ρ .
- Step 3 $d := \max \deg m_i$;
- Step 4 $V_d := \{a \in U(K) \mid a \text{ is a monomial of degree } \leq d \text{ such that } \rho(\text{Orb}_H(a)) \neq 0\}$;
- Step 5 Calculate a basis of S_ρ (relative to V_d), and let the first basis element be $\theta(v_1, v_1^*)$.
- Step 6 Calculate the action of the elements of a basis of L on S_ρ . If this yields a representation of L , then return that representation. Otherwise set $d := d + 1$; and go to Step 4.

The algorithm is straightforward. It calculates a basis of S_ρ and the matrices of the corresponding representation. Most of the steps are concerned with finding an appropriate set V_d of monomials relative to which we can represent all elements of S_ρ . First we consider the space C_ρ . We have that

$$\theta(v_i, v_j^*)(a) = v_j^*(\rho(a)v_i),$$

so that we can describe a function in C_ρ by giving its values on the monomials m_i constructed in Step 1. Now we let V_d be a subset of the set of all monomials of degree $\leq d$. So initially we set d equal to the maximum degree of a monomial m_i , ensuring that all these elements will be contained in V_d . By Lemma 6.6 we may discard all monomials a such that $\rho(\text{Orb}_H(a)) = 0$. Using Lemma 6.5 we calculate a basis of S_ρ , representing each function on the set V_d . Then we calculate the matrices of the action of the elements of a basis of L . If this yields a representation of L then we are done. Otherwise we apparently did not calculate all of S_ρ in the preceding step. This means that there are functions in S_ρ that cannot be described by giving their values on only the monomials in V_d . So in this case we set $d := d + 1$ and go through the process again. Since S_ρ is finite dimensional, the procedure will terminate.

Now we state the routine that also treats the special case.

Algorithm ExtendRepresentation

Input: $L = K \rtimes H$ and $\rho : K \rightarrow \mathfrak{gl}(V)$ such that $\rho(x) \cdot v_1 = 0$ for $x \in K$.

Output: An extension $\sigma : L \rightarrow \mathfrak{gl}(W)$.

if $H = \langle y \rangle$ and there is a $\tilde{y} \in K$ such that

$$y - \tilde{y} \in Z(L)$$

then

$$n := \dim V;$$

$$\sigma(y - \tilde{y}) := e_{1, n+1}^{n+1};$$

for x in a basis of K **do**

$\sigma(x) :=$ the $n + 1 \times n + 1$ matrix of which the $n \times n$ submatrix
in the top left corner is $\rho(x)$ and the other positions are 0;

od;

else $\sigma := \text{GeneralExtension}(L, \rho)$;

fi;

Proof. First we remark that finding a \tilde{y} such that $y - \tilde{y} \in Z(L)$ amounts to solving a system of linear equations.

We have to prove that the map σ constructed in the first part of the algorithm is a representation of L . Since $\rho(x) \cdot v_1 = 0$ for all $x \in K$, we have that the first column of the matrix $\rho(x)$ is zero. Hence $\sigma(y - \tilde{y})$ commutes with $\rho(x)$ for $x \in K$. \square

6.4 An effective version of Ado's theorem

Using the routines `ExtensionSeries` and `ExtendRepresentation`, we formulate an algorithm for calculating a finite-dimensional faithful representation of an arbitrary Lie algebra of characteristic 0.

Algorithm Representation

Input: A Lie algebra L .

Output: A finite-dimensional faithful representation σ of L .

```

[ $K_1, \dots, K_r, H_1, \dots, H_{r-1}$ ] := ExtensionSeries( $L$ );
 $\rho_1(x_i) := e_{1,i+1}^{s+1}$ ;
(Where  $\{x_1, \dots, x_s\}$  is a basis of  $K_1$ )
 $i := 2$ ;
while  $i \leq r - 1$  do
   $\rho_i := \text{ExtendRepresentation}(\rho_{i-1}, K_i)$ ;
   $i := i + 1$ ;
od;
if  $H_{r-1} \neq 0$  then
   $\rho_r := \text{ExtendRepresentation}(\rho_{r-1}, L)$ ;
   $\sigma := \text{DirectSum}(\rho_r, \text{ad})$ ;
else
   $\sigma := \rho_{r-1}$ ;
fi;

```

Proof. It is a trivial fact that the algorithm terminates. We prove that it outputs a faithful representation of L .

The algorithm starts with a representation of the commutative subalgebra K_1 . We remark that $\rho_1(x)$ is nilpotent for all $x \in \text{NR}(K_1) = K_1$. Then we successively construct representations ρ_i of K_i . By Lemma 6.4 an invariant of the process is that $\rho_i(x)v_1 = 0$ for $x \in K_i$. Hence we have the correct input for the subroutine `ExtendRepresentation`. Also by Proposition 6.2 and property 3 of the output of `ExtensionSeries` we have that ρ_i is always nilpotent on the nilradical of K_i so that S_{ρ_i} will be finite-dimensional. By Proposition 6.3 and property 4 of the output of `ExtensionSeries` we have that each time `ExtendRepresentation` will return a faithful representation.

The last step is the extension to the Lie algebra $R \rtimes S$, where R is the solvable radical of L and S is a Levi subalgebra. This time after having called `ExtendRepresentation` there is no guarantee that the resulting representation will be faithful. However it will be faithful on R and consequently on the centre of L . Then we take the direct sum with the adjoint representation obtaining a representation that is faithful on the centre as well as on the rest of L . \square

Corollary 6.7 (Ado's theorem) *Let L be a finite-dimensional Lie algebra over a field of characteristic zero. Then L has a faithful finite-dimensional representation. Moreover, a representation can be constructed such that the elements of the nilradical of L are represented by nilpotent matrices.*

Proof. Set $\sigma = \text{Representation}(L)$. First we suppose that L is solvable. Then by Proposition 6.2 we have that $\sigma(x)$ is nilpotent for all $x \in \text{NR}(L)$. If L is not solvable, then σ is the direct sum of the adjoint representation and a representation ρ , that was constructed by several extension steps. Now for $x \in \text{NR}(L)$ we have that $\text{ad}(x)$ is nilpotent. Also $\rho(x)$ is nilpotent; so the conclusion is that $\sigma(x)$ is nilpotent. \square

Now we consider bounding the degree (i.e., the dimension of the representation space) of the representation produced. In general we are not able to do this; however if L is nilpotent then we can provide a bound. For this we introduce a weight function w on $U(L)$, following [5]. So let L be a nilpotent Lie algebra of nilpotency class c . This means that the lower central series of L is

$$L = L^1 \supset L^2 \supset \dots \supset L^c \supset L^{c+1} = 0.$$

Then for $x \in L$ we let $w(x)$ be the number k such that $x \in L^k$ but $x \notin L^{k+1}$. We extend w to $U(L)$ by setting $w(ab) = w(a) + w(b)$ and $w(a + b) = \min(w(a), w(b))$ if $a + b \neq 0$. Furthermore we set $w(1) = 0$ and $w(0) = \infty$. Let $K_1 \subset K_2 \subset \dots \subset K_r = L$ be the series constructed in the algorithm `ExtensionSeries`. Then $K_{i+1} = K_i \rtimes \langle y_i \rangle$ and $K_1 = \langle x_1, \dots, x_s \rangle$. Let ρ_1 be a representation of K_1 given by $\rho_1(x_i) = e_{1,i+1}^{s+1}$. Then by successively extending ρ_1 we obtain representations ρ_i of K_i .

Proposition 6.8 *Suppose that $\rho_i(a) = 0$ for every element $a \in U(K_i)$ such that $w(a) \geq c + 1$. Then $\rho_{i+1}(b) = 0$ for all $b \in U(K_{i+1})$ such that $w(b) \geq c + 1$. Furthermore $f(b) = 0$ for all $f \in S_{\rho_i}$ and $b \in U(K_i)$ such that $w(b) \geq c + 1$.*

Proof. For the first statement let $b \in U(K_{i+1})$ be a monomial such that $w(b) \geq c + 1$. According to the construction of ρ_{i+1} there are two cases to be considered.

First we consider the case where there is a $\tilde{y}_i \in K_i$ such that $y_i - \tilde{y}_i \in Z(K_{i+1})$. This means that ρ_{i+1} is constructed in the first part of `ExtendRepresentation`. After replacing y_i by $y_i - \tilde{y}_i$ we may suppose that $\tilde{y}_i = 0$. Then $\rho_{i+1}(y_i)\rho_{i+1}(a) = \rho_{i+1}(a)\rho_{i+1}(y_i) = 0$ for all $a \in U(K_i) \setminus \{1\}$. Now if b contains a y_i , then $\rho_{i+1}(b) = 0$. Otherwise b is also an element of $U(K_i)$ and again from the construction of ρ_{i+1} it is seen that $\rho_{i+1}(b) = 0$.

Now we consider the case where ρ_{i+1} is constructed by `GeneralExtension`. Let a be an element of $U(K_i)$, then we claim that $w(y_i a - a y_i) \geq w(a) + w(y_i)$. First we have $w(a y_i) = w(y_i a) = w(a) + w(y_i)$. So if $y_i a - a y_i \neq 0$, then the claim follows from the fact that the weight of a sum is the least of the weights of its terms. On the other hand, if

$y_i a - a y_i = 0$ then its weight is ∞ . Let $f = y_i^l \cdot \theta(v_p, v_r^*)$ be an element of S_{ρ_i} , where $l \geq 0$. Then for $b' \in U(K_i)$ we calculate

$$(b \cdot f)(b') = \pm v_r^*(\rho_i(y_i^l \cdot (b \cdot b'))v_p).$$

Here $y_i \cdot g = y_i g - g y_i$ and $x \cdot g = g x$ for $x \in K_i$ and $g \in U(K_i)$. Now from our claim above it follows that $w(y_i^l \cdot (b \cdot b')) \geq w(b)$. (Note that $y_i^l \cdot (b \cdot b')$ lies in $U(K_i)$ whereas b lies in $U(K_{i+1})$.) Hence $\rho_i(y_i^l \cdot (b \cdot b')) = 0$ so that $b \cdot f = 0$. Now by Lemma 6.5 we have that $\rho_{i+1}(b) = 0$.

For the second statement let $b \in U(K_i)$ be an element such that $w(b) \geq c + 1$ and let $f = y_i^l \cdot \theta(v_p, v_r^*)$ be an element of S_{ρ_i} . Then $f(b) = \pm v_r^*(\rho_i(y_i^l \cdot b)v_p)$, which is 0 because $w(y_i^l \cdot b) \geq w(b) \geq c + 1$. \square

Corollary 6.9 *Let L be a nilpotent Lie algebra of dimension n and nilpotency class c . Set*

$$\sigma = \text{Representation}(L).$$

Then the degree of σ is bounded from above by $\binom{n+c}{c}$.

Proof. Since $\sigma = \rho_r$ we have that the representation space of σ is $S_{\rho_{r-1}}$. The representation ρ_1 of K_1 satisfies the requirement of Proposition 6.8. The conclusion is that $f(b) = 0$ for $f \in S_{\rho_i}$, $b \in U(K_i)$ such that $w(b) \geq c + 1$ and $i = 1, \dots, r - 1$. It follows that the degree of σ is bounded from above by the number of monomials in $U(L)$ of degree at most c , which is equal to $\binom{n+c}{c}$. \square

Remark. Since the maximal nilpotency class of a Lie algebra of dimension n is $c = n - 1$, we have that the bound of Corollary 6.9 in general is exponential in n . However, for Lie algebras of constant nilpotency class, the bound is polynomial in n .

