Technische Universiteit Eindhoven
Department of Mathematics and Computer Science

MASTER'S THESIS

Lie Algebras Generated by Extremal Elements

by
D.A. Roozemond

Supervisor: Prof. Dr. A.M. Cohen

Eindhoven, August 2005

Abstract

A Lie algebra L is a vector space over the field $\mathbb{F}$ accompanied by a bilinear map $[\cdot,\cdot]: L \times L \to L$ which is skew-symmetric (i.e. $[x,x] = 0$ for all $x \in L$) and satisfies the Jacobi identity: $[x,[y,z]] + [y,[z,x]] + [z,[x,y]] = 0$ for all $x,y,z \in L$. Lie algebras have their applications for example in physics (see for example [SW86] or [BK90]), and in the study of differential equations (see for example the PhD thesis by Jan Draisma [Dra02]).

An element $x \in L$ is called an *extremal element* if $[x,[x,L]] \subseteq \mathbb{F}x$. In this Master's thesis we study Lie algebras generated by finitely many extremal elements, building on results by Cohen, Steinbach, Ushirobira, and Wales [CSUW01]. In that paper various important properties of Lie algebras generated by extremal elements are proved, for example the fact that a Lie algebra generated by finitely many extremal elements is always finite dimensional. It is also proved that all simple Lie algebras can be generated by extremal elements. Moreover, the two and three extremal generator cases are extensively studied.

In this Master's thesis we continue with the four and five extremal generator cases. Let $\mathcal{S}_n$ be the set of Lie algebras generated by n extremal elements. Cohen et al. proved that both $\mathcal{S}_2$ and $\mathcal{S}_3$ contain a semi-simple Lie algebra of maximal dimension, A_1 and A_2, respectively. We find that also $\mathcal{S}_4$ contains a semi-simple Lie algebra of maximal dimension, namely D_4. However, we prove that the maximal dimension occurring in $\mathcal{S}_5$ is 537, and that no 537-dimensional semi-simple Lie algebra occurs in $\mathcal{S}_5$.

Moreover, we study degenerate cases of Lie algebras generated by four or five extremal elements, i.e. instances where one or more pairs of the generating extremal elements commute. Lastly, we show how the simple Lie algebras A_n and C_n can be generated by extremal elements.

Contents

Chapter 1

Introduction

There is no branch of mathematics, however abstract, which may not someday be applied to the phenomena of the real world.
– Nicolai Lobachevsky (1793-1856) [MQ]

A Lie algebra L is a vector space over the field $\mathbb{F}$ accompanied by a bilinear map $[\cdot,\cdot] : L \times L \to L$ which is skew-symmetric (i.e. $[x,x] = 0$ for all $x \in L$), and which satisfies the Jacobi identity:

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

Lie algebras have their applications for example in physics (see for example [SW86] or [BK90]), and in the study of differential equations (see for example the PhD thesis by Jan Draisma [Dra02]). To get a feeling for the applications in physics it might be worth skimming through Chapter 4 of [SW86]. Section 1.2 of [Dra02] provides an excellent overview of the connection between ordinary differential equations and finite dimensional Lie algebras.

An element $x \in L$ is called an *extremal element* if we have $[x,[x,y]] \in \mathbb{F}x$ for each $y \in L$. In this Master's thesis we study Lie algebras generated by finitely many extremal elements, especially those generated by four or five extremal elements. Applications for extremal elements stem from the fact that long root elements are extremal in Lie algebras of Chevalley type. They were used by Chernousov [Che89] in his proof of the Hasse principle for E_8. Sandwich elements, elements $x \in L$ with $[x,[x,y]] = 0$ for all $y \in L$, are a special kind of extremal elements, with an application in Lie algebras over fields of small characteristic [PS97].

In order to study Lie algebras generated by extremal elements we start with small cases, i.e. Lie algebras generated by a few extremal elements, and try to find patterns. Cohen et al. extensively study the two and three generator case [CSUW01], derive various important basic properties of Lie algebras generated by extremal elements, and give bounds on the number of extremal elements required to generate simple Lie algebras. In this Master's thesis we build on their results, and extensively study the four and five generator cases. We prove that the five generator case significantly differs from the smaller cases (see Section 8.1), and we show how two of the four classical Lie algebras are generated by extremal elements (see Chapter 9).

Chapter 2 of this report contains a general introduction to Lie algebras (inspired mainly by [Hal03] and [Hum72]) and Chapter 3 contains a general introduction to extremal elements. Though this chapter was mainly inspired by [CSUW01], Section 3.4 contains a new result. In Chapters 4 and 5 we consider Lie algebras generated by two and three extremal elements, respectively. These chapters serve both as an overview of the results of [CSUW01] and as an introduction to the following chapters.

In Chapter 6 we consider Lie algebras generated by four extremal elements. The first part of this chapter contains results from [CSUW01], but Sections 6.2 through 6.5 are new. In Chapter 7 we give an overview of the algorithms introduced in the previous chapter, and we introduce a new algorithm. Chapter 8 contains the analysis of Lie algebras generated by five extremal elements, almost entirely composed of new results. Especially the extensive analysis of degenerate cases in Section 8.2 appears to be new information. This analysis is a result of the algorithms described in Chapter 7.

Lastly, Chapter 9 contains three theorems and three conjectures on Lie algebras generated by arbitrary many extremal elements. A conclusion and some recommendations can be found in Chapter 10.

Unless mentioned otherwise, we work over fields of characteristic 0. To avoid confusion, we write 'generated' if and only if we mean 'generated as a Lie algebra', and we write 'spanned' if and only if we mean 'linearly generated'.

Following good practice, I end this introduction with some acknowledgements. First of all, I would like to thank Prof. Dr. A.M. Cohen, my supervisor, for the inspiring conversations, his valuable suggestions, and the thorough remarks he gave on proofs and various versions of this report. Furthermore, my acknowledgements go to Dr. F.G.M.T. Cuypers and Prof. Dr. Ir. J. de Graaf for being members of my graduation committee and for their useful comments on earlier versions of this report. I should also thank Willem de Graaf (University of Trento) for the short yet clarifying e-mail discussion we had about his work on this particular subject.

Dan Roozemond, August 2005

Chapter 2

Lie Algebras - An Introduction

This chapter presents an introduction to Lie algebras in general. Many great books exist on this topic, for instance the classic 'Introduction to Lie Algebras and Representation Theory' by James E. Humphreys [Hum72] or 'Lie Algebras' by Nathan Jacobson [Jac62]. For a thorough introduction one might also consult [Hal03] or [Var84]. Some very nice notes on the topic were written by Serre [Ser87], and for an introduction a bit more focussed on applications refer to [SW86]. This chapter was mainly inspired by [Hal03] and [Hum72].

2.1 Definition

Because this chapter provides an introduction to Lie algebras, it is only logical to start by defining the notion *Lie algebra*.

Definition 2.1. (Lie Algebra) A *finite-dimensional Lie Algebra* is a finite-dimensional vector space L over a field $\mathbb{F}$ together with a map $[\cdot,\cdot] : L \times L \to L$, with the following properties:

1. $[\cdot,\cdot]$ is bilinear: $[x+v,y] = [x,y]+[v,y]$ for all $x,v,y \in L$ and $[\alpha x,y] = \alpha[x,y]$ for all $x,y \in L$ and $\alpha \in \mathbb{F}$, $[x,y+w] = [x,y]+[x,w]$ for all $x,y,w \in L$ and $[x,\beta y] = \beta[x,y]$ for all $x,y \in L$ and $\beta \in \mathbb{F}$,

2. $[\cdot,\cdot]$ is skew-symmetric: $[x,x] = 0$ for all $x \in L$,

3. The Jacobi identity holds: $[x,[y,z]]+[y,[z,x]]+[z,[x,y]] = 0$ for all $x,y,z \in L$.

In order to get somewhat acquainted with the notion of a Lie algebra, we first give two examples.

Example 2.2. The first example is $\mathbb{R}^3$ with $[x,y] := x \times y$, the vector product, i.e.

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \times \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} x_2y_3 - x_3y_2 \\ x_3y_1 - x_1y_3 \\ x_1y_2 - x_2y_1 \end{pmatrix}. \tag{2.1}$$

It is straightforward to verify that indeed $[x,y]$ is bilinear and skew-symmetric. Verification of the Jacobi identity is straightforward as well.

Example 2.3. A whole range of examples can be found as follows. Let V be a vector space, and L the ring of all linear transformations $V \to V$. Define $[X,Y] = XY - YX$ for each $X, Y \in L$. Obviously, $[\cdot,\cdot]$ is bilinear, and $[X,X] = 0$. For the Jacobi identity we have:

$$\begin{gathered}
[X,[Y,Z]] + [Y,[Z,X]] + [Z,[X,Y]] = \\
[X, YZ - ZY] + [Y, ZX - XZ] + [Z, XY - YX] = \\
XYZ - XZY - YZX + ZYX + YZX - YXZ + \\
-ZXY + XZY + ZXY - ZYX - XYZ + YXZ = 0.
\end{gathered}$$

So indeed this gives us a Lie algebra, which we will call the *general linear algebra* $\mathfrak{gl}(V)$. It is clear that the above can be done for any associative algebra L.

Note that the first and second property in Definition 2.1 together imply $[x,y] = -[y,x]$ for all $x, y \in L$:

$$0 = [x+y, x+y] = [x,x] + [x,y] + [y,x] + [y,y] = [x,y] + [y,x]. \tag{2.2}$$

We define two more notions on Lie algebras.

Definition 2.4. Let L_1 and L_2 be Lie algebras. A linear map $\varphi : L_1 \to L_2$ is called a **Lie algebra homomorphism** if $\varphi([x,y]) = [\varphi(x), \varphi(y)]$ for all $x, y \in L$. If a Lie algebra homomorphism φ is a bijection, then φ is called a **Lie algebra isomorphism**. A Lie algebra homomorphism $\varphi : L \to L$ is called a **Lie algebra automorphism.**

If L is a Lie algebra, then we define for any $\varphi : L \to \mathfrak{gl}(L)$ that $[\varphi(x), \varphi(y)] = \varphi(x)\varphi(y) - \varphi(y)\varphi(x)$, analogous to Example 2.3.

Definition 2.5. (Ad) Let L be a Lie algebra. For $x \in L$, we define a linear map $\mathrm{ad}_x : L \to L$ by

$$\mathrm{ad}_x(y) = [x,y]. \tag{2.3}$$

Thus ad (i.e. the map $x \mapsto \mathrm{ad}_x$) is in fact a linear map from L into the space of linear operators from L to L.

The ad function is useful in the sense that it makes things definitely more readable: instead of writing $[x,[x,[x,[x,[x,y]]]]]$ we will now simply write $(\mathrm{ad}_x)^5(y)$. We have the following useful property for the ad function:

Lemma 2.6. *If L is a Lie algebra, then ad is a Lie algebra homomorphism from L to $\mathfrak{gl}(L)$.*

Proof Let L be a Lie algebra, and let $x, y \in L$. Then we have, for every $z \in L$, (by the Jacobi identity):

$$\begin{aligned}
\mathrm{ad}_{[x,y]}(z) &= [[x,y],z] \\
&= -[z,[x,y]] \\
&= [x,[y,z]] + [y,[z,x]] \\
&= [x,[y,z]] - [y,[x,z]] \\
&= \mathrm{ad}_x\mathrm{ad}_y(z) - \mathrm{ad}_y\mathrm{ad}_x(z) \\
&= (\mathrm{ad}_x\mathrm{ad}_y - \mathrm{ad}_y\mathrm{ad}_x)(z) \\
&= [\mathrm{ad}_x, \mathrm{ad}_y](z),
\end{aligned} \tag{2.4}$$

so indeed $\mathrm{ad}_{[x,y]} = [\mathrm{ad}_x, \mathrm{ad}_y]$, so ad is a homomorphism. $\square$

Definition 2.7. (Derivation) A *derivation* d is a Lie algebra homomorphism satisfying

$$d([x,y]) = [d(x), y] + [x, d(y)].$$

If we pick an image of ad in $\mathfrak{gl}(L)$, we see that it acts as a derivation on L:

Lemma 2.8. *If L is a Lie algebra and $x \in L$, then ad_x is a derivation, i.e.* $\mathrm{ad}_x([y,z]) = [y, \mathrm{ad}_x(z)] + [\mathrm{ad}_x(y), z]$.

Proof Let $x \in L$, and observe the action of ad_x on $[y,z]$ for $y, z \in L$:

$$\begin{aligned} \mathrm{ad}_x([y,z]) &= [x,[y,z]] \\ &= [y,[x,z]] - [z,[x,y]] \\ &= [y, \mathrm{ad}_x(z)] - [z, \mathrm{ad}_x(y)] \\ &= [y, \mathrm{ad}_x(z)] + [\mathrm{ad}_x(y), z]. \end{aligned} \tag{2.5}$$

□

Furthermore, we have the following notion:

Definition 2.9. (Monomial) A *monomial of length s* is a bracketing of the form

$$[x_1, [x_2, [x_3, \ldots [x_{s-1}, x_s] \ldots]]].$$

If we consider a Lie algebra generated by elements G, we usually take monomials to be those bracketings where $x_1, \ldots, x_s \in G$. Furthermore, we then have the notion of *reducible monomial*: A monomial is called reducible if it is a linear combination of monomials of strictly smaller length.

2.2 Representations

Again, let L be a Lie algebra over the field $\mathbb{F}$, and V a vector space over $\mathbb{F}$.

Definition 2.10. (Representation) A *representation* of L in V is a map $\varphi : L \to \mathrm{End}(V)$ such that

- φ is linear, and
- $\varphi([x,y]) = \varphi(x)\varphi(y) - \varphi(y)\varphi(x)$ for all $x, y \in L$.

If V is finite dimensional, the above is equivalent to saying that φ is a homomorphism of L into $\mathfrak{gl}(V)$. A well known representation is the *adjoint representation* of a Lie algebra: $\varphi : L \to \mathfrak{gl}(L)$, $x \mapsto \mathrm{ad}_x$.

There is a well-known result on representations of finite dimensional Lie algebras:

Theorem 2.11. ***(Ado's Theorem)*** *Every finite dimensional Lie algebra L of characteristic zero has a faithful finite dimensional representation.*

The proof of this theorem is beyond the scope of this document. It can be found in [Jac62, Chapter VI].

2.3 Ideals

Note that we defined $\mathfrak{gl}(V)$ as the set of linear transformations of a vector space V to itself, viewed as a Lie algebra (see Example 2.3).

Definition 2.12. (Linear Lie Algebra) A Lie algebra L is called a *linear Lie algebra* if it is isomorphic to a subalgebra of $\mathfrak{gl}(V)$ for some vector space V.

> **Example 2.13.** Since $\mathfrak{gl}(V)$ is always finite dimensional, any infinite dimensional Lie algebra is an example of a nonlinear Lie algebra. Take for example the Lie algebra $L = \langle a, b \rangle$, where we may create strings of arbitrary length consisting of a's and b's.

The notion of ideals in rings extends to ideals in Lie algebras:

Definition 2.14. (Ideal) A subspace I of a Lie algebra L is called an *ideal* if $[x, y] \in I$ if $x \in L$ and $y \in I$.

> **Example 2.15.** Let L be a Lie algebra. Obviously, $\{0\}$ and L are ideals.
>
> A more interesting example is the *center* of L:
>
> $$Z(L) = \{z \in L \mid [x, z] = 0 \text{ for all } x \in L\}. \tag{2.6}$$
>
> Indeed, if we let $y \in L$ and $z \in Z(L)$, then $[y, z] = 0 \in Z(L)$.

Lemma 2.16. *If I and J are both ideals of L, then $I + J = \{x + y \mid x \in I, y \in J\}$ is an ideal, and so is $[I, J] = \{\Sigma[x_i, y_i] \mid x_i \in I, y_i \in J\}$.*

Proof Let I and J be ideals of a Lie algebra L. It is straightforward that $I + J$ is an ideal of L, so we focus on $[I, J]$. Let $y \in L$ and $z \in [I, J]$, so $z = [a_1, b_1] + \ldots + [a_t, b_t]$, with $a_i \in I$ and $b_i \in J$. Then, by bilinearity,

$$[y, z] = [y, [a_1, b_1]] + \ldots + [y, [a_t, b_t]]. \tag{2.7}$$

By Jacobi, we have for every term of this equation,

$$[y, [a_i, b_i]] = [a_i, [y, b_i]] + [b_i, [a_i, y]] = [a_i, [y, b_i]] + [[y, a_i], b], \tag{2.8}$$

and both of these terms are of the form $[a, b]$ with $a \in I$ and $b \in J$. So indeed $[y, z] \in [I, J]$, as desired. □

A special case of the latter construction is $[L, L]$, the *derived algebra* of L. We end this section with one more definition.

Definition 2.17. (Abelian) A Lie algebra L is called Abelian if $[L, L] = 0$.

2.4 Simple Lie Algebras

A very important property of a Lie algebra is the following:

Definition 2.18. (Simple Lie Algebra) A Lie algebra L is said to be simple if $[L, L] \neq 0$ and L has no ideals except $\{0\}$ and L itself.

Corollary 2.19. If we consider ad as a map from L into the space of linear operators on L, we see that its kernel is equal to the center of L:

$$\ker(\mathrm{ad}) = \{x \in L \mid \mathrm{ad}_x(y) = 0 \text{ for all } y \in L\} = Z(L). \tag{2.9}$$

So if L is a simple Lie algebra, $\ker(\mathrm{ad}) = \{0\}$, hence ad is an isomorphism of L to $\mathfrak{gl}(L)$, so *any simple Lie algebra is a linear Lie algebra.*

> **Example 2.20.** Let V be a vector space over $\mathbb{F}$ of dimension n. Recall that the trace of a matrix M is the sum of its diagonal elements, commonly denoted by $\mathrm{tr}(M)$, and independent of the choice of basis. Then we let $\mathfrak{sl}(V)$ (or $\mathfrak{sl}_n(\mathbb{F})$ if $V = \mathbb{F}^n$) denote the set of endomorphisms of V having trace zero.
>
> Since $\mathrm{tr}(x + y) = \mathrm{tr}(x) + \mathrm{tr}(y)$ and $\mathrm{tr}(xy) = \mathrm{tr}(yx)$, we know that $\mathfrak{sl}(V)$ is a subalgebra of $\mathfrak{gl}(V)$. It is called the *special linear algebra*. It is easy to see that the dimension of $\mathfrak{sl}(V)$ is $n^2 - 1$.

It can be proved that only a very limited number of classes of simple Lie algebras exist. Over fields of characteristic 0 there exist four families called the *classical Lie algebras*, and five *exceptional* simple Lie algebras. In this report we limit ourselves to giving a list of these algebras. The accompanying proof, however, is certainly worth reading: consult for instance Chapter III of [Hum72].

The four families of classical Lie algebras are:

- A_n ($n \geq 1$): The Lie algebra of the special linear group in $n+1$ variables, also denoted $\mathfrak{sl}_{n+1}$, and most commonly represented by all $(n+1) \times (n+1)$ matrices with trace 0 (see Example 2.20). It is easy to see that this Lie algebra has dimension $(n+1)^2 - 1$.

- C_n ($n \geq 3$): The Lie algebra of the symplectic group in $2n$ variables, also denoted $\mathfrak{sp}_{2n}$. We let V be a vector space of dimension $2n$, and denote its elements as row vectors. We define the non-degenerate bilinear form g on V by the matrix G:
$$G = \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix}. \tag{2.10}$$
It is easy to see that g is an skew-symmetric bilinear function into $\mathbb{F}$. Now $\mathfrak{sp}_{2n}$ consists by definition of all endomorphisms x of V satisfying $g(x(v), w) = -g(v, x(w))$. It is not hard to see that the dimension of $\mathfrak{sp}_{2n}$ is $n(2n+1)$.

- B_n ($n \geq 2$): The Lie algebra of the special orthogonal group in $2n+1$ variables, also denoted $\mathfrak{o}_{2n+1}$. Similar to the previous case, we let V be a vector space of dimension $2n+1$, and define the non-degenerate bilinear form g on V by the matrix G:
$$G = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & I_n \\ 0 & I_n & 0 \end{pmatrix}. \tag{2.11}$$
Now $\mathfrak{o}_{2n+1}$ consists by definition of all endomorphisms x of V satisfying $g(x(v), w) = -g(v, x(w))$. Its dimension is $n(2n+1)$.

- D_n ($n \geq 4$): The Lie algebra of the special orthogonal group in $2n$ variables, also denoted $\mathfrak{o}_{2n}$. This Lie algebra is defined in the same way as $\mathfrak{o}_{2n+1}$, only V has dimension $2n$ again and G has the simpler form
$$G = \begin{pmatrix} 0 & I_n \\ I_n & 0 \end{pmatrix}. \tag{2.12}$$
The dimension of $\mathfrak{o}_{2n}$ is $n(2n-1)$.

Note that we could also define B_n and C_n for $n \geq 1$ and D_n for $n \geq 3$, but to avoid repetitions (because $A_1 = B_1 = C_1$, $B_2 = C_2$, and $A_3 = D_3$) we usually use the numbering above.

The exceptional Lie algebras are denoted by G_2 (of dimension 14), F_4 (of dimension 52), E_6 (of dimension 78), E_7 (of dimension 133), and E_8 (of dimension 248).

2.5 Solvability and Nilpotency

A given Lie algebra could be *solvable* or *nilpotent*:

Definition 2.21. (Solvability) Let L be a Lie algebra. We define a sequence of ideals of L by
$$L^{(0)} = L, L^{(1)} = [L, L], L^{(2)} = [L^{(1)}, L^{(1)}] = [[L, L], [L, L]], L^{(3)} = [L^{(2)}, L^{(2)}], \ldots. \tag{2.13}$$
L is called *solvable* if $L^{(n)} = 0$ for some n.

So from the definitions we immediately see that Abelian Lie algebras are always solvable, and simple Lie algebras are never solvable.

Definition 2.22. (Nilpotency) Let L be a Lie algebra. We define a sequence of ideals of L by

$$L^0 = L, L^1 = [L, L], L^2 = [L, L^1] = [L, [L, L]], L^3 = [L, L^2], \ldots. \tag{2.14}$$

L is called *nilpotent* if $L^n = 0$ for some n.

> **Example 2.23.** We let L be the Lie algebra over the field $\mathbb{F}$ generated by a, b, and c, such that $[a, b] = [a, c] = a$ and $[b, c] = 0$. It is easy to see that this Lie algebra satisfies the Jacobi identity:
>
> $$[a, [b, c]] + [b, [c, a]] + [c, [a, b]] = 0 - [b, a] + [c, a] = - - a + -a = 0. \tag{2.15}$$
>
> Furthermore, $[L, L] = \mathbb{F}a$ so $[[L, L], [L, L]] = 0$ and L is solvable. However, L is not nilpotent: for example, $[b, [b, [\ldots, [b, a]]]]$ is nonzero for any arbitrary number of b's in front.

Similarly, an ideal I is called solvable if $I^{(n)} = 0$ for some n, and it is called nilpotent if $I^n = 0$ for some n. It is easy to see that every nilpotent Lie algebra (ideal) is solvable, since $[L, L] \subseteq L$. A solvable Lie algebra (ideal), however, is not necessarily nilpotent.

Definition 2.24. (Radical) Let L be a Lie algebra. The *radical* of L, denoted by $\mathrm{Rad}(L)$, is the largest solvable ideal of L.

Definition 2.25. (Nilradical) Similarly, we define the nilradical of L, denoted by $\mathrm{NilRad}(L)$, as the largest nilpotent ideal of L.

It is straightforward to see that a Lie algebra has a unique radical. Suppose I and J are solvable ideals of a Lie algebra L. Then $J/(I \cap J)$ is solvable (because J is solvable), so $(I + J)/I$ is solvable (because $(I + J)/I \cong J/(I \cap J)$), so $I + J$ is solvable. So there is a unique solvable ideal containing all solvable ideals of L, which is exactly the radical of L. With a similar argument we see that the nilradical of a Lie algebra is unique.

Definition 2.26. (Semi-simple Lie Algebra) A Lie algebra L is said to be *semi-simple* if $\mathrm{Rad}(L) = 0$.

Theorem 2.27. *Let L be a Lie algebra. Then $L/\mathrm{Rad}(L)$ is semi-simple.*

Proof Let φ be the natural map of L onto $L/\mathrm{Rad}(L)$. If I is a solvable ideal of $L/\mathrm{Rad}(L)$, then $\varphi^{-1}(I)$ must be a solvable ideal of L, and we have $\mathrm{Rad}(L) \subseteq \varphi^{-1}(I)$. Moreover, by maximality of $\mathrm{Rad}(L)$ we have $\varphi^{-1}(I) \subseteq \mathrm{Rad}(L)$, hence $\mathrm{Rad}(L) = \varphi^{-1}(I)$. This shows that $I = 0$, hence L is semi-simple. □

We end this section with a very important theorem.

Theorem 2.28. *L is semi-simple if and only if it is isomorphic to a direct sum of simple Lie algebras.*

The proof of this theorem is beyond the scope of this report. Consult for instance Section 2 of [Hum72] or Chapter 6 of [Hal03]. It should be noted that in some books semi-simplicity is defined as in Theorem 2.28, and then Definition 2.26 is proved as a property.

2.6 Universal Enveloping Algebras

Now we arrive at the notion *universal enveloping algebra*.

Definition 2.29. (Universal Enveloping Algebra) Let L be a Lie algebra over the field $\mathbb{F}$. A pair $(\mathfrak{U}, \varphi)$, where $\mathfrak{U}$ is an associative algebra over $\mathbb{F}$ and φ is a linear mapping of L into $\mathfrak{U}$, is called a *universal enveloping algebra* of L if the following conditions are satisfied:

- $\varphi(L)$ generates $\mathfrak{U}$,
- $\varphi([x, y]) = \varphi(x)\varphi(y) - \varphi(y)\varphi(x)$ for all $x, y \in L$, and
- If $\mathfrak{U}'$ is any associative algebra and φ' is any linear map of L into $\mathfrak{U}'$ such that $\varphi'([x, y]) = \varphi'(x)\varphi'(y) - \varphi'(y)\varphi'(x)$ for all $x, y \in L$, then there exists a (unique) homomorphism θ of $\mathfrak{U}$ into $\mathfrak{U}'$ such that $\varphi'(x) = \theta(\varphi(x))$ for all $x \in L$.

One special universal enveloping algebra is the following. Let L be a Lie algebra and let T be the tensor algebra over the underlying vector space of L. For $x, y \in L$ we let

$$u_{x,y} = x \otimes y - y \otimes x - [x, y]. \tag{2.16}$$

By $\mathcal{L}$ we denote the two sided ideal $\sum_{x,y \in L} T \otimes u_{x,y} \otimes T$. We introduce the quotient algebra $U = T/\mathcal{L}$, and let φ be the natural homomorphism of T onto U. Since L generates T, we have that $\varphi(L)$ generates U. It is straightforward to verify that (U, φ) is indeed a universal enveloping algebra of L. It is a bit more elaborate to prove that φ is in fact injective, see [Hum72, Section 17.2].

