SEMISIMPLE SUBALGEBRAS OF SEMISIMPLE LIE ALGEBRAS

by

Mychelle Parker

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Mathematics

Approved:

Ian Anderson, Ph.D. Major Professor	Mark Fels, Ph.D. Committee Member
Zhaohu Nie, Ph.D. Committee Member	Richard S. Inouye, Ph.D. Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2019

ABSTRACT

Semisimple Subalgebras of Semisimple Lie Algebras

by

Mychelle Parker, Master of Science

Utah State University, 2019

Major Professor: Dr. Ian Anderson
Department: Mathematics and Statistics

Let $\mathfrak{g}$ be a Lie algebra. The subalgebra classification problem is to create a list of all subalgebras of $\mathfrak{g}$ up to equivalence. The purpose of this thesis is to provide a software toolkit within the Differential Geometry package of Maple for classifying subalgebras of In particular the thesis will focus on classifying those subalgebras which are isomorphic to the Lie algebra sl(2) and those subalgebras of which have a basis aligned with the root space decomposition (regular subalgebras).

(116 pages)

PUBLIC ABSTRACT

Semisimple Subalgebras of Semisimple Lie Algebras

Mychelle Parker

Let $\mathfrak{g}$ be a Lie algebra. The subalgebra classification problem is to create a list of all subalgebras of $\mathfrak{g}$ up to equivalence. The purpose of this thesis is to provide a software toolkit within the Differential Geometry package of Maple for classifying subalgebras of In particular the thesis will focus on classifying those subalgebras which are isomorphic to the Lie algebra sl(2) and those subalgebras of which have a basis aligned with the root space decomposition (regular subalgebras).

ACKNOWLEDGMENTS

I would like to thank Dr. Ian Anderson for guiding and supporting me on this project. He was always willing to help with any concerns that I had about school. I truly appreciate that he was willing to be my advisor.

I would also like to thank Dr. Mark Fels and Dr. Zhaohu Nie both for being members of my committee and for encouraging me to consider applying for graduate school.

Finally I would like to thank my family and friends for all they have done to support me as I worked on this project.

Mychelle Parker

Contents

1 Introduction

Let $\mathfrak{g}$ be a Lie algebra. Let $\mathcal{A} \in Aut(\mathfrak{g})$ be a subgroup of the automorphism group of $\mathfrak{g}$. Two subalgebras $\mathfrak{k}_1$ and $\mathfrak{k}_2$ are equivalent if there is an element $\varphi \in \mathcal{A}$ such that $\varphi(\mathfrak{k}_1) = \mathfrak{k}_2$. The subalgebra classification problem is creating a list of all subalgebras of $\mathfrak{g}$ up to this notion of equivalence. The purpose of this thesis is to provide a software toolkit for studying the subalgebras of simple Lie algebras. In particular software has been created to study those subalgebras which are isomorphic to $sl(2)$ and those subalgebras which are regular, that is the subalgebras with a basis aligned with the root space decomposition.

The problem of classifying subalgebras of a general Lie algebra is of fundamental importance in Lie theory and its applications in Differential Geometry to homogeneous spaces and studying group invariant solutions to differential equations, [22]. Already a complete classification of the subalgebras of $gl(5)$ is unknown so one sees that this is a difficult problem. Consequently, there is an extensive amount of literature on this problem which can be roughly split as belonging to one of two categories.

The first type deals with the subalgebra classification of specific Lie algebras. In their paper *Continuous Subgroups of the Fundamental Groups of Physics I.* Patera, Winternitz, and Zassenhaus classify the subalgebras of the Poincaré group. In their second paper *Continuous Subgroups of the Fundamental Groups of Physics II.* all subalgebras of the similitude algebra are classified up to equivalence. Patera and Winternitz also created a list of the subalgebras of all real Lie algebras of dimension $d \leq 4$. These results are summarized in tables in the paper *Subalgebras of real three- and four-dimensional Lie algebras.* The classification of the maximal solvable subalgebras of real Lie algebras $sl(n, \mathbb{R}), su^*(n), so^*(n), sp(n, \mathbb{R})$ and $usp(p, q)$ is given in Marcel Perroud's paper *The Maximal Solvable Subalgebras of the Real Classical Lie Algebras.* All subalgebras of $gl(3, \mathbb{R})$ have been classified by Thompson and Wick. An explicit classification of the semisimple complex Lie subalgebras of the simple complex Lie algebras up to rank 6 is given in Lorente and Gruber's paper *Classification of Semisimple Subalgebras of Simple Lie Algebras.* In the paper *The Subalgebras of G_2*, Evgeny Mayanskiy classifies all subalgebras of G_2 up to conjugacy.

The second type of literature are those papers which deal with classifying the subalgebras of semisimple Lie algebras. The classification of these subalgebras was extensively studied by E.B. Dynkin in his papers *Semisimple Subalgebras of Semisimple Lie Algebras* and *Maximal Subgroups of the Classical Groups.* Together Dynkin's two papers are over 280 pages long.

One topic that is discussed in Dynkin's papers are those subalgebras isomorphic to $sl(2)$ which are called $sl(2)$-triples. Kostant provides an algorithm which allows us to explicitly find an $sl(2)$-triple containing a given nilpotent element. For certain semisimple elements of a Lie algebra it is also possible to explicitly find the $sl(2)$-triple containing that element. Of particular importance are those $sl(2)$-triples which are principal. One application of the principal $sl(2)$-triple is that it can be used to find the exponents of the Lie algebra. Dynkin provides a classification of all $sl(2)$-triples of the exceptional Lie algebras up to conjugation.

Another topic discussed in Dynkin's papers are those subalgebras which are regular that is those subalgebras which can be written in the form $\mathfrak{f} = \mathfrak{h} \oplus_{\alpha\in\Gamma} \mathfrak{g}_\alpha$ where Γ is subset of the root system Δ of $\mathfrak{g}$, $\mathfrak{h}$ is a subset of the Cartan subalgebra of $\mathfrak{g}$, and $\mathfrak{g}_\alpha$ is the root space of the root α. Classifying all regular subalgebras of $\mathfrak{g}$ up to equivalence is the same as classifying all root subsystems up to conjugacy by the Weyl group. As a result of this relationship the Weyl group plays a central role in the classification of regular subalgebras.

If a subalgebras is not contained within a proper regular subalgebra it is defined to be an S-subalgebra. Dynkin also addresses the topic of classifying S-subalgebras. He also discussed those subalgebras which are not regular but are contained in a regular subalgebra which are defined to be R-subalgebras. If this project were to continue the next topic to discuss would be classifying the S-subalgebras and R-subalgebras.

While there are many software packages available for specialized computations in Lie theory, we believe that the toolkit provided in this thesis is unique to the Maple environment. Its inclusion in the DifferentialGeometry package will make this work widely accessible and support applications to other areas of differential geometry and its applications.

The thesis is organized as follows:

Chapter 2 will summarize the basic theory of Lie algebras and subalgebras. This section will include a discussion of both the Dynkin diagrams and the extended Dynkin diagrams of a simple Lie algebra. Maple already has a command that will return the Dynkin diagram of a simple Lie algebra and for this project a program was created to output the extended Dynkin diagram of a simple Lie algebra. The DG package in Maple already has commands for some properties of the Weyl group. One property of the Weyl group discussed in this chapter is that given a Weyl group element it is possible to find a corresponding Lie algebra automorphism that fixes the Cartan subalgebra. Given a Weyl group element the command WeylGroupElementToAutomorphism will return this mapping. Conversely the command AutomorphismToWeylGroupElement takes a Lie algebra automorhpism which preserves the Cartan subalgebra and returns the desired Weyl group element. Given two sets of roots the program EquivalenceOfSubsystems will return a Weyl group element which maps one set to the other.

Chapter 3 will continue to address the basic theory of Lie algebras, in particular this section will summarize the representation theory of $sl(2)$.

In Chapter 4, the notion of equivalence will be discussed. Two subalgebras $\mathfrak{f}_1$ and $\mathfrak{f}_2$ are equivalent if there is an element $\varphi \in \mathcal{A}$ such that $\varphi(\mathfrak{f}_1) = \mathfrak{f}_2$. One way to check if two simple subalgebras of a simple Lie algebra are not equivalent is to see if they have different Dynkin indices. The program DynkinIndex will return the Dynkin index of a given simple subalgebra.

In Chapter 5 we will discuss those subalgebras which are isomorphic to $sl(2)$, which we call $sl(2)$-triples. When finding these subalgebras one can use the Jacobson-Morosov Theorem which states that "Every nilpotent element of a complex semisimple Lie algebra can be embedded in an $sl(2)$-triple." Certain semisimple elements can also be contained within an $sl(2)$-triple. The goal of this section is to be able to list all of the $sl(2)$- subalgebras of a semisimple subalgebra up to conjugacy. To do this consider linear combinations (with coefficients 0, 1, or 2) of certain semisimple elements of the Lie algebra called the epsilon characteristics. For this chapter various programs have been written. The command ThreeDimensionalSubalgebra will return the $sl(2)$-triple containing a given nilpotent or semisimple element. The program EpsilonCharacteristics returns certain semisimple elements that can be used to list all $sl(2)$-triples. The command ALLThreeDimensionalSubalgebras uses the epsilon characteristics of a Lie algebra to explicitly list all non-conjugate $sl(2)$-triples. Using these programs we can verify Table 16 and Table 17 by Dynkin [11].

Let ϵ_i be the epsilon characteristics of a semisimple Lie algebra. Then the $sl(2)$ subalgebra containing the semisimple element $\sum 2\epsilon_i$ is defined to be the principal subalgebra. The program PrincipalSubalgebra will return this specific $sl(2)$-triple. Then in Chapter 6 the principal subalgebra can be used to find the Lie algebra exponents. The program LieAlegbraExponenets will return the desired exponents. However the results of this program are from a table rather than calculated each time.

In Chapter 7 the topic of regular subalgebras is addressed. A subalgebra $\mathfrak{f}$ of $\mathfrak{g}$ is regular with respect to Cartan subalgebra $\mathfrak{h}$ if it can be written in the form

$$\mathfrak{f} = \mathfrak{f}(\mathfrak{t}, \Gamma) = \mathfrak{t} \oplus \sum_{\alpha \in \Gamma} \mathfrak{g}_\alpha$$

where Γ is a closed root subsystem and $\mathfrak{t}$ is a certain subspace in $\mathfrak{h}$. Because regular subalgebras can be expressed this way there is a relationship between the root subsystems and the subalgebras. To study regular subalgebra the various programs have been created. The command VerifySubsystem will check to see if a given subsystem is closed, symmetric, or a π-system. The command VerifyRegular will check to see if a given subalgebra is a regular subalgebra. There are times where it is important to have a closed and symmetric subsystem so as a result the programs ClosedRootSystem and SymmetricRootSystem were created. ClosedRootSystem will find the smallest closed root subsystem containing the given subsystem. Similarly SymmetricRootSystem will find the smallest symmetric root subsystem containing the given subsystem. Because a regular subalgebra can be written in the form $\mathfrak{f} = \mathfrak{f}(\mathfrak{t}, \Gamma) = \mathfrak{t} \oplus \sum_{\alpha \in \Gamma} \mathfrak{g}_\alpha$ the program ClosedSubsystemToRegularSubalgebra takes a closed subsystem Γ and returns $\mathfrak{f}(\mathfrak{t}, \Gamma)$. Conversely the program RegularSubalgebraToRootSystem takes $\mathfrak{f}(\mathfrak{t}, \Gamma)$ and returns the closed subsystem Γ.

A subalgebra $\mathfrak{f}$ is reductive if rad $\mathfrak{f}$ consists only of semisimple elements. In Chapter 8 regular reductive subalgebras will be discussed. An important theorem for this section is that a subalgebra $\mathfrak{f}(\mathfrak{t}, \Gamma)$ is reductive if and only if the closed subsystem Γ is symmetric. For this section the program VerifyReductive was created to check if a given subalgebra is reductive.

In Chapters 9 we find the Maximal Regular Semisimple Subalgebras. To do this consider a set of simple roots Δ^0 and add the lowest root α_0 to create the extended root system $\tilde{\Delta}^0 = \Delta^0 \cup \{\alpha_0\}$. Then let $M = \tilde{\Delta}^0/\{\alpha_i\}$, where α_i is a certain root in the extended root system. Find $[M]$ which is all integral linear combinations of roots in M. Then the subalgebra $\mathfrak{f}(\mathfrak{t}, [M])$ will be a maximal regular semisimple subalgebra. Given a subsystem N the program SubsystemBracket will return $[N]$. Given a subsystem N the program MaximalPiSystem will find the largest π-system M such that $[M] \subset N$.

For the next Chapter let $M = \Delta_0/\{\alpha_i\}$ where α_i is a particular simple root. Then the subalgebra $\mathfrak{f}(\mathfrak{h}, [M])$ will be a maximal regular reductive (non-semisimple) subalgebra. We are able to list all the subalgebras of this type up to conjugacy. The results of Chapters 9 and 10 allowed us to explicitly verify the results of Tables 5 and 6 in [23]

In Chapter 11 we look at another type of regular subalgebra - those which are parabolic. There are already programs in Maple which deal with parabolic subalgebras. It is a theorem that every nonsemisimple maximal subalgebra of a complex semisimple Lie algebra is parabolic. As a result of this theorem we can conclude that the subalgebras discussed in Chapter 10 can be contained within a parabolic subalgebra.

Finally in Chapter 12 we will find all regular semisimple subalgebras of a given simple Lie algebra. To do this we take an extended root system and find a list of all root subsystems which are not related by any Weyl group element. (The program AllRootSystems will return this.) Then each of these subsystems will correspond to a non-conjugate regular semisimple subalgebra. This will verify the results of Table IV given by Lorente and Gruber [18].

We begin each chapter by summarizing the theory that will be used in that section as well as a list of commands in Maple that have been created for this project. After the summary each chapter will contain both examples of how the programs work and the theory behind them. Some of the given examples will illustrate results from various literature sources.

The appendix contains a summary of commands.

2 Basic Structure Theory of Lie Algebras

Let $\mathfrak{g}$ be a vector space with a bilinear map $\mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ called the Lie bracket, satisfying the following properties

- $[x, y] = -[y, x]$ for all $x, y \in \mathfrak{g}$
- $[x, [y, z]] + [z, [x, y]] + [y, [z, x]] = 0$ for all $x, y, z \in \mathfrak{g}$

then $\mathfrak{g}$ is said to be a **Lie algebra**. A vector subspace $\mathfrak{f}$ of $\mathfrak{g}$ is said to be a Lie **subalgebra** if $[x, y] \in \mathfrak{f}$ for all $x, y \in \mathfrak{f}$.

In this section we will discuss basic properties of Lie algebras, that will be relevant in later sections. The majority of the programs used in this section already existed with the exception of the following programs created to study extended Dynkin diagram and the Weyl group:

- **ExtendedDynkinDiagram:** This program returns the extended Dynkin diagram of a simple Lie algebra.
- **WeylGroupElementToAutomorphism:** This program takes an element of the Weyl group and returns a Lie algebra automorphism which fixes the Cartan subalgebra.
- **AutomorphismToWeylGroupElement:** This program takes a Lie algebra automorphism and returns the corresponding element of the Weyl group.
- **EquivalenceOfSubsystems:** Given two sets of roots Γ_1 and Γ_2, if there is a Weyl group element ω such that $\omega\Gamma_1 = \Gamma_2$ this program will find and return ω.

2.1 Basic Lie Algebra Properties

Let $\mathfrak{g}$ be a Lie algebra. A subset $\mathfrak{k}$ of $\mathfrak{g}$ is an **ideal** if it is a vector subspace of $\mathfrak{g}$ under addition and if for all $x \in \mathfrak{k}$ and $y \in \mathfrak{g}$ then $[x, y] \in \mathfrak{k}$. Note that both $\{0\}$ and $\mathfrak{g}$ will always be ideals. Any ideal of $\mathfrak{g}$ will be a subalgebra of $\mathfrak{g}$.

Let $\mathfrak{s} \subset \mathfrak{g}$ be a subalgebra. The **normalizer** of $\mathfrak{s}$ in $\mathfrak{g}$ is the largest subalgebra of $\mathfrak{g}$ which contains $\mathfrak{s}$ as an ideal.

$$\mathfrak{n}_{\mathfrak{s}}(\mathfrak{g}) = \{x \in \mathfrak{g} \mid [x, s] \in \mathfrak{s} \text{ for all } s \in \mathfrak{s}\}.$$

The normalizer of $\mathfrak{s}$ will contain $\mathfrak{s}$.

The **centralizer** of $\mathfrak{s}$ in $\mathfrak{g}$ is the set

$$C_{\mathfrak{s}}(\mathfrak{g}) = \{x \in \mathfrak{g} \mid [x, s] = 0 \text{ for all } s \in \mathfrak{s}\}.$$

The **Derived Series** of $\mathfrak{g}$ is the sequence of subalgebras defined by

$$\mathfrak{g}^{k+1} = [\mathfrak{g}^k, \mathfrak{g}^k]$$

where $\mathfrak{g}^0 = \mathfrak{g}$. A Lie algebra is **solvable** if $\mathfrak{g}^m = 0$ for some integer m.

We define the **Lower Central Series** of $\mathfrak{g}$ to be the sequence of subalgebras defined by

$$\mathfrak{g}_{k+1} = [\mathfrak{g}, \mathfrak{g}_k]$$

where $\mathfrak{g}_0 = \mathfrak{g}$. If this series ends in the zero subspace, then the $\mathfrak{g}$ is said to be **nilpotent**. Note that all nilpotent Lie algebras are also solvable.

The largest solvable ideal is said to be the **radical** of $\mathfrak{g}$ and is denoted $\operatorname{rad} \mathfrak{g}$.

Let $\mathfrak{g}_1$ and $\mathfrak{g}_2$ be Lie algebras. Then a linear mapping $\varphi : \mathfrak{g}_1 \to \mathfrak{g}_2$ is a **homomorphism** if for $x, y \in \mathfrak{g}_1$ we have

$$\varphi([x, y]) = [\varphi(x), \varphi(y)].$$

If φ is also bijective then φ is an **isomorphism.** If $\mathfrak{g}_1 = \mathfrak{g}_2$ and φ is an isomorphism, then φ is called an **automorphism**.

Let T be a linear transformation, then we define its exponential

$$\exp(T) = \sum_{k=1}^{\infty} \frac{1}{k!} T^k$$

where X^0 is the identity. Then $\exp(T)$ will also be a linear transformation.

For a fixed element $x \in \mathfrak{g}$, define the transformation $\mathrm{ad}_x : \mathfrak{g} \to \mathfrak{g}$ by $\mathrm{ad}_x(y) = [x, y]$ for all $y \in \mathfrak{g}$. We call the mapping $x \to ad_x$ the **adjoint representation** of $\mathfrak{g}$. For $x, y, z \in \mathfrak{g}$ we know that ad_x is a linear map and that

$$\mathrm{ad}_x[y, z] = [\mathrm{ad}_x(y), z] + [y, \mathrm{ad}_x(z)].$$

The Jacobi identity can be written as

$$([\mathrm{ad}_x, \mathrm{ad}_y])(z) = (\mathrm{ad}_{[x,y]})(z).$$

We define the linear transformation $\mathrm{Ad} : \mathfrak{g} \to \mathfrak{g}$ to be $\mathrm{Ad}(x) = \exp(\mathrm{ad}(x))$.

2.2 Killing Form

We now can define the **Killing form** on $\mathfrak{g}$ by

$$B(x, y) = \text{trace}(\text{ad}(x)\text{ad}(y))$$

for $x, y \in \mathfrak{g}$. Note that the Killing form is both bilinear and symmetric.

Let ϕ be a Lie algebra automorphism of $\mathfrak{g}$, then a property of the Killing form is

$$B(\phi(x), \phi(y)) = B(x, y).$$

Another property of the Killing form is that

$$B([x, y], z) = B(x, [y, z]).$$

The Killing form is said to be **non-degenerate** if $B(x, y) = 0$ for all $y \in \mathfrak{g}$ implies that $x = 0$.

2.3 Simple Lie Algebra

A Lie algebra $\mathfrak{g}$ is simple if it is non-abelian and its only ideals are the zero ideal and $\mathfrak{g}$. Not including the exceptional Lie algebras, any finite dimensional complex simple Lie algebra is isomorphic to either $sl(n)$, $so(n)$, or $sp(2n)$.

Those Lie algebras which are isomorphic to $sl(n)$, which are the trace free $n \times n$ invertible matrices, we classify as being a type A_{n-1}. Those Lie algebras which are isomorphic to the odd-dimensional orthogonal Lie algebra $so(2n+1)$ are type B_n. Those Lie algebras which are isomorphic to the symplectic Lie algebra $sp(2n)$ are type C_n. Those Lie algebras which are isomorphic to the even-dimensional orthogonal Lie algebra $so(2n)$ are type D_n.

There are also 5 exceptional Lie algebras: G_2, F_4, E_6, E_7, E_8 which are also simple Lie algebras, but not isomorphic to any of the Lie algebras listed above.

2.4 Semisimple Lie Algebra

A Lie algebra $\mathfrak{g}$ is defined to be **semisimple** if it is the direct sum of simple Lie algebras.

Cartan's Criterion says that $\mathfrak{g}$ is a complex **semisimple** Lie algebra if and only if its Killing form is non-degenerate.

The simple Lie algebras discussed above are also examples of semisimple Lie algebras.

2.5 Levi Decomposition

Let $\mathfrak{g}$ be a finite-dimensional Lie algebra. Then the **Levi decomposition** states that $\mathfrak{g}$ can be decomposed into the semi-direct sum $\mathfrak{g} = \mathfrak{r} + \mathfrak{s}$ where $\mathfrak{r}$ is the radical of $\mathfrak{g}$ and $\mathfrak{s}$ is a semisimple subalgebra of $\mathfrak{g}$. Therefore $\mathfrak{g}$ is a semi-direct product of a solvable Lie algebra and a semisimple Lie algebra.

2.6 Cartan Subalgebra

A subalgebra $\mathfrak{h}$ of $\mathfrak{g}$ is called a **Cartan subalgebra** if it is nilpotent and is equal to its normalizer. If $\mathfrak{g}$ is a complex semisimple Lie algebra then $\mathfrak{h}$ is a Cartan subalgebra if $\mathfrak{h}$ is a maximal commutative subalgebra of $\mathfrak{g}$ and ad_h is diagonalizable for every $h \in \mathfrak{h}$.