Remark. If the field over which L is defined is of characteristic $p > 0$, then L might not have a Levi decomposition. However, if L has a Levi decomposition, then the algorithm will yield a representation for L also in this case.

6.5 Examples, and practical experience

Let $a \in U(L)$ be a monomial. Then f_a will denote the element of $U(L)^*$ that takes the value 1 on a and 0 on all other monomials.

Example 6.10 Let $L = K \rtimes \langle y \rangle$, where K is a commutative subalgebra spanned by $\{x_1, \dots, x_t\}$. We suppose that L is not commutative and try to find a representation of

n	$\dim L_n$	nilpotency class	Degree	Runtime (s)
3	3	2	3	4
4	6	3	7	26
5	10	4	16	350
6	15	5	35	3311

Table 6.1: Degrees of the representation of the Lie algebra L_n found by the algorithm **Representation**. The last column displays the runtime of the process in seconds.

L . We start with a representation ρ of K given by $\rho(x_i) = e_{1,i+1}^{i+1}$. Then one extension step will yield a representation of L . First we calculate C_ρ . For $i > 1$ we have

$$\theta(v_i, v_j^*)(a) = v_j^*(\rho(a)v_i) = \delta_{j,1}f_{x_{i-1}}(a) + \delta_{i,j}f_1(a).$$

So a basis of C_ρ is given by

$$\{f_1, f_{x_1}, \dots, f_{x_i}\}.$$

We suppose that $[y, x_i] = \sum_j c_{ij}x_j$ and we calculate the action of y on C_ρ :

$$y \cdot f_{x_i}(a) = -f_{x_i}(ya - ay) = \sum_{k=1}^i -c_{ki}f_{x_k}(a).$$

It follows that C_ρ is already a module for L . The values of the representation $\sigma : L \rightarrow \mathfrak{gl}(C_\rho)$ are given by $\sigma(x_i) = e_{1,i+1}^{i+1}$ and $\sigma(y) = -(\text{ad } y)^\top$.

Example 6.11 Let $L = \mathfrak{gl}_n(F)$, the Lie algebra of all $n \times n$ matrices. The Levi decomposition of this Lie algebra is $L = \langle x \rangle \rtimes K$, where $K \cong \mathfrak{sl}_n(F)$ and $\langle x \rangle = Z(L)$. Then we start with a representation ρ of the 1-dimensional Lie algebra $\langle x \rangle$, given by $\rho(x) = e_{1,2}^2$. Now $C_\rho = \{f_1, f_x\}$. Since K commutes with $\langle x \rangle$ we have that C_ρ is a trivial module for L . Hence in this case we need to take the direct sum with the adjoint representation of L , obtaining a representation of degree $n^2 + 2$.

Example 6.12 We implemented the algorithm inside ELIAS. We tried the method on the Lie algebras L_n of strictly upper triangular matrices of order n , for $n = 3, 4, 5, 6$. The degrees of the resulting representations are shown in Table 6.1. It is seen that the resulting degree is much less than the bound provided by Corollary 6.9. However the algorithm seems to have an exponential behaviour. So for nilpotent Lie algebras of small dimension the algorithm works fine, but when the dimension and the nilpotency class increase, the algorithm may become slow.

Chapter 7

Practice

In this chapter we demonstrate how the system ELIAS can be used in two examples. In the first example (Section 7.1) we show how isomorphism of Lie algebras can be tested. Then in Section 7.2 the system is used to obtain information about centralisers of nilpotent elements in E_8 .

7.1 Isomorphism testing

Let L_1 and L_2 be two n -dimensional Lie algebras. We consider the problem of deciding whether L_1 and L_2 are isomorphic. A method for deciding this can be applied to the theory of Lie point symmetries. Here it can be used to identify the symmetry Lie algebra of a differential equation as a member of existing lists of isomorphism classes of Lie algebras. Also the problem is relevant for the construction of those lists. Several partially overlapping lists are known (see e.g., [4], [43], [46]). For the unification of those lists, the use of a computer seems indispensable.

A first approach to the problem is to calculate as many structural invariants of the Lie algebras as possible. A structural invariant is a function

$$f : \{\text{Lie algebras}\} \longrightarrow \{\text{objects}\},$$

such that $L_1 \cong L_2$ implies $f(L_1) = f(L_2)$. Examples of structural invariants are the centre, the derived and lower central series, the nilradical, etc. Algorithms for determining structural invariants have been given in the preceding chapters. By calculating these for L_1 and L_2 we might be able to demonstrate that they are *not* isomorphic. In some cases, for example when L_1 and L_2 are semisimple Lie algebras over a field of characteristic 0 (Chapter 5), we can always decide the isomorphism of L_1 and L_2 this way.

There is also a direct method for testing isomorphism. This method was described in [22]. Let $\{x_1, \dots, x_n\}$ be a basis of L_1 and let $\{y_1, \dots, y_n\}$ be a basis of L_2 . Let (c_{ij}^k) and (γ_{ij}^k)

be the structure constants of L_1 and L_2 , respectively. If $\phi : L_1 \rightarrow L_2$ is an isomorphism given by $\phi(x_i) = \sum_j a_{ij}y_j$, then

$$[\phi(x_i), \phi(x_j)] = \sum_{k,l=1}^n a_{ik}a_{jl}[y_k, y_l] = \sum_{k,l,m=1}^n \gamma_{kl}^m a_{ik}a_{jl}y_m,$$

and

$$\phi([x_i, x_j]) = \sum_{k=1}^n c_{ij}^k \phi(x_k) = \sum_{k,m=1}^n c_{ij}^k a_{km} y_m.$$

This amounts to the following equations in the variables a_{ij} :

$$\sum_{k,l=1}^n \gamma_{kl}^m a_{ik}a_{jl} - \sum_{k=1}^n c_{ij}^k a_{km} = 0, \quad (7.1)$$

for $1 \leq i < j \leq n$ and $1 \leq m \leq n$. Also the determinant of the matrix (a_{ij}) may not be 0. So in [22] the set of equations (7.1) together with $\det(a_{ij}) - d = 0$ is solved by a Gröbner basis calculation.

The advantages of this last method are clear: first of all it always gives the correct answer and secondly it is possible to construct the isomorphism matrix explicitly. The main disadvantage however is that the Gröbner basis computation tends to become very time consuming. Compared to this the calculation of structural invariants is very fast. So the idea presents itself to combine the two approaches. First we calculate as many structural invariants of the Lie algebras as possible. If this leads to a decision regarding their isomorphism then we are happy. If not, then we do a Gröbner basis computation, using the structural invariants that we computed to reduce the number of equations and the number of variables in (7.1), whenever possible. We illustrate this idea with an example.

Example 7.1 Let L_1 and L_2 be 4-dimensional Lie algebras over a field of characteristic 0, and let their Lie multiplication be given by

$$\begin{aligned} [x_2, x_3] &= x_1, & [x_2, x_4] &= x_2, & [x_3, x_4] &= -x_3 \\ [y_2, y_3] &= y_1, & [y_2, y_4] &= -y_3, & [y_3, y_4] &= y_2. \end{aligned}$$

(This is Example 2 of [22]). First we have that $Z(L_1) = \langle x_1 \rangle$ and $Z(L_2) = \langle y_1 \rangle$ so that $\phi(x_1) = a_{11}y_1$. Secondly $\text{NR}(L_1) = \langle x_1, x_2, x_3 \rangle$ and $\text{NR}(L_2) = \langle y_1, y_2, y_3 \rangle$. Hence

$$\begin{aligned} \phi(x_2) &= a_{21}y_1 + a_{22}y_2 + a_{23}y_3 \\ \phi(x_3) &= a_{31}y_1 + a_{32}y_2 + a_{33}y_3 \end{aligned}$$

Also a Cartan subalgebra of L_1 is spanned by x_1, x_4 and a Cartan subalgebra of L_2 is spanned by y_1, y_4 . Since all Cartan subalgebras are conjugate under the automorphism

group of L_1 we may assume that $\phi(x_4) = a_{41}y_1 + a_{44}y_4$. This leads to the following system of equations:

$$\{a_{22}a_{33} - a_{23}a_{32} - a_{11}, a_{21}, a_{23}a_{44} - a_{22}, a_{22}a_{44} + a_{23}, \\ a_{31}, a_{33}a_{44} + a_{32}, a_{32}a_{44} - a_{33}, d - a_{11}(a_{22}a_{33} - a_{23}a_{32})a_{44}\}$$

So we have 8 equations in 10 variables, which is much better than the set of 22 equations in 17 variables given in [22]. Also it is seen that we can easily factorise the determinant. This allows us to replace one polynomial of high degree by some polynomials of smaller degree.

Now we describe in generality the approach that we are using. We assume that the Lie algebras have rational structure constants, but are defined over the algebraic closure $\overline{\mathbb{Q}}$. Then there exists an isomorphism if and only if 1 is not an element of the Gröbner basis we calculate.

For a Lie algebra L we distinguish two types of structural invariants. First we have those that help us to reduce the number of variables and equations. The centre $Z(L)$ is an example of such an invariant. These are called *reduction invariants*. An example of a non-reduction invariant is the dimension of the associative algebra $(\text{ad } L)^*$. For a nilpotent Lie algebra L of nilpotency class c we consider the following reduction invariants:

- the derived series, $L = L_1 \supset L_2 \supset \cdots \supset L_{m+1} = 0$,
- the lower central series, $L = L^1 \supset L^2 \supset \cdots \supset L^{c+1} = 0$,
- the upper central series, $Z(L) = Z_1 \subset Z_1 \subset \cdots \subset Z_c = L$.

As non-reduction invariants we consider $\dim(\text{ad } L)^*$ and $\dim H^2(L, F)$. (The second can be calculated by solving a system of linear equations, we do not go into this here.) These invariants are also applied to the members of the above series.

If L is solvable, then we consider the following reduction invariants:

- the nilradical and its series,
- the derived, lower central and upper central series,
- a Cartan subalgebra.

Remark. The Lie algebra L does not have a unique Cartan subalgebra. However, since we have assumed that the field is algebraically closed, all Cartan subalgebras are conjugate under the automorphism group of L (Theorem IX.3 of [32]). Hence we may assume that an isomorphism of the Lie algebras L_1 and L_2 maps a given Cartan subalgebra of L_1 onto

a given Cartan subalgebra of L_2 .

The non-reduction invariants are the same as in the nilpotent case.

If L is not solvable and not nilpotent and defined over a field of characteristic 0, then we calculate a Levi decomposition $L = S \oplus R$ where S is a semisimple subalgebra and R the solvable radical. We calculate all invariants of the solvable Lie algebra R that are listed above. A Cartan subalgebra of S is a reduction invariant and if it splits, then so are the root spaces. Furthermore, the type of S is a non-reduction invariant.

Since nilpotent Lie algebras have the fewest reduction invariants, testing isomorphism of those algebras will be the most difficult. As an example we try to identify elements of the lists [4] and [43] as members of the list [46]. In this last paper the Lie algebras are listed by the sequence of dimensions of the elements of the upper central series. From each of the first two papers we choose a Lie algebra with upper central series dimensions 2,5,7. From [4] this is the Lie algebra with name n_{95}^7 and from [43] we choose $g_{7,106}$. First n_{95}^7 is quickly seen not to be isomorphic to 2, 5, 7A because the dimensions of the associative algebras generated by the adjoint representation differ. Regarding the isomorphism of n_{95}^7 and 2, 5, 7B, the structural invariants do not bring a decision. So we have to force a decision "in extra time" using the Gröbner basis method. We wrote a GAP program that writes the equations in a format accepted by Macaulay2 (see [26]) to a file called `eqs`. Then in Macaulay2 we do the following:

```
i1 = load "eqs"

--loaded eqs
i2 = timing d1*d2*d3*d4*d5 % I

                2
o2 = d1 d2 d4 d5
-- 49.36 seconds

o2 : Time

i3 = timing d1^ 0 % I

o3 = 1
-- 0.07 seconds

o3 : Time
```

Here I is the ideal defined in the file `eqs`. The determinant of the matrix (a_{ij}) (contain-

ing the variables we could not get rid of by using structural invariants) has factorisation $d1*d2*d3*d4*d5$. We determine whether this determinant is in the ideal I . The conclusion is that it is not in this ideal. Also 1 is not in I . It follows that the set of equations has a solution with a nonzero determinant. So the Lie algebras are isomorphic.