Because φ is injective on L, it is possible to identify L with its image $\varphi(L)$ in U. With this identification, U will be called *the universal enveloping algebra of L*.

Chapter 3

Extremal Elements

3.1 Introduction

Definition 3.1. (Extremal Element) A nonzero element x of a Lie algebra L over $\mathbb{F}$ is called an *extremal element* if $[x, [x, L]] \subseteq \mathbb{F}x$.

For extremal elements, ad has a very nice property: Let $x \in L$ be an extremal element, then, for any $y \in L$, $\mathrm{ad}_x^3(y) = [x, [x, [x, y]]] = [x, \alpha x]$ for a certain $\alpha \in \mathbb{F}$, so $\mathrm{ad}_x^3(y) = \alpha[x, x] = 0$. We say that x is ad-*nilpotent* of order at most 3.

In the remainder of this section we will study Lie algebras generated by extremal elements, using [CSUW01]. We will assume that L is a Lie algebra over the field $\mathbb{F}$ of characteristic not 2, and start with some general properties of extremal elements.

Since $[x, [x, y]]$ is linear in y, we know that an element $x \in L$ is extremal if and only if a linear functional $f_x : L \to K, y \mapsto f_x(y)$ exists, such that

$$[x, [x, y]] = f_x(y)x \text{ for all } y \in L. \tag{3.1}$$

Note that, (in characteristic not equal to 2) if x and y commute, we have $[x, y] = [y, x]$, and by Definition 2.1 we have $[x, y] = -[y, x]$, so $[x, y] = [y, x] = 0$. This implies $f_x(y) = 0$.

Following [CSUW01], we will write $\mathcal{E}$ for the set of non-zero extremal elements in L.

Lemma 3.2. *If $x, y \in \mathcal{E}$, then $f_x(y) = f_y(x)$.*

Proof Let $x, y \in \mathcal{E}$. We will compute $[y, [x, [x, y]]]$ in two different ways.
On the one hand,

$$[y, [x, [x, y]]] = [y, f_x(y)x] = f_x(y)[y, x], \tag{3.2}$$

but on the other hand (by the Jacobi identity),

$$\begin{aligned} [y, [x, [x, y]]] &= -[x, [[x, y]], y] - [[x, y], [y, x]] \\ &= -[x, [y, [y, x]]] \\ &= -[x, f_y(x)y] \\ &= f_y(x)[y, x]. \end{aligned} \tag{3.3}$$

So, provided that x and y do not commute, we may conclude $f_x(y) = f_y(x)$. If x and y commute, we have $f_x(y) = 0 = f_y(x)$, as noted above. □

We have some nice relations for calculations involving extremal elements. At least the first two rules go back to Premet. In the remainder of this report we will refer to them as (P1) to (P5).

Lemma 3.3. *Let L be a Lie algebra over a field of characteristic not 2. Let $x \in \mathcal{E}$ and $y, z \in L$. Then:*

(P1) $2[[x,y],[x,z]] = f_x([y,z])x + f_x(z)[x,y] - f_x(y)[x,z]$,

(P2) $2[x,[y,[x,z]]] = f_x([y,z])x - f_x(z)[x,y] - f_x(y)[x,z]$.

Now let $x, y \in \mathcal{E}$ and $z \in L$.

(P3) $2[[x,y],[x,[y,z]]] = f_y(z)f_x(y)x + f_x([y,z])[x,y] - f_x(y)[x,[y,z]]$,

(P4) $2[[x,y],[[x,y],z]] = (f_x([y,z]) - f_y([x,z]))[x,y] + f_x(y)(f_x(z)y + f_y(z)x - [y,[x,z]] - [x,[y,z]])$.

Now let $x, y, z \in \mathcal{E}$.

(P5) $2[[x,[y,z]],[y,[x,z]]] = -\frac{1}{2}\left(f_y(z)f_x([y,z])x + f_x([y,z])f_x(z)y + f_x([y,z])f_x(y)z\right) - f_y(z)f_x(z)[x,y] + f_y(z)f_x(y)[x,z] - f_x(z)f_x(y)[y,z]$.

Proof For sake of completeness we provide the straightforward proofs for these relations. Firstly, let $x \in \mathcal{E}$ and $y, z \in L$. Then, using the Jacobi identity, we find

$$\begin{aligned} [[x,y],[x,z]] &= -[[x,z],[x,y]] \\ &= [x,[y,[x,z]]] + [y,[[x,z],x]] \\ &= [x,[y,[x,z]]] + f_x(y)[x,y]. \end{aligned} \tag{3.4}$$

Similarly, interchanging y and z, we find

$$[[x,y],[x,z]] = -[x,[z,[x,y]]] - f_x(y)[x,z]. \tag{3.5}$$

Furthermore, again by the Jacobi identity,

$$[x,[z,[x,y]]] = -[x,[x,[y,z]]] - [x,[y,[z,x]]] = -f_x([y,z])x + [x,[y,[x,z]]], \tag{3.6}$$

so

$$0 = [x,[z,[x,y]]] + f_x([y,z])x - [x,[y,[x,z]]]. \tag{3.7}$$

Adding Equations 3.4, 3.5, and 3.7 yields

$$2[[x,y],[x,z]] = f_x([y,z])x + f_x(z)[x,y] - f_x(y)[x,z], \tag{3.8}$$

which proves the first identity.

Inserting the first identity into the result of Equation 3.6 yields:

$$\begin{aligned} 2[x,[y,[x,z]]] &= 2[[x,y],[x,z]] - 2f_x(z)[x,y] \\ &= f_x([y,z])x - f_x(z)[x,y] - f_x(y)[x,z], \end{aligned} \tag{3.9}$$

which proves the second identity.

Now let $x, y \in \mathcal{E}$ and $z \in L$. Substituting $[y,z]$ for z in the first identity gives

$$\begin{aligned} 2[[x,y],[x,[y,z]]] &= f_x([y,[y,z]])x + f_x([y,z])[x,y] - f_x(y)[x,[y,z]] \\ &= f_y(z)f_x()x + f_x([y,z])][x,y] - f_x(y)[x,[y,z]], \end{aligned} \tag{3.10}$$

which proves the third identity. For the last identity, we first apply the Jacobi identity, and then apply the third identity twice (using that $f_x(y) = f_y(x)$ when both x and y are extremal elements):

$$2[[x,y],[[x,y],z]] = -2[[x,y],[z,[x,y]]]$$

$$
\begin{aligned}
&= 2[[x,y],[x,[y,z]]] + 2[[x,y],[y,[z,x]]]\\
&= 2[[x,y],[x,[y,z]]] + 2[[y,x],[y,[x,z]]]\\
&= f_y(z)f_x(y)x + f_x([y,z])[x,y] - f_x(y)[x,[y,z]]\\
&\quad + f_x(z)f_y(x)y + f_y([x,z])[y,x] - f_y(x)[y,[x,z]]\\
&= f_x(y)\left(f_y(z)x - [x,[y,z]] + f_x(z)y - [y,[x,z]]\right)\\
&\quad + \left(f_x([y,z]) - f_y([x,z])\right)[x,y].
\end{aligned} \tag{3.11}
$$

For the last identity, we let $x,y,z \in \mathcal{E}$, and apply Jacobi to see that

$$
\begin{aligned}
[[x,[y,z]],[y,[x,z]]] &= -[[y,[x,z]],[x,[y,z]]]\\
&= [x,[[y,z],[y,[x,z]]]] - [[y,z],[x,[y,[x,z]]]].
\end{aligned} \tag{3.12}
$$

For the first term, we have

$$
\begin{aligned}
&[x,[[y,z],[y,[x,z]]]] =\\
&= [x, \tfrac{1}{2}f(y,[z,[x,z]])y + \tfrac{1}{2}f(y,[x,z])[y,z] - \tfrac{1}{2}f(y,z)[y,[x,z]]]\\
&= -\tfrac{1}{2}f(y,z)f(x,z)[x,y] - \tfrac{1}{2}f(x,[y,z])[x,[y,z]] - \tfrac{1}{2}f(y,z)[x,[y,[x,z]]]\\
&= -\tfrac{1}{2}f(y,z)f(x,z)[x,y] - \tfrac{1}{2}f(x,[y,z])[x,[y,z]]\\
&\quad -\tfrac{1}{4}f(y,z)f(x,[y,z])x + \tfrac{1}{4}f(y,z)f(x,y)[x,z] + \tfrac{1}{4}f(y,z)f(x,z)[x,y].
\end{aligned}
$$

For the second term, we have

$$
\begin{aligned}
&-[[y,z],[x,[y,[x,z]]]] =\\
&= -[[y,z], \tfrac{1}{2}f(x,[y,z])x - \tfrac{1}{2}f(x,y)[x,z] - \tfrac{1}{2}f(x,z)[x,y]]\\
&= \tfrac{1}{2}f(x,[y,z])[x,[y,z]]\\
&\quad -\tfrac{1}{4}f(x,y)f(x,[y,z])z + \tfrac{1}{4}f(x,y)f(y,z)[x,z] - \tfrac{1}{4}f(x,y)f(x,z)[y,z]\\
&\quad -\tfrac{1}{4}f(x,z)f(x,[y,z])y - \tfrac{1}{4}f(x,y)f(x,z)[y,z] - \tfrac{1}{4}f(x,z)f(y,z)[x,y].
\end{aligned}
$$

Adding the two terms, we find

$$
\begin{aligned}
&[[x,[y,z]],[y,[x,z]]] =\\
&= -\tfrac{1}{4}f(x,[y,z])f(y,z)x - \tfrac{1}{4}f(x,[y,z])f(x,z)y + \tfrac{1}{4}f(x,[y,z])f(x,y)z\\
&\quad -\tfrac{1}{2}f(x,z)f(y,z)[x,y] + \tfrac{1}{2}f(x,y)f(y,z)[x,z] - \tfrac{1}{2}f(x,y)f(x,z)[y,z],
\end{aligned}
$$

which is the statement to be proved.

This completes the proof of Lemma 3.3. □

Definition 3.4. (Exponential) We define, for $x \in \mathcal{E}$ and $s \in \mathbb{F}$, the following function:

$$
\exp(x,s) := 1 + s\mathrm{ad}_x + \frac{1}{2}s^2\mathrm{ad}_x^2. \tag{3.13}
$$

Since $\mathrm{ad}_x^3 = 0$ if $x \in \mathcal{E}$, this is the exponential of the derivation $s\mathrm{ad}_x$.

We verify that $\exp(x,s)$ is an automorphism of L. Firstly, note that $[\mathrm{ad}_x^2(y), \mathrm{ad}_x^2(z)] = f_x(y)f_x(z)[x,x] = 0$ and

$$
\begin{aligned}
[\mathrm{ad}_x(y), \mathrm{ad}_x^2(z)] + [\mathrm{ad}_x^2(y), \mathrm{ad}_x(z)] &= [[x,y],[x,[x,z]]] + [[x,[x,y]],[x,z]]\\
&= [[x,y], f_x(z)x] + [f_x(y)x,[x,z]]\\
&= -f_x(z)[x,[x,y]] + f_x(y)[x,[x,z]]\\
&= -f_x(z)f_x(y)x + f_x(y)f_x(z)x\\
&= 0.
\end{aligned} \tag{3.14}
$$

Now we have the following chain of identities:

$$
[\exp(x,s)(y), \exp(x,s)(z)]
$$

$$
\begin{aligned}
&= [y + s\mathrm{ad}_x(y) + \frac{1}{2}s^2\mathrm{ad}_x^2(y), z + s\mathrm{ad}_x(z) + \frac{1}{2}s^2\mathrm{ad}_x^2(z)] \\
&= [y, z] + [y, s\mathrm{ad}_x(z)] + [y, \frac{1}{2}s^2\mathrm{ad}_x^2(z)] \\
&\quad +[s\mathrm{ad}_x(y), z] + [s\mathrm{ad}_x(y), s\mathrm{ad}_x(z)] + [s\mathrm{ad}_x(y), \frac{1}{2}s^2\mathrm{ad}_x^2(z)] \\
&\quad +[\frac{1}{2}s^2\mathrm{ad}_x^2(z), z] + [\frac{1}{2}s^2\mathrm{ad}_x^2(z), s\mathrm{ad}_x(z)] + [\frac{1}{2}s^2\mathrm{ad}_x^2(z), \frac{1}{2}s^2\mathrm{ad}_x^2(z)] \\
&= [y, z] + s[y, \mathrm{ad}_x(z)] + s[\mathrm{ad}_x(y), z] \\
&\quad +2 \cdot \frac{1}{2}s^2[\mathrm{ad}_x(y), \mathrm{ad}_x(z)] + \frac{1}{2}s^2[y, \mathrm{ad}_x^2(z)] + \frac{1}{2}s^2[\mathrm{ad}_x^2(y), z] + 0 + 0 \\
&= [y, z] + s\mathrm{ad}_x([y, z]) + \frac{1}{2}s^2\mathrm{ad}_x^2([y, z]) \\
&= \exp(x, s)([y, z]). \qquad (3.15)
\end{aligned}
$$

So indeed $\exp(x, s)$ is an automorphism of L.

3.2 Lie Algebras Generated by Extremal Elements

Lemma 3.5. *If L is generated as a Lie algebra by extremal elements, then it is spanned by the set $\mathcal{E}$ of all extremal elements.*

Proof We consider $z \in L$ as a bracketing of elements from $\mathcal{E}$. We apply induction on the length of z. If the length of z is 1 then $z \in \mathcal{E}$ and we are done.

Now suppose all monomials of length n are indeed linear combinations of elements from $\mathcal{E}$ and z is a bracketing of length $n+1$. Then consider the last two elements of z, more precisely: $z = [\cdot, [\cdot, [\cdots, [\cdot, [x, y]] \cdots]]]$ with $x, y \in \mathcal{E}$. Now let $m = \exp(x, 1)y = y + [x, y] + \frac{1}{2}f(x, y)x$, and note that m is an extremal element since $\exp(x, 1)$ is an automorphism. This means that

$$
\begin{aligned}
z &= [\cdot, [\cdot, [\cdots, [\cdot, m - y - \frac{1}{2}f(x, y)x] \cdots]]] \\
&= [\cdot, [\cdot, [\cdots, [\cdot, m] \cdots]]] - [\cdot, [\cdot, [\cdots, [\cdot, y] \cdots]]] \\
&\quad -\frac{1}{2}f(x, y)[\cdot, [\cdot, [\cdots, [\cdot, x] \cdots]]], \qquad (3.16)
\end{aligned}
$$

and we apply the induction hypothesis. This finishes the proof. □

We show how the map f defined in the beginning of this section (see Identity 3.1) gives rise to a symmetric bilinear associative form:

Theorem 3.6. *Suppose the Lie algebra L over the field $\mathbb{F}$ is generated by the set of nonzero extremal elements $\mathcal{E}$. Then a unique bilinear symmetric form $f : L \times L \to K$ exists, such that the linear form f_x coincides with $y \mapsto f(x, y)$ for each $x \in \mathcal{E}$. This form is associative in the sense that $f(x, [y, z]) = f([x, y], z)$ for all triples $x, y, z \in L$.*

Proof By the previous lemma, we know a basis of L exists consisting of elements from $\mathcal{E}$, say $u_1, \ldots, u_t$. Note that if $x \in \mathcal{E}$ and $\alpha \in \mathbb{F}$, then $[\alpha x, [\alpha x, y]] = \alpha^2 f_x(y)x = \alpha f(x, y)\alpha x$, so αx is also extremal with $f(\alpha x, y) = \alpha f(x, y)$.

Now suppose $x = \sum_i \lambda_i u_i$, and define f_x by $\sum_i \lambda_i f_{u_i}$, which is equal to $\sum_i f_{\lambda_i u_i}$. Furthermore, suppose that both $\sum_i u_i$ and $\sum_i v_i$ are ways of writing an $x \in L$ as a sum of extremal elements. Now we let $z \in \mathcal{E}$ and see, by Lemma 3.2,

$$
f_x(z) = \sum_i f_{u_i}(z) = \sum_i f_z(u_i) = f_z(\sum_i u_i) = f_z(\sum_i v_i) = \sum_i f_z(v_i) = \sum_i f_{v_i}(z), \qquad (3.17)
$$

so, since $\mathcal{E}$ spans L (Lemma 3.5), we know that $\sum_i f_{u_i} = \sum_i f_{v_i}$. Hence, f_x is a well defined linear functional. It immediately follows that $f(x,y) = f_x(y)$ defines a bilinear form, which is symmetric by Lemmas 3.2 and 3.5.

It remains to show that f is associative in the sense described above. Take $x, y, z \in \mathcal{E}$. Interchanging x and y in (P4) gives

$$2[[x,y],[y,[x,z]]] = -f(x,z)f(x,y)y + f(y,[x,z])[x,y] + f(x,y)[y,[x,z]], \tag{3.18}$$

and Jacobi followed by the application of (P1) gives

$$\begin{aligned} 2[[x,y],[y,[x,z]]] &= -2[y,[[x,z],[x,y]]] - 2[[x,z],[[x,y],y]] \\ &= [y, f(x,[y,z])x + f(x,z)[x,y] - f(x,y)[x,z]] \\ &\quad -2f(x,y)[[x,z],y] \\ &= -f(x,[y,z])[x,y] - f(x,z)f(x,y)y \\ &\quad +f(x,y)[y,[x,z]]. \end{aligned} \tag{3.19}$$

We now distinguish two cases.

- If $[x,y] \neq 0$ we compare the coefficients of $[x,y]$ in the two expressions above, and see that $-f(x,[y,z]) = f(y,[x,z])$. From this it follows that

$$f(x,[z,y]) = -f(x,[y,z]) = f(y,[x,z]) = f([x,z],y), \tag{3.20}$$

  proving the statement if $[x,y] \neq 0$. By symmetry, we find $f(x,[y,z]) = f([x,y],z)$ if $[x,z] \neq 0$ and $f(y,[x,z]) = f([y,x],z)$ if $[y,z] \neq 0$.

- If $[x,y] = 0$ and $[x,z] \neq 0$ and $[y,z] \neq 0$, we have $f(x,[z,y]) = 0 = f([x,z],y)$ by the previous calculations. By symmetry, we are left with the case $[x,y] = 0, [x,z] = 0$. By Jacobi, we have $[x,[y,z]] = 0$, and applying ad yields $f(x,[y,z]) = 0 = f([x,z],y)$. So we proved the statement.

This implies that $f(x,[z,y]) = f([x,z],y)$ for all $x,y,z \in \mathcal{E}$. Since $\mathcal{E}$ spans L, this completes the proof. □

One of the main results on Lie algebras generated by finitely many extremal elements is the following theorem.

Theorem 3.7. *If L is generated as a Lie algebra by a finite number of extremal elements, then L is finite dimensional.*

This theorem is due to Zelmanov and Kostrikin [ZK90]. The proof by Zelmanov and Kostrikin requires the introduction of many notions, so we refrain from giving that here. Another, somewhat shorter, version of the proof can be found in [CSUW01].

3.3 The Radical and the Bilinear Form f

This section is a summary of the observations leading to one of the results of [CSUW01, Section 9]. Throughout this section, L is a Lie algebra over the field $\mathbb{F}$ generated by extremal elements, and $\mathcal{E}$ is the set of extremal elements of L.

We define the *radical* of the bilinear form f (as introduced in Theorem 3.6) as follows:

Definition 3.8. (Radical of f)

$$\mathrm{Rad}(f) = \{x \in L \mid f(x,y) = 0 \text{ for all } y \in L\}. \tag{3.21}$$

The remainder of this section is devoted to the proof of the following theorem.

Theorem 3.9. *If the characteristic of the underlying field is not* 2 *or* 3*, then* $\mathrm{Rad}(f) = \mathrm{Rad}(L)$.

Before proceeding to the proof of this theorem, we prove a few lemmas.

Lemma 3.10. *Let* J *be an ideal of* L*, and let* $N := \mathrm{span}_{\mathbb{F}}\{x \in \mathcal{E} \mid x \notin J\}$*. Then* $f(N, J) = 0$.

Proof Let $x \in \mathcal{E}\backslash J$ and $y \in J$. Then $f(x, y)x = [x, [x, y]] \in J$, so $f(x, y) = 0$. □

Lemma 3.11. *Let* K *be a solvable ideal of* L*. Then* $\mathcal{E} \cap K \subseteq \mathrm{Rad}(f)$.

Proof Let $x \in \mathcal{E} \cap K$, and let $y \in \mathcal{E}$. If $y \notin K$, then $f(x, y) = 0$ by the previous lemma. If, on the other hand, $y \in K$ and $f(x, y) \neq 0$, then $\langle x, y\rangle \equiv \mathfrak{sl}_2$, contradicting that K is solvable. This implies that $f(x, y) = 0$ for all $x \in \mathcal{E} \cap K$ and $y \in \mathcal{E}$, completing the proof as $\mathcal{E}$ spans L (Lemma 3.5). □

Lemma 3.12. *We have* $\mathrm{Rad}(L) \subseteq \mathrm{Rad}(f)$.

Proof Let K be a solvable ideal of L and let $x \in K$ and $y \in \mathcal{E}$. If $y \in K$ we have $f(x, y) = 0$ by Lemma 3.11. If $y \notin K$, we have $f(x, y) = 0$ by Lemma 3.10. □

We denote by $\mathrm{SanRad}(L)$ the linear span of all *sandwiches* of $\mathcal{E}$:

$$\mathrm{SanRad}(L) := \{x \in \mathcal{E} \mid \mathrm{ad}_x^2 = 0\}. \tag{3.22}$$

It is easy to see that $\mathrm{SanRad}(L)$ is an ideal of L. Moreover, as the restriction of f to $\mathrm{SanRad}(L)$ is identically zero, we know that $\mathrm{SanRad}(L)$ is a nilpotent Lie subalgebra of L, so $\mathrm{SanRad}(L) \subseteq \mathrm{NilRad}(L)$ (cf [ZK90] and Lemma 4.2 of [CSUW01]).

These observations lead to the following chain of inclusions:

$$\mathrm{SanRad}(L) \subseteq \mathrm{NilRad}(L) \subseteq \mathrm{Rad}(L) \subseteq \mathrm{Rad}(f). \tag{3.23}$$

Lemma 3.13. *If* $x \in \mathcal{E}\backslash\mathrm{Rad}(f)$ *and* $y \in \mathrm{Rad}(f)$*, then* $\mathrm{ad}_{[x,y]}^4 = 0$*, provided that* $\mathrm{char}(\mathbb{F}) \neq 2$.

Proof Write $X = \mathrm{ad}_x$ and $Y = \mathrm{ad}_y$. Then we have for $z \in L$, by (P2),

$$\begin{aligned} 2XYXYz &= 2[x, [y, [x, [y, z]]]] \\ &= f(x, [y, [y, z]])x - f(x, y)[x, [y, z]] - f(x, [y, z])[x, y] \\ &= f(y, z)f(x, y)x - f(y, x)[x, [y, z]] + f(y, [x, z])[x, y] \\ &= 0, \end{aligned} \tag{3.24}$$

so $2XYXY = 0$, and $XYXY = 0$ by the assumption that the characteristic of the field is not 2. Moreover,

$$X^2Yz = [x, [x, [y, z]]] = f(x, [y, z])x = -f(y, [x, z])x = 0, \tag{3.25}$$

so $X^2Y = 0$. Using these equations, we have for $\mathrm{ad}_{[x,y]}^4$:

$$\begin{aligned} \mathrm{ad}_{[x,y]}^4 &= (XY - YX)^4 \\ &= (XYXY - XY^2X - YX^2Y + YXYX)^2 \\ &= (YXYX - XY^2X) \\ &= YXYXYXYX - YXYXXY^2X - XY^2XYXYX + XY^2XXY^2X \\ &= Y(XYXY)XYX - YXY(X^2Y)YX - XY^2(XYXY)X + XY^2(X^2Y)YX \\ &= 0. \end{aligned} \tag{3.26}$$

□

Lemma 3.14. *Let K be an ideal of L, such that $K \subseteq \mathrm{Rad}(f)$. Then $\overline{L} := L/K$ is spanned by extremal elements, with induced form $\overline{f}$ defined by $\overline{f}(\overline{x}, \overline{y}) := f(x, y)$ for $x, y \in L$.*

Proof Since $K \in \mathrm{Rad}(f)$, the expression $\overline{f}(\overline{x}, \overline{y})$ is well-defined for $x, y \in L$. Furthermore, for $x \in \mathcal{E}$ and $y \in L$, we have $[\overline{x}, [\overline{x}, \overline{y}]] = \overline{f}(\overline{x}, \overline{y}) \cdot \overline{x} = f(x, y) \cdot \overline{x}$, so indeed $\overline{x} \in \mathcal{E}(\overline{L})$. □

The proof of the theorem is due to Gabor Ivanyos.

Proof of Theorem 3.9 Recall that Lemma 3.12 states that $\mathrm{Rad}(L) \subseteq \mathrm{Rad}(f)$, so if $\mathrm{Rad}(f) \subseteq Z(L)$, there is nothing to prove. Suppose therefore that $\mathrm{Rad}(f) \not\subseteq Z(L)$. We will first show that $\mathrm{SanRad}(L) \neq 0$.

By Lemma 3.13, there exists a nonzero $w \in \mathrm{Rad}(f)$ with $\mathrm{ad}_w^4 = 0$. Indeed, if $\mathcal{E} \cap \mathrm{Rad}(f) \neq \emptyset$, any $w \in \mathcal{E} \cap \mathrm{Rad}(f)$ is good enough. Otherwise, if $\mathcal{E} \cap \mathrm{Rad}(f) = \emptyset$, take $y \in \mathrm{Rad}(f)$ and $x \in \mathcal{E}$ such that $[x, y] \neq 0$ (which is possible by the fact that L is spanned by $\mathcal{E}$ and $\mathrm{Rad}(f) \not\subseteq Z(L)$), and use Lemma 3.13 to see that $\mathrm{ad}_w^4 = 0$ if we take $w = [x, y]$. Now, if $\mathrm{char}(\mathbb{F})$ is not 2 or 3, we have $\mathrm{ad}_w^3(x) = 0$ for any $x \in L$, by Proposition 2.1.5 of [Kos90]. Summing up, we have that there exists a nonzero $w \in \mathrm{Rad}(f)$ with $\mathrm{ad}_w^3 = 0$.