Every semisimple Lie algebra contains a Cartan subalgebra. A Cartan subalgebra of a complex semisimple Lie algebra $\mathfrak{g}$ is unique up to an automorphism of $\mathfrak{g}$. The dimension of the Cartan subalgebra is the **rank** of $\mathfrak{g}$.

2.7 Root Space Decomposition

Let $\mathfrak{g}$ be a semisimple Lie algebra with Cartan subalgebra $\mathfrak{h}$. Let $\mathfrak{h}^*$ be the set of linear functionals on $\mathfrak{h}$. Define $\alpha \in \mathfrak{h}^*$ to be a **root** of $\mathfrak{g}$ if $\alpha \neq 0$ and for some $x \in \mathfrak{g}$

$$[h, x] = \alpha(h) \cdot x$$

for all $h \in \mathfrak{h}$. The set of $x \in \mathfrak{g}$ that satisfy the above equation is called the **root space** of $\mathfrak{g}$ associated with α, which is denoted $\mathfrak{g}_\alpha$.

Let Δ be the system of all roots of $\mathfrak{g}$ with respect to $\mathfrak{h}$, then the **Root Space Decomposition** of $\mathfrak{g}$ with respect to $\mathfrak{h}$ is given by the direct sum

$$\mathfrak{g} = \mathfrak{h} \oplus \bigoplus_{\alpha \in \Delta} \mathfrak{g}_\alpha.$$

Some elementary properties of the root space decomposition are:

- Every root space $\mathfrak{g}_\alpha$ is one-dimensional
- If $\alpha \in \Delta$ then $-\alpha \in \Delta$
- If $\alpha \in \Delta$ and $k \neq \pm 1$, then $k\alpha \notin \Delta$ for $k \in \mathbb{R}$

- If $x \in \mathfrak{g}_\alpha$ and $y \in \mathfrak{g}_\beta$ then $[x, y] \in \mathfrak{g}_{\alpha+\beta}$ if $\alpha + \beta$ is also a root. Otherwise $[x, y] = 0$.
- If $x \in \mathfrak{g}_\alpha$ and $y \in \mathfrak{g}_{-\alpha}$ then $[x, y] \in \mathfrak{h}$. The three elements $\{x, y, [x, y]\}$ form a subalgebra which is isomorphic to $sl(2)$.
- If $\alpha \neq \beta$, then $B(\mathfrak{g}_\alpha, \mathfrak{g}_\beta) = 0$.

We can write Δ as the disjoint union $\Delta^+ \cup \Delta^-$, where if $\alpha \in \Delta^+$ then $-\alpha \in \Delta^-$. We say that Δ^+ are the **positive roots** and Δ^- are the **negative roots**. For a given root system Δ there are multiple choices for the positive and negative roots.

We say that a root $\alpha \in \Delta^+$ is a **simple root**, if α cannot be written as the sum of two other positive roots. We denote a choice of simple roots by Δ^0. The number of simple roots will be the same as the rank of the Lie algebra. For a given set of positive roots, there are multiple choices of simple roots.

Example 2.7.1: Roots of a Lie Algebra

Consider the Lie algebra $so(2, 2)$. This is a 6 dimensional Lie algebra with basis elements e_1, e_2, e_3, e_4, e_5 and e_6, with multiplication table

	e_1	e_2	e_3	e_4	e_5	e_6
e_1	0	e_2	$-e_3$	0	e_5	$-e_6$
e_2	$-e_2$	0	$e_1 - e_4$	e_2	0	0
e_3	e_3	$-e_1 + e_4$	0	$-e_3$	0	0
e_4	0	$-e_2$	e_3	0	e_5	$-e_6$
e_5	$-e_5$	0	0	$-e_5$	0	$-e_1 - e_4$
e_6	e_6	0	0	e_6	$e_1 + e_4$	0

Then a Cartan subalgebra for this Lie algebra is $\mathfrak{h} = \{e_1, e_4\}$. If $\mathfrak{h}^*$ has a basis $\{\alpha_1, \alpha_4\}$, then the roots of $\mathfrak{g}$ with respect to $\mathfrak{h}$ will be $\Delta = \{\alpha_1 - \alpha_4, \alpha_1 + \alpha_4, -\alpha_1 + \alpha_4, -\alpha_1 + \alpha_4\}$. A choice of simple roots is $\Delta^0 = \{\alpha_1 - \alpha_4, \alpha_1 + \alpha_4\}$. We can consider the matrix representation of the simple roots

$$\Delta_0 = \left[\begin{bmatrix} 1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \end{bmatrix} \right].$$

However to be consistent with the output of Maple we will write the matrix representation of a root space element as a column vector. So using this notation

$$\Delta_0 = \begin{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{bmatrix}.$$

We will also refer to the matrix representation of a root system as the root system.

Let B be the Killing form of $\mathfrak{g}$. Let $\tilde{B}$ be the Killing form restricted to $\mathfrak{h}$. Because the Killing form is non-degenerate on this restriction we can find it's inverse $\tilde{B}^{-1}$.

Then to define an inner product $(\cdot, \cdot)$ on $\mathfrak{h}^*$ consider the matrix representation of $\tilde{B}^{-1}$. Specifically given $\alpha, \beta \in \mathfrak{h}^*$, then define

$$(\alpha, \beta) := \alpha^T \tilde{B}^{-1} \beta$$

Therefore we have an inner product on the roots of a given Lie algebra.

If $\alpha, \beta \in \Delta$ and $\beta \neq \pm\alpha$, then the following will also be roots

$$\beta - p\alpha, \beta - (p-1)\alpha, \ldots, \beta - \alpha, \beta, \beta + \alpha, \ldots, \beta + q\alpha$$

which we call the **α-string through** β. A property of root systems is that $p + q \leq 3$ and that $p - q = 2\frac{(\alpha,\beta)}{(\alpha,\alpha)}$ and that the number given by $2\frac{(\alpha,\beta)}{(\alpha,\alpha)}$ is an integer.

2.8 Dynkin Diagrams

A **Dynkin diagram** is a graph where each node correspond to a simple root of the Lie algebra. Given two simple roots α, β, the number of edges $d_{\alpha\beta}$ between two nodes is

$$d_{\alpha,\beta} = 4\frac{(\alpha, \beta)(\beta, \alpha)}{(\alpha, \alpha)(\beta, \beta)}$$

where $(\cdot, \cdot)$ is the inner product on $\mathfrak{h}^*$ defined in Section 2.7. A property of Lie algebras says that $d_{\alpha\beta} \in \{0, 1, 2, 3\}$ [13]. If $d_{\alpha\beta} > 1$ then an arrow is drawn over the edges from the longer root to the shorter root.

Example 2.8.1 - Constructing a Dynkin Diagram

For the Lie algebra $so(4,3)$ one possible choice of simple roots is

$$[\alpha_1, \alpha_2, \alpha_3] = \left[\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right]$$

Where the Killing form restricted to the Cartan subalgebra is

$$B = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}.$$

From this information we can compute the following:

$$d_{\alpha_1\alpha_2} = 4\frac{(\alpha_1, \alpha_2)(\alpha_2, \alpha_1)}{(\alpha_1, \alpha_1)(\alpha_2, \alpha_2)} = 1$$

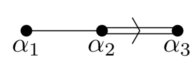
$$d_{\alpha_1\alpha_3} = 4\frac{(\alpha_1, \alpha_3)(\alpha_3, \alpha_1)}{(\alpha_1, \alpha_1)(\alpha_3, \alpha_3)} = 0$$

$$d_{\alpha_2\alpha_3} = 4\frac{(\alpha_2, \alpha_3)(\alpha_3, \alpha_2)}{(\alpha_2, \alpha_2)(\alpha_3, \alpha_3)} = 2.$$

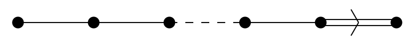
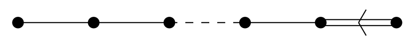
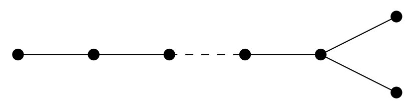
Because $d_{\alpha_2\alpha_3} > 1$ we need to see which is the longer root. Then because

$$(\alpha_2, \alpha_2) = \frac{1}{5} > \frac{1}{10} = (\alpha_3, \alpha_3)$$

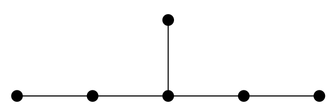
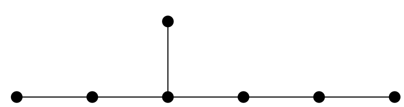
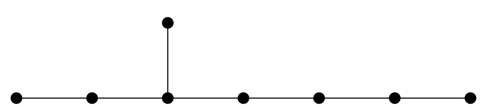
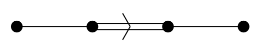
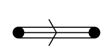
we can draw the following diagram:



The following table summarizes the Dynkin diagrams for the classical Lie algebras:

Lie Algebra		Dynkin Diagram
A_n	$n > 0$	
B_n	$n > 1$	
C_n	$n > 2$	
D_n	$n > 3$	

We can also create a table of the Dynkin diagrams of the exceptional Lie algebras.

Lie Algebra	Dynkin Diagram
E_6	
E_7	
E_8	
F_4	
G_2	

2.9 Extended Dynkin Diagrams

Let Δ be the system of all roots of a given Lie algebra $\mathfrak{g}$ with Cartan subalgebra $\mathfrak{h}$. Then let $\Delta^0 = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ be a choice of simple roots for this Lie algebra. Then every root $\beta \in \Delta$ can be written as a sum of simple roots

$$\beta = \sum_{\alpha \in \Delta^0} k_\alpha \alpha$$

The **height** of a root $\beta = \sum_{\alpha \in \Delta^0} k_\alpha \alpha$ is defined to be

$$\sum_{\alpha \in \Delta^0} k_\alpha$$

The **highest root** is a unique root δ such that the height of δ is greater than the height of any other root in Δ, $(\delta, \alpha) \geq 0$ for all simple roots $\alpha \in \Delta$, and all the $k_\alpha > 0$. If δ is the highest root of a given root system then $-\delta$ is defined to be the **lowest root**.

We define the set $\tilde{\Delta}^0 = \{-\delta, \alpha_1, \alpha_2, \ldots, \alpha_r\}$ to be the **extended root system** of the simple root system Δ^0. Then the Dynkin diagram with $r+1$ nodes corresponding to the extended root system is the **Extended Dynkin Diagram**.

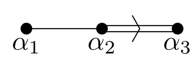
The program **ExtendedDynkinDiagram** was created to produce the extended Dynkin diagrams in Maple for the classical and exceptional Lie algebras.

Example 2.9.1 - Constructing an Extended Dynkin Diagram

From Example 2.8.1 we know that the Lie algebra $so(4,3)$ has the following choice of simple roots

$$[\alpha_1, \alpha_2, \alpha_3] = \left[\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right]$$

and that $so(4,3)$ has the following Dynkin diagram:



To find the Extended Dynkin Diagram we first need to find the highest root and the corresponding extended root system.

Consider the positive root $\begin{bmatrix}1\\0\\0\end{bmatrix} = \begin{bmatrix}1\\-1\\0\end{bmatrix} + \begin{bmatrix}0\\1\\-1\end{bmatrix} + \begin{bmatrix}0\\0\\1\end{bmatrix}$. This root has a height of 3. However the highest root will be $\begin{bmatrix}1\\1\\0\end{bmatrix} = \begin{bmatrix}1\\-1\\0\end{bmatrix} + 2\begin{bmatrix}0\\1\\-1\end{bmatrix} + 2\begin{bmatrix}0\\0\\1\end{bmatrix}$ which has a height of 5. Therefore the extended root system is

$$[-\delta, \alpha_1, \alpha_2, \alpha_3] = \left[\begin{bmatrix}-1\\-1\\0\end{bmatrix}, \begin{bmatrix}1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\-1\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}\right]$$

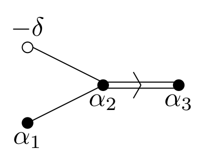
Then to find the extended Dynkin diagram we compute the following:

$$d_{-\delta,\alpha_1} = 4\frac{(-\delta,\alpha_1)(\alpha_1,-\delta)}{(-\delta,-\delta)(\alpha_1,\alpha_1)} = 0$$

$$d_{-\delta,\alpha_2} = 4\frac{(-\delta,\alpha_2)(\alpha_2,-\delta)}{(-\delta,-\delta)(\alpha_2,\alpha_2)} = 1$$

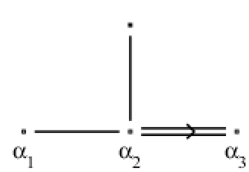
$$d_{-\delta,\alpha_3} = 4\frac{(-\delta,\alpha_3)(\alpha_3,-\delta)}{(-\delta,-\delta)(\alpha_3,\alpha_3)} = 0$$

Because $d_{-\delta,\alpha_2} = 1$ we draw a line between these two nodes in the Extended Dynkin diagram. With this information we can draw the following extended Dynkin diagram for $so(4,3)$:



Using the knowledge that the Lie algebra is the type B_3 we use the command ExtendedDynkin-Diagram in Maple to create this diagram.

```
> ExtendedDynkinDiagram("B3");
```



Example 2.9.2: Extended Dynkin Diagram of A_3

In this example we will consider the Lie algebra $sl(4)$ which is of type A_3. One choice of simple roots for this Lie algebra is

$$[\alpha_1, \alpha_2, \alpha_3] = \begin{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \end{bmatrix}$$

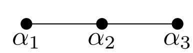
The highest root for this Lie algebra is

$$\delta = \alpha_1 + \alpha_2 + \alpha_3 = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}.$$

which has a height of 3. Therefore the extended root system is the roots

$$[-\delta, \alpha_1, \alpha_2, \alpha_3] = \begin{bmatrix} \begin{bmatrix} -2 \\ -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \end{bmatrix}.$$

From Section 2.8 we know that the Dynkin diagram for $sl(4)$ is



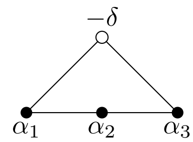
Then to draw the extended Dynkin diagram we need to determine which nodes from the simple roots connect to the node for the lowest root.

$$d_{-\delta,\alpha_1} = 4\frac{(-\delta, \alpha_1)(\alpha_1, -\delta)}{(-\delta, -\delta)(\alpha_1, \alpha_1)} = 1$$

$$d_{-\delta,\alpha_2} = 4\frac{(-\delta, \alpha_2)(\alpha_2, -\delta)}{(-\delta, -\delta)(\alpha_2, \alpha_2)} = 0$$

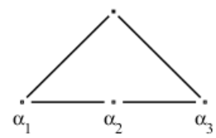
$$d_{-\delta,\alpha_3} = 4\frac{(-\delta, \alpha_3)(\alpha_3, -\delta)}{(-\delta, -\delta)(\alpha_3, \alpha_3)} = 1$$

So the node for $-\delta$ will be connected to both the nodes for α_1 and α_3 by one line. Thus the extended Dynkin diagram for $sl(4)$ will be

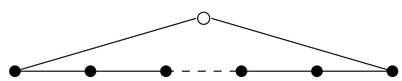
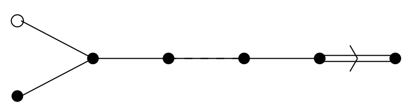
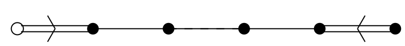
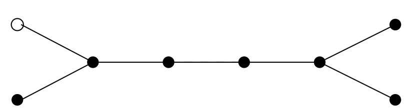


Again this diagram can be created in Maple using the command ExtendedDynkinDiagram.

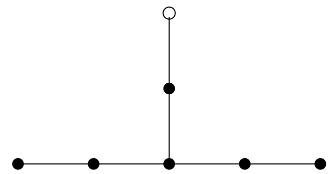
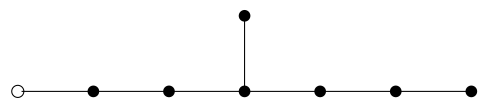
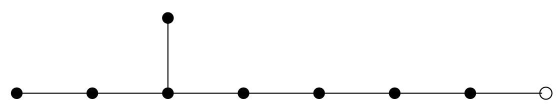
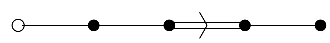
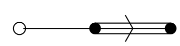
```
> ExtendedDynkinDiagram("A3");
```



The following table summarizes the extended Dynkin diagrams for the classical Lie algebras.

Lie Algebra		Extended Dynkin Diagram
A_n	$n > 0$	
B_n	$n > 1$	
C_n	$n > 2$	
D_n	$n > 3$	

Below are the extended Dynkin diagrams for the exceptional Lie algebras.

Lie Algebra	Extended Dynkin Diagram
E_6	
E_7	
E_8	
F_4	
G_2	

2.10 Weyl Group

Recall the inner product $(\cdot, \cdot)$ defined in Section 2.7, which is positive definite on $\mathfrak{h}^*$. Let Δ be the set of roots for $\mathfrak{g}$. Then for each root $\alpha \in \Delta$ we define $W_\alpha : \mathfrak{h}^* \to \mathfrak{h}^*$ to be the reflection about the hyperplane perpendicular to α. The group generated by these reflections is called the **Weyl group**.

For any $\beta \in \Delta$ we have that

$$W_\alpha(\beta) = \beta - 2\frac{(\alpha, \beta)}{(\alpha, \alpha)}\alpha$$

It also happens that $2\frac{(\alpha,\beta)}{(\alpha,\alpha)}$ is the Cartan matrix of the Lie algebra.

If we think of W_α as a linear transformation then the Weyl group element can be represented as an $r \times r$ matrix with respect to a basis of simple roots, where r is the rank of the Lie algebra.

Weyl Group properties include:

- A Weyl group element is also be viewed as a permutation on the roots Δ.
- If Δ^0, and $\tilde{\Delta}^0$ are both simple root systems of $\mathfrak{g}$ then there is a Weyl group element ω such that $\omega(\Delta^0) = \tilde{\Delta^0}$.
- The Weyl group can be generated by W_α, $\alpha \in \Delta^0$ (the Weyl group is generated by r reflections.)
- If $\beta \in \Delta$ then there is a Weyl group element U such that $\beta = U(\alpha)$ for some simple root α.

We call the r generating elements the **simple reflections.** Another way to think of a Weyl group element is as a product of the simple reflections.

Given any Weyl group element ω we can find a corresponding Lie algebra automorphism that preserves that Cartan subalgebra. We first write ω as a product of simple reflections; then for each simple reflection we can find the corresponding automorphism. Start by finding $X_\alpha \in \mathfrak{g}_\alpha$ and $Y_\alpha \in \mathfrak{g}_{-\alpha}$ that satisfy the bracket of $sl(2)$. Then compute the matrix

$$\exp(\mathrm{ad}(X_\alpha))\exp(\mathrm{ad}(-Y_\alpha))\exp(\mathrm{ad}(X_\alpha))$$

This linear transformation will be a Lie algebra automorphism that preserves the Cartan subalgebra. If we do this for each $\alpha \in \Delta^+$ we will have r Lie algebra automorphisms. To find the automorphism that corresponds to ω we compose the transformations in the same order that the simple reflections are multiplied.

A program in Maple was created to do this.

Example 2.10.1 - Weyl Group Element to Lie Algebra Automorphism

Start by creating the Lie algebra $sl(4)$, and use the command SimpleLieAlgebraProperties to create a record containing information about the Lie algebra.

```
> DGEnvironment[LieAlgebra]("sl(4)", sl4);
```

$$\mathit{Lie\ algebra} : \mathit{sl4}$$

```
> Prop := SimpleLieAlgebraProperties(sl4):
```

Then find a Weyl group element using the command WeylGroupElementPresentation.

```
> omega := WeylGroupElementPresentation([1], Prop, "Matrix");
```

$$\omega := \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Using the command WeylGroupElementToAutomorphism we can find the Lie algebra automorphism that corresponds to this Weyl group element.

```
> Φ := WeylGroupElementToAutomorphism(omega, Prop);
```

$$\Phi := e1 \to e2, e2 \to e1, e3 \to e3, e4 \to -e7, e5 \to -e8, e6 \to -e9, e7 \to -e4, e8 \to e5, e9 \to e6, e10 \to -e11, e11 \to e10, e12 \to e12, e13 \to -e14, e14 \to e13, e15 \to e15$$

We can verify that this is an automorphism and that it does preserve the Cartan subalgebra.

```
> Query(Phi, "Homomorphism");
```

$$true$$

```
> CSA := CartanSubalgebra(sl4);
```

$$CSA := [e1, e2, e3]$$

```
> ApplyLinearTransformation(Phi, CSA);
```

$$[e2, e1, e3]$$

In a similar manner a Maple program was written that takes a Lie algebra automorphism and returns the corresponding Weyl group element.

Example 2.10.2 - Lie Algebra Automorphism to Weyl Group Element

Continuing with the Lie algebra $sl4$ defined in Example 2.9.1 above start with an automorphism that preserves the Cartan subalgebra.

```
> Phi2 := LinearTransformation([[e1, e2 - e3], [e2, e1 - e3], [e3, -e3], [e4, -e7],
    [e5, e9], [e6, - e8], [e7, - e4], [e8, -e6], [e9, e5], [e10, e14], [e11, -e13],
    [e12, - e15], [e13, -e11], [e14, e10], [e15, -e12]]);
```

$$\Phi 2 := e1 \to e2 - e3, e2 \to e1 - e3, e3 \to -e3, e4 \to -e7, e5 \to e9, e6 \to -e8, e7 \to -e4, e8 \to -e6, e9 \to e5, e10 \to e14, e11 \to -e13, e12 \to -e15, e13 \to -e11, e14 \to e10, e15 \to -e12$$

Then using the command AutomorphismToWeylGroupElement we can find the corresponding Weyl group element.

```
> omega2 := AutomorhpismToWeylGroupElement(Phi2, Prop);
```

$$\omega 2 := \begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & -1 \\ 0 & 0 & -1 \end{bmatrix}$$

We can verify that this matrix squares to the identity

```
> omega2^2;
```

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

Thus we have $\omega 2$ which is a Weyl group element for the Lie algebra $sl(4)$.

Example 2.10.3 - Finding a Weyl Group Element

One question that we ask when given two root subsystems of a Lie algebra $\mathfrak{g}$ is if there is a Weyl group element ω that maps one root system to the other. If such a Weyl group element exists can we find it? The program EquivalenceOfSubsystems will do that for us.