In the case of $g_{7,106}$ the structural invariants suffice to decide that this Lie algebra is not isomorphic to 2, 5, 7A to 2, 5, 7H. Then $g_{7,106}$ turned out to be isomorphic to 2, 5, 7I. Macaulay2 used 90.63 seconds for the Gröbner basis computation.

In the following examples we let L_1 be a Lie algebra from the table given in [46]. Then L_2 is constructed from L_1 by a change of basis. If $\{x_1, \dots, x_7\}$ is a basis of L_1 , then $\{y_1, \dots, y_7\}$ is a basis of L_2 where $y_i = x_i + x_{i+1}$ for $i = 1, \dots, 6$ and $y_7 = x_7 + x_1$. Now since L_1 and L_2 are isomorphic we are sure to end up in the “extra time” of the Gröbner basis calculation. In Table 7.1 the timings of some Gröbner basis calculations are listed.

name	time (s)
2, 5, 7A	47
2, 4, 5, 7A	9
2, 3, 5, 7A	1455
2, 3, 4, 5, 7A	74
2, 3, 4, 5, 7C	1090
1, 2, 3, 4, 5, 7B	126379

Table 7.1: Timings of the Gröbner basis calculation relative to two isomorphic forms of some Lie algebras.

From this table it remains unclear whether the duration of the Gröbner basis computation is related to the structure of the Lie algebra. In some cases it is very fast, in other cases very slow or even infeasible.

In general the Gröbner basis procedure turns out to be a method of brute force, not elegant enough to make Lie algebras reveal their identity. However, on many occasions the somewhat more shy procedure of computing structural invariants succeeds in seducing the Lie algebras to concede that they are not isomorphic, or eventually to comply with the Gröbner basis method.

7.2 Calculations in E_8

7.2.1 Preliminaries

Here we briefly describe some concepts related to orbits of nilpotent elements in simple Lie algebras. The proofs that we omit can be found in [8], Chapter 5. In the sequel L will

be a simple Lie algebra over an algebraically closed field of characteristic 0 with algebraic group G . Then G acts on L via the adjoint representation

$$\text{Ad} : G \longrightarrow \text{GL}(L)$$

(see [30]). We consider orbits of nilpotent elements of L under the action of G . Let e be a nilpotent element of L , then by the Jacobson-Morozov theorem ([32], Theorem III.17) there are elements f, h of L such that

$$[e, f] = h, \quad [h, e] = 2e, \quad [h, f] = -2f,$$

i.e., e can be embedded in a subalgebra isomorphic to \mathfrak{sl}_2 .

Proposition 7.2 *Let e_1, e_2 be nilpotent elements of L and let $K_1 = \langle e_1, f_1, h_1 \rangle$ and $K_2 = \langle e_2, f_2, h_2 \rangle$ be two subalgebras isomorphic to \mathfrak{sl}_2 . Then the following are equivalent:*

1. e_1 and e_2 lie in the same G -orbit,
2. K_1 and K_2 lie in the same G -orbit,
3. h_1 and h_2 lie in the same G -orbit.

Let e be a nilpotent element contained in the subalgebra $K = \langle e, f, h \rangle$ isomorphic to \mathfrak{sl}_2 . Then via the adjoint representation K acts on L . By Corollary 7.2 of [29], we have that L splits as a direct sum

$$L = \bigoplus_{i \in \mathbb{Z}} L(i)$$

where $L(i) = \{x \in L \mid [h, x] = ix\}$. Now let H be a Cartan subalgebra of L containing h . Let

$$L = H \oplus \bigoplus_{\alpha \in \Phi} L_\alpha$$

be the Cartan decomposition of L with respect to H . Then we define a function $\eta : \Phi \rightarrow \mathbb{Z}$ by $\eta(\alpha) = \alpha(h)$.

Proposition 7.3 (Dynkin) *There is a fundamental system Δ of the root system Φ such that $\eta(\alpha) \in \{0, 1, 2\}$ for all $\alpha \in \Delta$.*

Now we can define the *weighted Dynkin diagram* $D(e)$ of e . Let $\Delta = \{\alpha_1, \dots, \alpha_l\}$ be the fundamental system of roots provided by Proposition 7.3. Then $D(e)$ will be the Dynkin diagram of Δ where each node i is labelled by the number $\eta(\alpha_i)$. By the following proposition the weighted Dynkin diagram of e identifies its nilpotent orbit.

Proposition 7.4 *If e_1, e_2 are nilpotent elements of L then they lie in the same G -orbit if and only if $D(e_1) = D(e_2)$.*

7.2.2 Centralisers of nilpotent elements in E_8

Now let L be a Lie algebra of exceptional type. Then the possible weighted Dynkin diagrams of nilpotent elements have been determined by Dynkin ([15], see also [8]). Here we concentrate on the particular case where L is of type E_8 .

Let $\Delta = \{\alpha_1, \dots, \alpha_8\}$ be a fundamental system of roots. Let D be a weighted Dynkin diagram with weights $\eta(\alpha_1), \dots, \eta(\alpha_8)$. We describe how an element h of a Cartan subalgebra H can be found with the property that $\alpha_i(h) = \eta(\alpha_i)$ for $1 \leq i \leq 8$. The Cartan subalgebra H has a basis consisting of elements h_{α_i} with the property that $\alpha_i(h_{\alpha_j}) = \langle \alpha_i, \alpha_j \rangle$ (see [29]). It follows that we can find an appropriate $h \in H$ by applying the inverse of the Cartan matrix of Δ to the vector $(\eta(\alpha_1), \dots, \eta(\alpha_8))$.

We decompose L with respect to the element h found above as a direct sum of eigenspaces $L(i)$. We now want an element $e \in L(2)$ that is a member of the nilpotent class corresponding to D . Representatives for every class of nilpotent elements in L are known (see [8], [15], [27]). These elements all have a *name* and a *diagram*. The diagram is defined as follows. For every root α we fix a nonzero root vector x_α . Then $L(2) = \langle x_\alpha \mid \alpha(h) = 2 \rangle$. Now let $x = x_{\beta_1} + \dots + x_{\beta_m}$ be an element of $L(2)$. The diagram corresponding to the element x has m nodes, labelled by $1, \dots, m$. If $\langle \beta_i, \beta_j \rangle = -1$ then node i is connected to node j by a single bond. If $\langle \beta_i, \beta_j \rangle = 1$ then i is connected to j by a dotted line. The remaining case is $\langle \beta_i, \beta_j \rangle = 0$; here we do not draw a bond. In Table 7.2 there is a list of the nilpotent classes in E_8 . For each class we have listed the diagram and the name of a representative of the class. We remark that in most cases more than one diagram is possible (see [17]).

Finally we look for an element $e \in L(2)$ that has the diagram corresponding to the nilpotent class of D . This can be done by a trial and error method, or by a small procedure trying all possibilities.

For every class of nilpotent elements we followed the procedure described above. The result is displayed in Table 7.2. The labels of the diagrams in Table 7.2 correspond to basis elements of the basis of E_8 used in GAP. We get the nilpotent element corresponding to a diagram by summing the basis elements of E_8 corresponding to the labels in the diagram (see the example below). Also in Table 7.2 there is some information about the centralisers of the nilpotent elements. The Levi decompositions can be found in [17]; there some mistakes that appeared in an earlier paper ([18]) were corrected. Here we also give the decomposition of the radical as a direct sum of irreducible representations of the Levi factor.

Example 7.5 Here we demonstrate how the data contained in Table 7.2 can be calculated in GAP. In this example we take the element with name D_5 .

In GAP we first issue the command that constructs E_8 . Then we get the nilpotent element by summing the basis elements of L corresponding to the labels in the appropriate diagram

of Table 7.2. We calculate the centraliser CL of the subalgebra generated by that element. Then a Levi decomposition of CL is computed. The first element of this decomposition is the semisimple part of CL. We determine the type of this part.

```
gap> L:=SimpleLieAlgebra("E",8,Rationals);
<Lie algebra of dimension 248 over Rationals>
gap> b:= BasisVectors( Basis( L ) );
gap> K:= Subalgebra( L, [ b[1]+b[7]+b[8]+b[44]+b[61] ] );
<Lie algebra over Rationals, with 1 generators>
gap> CL:= LieCentralizer( L, K );
<Lie algebra of dimension 48 over Rationals>
gap> ll:= LeviDecomposition( CL );
[ <Lie algebra of dimension 21 over Rationals>,
  <Lie algebra of dimension 27 over Rationals> ]
gap> SemiSimpleType( ll[1] );
"B3"
```

Now the solvable radical offers a representation of the semisimple part S of CL. By the representation theory of semisimple Lie algebras this solvable radical decomposes as a direct sum of irreducible modules and each module is determined by its *highest weight* (see [29]). Before we determine the weights of S on CL, we describe the basis of a Cartan subalgebra relative to which we describe these weights. Let $\alpha_1, \dots, \alpha_l$ be a fundamental system of roots of S (in the order given in [29]). Let x_{α_i} be a root vector belonging to α_i and similarly for $y_{-\alpha_i}$. Then $h_i = [x_{\alpha_i}, y_{-\alpha_i}]$ forms a basis of a Cartan subalgebra H . We multiply these elements by a scalar in order to ensure that $[h_i, x_{\alpha_i}] = 2x_{\alpha_i}$ and $[h_i, y_{-\alpha_i}] = -2y_{-\alpha_i}$. Now the weights are taken relative to this basis.