If $\mathrm{ad}_w^2 = 0$, we are done. Otherwise, there exists a $b \in \mathcal{E}$ with $x = \mathrm{ad}_w^2(b) \neq 0$. If $b \in \mathrm{Rad}(f)$ then $b \in \mathrm{Rad}(f) \cap \mathcal{E}$, so $[b, [b, c]] = 0$ for all $c \in L$, so $b \in \mathrm{SanRad}(L)$ and we are done. So we assume that $b \notin \mathrm{Rad}(f)$. Then, for $t \in L$,

$$\begin{aligned}
[x, [x, t]] &= f(x, t)x \\
&= f([w, [w, b]], t)x \\
&= f([[b, w], w], t)x \\
&= f(b, [w, [w, t]])[w, [w, b]] \\
&= [w, [w, [b, [b, [w, [w, t]]]]]]
\end{aligned} \tag{3.27}$$

so $\mathrm{ad}_x^2 = \mathrm{ad}_w^2 \mathrm{ad}_b^2 \mathrm{ad}_w^2$ (cf Lemma 1.7(iii) of [Ben77]). Since $w \in \mathrm{Rad}(f)$, we have $\mathrm{ad}_w^2(L) \subseteq \mathrm{Rad}(f)$. Since $b \in \mathcal{E} \backslash \mathrm{Rad}(f)$, we have $ad_b^2(\mathrm{Rad}(f)) = 0$. This implies that $\mathrm{ad}_x^2 = 0$, so $x \in \mathrm{SanRad}(L)$. Summing up, we have that $\mathrm{SanRad}(L) \neq 0$.

The reasoning above shows that if $\mathrm{Rad}(f) \not\subseteq Z(L)$, then $\mathrm{SanRad}(L) \neq 0$, hence $\mathrm{NilRad}(L) \neq 0$. But then, using Lemma 3.14 and by induction on the dimension, $\mathrm{Rad}(f)$ must be solvable. This completes the proof of Theorem 3.9. □

Remark 3.15. One of the consequences of this theorem is the following. Suppose we have a basis B (of dimension n) for a Lie algebra L over the field $\mathbb{F}$ generated by extremal elements, and we calculated (for a given bilinear form f) the $n \times n$ matrix M defined by

$$M_{i,j} = f(B_i, B_j).$$

We may then view M as a linear map from $\mathbb{F}^n$ to $\mathbb{F}^n$, and calculate its null space $\mathcal{N}$. There is an obvious one-to-one correspondence between $\mathcal{N}$ and $\mathrm{Rad}(f)$. Indeed, suppose that $\alpha_1 e_1 + \ldots + \alpha_n e_n \in \mathcal{N}$. This implies (by definition) that

$$\alpha_1 f(B_1, B_l) + \ldots + \alpha_n f(B_n, B_l) = 0, \text{ for } l = 1, \ldots, n,$$

so $\alpha_1 f(B_1, x) + \ldots + \alpha_n f(B_n, x) = 0$ for all $x \in L$ (since B spans L), and by bilinearity we have

$$f(\alpha_1 B_1 + \ldots + \alpha_n B_n, x) = 0,$$

so $\alpha_1 B_1 + \ldots + \alpha_n B_n \in \mathrm{Rad}(f)$. The proof in the other direction is similar.

We will use this observation to analyse the structure of various degenerate cases, for example in Section 6.5.

3.4 Generating Semi-Simple Lie Algebras

In this section we assume that we are working over the field $\mathbb{F}$ of characteristic not 2 or 3.

Theorem 3.16. *Suppose the semi-simple Lie algebra L_1 is generated (as a Lie algebra) by n extremal elements and no fewer, and the semi-simple Lie algebra L_2 is generated (as a Lie algebra) by m extremal elements and no fewer. Then the semi-simple Lie algebra $L_1 + L_2$ is generated by $n + m$ extremal elements and no fewer.*

Before giving the proof of this theorem, we prove a small lemma.

Lemma 3.17. *Suppose $x \in L_1$ is an extremal element in L_1. Then x is an extremal element in $L_1 + L_2$. Moreover, the bilinear form f on L_1 coincides with f on $L_1 + L_2$.*

Proof Let $l \in L_1 + L_2$, and let $l_1 \in L_1$ and $l_2 \in L_2$ be such that $l = l_1 + l_2$. Then $[x, [x, l_1]] = f(x, l_1)x$ and $f(x, l_2) = 0$ since $[x, [x, l_2]] = 0$. We have

$$[x, [x, l]] = f(x, l_1)x + 0x = (f(x, l_1) + f(x, l_2))x = f(x, l_1 + l_2)x = f(x, l)x,$$

so x is indeed an extremal element of $L_1 + L_2$. It also follows that f_x on $L_1 + L_2$ coincides with f_x on L_1. Since L_1 is spanned by extremal elements, the bilinear form f on $L_1 + L_2$ coincides with f on L_1 as well. □

From this argument it is clear that extremal elements of L_2 are extremal elements of $L_1 + L_2$ as well.

Proof of Theorem 3.16 Suppose that L_1 is generated by the extremal elements $x_1, \ldots, x_n$ and L_2 is generated by the extremal elements $y_1, \ldots, y_m$. We let $L = L_1 + L_2$. It is clear that $L_1 + L_2$ is generated by $x_1, \ldots, x_n, y_1, \ldots, y_m$, and that these elements are all extremal. Indeed, for example, $[x_i, [x_i, l]] = [x_i, [x_i, \alpha x_s + \beta y_t]] = \alpha f(x_i, x_s)x_i$, for some $\alpha, \beta \in \mathbb{F}$.

So it remains to prove that L cannot be generated by less than $n + m$ extremal elements. Suppose to the contrary that $L_1 + L_2$ is generated by less than $n+m$ extremal elements. Recall that there exists a set B of extremal elements spanning $L_1 + L_2$. If it were the case that for all $b \in B$ either $b \in L_1$ or $b \in L_2$, then $B = B_1 \cup B_2$, such that $B_1 \subseteq L_1$, $B_2 \subseteq L_2$, $B_1 \cap B_2 = \emptyset$, and $[B_1, B_2] = 0$. Moreover, B_1 spans L_1 and B_2 spans L_2. Now we need n extremal elements to generate L_1, and since these extremal elements are all in B_1 and $[B_1, L_2] = 0$, we need m additional extremal elements to generate L_2. This contradicts the assumption.

So from now on we assume that there exists a $z \in B$, such that $z = x + y$, $x \in L_1$ and $y \in L_2$, and $x, y \neq 0$. Suppose L_1 is spanned by extremal elements $x_1, \ldots, x_N$, and L_2 is spanned by extremal elements $y_1, \ldots, y_M$. Then $z = \alpha_1 x_1 + \ldots + \alpha_N x_N + \beta_1 y_1 + \ldots + \beta_M y_M$. We let $t \in L_1$. Using the the fact that z is extremal and the lemma above, we find

$$\begin{aligned} f(z,t)z &= [z,[z,t]] \\ &= \sum_{i=1}^{N} \alpha_i [z,[x_i,t]] + \sum_{i=1}^{M} \beta_i [z,[y_i,t]] \\ &= \sum_{i=1}^{N} \alpha_i [z,[x_i,t]] + 0 \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j [x_j,[x_i,t]] + \sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_i \beta_j [y_j,[x_i,t]] \end{aligned}$$

$$\begin{aligned} &= [x,[x,t]] + \sum_{i=1}^{N}\sum_{j=1}^{M} \alpha_i\beta_j(-[x_i,[t,y_j]] - [t,[y_j,x_i]]) \\ &= f(x,t)x + 0. \end{aligned} \tag{3.28}$$

The fact that $[y, t] = 0$ leads to the observation that $f(y, t) = 0$ (since $f(y, t)y = [y, [y, t]] = 0$), so

$$f(z,t)(x+y) = f(z,t)z = f(x,t)x + f(y,t)x = f(z,t)x,$$

and since $y \neq 0$ we have $f(z, t) = 0$. Similarly, we find $f(z, L_2) = 0$. This implies that $f(z, L) = 0$, so $z \in \mathrm{Rad}(f)$, and by Theorem 3.9 then $z \in \mathrm{Rad}(L)$, so $\dim(\mathrm{Rad}(L)) \geq 1$. This shows that L is not semi-simple, so it cannot be isomorphic to $L_1 + L_2$. This contradicts the assumption that L can be generated by less than $n + m$ extremal elements, so we proved the theorem. □

Repeatedly applying this theorem shows that the statement is true for the direct sum of finitely many simple Lie algebras.

Chapter 4

Lie Algebras Generated by Two Extremal Elements

In this chapter we study Lie algebras generated by two extremal elements, x and y. In the first section it is shown that such a Lie algebra is in general isomorphic to A_1. In the second section we take a different point of view, and come to the same conclusions. The first section is taken from [CSUW01].

4.1 Structure

Theorem 4.1. *Let L be a Lie algebra over the field $\mathbb{F}$ generated by two extremal elements $x, y \in \mathcal{E}$. Then exactly one of the following three assertions holds:*

1. $L = \mathbb{F}x + \mathbb{F}y$ *is Abelian,* $\mathcal{E} = L \backslash \{0\}$.
2. $L = \mathbb{F}x + \mathbb{F}y + \mathbb{F}z$*, where* $z = [x, y] \neq 0$*, and* $\mathcal{E} = L \backslash \{0\}$.
3. $L \cong \mathfrak{sl}_2$ *and* $\mathcal{E}$ *consists of all nilpotent elements of* L.

Proof We define $z = [x, y]$ and distinguish three cases:

- If $[x, y] = 0$:
  Lemma 3.5 implies that L is spanned by x and y, i.e. $L = \mathbb{F}x + \mathbb{F}y$. Since $[x, y] = 0 = [y, x]$, we have that x and y commute, hence L is Abelian. From this it immediately follows that $[v, [v, u]] = 0 = 0 \cdot v$ for each $v \in L \backslash \{0\}$, so indeed $\mathcal{E} = L \backslash \{0\}$. Furthermore, f is identically 0.

- If $[x, y] \neq 0$ and f is identically zero:
  We note that z is an extremal element by (P4). Hence, by Lemma 3.5 we have $L = \mathbb{F}x + \mathbb{F}y + \mathbb{F}z$. Similar reasoning as above leads to $\mathcal{E} = L \backslash \{0\}$. This Lie algebra is referred to as the *Heisenberg*-algebra, denoted by $\mathfrak{h}$.

- If $[x, y] \neq 0$ and f is not identically zero:
  Then L is isomorphic with $\mathfrak{sl}_2$. We identify

$$x \text{ with } \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, y \text{ with } \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \tag{4.1}$$

  and see that $[x, y] := xy - yx$ is equal to:

$$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{4.2}$$

Furthermore, we prove that $\mathcal{E}$ consists of all nilpotent elements of $\mathfrak{sl}_2$.

- On the one hand, let $u \in \mathfrak{sl}(2)$, $u \neq 0$, say $u = \begin{pmatrix} a & b \\ c & -a \end{pmatrix}$. If u is nilpotent, we have $\det(u) = 0$, hence $a^2 + bc = 0$.
  Now we distinguish two cases. Firstly, if $a = 0$, then $b = 0$ or $c = 0$, yielding either $u \in \mathbb{F}y$ or $u \in \mathbb{F}x$, so $u \in \mathcal{E}$. Secondly, if $a \neq 0$, then $u = a[x, y] + \delta x + \epsilon y$ with ϵ such that $\delta\epsilon = a^2$. In that case, too, $u \in \mathcal{E}$, because u is a linear combination of extremal elements.
- On the other hand, suppose u is an extremal element, write $u = \begin{pmatrix} a & b \\ c & -a \end{pmatrix}$.
  Then $[u, [u, [x, y]]] = \begin{pmatrix} -4bc & 4ab \\ 4ac & 4bc \end{pmatrix} = \alpha u$ for a certain $\alpha \in \mathbb{F}$. This yields $-4bc = a\alpha$, $4ab = \alpha b$, and $4ac = \alpha c$, so $\alpha = 4a$ and $a^2 + bc = 0$. (Note that we use the fact that the characteristic of the field is zero here). By the above $a^2 + bc = 0$ implies that u is nilpotent.

□

4.2 Classification by Structure Constants

Once we fix the basis of L to be x, y, and $[x, y]$, the structure of the resulting Lie algebra is completely determined by the functions $f(x, y)$ and $f(y, x)$. As noted before, we have $f(x, y) = f(y, x)$, so for every value we pick for $f(x, y)$ we might have a Lie algebra. This means we can consider the 'line' $\mathbb{F}$ as a line of Lie algebras, and wonder what Lie algebra corresponds to each point on the line. To help us study the structure, we write down the multiplication table of the basis elements of L in Table B.2.

At the origin (i.e. $f(x, y) = 0$), we have the case where $L = \mathbb{F}x + \mathbb{F}y + \mathbb{F}z$ (see Table B.3). On every other point we have $\mathfrak{sl}_2$, as noted before. Note that we explicitly demanded that the dimension of the resulting Lie algebra is 3, so here we do not see the case where $[x, y] = 0$.

Chapter 5

Lie Algebras Generated by Three Extremal Elements

In this chapter we study Lie algebras over the field $\mathbb{F}$ generated by three extremal elements x, y, and z. In the first section we show that the dimension of such a Lie algebra is 8, and in the second section we show that it is in general isomorphic to A_2. In the third section we study the Lie algebra using structure constants. The first two sections are largely inspired by [CSUW01, Section 5].

5.1 Dimension

In this section we study the Lie Algebra L generated by three extremal elements, x, y, and z. The five identities in Lemma 3.3 immediately show that

$$x, y, z, [x, y], [y, z], [x, z], [x, [y, z]], [y, [x, z]] \tag{5.1}$$

span L. This implies that L is at most 8-dimensional. It is easily checked that the above 8 elements give a basis in the free case, so $\dim L = 8$.

5.2 Structure

We study the actions of ad_x and ad_y on the linear generators of L, and see that they can be completely described by identities in terms of the four parameters $f(x, y)$, $f(x, z)$, $f(y, z)$, and $f(x, [y, z])$. We describe these parameters by drawing a triangle with vertices x, y, and z, and labeling the edge xy with 'edge' parameter $f(x, y)$, etc, and putting the 'central' parameter $f(x, [y, z])$ in the middle, with an indication of orientation. See Figure 5.1.

We will reduce $f(x, [y, z])$ to zero by transforming the generators, as follows. Let $s \in \mathbb{F}$, and consider the triple $x, y, \exp(x, s)z$. It has the following parameters:

$$\begin{aligned}
f(x, y) &= f(x, y) \\
f(x, \exp(x, s)z) &= f(x, z) \\
f(y, \exp(x, s)z) &= f(y, z) + sf(x, [y, z]) + \frac{1}{2}s^2 f(x, y), f(x, z) \\
f(x, [y, \exp(x, s)z]) &= f(x, [y, z]) - sf(x, y)f(x, z).
\end{aligned} \tag{5.2}$$

Proof For the second parameter, we have

$$f(x, \exp(x, s)z) = f(x, z) + sf(x, [x, z]) + \frac{1}{2}s^2 f(x, [x, [x, z]]) = f(x, z), \tag{5.3}$$

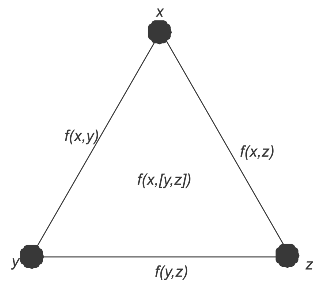


Figure 5.1: The three generator case

and for the third parameter, we have:

$$\begin{aligned} f(y, \exp(x,s)z) &= f(y,z) + sf(y,[x,z]) + \frac{1}{2}s^2 f(y,[x,[x,z]]) \\ &= f(y,z) + sf(x,[y,z]) + \frac{1}{2}s^2 f(y, f(x,z)x) \end{aligned} \tag{5.4}$$

For the last parameter, we have

$$f(x,[y,\exp(x,s)z]) = f(x,[y,z]) + sf(x,[y,[x,z]]) + \frac{1}{2}s^2 f(x,[y,[x,[x,z]]]) \tag{5.5}$$

The last term is obviously zero, and for the second term, we have

$$f(x,[y,[x,z]]) = -f(y,[x,[x,z]]) = -f(x,z)f(y,x) = -f(x,y)f(x,z).$$

So indeed $f(x,[y,\exp(x,s)z]) = f(x,[y,z]) - sf(x,y)f(x,z)$. □

Clearly, this triple again consists of extremal elements, and generates the same algebra as x, y, and z. We distinguish two cases:

- If at least two of the three edges have nonzero labels (say $f(x,z)$ and $f(x,y)$), we can transform the central parameter $f(x,[y,z])$ to zero by a suitable choice of s, in this case $s = \frac{f(x,[y,z])}{f(x,y)f(x,z)}$.
- If at most one edge has a nonzero label, say $f(x,y)$, and the central parameter $f(x,[y,z])$ is also nonzero, we can move to three extremal generators x, y, and $\exp(x,s)z$ with one more edge (i.e. $f(y,\exp(x,s)z)$). So, we can reduce to the first case, and thus we are always able to reduce to the case where the central parameter is zero.

Now we let $\alpha, \beta, \gamma \in \mathbb{F}\backslash\{0\}$, and scale x, y, z to $\alpha x, \beta y, \gamma z$. This leaves the central parameter 0, and changes the edge labels to $\alpha\beta f(x,y)$, $\alpha\gamma f(x,z)$, and $\beta\gamma f(y,z)$. This means that, maybe using a field extension of $\mathbb{F}$, we may transform all the edge labels to -2.

By the above reasoning we are left with only four essentially different cases, distinguished by the number of edges labeled nonzero in the triangle:

Theorem 5.1. *Suppose the Lie algebra L is generated by three extremal elements. After extending the field if necessary, L is generated by three extremal elements whose central parameter is zero, and whose nonzero edge parameters are -2. In particular, L is a quotient of a Lie algebra M generated by extremal elements x, y, z with $f(x,[y,z]) = 0$, and $\dim M = 8$.*

We distinguish four cases, depending on the number of nonzero edge parameters.

The four cases are as follows.

- $f(x, y) = f(x, z) = f(y, z) = 0$:
  If $f = 0$, then $\mathrm{Rad}(M) = M$ by Theorem 3.9, so M is solvable.
- $f(x, y) = -2, f(x, z) = f(y, z) = 0$:
  In this case, we have a five dimensional solvable radical R, spanned by $\{z, [x, z], [y, z], [x, [y, z]], [y, [x, z]]\}$, and the semi-simple part M/R of M is three dimensional and isomorphic to $\mathfrak{sl}_2$.
- $f(x, y) = f(x, z) = -2, f(y, z) = 0$:
  In this case, we again have a five dimensional solvable radical R, spannend by $\{y - \frac{1}{2}[y, [x, z]], z - \frac{1}{2}[y, [x, z]], [x, y] - [x, z], [y, z], [x, [y, z]]\}$, and the semi-simple part of M is again isomorphic to $\mathfrak{sl}_2$.
- $f(x, y) = f(x, z) = f(y, z) = -2$:
  In this case, $M \cong \mathfrak{sl}_3$. Take for example

$$x = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, y = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \text{ and } z = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ -2 & -2 & -2 \end{pmatrix}. \quad (5.6)$$

5.3 Classification by Structure Constants

In this section we follow Section 4.2, and analyse the structure of the three generator case using the evaluation of the function f defined in Theorem 3.6. First, we see that by $f(x, [y, z]) = f([x, y], z)$, $f(x, [y, z]) = -f(x, [z, y])$, and the symmetry of f, we only 'need' $f(x, y)$, $f(x, z)$, $f(y, z)$, and $f(x, [y, z])$. In Table B.4 the upper triangular part can be found (for ease of reading), the lower triangular part follows by the fact that the multiplication table of any Lie algebra is anti-symmetric.

Whereas the multiplication table of the two generator case only had 9 entries, this one already contains 64 entries, and thus is a bit to large to study by hand. Fortunately, we can use GAP [S+95] to help us. To this end, we enter the structure constants of the multiplication into GAP. For example,

$$[[x, y], [x, z]] = \frac{1}{2} f(x, [y, z])x + \frac{1}{2} f(x, z)[x, y] - \frac{1}{2} f(x, y)[x, z]$$

gives the multiplication of the fourth and the fifth basis element, expressed in the other basis elements, which yields the GAP command

```
SetEntrySCTable(T, 4, 5, [(1/2)*fxyz, 1, (1/2)*fxz, 4, -(1/2)*fxy, 5]);
```

Note that we have to define `fxy`,`fxz`,`fyz`, and `fxyz` in advance.

We do so for all possible multiplications, see Section C.1. This gives us the possibility to easily analyse the Lie algebra. We write f_{xy} for $f(x, y)$, f_{xz} for $f(x, z)$, f_{yz} for $f(y, z)$, and f_{xyz} for $f(x, [y, z])$. Some observations:

- Setting f_{xy}, f_{xz}, f_{yz}, and f_{xyz} nonzero (and distinct) indeed gives us A_2,
- Setting f_{xy}, f_{xz}, f_{yz} nonzero, and f_{xyz} to zero gives A_2 as well,
- Setting $f_{xy} = f_{xz} = f_{yz}$ (nonzero) and f_{xyz} to zero gives A_2 as well,
- Setting $f_{xy} = f_{xz} = f_{yz} = 0$ and f_{xyz} nonzero gives A_2 as well,
- Setting zero, one or two of f_{xy}, f_{xz}, nonzero and the others to zero gives a non semi-simple Lie algebra, as indicated in Section 5.2. Its semi-simple part is isomorphic to A_1 (i.e. $\mathfrak{sl}_2$).

This seems to be in line with the results of Section 5.2.

Chapter 6

Lie Algebras Generated by Four Extremal Elements

In this chapter we consider Lie algebras generated by four extremal elements. The first section, inspired by [CSUW01, Section 6], shows why such a Lie algebra is 28 dimensional. In Section 6.3 it is shown that only 12 degrees of freedom exist determining exactly which Lie algebra four given extremal elements generate. In the third section, we use this result to analyse $2^{12} = 4096$ different Lie algebras generated by four extremal elements. In Section 6.4 we present an algorithm that enables us to analyse various instances of Lie algebras generated by (arbitrary many) extremal elements. This algorithm is somewhat faster than what we used before, and it is needed for our analyses in Chapter 8. Lastly, in Section 6.5, we consider some degenerate cases.

6.1 Dimension

In this section we study Section 6 of [CSUW01], which shows why a Lie algebra generated by four extremal elements has dimension 28.

Theorem 6.1. *Let L be generated by four extremal elements x, y, z, and u. Then L is spanned by the following 28 monomials of length at most 5:*

$$\begin{gathered} x, y, z, u, \\ [x,y], [x,z], [x,u], [y,z], [y,u], [z,u], \\ [x,[y,z]], [x,[y,u]], [x,[z,u]], [y,[x,z]], [y,[x,u]], [y,[z,u]], [z,[x,u]], [z,[y,u]], \\ [x,[y,[z,u]]], [x,[z,[y,u]]], [y,[x,[z,u]]], [y,[z,[x,u]]], [z,[x,[y,u]]], [z,[y,[x,u]]], \\ [x,[y,[z,[x,u]]]], [y,[x,[z,[y,u]]]], [z,[x,[y,[z,u]]]], [u,[x,[y,[z,u]]]]. \end{gathered} \tag{6.1}$$

Proof Since L is spanned by the monomials in x, y, z, and u, we only have to show that each monomial can be written as a linear combination of the 28 elements above. We distinguish on the length of the monomials.

All the monomials of length 1 are on the list. From the $4 \cdot 3 = 12$ monomials of length 2 we only need half, by skew-symmetry. These 6 monomials are on the list.

All monomials of length 3 involving at most 2 letters are reducible, since x, y, z, and u are all extremal, so we study only those monomials of length 3 with 3 different letters. We have $\binom{3}{2}$ such monomials starting with x, and $\binom{3}{2}$ such monomials starting with y. By Jacobi, we only need those monomials starting with z that contain u and either x or y (but not both), i.e. 2 more monomials. All monomials containing u have now been considered, by the Jacobi identity. This yields the 8 monomials of length 3 above.

All monomials of length 4 involving at most 3 letters are reducible, by the extremality of x, y, z, and u, and (P2). So we study only those monomials of length 4 with 4 different letters. By the Jacobi identity we find that all monomials starting with x, y, or z can be written as a linear combination of the six monomials of length 4 on the list:

$$[x,[y,[z,u]]] - [x,[z,[y,u]]] + [x,[u,[y,z]]] = 0, \tag{6.2}$$

$$[y,[x,[z,u]]] - [y,[z,[x,u]]] + [y,[u,[x,z]]] = 0, \tag{6.3}$$

$$[z,[x,[y,u]]] - [z,[y,[x,u]]] + [z,[u,[x,y]]] = 0, \tag{6.4}$$

For the monomials starting with u, we find:

$$[u,[x,[y,z]]] - [x,[u,[y,z]]] + [[y,z],[u,x]] = 0, \tag{6.5}$$

$$[u,[y,[x,z]]] - [y,[u,[x,z]]] + [[x,z],[u,y]] = 0, \tag{6.6}$$

$$[u,[z,[x,y]]] - [z,[u,[x,y]]] + [[x,y],[u,z]] = 0, \tag{6.7}$$

where the elements of the form $[[a,b],[c,d]]$ can be expressed as follows:

$$[y,[z,[x,u]]] - [z,[y,[x,u]]] + [[x,u],[y,z]] = 0, \tag{6.8}$$

$$[x,[z,[y,u]]] - [z,[x,[y,u]]] + [[y,u],[x,z]] = 0, \tag{6.9}$$

$$[x,[y,[z,u]]] - [y,[x,[z,u]]] + [[z,u],[x,y]] = 0. \tag{6.10}$$

So indeed all monomials of length 4 may be written as a linear combination of the elements above.

For the monomials of length 5 we multiply a monomial $[a,[b,[c,d]]]$ of length 4 to the left with a letter. This yields four possibilities:

- $[a,[a,[b,[c,d]]]]$, which is obviously reducible,
- $[b,[a,[b,[c,d]]]]$, which is reducible by Identity (P2),
- $[c,[a,[b,[c,d]]]]$, which cannot be obviously reduced, and
- $[d,[a,[b,[c,d]]]]$, which can be rewritten to the case above by interchanging c and d.