For this example consider the Lie algebra $sl(5)$.

```
> DGEnvironment[LieAlgebra]("sl(5)", sl5);
```

$$\mathit{Lie\ algebra} : sl5$$

Next get a record of properties for this Lie algebra

```
> Prop := SimpleLieAlgebraProperties(sl5):
```

and find the root system Δ.

```
> PR := Prop :- PositiveRoots:
> Delta := simplify([op(PR), -op(PR)]):
```

Now choose two closed root subsystems.

```
> Gamma1 := [Delta[1], -Delta[1], Delta[3], -Delta[3]];
```

$$\Gamma 1 := \left[\begin{bmatrix}1\\-1\\0\\0\end{bmatrix}, \begin{bmatrix}-1\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\-1\end{bmatrix}, \begin{bmatrix}0\\0\\-1\\1\end{bmatrix}\right]$$

```
> Gamma2 := [Delta[8], - Delta[8], Delta[2], -Delta[2]];
```

$$\Gamma 2 := \left[\begin{bmatrix}1\\0\\0\\-1\end{bmatrix}, \begin{bmatrix}-1\\0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\-1\\1\\0\end{bmatrix}\right]$$

Given these two root subsystems of Δ we want to find a Weyl group element ω such that $\omega\Gamma_1 = \Gamma_2$.

```
> W := EquivalenceOfSubsystems(Gamma1, Gamma2, Prop);
```

$$W := \begin{bmatrix}0 & 0 & 1 & 0\\0 & 1 & 0 & 0\\1 & 0 & 0 & 0\\0 & 0 & 0 & 1\end{bmatrix}$$

We can check that this matrix satisfies the condition $\omega\Gamma_1 = \Gamma_2$.

```
> [seq(W.x, x in Gamma1)];
```

$$\left[\begin{bmatrix}0\\-1\\1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\-1\\0\end{bmatrix}, \begin{bmatrix}1\\0\\0\\-1\end{bmatrix}, \begin{bmatrix}-1\\0\\0\\1\end{bmatrix}\right]$$

Therefore there is at least one Weyl group element such that $\omega\Gamma_1 = \Gamma_2$. Now consider another set of roots Γ_3.

```
> Gamma3 := [Delta[1], -Delta[1], Delta[2], -Delta[2]];
```

$$\Gamma3 := \left[\begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ 1 \\ 0 \end{bmatrix}\right]$$

We want to know if there is a Weyl group element ω such that $\omega\Gamma_1 = \Gamma_3$.

```
> EquivalenceOfSubsystems(Gamma1, Gamma3, Prop);
```

$$[\ ]$$

Thus there is no Weyl group element such that $\omega\Gamma_1 = \Gamma_3$.

3 $sl(2)$ Representation Theory

A representation of a Lie algebra $\mathfrak{g}$ is a Lie algebra homomorphism

$$\varphi : \mathfrak{g} \to gl(V)$$

where V is a finite dimensional vector space. Let $\varphi : \mathfrak{g} \to gl(V)$ be a representation. Then the following properties will be true:

- $\mathrm{Im}(\varphi)$ is a subalgebra of $gl(V)$
- $\ker(\varphi)$ is an ideal of $\mathfrak{g}$

If the representation φ is injective then it is defined to be a **faithful representation.**

Let $\varphi : \mathfrak{g} \to gl(V)$ be a representation of a Lie algebra $\mathfrak{g}$. Then a subspace W of V is said to be **invariant** if $\varphi(x)(w) \in W$ for all $w \in W$ and $x \in \mathfrak{g}$. A nonzero representation is **irreducible** if the only invariant subspaces are V and $\{0\}$.

It is a common practice to call V the representation of $\mathfrak{g}$ and suppress the notation involving φ. So for $x \in \mathfrak{g}$ and $v \in V$ we would write $x \cdot v$ rather than $\varphi(x) \cdot v$.

3.1 Examples of Representations

Let $\mathfrak{g}$ be a Lie algebra and let V, W be finite dimensional vector spaces. Then we can consider the following examples of representations.

Example 3.1.1: The Trivial Representation

Let V be a field. Then define the mapping $\varphi : \mathfrak{g} \to gl(V)$ by $\varphi(x) = 0$ for all $x \in \mathfrak{g}$.This is defined to be the **trivial representation**. Every Lie algebra will has a trivial representation.

Note that the kernel of this representation will be the entire Lie algebra. Therefore for all non-zero $\mathfrak{g}$, φ is not one-to-one and thus the trivial representation will not be faithful.

Example 3.1.2: The Natural/Standard Representation

Let $\mathfrak{g}$ be a subalgebra of $gl(V)$. Then the inclusion map $i : \mathfrak{g} \to gl(V)$ is a Lie algebra homomorphism. The representation given by this mapping is defined to be the **natural or standard representation** of $\mathfrak{g}$. For example the Lie algebra $sl(V)$ will have a natural representation.

Because the inclusion map is a restriction of the identity map we know that i is injective; therefore the natural representation is always faithful.

Example 3.1.3: The Adjoint Representation

In Section 2.1 we defined the mapping $x \to ad_x$ where $ad_x(y) = [x, y]$ was said to be the **adjoint representation** of $\mathfrak{g}$. This is an important representation to study because several properties of a Lie algebra are related to the adjoint representation. For example the Killing form was defined using the adjoint representation.

Example 3.1.4: The Tensor Product

If V and W are representations then it is a fact that then tensor product $V \otimes W$ will also be a representation.

Example 3.1.5: Additional Representations

If V and W are representations of $\mathfrak{g}$ then the following will also be representations:

- The Direct Sum: $V \oplus W$
- The nth Tensor Product: $V^{\otimes n}$
- The Exterior Powers: $\wedge^n(V)$
- The Symmetric Powers: $\mathrm{Sym}^n(V)$
- The Dual $V^* = \mathrm{Hom}(V, \mathbb{C})$

3.2 Irreducible Representations of sl(2)

In this section we will define for integer d an irreducible representation of $sl(2)$.

When considering the irreducible representation of $sl(2)$ we will use the following basis $\{e, f, h\}$ for $sl(2)$ satisfying the following relations:

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

Which are the structure equations for $sl(2)$.

To find a representation of $sl(2)$ we first need to specify a vector space V. Consider the vector space $\mathbb{C}[X, Y]$ of polynomials in two variables with complex coefficients. Let d be an integer and define V_d to be the subspace of $\mathbb{C}[X, Y]$ consisting of polynomials in X and Y of degree d. For $d \geq 1$ the space V_d has a basis

$$X^d, X^{d-1}Y^1, X^{d-2}Y^2, \cdots, X^2Y^{d-2}, XY^{d-1}, Y^d.$$

Thus V_d is a $d+1$ dimensional complex vector space.

Now define a mapping $\varphi : sl(2) \to gl(V_d)$ by

$$\varphi(e) = X\frac{\partial}{\partial Y}, \quad \varphi(f) = Y\frac{\partial}{\partial X}, \quad \varphi(h) = X\frac{\partial}{\partial X} - Y\frac{\partial}{\partial Y}.$$

We can show this is a representation by showing that this mapping satisfies the structure equations of $sl(2)$.

$$\begin{aligned}
[\varphi(h), \varphi(e)] &= \left[X\frac{\partial}{\partial X} - Y\frac{\partial}{\partial Y}, X\frac{\partial}{\partial Y}\right] \\
&= X\left(1\frac{\partial}{\partial Y}\right) - Y\left(0\frac{\partial}{\partial Y}\right) - X\left(0\frac{\partial}{\partial X} - 1\frac{\partial}{\partial Y}\right) \\
&= 2X\frac{\partial}{\partial Y} \\
&= 2\varphi(e)
\end{aligned}$$

$$\begin{aligned}
[\varphi(h), \varphi(f)] &= \left[X\frac{\partial}{\partial X} - Y\frac{\partial}{\partial Y}, Y\frac{\partial}{\partial X}\right] \\
&= X\left(0\frac{\partial}{\partial X}\right) - Y\left(1\frac{\partial}{\partial X}\right) - Y\left(1\frac{\partial}{\partial X} - 0\frac{\partial}{\partial Y}\right) \\
&= -2Y\frac{\partial}{\partial X} \\
&= -2\varphi(f)
\end{aligned}$$

$$\begin{aligned}
[\varphi(e), \varphi(f)] &= \left[X\frac{\partial}{\partial Y}, Y\frac{\partial}{\partial X}\right] \\
&= X\left(1\frac{\partial}{\partial X}\right) - Y\left(1\frac{\partial}{\partial Y}\right) \\
&= \varphi(h)
\end{aligned}$$

Therefore this mapping is a representation of $sl(2)$.

Theorem 3.1. *The representation $\varphi : sl(2) \to gl(V_d)$ is irreducible.*

We can also consider the matrix representation corresponding to φ. The matrix of $\varphi(e)$ with respect to the basis for V_d listed above is

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & d \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

The matrix of $\varphi(f)$ with respect to the basis for V_d listed above is

$$\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ d & 0 & 0 & \cdots & 0 \\ 0 & d-1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}.$$

The matrix of $\varphi(h)$ with respect to the basis for V_d listed above is

$$\begin{bmatrix} d & 0 & 0 & \cdots & 0 \\ 0 & d-2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -d+2 & 0 \\ 0 & 0 & \cdots & 0 & -d \end{bmatrix}$$

where the diagonal entries are the numbers $d - 2k$ for $k = 0, 1, 2, \ldots, d$. If the bracket operation is the matrix commutator then these matrices will also satisfy the structure equations for sl(2).

Theorem 3.2. *Let V be a finite dimensional vector space. If $\rho : sl(2) \to gl(V)$ is an irreducible* *representation then V is isomorphic to one of the V_d*

Example 3.2.1: The Natural/Standard Representation

Let $d = 1$, Then we can look at the corresponding matrix representations:

$$\varphi(e) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \varphi(f) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \varphi(h) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

which satisfies the conditions of the Standard representation set forth is Example 3.1.2.

Example 3.2.2: The Adjoint Representation

Let $d = 2$, Then we can look at the corresponding matrix representations:

$$\varphi(e) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix}, \varphi(f) = \begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \varphi(h) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -2 \end{bmatrix}$$

This representation will correspond to the Adjoint representation of the Lie algebra.

Every representation V is completely reducible. That is a given representation V can be written as

$$V = (V_1)^{a_1} \oplus (V_2)^{a_2} \oplus \cdots \oplus (V_n)^{a_n}$$

where the V_i are irreducible representations.

Given a representation V, we can decompose V into a direct sum of irreducible representations using the Highest Weight Vector for $sl(2)$, $W = \{v \in V \mid e \cdot v = 0\}$.

4 Equivalence of Subalgebras

Let $\mathfrak{g}$ be a Lie algebra. Given two subalgebras $\mathfrak{f}_1$ and $\mathfrak{f}_2$ of $\mathfrak{g}$ one question that can be asked is if the two algebras are equivalent.

Two subalgebras $\mathfrak{f}_1$ and $\mathfrak{f}_2$ are said to be **conjugate** if there is a Lie algebra automorphism φ, mapping $\mathfrak{f}_1$ to $\mathfrak{f}_2$. Given a Lie algebra $\mathfrak{g}$, we would like to be able to list all of the subalgebras up to conjugacy. There are several papers that contain lists of subalgebras up to conjugacy. We would like to be able to recreate these lists using Maple.

Two subalgebras $\mathfrak{f}_1$ and $\mathfrak{f}_2$ of $\mathfrak{g}$ are said to be **linearly equivalent** if for every representation $\rho : \mathfrak{g} \to \mathfrak{gl}(V)$ the subalgebras $\rho(\mathfrak{f}_1), \rho(\mathfrak{f}_2)$ are conjugate under $GL(V)$. While the notion of linear equivalence is not discussed in this paper it would be one of the next related topics to cover.

If $\mathfrak{g}$ is a simple Lie algebra with a simple subalgebra $\mathfrak{s}$ then one way to address this issue is by looking at the **Dynkin index** of $\mathfrak{s}$. The Dynkin index of $\mathfrak{s}$ in $\mathfrak{g}$ is an invariant given by the equation

$$\mathrm{ind}(\mathfrak{s} \hookrightarrow \mathfrak{g}) = \frac{(x,x)_{\mathfrak{g}}}{(x,x)_{\mathfrak{s}}}, \quad x \in \mathfrak{s}$$

where $(\cdot,\cdot)_{\mathfrak{g}}$ is the normalized inner product on $\mathfrak{h}^*$ and $(\cdot,\cdot)_{\mathfrak{s}}$ is the normalized inner product on $\mathfrak{h}^*_{\mathfrak{s}}$. Note that this number does not depend on the choice of x.

Theorem 4.1. *The Dynkin index will be an integer.*

If two subalgebras are not conjugate then they will not have the same Dynkin index.

To determine if two subalgebras may be conjugate we have created the following program.

- **DynkinIndex**: This program takes a simple Lie algebra and a simple subalgebra and return the corresponding Dynkin index.

As an application of this program we are able to calculate the Dynkin index for a principal subalgebra of a simple Lie algebra and compare the results to those given in the paper *The Dynkin Index and* $sl(2)$*-Subalgebras of Simple Lie Algebras* by Dmitri I. Panyushev, [24].

4.1 Dynkin Index of A Subalgebra

Let $\mathfrak{g}$ be a simple Lie algebra with Cartan subalgebra $\mathfrak{h}$ and roots system Δ. As discussed in Section 2.7 there is the induced inner product $(\cdot,\cdot)$ on $\mathfrak{h}^*$. Let δ be the longest root in Δ. Then we normalize this inner product so that

$$(\delta,\delta)_{\mathfrak{g}} = 2$$

Let $\mathfrak{s}$ be a simple subalgebra of $\mathfrak{g}$. As a Lie algebra $\mathfrak{s}$ has its own Cartan subalgebra $\mathfrak{h}_{\mathfrak{s}}$. Then there is a normalized inner product $(\cdot,\cdot)_{\mathfrak{s}}$ on $\mathfrak{h}_{\mathfrak{s}}^*$.

We define the **Dynkin Index of $\mathfrak{s}$ in $\mathfrak{g}$** by

$$\operatorname{ind}(\mathfrak{s} \hookrightarrow \mathfrak{g}) := \frac{(x,x)_{\mathfrak{g}}}{(x,x)_{\mathfrak{s}}}, \quad x \in \mathfrak{s}$$

As stated in Theorem 4.1 the Dynkin index will be an integer. The Dynkin index is also a subalgebra invariant. In the paper *Semisimple Subalgebras of Semisimple Lie Algebras*, E. B. Dynkin writes that this index is "the simplest of the non-trivial invariants of the subalgebras."

A property of the Dynkin index is that if $\mathfrak{l}$ is a simple subalgebra of $\mathfrak{s}$ and $\mathfrak{s}$ is a simple subalgebra of $\mathfrak{g}$, then

$$\operatorname{ind}(\mathfrak{l} \hookrightarrow \mathfrak{s}) \cdot \operatorname{ind}(\mathfrak{s} \hookrightarrow \mathfrak{g}) = \operatorname{ind}(\mathfrak{l} \hookrightarrow \mathfrak{g})$$

Example 4.1.1: Finding the Dynkin Index of a Subalgebra

Let $\mathfrak{g}$ be the Lie algebra $g2$. Then as we will show in Example 5.4.1 the elements $[4e1+2e2, e8+e11, -6e5-10e14]$ of $g2$ define a basis for an $sl(2)$-subalgebra of $g2$. This can be verified using the Query command.

```
> a := evalDG([4 e1 + 2 e2, e8 + e11, -6 e5 - 10 e14]);
```

$$a := [4e1+2e2, e8+e11, -6e5-10e14]$$

```
> Query(a, "Subalgebra");
```

$$true$$

Next we want to find the normalized inner product $(\cdot,\cdot)_{\mathfrak{g}}$ for the Lie algebra $\mathfrak{g}$ restricted to the elements in $\mathfrak{a}$.

```
> Prop := SimpleLieAlgebraProperties(g2):
> Bg2 := KillingForm():
> Kg2 := NormalizedKillingForm(Bg2, Prop):
> IPg2 := Tensor:-TensorInnerProduct(Kg2, a, a);
```

$$IPg2 := \begin{bmatrix} 56 & 0 & 0 \\ 0 & 0 & 28 \\ 0 & 28 & 0 \end{bmatrix}$$

Then we will find the normalized inner product $(\cdot,\cdot)_{\mathfrak{a}}$ for the subalgebra $\mathfrak{a}$. To do this we first initialize $\mathfrak{a}$ as a Lie algebra.

```
> LD := LieAlgebraData(a, alg):
> DGEnvironment[LieAlgebra](LD);
```

$$Lie\,algebra : alg$$

Next we create a record that contains the Cartan subalgebra of $\mathfrak{a}$ and the simple roots.

```
> CSAalg := CartanSubalgebra(alg);
```

$$CSAalg := [e1]$$

```
> RSDalg := RootSpaceDecomposition(CSAalg):
> PRalg := PositiveRoots(RSDalg):
> SRalg := SimpleRoots(PRalg);
```

$$SRalg := [[2]]$$

```
> Rec := Record("CartanSubalgebra" = CSAalg, "SimpleRoots" = SR);
```

$$Rec := Record(CartanSubalgebra = [e1], SimpleRoots = [[2]])$$

Now we can find the Normalized Killing form.

```
> Bsub := Killing();
```

$$Bsub := \begin{bmatrix} 8 & 0 & 0 \\ 0 & 0 & 4 \\ 0 & 4 & 0 \end{bmatrix}$$

```
> IPalg := NormalizedKillingForm(Bsub, Rec);
```

$$IPalg := \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Then the Dynkin index is the integer n such that

$$n \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 56 & 0 & 0 \\ 0 & 0 & 28 \\ 0 & 28 & 0 \end{bmatrix}$$

In this case $n = 28$. Therefore the Dynkin index of this subalgebra is 28.

We created the command DynkinIndex to take a simple subalgebra and return the corresponding Dynkin index. If we use this command on the subalgebra in the example above we will again see that its Dynkin index is 28.

```
> DynkinIndex(a);
```

$$28$$

The Dynkin indices for all the $sl(2)$-triples of G_2 and F_4 are given Section 5.3 in Tables 5 and 6.

4.2 Dynkin Index of the Principal Subalgebra

If $\mathfrak{a}$ is the $sl(2)$ principal subalgebra of a Lie algebra $\mathfrak{g}$, then the Dynkin index of $\mathfrak{a}$ will be as listed in the following table

Dynkin Index of a Principle $sl(2)$ Subalgebra

$\mathfrak{g}$	A_n	B_n	C_n	D_n	E_6	E_7	E_8	F_4	G_2
$\operatorname{ind}(a \hookrightarrow g)$	$\binom{n+2}{3}$	$\frac{1}{2}\binom{2n+2}{3}$	$\binom{2n+1}{3}$	$\frac{1}{2}\binom{2n}{3}$	156 0	399	1240	156	28

Example 4.2.1: Finding the Dynkin Index of a Priniple Subalgebra of type A_n

For this example we will consider the algebra $sl(4)$. To find the Dynkin index we first want to find the Principal subalgebra using the command PrincipalSubalgebra.

```
> DGEnvironment[LieAlgebra]("sl(4)", sl4);
```

$$\mathit{Lie\ algebra} : sl4$$

```
> CSA := CartanSubalgebra(sl4):
> a := PrincipalSubalgebra(sl4, CSA);
```

$$a := [3\,e1 + e2 - e3,\, e4 + e8 + e12, 3\,e7 + 4\,e11 + 3\,e15]$$

Now use the command *DynkinIndex* to find the Dynkin index of this subalgebra.

```
> Ld := LieAlgebraData(a, alg1):
> DGEnvironment[LieAlgebra](Ld):
> DynkinIndex(a);
```

$$10$$

The Lie algebra $sl(4)$ is of type A_3 so according the table the Dynkin index will be

$$\binom{3+2}{3} = \frac{5!}{3!2!} = \frac{20}{2} = 10$$

5 $sl(2)$ Subalgebras

Let $\mathfrak{g}$ be a complex semisimple Lie algebra with Cartan subalgebra $\mathfrak{h}$. One type of subalgebra that we are interested in studying are those with the same structure equations as $sl(2)$. We call these types of subalgebras $sl(2)$-triples. To study these subalgebras there are two types of elements of $\mathfrak{g}$ that we are interested in. Let $x \in \mathfrak{g}$. Then if ad x is nilpotent we say that x is **nilpotent**. Similarly if ad x is diagonalizable we say that x is **semisimple**. Note only the zero element can be semisimple and nilpotent.

Accordingly there are two methods that can be used to find an $sl(2)$-triple containing a given element of $\mathfrak{g}$. If the element is nilpotent then the following theorem applies and there is an algorithm to find the corresponding $sl(2)$-triple.

Theorem 5.1 (Jacobson-Morosov Theorem). *Every nilpotent element of a complex semi-simple Lie algebra can be embedded in an $sl(2)$-triple.*

If the element of $\mathfrak{g}$ is semisimple it may not be contained in an $sl(2)$-triple, however if it is then we have an algorithm to determine what the $sl(2)$-triple will be.

To list all of the $sl(2)$ subalgebras of $\mathfrak{g}$ up to conjugacy we just need to find the $sl(2)$ subalgebras containing certain semisimple elements written in the form

$$x = \sum_{i=1}^{l} a_i \epsilon_i$$

where $a_1 = 0, 1$ or 2 and ϵ_i is a dual basis in $\mathfrak{h}$ to a choice of simple roots Δ^0.

Within $\mathfrak{g}$ there is a particular subalgebra that is unique up to conjugation, we call this subalgebra the principle subalgebra. Note that this is the subalgebra that we get when $a_i = 2$ for all values of i.

To study $sl(2)$ subalgebras we have created the following programs.

- **ThreeDimensionalSubalgebra**: This program takes an element of $\mathfrak{g}$ and returns the $sl(2)$-triple that contains the given element.
- **EpsilonCharacteristics**: This program takes the Cartan subalgebra of $\mathfrak{g}$ and returns the semisimple elements ϵ_i of $\mathfrak{g}$ which can be used to find all non-conjugate $sl(2)$ subalgebras of $\mathfrak{g}$ by considering linear combinations.
- **FindAllTDS**: This program returns a list of all the non-conjugate $sl(2)$ subalgebras of a given complex semisimple Lie algebra.

- **PrincipalSubalgebra**: This program finds the principal subalgebra of a given complex semisimple Lie algebra.

The goal of this section is to list all $sl(2)$ subalgebras of $\mathfrak{g}$ up to conjugation for low dimensional simple Lie algebras and the exceptional Lie algebras. Then we will compare our results to those given in the paper *Semisimple Subalgebras of Semisimple Lie Algebras* by E. B. Dynkin, [12].