In the example we first calculate the root system of S and look at the Cartan matrix. This matrix tells us how we have to reorder the roots.

```
gap> rr:= RootSystem( ll[1] );
gap> Print( rr.cartanmat );
[ [ 2, -2, -1 ], [ -1, 2, 0 ], [ -1, 0, 2 ] ]
```

From this it follows that $\alpha_3, \alpha_1, \alpha_2$ is the correct order of the fundamental roots. Now we take basis vectors of the root spaces corresponding to these roots and store them in the variable x .

```
gap> v:= rr.rootvecs;; r:= rr.roots;;
gap> x:= [ v[3], v[1], v[2] ];
[ v.125, v.124, v.32+(-1)*v.123 ]
```

(The vectors corresponding to the fundamental roots come first in the list `rr.rootvecs`.) The positive roots are listed first in the list `rr.roots` and then the negative ones. Now, to find the root vectors corresponding to the negative roots $-\alpha_i$ we first determine where

the negative roots start in the list of roots (then we also know where the corresponding vectors are in the list of vectors).

```
gap> Position( r, -r[1] );
10
gap> y:= [ v[12], v[10], v[11] ];
[ v.5, v.4, v.3+(-1)*v.152 ]
```

Now a basis of a Cartan subalgebra is given by $x[ii]*y[ii]$. We calculate this basis and if necessary multiply an element by a scalar.

```
gap> bH:= List( [1,2,3], ii -> x[ii]*y[ii] );;
gap> List( [1,2,3], ii -> bH[ii]*x[ii] );
[ (-2)*v.125, (-2)*v.124, (2)*v.32+(-2)*v.123 ]
gap> bH[1]:=-bH[1];; bH[2]:=-bH[2];;
gap> List( [1,2,3], ii -> bH[ii]*y[ii] );
[ (-2)*v.5, (-2)*v.4, (-2)*v.3+(2)*v.152 ]
```

We wrote a simple program for calculating the weights. It takes the basis vectors we just calculated and uses the fact that the matrices of the action of these elements on the solvable radical are already in diagonal form.

```
gap> Weights( bH, ll[2] );
[ [ -1, 0, 0 ] ]
[ [ -1, 0, 1 ], [ -1, 0, 1 ] ]
[ [ -1, 1, -1 ], [ -1, 1, -1 ] ]
[ [ -1, 1, 0 ] ]
[ [ 0, -1, 1 ], [ 0, -1, 1 ] ]
[ [ 0, -1, 2 ] ]
[ [ 0, 0, -1 ], [ 0, 0, -1 ] ]
[ [ 0, 0, 0 ], [ 0, 0, 0 ], [ 0, 0, 0 ], [ 0, 0, 0 ], [ 0, 0, 0 ] ]
[ [ 0, 0, 1 ], [ 0, 0, 1 ] ]
[ [ 0, 1, -2 ] ]
[ [ 0, 1, -1 ], [ 0, 1, -1 ] ]
[ [ 1, -1, 0 ] ]
[ [ 1, -1, 1 ], [ 1, -1, 1 ] ]
[ [ 1, 0, -1 ], [ 1, 0, -1 ] ]
[ [ 1, 0, 0 ] ]
```

The weight $[1, 0, 0]$ occurs. Using the function `Demazure` in `LiE` (see [12]) we see that the irreducible module of B_3 with this highest weight has dimension 7. We subtract the corresponding weights from the list above and continue. At the end we find that as an S -module the solvable radical decomposes as

$$R_7^{100} \oplus R_8^{001} \oplus R_8^{001} \oplus R_1^0$$

where R_m^λ denotes the irreducible module with highest weight λ and dimension m .

Table 7.2: Nilpotent elements and their centralisers in E_8 .

The labels in the diagrams correspond to basis elements of E_8 (where the basis of the GAP-implementation is used). S is the semisimple part of the centraliser and R is its solvable radical; in the fourth column the dimension of R is given or its decomposition as an S -module. Here R_m^λ means the irreducible module of dimension m and highest weight λ ; thus R_1^0 is the trivial module. The order of the roots as given in [29] is used to calculate the weights.

Diagram	Name	S	R
$\circ 120$	A_1	E_7	$R_{56}^{0000001} R_1^0$
$\circ 97 \circ 120$	$2A_1$	B_5	$R_{64}^{000001} R_{13}^{100000} R_1^0$
$\circ 74 \circ 104 \circ 118$	$3A_1$	$F_4 + A_1$	$R_{52}^{00011} R_{26}^{00010} R_2^{00001} R_1^0$
$\circ 8 \text{---} \circ 119$	A_2	E_6	$R_{27}^{100000} R_{27}^{000001} 2R_1^0$
$\circ 69 \circ 91 \circ 106 \circ 114$	$4A_1$	C_4	$R_8^{1000} R_{27}^{0100} R_{48}^{0010} R_1^0$
$\circ 47 \text{---} \circ 112 \circ 97$	$A_2 + A_1$	A_5	$2R_6^{10000} R_{15}^{01000} R_{20}^{00100}$ $+ R_{15}^{00010} 2R_6^{00001} 3R_1^0$
$\circ 61 \text{---} \circ 101 \circ 79 \circ 102$	$A_2 + 2A_1$	$B_3 + A_1$	$R_{32}^{0013} R_{21}^{1002} R_{16}^{0011} R_5^{0004} R_3^{0002} R_1^0$
$\circ 8 \text{---} \circ 97 \text{---} \circ 74$	A_3	B_5	$R_{32}^{00001} R_{11}^{10000} 2R_1^0$
$\circ 44 \text{---} \circ 96 \circ 80 \circ 90 \circ 99$	$A_2 + 3A_1$	$G_2 + A_1$	$R_{27}^{200} R_{28}^{011} R_{14}^{101} R_7^{100} R_1^0$
$\circ 1 \text{---} \circ 112 \text{---} \circ 44 \text{---} \circ 96$	$2A_2$	$2G_2$	$R_{49}^{1010} R_7^{1000} R_7^{0010} R_1^0$
$\circ 39 \text{---} \circ 81 \text{---} \circ 73 \text{---} \circ 93 \circ 74$	$2A_2 + A_1$	$G_2 + A_1$	$R_{21}^{102} 2R_{14}^{101} R_7^{100} R_4^{003}$ $+ R_3^{002} 2R_2^{001} 2R_1^0$
$\circ 22 \text{---} \circ 82 \text{---} \circ 81 \circ 80$	$A_3 + A_1$	$B_3 + A_1$	$R_{14}^{1001} R_{16}^{0011} R_7^{1000}$ $+ 2R_8^{0010} 2R_2^{0001} 3R_1^0$
$\begin{array}{c} \circ 68 \quad \circ 97 \\ \square \\ \circ 7 \quad \circ 15 \end{array}$	$D_4(a_1)$	D_4	$2R_8^{1000} 2R_8^{0010} 2R_8^{0001} 6R_1^0$

continued on next page

continued from previous page			
Diagram	Name	S	R
	D_4	F_4	$R_{26}^{0001} 2R_1^0$
	$2A_2 + 2A_1$	B_2	$R_{20}^{03} R_{10}^{02} R_{14}^{20} R_{16}^{11} R_5^{10} R_4^{01} R_1^0$
	$A_3 + 2A_1$	$B_2 + A_1$	$R_{12}^{012} R_3^{002} R_{10}^{101} 2R_8^{011}$ $+ 3R_2^{001} R_5^{100} 2R_4^{010} 3R_1^0$
	$D_4(a_1) + A_1$	$3A_1$	$R_8^{111} 2R_4^{110} 2R_4^{101} 2R_4^{011}$ $+ 4R_2^{100} 4R_2^{104} R_2^{001} 7R_1^0$
	$A_3 + A_2$	B_2	$3R_5^{10} 8R_4^{01} 13R_1^0$
	A_4	A_4	$2R_5^{1000} R_{10}^{0100} R_{10}^{0010} 2R_5^{0001} 4R_1^0$
	$A_3 + A_2 + A_1$	$2A_1$	$R_9^{80} R_{14}^{61} R_7^{60} R_{10}^{41} 2R_5^{40} R_6^{21} R_3^{20} R_1^0$
	$D_4 + A_1$	C_3	$2R_6^{100} R_{14}^{010} R_{14}^{001} 3R_1^0$
	$D_4(a_1) + A_2$	A_2	$R_{27}^{22} R_{10}^{30} R_{10}^{03} R_8^{11} R_1^0$
	$A_4 + A_1$	A_2	$6R_3^{10} 6R_3^{01} 16R_1^0$
	$2A_3$	B_2	$R_{16}^{11} R_{10}^{02} 2R_5^{10} 3R_4^{01} 2R_1^0$
	$D_5(a_1)$	A_3	$3R_4^{100} 2R_6^{100} 3R_4^{001} 7R_1^0$
	$A_4 + 2A_1$	A_1	$3R_3^{21} 14R_2^{16} 16R_1^0$

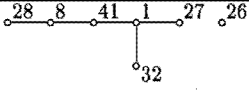
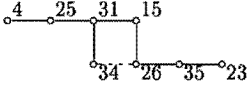
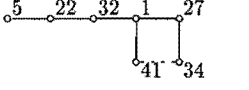
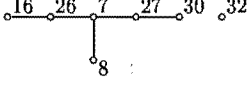
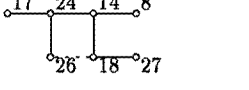
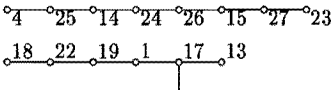
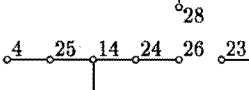
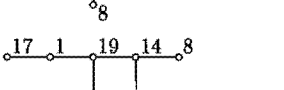
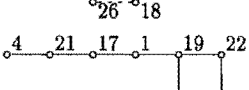
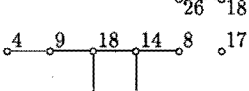
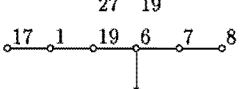
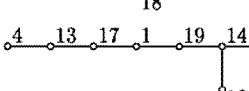
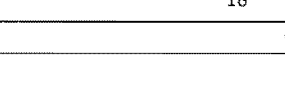
continued on next page

continued from previous page			
Diagram	Name	S	R
	$A_4 + A_2$	$2A_1$	$R_{12}^{15} R_8^{13} R_4^{11} R_7^{06} 2R_9^{04} 2R_3^{02} R_1^0$
	A_5	$G_2 + A_1$	$R_{14}^{101} 2R_7^{100} 2R_2^{001} 3R_1^0$
	$A_4 + A_2 + A_1$	A_1	$R_7^6 2R_6^5 2R_5^4 2R_4^3 2R_3^2 2R_2^1 2R_1^0$
	$A_5 + A_1$	G_2	$4R_7^{10} 8R_1^0$
	$A_4 + A_3$	A_1	$R_6^5 2R_5^4 3R_4^3 3R_3^2 2R_2^1 2R_1^0$
	$A_5 + A_1$	$2A_1$	$R_4^{30} R_6^{21} 2R_3^{20} 2R_4^{11} 4R_2^{10} 2R_2^{01} 4R_1^0$
	$D_5(a_1) + A_2$	A_1	$R_5^4 3R_4^3 4R_3^2 5R_2^1 4R_1^0$
	$D_6(a_2)$	$2A_1$	$2R_4^{15} 5R_2^{01} 5R_2^{10} 10R_1^0$
	$A_5 + 2A_1$	A_1	$R_4^3 4R_3^2 8R_2^1 9R_1^0$
	$A_5 + A_2$	A_1	$9R_1^2 21R_1^0$

continued on next page

continued from previous page			
Diagram	Name	S	R
	$D_5 + A_1$	$2A_1$	$R_8^{21} R_3^{20} 2R_4^{11} 2R_2^{01} 4R_2^{10} 5R_1^0$
	$A_5 + A_2 + A_1$	0	≤ 0
	A_6	$2A_1$	$R_8^{13} 2R_4^{11} R_3^{04} 3R_3^{02} 2R_1^0$
	$D_6(a_1)$	$2A_1$	$2R_4^{14} R_2^{10} 4R_2^{01} 8R_1^0$
	$A_6 + A_1$	A_1	$R_5^2 R_4^3 3R_3^2 4R_2^3 3R_1^0$
	$D_6(a_1) + A_1$	A_1	$8R_2^{17} R_1^0$
	$E_6(a_1)$	A_2	$3R_3^{12} 2R_1^0$
	$D_5 + A_2$	0	33
	D_6	B_2	$R_5^{10} 3R_4^{15} R_1^0$
	E_6	G_2	$2R_7^{10} 4R_1^0$
	$D_7(a_2)$	0	32
	A_7	A_1	$R_4^3 3R_3^2 5R_2^4 R_1^0$
	$E_6(a_1) + A_1$	0	30

continued on next page

<i>continued from previous page</i>			
Diagram	Name	<i>S</i>	<i>R</i>
	$D_6 + A_1$	A_1	$6R_2^{11}3R_1^0$
	$D_8(a_3)$	0	28
	$D_7(a_1)$	0	26
	$E_6 + A_1$	A_1	$R_4^3 2R_3^2 4R_2^5 R_1^0$
	$E_7(a_2)$	A_1	$5R_2^{11}11R_1^0$
	A_8	0	$2\pm$
	D_7	A_1	$R_3^5 5R_2^6 R_1^0$
	$E_6 + A_2$	0	22
	$E_7(a_1)$	A_1	$4R_2^9 9R_1^0$
	$D_8(a_1)$	0	20
	$E_7(a_1) + A_1$	0	18
	E_7	A_1	$3R_3^4 4R_1^0$
	D_8	0	16

continued on next page

continued from previous page			
Diagram	Name	S	R
	$E_7 + A_1$	0	14
	$E_8(a_2)$	0	12
	$E_8(a_1)$	0	10
	E_8	0	8