So we are left with the monomials of the form $[c,[a,[b,[c,d]]]]$, where $[a,[b,[c,d]]]$ is one of the above monomials of length 4. This yields the following 12 monomials, which we will call $m_{11},\ldots,m_{16},m_{21},\ldots,m_{26}$:

$$\begin{aligned}&[\mathbf{z},[\mathbf{x},[\mathbf{y},[\mathbf{z},\mathbf{u}]]]],[\mathbf{y},[\mathbf{x},[\mathbf{z},[\mathbf{y},\mathbf{u}]]]],[z,[y,[x,[z,u]]]],[\mathbf{x},[\mathbf{y},[\mathbf{z},[\mathbf{x},\mathbf{u}]]]],\\&\qquad\qquad\qquad\qquad\qquad\qquad[y,[z,[x,[y,u]]]],[x,[z,[y,[x,u]]]],\\&[\mathbf{u},[\mathbf{x},[\mathbf{y},[\mathbf{z},\mathbf{u}]]]],[u,[x,[z,[y,u]]]],[u,[y,[x,[z,u]]]],[u,[y,[z,[x,u]]]],\\&\qquad\qquad\qquad\qquad\qquad\qquad[u,[z,[x,[y,u]]]],[u,[z,[y,[x,u]]]].\end{aligned} \tag{6.11}$$

We will prove that only m_{11}, m_{12}, m_{14}, and m_{21} (boldfaced in Equation 6.11) are needed. Indeed, the other 8 elements can be expressed in these four:

- m_{13} and m_{23} using 6.10 and 6.7,
- m_{15} and m_{25} using 6.9 and 6.6,
- m_{16} and m_{26} using 6.8 and 6.5,
- m_{22} using 6.2, and
- m_{24} using 6.3.

It remains to prove that all monomials of length 6 are reducible. For this, we consider all the monomials above of length 5, and multiply them with a letter from the left. This yields four different cases:

- $[a,[c,[a,[b,[c,d]]]]]$: Obviously reducible by (P2).
- $[b,[c,[a,[b,[c,d]]]]]$: The term $[a,[b,[c,d]]]$ reduces, by the Jacobi identity, to $[b,[[c,d],a]]$ and $[[c,d],[a,b]]$. The latter passes to $[d,[c,[a,b]]]$ and $[c,[[a,b],d]]$, by the Jacobi identity. The resulting elements are all reducible by (P2).
- $[c,[c,[a,[b,[c,d]]]]]$: Obviously reducible.
- $[d,[c,[a,[b,[c,d]]]]]$: The term $[b,[c,d]]$ reduces, by the Jacobi identity, to $[c,[d,b]]$ and $[d,[c,b]]$. The former can be reduced by (P2), the latter has the form $[d,[c,[a,[d,[c,b]]]]]$, which reduces as follows:

$$\begin{aligned}[d,[c,[a,[d,[c,b]]]]] &= -[d,[c,[a,[c,[d,b]]]]]-[d,[c,[a,[d,[b,c]]]]]\\ &= [d,[c,[a,[c,[b,d]]]]]-[d,[c,[d,[a,[b,c]]]]]+[d,[c,[[b,c],[a,d]]]]\\ &= [d,[c,[a,[c,[b,d]]]]]-[d,[c,[d,[a,[b,c]]]]]-[d,[c,[[a,d],[b,c]]]]\\ &= [d,[c,[a,[c,[b,d]]]]]-[d,[c,[d,[a,[b,c]]]]]+[d,[c,[b,[c,[a,d]]]]]\\ &\quad -[d,[c,[c,[b,[a,d]]]]]. \qquad (6.12)\end{aligned}$$

  The last term reduces to a smaller case by extremality of c, and to the first three terms we apply (P2).

So indeed all monomials of length 6 are reducible to smaller cases.

This completes the proof of Theorem 6.1. □

6.2 Classification by Structure Constants

We consider a Lie algebra L generated by four extremal elements x, y, z, and u. As proved in the previous section, L is spanned by the 28 monomials in Equation 6.1. As in Section 5.3, we will analyse the structure of L using GAP. To this end, we need to know which evaluations of f control the structure of the resulting Lie algebra.

Theorem 6.2. *Let $a,b\in L$. Then $f(a,b)$ can be linearly expressed in terms of the following 12 monomials:*

$$\begin{gathered} f(x,y), f(x,z), f(x,u), f(y,z), f(y,u), f(z,u)\\ f(x,[y,z]), f(x,[y,u]), f(x,[z,u]), f(y,[z,u]),\\ f(x,[y,[z,u]]), f(x,[z,[y,u]]). \end{gathered} \qquad (6.13)$$

We call these the *primitive evaluations of f* or *primitives*. Before we give the proof, we give a few straightforward lemmas.

Lemma 6.3. *Suppose x is an extremal element, and $b,c\in L$, then*

$$f(x,[b,[x,c]]) = -f(b,[x,[x,c]]) = -f(b,x)f(x,c) = -f(x,b)f(x,c). \qquad (6.14)$$

Lemma 6.4. *Suppose $x,a\in L$, not necessarily extremal, then*

$$f(x,[x,a]) = f([x,x],a) = 0. \qquad (6.15)$$

Proof of Theorem 6.2 We let $a,b\in L$, and define M to be the set of monomials in Equation 6.1. Because L is spanned by the 28 monomials in M and f is a bilinear form, we only need to consider $f(a,b)$ with $a,b\in M$.

Moreover, we claim that we only need to consider $f(a, b)$ with $a \in \{x, y, z, u\}$ and $b \in M$. Indeed, let $a, b \in M$. If $a \in \{x, y, z, u\}$ we proved the claim, so suppose $a = [\tilde{a}, \tilde{A}]$, with $\tilde{a} \in \{x, y, z, u\}$. By associativity of f (see Theorem 3.6) we have

$$f(a, b) = f([\tilde{a}, \tilde{A}], b) = f(\tilde{a}, [\tilde{A}, b]),$$

and since $[\tilde{A}, b] \in L$, it is a linear combination of elements from M, so $f(a, b) = f(\tilde{a}, [\tilde{A}, b])$ is a linear combination of elements of the form $f(\tilde{a}, \tilde{b})$ with $\tilde{a} \in \{x, y, z, u\}$ and $\tilde{b} \in M$. This proves the claim.

From now on we consider $f(a, A)$ for an extremal element a and a monomial A of length at most 5. Furthermore, a, b, c, d, e, and g denote elements of the set $\{x, y, z, u\}$. We do a case analysis on the length of A.

1. We have something of the form $f(a, b)$. If $b = a$ we have

   $$f(a, b)a = f(a, a)a = [a, [a, a]] = 0 = 0a,$$

   so $f(a, b) = 0$. Otherwise, $f(a, b)$ is among the expressions in Equation 6.13.

2. We have $f(a, [b, c])$. If $b = a$ or $c = a$ we apply Lemma 6.4 to reduce the expression to a smaller case, otherwise $\pm f(a, [b, c])$ is among the expressions in Equation 6.13.

3. We have $f(a, [b, [c, d]])$. If b, c, d are not all distinct we have an obvious reduction to a smaller case: if $c = d$ it reduces to zero, if $b = c$ (or similarly, if $b = d$) we have

   $$f(a, [b, [c, d]]) = f(a, [b, [c, b]]) = f(b, c)f(a, b).$$

   So we assume b, c, d are all distinct.

   - If a is equal to b, we apply Lemma 6.4 to reduce the expression to a smaller case.
   - If a is equal to c (or similarly, equal to d), we apply Lemma 6.3.
   - So we assume a is distinct from b, c, and d. We verify that $f(x, [y, [z, u]])$ and $f(x, [z, [y, u]])$ are among the expressions in Equation 6.13, and

     $$f(x, [u, [y, z]]) = f(x, [z, [y, u]]) - f(x, [y, [z, u]]).$$

     Since $f(a, [b, [c, d]]) = -f(a, [b, [d, c]])$ this covers all $3! = 6$ cases where $a = x$. We can reduce the other $4! - 6 = 18$ cases to one of these by moving x to the front, which is possibly by associativity and symmetry of f, for example:

     $$f(y, [z, [x, u]]) = -f(y, [z, [u, x]]) = -f([[y, z], u], x) = f(x, [u, [y, z]]).$$

     So we covered all cases where a is distinct from b, c, and d.

   This completes the proof for the case where the second argument is a monomial of length 3.

4. We have $f(a, [b, [c, [d, e]]])$. If b, c, d, and e are not all distinct, we have an obvious reduction, either by extremality or by (P2). So we assume b, c, d, and e are distinct, so a must be equal to one of them.

   - If $a = b$, then we apply Lemma 6.4.
   - If $a = c$, then we apply Lemma 6.3.

- If $a = d$ (or similarly $a = e$) then $f(a, [b, [c, [d, e]]]) = f(a, [b, [c, [a, e]]]) = -f([b, a], [c, [a, e]]) = -f(b, [a, [c, [a, e]]])$, which reduces to a smaller case by (P2).

This completes the case where the length of the second argument of f is four.

5. We consider $f(a, [b, [c, [d, [e, g]]]])$. For sake of simplicity we only observe those $[b, [c, [d, [e, g]]]]$ that are elements of M (other instances easily reduce to those), so we may take $b = e$ and $b, c, d,$ and g distinct. So we consider $f(a, [b, [c, [d, [b, g]]]])$. Since $b, c, d,$ and g are all distinct, a must be equal to one of them.

   - If $a = b$, then we apply Lemma 6.4.
   - If $a = c$, then we apply Lemma 6.3.
   - If $a = d$, then

   $$f(a, [b, [c, [d, [e, g]]]]) = -f([b, a], [c, [a, [e, g]]]) = -f(b, [a, [c, [a, [e, g]]]]),$$

   which yields a reduction to a smaller case by (P2).

   - If $a = g$, then

   $$\begin{aligned} f(a, [b, [c, [d, [e, g]]]]) &= f(a, [b, [c, [d, [b, a]]]]) \\ &= f([[a, b], c], [d, [b, a]]) \\ &= -f([c, [a, b]], [d, [b, a]]) \\ &= f([a, [b, c]], [d, [b, a]]) \\ &\quad +f([b, [c, a]], [d, [b, a]]) \\ &= f([b, c], [a, [d, [a, b]]]) \\ &\quad +f([a, c], [b, [d, [b, a]]]), \end{aligned} \tag{6.16}$$

   and both terms reduce to a smaller case by (P2).

This completes the proof of Theorem 6.2. □

6.3 Using GAP to Find the Structure

In the previous section we proved that the structure of a Lie algebra generated by four extremal elements is completely determined by 12 evaluations of f. We use this knowledge to let GAP [S$^+$95] create the multiplication table of this Lie algebra automatically.

We let R be the ring $\mathbb{Q}[f(x, y), f(x, z), f(x, u), f(y, z), f(y, u), f(z, u), f(x, [y, z]), f(x, [y, u]), f(x, [z, u]), f(y, [z, u]), f(x, [y, [z, u]]), f(x, [z, [y, u]])]$. We consider the universal enveloping algebra M in R of a Lie algebra generated by four variables (x, y, z, and u), as defined in Section 2.6. We define the ideal I of M to be the ideal generated by the following 36 polynomials:

- 12 polynomials of the form $[a, [a, b]] - f(a, b)a$, for $a, b \in \{x, y, z, u\}$, and a and b distinct, and we translate $f(a, b)$ to $\pm$ one of the expressions in Equation 6.13. This indeed gives us $4 \cdot 3 = 12$ relations.

- 12 polynomials of the form $[a, [a, [b, c]]] - f(a, [b, c])a$, for $a, b, c \in \{x, y, z, u\}$ and a, b, and c distinct, and we translate $f(a, [b, c])$ to $\pm$ one of the expressions in Equation 6.13. Obviously, if we include $[a, [a, [b, c]]]$, we may omit $[a, [a, [c, b]]]$, so this indeed gives us $4 \cdot 3 \cdot 2 \cdot \frac{1}{2} = 12$ relations.

- 12 polynomials of the form $[a, [a, [b, [c, d]]]] - f(a, [b, [c, d]])a$, for $\{a, b, c, d\} = \{x, y, z, u\}$, and we translate $f(a, [b, [c, d]])$ to $\pm f(x, [y, [z, u]])$, $\pm f(x, [z, [y, u]])$, or a sum of those two. This gives $4!/2 = 12$ relations.

This means that I 'describes' the fact that x, y, z, and u are extremal elements. Using GBNP [CG05], we can calculate a Gröbner basis for I, and thus we are able to perform calculations in the quotient algebra M/I. In M/I we straightforwardly calculate a basis for the algebra generated by x, y, z, and u.

The following corollary immediately follows from Theorem 6.2.

Corollary 6.5. Using the above procedure every element of R corresponds to exactly one Lie algebra generated by four extremal elements.

We find a 28-dimensional algebra L in R, and verifying the Jacobi identity shows that L indeed is a Lie algebra. However, the structure is not determined until we fix f. As a first test we let all the values in Equation 6.13 be either 0 or 1, and for all these $2^{12} = 4096$ Lie algebras we store the following properties, where L is the Lie algebra and S is $\mathrm{Rad}(L)$:

- dim(S): the dimension of the radical of L,
- dim(L/S): the dimension of L/S,
- L/S simple: whether L/S is simple or not.

The results are summarized in Table 6.1. We conclude the following:

1. In more than half of the cases we find a simple Lie algebra of dimension 28, i.e. it is isomorphic to D_4,
2. In 1090 cases, L/S is a simple Lie algebra of dimension 21, i.e. it is isomorphic to B_3,
3. In 460 cases, L/S is simple of dimension 15, i.e. it is isomorphic to A_3,
4. In 104 cases, L/S is simple of dimension 14, i.e. it is isomorphic to G_2,
5. In 14 cases, L/S is simple of dimension 10, i.e. it is isomorphic to B_2,
6. In 174 cases, L/S is simple of dimension 8, i.e. it is isomorphic to A_2,
7. In 3 cases, L/S is not simple and of dimension 6. Verification of these three cases gives that each of them is isomorphic to $A_1 \times A_1$,
8. In 23 cases, L/S is simple of dimension 3, i.e. it is isomorphic to A_1, and
9. In 1 case we find a solvable Lie algebra. This is of course the case where $f \equiv 0$.

	Count	dim(S)	dim(L/S)	L/S simple
1	2227	0	28	true
2	1090	7	21	true
3	460	13	15	true
4	104	14	14	true
5	14	18	10	true
6	174	20	8	true
7	3	22	6	false
8	23	25	3	true
9	1	28	0	N/A

Table 6.1: Summary of 4096 Lie algebras generated by 4 extremal elements

These results are in line with what Cohen et al. predicted in their paper [CSUW01, Section 7]. Moreover, they agree with Theorem 3.16.

Remark 6.6. There is one remark we should make on the implementation of the above in GAP. Although 36 polynomials above should give an ideal I we can work with, the Gröbner basis algorithm is known for its slowness and huge intermediate results. Practice shows that the 36 polynomials above indeed give huge intermediate results, and the calculations do not finish in less than 15 hours. Therefore, we make the following observation.

Let x be an extremal element of L, and let $m \in M$. Then $\mathrm{ad}_x^2 m$ is in L since x is an extremal element, so $\mathrm{ad}_x^2 m = 0$ in M/L. Thus we may add x^2, y^2, z^2, and u^2 to the ideal I. Now we find that the Gröbner basis calculation ends within seconds.

6.4 The Nilpotent Case and Beyond

In this section we introduce a modification of the method presented in the previous section. The motivation for the development of this algorithm is twofold. Firstly, we wish to study the case where a Lie algebra is generated by five extremal elements, and practice shows that our current method is far too slow. Secondly, we wish to study some degenerate cases, as described in Section 6.5. For the remainder of this section, we will refer to the algorithm from the previous section as 'the old algorithm', and the one to be introduced here as 'the new algorithm'. Our new algorithm breaks down into two steps: first, we find a basis for the Lie algebra on hand, and after that we try to find the evaluation of the bilinear form f on these basis elements. Some more observations on these algorithms can be found in Chapter 7.

For the first step, we use an approach similar to the old algorithm, with one major difference: we take the bilinear form f to be identically zero. This is implemented as follows.

The basis will be the set B, containing monomial basis elements. In every step of the algorithm, monomial basis elements of length $m+1$ are calculated from monomial basis elements of length m. For ease of reading, the length of a monomial m is denoted by $|m|$. The (non-commutative) Gröbner basis describing extremality will be denoted by GB. Calculating the Gröbner basis of a set X is denoted by $\texttt{Grobner}(X)$. 'Cleaning' a set of potential basis elements is denoted by $\texttt{Clean}(B')$. This amounts to repeatedly trying to divide $b \in B'$ by $B' \backslash \{b\}$, and removing b from B' if the division was successful.

1. $B := \{1, \ldots, N\}$;
2. $m := 1$;
3. $G := \emptyset$;
4. While B changed:
   (a) $G := G \cup \{iij - 2iji + jji \mid i, j \in B, |i| = 1, |j| = m\}$;
   (b) $GB := \texttt{Grobner}(G)$;
   (c) $B' := \{(ij - ji) \mod GB \mid i, j \in B, |i| = 1, |j| = m\}$;
   (d) $B := B \cup \texttt{Clean}(B')$;
   (e) $m := m + 1$;
5. Return B;

This algorithm was implemented in GAP.

The main advantages of this new algorithm over the old one are the following:

- The Gröbner basis GB in the new algorithm contains only homogeneous polynomials ($iij - 2iji + jji$), opposed to the old algorithm, where it mainly contained non-homogeneous polynomials ($iij - 2iji + jji + \alpha i$). This enables the use of a *graded* Gröbner basis algorithm, which gives an enormous speed-up.
- We do not need to know the primitive evaluations of f in advance (c.f. Theorem 6.2). Especially in the five generator case this is a great advantage, as it is far from obvious which are the primitives.

The main disadvantage is that we are no longer able to calculate the multiplication table of the resulting Lie algebra simultaneously. This is why we invented the following algorithm, operating on the resulting basis. In short, this algorithm automates the proof of Theorem 6.2. The core of this algorithm is `try_to_write_as_irreducibles`, which employs various methods such as associativity, extremality (i.e. $f(y, [x, [x, z]]) = f(x, z)f(y, x)$), the first two Premet rules (Lemma 3.3), and the Jacobi identity (in the sense that it rewrites $[a, [b, c]]$ to $-[b, [c, a]] - [c, [a, b]]$). A sketch of the way the algorithm works on a basis B (of size n) is given below. We write P for the set of primitives.

1. $P := \emptyset$;
2. $M :=$ empty $n \times n$ matrix;
3. For i in {1, ..., n} do
   (a) For j in {1, ..., i} do
      i. If `try_to_write_as_irreducibles`$(f(B_i, B_j))$ returns a combination p of elements from P,
         then store $M_{i,j} := p$,
         otherwise store $M_{i,j} := f(B_i, B_j)$ and set $P := P \cup f(B_i, B_j)$;
4. Return (P, M).

This algorithm was implemented in C++. Note that, under the assumption that `try_to_write_as_irreducibles` indeed returns a reduced form if that is possible, this algorithm finds both a set of primitives and the evaluation of f on the basis elements. We then use the theorems presented in Section 3.3, particularly Theorem 3.9 and Remark 3.15, to analyse the structure of the Lie algebra generated by N extremal elements.

In practice, this new algorithm returns a 28-dimensional basis for the four generator case in about a second instead of two minutes, and the evaluation of f on the basis elements is found in less than five seconds. The analysis of 4096 cases (see Section 6.3) now takes about two minutes instead of eight or nine hours. The gain in performance is obvious. Moreover, the second part of the algorithm (which is the 'hardest' part) was implemented in C++, so it is subject to regular compiler optimization and can easily be executed on an arbitrary computer.

6.5 Analysis of Degenerate Cases

In this section we consider special cases of Lie algebras generated by 4 extremal elements. We study them by means of pictures such as the ones below, which are easiest explained by means of an example: In the picture for $D^{1.1}$ we see 4 extremal generators, labeled $1, 2, 3$, and 4. A line is drawn between two generators i and j if $[i, j] \neq 0$, and if no line is drawn between i and j then $[i, j] = 0$. Consequently, for $D^{1.1}$ we have $[1, 2] = [1, 3] = [2, 3] = 0$. If $[i, j] = 0$ for at least one pair (i, j), we call it a *degenerate case*.

It is important to note that a picture does not uniquely determine a Lie algebra: if no line is drawn we know $\dim(\langle i, j\rangle) = 2$, and if a line is drawn we know $\dim(\langle i, j\rangle) = 3$. However, if a line is drawn, either $\langle i, j\rangle \cong \mathfrak{sl}_2$ or $\langle i, j\rangle \cong \mathfrak{h}$ (see Theorem 4.1). This means that though the dimension of the Lie algebra is determined by the picture alone, the dimension of its radical certainly is not.

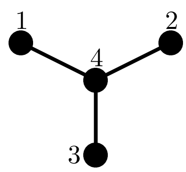
A Lie algebra is determined by a picture if we fix the evaluation of f. As indicated below, the most general case of $D^{1.1}$ has a 9-dimensional radical. This occurs for instance if we pick $f(1, 4) = f(2, 4) = f(3, 4) = 1$ (it is not hard to see that the other 9 primitive evaluations of f are equal to 0). However, if we would take $f \equiv 0$, we have a 12-dimensional radical (cf Theorem 3.9). Unless otherwise mentioned, when discussing 'the' Lie algebra induced by a picture we mean the most general one, i.e. the one with a radical of smallest dimension.

A straightforward approach to finding all possible degenerate cases is by considering all connected undirected simple graphs G on 4 vertices. We partition this space of connected undirected simple graphs by the sum of the degrees of the vertices. Then, for each partition (each possible sum S of degrees), we write down all possible combinations of positive degrees summing up to S. From these combinations we then draw 0 or more possible graphs.

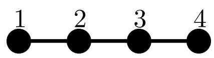
It is obvious that the sum S is always even, and the number of edges is $\frac{1}{2}S$. Moreover, since G should be connected, we must have at least 3 edges, so $S \geq 6$, and since G is simple, we have at most $\binom{4}{2} = 6$ vertices, so $S \leq 12$. Actually, the case where $S = 12$ is not a degenerate case, but we will include it for sake of completeness.

It should be noted that not all combinations of degrees summing up to S actually give a connected simple graph. Consider for example $S = 8$ and see that $8 = 3 + 3 + 1 + 1$ (case $D^{2.1}$ below). However, these degrees imply that 2 vertices are connected to all others, and the other 2 vertices are connected to only one other vertex, which is obviously impossible. From now on, we will write 'no associated Lie algebra' for cases such as this one.

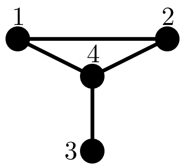
We used the algorithm described in the previous section to find the dimension of the Lie algebra and its radical, and we will use Theorem 3.16 and Appendix A to indicate which simple Lie algebras arises. These results were verified by Algorithm III, as described in Section 7.3.



$6 = 3{+}1{+}1{+}1$: $D^{1.1}$, where $[1, 2] = [1, 3] = [2, 3] = 0$, has dimension 12 and in general a radical of dimension 9, leaving a semi-simple part of dimension 3. This is A_1.

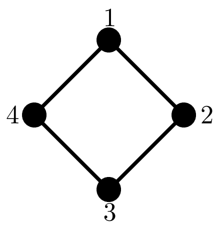


$6 = 2{+}2{+}1{+}1$: $D^{1.2}$, where $[1,3] = [1,4] = [2,4] = 0$, has dimension 10 and in general a trivial radical. This is C_2, see Section 9.2.



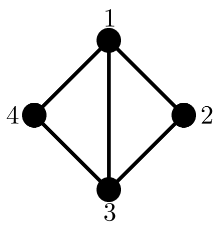
$8 = 3 + 3 + 1 + 1$: $D^{2.1}$, no associated Lie algebra.

$8 = 3 + 2 + 2 + 1$: $D^{2.2}$, where $[1,3] = [2,3] = 0$, has dimension 15 and in general a trivial radical. This is A_3.

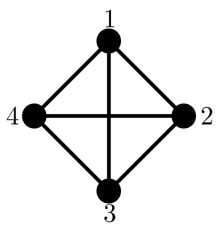


$8 = 2 + 2 + 2 + 2$: $D^{2.3}$, where $[1,3] = [2,4] = 0$, has dimension 15 and in general a trivial radical. This is A_3, see Section 9.1.

$10 = 3 + 3 + 3 + 1$: $D^{3.1}$, no associated Lie algebra.



$10 = 3 + 3 + 2 + 2$: $D^{3.2}$, where $[2,4] = 0$, has dimension 21 and in general a trivial radical. This must be B_3.



$12 = 3 + 3 + 3 + 3$: The most general case, dimension 28, discussed in previous sections.

Remark 6.7. As can be found in Appendix A, the simple Lie algebras generated by four extremal elements and no fewer are B_2, G_2, A_3, B_3, and D_4. In the analysis above, we find B_2, A_3, B_3, and D_4 as general instance of a degenerate case. However, G_2 is not found that way. It is, however, the semi-simple part of one of the instances of the 28-dimensional Lie algebra discussed in Section 6.3, and it is the semi-simple part of one of the instances of $D^{3.2}$ as well.

Chapter 7

Intermezzo: Algorithms

In this chapter we give an overview of the algorithms introduced in previous chapters, consider their advantages and disadvantages, and present a new algorithm. Algorithm I was introduced in Section 6.2, and depends on the knowledge of the primitive evaluations of f in advance, which is a big drawback. Given values for these primitive evaluations, it returns the appropriate Lie algebra. Algorithm II, introduced in Section 6.4, does not suffer from this drawback, but it is a bit slower. Given the number of extremal generators (and possibly some commutators), this algorithm returns a basis for the Lie algebra, and the evaluation of f on the basis elements. Using this data it is possible to calculate the minimal dimension of the radical. Algorithm III, to be introduced in Section 7.3, is a combination of these two algorithms.

7.1 Algorithm I

The first algorithm was introduced in Section 6.2 and was implemented in GAP [S+95], heavily depending on the GBNP package[CG05]. This algorithm is useful if we know the primitive evaluations of f. Given values for these evaluations, it returns the appropriate Lie algebra. Indeed, in the four generator case we know the 12 primitive evaluations of f, and we are therefore able to construct a quotient algebra containing the required Lie algebra. Given these primitive evaluations, this algorithm is not very fast, but it is not very slow either.

The main drawback of this algorithm is that if these primitive evaluations are not known, we are not able to give a basis for the required ideal and this algorithm is worthless. The main advantage of this algorithm is that once it is finished and we have fixed f, we can use all the functions GAP provides for working with Lie algebras. This means that we can ask for the `SemiSimpleType` or calculate its `LieSolvableRadical`, for example.

The (slightly annotated) implementation of this algorithm can be found in Appendix C.2.

Remark 7.1. Note that we sometimes fix the primitive evaluations of f in advance. This means that we carry out our calculations in $\mathbb{Q}$ instead of in $\mathbb{Q}[f(x,y), f(x,z), f(x,u), f(y,z), f(y,u), f(z,u), f(x,[y,z]), f(x,[y,u]), f(x,[z,u]), f(y,[z,u]), f(x,[y,[z,u]]), f(x,[z,[y,u]])]$ (in the four generator case), and this greatly speeds up the algorithm. Especially in the five generator cases this optimization is needed to make sure that the calculations end before this Master's thesis was finished. The drawback is that applying this optimization makes the algorithm return exactly one Lie algebra instead of an entire range of Lie algebras.