As an application can use these programs to study properties of the principal subalgebra, which will be discussed more in the next chapter.

5.1 Nilpotent Elements

Let $\mathfrak{g}$ be a semisimple Lie algebra. Let $\mathfrak{a}$ be a semisimple subalgebra of $\mathfrak{g}$ then the following properties are true:

- Any nilpotent element of $\mathfrak{a}$ is also nilpotent in $\mathfrak{g}$.
- Any semisimple element of $\mathfrak{a}$ is a semisimple element of $\mathfrak{g}$.
- Any $x \in \mathfrak{a}$ will either be a semisimple or nilpotent element

If we have a complex semi-simple Lie algebra $\mathfrak{g}$, we want to determine all $sl(2)$-triples, in $\mathfrak{g}$. One way that we try to do this is by starting with the nilpotent elements of $\mathfrak{g}$.

The **Jacobson-Morosov Theorem** states that "every nilpotent element of a complex semi-simple Lie algebra can be embedded in an $sl(2)$-triple."

The proof of this theorem gives an algorithm to find an $sl(2)$-triple given a nilpotent element $e \in \mathfrak{g}$, which contains e.

The algorithm is outlined below:

Let $e \in \mathfrak{g}$ be a nonzero nilpotent element of $\mathfrak{g}$. We first find an $f \in \mathfrak{g}$ such that $[[f, e], e] = 2e$. Then let $x = [f, e]$, so that $[x, e] = 2e$.

Now we have two cases to consider. If $[x, f] = -2f$ then we would let $e_- = f$. Now if $[x, f]+2f \neq 0$, we can say that $[[x, f] + 2f, e] = 0$, leading us to conclude that $[x, f] + 2f \in C_{\{e\}}(\mathfrak{g})$. Then there is a $g \in C_{\{e\}}(\mathfrak{g})$ such that

$$[x, f] + 2f = [x, g], +2g.$$

Now if we let $e = e_+, x = x$, and $e_- = g - f$, then we have the desired commutation relations.

Example 5.1.1 - Finding an $sl(2)$-triple given a Nilpotent Element

Start by initializing the Lie algebra so(3,2).

```
> DGEnvironment[LieAlgebra]("so(3,2)", so32);
```

$$Liealgebra : so32$$

Next choose an element of the Lie algebra, say the element $e2$. We can verify that this element is Nilpotent by showing that the adjoint matrix raised to a finite power is the zero matrix.

```
> AdE3 := Adjoint(e2)^3;
```

$$AdE3 := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Now that we have a nilpotent element we want to find the corresponding $sl(2)$-triple. To do this we first create an arbitrary element Z.

```
> vars := [seq(z||i, i = 1 ..  10)]:
> Vec := DGinformation("FrameBaseVectors"):
> Z := DGzip(vars, Vec);
```

$$Z := z1e1 + z2e2 + z3e3 + z4e4 + z5e5 + z6e6 + z7e7 + z8e8 + z9e9 + z10e10$$

Next we want to find a vector f such that $[[f, e2], e2] = 2e2$.

```
> Eq := evalDG(LieBracket(LieBracket(Z, E), E) - 2*E):
> EqCoef := DGinformation(Eq, "CoefficientSet"):
```

```
> Sol := solve(EqCoef):
> f := DGsimplify(subs(seq(z||i = 0, i = 1 .. 10), subs(Sol, Z)));
```

$$f := -e3$$

Define x to be $[f, e2]$

```
> x := LieBracket(f, e2);
```

$$x := e1 - e4$$

At this point we have our nil-positive element and the neutral element, we only have to find e_-. To find e_- we first find a vector g such that $[x, f] + 2f = [x, g] + 2g$.

```
> Eq2 := evalDG(LieBracket(X, f) + 2*f - LieBracket(X, Z) - 2*Z):
> EqCoef2 := DGinformation(Eq2, "CoefficientSet"):
> Sol2 := solve(EqCoef2):
> g := DGsimplify(subs(seq(z||i = 0, i = 1 .. 10), subs(Sol2, Z)));
```

$$g := 0e1$$

Then define $e_- = g - f$.

```
> Em := evalDG(g - f);
```

$$Em := e3$$

Therefore we have an $sl(2)$-triple $[x, e_+, e_-] = [e1 - e4, e2, e3]$, which we can show satisfies the required bracket equations.

```
> LieAlgebraData([e1-e4, e2, e3], LD);
```

$$[e1, e2] = 2e2,\ [e1, e3] = -2e3,\ [e2, e3] = e1$$

The Maple command ThreeDimensionalSubalgebra will take a nilpotent element and return the corresponding $sl(2)$-triple.

```
> TDS := ThreeDimensionalSubalgebra(e2, "nilpotent");
```

$$TDS := [e1 - e4, e2, e3]$$

5.2 Semisimple Elements

Let $\mathfrak{a} = \{x, e_+, e_-\}$ be an $sl(2)$-triple. We will call x the neutral element and e_+ the nil-positive element. Given a semisimple element $x \in \mathfrak{g}$, we want to know which $sl(2)$-triples, if any, contain x as the neutral element. If ad$x$ on $\mathfrak{a}$ has eigenvalues $-2, 0, 2$ we say that x is a mono-semisimple element.

The eigenvalues of ad x on $\mathfrak{g}$ will be integers. Denote $\mathfrak{g}_p$ to be the eigenspace of ad$x$ for the eigenvalue p. Kostant claims that any nil-positive element in an $sl(2)$-triple will belong to $\mathfrak{g}_2$. Consider $\mathfrak{g}_0$ which is the kernel of ad$x$ which is a subalgebra. Then for each $e \in \mathfrak{g}_2$ we get a mapping ade : $\mathfrak{g}_0 \to \mathfrak{g}_2$.

Define T_e to be the restriction of ad$e$ to $\mathfrak{g}_0$ and define $\hat{\mathfrak{g}}_2 := \{e \in \mathfrak{g}_2 \mid T_e \text{ maps } \mathfrak{g}_0 \text{ onto } \mathfrak{g}_2\}$. Therefore $e \in \hat{\mathfrak{g}}_2$ if and only if the rank of T_e equals the dimension of $\mathfrak{g}_2$.

Thus when finding the nil-positive element of an $sl(2)$-triple it is necessary that $e_+ \in \hat{\mathfrak{g}}_2$.

Once we have the nil-positive element e we can use the algorithm described above to find what the remaining basis element, e_- will be.

Example 5.2.1 - Not all Semisimple Elements are Contained in an $sl(2)$-Triple

Start by creating the Lie algebra so(2,2).

```
> DGEnvironment[LieAlgebra]("so(2,2)", so22);
```

$$Liealgebra : so22$$

We know that every element of the Cartan subalgebra is semisimple. Therefore to find a suitable monosemisimple element we first calculate the Cartan subalgebra.

```
> CSA := CartanSubalgebra(so22);
```

$$CSA := [e1, e4]$$

Now choose an element $e1$ from the Cartan subalgebra. We know that this element will be semisimple, so we just need to test if it is monosemisimple. Start by finding the adjoint

```
> AdX := Adjoint(e1);
```

$$AdX := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Next we find the characteristic polynomial that is associated with the adjoint matrix.

```
> CP := LinearAlgebra:-CharacteristicPolynomial(AdX, lambda);
```

$$CP := \lambda^6 - 2\lambda^4 + \lambda^2$$

To determine if 2 is an eigenvalue we evaluate the characteristic polynomial at $\lambda = 2$ and see if it is equal to 0.

```
> subs(lambda = 2, CP);
```

$$36$$

Therefore we can conclude that although $e1$ is a semisimple element it is not contained within an $sl(2)$-triple.

Example 5.2.2 - Finding an $sl(2)$-triple given a Semisimple Element

We will continue with the Lie algebra defined in example 6.3.1. Choose the semisimple element $e1 - e4$. We need to verify that this is a monosemisimple element, so we find the adjoint matrix.

```
> AdX := Adjoint(e1 - e4);
```

$$AdX := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Because the matrix is diagonal we can conclude that the element $e1 - e4$ is semisimple. We can also see that 2 and -2 is are eigenvalues of the adjoint matrix. Therefore $e1-e4$ is a monosemisimple element. Next we will find g_2 which is the eigenspace of $ad\,x$ for eigenvalue 2.

```
> Z := DGzip([seq(t||i, i = 1 .. 6)], [e1, e2, e3, e4, e5, e6]):
> Eq := evalDG(LieBracket(e1-e4, Z) - 2*Z):
> EqCoef := DGinformation(Eq, "CoefficientSet"):
> A,b := LinearAlgebra:-GenerateMatrix(EqCoef, [seq(t||i, i = 1 .. 6)]):
> NullSp := convert(LinearAlgebra:-NullSpace(A), list):
> g2 := DGzip(NullSp, Vec);
```

$$g2 := [e2]$$

We now want to find the rank of T_{e2}, the restriction of $\operatorname{ad} e2$ to the centralizer of $e1 - e4$.

```
> Cent := Centralizer([e1-e4]);
```

$$[e6, e5, e4, e1]$$

```
> LD := LieDerivative(e2, Cent):
> GC := GetComponents(LD, g1):
> A :=Matrix(GC):
> LinearAlgebra:-Rank(A);
```

$$1$$

Therefore T_{e2} maps the centralizer onto g_2 and $e2 \in \hat{g}_2$. Thus $e2$ is the nil-positive element corresponding the the neutral element $e1 - e4$ in an $sl(2)$-triple. To find the $sl(2)$-triple we can use the algorithm described above for the nilpotent element $e2$.

```
> TDS := ThreeDimensionalSubalgebra(e2, "nilpotent")
```

$$TDS := [e1 - e4, e2, e3]$$

We can verify that this is indeed an $sl(2)$-triple by seeing that the bracket equations are satisfied.

```
> LieAlgebraData(TDS, ld);
```

$$[e1, e2] = 2\, e2,\ [e1, e3] = -2\, e3,\ [e2, e3] = e1]$$

Given a monosemisimple element the command ThreeDimensionalSubalgebra will return the corresponding $sl(2)$-triple.

```
> ThreeDimensionalSubalgebra(evalDG(e1-e4), "semisimple");
```

$$[e1 - e4, e2, e3]$$

We will use the monosemisimple elements to determine all of the $sl(2)$-triples up to conjugacy.

5.3 Finding All $sl(2)$ Subalgebras

Now that we can find an $sl(2)$-triple given an element of $\mathfrak{g}$, we want to be able to list them all up to conjugacy. One way that we do this is by looking at certain semisimple elements of $\mathfrak{g}$.

Let $\{\alpha_1, \ldots, \alpha_r\} = \Delta^0$ be a set of simple roots for a Lie algebra $\mathfrak{g}$. Let $\mathfrak{h}$ be the Cartan subalgebra. Then we define ϵ_i for $i = 1, 2, \ldots, r$ to be the dual basis in $\mathfrak{h}$ to the choice of simple roots, α_j.

The following algorithm will give the ϵ_i for a Lie algebra $\mathfrak{g}$ with Cartan subalgebra $\mathfrak{h}$.

Start by finding a choice of simple roots, Δ^0 for the Lie algebra. Let h^* be the dual space to the Cartan algebra. Then we write the simple roots as a linear combination of the basis elements of h^*. The dual basis to these linear combinations will be the ϵ_i.

If $x \in \mathfrak{g}$ is a monosemisimple element of an $sl(2)$-triple then it can be written as

$$x = \sum_{i=1}^{r} a_i \epsilon_i$$

where $a_i = 0, 1$ or 2. It is a theorem that the a_i will be non-negative integers. In Kostant's paper it was proven that the a_i's must be less than or equal to 2. Therefore the only possible choices that the a_i can take are 0, 1, or 2.

If $\{x_1, x_2, \ldots, x_b\}$ is the set of all monosemisimple elements that are contained in at least one $sl(2)$-triple there will be exactly b conjugate classes. Note that $b < 3^r$.

There is one particular semisimple element

$$x = \sum_{i=1}^{r} 2\epsilon_i$$

which will always be contained within an $sl(2)$-triple. This $sl(2)$-triple is called the **Principal Subalgebra** and will be discussed in greater details below.

To find all $sl(2)$-triples we first find the ϵ_i's and then consider all the possible linear combinations of which there will be 3^r. These will be the possible monosemisimple elements contained in an $sl(2)$-triple. Then we test each of the combinations to see which x_i are the neutral elements.

Example 5.3.1 - Finding the ϵ_i's of a Lie Algebra

Start by creating the Lie algebra G_2 and finding properties about the Lie algebra including the Cartan subalgebra and simple roots.

```
> DGEnvironment[LieAlgebra]("g(2, [6,8]", g2));
```

$$\textit{Lie algebra} : g2$$

```
> CSA := CartanSubalgebra(g2);
```

$$CSA := [e1, e2]$$

```
> RSD := RootSpaceDecomposition(CSA):
> PR := PositiveRoots(RSD):
> SR := SimpleRoots(PR);
```

$$SR := \left[\begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right]$$

Next find the the dual basis of the Cartan subalgebra and express the simple roots as linear combinations of the dual basis.

```
> Dual := [theta1, theta2];
```

$$Dual := [\theta 1, \theta 2]$$

```
> DB := DGzip(SR, Dual);
```

$$DB := [\theta 1 - \theta 2, \theta 2]$$

Then finding the dual basis of the dual basis will give the ϵ_i's.

```
> epsilon := DualBasis(DB, CSA);
```

$$\epsilon := [e1, e1 + e2]$$

The program EpsilonCharacteristics will return these monosemisimple elements.

```
> epsilon := EpsilonCharacteristics(CSA);
```

$$\epsilon := [e1, e1 + e2]$$

Now that we have found the epsilons we can find all the combinations of them which will be monosemisimple elements contained in an $sl(2)$-triple. The command AllThreeDimensionalSubalgebras will return the ϵ_i's as well as a list of which combinations correspond to non-conjugate $sl(2)$-triples. If a different choice of simple roots is used then the ϵ_i's will be different, but the combinations of the ϵ_i's which are in an $sl(2)$-triple will not change based on choice of simple roots.

```
> All := AllThreeDimensionalSubalgebras(g2, "characteristics", [e1, e2]);
```

$$All := [[[2, 2], [2, 0], [1, 0], [0, 1]], [e1 + e2, e1]]$$

Note that these ϵ characteristics match those listed by Dynkin in Table 16 [11].

By changing the command from "characteristics" to "subalgebras", the program will give an explicit list of all non-conjugate $sl(2)$-triples.

```
> AllThreeDimensionalSubalgebras(g2, "subalgebras", [e1, e2]);
```

$$[[2\,e1 + 8\,e2,\ e6 + e13, -10\,e7 - 6\,e12], [2\,e2, e5 + e9, -(2/3)\,e3 - (4/3)\,e7 - (2/3)\,e11 + (4/3)\,e14],$$

$$[e2, e4, -e10], [e1 + 3\,e2, e5, -e11]]$$

We can create tables that have information about the subalgebras including the Dynkin index and the epsilon characteristics.

Table 1: $sl(2)$-triples of $sl(3)$

Dynkin Index	Epsilon Characteristics	$sl(2)$-triple
1	1 1	$[e1,\ e4,\ e7]$
4	2 2	$[2\,e1,\ e3 + e6,\ 2\,e5 + 2\,e8]$

Table 2: $sl(2)$-triples of $sl(4)$

Dynkin Index	Epsilon Characteristics	$sl(2)$-triple
1	0 1 1	$[e1,\ e6,\ e13]$
2	2 0 0	$[e1 + e2 - e3,\ e6 + e8,\ e11 + e13]$
4	0 2 2	$[2\,e1,\ e4 + e9\,2\,e7 + 2\,e14]$
10	2 2 2	$[3\,e1 + e2 - e3,\ e4 + e8 + e12,\ 3\,e7 + 4\,e11 + 3\,e15]$

Table 3: $sl(2)$-triples of $sl(5)$

Dynkin Index	Epsilon Characteristics	$sl(2)$-triple
1	0 1 0 1	$[e1,\ e8,\ e21]$
2	1 0 1 0	$[e1 + e2 - e4,\ e8 + e11,\ e18 + e21]$
4	0 2 0 2	$[2\,e1,\ e6 + e16,\ 2\,e13 + 2\,e23]$
5	1 1 1 1	$[2\,e1 + e2 - e4,\ e6 + e11 + e16,\ 2\,e13 + e18 + 2\,e23]$
10	1 2 1 2	$[3\,e1 + e2 - e4,\ e5 + e11 + e20,\ 3\,e9 + 4\,e18 + 3\,e24]$
20	2 2 2 2	$[4\,e1 + 2\,e2 - 2\,e4,\ e5 + e10 + e15 + e20, 4\,e9 + 6\,e14 + 6\,e19 + 4\,e24]$

Table 4: $sl(2)$**-triples of** $sl(6)$

Dynkin Index	Epsilon Characteristics	$sl(2)$-triple
1	1 0 0 1 0	$[e1,\ e10,\ e31]$
2	0 0 1 0 1	$[e1 + e2 - e5,\ e10 + e14,\ e27 + e31]$
3	0 2 0 0 0	$[e1 + e2 + e3 - e4 - e5,\ e9 + e13 + e20,\ e22 + e26 + e33]$
4	2 0 0 2 0	$[2\,e1,\ e6 + e15,\ 2\,e11 + 2\,e32]$
5	1 0 1 1 1	$[2\,e1 + e2 - e5,\ e7 + e14 + e20,\ 2\,e16 + e27 + 2\,e33]$
8	0 0 2 0 2	$[2\,e1 + 2\,e2 - 2\,e5,\ e8 + e12 + e20 + e24,\ 2\,e17 + 2\,e21 + 2\,e29 + 2\,e33]$
10	2 0 1 2 1	$[3\,e1 + e2 - e5,\ e6 + e14 + e30,\ 3\,e11 + 4\,e27 + 3\,e35]$
11	2 2 0 2 0	$[3\,e1 + e2 + e3 - e4 - e5,\ e7 + e14 + e18 + e25,\ 3\,e16 + 4\,e23 + e27 + 3\,e34]$
20	2 0 2 2 2	$[4\,e1 + 2\,e2 - 2\,e5,\ e6 + e12 + e19 + e30,\ 4\,e11 + 6\,e17 + 6\,e28 + 4\,e35]$
35	2 2 2 2 2	$[5\,e1 + 3\,e2 + e3 - e4 - 3\,e5,\ e6 + e12 + e18 + e24 + e30,\ 5\,e11 + 8\,e17 + 9\,e23 + 8\,e29 + 5\,e35]$

Table 5: $sl(2)$**-triples of G2**

Dynkin index	Epsilon Characteristics	$sl(2)$-triple
1	1 0	$[e2, e4, -e10]$
3	0 1	$[e1 + 3\,e2, e5, -e11]$
4	2 0	$[2\,e2, e5 + e9, -(2/3)\,e3 - (4/3)\,e7 - (2/3)\,e11 + (4/3)\,e14]$
28	2 2	$[2\,e1 + 8\,e2,\ e6 + e13, -10\,e7 - 6\,e12]$

Table 6: $sl(2)$-triples of F4

Dynkin Index	Epsilon Characteristics	$sl(2)$-triple
1	1 0 0 0	$[e2,\ e6,\ -e30]$
2	0 0 0 1	$[2\,e1+4\,e2+2\,e3+e4, e21, -e45]$
3	0 1 0 0	$[e1+2\,e2, e18+e31, -e7-e42]$
4	2 0 0 0	$[2\,e2,\ e9+e29, -2\,e35-2\,e33]$
6	0 0 1 0	$[2\,e1+4\,e2+e3,\ e27+e31+e44, -e7-2\,e20-2\,e51]$
8	0 0 0 2	$[4\,e1+8\,e2+4\,e3+2\,e4,\ e25+e35, -2\,e11-2\,e49]$
9	0 1 0 1	$[3\,e1+6\,e2+2\,e3+e4,\ e10+e16+e50,\ -e26-2\,e34-2\,e40]$
10	2 0 0 1	$[2\,e1+6\,e2+2\,e3+e4,\ e28+e43,\ -3\,e19-4\,e52]$
11	1 0 1 0	$\left[2\,e1+5\,e2+e3,\ e12+e14+e43,\ -\left(\frac{3}{2}\right)e19+\left(\frac{3}{2}\right)e20-\left(\frac{5}{2}\right)e36-\left(\frac{3}{2}\right)e38-\left(\frac{3}{2}\right)e52\right]$

Table 6: $sl(2)$-triples of F4 cont.

Dynkin Index	Epsilon Characteristics	$sl(2)$-triple
12	0 2 0 0	$[2\,e1 + 4\,e2,\ e10 + e12 + e14 + e41,\ e13 - e15 - 2\,e17 + e19 + e20$ $+2\,e22 - e34 - e36 - 2\,e38 + e40 + e51 + 2\,e52]$
28	2 2 0 0	$[2\,e1 + 6\,e2,\ e12 + e29 + e37,\ -10\,e5 - 6\,e13 - 6\,e36]$
35	1 0 1 2	$[6\,e1 + 13\,e2 + 5\,e3 + 2\,e4,\ e25 + e43 + e44,\ -5\,e19 - 9\,e20 - 8\,e49]$
36	0 2 0 2	$[6\,e1 + 12\,e2 + 4\,e3 + 2\,e4,\ e23 + e25 + e39 + e44,\ -10\,e15 - 4\,e17 + 4\,e19$ $-10\,e20 - 4\,e47 - 4\,e49]$
60	2 2 0 2	$[6\,e1 + 14\,e2 + 4\,e3 + 2\,e4, e23 + e25 + e29 + e41, -14\,e5 - 5, e13 - 13\,e17$ $-5\,e20 - 5\,e47 - 5\,e49]$
156	2 2 2 2	$[10\,e1 + 22\,e2 + 6\,e3 + 2\,e4,\ e25 + e29 + e32 + e37,\ -22\,e5 - 30\,e8 - 42\,e13 - 16\,e49]$

5.4 Principal Subalgebra

As stated above for any simple subalgebra $\mathfrak{g}$ the element

$$x = \sum_{i=1}^{l} 2\epsilon_i$$

will be contained with in an $sl(2)$-triple called the **Principal Subalgebra.** This neutral element x is called the **principal regular element** of $\mathfrak{g}$. The corresponding nil-positive element e_+ will be called a **principal nilpotent element** of $\mathfrak{g}$. In the paper *The Principal Three-Dimensional Subgroup and the Betti Numbers of a Complex Simple Lie Group*, Bertram Kostant shows that this subalgebra is unique up to conjugation. Therefore we can conclude that every complex simple Lie algebra contains a uniquely defined $sl(2)$-triple.