Let f be an element of the dual space L^* . Then we set

$$L^f = \{x \in L \mid f([x, y]) = 0 \text{ for all } y \in L\}.$$

The *index* of L is defined as the number $r(L) = \inf_{f \in L^*} \dim L^f$. For a semisimple Lie algebra it is known that its index is equal to its rank (see [14]). So the index of E_8 is 8. It was a conjecture communicated to the author by A. Elashvili that the index of a centraliser of a nilpotent element in E_8 is also equal to 8. Here we are concerned with the calculation of these indices. Let K be a subalgebra of L with basis $\{x_1, \dots, x_n\}$ and let $\{x_1^*, \dots, x_n^*\}$ be the corresponding basis of K^* . Let $f = \sum_i T_i x_i^*$ be an element of K^* and let $x = \sum_i \alpha_i x_i$ be an element of K ; then $x \in K^f$ if and only if

$$\sum_{i=1}^n \left(\sum_{k=1}^n c_{ij}^k T_k \right) \alpha_i = 0 \quad \text{for } j = 1, \dots, n$$

(where the c_{ij}^k are the structure constants of K). It follows that the dimension of K^f is minimal if and only if the rank of the matrix $A = (\sum_k c_{ij}^k T_k)_{1 \leq i, j \leq n}$ is maximal. The rank of A is not maximal if and only if some polynomials (determinants of certain minors of A) in the T_k vanish. The conclusion is that if we choose some random substitution for the T_k , then with high probability the rank of A will be maximal. So we have an efficient probabilistic algorithm for testing whether the index of a centraliser is 8.

Algorithm Index

Input: A centraliser K of a nilpotent element in E_8 .

Output: true if the index of K is 8.

Step 1 Choose a random vector $(\lambda_1, \dots, \lambda_n) \in F^n$.

Step 2 Calculate the rank r of the matrix A where the λ_k are substituted for the T_k .

Step 3 If $n - r = 8$ then return `true` else return to Step 1.

Remark. If the index of a centraliser is bigger than 8, then the algorithm `Index` will loop forever. However, if the index is equal to 8, then the algorithm will output the correct answer after a few steps.

Now we show an example of the calculation of the index of a centraliser.

```
gap> K:=Subalgebra( L, [ b[39]+b[73]+b[74]+b[81]+b[93] ] );;
gap> CL:= LieCentralizer( L, K );
<Lie algebra of dimension 86 over Rationals>
gap> Index( CL );;
8
gap> Runtime()-t;
2272516
```

The procedure prints the estimate of the index in each round of the iteration. When the estimate is equal to 8 the algorithm stops. It is seen that after one step we are done. We remark that most of the time is spent on calculating a table of structure constants of CL.

We did the same computations for every element of the list of Table 7.2. All centralisers turned out to have index 8.

Corollary 7.6 *Let L be the simple Lie algebra of type E_8 . Let e be a representative of a nilpotent class in L . Then the centraliser $Z_L(e)$ has index 8.*

Appendix A

Manual of ELIAS

Here we give a brief description of the functions that constitute the ELIAS package inside GAP4.

A first concern is how to represent a Lie algebra on a computer (see Section 1.2). In ELIAS it is possible to present Lie algebras in two ways. First there is the possibility of constructing a Lie algebra by a table of structure constants (see A.3, A.4 and A.5). Secondly, a Lie algebra can be given by some matrices that generate the Lie algebra as a subalgebra of the full matrix algebra (see A.6). Also we can make direct sums of Lie algebras (see A.8).

Section A.7 describes a function that constructs the simple Lie algebras.

The Sections A.10, A.11, A.12, A.13, A.14, A.15, A.16 and A.17 describe the construction of several distinguished subalgebras and ideals.

The next sections describe the construction of series of ideals (see A.18, A.19 and A.20).

The next sections describe functions related to decompositions of a Lie algebra into a direct sum of subspaces (see A.21, A.22).

The next section describes several property tests for Lie algebras (see A.23).

The next section describes a function that calculates the type of a semisimple Lie algebra of characteristic 0 (see A.24).

The next section (A.25) describes the construction of the associative algebra generated by the adjoint matrices of the elements of the Lie algebra.

The next section (A.26) describes the construction of the matrix of the Killing form.

The last sections describe functions related to elements of the Lie algebra (see A.27 and A.28).

A.3 About Structure Constants

Here we consider representing a Lie algebra by a table of structure constants. Table A.3 is the multiplication table of \mathfrak{sl}_2 (see also Example 1.2).

In ELIAS such a table is represented using lists. The obvious way to do this is to construct a “three-dimensional” list T such that $T[i][j][k]$ equals c_{ij}^k . But it often happens that

	x_1	x_2	x_3
x_1	0	x_3	$-2x_1$
x_2	$-x_3$	0	$2x_2$
x_3	$2x_1$	$-2x_2$	0

Table A.3: Multiplication table of \mathfrak{sl}_2

many of these constants are 0. Therefore a more complicated structure is used in order to be able to forget the zeros. A multiplication table of an n -dimensional Lie algebra is an $n \times n$ array T such that $T[i][j]$ describes the product of the i -th and the j -th basis element. This product is encoded in the following way. The entry $T[i][j]$ is a list of two elements. The first of these is a list of indices k such that c_{ij}^k is nonzero. The second list contains the corresponding constants c_{ij}^k . For example, if T is the table displayed in Table A.3, then $T[1][3]$ equals the list $[[1], [-2]]$, meaning that the product of first and third basis element of the Lie algebra equals -2 times the first basis element. Furthermore $T[3][3]$ is the list $[[], []]$ which means that the product of the third basis element with itself is zero. Now suppose that S is the table of a Lie algebra with basis $\{x_1, \dots, x_n\}$ and that $S[3][7]$ equals $[[2, 4, 6], [1/2, 2, 2/3]]$. Then in the Lie algebra we have the relation

$$[x_3, x_7] = \frac{1}{2}x_2 + 2x_4 + \frac{2}{3}x_6.$$

Finally two numbers are added to the table. In the case where T is the table of a Lie algebra, the first number is -1 , expressing the fact that the multiplication is anticommutative. The second element that is added is the zero-element of the field over which the Lie algebra is defined.

A.4 TestJacobi

`TestJacobi(T)`

Before constructing a Lie algebra by means of a table of structure constants it is advisable to check whether the resulting algebra satisfies the Jacobi identity. `TestJacobi(T)` returns `true` if in the algebra defined by T the Jacobi identity is satisfied, `false` otherwise. The table T in the next example is the same as the one in Table A.3.

```
gap> T:= [ [ [ [ ], [ ] ], [ [ 3 ], [ 1 ] ], [ [ 1 ], [ -2 ] ] ],
[ [ [ 3 ], [ -1 ] ], [ [ ], [ ] ], [ [ 2 ], [ 2 ] ] ],
[ [ [ 1 ], [ 2 ] ], [ [ 2 ], [ -2 ] ], [ [ ], [ ] ] ], -1, 0 ];
gap> TestJacobi( T );
true
```

A.5 LieAlgebraByStructureConstants

LieAlgebraByStructureConstants(F , T)

This function returns the Lie algebra over the field F defined by the table of structure constants T .

```
gap> T:= [ [ [ [ ], [ ] ], [ [ 3 ], [ 1 ] ], [ [ 1 ], [ -2 ] ] ],
[ [ [ 3 ], [ -1 ] ], [ [ ], [ ] ], [ [ 2 ], [ 2 ] ] ],
[ [ [ 1 ], [ 2 ] ], [ [ 2 ], [ -2 ] ], [ [ ], [ ] ] ], -1, 0 ];
gap> L:=LieAlgebraByStructureConstants( Rationals, T );
<Lie algebra of dimension 3 over Rationals>
```

A.6 AlgebraByGenerators

AlgebraByGenerators(F , $mats$)

In this section we describe the other way to present a Lie algebra, namely by matrices. AlgebraByGenerators(F , $mats$) returns the (matrix) Lie algebra over the field F generated by the elements of the list (of matrices) $mats$. Here we use the Lie algebra spanned by the matrices A_1 , A_2 and A_3 as in Example 1.3.

```
gap> mats:= [ [ [ 0, 1 ], [ 0, 0 ] ], [ [ 0, 0 ], [ 1, 0 ] ],
[ [ 1, 0 ], [ 0, -1 ] ] ];;
gap> mats:=List( mats, x -> LieObject( x ) );;
gap> K:=AlgebraByGenerators( Rationals, mats );
<Lie algebra over Rationals, with 3 generators>
```

A.7 SimpleLieAlgebra

SimpleLieAlgebra(X , n , F)

This function constructs the simple Lie algebra of type X_n over the field F . The result is a Lie algebra defined by a multiplication table. Here X can be one of "A", "B", "C", "D", "E", "F", "G", "W", "S", "H", and "K". For the types "A" to "G" n must be an integer greater or equal to 1. The other types only exist over fields of characteristic $p > 0$. In this case n must be a list of integers ≥ 1 . If X is "H" then this must be a list of even length and it must have odd length if X is "K".