The following GAP listing shows how this algorithm was used for the degenerate case $D^{2,2}$ (see Section 6.5):

```
Read("algorithm1.g");;

N := 4;
commutators := [[1,3], [2,3]];
KI := BaseKI(N, commutators);

f21 := 1;
f41 := 2;
f42 := 3;
f43 := 4;
f124 := 5;

addext2(KI, elt(2), elt(1), f21);
addext2(KI, elt(4), elt(1), f41);
addext2(KI, elt(4), elt(2), f42);
addext2(KI, elt(4), elt(3), f43);
addext3(KI, elt(1), elt(2), elt(4), f124);

result := FindLieAlgebra(N, KI);
```

Executing this in GAP gives:

```
Input KI size 17
Size of GB is 26

Generating basis...
The basis has size 15. Cleaning...
Cleaned. The basis has size 15.

The resulting Lie Algebra basis has size 15

Constructing Table of Structure Constants...... Done.
Constructing Lie Algebra by Structure Constants...Done
  Algebra( Rationals, [ v.1, v.2, v.3, v.4, v.5, v.6, v.7,
  v.8, v.9, v.10, v.11, v.12, v.13, v.14, v.15 ] )
  Lie Algebra of dim: 15
  Radical of dim    : 0
  Simple            : true
  Semi Simple Type  : A3
Time used:
  Groebner Basis    : 70
  Lie Algebra Basis : 771
  Clean LA Basis:   : 0
  Lie Algebra Constr: 160
  SemiSimpleType    : 1122
  TOTAL             : 2123
```

7.2 Algorithm II

The second algorithm was introduced in Section 6.4 and was implemented in a combination of GAP (again using GBNP) and C++. This algorithm is useful if we do not know the primitive evaluations of f in advance. Given the number of extremal generators, and possibly some commutators, the first step of this algorithm generates a basis of the Lie algebra. The second step determines the evaluation of f on these basis elements, finding primitive evaluations of f in the process. The first step of this algorithm is rather fast, but the second step is too slow, especially in the five generator case.

The main drawback of this algorithm is that the second step is too slow for our purposes. Moreover, even if the second step finishes in time, we only have the minimal dimension of the radical (see Remark 3.15), and almost no information on the semi-simple part of the Lie algebra. This second drawback however, is not to be taken too seriously, as Theorem 3.16 and Appendix A mostly provide us with enough information to deduce the structure of the semi-simple part. The main advantage of this algorithm is that it does not need the primitive evaluations of f in advance, so we are able to calculate a basis for any Lie algebra generated by extremal elements. Moreover, as a side effect, it returns the primitive evaluations of f.

For sake of completeness, we have included the implementation of the first step of this algorithm in Appendix C.3. For example, for the degenerate case $D^{2.2}$, we would use this algorithm as follows:

```
Read("algorithm2a.g");;

N := 4;
commutators := [[1,3], [2,3]];
findeebasis(N, commutators);;
```

The result of executing this in GAP is as follows:

```
LOG: Starting groebner basis calculation at degree 1
LOG: Groebner basis calculation finished. Runtime: 40 msces.
LOG: Generating elements of length 2
LOG: Generated 8 elements of length 2
LOG: Cleaning basis, input elements: 8
LOG: Removing element 1 from basis.
LOG: Removing element 1 from basis.
LOG: Removing element 2 from basis.
LOG: Removing element 2 from basis.
LOG: Basis cleaned, output elements: 4
LOG: --> Adding 4 of 8 bracketings of length 2, 8 so far.
LOG: Starting groebner basis calculation at degree 4
<<Generation of elements of length 3 to 5 removed>>
LOG: Groebner basis calculation finished. Runtime: 70 msces.
LOG: Generating elements of length 6
LOG: Generated 0 elements of length 6
LOG: Basis of 15 elements found
LOG: Total time taken: 591 msecs.
```

We applied this algorithm to find the types of the degenerate cases of Lie algebras generated by four extremal elements (Section 6.5), and some generated by five extremal elements ($E^{1.x}$ and $E^{2.x}$ in Section 8.2). From $E^{3.x}$ onwards this algorithm was too slow.

7.3 Algorithm III

Comparing the advantages and the drawbacks of these algorithms, it seems like a good idea to use a combination of the two algorithms. Indeed, Algorithm II gives primitive evaluations of f as a side effect, which can serve as input for Algorithm I. In this approach, two characteristics of the algorithms are very convenient. Firstly, the second step of Algorithm II will return some (maybe all) primitive evaluations of f 'early' in the process, because of the way the basis elements are traversed. Secondly, if Algorithm I gives a result, we know it is a correct result. Indeed, we only insert relations that we know to be true into the basis of the proposed ideal. Once the result of Algorithm I is a

Lie algebra of dimension at most the dimension of the Lie algebra returned by the first step of Algorithm II, we know it is the Lie algebra we were searching.

The combination of these characteristics makes it possible to start with Algorithm II, cancel it halfway into its second step, and fiddle around with the primitive evaluations of f in Algorithm I until it gives a result. We applied this algorithm successfully to verify the types of the degenerate cases in Section 6.5, to verify the types of $E^{1.x}$ and $E^{2.x}$ in Section 8.2, and to find the types of $E^{3.x}$ in Section 8.2.

Remark 7.2. It should be noted that both Algorithm I and II (and therefore Algorithm III as well) can be easily adapted to work with degenerate cases, such as those in Sections 6.5 and 8.2. For example, if $[a, b] = 0$, we may add $ab - ba$ to the ideal in Algorithm I and the first step of Algorithm II, and we tell the second step of Algorithm II that $[a, b]$ can be reduced to 0.

Chapter 8

Lie Algebras Generated by Five Extremal Elements

In this chapter we study Lie algebras generated by five extremal elements. In Section 8.1 we show that the five generator case significantly differs from the two, three, and four generator cases: there is no semi-simple Lie algebra generated by five extremal elements of maximal dimension, i.e. a 537-dimensional Lie algebra generated by five extremal elements always has a non-trivial radical. In Section 8.2 we study the degenerate cases.

8.1 Structure

As stated in Section 4 of [CSUW01], the general Lie algebra generated by five extremal elements has dimension 537. Using the first step of Algorithm II we have been able to reproduce this dimension. Unfortunately, the second step is far too slow to calculate the radical of this Lie algebra.

Recall that a Lie algebra generated by two extremal elements has dimension 3, and there exists a semi-simple Lie algebra of dimension 3 generated by two extremal elements, namely A_1 (see Section 4.1). Similarly, there exists a semi-simple Lie algebra of dimension 8 generated by three extremal elements, namely A_2 (see Section 5.2), and there exists a semi-simple Lie algebra of dimension 28 generated by four extremal elements, namely D_4 (see Section 6.3). The following corollary, a direct consequence of Theorem 3.16, states that no semi-simple Lie algebra of dimension 537 generated by five extremal elements exists.

Corollary 8.1. There exists no simple or semi-simple Lie algebra of dimension 537 generated by 5 extremal elements.

Proof As stated in Chapter 8, a Lie algebra L generated by 5 extremal elements has dimension 537 in general. However, Theorem 3.16 implies that with 5 extremal elements we are only able to generate a small number of different semi-simple Lie algebras.

We consider the data in Appendix A, and note that a Lie algebra generated by 1 extremal element is solvable, so it does not contribute to a higher-dimensional semi-simple part. Now we see that with 5 extremal elements (and no fewer) we are able to generate the semi-simple Lie algebra $A_2 + A_1$ (dimension 11) or a simple Lie algebra of dimension at most 248 (i.e. E_8). This implies that no semi-simple Lie algebra of dimension 537 generated by five extremal elements exists. □

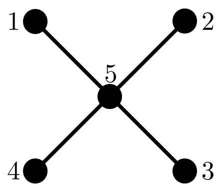
Unfortunately, our algorithms are too slow to find the generic radical. However, as found by Knopper [Kn004, Section 6.4], the simple Lie algebra E_8 only has a trivial quadratic module. This implies that the $537 - 248 = 289$ dimensional radical consists of 289 trivial 1-dimensional radicals.

Moreover, using Appendix A, we see that at least 20 extremal elements are needed to generate a 537-dimensional semi-simple Lie algebra: Either $E_8 + E_8 + A_5 + A_1 + A_1$, which has dimension $2 \cdot 248 + 35 + 2 \cdot 3 = 537$, and requires $2 \cdot 5 + 6 + 2 \cdot 2 = 20$ extremal elements, or $E_8 + E_8 + D_4 + B_2 + A_1$, which has dimension $2 \cdot 248 + 28 + 10 + 3 = 537$, and requires $2 \cdot 5 + 4 + 4 + 2 = 20$ extremal elements as well.

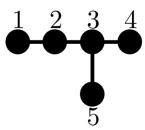
8.2 Analysis of Degenerate Cases

In this section we study degenerate cases of Lie algebras generated by five extremal elements. The meaning of the pictures below is explained in Section 6.5. We quickly see that the number of edges now is at least 4 and at most $\binom{5}{2} = 10$, so $8 \leq S \leq 20$. The most notable difference between these degenerate cases and the ones generated by four extremal elements, is that one combination of degrees may give rise to more than one graph: see for example $E^{2.4}$ and $E^{2.5}$.

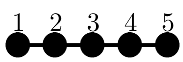
For the cases with 4 or 5 edges, we used Algorithm II to find the dimension of the most general instance and its radical, and Theorem 3.16 and Appendix A to indicate which simple Lie algebras arises. We used Algorithm III to verify these results. For the cases with 6 edges, we used Algorithm III to find the most general type. For the cases with 7 or more edges our algorithms were too slow, so we can only guess what their type is.



$8 = 4 + 1 + 1 + 1 + 1$: $E^{1.1}$, where $[1,2] = [1,3] = [1,4] = [2,3] = [2,4] = [3,4] = 0$, has dimension 28 and in general a radical of dimension 0. This implies that this Lie algebra is isomorphic to D_4.

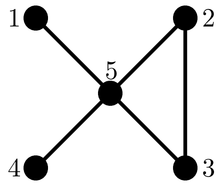


$8 = 3 + 2 + 1 + 1 + 1$: $E^{1.2}$, where $[1,3] = [1,4] = [1,5] = [2,4] = [2,5] = [4,5] = 0$, has dimension 20 and in general a radical of dimension 10, leaving a semi-simple part of dimension 10. This semi-simple part must be of type C_2.

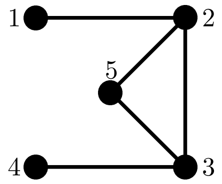


$8 = 2 + 2 + 2 + 1 + 1$: $E^{1.3}$, where $[1,3] = [1,4] = [1,5] = [2,4] = [2,5] = [3,5] = 0$, has dimension 15 and in general a radical of dimension 5, leaving a semi-simple part of dimension 10. This part must be of type C_2 as well.

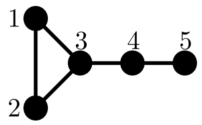
$10 = 4+3+1+1+1$: $E^{2.1}$, no associated Lie algebra.



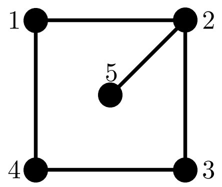
$10 = 4+2+2+1+1$: $E^{2.2}$, where $[1,2] = [1,3] = [1,4] = [2,4] = [3,4] = 0$, has dimension 36 and in general a trivial radical. This Lie algebra must be of type B_4.



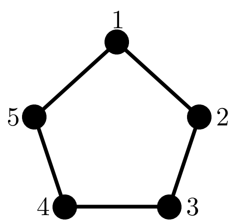
$10 = 3+3+2+1+1$: $E^{2.3}$, where $[1,3] = [1,4] = [1,5] = [2,4] = [4,5] = 0$, has dimension 30 and in general a radical of dimension 15, leaving a semi-simple part of dimension 15. This semi-simple part must be A_3.



$10 = 3+2+2+2+1$: $E^{2.4}$, where $[1,4] = [1,5] = [2,4] = [2,5] = [3,5] = 0$, has dimension 24 and in general a trivial radical. This Lie algebra must be isomorphic to A_4.

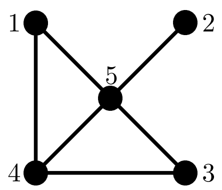


$10 = 3+2+2+2+1$: $E^{2.5}$, where $[1,3] = [1,5] = [2,4] = [3,5] = [4,5] = 0$, has dimension 30 and in general a radical of dimension 15, leaving a semi-simple part of dimension 15. This semi-simple part must be of type A_3.

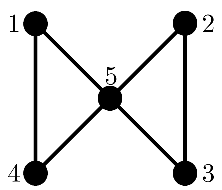


$10 = 2+2+2+2+2$: $E^{2.6}$, where $[1,3] = [1,4] = [2,4] = [2,5] = [3,5] = 0$, has dimension 24 and in general an empty radical. This must be A_4, see Section 9.1.

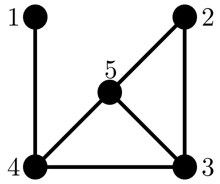
$12 = 4+4+2+1+1$: $E^{3.1}$, no associated Lie algebra.
$12 = 4+3+3+1+1$: $E^{3.2}$, no associated Lie algebra.



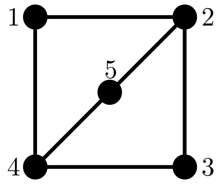
$12 = 4+3+2+2+1$: $E^{3.3}$, where $[1,2] = [1,3] = [2,3] = [2,4] = 0$, has dimension 52. This Lie algebra is of type F_4.



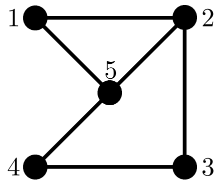
$12 = 4+2+2+2+2$: $E^{3.4}$, where $[1,2] = [1,3] = [2,4] = [3,4] = 0$, has dimension 45. This Lie algebra is of type D_5.



$12 = 3 + 3 + 3 + 2 + 1$: $E^{3.5}$, where $[1, 2] = [1, 3] = [1, 5] = [2, 4] = 0$, has dimension 45. This Lie algebra is of type D_5.

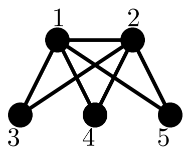


$12 = 3 + 3 + 2 + 2 + 2$: $E^{3.6}$, where $[1, 3] = [1, 5] = [2, 4] = [3, 5] = 0$, has dimension 52. This Lie algebra is of type F_4.

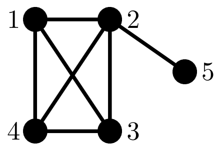


$12 = 3 + 3 + 2 + 2 + 2$: $E^{3.7}$, where $[1, 3] = [1, 4] = [2, 4] = [3, 5] = 0$, has dimension 45. This Lie algebra is of type D_5.

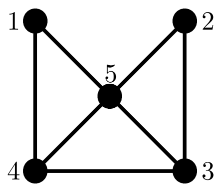
$14 = 4 + 4 + 4 + 1 + 1$: $E^{4.1}$, no associated Lie algebra.



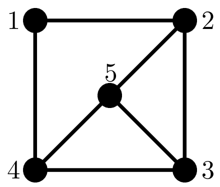
$14 = 4 + 4 + 2 + 2 + 2$: $E^{4.2}$, where $[3, 4] = [3, 5] = [4, 5] = 0$, has dimension 86. This could be E_6, which has dimension 78, and an 8-dimensional radical (see Section 8.3).



$14 = 4 + 3 + 3 + 3 + 1$: $E^{4.3}$, where $[1, 5] = [3, 5] = [4, 5] = 0$, has dimension 78. This probably is E_6 (see Section 8.3).

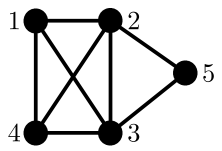


$14 = 4 + 3 + 3 + 2 + 2$: $E^{4.4}$, where $[1, 2] = [1, 3] = [2, 4] = 0$, has dimension 78. This probably is E_6 (see Section 8.3).

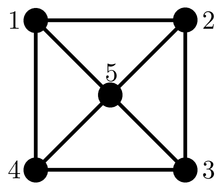


$14 = 3 + 3 + 3 + 3 + 2$: $E^{4.5}$, where $[1, 3] = [1, 5] = [2, 4] = 0$, has dimension 78. This probably is E_6 (see Section 8.3).

$16 = 4+4+4+3+1$: $E^{5.1}$, no associated Lie algebra.
$16 = 4+4+4+2+2$: $E^{5.2}$, no associated Lie algebra.

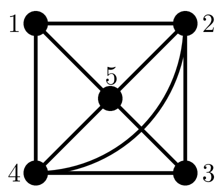


$16 = 4+4+3+3+2$: $E^{5.3}$, where $[1,5] = [4,5] = 0$, has dimension 134. Our best guess is that this is E_7 and a one dimensional radical (see Section 8.3).

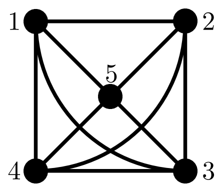


$16 = 4{+}3{+}3{+}3{+}3$: $E^{5.4}$, where $[1,3] = [2,4] = 0$, has dimension 133. We guess that this is E_7 (see Section 8.3).

$18 = 4+4+4+4+2$: $E^{6.1}$, no associated Lie algebra.



$18 = 4+4+4+3+3$: $E^{6.2}$, where $[1,3] = 0$, has dimension 249. This most probably is E_8 and an additional one dimensional radical.



$20 = 4{+}4{+}4{+}4{+}4$: The most general case, dimension 537.

Remark 8.2. Similarly to the discussion in Remark 6.7 we observe which simple Lie algebras generated by five extremal elements and no fewer occur as general instance of a degenerate case. From the data in Appendix A we see that $A_4, B_4, D_5, F_4, E_6, E_7$, and E_8 are generated by five extremal elements and no fewer. We suspect that all of these occur as general instance of a degenerate case. For A_4, B_4, D_5, and F_4 we showed this, and for E_6, E_7, and E_8 we think this is the case.

This would imply that G_2 is an exception among the exceptional Lie algebras: it is the only one that is not the general instance of a degenerate case.

8.3 Isomorphic Degenerate Cases

As mentioned in the introduction of the previous section, we were unable to calculate the Lie type of cases $E^{4.x}$, $E^{5.x}$, and $E^{6.x}$. In this section we give some more information on $E^{4.x}$ and $E^{5.x}$. To do that, we introduce a way of rewriting a picture such as those in the previous section to an (almost) isomorphic picture.

We adopt the usual setting of a Lie algebra L over a field $\mathbb{F}$ generated by extremal elements, and let $x \in \mathcal{E}$. Recall that $\exp(x,s) = 1 + s\mathrm{ad}_x + \frac{1}{2}s^2\mathrm{ad}_x^2$, and remember that $\exp(x,s)$ is a Lie algebra automorphism (cf Definition 3.4). We again consider a Lie algebra described by a picture, and study what happens if we replace one of the 'nodes' y by $\exp(x,s)y$. We write y^x for $\exp(x,s)y$ and note that nothing changes if

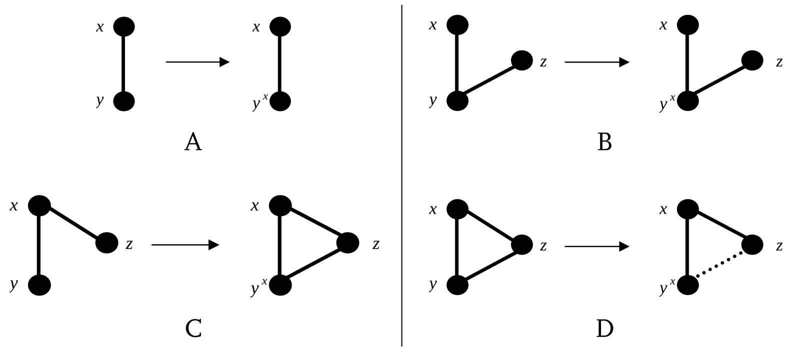


Figure 8.1: Replacing y by $\exp(x, s)y$

$[x, y] = 0$ since $y^x = y$. For the remaining situations, we distinguish four cases. See Figure 8.1.

A. If $\langle x, y\rangle \cong \mathfrak{sl}_2$ then we have

$$\langle x, y^x\rangle = \langle x^x, y^x\rangle = \langle x, y\rangle^x.$$

Since exp is an automorphism, we have $\langle x, y^x\rangle \cong \mathfrak{sl}_2$. This explains Figure 8.1A.

B. If $\langle x, y\rangle \cong \mathfrak{sl}_2$ and z is such that $\langle y, z\rangle \cong \mathfrak{sl}_2$ and $[x, z] = 0$, then we have $\langle x, y^x\rangle$ as above, and

$$\langle y^x, z\rangle = \langle y^x, z^x\rangle = \langle y, z\rangle^x.$$

So $\langle y^x, z\rangle \cong \mathfrak{sl}_2$. This explains Figure 8.1B.

C. If $\langle x, y\rangle \cong \mathfrak{sl}_2$ and z is such that $\langle x, z\rangle \cong \mathfrak{sl}_2$ and $[y, z] = 0$, then $[y^x, z] = s[z, [x, y]] + \frac{1}{2}s^2 f(x, y)[x, z]$. Now we have $\langle x, y^x\rangle$ as above and

$$\begin{aligned} [z, [y^x, z]] &= -f(z, y^x)z \\ &= -(sf(z, [x, y]) + \frac{1}{2}s^2 f(x, y)f(z, x))z \\ &= -(sf(x, [y, z]) + \frac{1}{2}s^2 f(x, y)f(z, x))z \\ &= -\frac{1}{2}s^2 f(x, y)f(x, z)z. \end{aligned}$$

Moreover, straightforward calculations show that

$$[y^x, [y^x, z]] = (\frac{1}{2}s^2 f(x, y)f(x, z))(y+s[x, y]+\frac{1}{2}s^2 f(x, y)x) = \frac{1}{2}s^2 f(x, y)f(x, z)y^x.$$

This means that if we set $\alpha = \frac{1}{2}s^2 f(x, y)f(x, z)$, we have $[z, [y^x, z]] = -\alpha z$ and $[y^x, [y^x, z]] = \alpha y^x$. So, provided that $\alpha \neq 0$, we have $\langle y^x, z\rangle \cong \mathfrak{sl}_2$. Since $\langle x, y\rangle \cong \mathfrak{sl}_2$ and $\langle x, z\rangle \cong \mathfrak{sl}_2$, we have $f(x, y) \neq 0$ and $f(x, z) \neq 0$, so picking any $s \neq 0$ gives $\langle y^x, z\rangle \cong \mathfrak{sl}_2$. This explains Figure 8.1C.

D. If $\langle x, y\rangle \cong \mathfrak{sl}_2$ and z is such that both $\langle y, z\rangle$ and $\langle x, z\rangle$ are isomorphic to $\mathfrak{sl}_2$, we have $[y^x, z] = [y, z] - s[z, [x, y]] + \frac{1}{2}s^2 f(x, y)[x, z]$. Now we have $\langle x, y^x\rangle$ as above and

$$\begin{aligned} [z, [y^x, z]] &= -(f(z, y^x)z) \\ &= -(f(y, z) + sf(x, [y, z]) + \frac{1}{2}s^2 f(x, y)f(x, z))z. \end{aligned}$$

Moreover, straightforward calculations show that

$$[y^x, [y^x, z]] = (f(y, z) + sf(x, [y, z]) + \frac{1}{2}s^2 f(x, y)f(x, z))y^x.$$

This implies that if we set $\alpha = f(y, z) + sf(x, [y, z]) + \frac{1}{2}s^2 f(x, y)f(x, z)$, we have $[z, [y^x, z]] = -\alpha z$ and $[y^x, [y^x, z]] = \alpha y^x$. This means that $\langle y^x, z\rangle \cong \mathfrak{sl}_2$ if we have $\alpha \neq 0$. On the other hand, if $\alpha = 0$, then $\langle y^x, z\rangle$ is nilpotent (cf Case 2 of Theorem 4.1). Solving the quadratic equation $\alpha = 0$ to s gives

$$s = \frac{1}{f(x, y)f(x, z)}(-f(x, [y, z]) \pm \sqrt{f(x, [y, z])^2 - 2f(x, y)f(x, z)f(y, z)}).$$

Always at least one of these two solutions is non-zero: Indeed, if $f(x, [y, z]) = 0$, both solutions are non-zero as $f(x, y), f(x, z), f(y, z) \neq 0$. If $f(x, [y, z]) \neq 0$, it depends on the value of $f(x, [y, z])^2 - 2f(x, y)f(x, z)f(y, z)$ whether we have one or two non-zero solutions. Note that it may require a field extension to be able to use such a non-zero solution, though. This explains Figure 8.1D.

Now to see why we can use these rewrite rules, we consider L the original Lie algebra generated by extremal elements, so $L = \langle x, y, z, \ldots\rangle$, and L' the new one, i.e. $L' = \langle x, y^x, z, \ldots\rangle$. We show that $y \in L'$. Observe $y = y^x - s[x, y] - \frac{1}{2}s^2 f(x, y)x$ and obviously $y^x \in L'$ and $\frac{1}{2}s^2 f(x, y)x \in L'$, and furthermore $[x, y] \in L'$ since $[x, y^x] = [x, y] + sf(x, y)x$ and $[x, y^x] \in L'$ and $sf(x, y)x \in L'$. So indeed $y \in L'$. However, we must be careful when using these rules: if we start with a semi-simple Lie algebra, after applying rewrite rules the evaluation of f may have changed, so the result may be a Lie algebra containing a non-trivial radical.

So below we show that at least one of the Lie algebras induced by the picture we end up with after applying the rewrite rules, is isomorphic to the most general Lie algebra induced by the picture we started with. We denote this relation by $\to$. We show $E^{4.4} \to E^{4.3}$, $E^{4.4} \to E^{4.5}$, $E^{4.5} \to E^{4.2}$, and $E^{5.4} \to E^{5.3}$. First, we consider $E^{4.x}$:

- $E^{4.4} \to E^{4.3}$: Pick $x = 5$ and $y = 3$. Then either $\langle 2, 3\rangle$ or $\langle 3, 4\rangle$ disappears into the radical (depending on our choice of s). We choose s such that $\langle 2, 3\rangle$ disappears. $\langle 1, 3\rangle$ appears, and we are done.
- $E^{4.4} \to E^{4.5}$: Pick $x = 4$ and $y = 1$. Then $\langle 1, 5\rangle$ goes into the radical, $\langle 1, 3\rangle$ appears, and we are done.
- $E^{4.5} \to E^{4.2}$: Pick $x = 2$ and $y = 5$. Then $\langle 5, 3\rangle$ goes into the radical, $\langle 1, 3\rangle$ appears, and we are done.