Example 5.4.1 - Finding the Principal Subalgebra

Continuing from Example 5.3.1 we know that the semisimple ϵ_i's of G_2 are the elements $[e1,\ e1+e2]$. To find the principal subalgebra we start with the semisimple element

$$x = 2\,e1 + 2(e1 + e2)$$

and find the corresponding $sl(2)$-triple.

```
> X := evalDG(2*e1 + 2*(e1 + e2));
```

$$X := 4\,e1 + 2\,e2$$

```
> ThreeDimensionalSubalgebra(X, "semisimple");
```

$$[4\,e1 + 2\,e2,\ e8 + e11,\ -6\,e5 - 10\,e14]$$

These elements will be the basis of the principal subalgebra of G_2.

We were able to create the Maple program, PrincipalSubalgebra, that will return a principal subalgebra given a Cartan Subalgebra.

```
> PrincipalSubalgebra(g2, CSA);
```

$$[4\,e1 + 2\,e2,\ e8 + e11,\ -6\,e5 - 10\,e14]$$

6 Simple Lie Algebras as $sl(2)$ Representation Spaces

Let $\mathfrak{g}$ be a complex simple Lie algebra and let G be the adjoint group of $\mathfrak{g}$. A famous theorem in algebra states that the Poincaré polynomial, $p_G(t)$, of G factors into the form $p_G(t) = \Pi_{i=1}^{r}(1 + t^{d_i})$ where r is the rank of $\mathfrak{g}$ and the d_i are odd integers. The integers m_i defined by $d_i = 2m_i + 1$, are defined to be the **exponents** of the Lie algebra $\mathfrak{g}$. [17]

The problem of finding the exponents from the root structure of a Lie algebra was studied by Bott, Shapiro and Kostant. There are tables which list the exponents for the simple Lie algebras and the exceptional Lie algebras. [1]

For this section the following programs have been created.

- **LieAlgebraExponents**: This program reproduces the Lie algebra exponents from a table [1].

As shown by Kostant, [17], there is a relationship between the exponents and the principal subalgebra. In this section we will illustrate this theorem.

6.1 Lie Algebra Exponents

The first person to find a procedure to determine the exponents of a Lie algebra from its root system was R. Bott. When following his method we start with a system of roots Δ for a given Lie algebra $\mathfrak{g}$. Then let Δ^+ be the positive roots and Δ^0 a system of simple roots. Then for $\alpha \in \Delta^+$ we define $\sigma(\alpha)$ to be the height of the vector α. Then define b_k to be the number of roots $\beta \in \Delta^+$ such that $\sigma(\alpha) = k$. Then Bott's method claims that $b_k - b_{k+1}$ is the number of times that k is an exponent of $\mathfrak{g}$

Then later an alternate way was discovered by Arnold Shapiro. His method is as follows: Let $\mathfrak{a}$ be the principal subalgebra of a Lie algebra $\mathfrak{g}$. Then decompose the adjoint representation of $\mathfrak{a}$ into a direct sum of l irreducible representations. The dimensions of these irreducible components we will write as d_i. Then the **exponents** for $i = 1 \ldots l$ will be the integers

$$m_i = \frac{d_i - 1}{2}.$$

Example 6.1.1 - Finding the Exponents

Start by initializing the Lie algebra $so(4, 3)$ and finding its Cartan subalgebra.

```
> DGEnvironment[LieAlgebra]("so(4,3)", so43);
```

$$Lie\,algebra : so43$$

```
> CSA := CartanSubalgebra(so43);
```

$$CSA := [e1,\, e5,\, e9]$$

Next we find the Principal subalgebra, initialize it as a Lie algebra, and create a record of relevant properties

```
> PS := PrincipalSubalgebra(so43, CSA);
```

$$PS := [6\,e1 + 4\,e5 + 2\,e9,\; e2 + e6 + e18,\; 6\,e4 + 10\,e8 - 12\,e21]$$

```
> LD := LieAlgebraData(PS, alg):
> DGEnvironment[LieAlgebra](LD, vectorlabels = [f]);
```

$$Lie\,algebra : alg$$

```
> Prop := SimpleLieAlgebraProperties([f1]):
```

Now find the adjoint representation for each element of the Principal subalgebra and create a representation ρ mapping alg to $so(4,3)$.

```
> AdPS := map(Adjoint, PS);
> rho := Representation(alg, so43, AdPS):
```

Next find the Highest weigh vectors

```
> HWV := HighestWeightVectors(rho, Prop);
```

$$HWV := [e10,\; e12 + e16,\; e2 + e6 + e18]$$

Then find the irreducible representations

```
> IR := seq(GenerateIrreducibleRepresentation(rho, HWV[i], Prop), i = 1..2);
```

$$IR := \Big[\big[e1 - 4e5 + 5e9, e2 - \left(\tfrac{9}{5}\right) e6 + e18, e3 - e17, e4 - 3e8 - 2e21, e7 + 2e21, e10,$$

$$e11, e12 - 2e16, e13, e14, e15 - e19], [e1 - e5 - e9, e2 - \left(\frac{1}{2}\right) e18, e3 + \left(\frac{1}{2}\right) e17, e4 + e21,$$

$$e7 - e20, e12 + e16, e15 + \left(\frac{1}{2}\right) e19], [e1 + \left(\frac{2}{3}\right) e5 + \left(\frac{1}{3}\right) e9, e2 + e6 + e18, e4 + \left(\frac{5}{3}\right) e8 - 2e21]]$$

Then if we find the dimensions of the irreducible representations we will have the d_i

```
> d := seq(nops(IR[i]), i = 1..3);
```

$$d := 11,\ 7,\ 3$$

Then the exponent will be

```
> m := seq((d[i] - 1)/2, i = 1..3);
```

$$m = 5,\ 3,\ 1$$

We can verify that these are the exponents using the program LieAlgebraExponents

```
> LieAlgebraExponents("B", 3);
```

$$1,\ 3,\ 5$$

The following table found in the book *Algebraic Integrability, Painlevé Geometry and Lie Algebras* [1] summarizes the exponents of the simple Lie algebras.

$\mathfrak{g}$	Exponents	$\mathfrak{g}$	Exponents	$\mathfrak{g}$	Exponents
A_r	$1, 2, \ldots, r$	D_r	$1, 3, \ldots 2r-3, r-1$	E_6	$1, 4, 5, 7, 8, 11$
B_r	$1, 3, 5, \ldots, 2r-1$	F_4	$1, 5, 7, 11$	E_7	$1, 5, 7, 9, 11, 13, 17$
C_r	$1, 3, 5, \ldots, 2r-1$	G_2	$1, 5$	E_8	$1, 7, 11, 13, 17, 19, 23, 29$

The problem of determining the exponents of a given Lie algebra was further studied by H. M. Coxeter. His observations on the subject were related to the Weyl group but will not be discussed in this paper.

7 Regular Subalgebras

Let $\mathfrak{g}$ be a complex semisimple Lie algebra with Cartan subalgebra $\mathfrak{h}$. A subalgebra $\mathfrak{f}$ of $\mathfrak{g}$ is said to be **regular with respect to the Cartan subalgebra** $\mathfrak{h}$ if $\mathfrak{h}$ is contained in the normalizer, $n_{\mathfrak{f}}(\mathfrak{g})$; that is to say $\mathfrak{f}$ is regular with respect to $\mathfrak{h}$ if $[\mathfrak{h}, \mathfrak{f}] \subset \mathfrak{f}$. A subalgebra $\mathfrak{f}$ is defined to be **regular** if it is regular with respect to some Cartan subalgebra. The basic theorem regarding regular subalgebras is:

Theorem 7.1. *If $\mathfrak{f}$ is a regular subalgebra of $\mathfrak{g}$ with respect to $\mathfrak{h}$ then it can be written in the form*

$$\mathfrak{f} = f(\mathfrak{t}, \Gamma) = \mathfrak{t} \oplus \sum_{\alpha \in \Gamma} \mathfrak{g}_\alpha$$

where Γ is a closed subsystem in the root system Δ and $\mathfrak{t}$ is a subspace in $\mathfrak{h}$ containing the elements h_α for $\alpha \in \Gamma \cap (-\Gamma)$.

Recall that for any $\alpha \in \Delta$, $h_\alpha = [\mathfrak{g}_\alpha, \mathfrak{g}_{-\alpha}]$. A subsystem of roots Γ is closed if for all $\alpha, \beta \in \Gamma$ that $\alpha + \beta \in \Gamma$ if $\alpha + \beta \in \Delta$

As a consequence of Theorem 7.1 we conclude that regular subalgebras are those which can be aligned with a root space decomposition. To study regular subalgebras we have created the following programs.

- **VerifySubsystem**: This program is used to check to see if a subsystem of roots is closed or symmetric.
- **VerifyRegular**: This program is used to check to see if a subalgebra of $\mathfrak{g}$ is a regular subalgebra.
- **ClosedRootSystem**: This program takes a set of roots and returns the smallest closed subsystem containing them.
- **SymmetricRootSystem**: This program takes a set of roots and returns the smallest symmetric subsystem containing them.
- **ClosedSubsysetmToRegularSubalgebra**: This command takes a closed subsystem and creates the corresponding regular subalgebra as defined by the above equation.
- **RegularSubalgebraToRootSystem**:This command takes a regular subalgebra (which can be checked using the VerifySubsystem proram) and return a closed subsystem.

We use these programs to illustrate theorems relating to regular subalgebras. As an application of these commands we calculate the semisimple regular subalgebras of some small dimensional subalgebra and compare the results to those given in the paper *Classification of Semisimple Subalgebras of Simple Lie Algebras*, by M. Lorente and B. Gruber, [18].

7.1 Root Systems

Let Δ be a root system. Let Γ be a subsystem of Δ. Let $\alpha, \beta \in \Gamma$. Then if $\alpha + \beta \in \Gamma$ for all $\alpha + \beta \in \Delta$ we say that Γ is a **closed** subsystem. A subsystem Γ is said to be **symmetric** if $-\Gamma = \Gamma$, that is if $\alpha \in \Gamma$ then $-\alpha \in \Gamma$. Given a subsystem Γ we are able to test if it is closed and if it is not we can find its closure.

Example 7.1.1 - Closed and Symmetric Subsystems

For this example we will consider the root system for a Lie algebra of type B3. Start by using the command **AbstractRootSystem** which will return a record containing the following properties of B3 as an abstract root system: Rank, RootType, NumberOfPositiveRoots, PositiveRoots, CartanMatrix, RootTable, HighestRoot, LowestForm, SimpleRoots, InverseSimRts, and CrossedSimpleRoots.

```
> Prop := AbstractRootSystem("B3"):
```

The record Prop will contain a choice of positive roots for the Lie algebra and using this information we can determine Δ.

```
> Delta := [op(Prop:-PositiveRoots), seq(-Prop:-PositiveRoots[i], i =1..9)];
```

$$\Delta := \left[\begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}1\\1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \begin{bmatrix}1\\1\\1\end{bmatrix}, \begin{bmatrix}0\\1\\2\end{bmatrix}, \begin{bmatrix}1\\1\\2\end{bmatrix}, \begin{bmatrix}1\\2\\2\end{bmatrix}, \begin{bmatrix}-1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\-1\end{bmatrix}, \begin{bmatrix}-1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\-1\\-1\end{bmatrix}, \begin{bmatrix}-1\\-1\\-1\end{bmatrix}, \begin{bmatrix}0\\-1\\-2\end{bmatrix}, \begin{bmatrix}-1\\-1\\-2\end{bmatrix}, \begin{bmatrix}-1\\-2\\-2\end{bmatrix} \right]$$

Now that we have a root system Δ we can choose a subsystem Γ.

```
> Gamma := [Delta[1], Delta[7], Delta[3]
```

$$\Gamma := \left[\begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\2\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}\right]$$

We want to know if Γ is closed or symmetric. Note that

$$\begin{bmatrix}1\\0\\0\end{bmatrix} + \begin{bmatrix}0\\1\\2\end{bmatrix} = \begin{bmatrix}1\\1\\2\end{bmatrix}$$

which is an element of Δ but not of Γ so we would conclude that Γ is not closed. Similarly $\begin{bmatrix}1\\0\\0\end{bmatrix} \in \Gamma$ but $\begin{bmatrix}-1\\0\\0\end{bmatrix} \notin \Gamma$ so we conclude that Γ is not symmetric. The program **VerifySubsystem** will test both of these conditions for a given subsystem.

```
> VerifySubsystem(Gamma, Delta, "closed");
```

$$false$$

```
> VerifySubsystem(Gamma, Delta, "symmetric");
```

$$false$$

The program **ClosedRootSystem** will return the smallest closed subsystem containing Γ.

```
> Gamma2:= ClosedRootSystem(Gamma, Delta);
```

$$\Gamma 2 := \left[\begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\2\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}1\\1\\2\end{bmatrix}\right]$$

We can verify that this system is in fact closed.

```
> VerifySubsyshrtem(Gamma2, Delta, "closed");
```

$$true$$

The program **SymmetricRootSystem** will return the smallest symmetric subsystem containing Γ.

```
> Gamma3 := SymmetricRootSystem(Gamma, Delta);
```

$$\Gamma := \left[\begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\2\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}-1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\-1\\-2\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}\right]$$

Again we can verify that this subsystem is symmetric.

```
> VerifySubsystem(Gamma3, Delta, "symmetric");
```

$$true$$

7.2 Subalgebra Corresponding to a Root Subsystem

Recall from above that any regular subalgebra can be written in the form

$$\mathfrak{f} = f(\mathfrak{t}, \Gamma) = \mathfrak{t} \oplus \sum_{\alpha \in \Gamma} \mathfrak{g}_\alpha$$

where Γ is a closed subsystem in the root system Δ and $\mathfrak{t}$ is a subspace in $\mathfrak{h}$ containing the element h_α for any $\alpha \in \Gamma \cap (-\Gamma)$.

Example 7.2.1 - $sl(2)$ is a regular subalgebra

Let $\mathfrak{g}$ be a semisimple Lie algebra with Cartan subalgebra $\mathfrak{h}$ and root space Δ. Let $\alpha \in \Delta$, then as described in Section 2.7 we know that the set $\{\mathfrak{g}_\alpha, \mathfrak{g}_{-\alpha}, [\mathfrak{g}_\alpha, \mathfrak{g}_{-\alpha}]\}$ forms a basis for a subalgebra $\mathfrak{f}$ which is isomorphic to $sl(2)$. We first want to show that $\mathfrak{f}$ is a regular subalgebra.

We know for any $x \in \mathfrak{g}_\alpha$, $y \in \mathfrak{g}_{-\alpha}$ and $h \in \mathfrak{h}$, that $[h, x] \in \mathfrak{f}$ and $[h, y] \in \mathfrak{f}$. Because $\mathfrak{g}$ is a semisimple Lie algebra we know that $\mathfrak{h}$ is commutative and thus for any $z \in [\mathfrak{g}_\alpha, \mathfrak{g}_{-\alpha}]$, $[h, z] = 0$.

Thus $\mathfrak{f}$ is a regular subalgebra with respect to $\mathfrak{h}$.

Note that the set $\Gamma = \{\alpha, -\alpha\}$ is a closed subsystem of Δ. It is clear that the subalgebra $\mathfrak{f}$ can be written as

$$\mathfrak{f} = \{h_\alpha\} \oplus \{\mathfrak{g}_\alpha\} \oplus \{\mathfrak{g}_{-\alpha}\} = f(h_\alpha, \Gamma)$$

Therefore we have an example of a regular subalgebra of $\mathfrak{g}$ with respect to $\mathfrak{h}$ which is written in the form $\mathfrak{f} = t \oplus \sum_{\alpha \in \Gamma} \mathfrak{g}_\alpha$.

Given a closed subsystem Γ we can use the formula $\mathfrak{f} = t \oplus \sum_{\alpha \in \Gamma} \mathfrak{g}_\alpha$ to find a corresponding regular subalgebra.

Start by finding all the $\alpha \in \Gamma \cap (-\Gamma)$, then let $\tilde{t} = \{h_\alpha \mid \alpha \in \Gamma \cap (-\Gamma)\}$. Define t to be a linearly independent spanning set of $\tilde{t}$.

Let $\mathfrak{f}$ be the Lie algebra with a basis $\{t, g_\beta\}$ for each $\beta \in \Gamma$, then $\mathfrak{f}$ will be a regular subalgebra of $\mathfrak{g}$.

Example 7.2.2

Consider the Lie algebra sl(5). Using Maple we can find a choice of root system Δ and determine a subsystem Γ.

```
> DGEnvironment[LieAlgebra]("sl(5)", sl5);
```

$$Lie\,algebra : sl5$$

```
> CSA := CartanSubalgebra(sl5);
```

$$CSA := [e1, e2, e3, e4]$$

```
> RSD := RootSpaceDecomposition(CSA):
> Delta := [op(PositiveRoots(RSD)), op(simplify(-PositiveRoots(RSD)))]:
> Gamma := [Delta[1], Delta[2], Delta[14], Delta[12];
```

$$\Gamma := \left[\begin{bmatrix}1\\0\\0\\-1\end{bmatrix}, \begin{bmatrix}0\\1\\-1\\0\end{bmatrix}, \begin{bmatrix}-2\\-1\\-1\\-1\end{bmatrix}, \begin{bmatrix}0\\-1\\1\\0\end{bmatrix}\right]$$

Next we want to determine if Γ is a closed subsystem.

```
> VerifySubsystem(Gamma, Delta, "closed");
```

$$false$$

This is not a closed subsystem so we will find its closure, $\tilde{\Gamma}$

```
> GammaClosed := ClosedRootSystem(Gamma, Delta);
```

$$GammaClosed := \left[\begin{bmatrix}1\\0\\0\\-1\end{bmatrix}, \begin{bmatrix}0\\1\\-1\\0\end{bmatrix}, \begin{bmatrix}-2\\-1\\-1\\-1\end{bmatrix}, \begin{bmatrix}0\\-1\\1\\0\end{bmatrix}, \begin{bmatrix}-1\\-1\\-1\\-2\end{bmatrix}\right]$$

Now that we have a closed subsystem $\tilde{\Gamma}$ we need to determine all of the $\alpha \in \tilde{\Gamma} \cap (-\tilde{\Gamma})$. Start by determining $-\tilde{\Gamma}$

```
> GammaClosedNeg := simplify(-GammaClosed);
```

$$GammaClosedNeg := \left[\begin{bmatrix}-1\\0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\-1\\1\\0\end{bmatrix}, \begin{bmatrix}2\\1\\1\\1\end{bmatrix}, \begin{bmatrix}0\\1\\-1\\0\end{bmatrix}, \begin{bmatrix}1\\1\\1\\2\end{bmatrix}\right]$$

Note that there are two elements in $\tilde{\Gamma} \cap (-\tilde{\Gamma})$

```
> GammaIntersection := [GammaClosed[2], GammaClosed[4]];
```

$$GammaIntersection := \left[\begin{bmatrix}0\\-1\\1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\-1\\0\end{bmatrix}\right]$$

For each $\alpha \in \tilde{\Gamma} \cap (-\tilde{\Gamma})$ we can find h_α.

```
> hAlpha := [seq(RootToCartanSubalgebraElementH(GammaIntersection[i], RSD), i =1..2)];
```

$$hAlpha := [e2 - e3, -e2 + e3]$$

Then we let t be a basis of these two elements

```
> t := DGbasis(hAlpha);
```

$$t := [e2 - e3]$$

For each $\alpha \in \Gamma$ we now find $\mathfrak{g}_\alpha$

```
> S := [seq(RootSpace(GammaClosed[i], RSD), i=1..5)];
```

$$S := [e7,\ e10,\ e21,\ e14,\ e24]$$

We claim that the subalgebra $\mathfrak{f}$ with basis elements $\{t, S\} = \{e2 - e3,\ e7,\ e10,\ e21,\ e14,\ e24\}$ will be a regular subalgebra. We can first verify that it is in fact a subalgebra.

```
> f := evalDG([op(t), op(S)]);
```

$$f := [e2 - e3, e7,\ e10,\ e21,\ e14,\ e24]$$

```
> Query(f, "Subalgebra");
```

$$true$$

Now show that $[\mathfrak{h}, \mathfrak{f}] \in \mathfrak{f}$ to verify that $\mathfrak{f}$ is a regular subalgebra.

```
> BracketOfSubspaces(CSA, f);
```

$$[e7, -2e21, -e24, e10, -e14]$$

Therefore $\mathfrak{f}$ is a regular subalgebra. This can also be checked using the command **VerifyRegular.**

```
> VerifyRegular(f, CSA);
```

$$true$$

The command **ClosedSubsystemToRegularSubalgebra** will return this regular subalgebra.

```
> ClosedSubsystemToRegularSubalgebra(GammaClosed, Delta, RSD);
```

$$[-e2 + e3,\ e7,\ e10,\ e21,\ e14,\ e24]$$

7.3 Conjugate Regular Subalgebras and the Weyl Group

Let Γ_1 and Γ_2 be two closed subsystems of a root system Δ of a complex semisimple Lie algebra $\mathfrak{g}$.

Theorem 7.2. *The subalgebras $f(\mathfrak{h}, \Gamma_1)$ and $f(\mathfrak{h}, \Gamma_2)$ are conjugate in $\mathfrak{g}$ (through an automorphism which fixes that Cartan subalgebra) if and only if there is element W of the Weyl group such that $W\Gamma_1 = \Gamma_2$.*

Proof. Assume that the subalgebras $\mathfrak{f}_1 = f(\mathfrak{h}, \Gamma_1)$ and $\mathfrak{f}_2 = f(\mathfrak{h}, \Gamma_2)$ are conjugate in $\mathfrak{g}$. This means that there is a Lie algebra automorphism φ of $\mathfrak{g}$ such that $\varphi(\mathfrak{f}_1) = \mathfrak{f}_2$ which fixes the Cartan subalgebra. Now the roots of $\mathfrak{f}_1$ are Γ_1 and the roots of $\mathfrak{f}_2$ are Γ_2. Then as discussed in Section 2.9, for a given Lie algebra automorphism there is a corresponding Weyl group element, which maps the roots Γ_1 in Δ to another set of roots in Δ, in this case to Γ_2.

Assume that there is an element W of the Weyl group such that $W\Gamma_1 = \Gamma_2$. Then as discussed in section 2.8 we can find the corresponding Lie algebra automorphism ψ of $\mathfrak{g}$ such that $\psi(\mathfrak{h}) = \mathfrak{h}$. This automorphism will map $\mathfrak{f}_1$ to $\mathfrak{f}_2$. Then we can conclude that $\mathfrak{f}_1 = f(\mathfrak{h}, \Gamma_1)$ and $\mathfrak{f}_2 = f(\mathfrak{h}, \Gamma_2)$ are conjugate in $\mathfrak{g}$.