In a few cases the Lie algebra returned by this function is not simple. Examples are the Lie algebras of type A_n over a field of characteristic $p > 0$ where p divides $n + 1$, and the Lie algebras of type K_n where n is a list of length 1.

```

gap> L:=SimpleLieAlgebra( "D", 7, Rationals );
<Lie algebra of dimension 91 over Rationals>
gap> L:=SimpleLieAlgebra( "F", 4, GF(7) );
<Lie algebra of dimension 52 over GF(7)>
gap> L:=SimpleLieAlgebra( "W", [1,1], GF(5) );
<Lie algebra of dimension 50 over GF(5)>
gap> L:=SimpleLieAlgebra( "S", [1,2], GF(5) );
<Lie algebra over GF(5), with 124 generators>
gap> L:=SimpleLieAlgebra( "H", [2,1], GF(5) );
<Lie algebra of dimension 123 over GF(5)>
gap> L:=SimpleLieAlgebra( "K", [1,1,1], GF(5) );
<Lie algebra of dimension 125 over GF(5)>

```

A.8 DirectSumOfAlgebras

`DirectSumOfAlgebras(L_1 , L_2)`

This function returns the direct sum of the (Lie) algebras L_1 and L_2 . It is assumed that either both Lie algebras are given by a table or they both are matrix Lie algebras.

```

gap> L1:=SimpleLieAlgebra( "B", 2, Rationals );
<Lie algebra of dimension 10 over Rationals>
gap> L2:=SimpleLieAlgebra( "C", 3, Rationals );
<Lie algebra of dimension 21 over Rationals>
gap> DirectSumOfAlgebras( L1, L2 );
<Lie algebra of dimension 31 over Rationals>
gap> mats:= [ [ [ 0, 1 ], [ 0, 0 ] ], [ [ 0, 0 ], [ 1, 0 ] ],
[ [ 1, 0 ], [ 0, -1 ] ] ];;
gap> mats:=List( mats, x -> LieObject( x ) );;
gap> K:=AlgebraByGenerators( Rationals, mats );
<Lie algebra over Rationals, with 3 generators>
gap> DirectSumOfAlgebras( K, K );
<Lie algebra over Rationals, with 6 generators>

```

A.9 RootSystem

`RootSystem(L)`

For a semisimple Lie algebra L with a split Cartan subalgebra, this function computes the root system. The output is a record with the following components:

- **roots** This is the set of roots of 'L' with respect to the Cartan subalgebra that is output by `CartanSubalgebra(L)`. First the positive roots are listed according to increasing height. The second half of the list consists of the negative roots.
- **rootvecs** The set of elements of L that are the root vectors corresponding to the roots in `roots` (so the first vector corresponds to the first root and so on).
- **fundroots** A set of fundamental roots.
- **cartanmat** The Cartan matrix of the set of fundamental roots.

```
gap> L:=SimpleLieAlgebra( "G", 2, Rationals );
<Lie algebra of dimension 14 over Rationals>
gap> R:=RootSystem( L );
rec(
roots := [ [ 1, -1 ], [ 0, 1 ], [ 1, 0 ], [ 2, -1 ], [ 3, -2 ], [ 3, -1 ],
[ -1, 1 ], [ 0, -1 ], [ -1, 0 ], [ -2, 1 ], [ -3, 2 ], [ -3, 1 ] ],
rootvecs := [ v.8, v.13, v.9, v.1, v.4, v.11, v.5, v.14, v.10, v.3, v.2,
v.12 ],
fundroots := [ [ 1, -1 ], [ 0, 1 ] ],
cartanmat := [ [ 2, -1 ], [ -3, 2 ] ] )
```

A.10 LieCentre

`LieCentre(L)`

This function returns the centre of L . For the algorithm we refer to Section 1.4.2.

```
gap> L:=SimpleLieAlgebra( "C", 3, Rationals );
<Lie algebra of dimension 21 over Rationals>
gap> LieCentre( L );
<Lie algebra of dimension 0 over Rationals>
```

Note that the definition of centre differs for associative algebras and Lie algebras. That is the reason why this function is called `LieCentre` (the same applies for the functions `LieCentralizer` and `LieNormalizer`). We illustrate this difference with an example.

```
gap> L:=SimpleLieAlgebra( "W", [1,1], GF(2) );
<Lie algebra of dimension 8 over GF(2)>
gap> LieCentre( L );
<Lie algebra of dimension 0 over GF(2)>
gap> Centre( L );
<Lie algebra of dimension 8 over GF(2)>
```

A.11 LieCentralizer

LieCentralizer(L , K)

This function returns the centralizer of the subalgebra K in its parent Lie algebra L . The algorithm was described in Section 1.4.3.

```
gap> L:=SimpleLieAlgebra( "D", 7, Rationals );
<Lie algebra of dimension 91 over Rationals>
gap> b:=BasisVectors( Basis( L ) );
gap> K:=Subalgebra( L, [ b[1], b[2], b[3] ] );
<Lie algebra over Rationals, with 3 generators>
gap> LieCentralizer( L, K );
<Lie algebra of dimension 49 over Rationals>
```

A.12 LieNormalizer

LieNormalizer(L , K)

LieNormalizer(L , K) returns the normalizer of K in its parent algebra L . The algorithm can be found in Section 1.4.4.

```
gap> L:=SimpleLieAlgebra( "D", 7, Rationals );
<Lie algebra of dimension 91 over Rationals>
gap> b:=BasisVectors( Basis( L ) );
gap> K:=Subalgebra( L, [ b[1], b[2], b[3] ] );
<Lie algebra over Rationals, with 3 generators>
gap> LieNormalizer( L, K );
<Lie algebra of dimension 58 over Rationals>
```

A.13 DerivedSubalgebra

DerivedSubalgebra(L)

This function returns the product space of L with itself. See Section 1.4.1 for the algorithm.

```
gap> L:=SimpleLieAlgebra( "E", 8, Rationals );
<Lie algebra of dimension 248 over Rationals>
gap> b:=BasisVectors( Basis( L ) );
gap> K:=Subalgebra( L, [ b[1]+b[4]+b[8]+b[13]+b[14]+b[17]+b[18]+b[19] ] );
<Lie algebra over Rationals, with 1 generators>
gap> CL:=LieCentralizer( L, K );
```



```

<Lie algebra of dimension 16 over Rationals>
gap> DerivedSubalgebra( CL );
<Lie algebra of dimension 10 over Rationals>

```

A.14 SolvableRadical

SolvableRadical(L)

This function returns the solvable radical of L . The algorithm was described in Section 1.4.5.

```

gap> L:=SimpleLieAlgebra( "E", 8, Rationals );;
gap> b:=BasisVectors( Basis( L ) );;
gap> K:=Subalgebra( L, [ b[1]+b[6]+b[7]+b[8]+b[17]+b[18]+b[19] ] );;
gap> CL:=LieCentralizer( L, K );
<Lie algebra of dimension 16 over Rationals>
gap> SolvableRadical( CL );
<Lie algebra of dimension 13 over Rationals>

```

A.15 NilRadical

This function calculates the nilradical of L . In the case of characteristic 0 it uses the “downward” method described in Chapter 2. Otherwise the radical of the associative algebra $(\text{ad } L)^*$ is calculated.

```

gap> L:=SimpleLieAlgebra( "E", 8, Rationals );;
gap> b:=BasisVectors( Basis( L ) );;
gap> K:=Subalgebra(L, [ b[1]+b[6]+b[7]+b[8]+b[17]+b[18]+b[19] ] );;
gap> CL:=LieCentralizer(L,K);
<Lie algebra of dimension 16 over Rationals>
gap> NilRadical( CL );
<Lie algebra of dimension 13 over Rationals>

```

A.16 CartanSubalgebra

CartanSubalgebra(L)

CartanSubalgebra(L) returns a Cartan subalgebra of L . The algorithm works for Lie algebras L defined over a field F such that $|F| > \dim L$ and for restricted Lie algebras of

characteristic p . This function uses the algorithms LRE and RestrictedCartan given in Chapter 3.

```
gap> L:=SimpleLieAlgebra( "E", 8, Rationals );;
gap> b:=BasisVectors( Basis( L ) );;
gap> K:=Subalgebra( L, [ b[1]+b[6]+b[7]+b[8]+b[17]+b[18]+b[19] ] );;
gap> CL:=LieCentralizer( L, K );
<Lie algebra of dimension 16 over Rationals>
gap> CartanSubalgebra( CL );
<Lie algebra of dimension 8 over Rationals>
```

A.17 FindSl2

FindSl2(L, x)

If x is a nilpotent element of a Lie algebra L , then there may exist a subalgebra of L that is isomorphic to \mathfrak{sl}_2 and contains x . By a theorem of Jacobson-Morozov, this is certainly the case if L is semisimple. This function returns a three dimensional subalgebra of L , isomorphic to \mathfrak{sl}_2 and containing x , if it exists. It returns **false** if there is no such subalgebra.

The proof of Morozov's lemma ([32], p. 99) is already constructive. This function is a straightforward implementation of that proof.

```
gap> L:=SimpleLieAlgebra( "G", 2, Rationals );;
<Lie algebra of dimension 14 over Rationals>
gap> b:=BasisVectors( Basis( L ) );;
gap> FindSl2( L, b[1] );
<Lie algebra of dimension 3 over Rationals>
```

A.18 DerivedSeries

DerivedSeries(L)

This function returns a list of ideals of L that form its derived series. It is calculated by repeatedly applying the algorithm for product spaces described in Section 1.4.1.

```
gap> L:=SimpleLieAlgebra( "E", 8, Rationals );;
gap> b:=BasisVectors( Basis( L ) );;
gap> K:=Subalgebra( L, [ b[1]+b[4]+b[8]+b[13]+b[14]+b[17]+b[18]+b[19] ] );;
gap> CL:=LieCentralizer( L, K );
<Lie algebra of dimension 16 over Rationals>
gap> DerivedSeries( CL );
```

```
[ <Lie algebra of dimension 16 over Rationals>,
<Lie algebra of dimension 10 over Rationals>,
<Lie algebra of dimension 2 over Rationals>,
<Lie algebra of dimension 0 over Rationals> ]
```

A.19 LowerCentralSeries

LowerCentralSeries(*L*)

This function calculates the lower central series of *L*. Again the method given in Section 1.4.1 is used.

```
gap> L:=SimpleLieAlgebra( "E", 8, Rationals );;
gap> b:=BasisVectors( Basis( L ) );;
gap> K:=Subalgebra( L, [ b[6]+b[7]+b[8]+b[9]+b[10]+b[11]+b[12]+b[25] ] );;
gap> CL:=LieCentralizer( L, K );
<Lie algebra of dimension 14 over Rationals>
gap> LowerCentralSeries( CL );
[ <Lie algebra of dimension 14 over Rationals>,
<Lie algebra of dimension 7 over Rationals>,
<Lie algebra of dimension 5 over Rationals>,
<Lie algebra of dimension 3 over Rationals>,
<Lie algebra of dimension 2 over Rationals>,
<Lie algebra of dimension 0 over Rationals> ]
```

A.20 UpperCentralSeries

UpperCentralSeries(*L*)

This function calculates the upper central series of *L*. It repeatedly uses the algorithm for the centre while keeping track of the pre-images of the ideals factored out. In ELIAS the upper central series is presented in the reversed order (starting with the hypercenter).

```
gap> L:=SimpleLieAlgebra( "E", 8, Rationals );;
gap> b:=BasisVectors( Basis( L ) );;
gap> K:=Subalgebra( L, [ b[6]+b[7]+b[8]+b[9]+b[10]+b[11]+b[12]+b[25] ] );;
gap> CL:=LieCentralizer( L, K );
<Lie algebra of dimension 14 over Rationals>
gap> UpperCentralSeries( CL );
[ <Lie algebra of dimension 14 over Rationals>,
<Lie algebra of dimension 12 over Rationals>,
<Lie algebra of dimension 11 over Rationals>,
```

```

<Lie algebra of dimension 8 over Rationals>,
<Lie algebra of dimension 6 over Rationals>,
<Lie algebra over Rationals, with 0 generators> ]

```

A.21 LeviDecomposition

LeviDecomposition(L)

The output of LeviDecomposition(L) is a list of two elements. The first element is the semisimple subalgebra and the second is the solvable radical of L . If L is solvable then the first component is the zero subalgebra. We use a similar algorithm to the one described in Section 1.4.7, but using the derived series instead of the lower central series. The reason for this is that the calculation of the derived series in many cases is faster than the computation of the lower central series. If L is a Lie algebra of characteristic $p > 0$ then it need not have a Levi decomposition. However, if it has, then this function will find one.

```

gap> L:=SimpleLieAlgebra( "E", 8, Rationals );;
gap> b:=BasisVectors( Basis( L ) );;
gap> K:=Subalgebra(L, [ b[8]+b[39]+b[69]+b[74] ]);;
gap> CL:=LieCentralizer( L, K );
<Lie algebra of dimension 68 over Rationals>
gap> LeviDecomposition( CL );
[ <Lie algebra of dimension 24 over Rationals>,
  <Lie algebra of dimension 44 over Rationals> ]

```

A.22 DirectSumDecomposition

DirectSumDecomposition(L)