Now we consider $E^{5.x}$:

- $E^{5.4} \to E^{5.3}$: Pick $y = 1$ and $x = 5$. Then either $\langle 1, 2\rangle$ or $\langle 1, 4\rangle$ disappears into the radical (depending on our choice of s). We choose s such that $\langle 1, 2\rangle$ disappears. $\langle 1, 3\rangle$ appears, and we are done.

So for the cases we were unable to calculate using one of the algorithms described in Chapter 7, it seems very probably that only three essentially different Lie algebras remain, probably E_6, E_7, and E_8 (dimensions 78, 133, and 248).

Chapter 9

Lie Algebras Generated by n Extremal Elements

This chapter contains statements on Lie algebras generated by arbitrary many extremal elements, fulfilling certain demands. More specifically, we show which degenerate cases are isomorphic to A_n ($n \geq 1$), and which degenerate cases are isomorphic to C_n ($n \geq 2$). Furthermore, in Section 9.4 we give three conjectures, in which we suggest degenerate cases isomorphic to B_n ($n \geq 4$) and D_n ($n \geq 5$).

9.1 A_n

Theorem 9.1. *Let L be a Lie algebra generated by n (where $n \geq 2$) extremal elements $x_1, \ldots, x_n$, such that*

$$[x_i, x_j] = 0 \text{ if } |i-j| \geq 2 \text{ and } \{i,j\} \neq \{1,n\}, \tag{9.1}$$

and

$$[x_i, x_j] \neq 0 \text{ if } |i-j| = 1 \text{ or } \{i,j\} = \{1,n\} \tag{9.2}$$

(cf $D^{2.3}$ and $E^{2.6}$ in Sections 6.5 and 8.2, respectively). Then L is linearly generated by the following $n^2 - 1$ monomials (for ease of reading, we identify x_{n+i} and x_i if $i \geq 1$):

$$\begin{array}{ll} [x_i,[x_{i+1},[\ldots,[x_{i+l-1},x_{i+l}]]]], & 1 \leq i \leq n, 0 \leq l \leq n-2 \\ [x_i,[x_{i+1},[\ldots,[x_{i+n-2},x_{i+n-1}]]]], & 1 \leq i < n, \end{array} \tag{9.3}$$

i.e. $n(n-1)$ monomials of length less than n, and $n-1$ monomials of length n.

Proof First, we prove that every monomial of L (of length up to n) can be written as a linear combination of monomials of the above form. For now, we will identify x_{n+i} and x_i if $i \geq 1$. We will prove this statement by induction on the length, the case where the length of the monomial is 1 being trivial.

So suppose that we proved the statement for monomials of length up to $l+1$ ($l+1 \neq n$), and let $x \in L$ be a monomial of length $l+2$. By the induction hypothesis, we know that

$$x = [x_p,[x_i,[x_{i+1},[\ldots,[x_{i+l-1},x_{i+l}]]]]] \tag{9.4}$$

for some p and some i, with $1 \leq p, i \leq n$. If $p = i-1$, we proved the induction step. If $p = i$ we apply extremality of x_i, and the induction hypothesis gives the induction step. If $p = i+1$ we apply (P2), and the induction hypothesis gives the induction step.

In all other cases, we note (by the Jacobi identity) that

$$x \quad = \quad [x_i,[x_p,[x_{i+1},[\ldots,[x_{i+l-1},x_{i+l}]]]]]$$

$$\begin{aligned} & +[[x_i, x_p], [x_{i+1}, [\ldots, [x_{i+l-1}, x_{i+l}]]]]] \\ = \quad & [x_i, [x_p, [x_{i+1}, [\ldots, [x_{i+l-1}, x_{i+l}]]]]]], \end{aligned} \tag{9.5}$$

since x_i and x_p commute. We repeatedly use this observation, and if $i \leq p \leq i+l$, we can sooner or later apply rule 2, and the induction hypothesis gives the induction step. Otherwise, we end up with

$$x = [x_i, [x_{i+1}, [\ldots, [x_{i+l-1}, [x_{i+l}, x_p]]]]]. \tag{9.6}$$

Now if $p = i + l + 1$, the induction step is proved immediately. Otherwise, as we excluded the case that $p = i + l$ already, we have $x = 0$, and the induction step is proved as well. So every element of L of length up to n can indeed be written as a linear combination of monomials of the above form.

Suppose $x \in L$ is a monomial of length $n+1$. Then, by the previous statement, we may assume

$$x = [x_p, [x_i, [x_{i+1}, [\ldots, [x_{i+n-2}, x_{i+n-1}]]]]] \tag{9.7}$$

for some p and some i, with $1 \leq p, i \leq n$. If $p = i$ or $p = i+1$ extremality of x_i and x_{i+1}, respectively, reduces x to a monomial of length at most n. Otherwise, x_p and x_i commute, and we have, similar to the observation in Equation 9.5:

$$x = [x_i, [x_p, [x_{i+1}, [\ldots, [x_{i+n-2}, x_{i+n-1}]]]]]. \tag{9.8}$$

Now we are either able to apply rule 2 after at most $n-1$ applications of this step, or we end up in the case where $p = i+n-1$ and $x = 0$. This proves that every element of L of arbitrary length can be written as a linear combination of monomials of the above form.

It remains to prove that one of the monomials of length n may be omitted. From now on, we will no longer identify x_{n+i} and x_i. Note that

$$\begin{aligned} & [x_1, [x_2, [x_3, [\ldots, [x_{n-1}, x_n]]]]] \\ = \quad & [x_2, [x_1, [x_3, [\ldots, [x_{n-1}, x_n]]]]] \\ & +[[x_1, x_2], [x_3, [\ldots, [x_{n-1}, x_n]]]]. \end{aligned} \tag{9.9}$$

The first term is equal to $-[x_2, [x_3, [\ldots, [x_{n-1}, [x_n, x_1]]]]]$, and for the second term we have

$$\begin{aligned} & [[x_1, x_2], [x_3, [x_4, [\ldots, [x_{n-1}, x_n]]]]] \\ = \quad & [x_3, [[x_1, x_2], [x_4, [\ldots, [x_{n-1}, x_n]]]]] \\ & +[[[x_1, x_2], x_3], [x_4, [\ldots, [x_{n-1}, x_n]]]]. \end{aligned} \tag{9.10}$$

Again, the first term is equal to $-[x_3, [x_4, [\ldots, [x_n, [x_1, x_2]]]]]$, and the second term is equal to

$$[[x_1, [x_2, x_3]], [x_4, [\ldots, [x_{n-1}, x_n]]]]. \tag{9.11}$$

Generalizing these observations shows:

$$\begin{aligned} & [[x_1, [x_2, [\ldots, [x_{j-1}, x_j]]]], [x_{j+1}, [x_{j+2}, [\ldots, [x_{n-1}, x_n]]]]] \\ = \quad & [x_{j+1}, [[x_1, [x_2, [\ldots, [x_{j-1}, x_j]]]], [x_{j+2}, [\ldots, [x_{n-1}, x_n]]]]] \\ & +[[[x_1, [x_2, [\ldots, [x_{j-1}, x_j]]]], x_{j+1}], [x_{j+2}, [\ldots, [x_{n-1}, x_n]]]] \\ = \quad & -[x_{j+1}, [x_{j+2}, [\ldots, [x_n, [x_1, [x_2, [\ldots, [x_{j-1}, x_j]]]]]]]] \qquad \text{(+)} \\ & +[[x_1, [x_2, [\ldots, [x_{j-1}, [x_j, x_{j+1}]]]]], [x_{j+2}, [\ldots, [x_{n-1}, x_n]]]]. \qquad \text{(++)} \end{aligned} \tag{9.12}$$

(+) is valid because $x_1, \ldots, x_j$ commute with $x_{j+2}, \ldots, x_n$, and (++) is valid because x_{j+1} commutes with $x_1, \ldots, x_{j-1}$.

Applying this rule $n-3$ times to $[x_1,[x_2,[x_3,[\ldots,[x_{n-1},x_n]]]]]$, taking the first of the two terms along every time, leaves us with $[[x_1,[x_2,[\ldots,[x_{n-3},x_{n-2}]]]],[x_{n-1},x_n]]$. For this bracketing, we have

$$\begin{aligned} & [[x_1,[x_2,[\ldots,[x_{n-3},x_{n-2}]]]],[x_{n-1},x_n]] \\ = \; & [x_{n-1},[[x_1,[x_2,[\ldots,[x_{n-3},x_{n-2}]]]],x_n]] \\ & +[[[x_1,[x_2,[\ldots,[x_{n-3},x_{n-2}]]]],x_{n-1}],x_n] \\ = \; & -[x_{n-1},[x_n,[x_1,[x_2,[\ldots,[x_{n-3},x_{n-2}]]]]]] \\ & -[x_n,[x_1,[x_2,[x_3,[\ldots,[x_{n-2},x_{n-1}]]]]]]. \end{aligned} \tag{9.13}$$

This shows that

$$\begin{aligned} & [x_1,[x_2,[x_3,[\ldots,[x_{n-1},x_n]]]]] \\ = \; & -[x_2,[x_3,[x_4,[\ldots,[x_n,x_1]]]]] \\ & -[x_3,[x_4,[x_5,[\ldots,[x_1,x_2]]]]] \\ & \vdots \\ & -[x_n,[x_1,[x_2,[\ldots,[x_{n-2},x_{n-1}]]]]]. \end{aligned} \tag{9.14}$$

So indeed, we may omit $[x_n,[x_1,[x_2,[\ldots,[x_{n-2},x_{n-1}]]]]]$ from our proposed basis.

This completes the proof of Theorem 9.1. □

We proved that a Lie algebra such as the degenerate $D^{2.3}$ and $E^{2.6}$ (see Section 6.5 and Section 8.2, respectively) in general has dimension n^2-1. The attentive reader may have noticed that this is exactly the dimension of A_{n-1} (see Section 2.4). The remainder of this section shows that indeed a Lie algebra as above is isomorphic to A_{n-1}.

Theorem 9.2. *Let L be a Lie algebra generated by n (where $n \geq 2$) extremal elements $x_1,\ldots,x_n$, such that*

$$[x_i,x_j]=0 \text{ if } |i-j|\geq 2 \text{ and } \{i,j\}\neq\{1,n\}, \tag{9.15}$$

$$\langle x_i,x_j\rangle \cong \mathfrak{h} \text{ if } |i-j|=1 \text{ or } \{i,j\}=\{1,n\}, \tag{9.16}$$

and

$$f(x_1,[x_2,\ldots,[x_{n-1},x_n]])\neq 0. \tag{9.17}$$

Then L is isomorphic to the simple Lie algebra A_{n-1}.

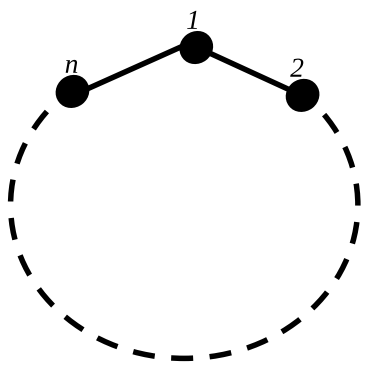


Figure 9.1: Extremal elements generating A_{n-1}

Proof We denote the matrix M with $M_{ij} = 1$ and all other entries 0 by e_{ij}. Similarly, we denote the vector v with $v_i = 1$ and all other entries 0 by e_i. As noted in [Hum72, Section 1.1], A_{n-1} is generated by the $n \times n$ matrices M with $\mathrm{tr}(M) = 0$, i.e:

- the $n(n-1)$ matrices e_{ij}, with $1 \leq i, j, \leq n$ and $i \neq j$, and
- the $n-1$ matrices $e_{i,i} - e_{i+1,i+1}$, with $1 \leq i < n$.

We take φ to be the morphism from L into $\mathfrak{sl}_n$, defined as follows:

$$\begin{aligned} x_i &\mapsto \begin{cases} e_{i,i+1} & \text{if } 1 \leq i < n \\ e_{n,1} & \text{if } i = n \end{cases} \\ [x_i, x_j] &\mapsto \varphi(x_i)\varphi(x_j) - \varphi(x_j)\varphi(x_i). \end{aligned}$$

We will show how φ naturally induces a bijection between the above basis elements of A_{n-1} and the basis elements of L, defined in Theorem 9.1. We will distinguish three cases: e_{ij} for $i < j$, e_{ij} for $i > j$, and $e_{i,i} - e_{i+1,i+1}$.

Case 1: $e_{ij}, 1 \leq i < j \leq n$. We claim

$$e_{ij} = \varphi([x_i, [x_{i+1}, [\ldots, [x_{j-2}, x_{j-1}]]]]). \tag{9.18}$$

We will prove this by induction on $j - i$. If $j - i = 1$ we are fine, since $e_{i,i+1} = \varphi(x_i)$. Suppose we proved the claim for $j - i \leq k$, for some k, and let i, j be such that $j - i = k+1$. Let $M = e_{i,i+1}$ and $N = e_{i+1,j}$, and observe that $MN - NM = e_{ij}$. Furthermore, by the induction hypothesis $M = \varphi(x_i)$ and $N = \varphi([x_{i+1}, [\ldots, [x_{j-2}, x_{j-1}]]])$, so $e_{ij} = \varphi([x_i, [x_{i+1}, [\ldots, [x_{j-2}, x_{j-1}]]]])$ as required.

Case 2: $e_{ij}, 1 \leq j < i \leq n$. We claim

$$e_{ij} = \varphi([x_i, [x_{i+1}, [\ldots, [x_n, [x_1, [x_2, [\ldots, [x_{j-2}, x_{j-1}]]]]]]]]). \tag{9.19}$$

First, take $i = n$. If $j = 1$ then $e_{ij} = e_{n,1} = \varphi(x_n)$. If $j > 1$ then take $M = e_{n,j-1}$ and $N = e_{j-1,j}$ and note that $MN - NM = e_{n,j}$ and

$$\begin{aligned} e_{ij} &= \varphi([[x_n, [\ldots, [x_{j-3}, x_{j-2}]]], x_{j-1}]) \\ &= \varphi([x_n, [x_1, [x_2, [\ldots, [x_{j-2}, x_{j-1}]]]]]), \end{aligned} \tag{9.20}$$

by the induction hypothesis and the fact that x_{j-1} commutes with x_n and $x_1, \ldots, x_{j-3}$. This completes the proof for the case where $j < i$ and $i = n$.

For the case where $i < n$ we apply induction on i, the case where $i = n$ being proved above. Now let $i < n$, and take $M = e_{i,i+1}$ and $N = e_{i+1,j}$. Now $MN - NM = e_{ij}$, as requested, and

$$e_{ij} = \varphi([x_i, [x_{i+1}, [x_{i+2}, [\ldots, [x_n, [x_1, [x_2, [\ldots, [x_{j-2}, x_{j-1}]]]]]]]]]), \tag{9.21}$$

by the induction hypothesis. This completes the proof for the case $e_{ij}, 1 \leq j < i \leq n$.

Case 3: $e_{i,i} - e_{i+1,i+1}, 1 \leq i < n$. We claim

$$e_{i,i} - e_{i+1,i+1} = \varphi([x_i, [x_{i+1}, [\ldots, [x_n, [x_1, [x_2, [\ldots, [x_{i-2}, x_{i-1}]]]]]]]]). \tag{9.22}$$

We take $M = e_{i,i+1}$ and $N = e_{i+1,i}$. Obviously, $MN - NM = e_{i,i} - e_{i+1,i+1}$. Moreover, we have $M = \varphi(x_i)$ by definition, and $N = \varphi([x_{i+1}, [x_{i+2}, [\ldots, [x_n, [x_1, [x_2, [\ldots, [x_{i-2}, x_{i-1}]]]]]]]])$ by case 2. Now $e_{i,i} - e_{i+1,i+1}$ is as required.

Conclusion: The above three cases show that each basis element of A_{n-1} is the φ of a unique monomial in the basis of L. Indeed, Case 1 proves this for $\frac{1}{2}n(n-1)$

cases where the monomial is of length smaller than n, and does not contain x_n. Case 2 proves this for $\frac{1}{2}n(n-1)$ where the monomial is of length smaller than n, and does contain x_n. Lastly, Case 3 proves this for $n-1$ cases where the monomial is of length n. This indeed amounts to n^2-1 cases in general.

Let i, j be such that $i+1=j$. Since $\langle x_i, x_j\rangle \cong \mathfrak{h}$, we have $f(x_i, x_j) = 0$. This corresponds to the fact that

$$\begin{aligned}&[\varphi(x_i), [\varphi(x_i), \varphi(x_j)]] = \\ &\quad e_{i,i+1}e_{i,i+1}e_{i+1,i+2} - 2e_{i,i+1}e_{i+1,i+2}e_{i,i+1} + e_{i+1,i+2}e_{i,i+1}e_{i,i+1} = 0.\end{aligned} \tag{9.23}$$

(similarly for the case where $i=n$ and $j=1$). However, it cannot be the case that f is identically zero, because that would mean $\mathrm{Rad}(L) = \mathrm{Rad}(f) = L$ by Theorem 3.9. As proved in Case 2 above, $\varphi([x_2, \ldots, [x_{n-1}, x_n]]) = e_{21}$, so

$$\begin{aligned}\varphi([x_1, [x_1, [x_2, \ldots, [x_{n-1}, x_n]]]]) &= e_{12}e_{12}e_{21} - 2e_{12}e_{21}e_{12} + e_{21}e_{12}e_{12} \\ &= -2e_{12}e_{21}e_{12} = -2e_{11}e_{12} = -2e_{12},\end{aligned} \tag{9.24}$$

so $f(x_1, [x_2, \ldots, [x_{n-1}, x_n]]) = -2$ in general. This implies that by multiplying for example $\varphi(x_1)$ with a suitable scalar, we have an isomorphism for arbitrary non-zero $f(x_1, [x_2, \ldots, [x_{n-1}, x_n]])$, as required. Most of the other evaluations of f are zero. For example, if $n \geq 5$, then

$$\begin{aligned}f(x_1, [x_2, [x_3, x_4]]) &= -f(x_2, [x_1, [x_3, x_4]]) = \\ &f(x_2, [x_3, [x_4, x_1]]) + f(x_2, [x_4, [x_1, x_3]]) = 0.\end{aligned} \tag{9.25}$$

This completes the proof of Theorem 9.2. □

9.2 C_n

Theorem 9.3. *Let L be a Lie algebra generated by n (where $n \geq 2$) extremal elements $x_1, \ldots, x_n$, such that*

$$[x_i, x_j] = 0 \text{ if } |i-j| \geq 2, \tag{9.26}$$

and

$$[x_i, x_j] \neq 0 \text{ if } |i-j| = 1 \tag{9.27}$$

(cf $D^{1,2}$ and $E^{1,3}$ in Sections 6.5 and 8.2, respectively). Then L is linearly generated by the following $\frac{1}{2}n^2 + \frac{1}{2}n$ monomials:

$$[x_i, [x_{i+1}, [\ldots, [x_{i+l-1}, x_{i+l}]]]], 1 \leq i \leq n-l, 0 \leq l \leq n-1. \tag{9.28}$$

Proof This proof is along the lines of the proof of Theorem 9.1, and depends on Identity 9.5. We again prove by induction that every monomial can be written as a linear combination of the proposed basis elements. The case where $l=0$ is trivial, so suppose that we proved the statement for monomials of length up to $l+1$, and let $x \in L$ be a monomial of length $l+2$. By the induction hypothesis, we know that

$$x = [x_p, [x_i, [x_{i+1}, [\ldots, [x_{i+l-1}, x_{i+l}]]]]] \tag{9.29}$$

for some p and some i, with $1 \leq p, i \leq n$. If $p = i-1$, we proved the induction step. If $p=i$ we apply extremality of x_i, and the induction hypothesis gives the induction step. If $p = i+1$ we apply (P2), and the induction hypothesis gives the induction step.

In all other cases, we apply the observation in Equation 9.5 until either:

1. We find $x=0$ (if $p < i-1$ or $p > i+l+1$), and we are done immediately, or

2. We are able to apply rule 2 (if $i+2 \leq p \leq i+l$), and we apply the induction hypothesis to show the induction step, or

3. We find
$$x = -[x_i, [x_{i+1}, [\ldots, [x_{i+l}, x_{i+l+1}]]]], \tag{9.30}$$
and we see the induction step as well.

Reasoning as in the proof of Theorem 9.1 shows that all monomials of length more than n can be written as linear combination of the proposed basis elements.

This indeed amounts to $\frac{1}{2}n^2 + \frac{1}{2}n$ monomials:

$$\sum_{l=0}^{n-1} n - l = n^2 - \sum_{l=0}^{n-1} l = n^2 - \sum_{l=1}^{n-1} l = n^2 - \frac{1}{2}(n-1)n = \frac{1}{2}n^2 + \frac{1}{2}n. \tag{9.31}$$

This completes the proof of Theorem 9.3. $\square$

We now arrive at the central theorem of this section.

Theorem 9.4. *Let L be a Lie algebra over the field $\mathbb{F}$ generated by $2l$ (where $l \geq 1$) extremal elements $x_1, \ldots, x_{2l}$, such that*

$$[x_i, x_j] = 0 \text{ if } |i-j| \geq 2, \tag{9.32}$$

and

$$\langle x_i, x_j \rangle \cong \mathfrak{sl}_2 \text{ if } |i-j| = 1. \tag{9.33}$$

Then L is isomorphic to the simple Lie algebra C_l.

Before proceeding to the proof of this theorem, we introduce a few notions and we prove some lemmas. As defined in Section 2.4, the simple Lie algebra C_l is $\mathfrak{sp}(V)$, where V is a vectorspace of dimension $2l$. The Lie algebra $\mathfrak{sp}(V)$ was defined as the algebra of all endomorphisms x of V satisfying $g(x(v), w) = -g(v, x(w))$, where g is the skew-symmetric bilinear function defined by the matrix G:

$$G = \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix}. \tag{9.34}$$

Furthermore, we define the map E from V into $\mathrm{End}(V)$ as follows:

$$E_a : V \to V, x \mapsto g(a,x)a, \text{ where } a \in V. \tag{9.35}$$

For all endomorphisms x, y of V we define $[x,y] = xy - yx$. Now we let $a \in V$ and derive the following properties of E_a:

P1. $E_a \in \mathfrak{sp}(V)$. Indeed, for $x, y \in V$:

$$\begin{aligned} g(E_a(x), y) + g(x, E_a(y)) &= g(g(a,x)a, y) + g(x, g(a,y)a) \\ &= g(a,x)g(a,y) + g(a,y)g(x,a) \\ &= 0. \end{aligned} \tag{9.36}$$

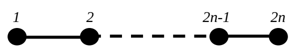


Figure 9.2: Extremal elements generating C_n

P2. $E_a^2 = 0$. Indeed, for $x \in V$:

$$\begin{aligned} E_a E_a(x) &= E_a g(a,x)a \\ &= g(a,x)g(a,a)a \\ &= 0. \end{aligned} \quad (9.37)$$

P3. $[E_a, [E_a, B]] = \alpha B$ where $\alpha \in \mathbb{F}$. Indeed, for $B \in \mathrm{End}(V)$ and $x \in V$, using P2:

$$\begin{aligned} [E_a, [E_a, B]](x) &= -2E_a B E_a(x) \\ &= -2E_a B g(a,x)a \\ &= -2g(a,x)E_a(Ba) \\ &= -2g(a,x)g(a,Ba)a \\ &= -2g(a,Ba)E_a(x). \end{aligned} \quad (9.38)$$

P4. $E_{a+b}(x) - E_a(x) - E_b(x) = g(a,x)b + g(b,x)a$. Indeed, let $b, x \in V$:

$$\begin{aligned} E_{a+b}(x) &= g(a+b,x)(a+b) \\ &= g(a,x)a + g(a,x)b + g(b,x)a + g(b,x)b \\ &= E_a(x) + E_b(x) + g(b,x)a + g(b,x)b. \end{aligned} \quad (9.39)$$

P5. $[E_a, E_b] = g(a,b)(E_{a+b} - E_a - E_b)$. Indeed, let $b, x \in V$, and use P4:

$$\begin{aligned} [E_a, E_b](x) &= E_a E_b(x) - E_b E_a(x) \\ &= g(b,x)g(a,b)a - g(a,x)g(b,a)b \\ &= g(a,b)(g(b,x)a + g(a,x)b) \\ &= g(a,b)(E_{a+b}(x) - E_a(x) - E_b(x)). \end{aligned} \quad (9.40)$$

From this observation it follows that $[E_a, E_b] \in \mathfrak{sp}(V)$ as well. Note that $[E_a, E_b]$ is in general not extremal.

We now give a lemma on the combination of various E_a.

Lemma 9.5. *Let $a_1, \ldots, a_n \in V$ and let $b_1, \ldots, b_k \in V$ be such that for every l we have $b_l = a_{l_1} + \ldots + a_{l_t}$ for some $t \in \mathbb{N}$. Moreover, suppose the following two equations hold:*

$$\mu_1 a_1 + \ldots + \mu_k a_n \Rightarrow \mu_1 = \ldots = \mu_n = 0, \quad (9.41)$$

$$\lambda_1 E_{a_1} + \ldots + \lambda_n E_{a_n} + \lambda_{n+1} E_{b_1} + \ldots + \lambda_{n+k} E_{b_k} = 0 \Rightarrow \lambda_1 = \ldots = \lambda_{n+k} = 0, \quad (9.42)$$

i.e. the a_i are linearly independent, and the E_{a_i} and E_{b_i} are all linearly independent.

Then, if we let $b \in V$ be such that $b \neq 0$, either $b = a_i + a_j$ or $b = a_i + b_j$, and $b \notin \{a_1, \ldots, a_n, b_1, \ldots, b_k\}$, we have

$$\lambda_1 E_{a_1} + \ldots + \lambda_{n+k} E_{b_k} + \lambda E_b = 0 \Rightarrow \lambda_1 = \ldots = \lambda_{n+k} = \lambda = 0, \quad (9.43)$$

i.e. E_b is linearly independent from all previous E_{a_i} and E_{b_j}.

Proof Without loss of generality, we assume $b = a_1 + \ldots + a_t$. Suppose to the contrary that our lemma is false, so there exist $\lambda_1, \ldots, \lambda_{n+k}, \lambda$ not all equal to zero, such that $\lambda_1 E_{a_1} + \ldots + \lambda_{n+k} E_{b_k} + \lambda E_b = 0$, or equivalently:

$$\lambda_1 E_{a_1}(x) + \ldots + \lambda_{n+k} E_{b_k}(x) + \lambda E_b(x) = 0 \text{ for all } x \in V. \quad (9.44)$$

If $\lambda = 0$ this is a contradiction with Equation 9.42, so $\lambda \neq 0$.