$\square$

Example 7.3.1

Consider the Lie algebra $so(3,3)$. We can use Maple to find the root system Δ for this Lie algebra.

```
> DGEnvironment[LieAlgebra]("so(3,3)", so33);
```

$$Lie\,algebra : so33$$

```
> Prop := SimpleLieAlgebraProperties(so33):
> CSA := Prop:-CartanSubalgebra;
```

$$CSA := [e1, e5, e9]$$

```
> RSD := Prop:-RootSpaceDecomposition;
> Delta := [op(Prop:-PostiveRoots), op(Prop:-NegativeRoots)]:
```

Now that we have a root systemt Δ we can find a closed subsystem Γ_1.

```
> Gamma1 := [Delta[1], Delta[3], Delta[5]];
```

$$\Gamma 1 := \left[\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \right]$$

```
> VerifySubsystem(Gamma1, Delta, "closed");
```

$$true$$

Now we can find an element W of the Weyl group

```
> W := WeylGroupElementPresentation([3,2], Prop, "Matrix");
```

$$W := \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Next find the closed subsystem $\Gamma_2 = W\Gamma_1$

```
> Gamma2 := [seq(W.GammaC[i], i =1..3)];
```

$$\Gamma 2 := \left[\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \right]$$

```
> VerifySubsystem(Gamma2, Delta, "closed");
```

$$true$$

First we want to find the subalgebra $\mathfrak{f}_1 = f(\mathfrak{h}, \Gamma_1)$

```
> f1 := ClosedSubsystemToRegularSubalgebra(Gamma1, Delta, RSD, CartanSubalgebra = CSA);
```

$$f1 := [e1, e5, e9, e2, e12, e11];$$

```
> Query(f1, "Subalgebra");
```

$$true$$

Then we find the regular subalgebra $\mathfrak{f}_2 = f(\mathfrak{h}, \Gamma_2)$

```
> f2 := ClosedSubsystemToRegularSubalgebra(Gamma2, Delta, RSD, CartanSubalgebra = CSA);
```

$$f2 := [e1, e5, e9, e10, e15, e3]$$

Next we will find the Lie algebra automorphism that corresponds to W and apply this transformation to $\mathfrak{f}_1$.

```
> Phi := WeylGroupElementToAutomorphism(W, Prop):
> ApplyLinearTransformation(Phi, f1);
```

$$[e1, -e5, -e9, e10, e15, e3]$$

Notice that $\Phi(\mathfrak{f}_1) = \mathfrak{f}_2$.

8 Regular Reductive Subalgebras

Let $\mathfrak{g}$ be a complex semisimple Lie algebra with Cartan subalgebra $\mathfrak{h}$. A subalgebra $\mathfrak{f}$ of $\mathfrak{g}$ is defined to be **reductive** if $\operatorname{rad}\mathfrak{f}$ consists only of semisimple elements. Equivalently $\mathfrak{f}$ is reductive if the adjoint representation of $\mathfrak{f}$ in $\mathfrak{g}$ is completely reducible.

When studying regular reductive subalgebras we consider the subsystems of the Lie algebra root system Δ. Recall Theorem 7.1 that for a Lie algebra $\mathfrak{g}$ with Cartan subalgebra $\mathfrak{h}$, a regular subalgebra $\mathfrak{f}$ can we written in the form $\mathfrak{f} = \mathfrak{f}(\mathfrak{t}, \Gamma) = \mathfrak{t} \oplus \sum_{\alpha \in \Gamma} \mathfrak{g}_\alpha$.

Theorem 8.1. *A regular subalgebra $\mathfrak{f}(\mathfrak{t}, \Gamma)$ is reductive in $\mathfrak{g}$ if and only if the closed subsystem Γ is symmetric. A regular reductive subalgebra $\mathfrak{f} = \mathfrak{f}(t, \Gamma)$ is semisimple if and only if $\mathfrak{t}$ is generated by $\mathfrak{h}_\alpha$, where $\alpha \in \Gamma$.*

To study regular reductive subalgebras we have created the following programs.

- **VerifyReductive**: This program is used to check to see if a subalgebra of $\mathfrak{g}$ is a reductive subalgebra.

We will use this program (as well as those described in above sections) to illustrate theorems relating to reductive subalgebras.

In Chapter 11 we will focus on maximal reductive non-semisimple subalgebras.

8.1 Reductive Subalgebras and Symmetric Root Systems

In this section we will show how to find a regular reductive subalgebra given a root subsystem. We will illustrate Theorem 8.1 with an example:

Example 8.1.1: Symmtetric Subsystem to Reductive Lie Algebra

For this example we will consider the Lie algebra $sl(4)$.

Start by initializing the Lie algebra.

```
> DGEnvironment[LieAlgebra]("sl(4)", sl4);
```

$$\text{Lie algebra} : sl4$$

Next we want to find the Cartan subalgebra:

```
> CSA := CartanSubalgebra(sl4);
```

$$CSA := [e1, e2, e3]$$

Then we want to get a list of all the roots of the Lie algebra

```
> RSD := RootSpaceDecomposition(CSA):
> PR := PostivieRoots(RSD);
```

$$PR := \left[\begin{bmatrix}2\\1\\1\end{bmatrix}, \begin{bmatrix}1\\0\\-1\end{bmatrix}, \begin{bmatrix}1\\2\\1\end{bmatrix}, \begin{bmatrix}1\\1\\2\end{bmatrix}, \begin{bmatrix}0\\1\\-1\end{bmatrix}, \begin{bmatrix}1\\-1\\0\end{bmatrix}\right]$$

```
> Delta := simplify([op(PR), -op(PR)]);
```

$$\Delta := \left[\begin{bmatrix}2\\1\\1\end{bmatrix}, \begin{bmatrix}1\\0\\-1\end{bmatrix}, \begin{bmatrix}1\\2\\1\end{bmatrix}, \begin{bmatrix}1\\1\\2\end{bmatrix}, \begin{bmatrix}0\\1\\-1\end{bmatrix}, \begin{bmatrix}1\\-1\\0\end{bmatrix}, \begin{bmatrix}-2\\-1\\-1\end{bmatrix}, \begin{bmatrix}-1\\0\\1\end{bmatrix}, \begin{bmatrix}-1\\-2\\-1\end{bmatrix}, \begin{bmatrix}-1\\-1\\-2\end{bmatrix}, \begin{bmatrix}0\\-1\\1\end{bmatrix}, \begin{bmatrix}-1\\1\\0\end{bmatrix}\right]$$

Now that we have our root system Δ we choose a closed and symmetric subsystem Γ.

```
> Gamma := [Delta[1], Delta[3], -Delta[1], -Delta[3], Delta[6], -Delta[6]];
```

$$\Gamma := \left[\begin{bmatrix}2\\1\\1\end{bmatrix}, \begin{bmatrix}1\\2\\1\end{bmatrix}, \begin{bmatrix}-2\\-1\\-1\end{bmatrix}, \begin{bmatrix}-1\\-2\\-1\end{bmatrix}, \begin{bmatrix}1\\-1\\0\end{bmatrix}, \begin{bmatrix}-1\\1\\0\end{bmatrix}\right]$$

We verify that Γ is both closed and symmetric.

```
> VerifySubsystem(Gamma, Delta, "symmetric");
```

$$true$$

```
> VerifySubsystem(Gamma, Delta, "closed");
```

$$true$$

We can use the equation from Theorem 7.1 to find the corresponding subalgebra, $\mathfrak{f} = \mathfrak{f}(\mathfrak{h}, \Gamma)$.

```
> f := ClosedSubsystemToRegularSubalgebra(Gamma, Delta, RSD, CartanSubalgebra = CSA);
```

$$f := [e1, e2, e3, e6, e9, e13, e14, e4, e7]$$

Now that we have a subalgebra we want to verify that it is reductive, so we need to find the radical of the subalgebra. To do this we first initialize the subalgebra as its own Lie algebra.

```
> LD := LieALgebraData(f, alg):
> DGEnvironment[LieAlgebra](LD);
```

$$Lie\ algebra : alg$$

Now we use the command Radical to find $rad\,\mathfrak{f}$.

```
> radf := Radical(alg);
```

$$radf := \left[-\left(\frac{1}{3}\right)e1 - \left(\frac{1}{3}\right)e2 + e3\right]$$

Next we need to check the element in the radical is in fact a semisimple element of the subalgebra.

```
> Adjoint(op(radf), [e1, e2, e3, e4, e5, e6, e7, e8, e9]);
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Note that because this adjoint matrix is the zero matrix that in this case the radical of the subalgbera is contained in the center of the subalgebra. The adjoint matrix of the element in $rad\,\mathfrak{f}$ is diagonalizable so we conclude that it is a semisimple element.Therefore the subalgebra $\mathfrak{f}$ is a reductive subalgebra. We can show this with the program VerifyReductive.

```
> VerifyReductive(alg);
```

$$true$$

Example 8.1.2: Continuing Example When Γ is not Symmetric

Still working with the Lie algebra $sl(4)$ from Example 8.1.1 we will now choose Γ to be the following subsystem:

```
> Gamma := [Delta[1], Delta[3]];
```

$$\Gamma := \left[\begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \right]$$

We can show that Γ is closed but not symmetric.

```
> VerifySubsystem(Gamma, Delta, "closed");
```

$$true$$

```
> VerifySubsystem(Gamma, Delta, "symmetric");
```

$$false$$

Next create the subalgebra $\mathfrak{f} = \mathfrak{f}(\mathfrak{h}, \Gamma)$ as shown in Theorem 7.1

```
> f := ClosedSubsystemToRegularSubalgebra(Gamma, Delta, RSD, CartanSubalgebra = CSA);
```

$$f := [e1, e2, e3, e6, e9]$$

We can verify that this really is a subalgebra of $sl(4)$.

```
> Query(f, "Subalgebra");
```

$$true$$

To test if $\mathfrak{f}$ is reductive we want to set up $\mathfrak{f}$ as its own Lie algebra.

```
> LD := LieAlgebraData(f, alg):
> DGEnvironment[LieAlgebra](LD);
```

$$Lie\ algebra : alg$$

Then find the radical of this Lie algebra

```
> rad := Radical(alg);
```

$$rad := [e5, e4, e3, e2, e1]$$

Next consider the element $e5$ in $rad\,\mathfrak{f}$, and look at the adjoint matrix

```
> Ad := Adjoint(rad[1]);
```

$$Ad := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & -1 & 0 & 0 \end{bmatrix}$$

Then see if this matrix is diagonalizable.

```
> LinearAlgebra:-JordanForm(Ad);
```

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

The adjoint matrix for $e5$ is not diagonalizable so we can conclude that $e5$ is not semisimple. Therefore $\mathfrak{f}$ is not a reductive Lie algebra. We can also verify this using the command VerifyReducitve.

```
> VerifyReductive(alg);
```

$$false$$

Thus staring with a non-symmetric closed subsystem Γ the corresponding subalgebra $\mathfrak{f}(\mathfrak{h}, \Gamma)$ is not reductive.

8.2 Reductive Semisimple Subalgebras

In this section we will use the programs created to show the second part of Theorem 8.1.

Example 8.2.1: Reductive Semisimple Subalgebra

Recall the Lie algebra $so(3,3)$ with root system Δ. Begin by initializing this as a Lie algebra and find its root system Δ.

```
> DGEnvironment[LieAlgebra]("so(3,3)", so33);
```

$$\textit{Lie algebra} : so33$$

```
> Prop := SimpleLieAlgebraProperties(so33):
> CSA := Prop:-CartanSubalgebra:
> RSD := Prop:-RootSpaceDecomposition:
> PR := Prop:-PositiveRoots;
```

$$PR := \left[\begin{bmatrix}1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\-1\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \begin{bmatrix}1\\0\\-1\end{bmatrix}, \begin{bmatrix}1\\0\\1\end{bmatrix}, \begin{bmatrix}1\\1\\0\end{bmatrix}\right]$$

```
> Delta := simplify([op(PR), -op(PR)]);
```

$$\Delta := \left[\begin{bmatrix}1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\-1\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \begin{bmatrix}1\\0\\-1\end{bmatrix}, \begin{bmatrix}1\\0\\1\end{bmatrix}, \begin{bmatrix}1\\1\\0\end{bmatrix}, \begin{bmatrix}-1\\1\\0\end{bmatrix}, \begin{bmatrix}0\\-1\\1\end{bmatrix}, \begin{bmatrix}0\\-1\\-1\end{bmatrix}, \begin{bmatrix}-1\\0\\1\end{bmatrix}, \begin{bmatrix}-1\\0\\-1\end{bmatrix}, \begin{bmatrix}-1\\-1\\0\end{bmatrix}\right]$$

We can choose a symmetric subsystem Γ of Δ.

```
> Gamma := [Delta[1], -Delta[1]];
```

$$\Gamma := \left[\begin{bmatrix}1\\-1\\0\end{bmatrix}, \begin{bmatrix}-1\\1\\0\end{bmatrix}\right]$$

It is clear that this is both a closed and symmetric subsystem. So by the first part of Theorem 8.1 we know that the corresponding subalgebra will be a reductive subalgebra. We will first consider the case where $\mathfrak{t}$ is generated by $\mathfrak{h}_\alpha$, where $\alpha \in \Gamma$.

```
> f1 := ClosedSubsystemToRegularSubalgebra(Gamma, Delta, RSD);
```

$$f1 := [-e1 + e5, e2, e4]$$

In this case $\mathfrak{t}$ is the single element $-e1 + e5$

We check that this is a subalgebra of $so(3,3)$.

```
> Query(f1, "Subalgebra");
```

$$true$$

We also check whether it is regular, reductive, or semisimple.

```
> Ld1 := LieAlgebraData(f1, alg1):
> DGEnvironment[LieAlgebra](Ld1):
> Query(alg1, "Semisimple");
```

$$true$$

```
> VerifyRegular(f1, CSA);
```

$$true$$

```
> VerifyReductive(alg1);
```

$$true$$

Therefore $f = \mathfrak{f}(t, \Gamma)$ is semisimple when $\mathfrak{t}$ is generated by $\mathfrak{h}_\alpha$, where $\alpha \in \Gamma$.

Next we will consider the case where $\mathfrak{t} = [e1, e5, e9]$ which is the entire Cartan subalgebra.

```
> f2 := ClosedSubsystemToRegularSubalgebra(Gamma, Delta, RSD, CartanSubalgebra = CSA);
```

$$f2 := [e1, e5, e9, e2, e4]$$

We check that this is a subalgebra of $so(3,3)$.

```
> Query(f2, "Subalgebra");
```

$$true$$

We also check whether or not it is reductive and semisimple.

```
> Ld2 := LieAlgebraData(f2, alg2):
> DGEnvironment[LieAlgebra](Ld2, vectorlabels = [E]):
> Query(alg2, "Semisimple");
```

$$false$$

```
> VerifyReductive(alg2);
```

$$true$$

So we see that because $\mathfrak{t}$ is not generated by $\mathfrak{h}_\alpha$, where $\alpha \in \Gamma$ that the subalgebra is reductive but is not semisimple. Consider the Levi decomposition of this subalgebra.

```
> LeviDecomposition(alg2);
```

$$[[E1 + E2, E3], [E1 - E2, E4, E5]]$$

Now if we write this back in terms of our Lie algebra $so(3,3)$ we see that the radical of $\mathfrak{f}_2$ is $[e1 + e5, e9]$ and that the semisimple portion is $[e1 - e5, e4, e5]$ which is the subalgebra $\mathfrak{f}_1$.

9 Maximal Regular Semisimple Subalgebras

Let Δ be a root system of a Lie algebra $\mathfrak{g}$. To study the maximal regular semisimple subalgebras of $\mathfrak{g}$ we must look at maximal closed symmetric subsystems of Δ.

A subalgebra $\mathfrak{f} \neq \mathfrak{g}$ is defined to be a **maximal** subalgebra if there is no other subalgebra $\mathfrak{f}_1$ such that $\mathfrak{f}_1 \neq \mathfrak{f}$, $\mathfrak{f}_1 \neq \mathfrak{g}$, and $\mathfrak{f} \subset \mathfrak{f}_1 \subset \mathfrak{g}$.

A subsystem $M \subset \Delta$ is called a **π-system** if M is linearly independent and $\alpha - \beta \notin \Delta$ for all $\alpha, \beta \in M$. From this definition we can see that a π-system will not be closed or symmetric. Conversely, if Γ is a closed and symmetric subsystem then there is a π-system M such that $[M] = \Gamma$.

A key property of π-systems is that they are always subsets of simple roots. Similarly, any system of simple roots of a closed symmetric subsystem is a π-system.

Given a subsystem N of Δ we define $[N]$ which is the set of all roots in Δ that are integral linear combinations of the roots in N.

$$[N] = \left\{x = \sum m_i \alpha_i \mid m_i \in \mathbb{Z}, \alpha_i \in N, x \in \Delta\right\}$$

Let $x, y \in [N]$ then we can show that $x + y \in [N]$ and $-x \in [N]$; therefore $[N]$ is both closed and symmetric. Then using the equation given in Theorem 7.1 we can find the regular semisimple subalgebra that corresponds to $[N]$.

Let $\Delta^0 = \{\alpha_1, \ldots, \alpha_r\}$ be a set of simple roots of $\mathfrak{g}$. Let α_0 be the lowest root of Δ.

Note that there are unique positive integers n_i such that

$$-\alpha_0 = \sum_{i=1}^{r} n_i \alpha_i \tag{9.1}$$

Recall from Section 2.9 that the extended root system is $\tilde{\Delta}^0 = \Delta^0 \cup \{\alpha_0\}$.

Theorem 9.1. *Let $\Gamma = [M]$ be a closed subsystem with associated π-system M. Then Γ is a maximal proper closed symmetric subsystem in Δ if and only if the π-system M can be obtained from the extended system of simple roots $\tilde{\Delta}^0 = \{\alpha_0, \alpha_1, \ldots, \alpha_r\}$ in one of the following ways:*

1. *$M = \tilde{\Delta}^0 \backslash \{\alpha_i\}$, where n_i is a prime number (from equation (10.1))*

2. *$M = \tilde{\Delta}^0 \backslash \{\alpha_0, \alpha_i\} = \Delta^0 \backslash \{\alpha_i\}$, where $i > 0$ and $n_i = 1$.*

Here the rank of Γ is r and $r - 1$ respectively.

As a result of Theorem 9.1 the problem of classifying all non-conjugate maximal regular subalgebras is equivalent to the problem of listing all maximal closed symmetric subsystems in the above forms, where the subsystems are not related by a Weyl group element.

In this chapter we will consider the case when the maximal closed symmetric subsystem is of the form $[\tilde{\Delta}^0\backslash\{\alpha_i\}]$, then the corresponding subalgebra will be a maximal regular semisimple subalgebra. If the the maximal closed symmetric subsystem is of the form $[\tilde{\Delta}^0\backslash\{\alpha_0, \alpha_i\}] = [\Delta^0\backslash\{\alpha_i\}]$ then the corresponding subalgebra will be a maximal non-semisimple reductive subalgebra, (which will be discussed in Chapter 11.)

For this section the following programs have been created.

- **SubsystemBracket**: This command takes a subsystem N and returns the smallest subsystem which contains all integral linear combinations roots in N. The output will be $[N]$.
- **MaximalPiSystem**: This command takes a subsystem N and finds the largest π-system M contained in the subsystem. The results will be the largest π-system M such that $[M] \subset N$.
- **VerifySubsystem**: This program is used to check if a given subsystem of roots N is a π-system.

We can use these programs to verify the results given in Table 5 of the book *Lie Groups and Lie Algebras III* by Onishchik and Vinberg, [23].

9.1 π-systems

In this section we will illustrate the programs SubsystemBracket, MaximalPiSystem, and VerifySubsystem.

Example 9.1.1 - Find $[N]$

Consider the abstract root system for the Lie algebra B5 and find its root space Δ.

```
> ARS := AbstractRootSystem("B5"):
> Delta := simplify([op(ARS:-PositiveRoots), op(-(ARS:-PositiveRoots))]:)
```

Now choose a subsystem N of Δ.

```
> N := [Delta[3], Delta[7]];
```

$$N := \left[\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \right]$$

The command SubsystemBracket will find all the roots in Δ which are integer linear combinations of roots in M.

```
> BracketN := SubsystemBracket(N, Delta);
```

$$BracketN := \left[\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right]$$

We can verify that $[N]$ is both closed and symmetric

```
> VerifySubsystem(BracketN, Delta, "closed");
```

$$true$$

```
> VerifySubsystem(BracketN, Delta, "symmetric");
```

$$true$$

We can use the program VerifySubsystem to check if a subsystem is a π-system or not.

```
> VerifySubsystem(BracketN, Delta, "pisystem");
```

$$false$$

Therefore BracketN is closed and symmetric but not a π-system.

Now that we have a closed and symmetric subsystem we can use the program MaximalPiSystem to find a π-system M such that $[M] =$ BracketN.

```
> M := MaximalPiSystem(BracketN, Delta);
```

$$M := \left[\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ -1 \\ 0 \\ 0 \end{bmatrix} \right]$$

We can verify that this subspace is not closed or symmetric but is a π-system.

```
> VerifySubsystem(M, Delta, "closed");
```

$$false$$

```
> VerifySubsystem(M, Delta, "symmetric");
```

$$false$$

```
> VerifySubsystem(M, Delta, "pisystem");
```

$$true$$

Therefore M is a π-system and we can show that $[M] =$ BracketN.

```
> BracketM := SubsystemBracket(M, Delta);
```

$$BracketM := \left[\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right]$$

9.2 Listing All Maximal Regular Semisimple Subalgebras

If this section we will use the programs created to list the maximal regular semisimple subalgebras of a given Lie algebra.

Example 9.2.1: Maximal Regular Semisimple Subalgebras using Maple

Start by initializing the Lie algebra F_4.

```
> DGEnvironment[LieAlgebra]("f(4, [28,24])", f4);
```

$$Lie\,algebra : f4$$

Then find the Cartan subalgebra

```
> CSA := CartanSubalgebra(f4);
```

$$CSA := [e1, e2, e3, e4]$$

From the Cartan subalgebra we can find the root space decomposition, the positive roots, and the simple roots.