If L is a direct sum of two or more ideals, then this function returns a list of these ideals, otherwise the output is a list consisting only of the element L . If L is semisimple, then the algorithms given in Chapter 4 are used. If L is defined over a large field, then the randomised method for finding a splitting element is chosen. Otherwise decomposable elements are used. If L is not semisimple, then the general algorithm described in Section 1.4.6 is used.

```

gap> K:=SimpleLieAlgebra( "A", 1, Rationals );;
<Lie algebra of dimension 3 over Rationals>
gap> L:=DirectSumOfAlgebras( K, K );;
<Lie algebra of dimension 6 over Rationals>
gap> L:=DirectSumOfAlgebras( L, K );;

```

```

<Lie algebra of dimension 9 over Rationals>
gap> DirectSumDecomposition( L );
[ <Lie algebra of dimension 3 over Rationals>,
  <Lie algebra of dimension 3 over Rationals>,
  <Lie algebra of dimension 3 over Rationals> ]

```

A.23 Property Tests for Lie Algebras

- `IsAbelianLieAlgebra(L)`
returns true if the multiplication of two elements from L always returns zero, false otherwise.
- `IsSolvable(L)`
returns true if L is solvable, false otherwise.
- `IsNilpotent(L)`
returns true if L is nilpotent, false otherwise.
- `IsRestrictedLieAlgebra(L)`
Suppose L is a Lie algebra over a field of characteristic $p > 0$. Then L is called restricted if the Lie algebra $\text{ad } L$ is closed under the p -th power map (that associates to a matrix its p -th power). This function tests whether L is restricted. Lie algebras of characteristic zero are never restricted. Furthermore, it is enough to test the property for a basis of L (see Theorem 5.11 of [32]).

```

gap> IsAbelianLieAlgebra( SimpleLieAlgebra( "A", 2, Rationals ) );
false
gap> T:=EmptySCTable( 10, 0, "antisymmetric" );;
gap> L:=LieAlgebraByStructureConstants( Rationals, T );;
gap> IsAbelianLieAlgebra( L );
true
gap> K:=SimpleLieAlgebra( "A", 2, Rationals );
<Lie algebra of dimension 8 over Rationals>
gap> IsSolvable( K );
false
gap> IsNilpotent( K );
false
gap> IsRestrictedLieAlgebra( K );
false
gap> M:=SimpleLieAlgebra( "W", [1], GF(5) );
<Lie algebra of dimension 5 over GF(5)>
gap> IsRestrictedLieAlgebra( M );

```

```

true
gap> N:=SimpleLieAlgebra( "W", [2], GF(5) );
<Lie algebra of dimension 25 over GF(5)>
gap> IsRestrictedLieAlgebra( L );
false

```

A.24 SemiSimpleType

`SemiSimpleType(L)`

If L is a Lie algebra with a nondegenerate Killing form, then `SemiSimpleType(L)` returns the type of L . It uses the method given in Chapter 5.

```

gap> L:=SimpleLieAlgebra( "E" , 8, Rationals );;
gap> b:=BasisVectors( Basis( L ) );;
gap> K:=Subalgebra( L, [ b[22]+b[23]+b[24]+b[61]+b[62] ] );;
gap> CL:=LieCentralizer( L, K );
<Lie algebra of dimension 52 over Rationals>
gap> ll:=LeviDecomposition( CL );
[ <Lie algebra of dimension 17 over Rationals>,
  <Lie algebra of dimension 35 over Rationals> ]
gap> SemiSimpleType( ll[1] );
"A1 G2"

```

A.25 AdjointAssociativeAlgebra

`AdjointAssociativeAlgebra(L)`

If L is a Lie algebra, then the matrices adx for $x \in L$ generate an associative algebra. The dimension of this algebra is in general higher than the dimension of the Lie algebra L . `AdjointAssociativeAlgebra(L)` calculates a basis of this associative algebra and returns the algebra.

```

gap> K:=SimpleLieAlgebra( "A", 1, Rationals );
<Lie algebra of dimension 3 over Rationals>
gap> AdjointAssociativeAlgebra( K );
<algebra of dimension 9 over Rationals>

```

A.26 KillingMatrix

KillingMatrix(B)

If $\{x_1, \dots, x_n\}$ is a basis of the Lie algebra L , then the matrix $(\kappa(x_i, x_j))$ is the matrix of the Killing form with respect to the basis B of L .

```
gap> K:=SimpleLieAlgebra( "A", 1, Rationals );
<Lie algebra of dimension 3 over Rationals>
gap> KillingMatrix( Basis( K ) );
[ [ 0, 4, 0 ], [ 4, 0, 0 ], [ 0, 0, 8 ] ]
```

A.27 AdjointMatrix

AdjointMatrix(B, x)

This function returns the matrix of $\text{ad } x$, with respect to the basis B of L .

```
gap> K:=SimpleLieAlgebra( "A", 1, Rationals );
<Lie algebra of dimension 3 over Rationals>
gap> b:=BasisVectors( Basis( K ) );
gap> AdjointMatrix( Basis(K), b[1] );
[ [ 0, 0, -2 ], [ 0, 0, 0 ], [ 0, 1, 0 ] ]
```

A.28 NonNilpotentElement

NonNilpotentElement(L)

This function returns an element of L that is *not* nilpotent, or **false** if no such element exists. The method described in Section 3.3 is used.

```
gap> K:=SimpleLieAlgebra( "A", 1, Rationals );
<Lie algebra of dimension 3 over Rationals>
gap> NonNilpotentElement( K );
v.3
```


Bibliography

- [1] D. W. Barnes. On Cartan subalgebras of Lie algebras. *Mathematische Zeitschrift*, 101:350–355, 1967.
- [2] R. E. Beck, B. Kolman, and I. N. Stewart. Computing the structure of a Lie algebra. In R. E. Beck and B. Kolman, editors, *Non-associative rings and algebras*, pages 167–188. Academic Press, New York, 1977.
- [3] E. R. Berlekamp. *Algebraic Coding Theory*. McGraw-Hill, New York, 1968.
- [4] J. M. Ancochea Bermudez and M. Goze. Classification des algèbres de Lie nilpotentes complexes de dimension 7. *Archiv der Mathematik*, 52:175–185, 1989.
- [5] G. Birkhoff. Representability of Lie algebras and Lie groups by matrices. *Annals of Mathematics*, 38:526–532, 1937.
- [6] N. Bourbaki. *Groupes et Algèbres de Lie, Chapitre I*. Hermann, Paris, 1971.
- [7] D. G. Cantor and H. Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, 36:587–592, 1981.
- [8] R. W. Carter. *Finite Groups of Lie Type, conjugacy classes and complex characters*. John Wiley & Sons, Chichester, 1985.
- [9] B. Champagne, W. Hereman, and P. Winternitz. The computer calculation of Lie point symmetries of large systems of differential equations. *Computer Physics Communications*, 66:319–340, 1991.
- [10] A. M. Cohen and W. A. de Graaf. Lie algebraic computation. *Computer Physics Communications*, 97:53–62, 1996.
- [11] A. M. Cohen, G. Ivanyos, and D. B. Wales. Finding the radical of an algebra of linear transformations. *Journal of Pure and Applied Algebra*, 1997.
- [12] A. M. Cohen, M. A. A. van Leeuwen, and B. Lisser. *LiE a Package for Lie Group Computations*. CAN, Amsterdam, 1992.

- [13] L. E. Dickson. *Algebras and Their Arithmetics*. University of Chicago Press, Chicago, 1923.
- [14] J. Dixmier. *Algèbres Enveloppantes*. Gauthier-Villars, Paris, Bruxelles, Montréal, 1974.
- [15] E. B. Dynkin. Semisimple subalgebras of semisimple Lie algebras. *Amer. Math. Soc. Transl.*, 6:111–244, 1957.
- [16] W. Eberly and M. Giesbrecht. Efficient decomposition of associative algebras. Technical Report 96/03, Dept. of Computer Science, University of Manitoba, jan 1996.
- [17] A. Elashvili. Centralizers of nilpotent elements in Lie algebras. *Sakharth. SSR Mecn. Akad. Math. Inst. Srom.*, 46:109–132, 1975.
- [18] G. B. Elkington. Centralizers of unipotent elements in semisimple algebraic groups. *Journal of Algebra*, 23:137–163, 1972.
- [19] R. Farnsteiner and H. Strade. *Modular Lie Algebras and Their Representations*. Marcel Dekker, New York and Basel, 1988.
- [20] W. Fulton and J. Harris. *Representation Theory*. Springer Verlag, New York, Heidelberg, Berlin, 1991.
- [21] V. P. Gerdt and V. V. Kornyak. Construction of finitely presented Lie algebras and superalgebras. *J. Symbolic Computation*, 21(3):337–349, 1996.
- [22] V. P. Gerdt and W. Lassner. Isomorphism verification for complex and real Lie algebras by Gröbner basis technique. In N. H. Ibragimov et al., editor, *Modern Group Analysis: Advanced Analytical and Computational Methods in Mathematical Physics*, pages 245–254, Dordrecht, 1993. Kluwer Academic Publishers.
- [23] W. A. de Graaf. Calculating the structure of a semisimple Lie algebra. *J. of Pure and Applied Algebra*, 117&118:319–329, 1997.
- [24] W. A. de Graaf, G. Ivanyos, A. Küronya, and L. Rónyai. Computing Levi decompositions. *Appl. Alg. in Eng., Comm. and Comp.*, to appear.
- [25] W. A. de Graaf, G. Ivanyos, and L. Rónyai. Computing Cartan subalgebras of Lie algebras. *Appl. Alg. in Eng., Comm. and Comp.*, 7(5):339–349, 1996.
- [26] D. R. Grayson and M. E. Stillman. Macaulay2. <http://www.math.uiuc.edu/Macaulay2/>.
- [27] G. Grélaud, C. Quitté, and P. Tauvel. Bases de Chevalley et $\mathfrak{sl}(2)$ -triplets des algèbres de Lie simples exceptionnelles. Technical Report 53, Université de Poitiers, 1990.

- [28] D. F. Holt and S. Rees. Testing modules for irreducibility. *Journal of the Australian Mathematical Society (Series A)*, 57:1–16, 1994.
- [29] J. E. Humphreys. *Introduction to Lie Algebras and Representation Theory*. Springer Verlag, New York, Heidelberg, Berlin, 1972.
- [30] J. E. Humphreys. *Linear Algebraic Groups*. Springer Verlag, New York, Heidelberg, Berlin, 1975.
- [31] G. Ivanyos. *Algorithms for Algebras over Global Fields*. PhD thesis, Hungarian Academy of Sciences, 1996.
- [32] N. Jacobson. *Lie Algebras*. Dover, New York, 1979.
- [33] A. I. Kostrikin. *Introduction to Algebra*. Springer Verlag, New York, Heidelberg, Berlin, 1982.
- [34] S. Lang. *Algebraic Number Theory*. Springer Verlag, New York, Heidelberg, Berlin, 1994.
- [35] M. A. A. van Leeuwen and M. Roelofs. Algorithms for constructing algebras given by generators and relations. *J. of Pure and Applied Algebra*, to appear, to appear.
- [36] A. K. Lenstra. Factorisation of polynomials. In H. W. Lenstra jr. and R. Tijdeman, editors, *Computational methods in number theory*, volume 154 of *Mathematical Centre Tract*, pages 169–198. Mathematisch Centrum, Amsterdam, 1982.
- [37] A. K. Lenstra, H. W. Lenstra jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.
- [38] R. Lidl and H. Niederreiter. *Finite Fields*, volume 20 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley, Reading, 1983.
- [39] P. J. Olver. *Applications of Lie Groups to Differential Equations*. Springer Verlag, 1993.
- [40] R. S. Pierce. *Associative Algebras*. Springer-Verlag, New York, Heidelberg, Berlin, 1982.
- [41] 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*, 109:197–246, 1988.
- [42] C. Reutenauer. *Free Lie Algebras*, volume 7 of *New Series*. Oxford University Press, Oxford, 1993.
- [43] M. Romdhani. Classification of real and complex nilpotent Lie algebras of dimension 7. *Linear and Multilinear Algebra*, 24:167–189, 1989.