We rearrange the terms of the summation, and find, for all $x \in V$:

$$0 = a_1(\lambda_1 g(a_1, x) + g(T_{a_1}, x) + \lambda g(a_1 + \ldots + a_t, x))$$

$$\begin{aligned}
&\vdots \\
&+a_t(\lambda_t g(a_t,x) + g(T_{a_t},x) + \lambda g(a_1+\ldots+a_t,x)) \\
&+a_{t+1}g(T_{a_{t+1}},x) \\
&\vdots \\
&+a_n g(T_{a_n},x), \qquad (9.45)
\end{aligned}$$

where T_{a_l} is some combination of terms of the form $\lambda_j g(a_{l_1}+\ldots+a_{l_s},x)$ for some $s\in\mathbb{N}$, depending on the values of $b_1,\ldots,b_k$.

By Assumption 9.41, we now find

$$\begin{aligned}
0 &= \lambda_1 g(a_1,x) + g(T_{a_1},x) + \lambda g(a_1+\ldots+a_t,x), \text{ and} \\
0 &= \lambda_2 g(a_2,x) + g(T_{a_2},x) + \lambda g(a_1+\ldots+a_t,x), \text{ for all } x\in V.
\end{aligned}$$

Because $\lambda\neq 0$ and there exists an $x\in V$ such that $g(a_1+\ldots+a_t,x)\neq 0$, and because we assumed that $b\notin\{a_1,\ldots,a_n,b_1,\ldots,b_k\}$, we know that $\lambda_1\neq 0$, or at least one of the λ_i hidden in T_{a_1} is non-zero. Moreover, the above implies that $\lambda_1 g(a_1,x)+g(T_{a_1},x) = \lambda_2 g(a_2,x)+f(T_{a_2},x)$, or equivalently

$$g(\lambda_1 a_1 + T_{a_1} - \lambda_2 a_2 - T_{a_2},x) = 0 \text{ for all } x\in V. \qquad (9.46)$$

This shows $\lambda_1 a_1 + T_{a_1} - \lambda_2 a_2 - T_{a_2} = 0$, but not all λ_i are zero, contradicting Assumption 9.41. So $\lambda = 0$, and we proved Lemma 9.5 □

A direct consequence of this lemma is the following:

Lemma 9.6. *Let $a_1,\ldots,a_n\in V$ and let $b_1,\ldots,b_k\in V$ be such that for every l we have $b_l = a_{l_1}+\ldots+a_{l_t}$ for some $t\in\mathbb{N}$. Moreover, suppose the following two equations hold:*

$$\mu_1 a_1 + \ldots + \mu_n a_n \Rightarrow \mu_1 = \ldots = \mu_n = 0, \qquad (9.47)$$

$$\lambda_1 E_{a_1}+\ldots+\lambda_n E_{a_n}+\lambda_{n+1}E_{b_1}+\ldots+\lambda_{n+k}E_{b_k} = 0 \Rightarrow \lambda_1 = \ldots = \lambda_{n+k} = 0, \qquad (9.48)$$

i.e. the a_i are linearly independent, and the E_{a_i} and E_{b_i} are all linearly independent.

Then if we let $B = [E_{a_i},E_{b_j}]$, such that $g(a_i,b_j)\neq 0$ and $a_i+b_j\notin\{a_1,\ldots,a_n,b_1,\ldots,b_k\}$, we have

$$\lambda_1 E_{a_1}+\ldots+\lambda_{n+k}E_{b_k}+\lambda E_{a_i+b_j} = 0 \Rightarrow \lambda_1 = \ldots = \lambda_{n+k} = \lambda = 0, \text{ and} \qquad (9.49)$$

$$\lambda_1 E_{a_1}+\ldots+\lambda_{n+k}E_{b_k}+\lambda B = 0 \Rightarrow \lambda_1 = \ldots = \lambda_{n+k} = \lambda = 0. \qquad (9.50)$$

Proof We assume, without loss of generality, that $B = [E_{a_1},E_{b_1}]$. Lemma 9.5 immediately gives us

$$\lambda_1 E_{a_1}+\ldots+\lambda_{n+k}E_{b_k}+\lambda E_{a_1+b_1} = 0 \Rightarrow \lambda_1 = \ldots = \lambda_{n+k} = \lambda = 0, \qquad (9.51)$$

proving the first claim. We set $\gamma = g(a_1,b_1)$ and apply Property P5 to see that $B = \gamma(E_{a_1+b_1} - E_{a_1} - E_{b_1})$, which gives

$$\begin{aligned}
0 &= \lambda_1 E_{a_1}+\ldots+\lambda_{n+k}E_{b_k}+\lambda B \\
&= (\lambda_1-\lambda\gamma)E_{a_1}+\ldots+\lambda_n E_{a_n}+(\lambda_{n+1}-\lambda\gamma)E_{b_1}+\ldots+\lambda_{n+k}E_{b_k}+\lambda\gamma E_{a_1+b_1}.
\end{aligned}$$

Applying the first claim now shows that $\lambda_1 = \ldots = \lambda_{n+k} = \lambda = 0$, proving the second claim as well.

This completes the proof of Lemma 9.6 □

Now these two lemmas are enough to prove Theorem 9.4.

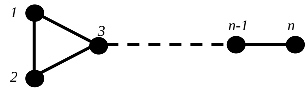


Figure 9.3: Extremal elements generating A_{n-1}

Proof of Theorem 9.4 Observe the basis elements of L, as proved in Theorem 9.3. We let V be a $2l$-dimensional vector space over $\mathbb{F}$, and let e_i denote the vector $v \in V$ with $v_i = 1$ and all other entries 0. We let $\psi : L \to V$ map the basis elements of L to elements of V as follows:

- $\psi(x_{2i-1}) = e_i$ for $1 \leq i \leq l$,
- $\psi(x_{2i}) = e_{i+l} + e_{i+l+1}$ if $1 \leq i < l$, and
- $\psi(x_{2l}) = e_{2l}$.

For example, if $l = 2$, then $x_1 \to (1, 0, 0, 0)$, $x_2 \to (0, 0, 1, 1)$, $x_3 \to (0, 1, 0, 0)$, and $x_4 \to (0, 0, 0, 1)$. It is immediately clear that the $\psi(x_i)$ are linearly independent, i.e. they fulfill the first assumption of Lemma 9.6.

We now define the map $\varphi : L \to \mathfrak{sp}(V)$:

$$\begin{array}{rcl} x_i & \mapsto & E_{\psi(x_i)} \\ [x_i, x_j] & \mapsto & [\varphi(x_i), \varphi(x_j)]. \end{array} \tag{9.52}$$

To prove that φ maps basis elements of L to basis elements of $\mathfrak{sp}(V)$, we first let $B = \{\varphi(x_i) \mid 1 \leq i \leq 2l\}$, then add the basis monomials of L of length 2 to B, then we add the basis monomials of length 3, etc. It is straightforward to see that the basis elements of L from Theorem 9.3 are such that they satisfy the demands on a_i and b_j in Lemma 9.6 during each step of this procedure. Indeed, for $[\varphi(x_i), \varphi(x_{i+1})]$ (a basis monomial of length 2) we have $g(\psi(x_i), \psi(x_{i+1})) \neq 0$, and $\psi(x_i) + \psi(x_{i+1})$ is different from all $\psi(x_i)$. The reasoning for longer basis monomials is exactly the same. By induction, using Lemma 9.6 for the induction step, we find that the above procedure gives a set of linearly independent elements of $\mathfrak{sp}(V)$. Moreover, the dimension of L is $\frac{1}{2}(2l)^2 + \frac{1}{2}(2l) = 2l^2 + l$, which is exactly the dimension of $\mathfrak{sp}(V)$.

Furthermore, using Property P3, we have

$$[E_a, [E_a, E_b]](v) = -2g(a, E_b a)E_a(v) = 2g(a, b)^2 E_a(v). \tag{9.53}$$

Since we chose $\psi(x_i)$ such that $g(\psi(x_i), \psi(x_j)) \neq 0$ if and only if $|i - j| = 1$, we have that $f(x_i, x_j) \neq 0$ translates to $[E_{\psi(x_i)}, [E_{\psi(x_i)}, E_{\psi(x_j)}]] \neq 0$, as required. This observation and the above reasoning show that φ faithfully maps basis elements of L to basis elements of $\mathfrak{sp}(V)$, showing that L and C_l are indeed isomorphic. This completes the proof of Theorem 9.4. $\square$

9.3 A_n Revisited

In this section we again consider a picture describing Lie algebras generated by n extremal elements, and prove that one of its instances is isomorphic to A_{n-1}. To do that, we use the rewrite rules introduced in Section 8.3.

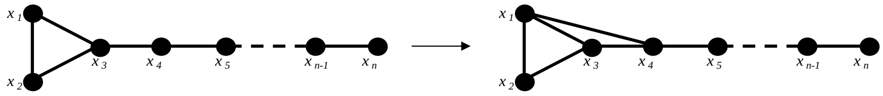


Figure 9.4: First rewrite move

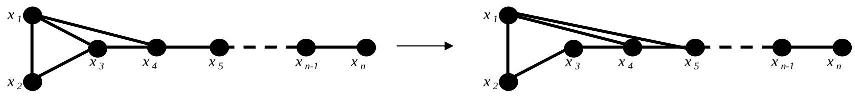


Figure 9.5: Middle rewrite moves

Theorem 9.7. *Let L be a Lie algebra over the field $\mathbb{F}$ generated by l (where $l \geq 4$) extremal elements $x_1, \ldots, x_l$, such that*

$$[x_1, x_2], [x_1, x_3], [x_2, x_3] \neq 0,$$

$$[x_i, x_{i+1}] \neq 0, \text{ where } i \in \{3, \ldots, l-1\},$$

and

$$[x_i, x_j] = 0 \text{ if } |i - j| \geq 2 \text{ and } \{i, j\} \not\subseteq \{1, 2, 3\},$$

see Figure 9.3. Then, for a certain evaluation of f, L is isomorphic to the simple Lie algebra A_{l-1}.

Proof We will prove this theorem by repeatedly applying the replacement of y by y^x, as introduced in Section 8.3. For the first move (see Figure 9.4), we set $x = x_3$, $y = x_1$, and $z = x_4$, and we are in Case C. Since $f(x_1, x_3) \neq 0$ and $f(x_3, x_4) \neq 0$ (as $\langle x_1, x_3\rangle \cong \mathfrak{sl}_2$ and $\langle x_3, x_4\rangle \cong \mathfrak{sl}_2$), we pick s non-zero and have an isomorphism between the original Lie algebra and the one depicted on the right hand side of figure 9.4. Indeed, now $\langle x_1^{x_3}, x_4\rangle \cong \mathfrak{sl}_2$.

Note that $\langle x_1, x_2\rangle \cong \mathfrak{sl}_2$ and $\langle x_2, x_3\rangle \cong \mathfrak{sl}_2$ (Case D above), so we should pick s such that $\langle x_1^{x_3}, x_2\rangle$ is still isomorphic to $\mathfrak{sl}_2$. This means that we should pick s such that $s \neq 0$ and $f(x_1, x_2) + sf(x_3, [x_1, x_2]) + \frac{1}{2}s^2 f(x_1, x_3) f(x_2, x_3) \neq 0$ (cf Case C and D above). This is not really a problem, since $f(x_1, x_2), f(x_1, x_3), f(x_2, x_3) \neq 0$.

We proceed with the right hand side of Figure 9.4, i.e. the left hand side of Figure 9.5. Now we first set $x = x_4$, $y = x_1$, and $z = x_3$, and see that we are in Case D. We pick $s \neq 0$ such that $\alpha = 0$ (which is possible, as proved above), and see that $\langle x_1^{x_4}, x_3\rangle$ disappears into the radical of the Lie algebra. Moreover, setting $z = x_5$, we see that $\langle x_1^{x_4}, x_5\rangle \cong \mathfrak{sl}_2$ since we picked s such that $s \neq 0$, and $f(x_1, x_4) \neq 0$ (by the previous step) and $f(x_4, x_5) \neq 0$ (by definition).

We apply this step $l - 4$ times and end up in the left hand side of Figure 9.6. In this situation we pick $x = x_n$, $y = x_1$, and $z = x_{n-1}$, and we see that we are in Case

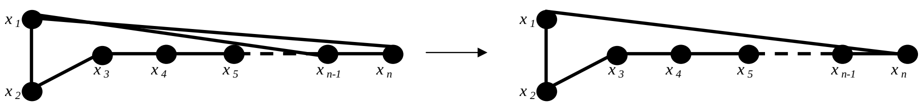


Figure 9.6: Last rewrite move

D again. We pick $s \neq 0$ such that $\alpha = 0$ (which is possible), and see that $\langle x_1^{x_n}, x_{n-1}\rangle$ disappears into the radical of the Lie algebra.

Now the right hand side of Figure 9.6 is isomorphic to A_{n-1} by Theorem 9.2, proving Theorem 9.7. □

9.4 Three Conjectures

In this Section we present three theorems we believe to be true, but could not prove. The first one is a refinement of Theorem 9.7, and the other two are theorems similar to 9.2 and 9.4.

Conjecture 9.8. *Let L be a Lie algebra over the field $\mathbb{F}$ generated by l (where $l \geq 4$) extremal elements $x_1, \ldots, x_l$, such that*

$$\langle x_1, x_2\rangle, \langle x_1, x_3\rangle, \langle x_2, x_3\rangle \cong \mathfrak{sl}_2,$$

$$\langle x_i, x_{i+1}\rangle \cong \mathfrak{sl}_2, \text{ where } i \in \{3, \ldots, l-1\},$$

and

$$[x_i, x_j] = 0 \text{ if } |i-j| \geq 2 \text{ and } \{i,j\} \not\subseteq \{1,2,3\},$$

see Figure 9.3. Then L is isomorphic to the simple Lie algebra A_{l-1}.

We verified this conjecture using Algorithm II for l up to 6, and using only the first step of Algorithm II (which means we only verified the dimension) for l up to 10.

Below we state two conjectures on the generation of B_n and D_n by extremal elements.

Conjecture 9.9. *Let L be a Lie algebra over the field $\mathbb{F}$ generated by l (where $l \geq 5$) extremal elements $x_1, \ldots, x_l$, such that*

$$\langle x_1, x_2\rangle, \langle x_1, x_3\rangle, \langle x_2, x_3\rangle, \cong \mathfrak{sl}_2,$$

$$\langle x_i, x_{i+1}\rangle \cong \mathfrak{sl}_2, \text{ where } i \in \{3, \ldots, l-3\},$$

$$\langle x_{l-2}, x_{l-1}\rangle, \langle x_{l-2}, x_l\rangle \cong \mathfrak{sl}_2,$$

and

$$[x_i, x_j] = 0 \text{ if } |i-j| \geq 2 \text{ and } \{i,j\} \not\subseteq \{1,2,3,l-2,l-1,l\},$$

$$[x_{l-1}, x_l] = 0,$$

see Figure 9.7. Then L is isomorphic to the simple Lie algebra B_{l-1}.

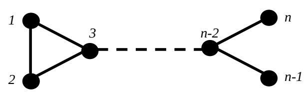


Figure 9.7: Extremal elements generating B_{n-1}

Conjecture 9.10. *Let L be a Lie algebra over the field $\mathbb{F}$ generated by l (where $l \geq 5$) extremal elements $x_1, \ldots, x_l$, such that*

$$\langle x_1, x_2\rangle, \langle x_1, x_3\rangle, \langle x_2, x_3\rangle, \cong \mathfrak{sl}_2,$$

$$\langle x_i, x_{i+1}\rangle \cong \mathfrak{sl}_2, \text{ where } i \in \{3, \ldots, l-3\},$$

$$\langle x_{l-2}, x_{l-1}\rangle, \langle x_{l-2}, x_l\rangle, \langle x_{l-1}, x_l\rangle \cong \mathfrak{sl}_2,$$

and

$$[x_i, x_j] = 0 \text{ if } |i-j| \geq 2 \text{ and } \{i, j\} \not\subseteq \{1, 2, 3, l-2, l-1, l\},$$

see Figure 9.8. Then L is isomorphic to the simple Lie algebra D_l.

We verified these two conjectures for l up to 6 using Algorithm II. Moreover, for l up to 10, we verified that the dimensions of the proposed isomorphic Lie algebras match.

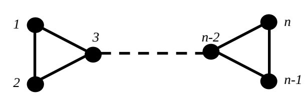


Figure 9.8: Extremal elements generating D_n

Chapter 10

Conclusion and Recommendations

In this Master's thesis we studied Lie algebras generated by extremal elements, mostly using and carrying on with the work by Cohen et al. [CSUW01]. The most important results include the observation that a Lie algebra generated by four elements is isomorphic to D_4 in general (Section 6.3), the detailed analysis of degenerate cases (Sections 6.5 and 8.2), and the observation that no semi-simple 537-dimensional Lie algebra generated by five extremal elements exists. The latter observation shows that the five generator case really differs from the smaller cases, and makes us suspect that the two, three, and four generator case are special in the fact that a semi-simple Lie algebra of maximal dimension does occur.

The results in Chapter 9 support the findings in [CSUW01] and show how two of the families of classical Lie algebras can be generated by extremal elements. Moreover, in Section 9.4 we give similar conjectures on the other two families of classical Lie algebras. As for the exceptional Lie algebras: G_2 was discovered in Section 6.3, and F_4 in Section 8.2. We suspect that E_6, E_7, and E_8 were found in Section 8.2 as well, but that remains to be proved.

Two things suitable for future research come to mind. First of all, it would be worth finding out if (and how) extremal elements can be applied in physics in general and, more particular, in differential geometry. This could be a nice project for someone acquainted with both physics and algebraic geometry.

Secondly, from a mathematical point of view, it would be very satisfying to find a proof for Conjectures 9.9 and 9.10 and the occurrence of E_6, E_7, and E_8 in Section 8.2. It would then be known how to generate every simple Lie algebra by extremal elements. However, I suspect some work has to be done before these conjectures are proved.

Of course, many more possibilities for future research exist. One could, for example, analyze Lie algebras generated by six extremal elements (whose dimension is suspected to be more than 20.000), or focus on Lie algebras over fields of non-zero characteristic. However, the series 1, 3, 8, 28, 537, ... remains the most intriguing problem to me. Maybe this is where the key to all there is to know about Lie algebras generated by extremal elements is hidden?

Bibliography

[Ben77] G. Benkart. Inner ideals of lie algebras. *Transations of the American Mathematical Society*, 232:61–81, 1977.

[BK90] G.G.A. Bäuerle and E.A. de Kerf. *Finite and Infinite Dimensional Lie Algebras and Applications in Physics*, volume 1 of *Studies in Mathematical Physics*. Elsevier Science Publishers, 1990.

[CG05] A.M. Cohen and D.A.H. Gijsbers. GBNP 0.9.2 (Non-commutative Gröbner bases), 2005. `http://www.win.tue.nl/~amc/pub/grobner/`.

[Che89] V. I. Chernousov. On the Hasse principle for groups of type e_8. *Dokl. Akad. Nauk SSSR*, 306, 25:1059–1063, 1989. Translation in Soviet Math. Dokl. 39, 592–596, 1989.

[CSUW01] A. M. Cohen, A. Steinbach, R. Ushirobira, and D. Wales. Lie algebras generated by extremal elements. *J. Algebra*, 236(1):122–154, 2001.

[Dra02] Jan Draisma. *Lie Algebras of Vector Fields*. PhD thesis, Technische Universiteit Eindhoven, 2002.

[Hal03] Brian C. Hall. *Lie Groups, Lie Algebras, and Representations, An Elementary Introduction*. Graduate Texts in Mathematics. Springer-Verlag New York, 2003.

[Hum72] J. E. Humphreys. *Introduction to Lie Algebras and Representation Theory*. Graduate Texts in Methematics. Springer-Verlag New York, 1972.

[Jac62] N. Jacobson. *Lie Algebras*, volume 10 of *Interscience Tracts in Pure and Applied Mathematics*. Interscience Publishers, 1962.

[Kno04] Jan Willem Knopper. GBNP and vector enumeration. 2004.

[Kos90] A.I. Kostrikin. *Around Burnside*. Springer-Verlag, 1990.

[MQ] Quotations about mathematics and education. `http://www.pen.k12.va.us/Div/Winchester/jhhs/math/quotes.html`.

[PS97] A. Premet and H Strade. Simple Lie algebras of small characteristic. I. Sandwich elements. *J. Algebra*, 189:419–480, 1997.

[S$^+$95] Martin Schönert et al. *GAP – Groups, Algorithms, and Programming*. Lehrstuhl D für Mathematik, Rheinisch Westfälische Technische Hochschule, Aachen, Germany, fifth edition, 1995.

[Ser87] J.P. Serre. *Complex Semisimple Lie Algebras*. Springer-Verlag New York, 1987.

[SW86] D.H. Sattinger and O.L. Weaver. *Lie Groups and Algebras With Applications to Physics, Geometry, and Mechanics*, volume 61 of *Applied Mathematical Sciences*. Springer-Verlag New York, 1986.

[Var84] V. S. Varadarajan. *Lie Groups, Lie Algebras, and Their Representations*, volume 102 of *Graduate Texts in Mathematics*. Springer-Verlag New York, 1984.

[ZK90] E. I. Zel′manov and A. I. Kostrikin. A theorem on sandwich algebras. *Trudy Mat. Inst. Steklov.*, 183:106–111, 225, 1990. Translated in Proc. Steklov Inst. Math. **1991**, no. 4, 121–126, Galois theory, rings, algebraic groups and their applications (Russian).

Index

Appendix A

Simple Lie Algebras

An overview of all simple Lie algebras of dimension at most 1000:

Lie Algebra	Dim	#
$A_1 = B_1 = C_1$	3	2
A_2	8	3
$B_2 = C_2$	10	4
G_2	14	4
A_3	15	4
B_3	21	4
C_3	21	6
A_4	24	5
D_4	28	4
A_5	35	6
B_4	36	5
C_4	36	8
D_5	45	5
A_6	48	7
F_4	52	5
B_5	55	6
C_5	55	10
A_7	63	8
D_6	66	6
B_6	78	7
C_6	78	12
E_6	78	5
A_8	80	9
D_7	91	7
A_9	99	10
B_7	105	8
C_7	105	14
A_{10}	120	11
D_8	120	8
E_7	133	5
B_8	136	9
C_8	136	16

Lie Algebra	Dim	#
A_{11}	143	12
D_9	153	9
A_{12}	168	13
B_9	171	10
C_9	171	18
D_{10}	190	10
A_{13}	195	14
B_{10}	210	11
C_{10}	210	20
A_{14}	224	15
D_{11}	231	11
E_8	248	5
B_{11}	253	12
C_{11}	253	22
A_{15}	255	16
D_{12}	276	12
A_{16}	288	17
B_{12}	300	13
C_{12}	300	24
A_{17}	323	18
D_{13}	325	13
B_{13}	351	14
C_{13}	351	26
A_{18}	360	19
D_{14}	378	14
A_{19}	399	20
B_{14}	406	15
C_{14}	406	28
D_{15}	435	15
A_{20}	440	21
B_{15}	465	16
C_{15}	465	30

Lie Algebra	Dim	#
A_{21}	483	22
D_{16}	496	16
A_{22}	528	23
B_{16}	528	17
C_{16}	528	32
D_{17}	561	17
A_{23}	575	24
B_{17}	595	18
C_{17}	595	34
A_{24}	624	25
D_{18}	630	18
B_{18}	666	19
C_{18}	666	36
A_{25}	675	26
D_{19}	703	19
A_{26}	728	27
B_{19}	741	20
C_{19}	741	38
D_{20}	780	20
A_{27}	783	28
B_{20}	820	21
C_{20}	820	40
A_{28}	840	29
D_{21}	861	21
A_{29}	899	30
B_{21}	903	22
C_{21}	903	42
D_{22}	946	22
A_{30}	960	31
B_{22}	990	23
C_{22}	990	44

'Dim' denotes the dimension of the Lie algebra (see Section 2.4), '#' denotes the minimum number of extremal elements required to generate this Lie algebra (see [CSUW01, Section 8]).