```
> RSD := RootSpaceDecomposition(CSA):
> PR := PositiveRoots(RSD):
> SR := SimpleRoots(SR);
```

$$SR := \left[\begin{bmatrix} 0 \\ 0 \\ 1 \\ -2 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -2 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ -2 \\ 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}\right]$$

We can also list all 48 roots of F_4.

```
> Delta := simplify([op(PR), - op(PR)]):
```

We can find the highest root of F_4 and the corresponding lowest root

```
> HR := Vector([2,-1,0,0]);
```

$$HR := \begin{bmatrix} 2 \\ -1 \\ 0 \\ 0 \end{bmatrix}$$

```
> LR := simplify(-HR);
```

$$LR := \begin{bmatrix} -2 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

We can find the n_i that satisfy equation (10.1)

```
> 2*SR[1] + 3*SR[2] + 2*SR[3] + 4*SR[4]
```

$$\begin{bmatrix} 2 \\ -1 \\ 0 \\ 0 \end{bmatrix}$$

```
> simplify(-LR)
```

$$\begin{bmatrix} 2 \\ -1 \\ 0 \\ 0 \end{bmatrix}$$

Therefore the n_i are $[2, 3, 2, 4]$.

Knowing what the lowest root is we are able to create the extended root system

```
> ExtSR := [op(SR), LR];
```

$$ExtSr := \left[\begin{bmatrix} 0 \\ 0 \\ 1 \\ -2 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -2 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ -2 \\ 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -2 \\ 1 \\ 0 \\ 0 \end{bmatrix}\right]$$

Then to find a maximal semisimple regular subalgebra we remove one root from the extended simple roots. For this example we choose to remove the root SR[3] because $n_3 = 2$ which is prime.

```
> M := [SR[1], SR[2], SR[4], LR];
```

$$M := \left[\begin{bmatrix} 0 \\ 0 \\ 1 \\ -2 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -2 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -2 \\ 1 \\ 0 \\ 0 \end{bmatrix}\right]$$

In order to find the desired subalgebra first determine $[M]$ which is a closed and symmetric subsystem

```
> Bm := SubsystemBracket(M, Delta):
> VerifySubsystem(Bm, Delta, "closed");
```

$$true$$

```
> VerifySubsystem(Bm, Delta, "symmetric");
```

$$true$$

Then we can use the program ClosedSubsystemToRegularSubalgebra discussed in Chapter 7 to find the corresponding subalgebra.

```
> f := ClosedSubsystemToRegularSubalgebra(Bm, Delta, RSD);
```

$$f := [-e1, -e1 - 2e2 - e3, -2e1 - 4e2 - 2e3 - e4, e3 + e4, e50, e32, e25, e29, e26, e8, e49, e5, e31, e23, e7, e47, e35, e24, e11, e48, e21, e45, e18, e42]$$

We can check to see if this is a regular subalgebra

```
> Query(f, "Subalgebra");
```

$$true$$

```
> VerifyRegular(f, CSA);
```

$$true$$

We now initialize this Lie algebra to verify that is in in fact semisimple.

```
> LD := LieAlgebraData(f, alg):
> DGEnvironment[LieAlgebra](LD);
```

$$Lie\,algebra : alg$$

```
> Query(alg, "Semisimple");
```

$$true$$

We are able to find the simple roots for this Lie algebra

```
> CSA1 := CartanSubalgebra(alg):
```

```
> RSD1 := RootSpaceDecomposition(CSA1):
> PR1 := PositiveRoots(RSD1);
> SR1 := SimpleRoots(PR1);
```

$$SR1 := \left[\begin{bmatrix} 0 \\ 0 \\ 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} \right]$$

To see which semisimple algebra our subalgebra is isomorphic to we find the Cartan matrix

```
> CM := CartanMatrix(SR1, RSD1);
```

$$CM := \begin{bmatrix} 2 & 0 & 0 & -2 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}$$

Then use the program CartanMatrixToStandardForm

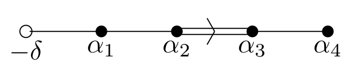
```
> CartanMatrixToStandardForm(CM, SR1);
```

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0-2 & 2 & \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \text{"}A1C3\text{"}$$

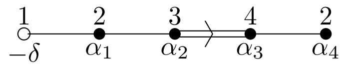
Therefore our subalgebra is isomorphic to the Lie algebra $sl(2) \oplus sp(6)$.

Example 9.2.2: Maximal Regular Semisimple Subalgebras using Extended Dynkin Diagrams

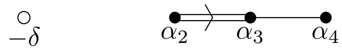
We will continue to work with the Lie algebra F_4. Recall from Section 2.9 that F_4 has the following Extended Dynkin Diagram:



We can add the n_i values discussed above to the extended Dynkin diagram



Because $n_1 = 2$ which is prime we can remove the root α_1 from this extended root system we are left with the following Dynkin diagram:

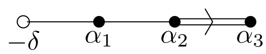


which is the Dynkin diagram for the semisimple Lie algebra of type $A_1 \oplus C_3$. Therefore this subalgebra is isomorphic to $sl(2) \oplus sp(6)$ as was shown in Example 9.1.1.

We are also able to remove the root α_2 because $n_2 = 3$ which is prime. Removing this root gives the following Dynkin diagram:

which is isomorphic to $sl(3) \oplus sl(3)$.

The last n_i which is prime is $n_4 = 2$; removing this root gives the following Dynkin diagram

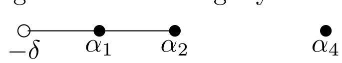


which is isomorphic to $so(9)$.

Thus by Theorem 9.1 we can conclude that every maximal regular semisimple subalgebra of F_4 will be isomorphic to one of the following Lie algebras:

- $sl(2) \oplus sp(6)$
- $sl(3) \oplus sl(3)$
- $so(9)$

Now we can ask what will happen if we remove the root α_3. Note that $n_3 = 4$ which is not a prime number. Removing this root gives the following Dynkin diagram



which is isomorphic to $sl(4) \oplus sl(2)$. It is obvious that this subalgebra will be semisimple. It will also be a regular subalgebra. The reason that this subalgebra is excluded by Theorem 9.1 is because it is not a maximal subalgebra. We can explicitly show that there is a subalgebra of type $A_3 \oplus A_1$ of B_4. Start by creating $so(5,4)$ as its own Lie algebra.

```
> DGEnvironment[LieAlgebra]("so(5,4)", so54);
```

$$\mathit{Lie\ algebra} : so54$$

Now we can find a record that contains various properties about this Lie algebra.

```
> Prop := SimpleLieAlgebraProperties(so54):
```

Now we can use this information to find Δ which is the list of all roots of $so(5,4)$.

```
> CSA := Prop:-CartanSubalgebra;
```

$$CSA := [e1, e6, e11, e16]$$

```
> RSD := Prop:-RootSpaceDecomposition:
> PR := Prop:-PositiveRoots:
> SR := Prop:-SimpleRoots;
```

$$SR := \left[\begin{bmatrix}1\\-1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\-1\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right]$$

```
> Delta := simplify([op(PR), op(-PR)]):
```

Then we take the lowest root and create the extended root system

```
> LR := Delta[-1];
> ER := [LR, op(SR)];
```

$$ER := \left[\begin{bmatrix}-1\\-1\\0\\0\end{bmatrix}, \begin{bmatrix}1\\-1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\-1\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right]$$

Now consider the following π-system which is a subsystem of the extended root system

```
> M := [ER[1], ER[3], ER[4], ER[5]]
```

$$M := \left[\begin{bmatrix}-1\\-1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\-1\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right]$$

We then find $[M]$ and the corresponding semisimple regular subalgebra $\mathfrak{f}(\mathfrak{t},[M])$

```
> BracketM := SubspaceBracket(M, Delta):
> f := ClosedSubsystemToRegularSubalebra(BracketM, Delta, RSD);
```

$$f := [-e1-e6, -e1-e11, -e1+e11, -2e16, e2, e7, e32, e23, e5, e10, e36, e17, e3, e24, e9, e18, e26, e20]$$

We now want to show that this subalgebra of $so(5,4)$ is of type $A3 \oplus A1$. Start by initializing this as its own Lie algebra.

```
> Ld := LieAlgebraData(f, alg);
> DGEnvironment[LieAlgebra](Ld);
```

$$\textit{Lie algebra} : alg$$

Then find the simple roots of this Lie algebra

```
> CSA1 := CartanSubalgebra(alg);
```

$$CSA1 := [e1, e2, e3, e4]$$

```
> RSD1 := RootSpaceDecomposition(CSA1):
> PR1 := PositiveRoots(RSD1):
> SR1 := SimpleRoots(PR1);
```

$$SR1 := \left[\begin{bmatrix}0\\1\\1\\0\end{bmatrix}, \begin{bmatrix}1\\-1\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\2\end{bmatrix}, \begin{bmatrix}1\\1\\-1\\0\end{bmatrix}\right]$$

Next find the Cartan matrix

```
> CM := CartanMatrix(SR1, RSD1);
```

$$CM := \begin{bmatrix}2 & -1 & 0 & -1\\ -1 & 2 & 0 & 0\\ 0 & 0 & 2 & 0\\ -1 & 0 & 0 & 2\end{bmatrix}$$

Then we can see what type of semisimple subalgebra we have

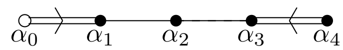
```
> CartanMatrixToStandardForm(CM, SR1);
```

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{"}A1A3\text{"}$$

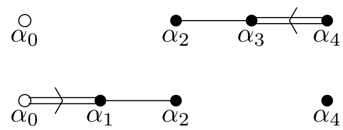
Therefore $A_1 \oplus A_3 \subset B_4 \subset F_4$. Thus $sl(2) \oplus sl(4)$ is not a maximal regular subalgebra of F_4.

Example 9.2.3: Conjugate Maximal Closed Regular Subalgebras and the Weyl Group

For this example consider the simple Lie algebra C_4. Recall from Section 2.9 that the extended Dynkin diagram for this Lie algebra is



where for the lowest root we have $-\alpha_0 = 2\alpha_1 + 2\alpha_2 + 2\alpha_3 + \alpha_4$. By Theorem 9.1 we know that the subalgebra corresponding to $[\tilde{\Delta}^0 \backslash \{\alpha_i\}]$ where α_i is one of the roots α_1, α_2 or α_3 will be a maximal semisimple regular subalgebra. However note that removing either α_1 or α_3 from the extended Dynkin diagram of C_4 will result in the Dynkin diagram of $A_1 \oplus C_3$.



Therefore the two subalgebras are conjugate. Then as discussed in Section 7.3 because the two subalgebras are conjugate there will be an element ω of the Weyl group such that $\omega(\tilde{\Delta}^0 \backslash \{\alpha_1\}) = \tilde{\Delta}^0 \backslash \{\alpha_3\}$. This can be explicitly shown in Maple.

Start by creating the Lie algebra $sp(8)$ and find the record containing information about this Lie algebra.

```
> DGEnvironment[LieAlgebra]("sp(8)", sp8);
```

Lie algebra : sp8

```
> Prop := SimpleLieAlgebraProperties(sp8):
```

From this record we can find the Cartan subalgebra, the list of all roots Δ, and the simple roots.

```
> CSA := Prop:-CartanSubalgebra;
```

$$CSA := [e7, e11, e14, e16]$$

```
> RSD := Prop:-RootSpaceDecomposition:
> PR := Prop:-PositiveRoots:
> Delta := simplify([op(PR), - op(PR)]):
> SR := Prop:-SimpleRoots;
```

$$SR := \left[\begin{bmatrix} I \\ -I \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ I \\ -I \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ -I \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2I \end{bmatrix} \right]$$

We can also find the highest root

```
> HR := Vector([2*I, 0, 0, 0]);
```

$$\begin{bmatrix} 2I \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Therefore the extended root system $[\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4]$ is

```
> ER := [-HR, op(SR)];
```

$$ER := \left[\begin{bmatrix} -2I \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} I \\ -I \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ I \\ -I \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ I \\ -I \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2I \end{bmatrix} \right]$$

Removing α_1 from the extended root system gives the following π-system

```
> SR1 := [-HR, SR[2], SR[3], SR[4]];
```

$$SR1 := \left[\begin{bmatrix}-2I\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\I\\-I\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\-I\end{bmatrix}, \begin{bmatrix}0\\0\\0\\2I\end{bmatrix}\right]$$

Also consider the π-system we get by removing α_3 from the extended root system

```
> SR2 := [-HR, SR[1], SR[2], SR[4]];
```

$$SR2 := \left[\begin{bmatrix}-2I\\0\\0\\0\end{bmatrix}, \begin{bmatrix}I\\-I\\0\\0\end{bmatrix}, \begin{bmatrix}0\\I\\-I\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\2I\end{bmatrix}\right]$$

We claim that there is a Weyl group element that will map SR1 to SR2 which we can find using the command EquivalenceOfSubsystems

```
> omega := EquivalenceOfSubsystems(SR1, SR2, Prop);
```

$$\omega := \begin{bmatrix}0 & 0 & 0 & -1\\0 & 0 & -1 & 0\\0 & -1 & 0 & 0\\-1 & 0 & 0 & 1\end{bmatrix}$$

Now apply the Weyl group element to SR1

```
> seq(omega.x, x in SR1);
```

$$\left[\begin{bmatrix}-2I\\0\\0\\0\end{bmatrix}, \begin{bmatrix}I\\-I\\0\\0\end{bmatrix}, \begin{bmatrix}0\\I\\-I\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\2I\end{bmatrix}\right]$$

Thus we see that ωSR1 = SR2. Then because the two π-systems are related by a Weyl group element we claim that the corresponding subalgebras will be conjugate through a Lie algebra automorphism.

Start by finding $[SR1]$

```
> SR1c := ClosedRootSystem(SR1, Delta):
> BracketSR1 := SymmetricRootSystem(SR1c):
```

Then find the subalgebra $\mathfrak{f}_1 = \mathfrak{f}(\mathfrak{t}, [SR1])$.

```
> f1 := ClosedSubsystemToRegularSubalgebra(BracketSR1, Delta, RSD);
```

$$f1 := [I\,e16, I\,e14, I\,e11, I\,e7, e17 + I\,e27, e4 + I\,e12, e6 + I\,e15, e26 - I\,e36, e5 + I\,e13, e25 - I\,e35, e23 - I\,e33, e24 - I\,e34, e22 - I\,e32, e21 - I\,e31, e17 - I\,e27, e4 - I\,e12, e6 - I\,e15, e26 + I\,e36, e5 - I\,e13, e25 + I\,e35, e23 + I\,e33, e24 + I\,e34, e22 + I\,e32, e21 + I\,e31]$$

Next find $[SR2]$

```
> SR2c := ClosedRootSystem(SR2, Delta):
> BracketSR2 := SymmetricRootSystem(SR2c):
```

Then find the subalgebra $\mathfrak{f}_2 = \mathfrak{f}(\mathfrak{t}, [SR2])$.

```
> f2 := ClosedSubsystemToRegularSubalgebra(BracketSR1, Delta, RSD);
```

$$f2 := [I\,e16, I\,e14, I\,e11, I\,e7, e17 + I\,e27, e1 + I\,e8, e4 + I\,e12, e26 - I\,e36, e18 + I\,e28, e2 + I\,e9, e19 + I\,e29, e21 + I\,e31, e22 + I\,e32, e24 + I\,e34, e17 - I\,e27, e1 - I\,e8, e4 - I\,e12, e26 + I\,e36, e18 - I\,e28, e2 - I\,e9, e19 - I\,e29, e21 - I\,e31, e22 - I\,e32, e24 - I\,e34]$$

Recall from Section 2.9 that given a Weyl group element we can find a Lie algebra automorphism which fixes the Cartan subalgebra.

```
> Phi := WeylGroupElelmentToAutomorphism(omega, Prop);
```

$$\Phi := e1 \to e6, e2 \to -e5, e3 \to e3, e4 \to e4, e5 \to -e2, e6 \to e1, e7 \to -e16, e8 \to e15, e9 \to -e13, e10 \to e10, e11 \to -e14, e12 \to e12, e13 \to -e9, e14 \to -e11, e15 \to e8, e16 \to -e7, e17 \to \left(\tfrac{65537}{512}\right) e26 + \left(\tfrac{65535I}{512}\right) e36, e18 \to -\left(\tfrac{65537}{512}\right) e25 - \left(\tfrac{65535I}{512}\right) e35, e19 \to \left(\tfrac{65537}{512}\right) e23 + \left(\tfrac{65535I}{512}\right) e33, e20 \to -\left(\tfrac{65537}{512}\right) e20 - \left(\tfrac{65535I}{512}\right) e30, e21 \to \left(\tfrac{65537}{512}\right) e24 + \left(\tfrac{65535I}{512}\right) e34, e22 \to -\left(\tfrac{65537}{512}\right) e22 - \left(\tfrac{65535I}{512}\right) e32, e23 \to \left(\tfrac{65537}{512}\right) e19 + \left(\tfrac{65535I}{512}\right) e29, e24 \to \left(\tfrac{65537}{512}\right) e21 + \left(\tfrac{65535I}{512}\right) e3, e25 \to -\left(\tfrac{65537}{512}\right) e18 - \left(\tfrac{65535I}{512}\right) e28, e26 \to \left(\tfrac{65537}{512}\right) e17 + \left(\tfrac{65535I}{512}\right) e27, e27 \to \left(\tfrac{65535I}{512}\right) e26 - \left(\tfrac{65537}{512}\right) e36, e28 \to -\left(\tfrac{65535I}{512}\right) e25 + \left(\tfrac{65537}{512}\right) e35, e29 \to$$

$$\left(\frac{65535I}{512}\right)e23-\left(\frac{65537}{512}\right)e33, e30\to-\left(\frac{65535I}{512}\right)e20+\left(\frac{65537}{512}\right)e30, e31\to \left(\frac{65535I}{512}\right)e24-\left(\frac{65537}{512}\right)e34, e32\to-\left(\frac{65535I}{512}\right)e22+\left(\frac{65537}{512}\right)e32, e33\to\left(\frac{65535I}{512}\right)e19-\left(\frac{65537}{512}\right)e29, e34\to \left(\frac{65535I}{512}\right)e21-\left(\frac{65537}{512}\right)e31, e35\to-\left(\frac{65535I}{512}\right)e18+\left(\frac{65537}{512}\right)e28, e36\to\left(\frac{65535I}{512}\right)e17-\left(\frac{65537}{512}\right)e27$$

We can then apply Φ to $\mathfrak{f}_1$

```
> f3 := ApplyLinearTransformation(Phi, f1);
```

$$f3 :=$$

$$[-Ie7, -Ie11, -Ie14, -Ie16, \left(\frac{1}{256}\right)e26-\left(\frac{I}{256}\right)e36, e4+Ie12, e1+Ie8, 256e17+(256I)\,e27, -e2-Ie9, -256e18-(256I)e28, 256e19+(256I)e29, 256e21+(256I)e31, -256e22-(256I)*e32, 256e24+(256I)e34, 256e26+(256I)e36, e4-Ie12, e1-Ie8, \left(\frac{1}{256}\right)e17-\left(\frac{I}{256}\right)e27, -e2+Ie9, -\left(\frac{1}{256}\right)e18+\left(\frac{I}{256}\right)e28, \left(\frac{1}{256}\right)e19-\left(\frac{I}{256}\right)e29, \left(\frac{1}{256}\right)e21-\left(\frac{I}{256}\right)e31, -\left(\frac{1}{256}\right)e22+\left(\frac{I}{256}\right)e32, \left(\frac{1}{256}\right)e24-\left(\frac{I}{256}\right)e34]$$

Which we can show is the same $\mathfrak{f}_2$

```
> GetComponents(f3, f2, trueorfalse = "on");
```

$$true$$

10 Maximal Reductive (Non-Semisimple) Subalgebras

Let $\mathfrak{g}$ be a complex semisimple Lie algebra with root space Δ. Recall from Theorem 9.1 that if the π-system $M = \Delta^0 \backslash \{\alpha_i\}$ where $i > 0$ and $n_i = 1$ then $\Gamma = [M]$ is a maximal closed symmetric subsystem in Δ with rank $r - 1$. The corresponding subalgebras $\mathfrak{f} = \mathfrak{f}(\mathfrak{h}, [M])$ will be the maximal reductive non-semisimple subalgebras

We can use programs created in above sections to verify the results given in Table 6 of the book *Lie Groups and Lie Algebras III* by Onishchik and Vinberg, [23].

10.1 Listing All Maximal Non-Semisimple Reductive Subalgebras

In this section we will list all the maximal non-semisimple reductive subalgebras of a complex semisimple Lie algebra which as a result of Theorem 9.1 is equivalent to the problem of listing all maximal closed symmetric subsystems of the form $[\tilde{\Delta}^0 \backslash \{\alpha_0, \alpha_i\}] = [\Delta^0 \backslash \{\alpha_i\}]$ where $i > 0$ and $n_i = 1$ which are not related by a Weyl group element.

Example 10.1.1: Finding a Maximal Non-Semisimple Reductive Subalgebra

For this example we consider the Lie algebra $sl(7)$. Start by initializing the Lie algebra and finding a record containing various properties of $sl(7)$.

```
> DGEnvironment[LieAlgebra]("sl(7)", sl7);
```

$$Lie\ algebra : sl7$$

From this record we can find the Cartan subalgebra, the root space decomposition, and the simple roots.

```
> RSD := Prop:-RootSpaceDecomposition:
> CSA := Prop:-CartanSubalgebra;
```

$$CSA := e1, e2, e3, e4, e5, e6]$$

```
> SR := Prop:-SimpleRoots;
```

$$SR := \left[\begin{bmatrix}1\\-1\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\-1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\-1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\\1\\-1\end{bmatrix}, \begin{bmatrix}1\\1\\1\\1\\1\\2\end{bmatrix} 0\right]$$

We are also able to find the list Δ of all the roots of $sl(7)$.

```
> PR := Prop:-PositiveRoots:
> Delta := simplify([op(PR), -op(PR)]):
```

Then find the highest root

```
> HR := Delta[21];
```

$$HR := \begin{bmatrix}2\\1\\1\\1\\1\\1\end{bmatrix}$$

Next find the n_i's that satisfy Equation 9.1.