- [44] L. Rónyai. Computing the structure of finite algebras. *J. of Symbolic Computation*, 9:355–373, 1990.
- [45] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27:701–717, 1980.
- [46] C. Seeley. 7-Dimensional nilpotent Lie algebras. *Transactions of the American Mathematical Society*, 335:479–496, 1993.
- [47] G. B. Seligman. *On Lie algebras of Prime Characteristic*. Memoirs of the A.M.S., No. 19, 1956.
- [48] G. B. Seligman. *Modular Lie Algebras*. Springer Verlag, New York, Heidelberg, Berlin, 1967.
- [49] D. B. Shmoys and É. Tardos. Computational complexity. In R. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, pages 1599–1645. Elsevier, Amsterdam, 1995.
- [50] D. J. Winter. *Abstract Lie Algebras*. M.I.T. Press, Cambridge, Mass., 1972.
- [51] H. Zassenhaus. Über eine Verallgemeinerung des Henselschen Lemmas. *Archiv der Mathematik*, V:317–325, 1954.
- [52] H. Zassenhaus. On the Cartan subalgebra of a Lie algebra. *Linear Algebra and its Applications*, 52/53:743–761, 1983.

Index of Terminology

- Abelian Lie algebra, 3
- absolutely simple Lie algebra, 60
- Cartan decomposition, 6
 - generalised, 40
- Cartan matrix, 55
- Cartan subalgebra, 5
- central component, 11
- centraliser, 3
- centre, 3
- decomposable element, 45
- derived series, 4
- direct sum of Lie algebras, 3
- f-algorithm, 9
- Fitting decomposition, 6
- Fitting null component, 6
- Fitting one component, 6
- fundamental system, 54
- hypercentre, 4
- ideal, 3
- idempotent, 11
 - orthogonal, 11
 - primitive, 11
- index of a Lie algebra, 89
- Jacobi identity, 2
- Killing form, 3
- Las Vegas algorithm, 8
- Levi subalgebra, 5
- Lie algebra, 2
- locally regular element, 28
- lower central series, 4
- nilpotent, 4
- nilradical, 4
- non-nilpotent element, 32
- normaliser, 4
- polynomial time algorithm, 8
- radical of an associative algebra, 11
- representation, 2
 - adjoint, 3
 - faithful, 3
- restricted Lie algebra, 5
- root, 6
- root space, 6
- root system, 6
- semidirect product, 3
- semisimple Lie algebra, 5
- simple Lie algebra, 5
- solvable, 4
- solvable radical, 4
- splitting element, 42
- structural invariant, 75
- structure constants, 7
- subalgebra, 3
- type of a semisimple Lie algebra, 51
- upper central series, 4
- weighted Dynkin diagram, 80

Index of Symbols

C_ρ , 65

$K \times I$, 3

$K \oplus I$, 3

L^k , 4

L_k , 4

$N_L(K)$, 4

S_ρ , 65

$Z(L)$, 3

$Z_L(K)$, 3

Z_k , 4

$Z_\infty(L)$, 4

$[K_1, K_2]$, 4

$[\cdot, \cdot]$, 2

$\text{NR}(L)$, 4

κ , 3

$\langle \alpha, \beta \rangle$, 54

$\text{Rad}(A)$, 11

$\text{R}(L)$, 4

c_{ij}^k , 7

e_{ij}^n , 35

Acknowledgements

Here I would like to thank everyone who contributed to this thesis. First of all I express my gratitude to my supervisor Arjeh Cohen for guiding me and my research. I thank Gábor Ivanyos, Lajos Rónyai and Alex Küronya for tolerating me as a co-author and being my host while I was in Budapest. I thank Andrea Caranti for being my host in Trento. I also thank Thomas Breuer for his patience in including my code into GAP. I thank Hans Cuypers for reading my papers. Finally I thank Arjeh Cohen, Wilberd van der Kallen and Ronald de Man for their careful reading of previous versions of the manuscript.

Samenvatting

In dit proefschrift worden algoritmen gegeven die opereren op eindig dimensionale Lie algebras gegeven door een vermenigvuldigingstabel.

In hoofdstuk 1 wordt ter inleiding kort ingegaan op de structuurtheorie van Lie algebras. Ook worden enkele algoritmen beschreven die niet aan de orde zullen komen in de latere hoofdstukken.

In hoofdstuk 2 wordt een nieuw algoritme besproken voor de berekening van het nilradicaal. Dit algoritme wordt vergeleken met enkele bekende algoritmen.

Cartan deelalgebras spelen een centrale rol in de structuurtheorie van halfnkelvoudige Lie algebras. In hoofdstuk 3 wordt ingegaan op de berekening van dergelijke deelalgebras. We geven een algoritme voor het vinden van een niet nilpotent element in een Lie algebra. Op basis hiervan wordt een algoritme voor de berekening van een Cartan deelalgebra geformuleerd. Het algoritme wordt in een praktische situatie vergeleken met enkele andere algoritmen.

Halfnkelvoudige Lie algebras splitsen als directe som van enkelvoudige Lie algebras. In hoofdstuk 4 worden algoritmen beschreven voor het bepalen van een dergelijke directe som decompositie. Het algoritme gegeven in het vorige hoofdstuk wordt gebruikt om een Cartan deelalgebra te vinden. De actie van deze deelalgebra wordt dan gebruikt om de Lie algebra te splijten. Aan het eind van het hoofdstuk worden de algoritmen met behulp van een praktisch voorbeeld vergeleken.

In hoofdstuk 5 wordt een algoritme beschreven voor de bepaling van het type van een halfnkelvoudige Lie algebra. Dit wordt gedaan door de Lie algebra te reduceren modulo een geschikt gekozen priemgetal. Het isomorfieprobleem voor halfnkelvoudige Lie algebras wordt met dit algoritme opgelost. We geven nog een oplossing van dit probleem. Deze blijkt in de praktijk echter niet goed te werken (dit in tegenstelling tot het algoritme voor de bepaling van het type).

In hoofdstuk 6 wordt ingegaan op het vinden van een eindig dimensionale representatie van een Lie algebra gegeven door een vermenigvuldigingstabel. De stelling van Ado zegt dat een dergelijke representatie bestaat, maar de bekende bewijzen bevatten geen effectieve constructie. Voor een groot gedeelte bewijzen we de stelling van Ado opnieuw. Dit levert wel een constructie die op een computer uitgevoerd kan worden.

De algoritmen beschreven in dit proefschrift zijn door de auteur geïmplementeerd in het computeralgebra systeem GAP4. Dit heeft geleid tot een pakket met de naam ELIAS (Eindhoven Lie Algebra System). In hoofdstuk 7 wordt aan de hand van twee praktische problemen geïllustreerd hoe dit pakket gebruikt kan worden. Het eerste probleem is de bepaling van isomorfie van Lie algebras. Het tweede is de berekening van de index van centralisatoren van nilpotente elementen in E_8 .

Tot slot is er een appendix met een beschrijving van ELIAS.

Curriculum Vitae

Willem de Graaf was born on August 21, 1969 in Gorinchem, The Netherlands. From 1988 to 1993 he studied mathematics and philosophy at the University of Groningen. He graduated in 1992 in pure mathematics with prof. M. van der Put as supervisor. In 1993 he graduated in technical mechanics with prof. A. Veldman as supervisor. In that same year he started his PhD research at the Technical University of Eindhoven.

Stellingen

behorende bij het proefschrift

Algorithms for Finite-Dimensional Lie Algebras

Willem A. de Graaf

1. Zij L een Lie algebra over een lichaam van karakteristiek 0. Laat x_1, \dots, x_n een basis zijn van L zo dat x_1, \dots, x_s een basis is van het centrum van L ($s \leq n$). Definieer polynomen f_1, \dots, f_n in de variabele X als volgt:

(a) $f_i = X^2$ voor $1 \leq i \leq s$,

(b) als $s < i \leq n$ dan is f_i het minimumpolynoom van $\text{ad } x_i$.

Zij I het ideaal van de universeel omhullende van L voortgebracht door de elementen $f_1(x_1), \dots, f_n(x_n)$; dan geldt in veel gevallen dat $L \cap I = 0$. Een algemeen bewijs hiervan zou tot een nieuw bewijs van de stelling van Ado leiden.

2. Het bewijs van Corollarium 4.4.1.2 in [1] is fout (het corollarium zelf is overigens correct, zie [2]).

[1] D. J. Winter. *Abstract Lie Algebras*. M.I.T. Press, Cambridge, Mass., 1972.

[2] D. W. Barnes. On Cartan Subalgebras of Lie Algebras. *Math. Z.*, 101:350–355, 1967.

3. Propositie 6 in [1] is fout; een tegenvoorbeeld is de Lie algebra met basis $\{x_1, y_1, h_1, x_2, y_2, h_2\}$ en vermenigvuldigingstabel:

$$\begin{array}{llll} [h_1, x_1] & = & y_1 & [h_2, x_1] & = & y_1 & [x_1, y_1] & = & \frac{1}{2}h_1 + \frac{1}{2}h_2 \\ [h_1, y_1] & = & -x_1 & [h_2, y_1] & = & -x_1 & [x_2, y_2] & = & \frac{1}{2}h_1 - \frac{1}{2}h_2 \\ [h_1, x_2] & = & y_2 & [h_2, x_2] & = & -y_2 & & & \\ [h_1, y_2] & = & -x_2 & [h_2, y_2] & = & x_2 & & & \end{array}$$

(niet getoonde produkten van basis elementen worden 0 verondersteld). De ruimte opgespannen door h_1 en h_2 is een Cartan deelalgebra. Een decompositie als beschreven in [1] wordt gegeven door

$$\langle h_1, h_2 \rangle \oplus \langle x_1, x_2, y_1, y_2 \rangle.$$

Deze leidt echter niet tot een directe som decompositie van de Lie algebra.

[1] W. A. de Graaf, Calculating the Structure of a Semisimple Lie Algebra, *J. of Pure and Applied Algebra*, 117&118:319–329, 1997.

4. Bij een inleidende cursus over Lie algebras verdient het boek van Jacobson ([1]) aandacht; de formuleringen van de bewijzen in dit boek zijn uitzonderlijk helder.

[1] N. Jacobson, *Lie Algebras*. Dover, New York, 1979.

5. De uitspraken van de complexiteitstheorie moet men niet in *morele* zin opvatten; “polynomialiteit” impliceert niet altijd “praktisch bruikbaar” (of *goed*) en “exponentieel” betekent niet “onbruikbaar” (of *slecht*).
6. Als men een tekst die zich in het geheugen van een computer bevindt grondig wil bestuderen, dan drukt men deze gewoonlijk af; hieruit kan men concluderen dat een mens met een beschreven blad een veel *intiemere* relatie kan onderhouden dan met een beeldscherm.
7. De Nederlanders hebben de spreuk “God zij met ons” op hun munten gezet; er is geen betere illustratie van hun koopmansgeest.
8. Als een elite spreekt van “moreel verval”, dan betekent dit veelal dat zij haar gewoontes overgenomen ziet worden door “het volk”.
9. Principes zijn veelal slechts een vrijbrief voor redeloosheid.