Appendix B

Multiplication Tables

	x	**y**	**z**
x	0	z	$f(x,y)x$
y	$-z$	0	$-f(y,x)y$
z	$-f(x,y)x$	$f(y,x)y$	0

Table B.1: Multiplication Table for the 2 generator case

	x	**y**	**z**
x	0	z	$f(x,y)x$
y	$-z$	0	$-f(x,y)y$
z	$-f(x,y)x$	$f(x,y)y$	0

Table B.2: Simplified Table for the 2 generator case

	x	**y**	**z**
x	0	z	0
y	$-z$	0	0
z	0	0	0

Table B.3: Simplified Table for the 2 generator case where $f = 0$

	$\mathbf{x}$	$\mathbf{y}$	$\mathbf{z}$	$[\mathbf{x},\mathbf{y}]$	$[\mathbf{x},\mathbf{z}]$	$[\mathbf{y},\mathbf{z}]$
$\mathbf{x}$	0	$[x,y]$	$[x,z]$	$f(x,y)x$	$f(x,z)x$	$[x,[y,z]]$
$\mathbf{y}$		0	$[y,z]$	$-f(x,y)y$	$[y,[x,z]]$	$f(y,z)y$
$\mathbf{z}$			0	$-[x,[y,z]]+[y,[x,z]]$	$-f(x,z)z$	$-f(y,z)z$
$[\mathbf{x},\mathbf{y}]$				0	$\frac{1}{2}f(x,[y,z])x+\frac{1}{2}f(x,z)[x,y]$ $-\frac{1}{2}f(x,y)[x,z]$	$\frac{1}{2}f(x,[y,z])y+\frac{1}{2}f(y,z)[x,y]$ $+\frac{1}{2}f(x,y)[y,z]$
$[\mathbf{x},\mathbf{z}]$					0	$+\frac{1}{2}f(x,[y,z])z-\frac{1}{2}f(y,z)[x,z]$ $+\frac{1}{2}f(x,z)[y,z]$
$[\mathbf{y},\mathbf{z}]$						0
$[\mathbf{x},[\mathbf{y},\mathbf{z}]]$						
$[\mathbf{y},[\mathbf{x},\mathbf{z}]]$						

	$[\mathbf{x},[\mathbf{y},\mathbf{z}]]$	$[\mathbf{y},[\mathbf{x},\mathbf{z}]]$
$\mathbf{x}$	$f(x,[y,z])x$	$\frac{1}{2}f(x,[y,z])x-\frac{1}{2}f(x,z)[x,y]$ $-\frac{1}{2}f(x,y)[x,z]$
$\mathbf{y}$	$-\frac{1}{2}f(x,[y,z])y+\frac{1}{2}f(y,z)[x,y]$ $-\frac{1}{2}f(x,y)[y,z]$	$-f(x,[y,z])y$
$\mathbf{z}$	$-\frac{1}{2}f(x,[y,z])z-\frac{1}{2}f(y,z)[x,z]$ $-\frac{1}{2}f(x,z)[y,z]$	$\frac{1}{2}f(x,[y,z])z-\frac{1}{2}f(x,z)[y,z]$ $-\frac{1}{2}f(y,z)[x,z]$
$[\mathbf{x},\mathbf{y}]$	$\frac{1}{2}f(y,z)f(x,y)x+\frac{1}{2}f(x,[y,z])[x,y]$ $-\frac{1}{2}f(x,y)[x,[y,z]]$	$-\frac{1}{2}f(x,z)f(x,y)y-\frac{1}{2}f(x,[y,z])[x,y]$ $+\frac{1}{2}f(x,y)[y,[x,z]]$
$[\mathbf{x},\mathbf{z}]$	$-\frac{1}{2}f(y,z)f(x,z)x+\frac{1}{2}f(x,[y,z])[x,z]$ $-\frac{1}{2}f(x,z)[x,[y,z]]$	$f(x,[y,z])[x,z]-\frac{1}{2}f(x,z)\cdot(f(x,y)z+f(y,z)x)$ $-\frac{1}{2}f(x,z)\cdot(2[x,[y,z]]-[y,[x,z]])$
$[\mathbf{y},\mathbf{z}]$	$-f(x,[y,z])[y,z]-\frac{1}{2}f(y,z)\cdot(f(x,y)z+f(x,z)y)$ $-\frac{1}{2}f(y,z)\cdot(2[y,[x,z]]-[x,[y,z]])$	$-\frac{1}{2}f(x,z)f(y,z)y-\frac{1}{2}f(x,[y,z])[y,z]$ $-\frac{1}{2}f(y,z)[y,[x,z]]$
$[\mathbf{x},[\mathbf{y},\mathbf{z}]]$	0	$-\frac{1}{4}(f(y,z)f(x,[y,z])x+f(x,[y,z])f(x,z)y)$ $-\frac{1}{4}f(x,[y,z])f(x,y)z-\frac{1}{2}f(y,z)f(x,z)[x,y]$ $+\frac{1}{2}(f(y,z)f(x,y)[x,z]-\frac{1}{2}f(x,z)f(x,y)[y,z])$
$[\mathbf{y},[\mathbf{x},\mathbf{z}]]$		0

Table B.4: Simplified Multiplication Table for the 3 generator case

Appendix C

GAP Code

C.1 Three Generator Case

```
fxy:=13;;fxz:=17;;fyz:=19;;fxyz:=23;;
T := EmptySCTable( 8, 0, "antisymmetric");
SetEntrySCTable( T, 1,2, [1,4] );
SetEntrySCTable( T, 1,3, [1,5] );
SetEntrySCTable( T, 1,4, [fxy,1] );
SetEntrySCTable( T, 1,5, [fxz,1] );
SetEntrySCTable( T, 1,6, [1,7] );
SetEntrySCTable( T, 1,7, [fxyz,1] );
SetEntrySCTable( T, 1,8, [(1/2)* fxyz,1, - (1/2)* fxz,4, - (1/2)* fxy,5] );
SetEntrySCTable( T, 2,3, [1,6] );
SetEntrySCTable( T, 2,4, [-fxy,2] );
SetEntrySCTable( T, 2,5, [1,8] );
SetEntrySCTable( T, 2,6, [fyz,2] );
SetEntrySCTable( T, 2,7, [- (1/2)* fxyz,2, (1/2)* fyz,4,  - (1/2)*fxy,6] );
SetEntrySCTable( T, 2,8, [-fxyz,2] );
SetEntrySCTable( T, 3,4, [-1,7,1,8] );
SetEntrySCTable( T, 3,5, [-fxz,3] );
SetEntrySCTable( T, 3,6, [-fyz,3] );
SetEntrySCTable( T, 3,7, [-(1/2)* fxyz,3, - (1/2)* fyz,5, - (1/2)* fxz,6] );
SetEntrySCTable( T, 3,8, [(1/2)* fxyz,3, - (1/2)* fxz,6, - (1/2)* fyz,5] );
SetEntrySCTable( T, 4,5, [(1/2)* fxyz,1, (1/2)* fxz,4, - (1/2)* fxy,5] );
SetEntrySCTable( T, 4,6, [ (1/2)* fxyz,2, (1/2)* fyz,4, (1/2)* fxy,6] );
SetEntrySCTable( T, 4,7, [ (1/2)* fyz* fxy,1, (1/2)* fxyz,4, - (1/2)* fxy,7] );
SetEntrySCTable( T, 4,8, [ - (1/2)* fxz* fxy,2, -(1/2)* fxyz,4, + (1/2)* fxy,8] );
SetEntrySCTable( T, 5,6, [(1/2)* fxyz,3, - (1/2)* fyz,5, + (1/2)* fxz,6] );
SetEntrySCTable( T, 5,7, [- (1/2)* fyz*fxz,1, (1/2)* fxyz,5, - (1/2)* fxz,7] );
SetEntrySCTable( T, 5,8, [fxyz,5,-(1/2)*fxz*fxy,3,-(1/2)*fxz*fyz,1,-(1/2)*fxz*2,
    7,-(1/2)*fxz*(-1),8] );
SetEntrySCTable( T, 6,7, [-fxyz,6,-(1/2)*fyz*fxy,3,-(1/2)*fyz*fxz,2,-(1/2)*fyz*2,
    8,-(1/2)*fyz*(-1),7] );
SetEntrySCTable( T, 6,8, [-(1/2)*fxz*fyz,2,-(1/2)*fxyz,6,-(1/2)*fyz,8] );
SetEntrySCTable( T, 7,8, [ -(1/4)*fyz*fxyz,1, -(1/4)*fxyz*fxz,2, -(1/4)*fxyz*fxy,3,
     - (1/2)* fyz* fxz,4, + (1/2)* ( fyz*fxy),5, - (1/2)*fxz*fxy,6] );
TestJacobi(T);
L := LieAlgebraByStructureConstants(Rationals, T);
SemiSimpleType(L);
```

C.2 Algorithm I

```
RequirePackage("gbnp");

##################################################################################
## Elt(i)
##
## Returns the monomial i in GBNP format
##
elt := function(i)
    return [[[i]],[1]];
end;

##################################################################################
## bprod(a,b)
##
## Returns ab-ba
##
bprod := function(a,b)
    return AddNP(MulNP(a,b),MulNP(b,a),1,-1);
end;

##################################################################################
## rbprod(a,b, basis)
##
## Returns ab-ba, cleaned and reduced wrt to the basis.
##
rbprod := function(a,b,basis)
    return CleanNP(StrongNormalFormNP(bprod(a,b),basis));
end;

##################################################################################
## DivNP(pol, B)
##
## Returns [[Q], r], where:
##   * Q is a list of size Size(lst)
##   * sum Q[i] lst[i] + r = a
##
DivNP := function(a, lst)
    local b,pos,r,Q, ltms;

    r := a; Q := 0*[1..Size(lst)]; ltms := LTermsNP(lst);

    if (not(r = [[],[]])) then pos := Position(ltms, r[1][1]); fi;
    while (not((r = [[],[]]) or (pos = fail))) do
        Q[pos] := Q[pos] + r[2][1];
        r := CleanNP(AddNP(r, lst[pos], 1, -r[2][1]));

        if (not(r = [[],[]])) then pos := Position(ltms, r[1][1]); fi;
    od;

    return [Q, r];
end;

##################################################################################
## LstToSC(l)
##
## Transforms [a,b,0,d] into [a,1,b,2,d,4] i.e. list to structure constants
```

```
##
LstToSC := function(l)
    local a,i;
    a := [];

    for i in [1..Size(l)] do
        if (l[i] <> 0) then
            Add(a, l[i]);
            Add(a, i);
        fi;
    od;
    return a;
end;



####################################################################################
## pol_ee(a, b, fab)
## addext2(KI, a, b, fab)
## addext3(KI, a, b, c, fabc)
##
## Functions to create polynomials for input to the groebner basis algorithm
##
pol_ee := function(a, b, fab)
    return CleanNP(AddNP(bprod(a, bprod(a, b)), a, 1, -fab));
end;
addext2 := function(KI, a, b, fab);
    Add(KI, pol_ee(a, b, fab));
    Add(KI, pol_ee(b, a, fab));
end;
addext3 := function(KI, a, b, c, fabc);
    Add(KI, pol_ee(a, bprod(b, c), fabc));
    Add(KI, pol_ee(b, bprod(a, c), -fabc));
    Add(KI, pol_ee(c, bprod(a, b), fabc));
end;

####################################################################################
## BaseKI(N, commutators)
##
## Provides a standard set of polynomials to input into the groebner basis algorithm
##
BaseKI := function(N, commutators)
    local i,r;

    r := [];
    for i in [1..N] do
        Add(r, CleanNP([[[i,i]],[1]]));
    od;

    for i in commutators do
        Add(r, bprod(elt(i[1]), elt(i[2])));
    od;

    return r;
end;
```

```
################################################################################
## FillBasis(basis, tmpsc, groebnerbasis, tracedbasis,
##                     arg1from, arg1to, arg2from, arg2to)
##
## Consider the Lie brackets of basis elements arg1from..arg1to with basis
##  elements ##  arg2from..arg2to, and see if the products are all in the basis.
##  If not, add them, meanwhile keeping track of things in tmpsc (the structure
##  constants) and tracedbasis (basis monomials).
##
FillBasis := function(basis, tmpsc, groebnerbasis, tracedbasis, arg1from, arg1to,
                arg2from, arg2to)
    local added, div, i, j, newStartAt, startAt, p, innerLoopFrom, tsize, newsize;

    added := 0;
    for i in [arg1from..arg1to] do
        innerLoopFrom := Maximum(arg2from, i+1);
        for j in [innerLoopFrom..arg2to] do
            p := rbprod(basis[i],basis[j],groebnerbasis);
            div := DivNP(p, basis);
            if (not(div[2] = [[],[]])) then
                Add(basis, MkMonicNP(p));
                Add(tracedbasis, [i, j, p[2][1]]);

                added := added + 1;
                Add(tmpsc, [i, j, [p[2][1], Size(basis)]]);
            else
                Add(tmpsc, [i, j, LstToSC(div[1])]);
            fi;
        od;
    od;

    if (added > 0) then
        newsize := Size(basis);
        FillBasis(basis, tmpsc, groebnerbasis, tracedbasis,
            1, arg2to, arg2to + 1, newsize);
        FillBasis(basis, tmpsc, groebnerbasis, tracedbasis,
            arg2to + 1, newsize, arg2to + 1, newsize);
    fi;
end;

################################################################################
## TryToCleanBasis(in_basis, groebnerbasis, in_tmpsc)
##
## Do trivial things to remove redundant basis elements.
##
## Note: TracedBasis data is useless once this procedure is used on a basis
##
TryToCleanBasis := function(in_basis, groebnerbasis, in_tmpsc)
    local basis, coeff, belt, c1, c2, prod, tmpsc, i, j, t, u, div, p, newtmpsc,
        prodlst, basistmp, changed;
    basis := in_basis;
    tmpsc := in_tmpsc;
    changed := true;

    while (changed) do
        changed := false;
        for i in [1..(Length(basis))] do
            basistmp := StructuralCopy(basis{Union([1..(i-1)],
```

```
                    [(i+1)..Length(basis)])});
                p := CleanNP(StrongNormalFormNP(basis[i], groebnerbasis));
                div := DivNP(p, basistmp);
                if (div[2] = [[],[]]) then
                    #We may remove this element, but we have to change
                    #the structure constraints to reflect that removal
                    changed := true;

                    newtmpsc := [];
                    for u in tmpsc do
                        c1 := u[1]; c2 := u[2]; prod := u[3];
                        if (c1 <> i and c2 <> i) then
                            if (c1 > i) then c1 := c1 - 1; fi;
                            if (c2 > i) then c2 := c2 - 1; fi;

                            prodlst := List([1..(Length(basis)-1)], i -> 0);
                            for t in [0..((Length(prod)/2)-1)] do
                                coeff := prod[2*t+1]; belt := prod[2*t+2];
                                if (belt < i) then
                                    prodlst[belt] := prodlst[belt] + coeff;
                                elif (belt = i) then
                                    if (Length(div[1]) <> Length(basis) - 1) then
                                        Error();
                                    fi;
                                    for j in [1..Length(div[1])] do
                                        if (div[1][j] <> 0) then
                                            prodlst[j] := prodlst[j] + coeff*div[1][j];
                                        fi;
                                    od;
                                elif (belt > i) then
                                    prodlst[belt-1] := prodlst[belt-1] + coeff;
                                fi;
                            od;

                            Add(newtmpsc, [c1, c2, LstToSC(prodlst)]);
                        fi;
                    od;
                    tmpsc := StructuralCopy(newtmpsc);

                    basis := basistmp;

                    break;
                fi;
            od;
    od;

    return rec(basis := basis, tmpsc := tmpsc);
end;

###################################################################################
## FindLieAlgebra(N, KI)
##
## The function calling all of the above in the correct order.
FindLieAlgebra := function(N, KI)
    local GB, t1, t0, t1a, t2, t3, t4, tmpsc, tracedbasis, i, r, B, T, u, b, L, S;

    #Create Groebner Basis
    t0 := Runtime();
```

```
    Print("Input KI size ", Length(KI), "\n");
    GB := SGrobner(KI);
    t1 := Runtime();
    Print("\nSize of GB is ", Size(GB), "\n");

    #Generate Lie Algebra Basis
    Print("\nGenerating basis...\n");
    B := [];
    for i in [1..N] do
        Add(B, [[[i]],[1]]);
    od;
    Apply(B, i -> MkMonicNP(CleanNP(i)));
    tmpsc := [];
    tracedbasis := List([1..Length(B)], i->[i]);
    FillBasis(B, tmpsc, GB, tracedbasis, 1, Size(B), 1, Size(B));
    Print("\nThe basis has size ", Size(B), ". Cleaning...\n");

    #Clean Lie Algebra basis
    t1a := Runtime();
    r := TryToCleanBasis(B, GB, tmpsc);
    B := r.basis;
    tmpsc := r.tmpsc;
    Print("\nCleaned. The basis has size ", Size(B), ".\n");

    #Construct Lie Algebra
    t2 := Runtime();
    Print("\nThe resulting Lie Algebra basis has size ", Size(B), "\n");
    Print("\nConstructing Table of Structure Constants......");
    T := EmptySCTable( Size(B), 0, "antisymmetric");
    for u in tmpsc do
        SetEntrySCTable(T, u[1], u[2], u[3]);
    od;
    Print(" Done. \n");

    #Output properties of Lie algebra
    b := TestJacobi(T);
    if (not(b = true)) then
        Error("T does not fulfill the Jacobi identity.");
    fi;

    Print("Constructing Lie Algebra by Structure Constants...");
    L := LieAlgebraByStructureConstants(Rationals, T);
    t3 := Runtime();
    Print("Done\n");
    Print("  ", L, "\n");
    S := LieSolvableRadical(L);
    Print("  Lie Algebra of dim: ", Dimension(L), "\n");
    Print("  Radical of dim    : ", Dimension(S), "\n");
    Print("  Simple            : ", IsSimpleAlgebra(L/S), "\n");
    Print("  Semi Simple Type  : ", SemiSimpleType(L/S), "\n");
    t4 := Runtime();

    Print("Time used: \n");
    Print("  Groebner Basis    : ", (t1 - t0), "\n");
    Print("  Lie Algebra Basis : ", (t1a - t1), "\n");
    Print("  Clean LA Basis:   : ", (t2 - t1a), "\n");
    Print("  Lie Algebra Constr: ", (t3 - t2), "\n");
    Print("  SemiSimpleType    : ", (t4 - t3), "\n");
```

```
    Print("  TOTAL              : ", (t4 - t0), "\n");

    return rec(L := L, B := B, S := S);
end;
```

C.3 Algorithm II - Step 1

```
RequirePackage("gbnp");

##############################################################################
## Elt(i)
##
## Returns the monomial i in GBNP format
##
elt := function(i)
    return [[[i]],[1]];
end;

##############################################################################
## bprod(a,b)
##
## Returns ab-ba
##
bprod := function(a,b)
    return AddNP(MulNP(a,b),MulNP(b,a),1,-1);
end;

##############################################################################
## DivNP(pol, B)
##
## Returns [[Q], r], where:
##   * Q is a list of size Size(lst)
##   * sum Q[i] lst[i] + r = a
##
DivNP := function(a, lst)
    local b,pos,r,Q, ltms;

    r := a; Q := 0*[1..Size(lst)]; ltms := LTermsNP(lst);

    if (not(r = [[],[]])) then pos := Position(ltms, r[1][1]); fi;
    while (not((r = [[],[]]) or (pos = fail))) do
        Q[pos] := Q[pos] + r[2][1];
        r := CleanNP(AddNP(r, lst[pos], 1, -r[2][1]));

        if (not(r = [[],[]])) then pos := Position(ltms, r[1][1]); fi;
    od;

    return [Q, r];
end;

##############################################################################
## LstToSC(l)
##
## Transforms [a,b,0,d] into [a,1,b,2,d,4] i.e. list to structure constants
##
LstToSC := function(l)
    local a,i;
    a := [];
```

```
    for i in [1..Size(l)] do
        if (l[i] <> 0) then
            Add(a, l[i]);
            Add(a, i);
        fi;
    od;
    return a;
end;




################################################################################
## pol_ee(a, b, fab)
##
## Functions to create polynomials for input to the groebner basis algorithm
##
pol_ee := function(a, b)
    return CleanNP(bprod(a, bprod(a, b)));
end;

################################################################################
## AddEEPolsRec(GBIn, lst, paramlen_togo, pol)
## AddEEPols(GBIn, lst, paramlen)
##
## Functions to add polynomials describing the extremality of elements during
##  our calculations
##
AddEEPolsRec := function(GBIn, lst, paramlen_togo, pol)
    local a, p;
    if (paramlen_togo = 0) then
        for a in lst do
            p := CleanNP(bprod(elt(a), bprod(elt(a), pol)));
            if (p <> [[],[]]) then
                Add(GBIn, p);
            fi;
        od;
    elif (pol = []) then
        for a in lst do
            AddEEPolsRec(GBIn, lst, paramlen_togo - 1, elt(a));
        od;
    elif (pol <> []) then
        for a in lst do
            AddEEPolsRec(GBIn, lst, paramlen_togo - 1, bprod(elt(a), pol));
        od;
    fi;
end;
AddEEPols := function(GBIn, lst, paramlen);
    AddEEPolsRec(GBIn, lst, paramlen, []);
end;

################################################################################
## CalculateGBFromGradedGB(GradedGB, IntermediateResult, Changed, N)
##
## Calculates a Groebner Basis from Graded polynomials; using intermediate results
##  if possible.
##
```

```
## Input is st GradedGB[d] only contains homogeneous polynomials of degree d.
##
CalculateGBFromGradedGB := function(GradedGB, IntermediateResult, Changed, N)
    local start,d,e,t,time0,time1,GB;

    start := Position(Changed, true);
    if (start = fail or start = 1) then
        start := 1;
        GB := [];
    else
        GB := StructuralCopy(IntermediateResult[start - 1]);
    fi;
    time0 := Runtime();
    Print("LOG: Starting groebner basis calculation at degree ", start, "\n");

    for d in [start..Length(GradedGB)] do
        if (Length(GradedGB[d]) > 0) then
            for e in GradedGB[d] do
                Add(GB, e);
            od;
            GB := SGrobnerTrunc(GB, d, List([1..N], i->1), 1);
        fi;
        IntermediateResult[d] := StructuralCopy(GB);
    od;

    time1 := Runtime();
    Print("LOG: Groebner basis calculation finished. Runtime: ", time1-time0);
    Print(" msces.\n");

    return GB;
end;

###################################################################################
## CleanTracedBElt(elt)
##
## Clean up an element of the traced basis. Example: If m = [x,y] + y is in the
##  basis, and we find a 'new' element, namely [x,m], then the traced basis will
##  contain [x,[x,y]] + [x,y]. This function fixes this, and returns [x,y], since
##  [x,[x,y]] = 0.
##
CleanTracedBElt := function(elt)
    local m, r, eltin, pos;

    eltin := StructuralCopy(elt);
    r := [];

    while (Length(eltin) > 0) do
        m := eltin[1];
        eltin := eltin{[2..(Length(eltin))]};

        pos := PositionProperty(eltin, i->(i[2] = m[2]));
        while (pos <> fail) do
            m[1] := m[1] + eltin[pos][1];
            eltin := eltin{Union([1..(pos - 1)], [(pos + 1)..(Length(eltin))])};
            pos := PositionProperty(eltin, i->(i[2] = m[2]));
        od;

        if (m[1] <> 0) then
```

```
            Add(r, m);
        fi;
    od;

    return r;

end;

################################################################################
## CleanBasis(B, GB, TracedB)
##
## Returns: List of elements that can be removed from the basis, updating TracedB
##  in the process.
##
CleanBasis := function(B, GB, TracedB)
    local a,b,c,lc,i,k,origi,j,max,t,p,div,quotient, remainder, cntok,cntrem, B2,
        BTemp, TracedB2, basischanged;

    basischanged := true;
    B2 := StructuralCopy(B);
    TracedB2 := StructuralCopy(TracedB);

    Print("LOG: Cleaning basis, input elements: ", Length(B2), "\n");

    while (basischanged) do
        cntok := 0; cntrem := 0;
        i := 1;
        basischanged := false;

        while (i <= Length(B2) and (not(basischanged))) do
            p := CleanNP(StrongNormalFormNP(B2[i], GB));
            BTemp := StructuralCopy(B2{Union([1..(i-1)], [(i+1)..Length(B2)])});
            div := DivNP(p, BTemp);
            if ((div[2] = [[],[]])) then
                B2 := BTemp;
                TracedB2 := StructuralCopy(TracedB2{Union([1..(i-1)],
                    [(i+1)..Length(TracedB2)])});
                Print("LOG: Removing element ", i, " from basis.\n");
            elif (div[1] = List([1..Length(div[1])], i->0)) then
                i := i + 1;
            else
                basischanged := true;
                remainder := div[2];
                quotient := div[1]{[1..(i-1)]}; Add(quotient, 0);
                Append(quotient, div[1]{[i..(Length(div[1]))]});

                B2[i] := StructuralCopy(CleanNP(MkMonicNP(remainder)));

                lc := remainder[2][1];

                for j in [1..Length(quotient)] do
                    if (quotient[j] <> 0) then
                        for k in TracedB2[j] do
                            Add(TracedB2[i],StructuralCopy(
                                [-1*quotient[j]*k[1], k[2] ])
                            );
                        od;
                    fi;
```

```
                od;

                for k in TracedB2[i] do
                    k[1] := k[1] / lc;
                od;

            fi;
        od;

    od;

    Print("LOG: Basis cleaned, output elements: ", Length(B2), "\n");

    return rec(B := B2, TracedB := TracedB2);
end;


##########################################################################################
## findeebasis(N, commutators)
##
## The main function, using the above. Returns record containing:
##  B       : Basis elements (in Universal Enveloping Algebra)
##  BPos    : Pointers to where basis elements of specified degrees start
##  GB      : Groebner basis describing extremality
##  N       : Number of extremal generators
##  TracedB : Traced basis elements
##
findeebasis := function(N, commutators)
    local added,a1, a2, b1, b2, c, i, j, k, K, div,m,p, p0, r, BPos, B, NewB,
        NewBCleaned, GB, GradedGB, GBChanged, GBIntermediate, tpol, maxlen,
        len, obsolete, time0, time1, TracedB, NewTracedB, NewTracedBCleaned;

    time0 := Runtime();

    #Initialization
    B := List([1..N], i->elt(i));
    TracedB := List([1..N], i->([[1, i]]));
    #Polynomials of degree i can be found in B[BPos[i]] till B[BPos[i+1]]
    BPos := [1, N+1];

    GradedGB := [];
    GradedGB[2] := List([1..N], i->[[[i, i]],[1]]);
    GradedGB[3] := []; AddEEPols(GradedGB[3], [1..N], 1);

    for c in commutators do
        Add(GradedGB[2], bprod(elt(c[1]), elt(c[2])));
    od;

    GBIntermediate := List([1..Length(GradedGB)], i->[]);
    GBChanged := List([1..Length(GradedGB)], i->true);

    len := 1;
    maxlen := 25;

    added := 1;
    while ((len < maxlen) and (added > 0)) do
        #Update groebner basis
        len := len + 1;
```

```
    for i in [1..(len + 2)] do
        if (not(IsBound(GradedGB[i]))) then
            GradedGB[i] := [];
            GBIntermediate[i] := [];
        fi;
    od;
    GB := CalculateGBFromGradedGB(GradedGB, GBIntermediate, GBChanged, N);
    GBChanged := List([1..Length(GradedGB)], i->false);

    #Find new monomials
    added := 0;
    a1 := BPos[1]; a2 := BPos[2] - 1;
    b1 := BPos[len - 1]; b2 := BPos[len] - 1;
    NewB := [];
    NewTracedB := [];
    Print("LOG: Generating elements of length ", len, "\n");
    for i in [a1..a2] do
        for j in [b1..b2] do
            p0 := CleanNP(bprod(B[i], B[j]));
            p := CleanNP(StrongNormalFormNP(p0, GB));
            div := DivNP(p, B);

            if (not(div[2] = [[],[]])) then
                Add(NewB, CleanNP(MkMonicNP(p)));
                m := [];
                for k in TracedB[j] do
                    Add(m,
                        [ (1/(p[2][1])) * k[1], [i, StructuralCopy(k[2]) ] ]
                    );
                od;
                Add(NewTracedB, m);
                added := added + 1;
            fi;
        od;
    od;
    Print("LOG: Generated ", Length(NewB), " elements of length ", len, "\n");

    # Keep track of what changed
    if (added > 0) then
        r := CleanBasis(NewB, GB, NewTracedB);
        NewBCleaned := r.B;
        NewTracedBCleaned := r.TracedB;
        Print("LOG: --> Adding ", Length(NewBCleaned), " of ", Length(NewB));
        Print(" bracketings of length ", len, ", ");
        Print(Length(NewBCleaned) + Length(B), " so far.\n");

        for p in [1..Length(NewBCleaned)] do
            Add(B, NewBCleaned[p]);
            Add(TracedB, CleanTracedBElt(NewTracedBCleaned[p]));
            for k in [1..N] do
                tpol := CleanNP(bprod(elt(k), bprod(elt(k), NewBCleaned[p])));
                Add(GradedGB[len + 2], tpol);
                GBChanged[len + 2] := true;
            od;
        od;

        Add(BPos, Length(B) + 1);
```

```
        fi;

    od;

    time1 := Runtime();

    Print("LOG: Basis of ", Length(B), " elements found\n");
    Print("LOG: Total time taken: ", time1-time0, " msecs.\n");

    return rec(B := B, BPos := BPos, GB := GB, N := N, TracedB := TracedB );
end;
```