```
> GetComponents(HR, SR);
```

$$[1, 1, 1, 1, 1, 1]$$

Because all of the n_i's are 1, by Theorem 9.1 we can remove any root from the simple root system and the corresponding subalgebra will be a maximal non-semisimple reductive subalgebra.

For this example we choose to remove the root α_3, giving the following π-system

```
> M := [SR[1], SR[2], SR[4], SR[5], SR[6]];
```

$$M := \left[\begin{bmatrix}1\\-1\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\-1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\\1\\-1\end{bmatrix}, \begin{bmatrix}1\\1\\1\\1\\1\\2\end{bmatrix}\right]$$

Next find the closed and symmetric subsystem $[M]$

```
> BracketM := SubsystemBracket(M, Delta):
> VerifySubsystem(BracketM, Delta, "closed");
```

$$true$$

```
> VerifySubsystem(BracketM, Delta, "symmetric");
```

$$true$$

Now because we want a maximal reductive subalgebra we find $\mathfrak{f} = \mathfrak{f}(\mathfrak{h}, [M])$ where $\mathfrak{h}$ is the Cartan subalgebra.

```
> f := ClosedSubsystemToRegularSubalgebra(BracketM, Delta, RSD, CartanSubalgebra =
CSA);
```

$$f :=$$

$$[e1, e2, e3, e4, e4, e6, e6, e14, e28, e35, e42, e13, e20, e34, e41, e48, e8, e29.e36, e19, e40, e47, e30, e46]$$

We can test to see if this is a regular subalgebra

```
> Query(f, "Subalgebra");
```

$$true$$

```
> VerifyRegular(f, CSA);
```

$$true$$

Initialize this subalgebra as its own Lie algebra

```
> Ld := LieAlgebraData(f, alg):
```

```
> DGEnvironment[LieAlgebra](Ld);
```

We can test to see if it is reductive and non-semisimple.

```
> Query(alg, "Semisimple");
```

$$false$$

```
> VerifyReductive(alg);
```

$$true$$

Now that we have a regular non-semisimple reductive subalgebra, we would like to know which subalgebra this is, we start by finding the Levi decomposition of $\mathfrak{f}$.

```
> LeDc := LeviDecomposition(alg);
```

$$LeDc := \left[\left[e1 + e2 + e3 - \left(\tfrac{3}{4}\right) e4 - \left(\tfrac{3}{4}\right) e5 - \left(\tfrac{3}{4}\right) e6\right], [e1 - e3, e2 - e3, e4, e5, e6, e7, e8,\right.$$

$$\left. e9, e10, e11, e12, e13, e14, e15, e16, e17, e18, e19, e20, e21, e22, e23, e24]\right]$$

The reductive part of the subalgebra $\left[e1 + e2 + e3 - \left(\frac{3}{4}\right) e4 - \left(\frac{3}{4}\right) e5 - \left(\frac{3}{4}\right) e6\right]$ is one dimensional and so we can say that it is isomorphic to $\mathbb{C}$.

Now consider the semisimple part of the subalgebra as its own Lie algebra.

```
> Ld2 := LieAlgebraData(LeDc[2], alg2):
> DGEnvironment[LieAlgebra](Ld2);
```

$$Lie\ algebra : alg2$$

For this subalgebra we can find the Cartan subalgebra, positive roots, and simple roots.

```
> CSA2 := CartanSubalgebra(alg2);
```

$$CSA2 := [e1, e2, e3, e4, e5]$$

```
> RSD2 := RootSpaceDecomposition(CSA2):
> PR2 := PositiveRoots(RSD2):
> SR2 := SimpleRoots(PR2);
```

$$SR2 := \left[\begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} \right]$$

From these simple roots we can find the Cartan matrix

```
> CM := CartanMatrix(SR2, RSD2);
```

$$CM := \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

Then we can use the program CartanMatrixToStandardForm to see what are subalgebra is

```
> CartanMatrixToStandardForm(CM, SR2);
```

$$\begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \text{"}A2A3\text{"}$$

So the semisimple portion of our subalgebra is isomorphic to $sl(3) \oplus sl(4)$.

Therefore the maximal non-semisimple reductive subalgebra we have is isomorphic to $sl(3) \oplus sl(4) \oplus \mathbb{C}$.

11 Parabolic Subalgebras

In this paper we have discussed various types of regular subalgebras of a Lie algebra. In this section we will discuss an additional type of regular subalgebra called a **Parabolic subalgebra.** To study the parabolic subalgebras of a Lie algebra we first look at a particular case called the Borel Subalgebra.

11.1 Borel Subalgebra

Let $\mathfrak{g}$ be a semisimple Lie algebra with Cartan subalgebra $\mathfrak{h}$. A subalgebra $\mathfrak{b}$ of $\mathfrak{g}$ is defined to be a **Borel Subalgebra** if $\mathfrak{b}$ is a maximal solvable subalgebra. Recall Theorem 7.1, then $\mathfrak{f}(\mathfrak{h}, \Delta^+)$ is a Borel subalgebra.

Let Δ be the root space of $\mathfrak{g}$ and let Δ^+ be a choice of positive roots with simple roots Δ^0. The subalgebra

$$\mathfrak{b} = \mathfrak{h} \oplus \bigoplus_{\alpha \in \Delta^+} \mathfrak{g}_\alpha$$

is called the **Standard Borel Subalgebra**.

11.2 Parabolic Subalgebra

A subalgebra $\mathfrak{p}$ of a Lie algebra $\mathfrak{g}$ is said to be **Parabolic** if $\mathfrak{p}$ contains a Borel subalgebra. A parabolic subalgebra containing a standard Borel subalgebra is called a **Standard Parabolic Subalgebra.**

Let N be a subsystem of the simple roots Δ^0. Define the subsystem $\Gamma_N = [N] \cup \Delta^+$ which is a closed subsystem of Δ containing Δ^+. Then the corresponding regular subalgebra $\mathfrak{f}(\mathfrak{t}, \Gamma_N)$ is a parabolic subalgebra. Using this we can write a standard parabolic subalgebra in the form

$$\mathfrak{p} = \mathfrak{h} \oplus \bigoplus_{\alpha \in \Gamma_N} \mathfrak{g}_\alpha = \mathfrak{h} \oplus \bigoplus_{\alpha \in \Delta^+} \mathfrak{g}_\alpha \oplus \bigoplus_{\alpha \in \Gamma_N / \Delta^+} \mathfrak{g}_\alpha$$

There are already two programs in the Differential Geometry package of Maple that deal with parabolic subalgebras. The program ParabolicSubalgebra will find the parabolic subalgebra defined by a set of simple roots and the program ParabolicSubalgebraRoots finds the simple roots which generate a parabolic subalgebra.

In the book *Lie Groups and Lie Algebras* by Onishchik and Vinberg [23] we read the following theorem:

Theorem 11.1. *Any non-semisimple maximal subalgebra of a complex semisimple Lie algebra is parabolic.*

A result of this theorem is that the subalgebras from Chapter 10 are not maximal among all non-semisimple subalgebras because these subalgebras are not parabolic. However any maximal reductive non-semisimple subalgebra can be contained in a parabolic subalgebra. Consider a maximal reductive non-semisimple subalgebra $\mathfrak{f} = \mathfrak{f}(\mathfrak{h}, [\Delta^0 \backslash \{\alpha_i\}])$ as discussed in Chapter 11. Then $\mathfrak{f} \subset \mathfrak{p}$ where

$$\mathfrak{p} = \mathfrak{f}(\mathfrak{h}, \Gamma_{\Delta^0 \backslash \{\alpha_i\}})$$

Example 11.2.1: Continuing Example 10.1.1

Recall from Example 10.1.1 that we were working with the Lie algebra $sl(7)$ and have the following subalgebra

```
> f;
```

$$f :=$$

$$[e1, e2, e3, e4, e4, e6, e6, e14, e28, e35, e42, e13, e20, e34, e41, e48, e8, e29.e36, e19, e40, e47, e30, e46]$$

To find the parabolic subalgebra that contains this subalgebra we need to find the set of roots $[M] \cup \Delta^+$. However to do this in Maple we need to convert our list of vectors $[M]$ and PR into a set of lists.

```
> MList := map(convert, BracketM, list):
> MListSet := convert(MList, set):
> PRList := map(convert, PR, list):
> PRListSet := convert(PRList, set):
```

Now we take the union of these two sets

```
> Int1 := MListSet union PRListSet:
```

Which we can then map back to vectors

```
> UnionVec := map(Vector, convert(Int1, list)):
```

Then find the subalgebra $\mathfrak{p} = \mathfrak{f}(\mathfrak{h}, UnionVec)$

```
> p := ClosedSubsystemToRegularSubalgebra(UnionVec, Delta, RSD, CartanSubalgebra =
CSA);
```

$$p := [e1, e2, e3, e4, e5, e6, e46, e47, e48, e19, e13, e20, e40, e34, e41, e35, e28, e29, e21, e22, e23, e14,$$
$$e15, e16, e17, e7, e8, e9, e10, e11, e42, e36, e30, e24, e18, e12]$$

We can test that this is a parabolic subalgebra

```
> Query(p, "parapolic");
```

$$true$$

```
> Query(p, "Subalgebra");
```

$$true$$

We can see that the subalgebra $\mathfrak{f}$ is a subalgebra of $\mathfrak{p}$.

```
> GetComponents(f, p, trueorfalse = "on");
```

$$true$$

12 Classifying all Regular Semisimple Subalgebras of a Semisimple Lie Algebra

Let $\mathfrak{g}$ be a semisimple Lie algebra with Cartan subalgebra $\mathfrak{h}$ and root space Δ. Let Δ^0 be a choice of simple roots and α_0 the lowest root. Then recall the extended root system $\tilde{\Delta}^0 = \Delta^0 \cup \{\alpha_0\}$. Any subset M of the extended root system will be a π-system.

Recall that a regular semisimple subalgebra can be written in the form $\mathfrak{f} = \mathfrak{f}(\mathfrak{t}, [M])$ where M is a π-system. The classification of regular semisimple subalgebras is equivalent to listing all the π-systems not related by a Weyl group element.

The following program has been created to list all possible non-conjugate root subsystems:

- **AllRootSystems:** This program takes an extended root system and a record of properties about the Lie algebra and returns a list of all root subsystems which are not related by any Weyl group element. To do this the program first creates a list of all possible root subsystems and well as listing all Weyl group elements. Then pick the first root subsystem. Remove all other root subsystems which are related by any Weyl group element. Now pick the next root subsystem remaining and repeat the process. The end result is a list of root subsystems not conjugate under the Weyl group.

We can use this program (as well as those discussed in above sections) to list all the possible regular semisimple subalgebras of a given semisimple Lie algebra. We can also verify the result of Table IV in the paper, *The Classification of Semisimple Lie Algebras of Simple Lie Algebras* by Lorente and Gruber.

Example 12.1.1 - Listing All Regular Semisimple Subalgebras of B_3

Begin by initializing $so(4,3)$ as a Lie algebra.

```
> DGEnvironment[LieAlgebra]("so(4,3)", so43);
```

$$\textit{Lie algebra} : so43$$

Then create a list of properties for this Lie algebra

```
> Prop := SimpleLieAlgebraProperties(so43):
```

From this list of properties we are able to create a list Δ of all the roots of $so(4,3)$

```
> RSD := Prop:-RootSpaceDecomposition:
> PR := PositiveRoots:
> Delta := simplify([op(PR), op(-PR)]):
```

Next we want to find the extended roots.

```
> SR := Prop:-SimpleRoots
```

$$SR := \left[\begin{bmatrix}1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\-1\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}\right]$$

```
> LR := Delta[-1];
```

$$LR := \begin{bmatrix}-1\\-1\\0\end{bmatrix}$$

```
> ER := [LR, op(SR)];
```

$$ER := \left[\begin{bmatrix}-1\\-1\\0\end{bmatrix}, \begin{bmatrix}1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\-1\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}\right]$$

Next find a list of all π-systems not related by a Weyl group element.

```
> Gamma := AllRootSystems(ER, Prop);
```

$$\Gamma := \left[\left[\begin{bmatrix}1\\-1\\0\end{bmatrix}\right], \left[\begin{bmatrix}0\\0\\1\end{bmatrix}\right], \left[\begin{bmatrix}1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\-1\end{bmatrix}\right], \left[\begin{bmatrix}1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}\right], \left[\begin{bmatrix}1\\-1\\0\end{bmatrix}, \begin{bmatrix}-1\\-1\\0\end{bmatrix}\right],\right.$$

$$\left.\left[\begin{bmatrix}0\\1\\-1\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}\right], \left[\begin{bmatrix}1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\-1\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}\right], \left[\begin{bmatrix}1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\-1\end{bmatrix}, \begin{bmatrix}-1\\-1\\0\end{bmatrix}\right], \left[\begin{bmatrix}1\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}-1\\-1\\0\end{bmatrix}\right]\right]$$

Note that there are 9 root systems returned so we claim that there are 9 non-conjugate regular semisimple subalgebras of B_3. We will consider each of these root systems and find which subalgebra they correspond to.

Consider the first root system Γ_1

```
> Gamma1 := Gamma[1];
```

$$\Gamma 1 := \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

Then we can find the closed and symmetric subsystem $[\Gamma_1]$

```
> Bracket1 := SubsystemBracket(Gamma1, Delta);
```

$$Bracket1 := \left[\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \right]$$

Next find the corresponding subalgebra $\mathfrak{f}_1 = \mathfrak{f}(\mathfrak{t}, [\Gamma_1])$.

```
> f1 := ClosedSubsystemToRegularSubalgebra(Bracket1, Delta, RSD);
```

$$f1 := [-e1 + e5, e2, e4]$$

We can look at the structure equation and see that this is the subalgebra A_1

```
> LD1 := LieAlgebraData(f1, alg1);
```

$$LD1 := [e1, e2] = -2e2, [e1, e3] = 2e3, [e2, e3] = -e1$$

Now that we have an A_1 subalgebra we can check its Dynkin index.

```
> DynkinIndex(f1);
```

$$1$$

Therefore this is the subalgebra A_1 with Dynkin index 1.

Now we can move onto the next listed root subsystem Γ_2

```
> Gamma2 := Gamma[2];
```

$$\Gamma2 := \left[\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right]$$

Then we can repeat the process listed above to determine which subalgebra this is

```
> Bracket2 := SubsystemBracket(Gamma2, Delta):
> f2 := ClosedSubsystemToRegularSubalgebra(Bracket2, Delta, RSD);
```

$$f2 := [-2e9, e18, e21]$$

This subalgebra will also be of type A_1. However note that this subalgebra has a different Dynkin index that the one found above.

```
> DynkinIndex(f2);
```

$$2$$

Therefore this is the subalgebra A_1 with a Dynkin index of 2 denoted by A_1^2.

We have now found two non-conjugate subalgebras and will now consider the third root subsystem Γ_3

```
> Gamma3 := Gamma[3];
```

$$\Gamma3 := \left[\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}\right]$$

Find $[\Gamma_3]$ and the corresponding subalgebra $\mathfrak{f}_3 = \mathfrak{f}(\mathfrak{t}, [\Gamma_3])$

```
> Bracket3 := SubsystemBracket(Gamma3, Delta):
> f3 := ClosedSubsystemToRegularSubalgebra(Bracket3, Delta, RSD);
```

$$f3 := [-e1 + e9, -e1 + e5, e2, e6, e4, e8, e3, e7]$$

To determine which semisimple subalgebra this is we initialize this as its own Lie algebra and find the Cartan matrix in standard form.

```
> LD3 := LieAlgebraData(f3, alg3):
> DGEnvironment[LieAlgebra](LD3, vectorlabels = [E]):
> CSA3 := CartanSubalgebra(alg3):
> RSD3 := RootSpaceDecomposition(CSA3):
> PR3 := PositiveRoots(RSD3):
> SR3 := SimpleRoots(PR3):
> CM3 := CartanMatrix(SR3, RSD3):
> CartanMatrixToStandardForm(CM3, SR3);
```

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \text{"}A2\text{"}$$

We can check the Dynkin index for this simple subalgebra

```
> DynkinIndex(f3);
```

$$1$$

Thus this is the subalgebra A_2.

We can now consider the fourth listed root system Γ_4

```
> Gamma4 := Gamma[4];
```

$$\Gamma 4 := \left[\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} \right], \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Then find the subalgebra $\mathfrak{f}_4 = \mathfrak{f}(\mathfrak{t}, [\Gamma_4])$

```
> Bracket4 := SubsystemBracket(Gamma4, Delta):
> f4 := ClosedSubsystemToRegularSubalgebra(Bracket4, Delta, RSD);
```

$$f4 :=] - e1 + e5, -2e9, e2, e18, e4, e21]$$

To determine which semisimple subalgebra this is we initialize this as its own Lie algebra and find the Cartan matrix in standard form.

```
> LD4 := LieAlgebraData(f4, alg4):
> DGEnvironment[LieAlgebra](LD4, vectorlabels = [E]):
> CSA4 := CartanSubalgebra(alg4):
> RSD4 := RootSpaceDecomposition(CSA4):
> PR4 := PositiveRoots(RSD4):
> SR4 := SimpleRoots(PR4):
> CM4 := CartanMatrix(SR4, RSD4):
> CartanMatrixToStandardForm(CM4, SR4);
```

$$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, "A1A1"$$

So we have a subalgebra of type $A_1 \oplus A_1$ but we would like to know what the Dynkin index is for each piece. Note the subalgebra can be decomposed as $f4 = TDS1 \oplus TDS2$ where

```
> TDS1 := ThreeDimensionalSubalgebra(evalDG(-e1+e5), "semisimple");
```

$$TSD1 := [-e1 + e5, e4, e2]$$

```
> TDS2 := ThreeDimensionalSubalgebra(evalDG(-2*e9), "semisimple");
```

$$TDS2 := [-2e9, e21, -2e18]$$

and that

```
> DynkinIndex(TDS1);
```

$$1$$

```
> DynkinIndex(TDS2);
```

$$2$$

Therefore this is the subalgebra $A_1 \oplus A_1^2$. So we have already listed 4 non-conjugate subalgebras of B_3.

Continuing in this manner we can show that

$$\mathfrak{f}_5 = \mathfrak{f}(\mathfrak{t}, [\Gamma_5]) = A_1 \oplus A_1$$

$$\mathfrak{f}_6 = \mathfrak{f}(\mathfrak{t}, [\Gamma_6]) = B_2$$

$$\mathfrak{f}_7 = \mathfrak{f}(\mathfrak{t}, [\Gamma_7]) = B_3$$

$$\mathfrak{f}_8 = \mathfrak{f}(\mathfrak{t}, [\Gamma_8]) = A_3$$

$$\mathfrak{f}_9 = \mathfrak{f}(\mathfrak{t}, [\Gamma_9]) = 2A_1 \oplus A_1^2$$

Therefore the regular semisimple proper subalgebras of B_3 are A_1, A_1^2, A_2, $A_1 \oplus A_1^2$, $A_1 \oplus A_1$, B_2, A_3, and $2A_1 \oplus A_1^2$ which corresponds to the regular semisimple subalgebras of B_3 given in Table IV of [18].

13 Summary of Commands

ThreeDimensionalSubalgebra(X, method)

- X is a semisimple or nilpotent element of a Lie algebra
- method is a string "semisimple" or "nilpotent"

For any nilpotent element X this program will return the $sl(2)$ subalgebra containing that element. (See Example 5.1.1) For certain semisimple elements this program will return the $sl(2)$ subalgebra containing X. (See Example 5.2.2)

EpsilonCharacterisics(CSA);

- CSA is a choice of Cartan subalgebra for a given Lie algebra

For a choice of Cartan subalgebra this program will return certain monosemisimple elements. (See Example 5.3.1) By taking linear combinations of these elements and looking at the subalgebras containing them, we are able to list all non-conjugate $sl(2)$ subalgebra.

AllThreeDimensionalSubalgebras(alg, method, CSA)

- alg is a Lie algebra
- method is a string "characteristics" or "subalgebras"
- CSA is a Cartan subalgebra of the Lie algebra alg

For a given Lie algebra and Cartan subalgebra this program will find all the non-conjugate $sl(2)$ subalgebras. These algebras have semisimple elements of the form $X = \sum \alpha_i \epsilon_i$ where the α_is are 0,1, or 2, and the ϵ_is are the epsilon characteristics. If the method is "characteristics" then the output will be a list of what α_is to choose and the epsilon characteristics. If the method is "subalgbras" then the program will return a list of all non-conjugate $sl(2)$ subalgebras. (See Example 5.3.1)

PrincipalSubalgebra(CSA)

- CSA is a choice of Cartan subalgebra for a given Lie algebra

For a choice of Cartan subalgebra this program will return a principal subalgebra, that is the $sl(2)$ subalgebra where the semisimple element is written in the form $X = \sum 2\epsilon_i$, the ϵ_i's are the epsilon characteristics. (See Example 5.4.1)

LieAlgebraExponents(algType, Rank)

- algType is the type of simple Lie algebra A, B, C, D, E, F, or G as a string
- Rank is an integer

For a given simple Lie algebra this program will return the Lie algebra exponents (See Example 6.1.1)

DynkinIndex(subalg)

- subalg is a list of vectors that define a simple subalgebra of a simple Lie algebra

This program takes a simple subalgebra of a simple Lie algebra and returns the Dynkin index. (See Example 4.1.1)

VerifySubsystem(Gamma, Delta, method)

- Delta is the root system of a Lie algebra
- Gamma is a root subsystem of Delta
- method is a string either "closed", "symmetric", or "pisystem"

Given the root system Δ of a Lie algebra $\mathfrak{g}$, this program will test to see if a subsystem Γ of Δ is a closed subsystem , a symmetric subsystem, or a π-system. (See Example 7.1.1 and Example 9.1.1)

ClosedRootSystem(Gamma, Delta)

- Delta is the root system of a Lie algebra
- Gamma is a root subsystem of Delta

This program takes an arbitrary root subsystem Γ of the root system Δ and returns the smallest closed subsystem containing Γ. (See Example 7.1.1)

SymmetricRootSystem(Gamma, Delta)

- Delta is the root system of a Lie algebra
- Gamma is a root subsystem of Delta

This program takes an arbitrary root subsystem Γ of the root system Δ and returns the smallest symmetric subsystem containing Γ. (See Example 7.1.1)

SubsystemBracket(N, Delta)

- Delta is the root system of a Lie algebra

- N is a root subsystem of Delta

This program takes an arbitrary subsystem N of a root system Δ and returns the smallest subsystem containing all integer linear combinations of the roots in N. Using the notation discussed Chapter 10, the output will be $[N]$. (See Example 9.1.1)

MaximalPiSystem(N, Delta)

- Delta is the root system of a Lie algebra
- N is a root subsystem of Delta

This program takes an arbitrary subsystem N of a root system Δ and returns the maximal π-system M contained within N. (See Example 9.1.1)

VerifyRegular(subAlg, CSA)

- subAlg is a list of vectors defining a subalgebra of Lie algebra $\mathfrak{g}$
- CSA is a Cartan subalgebra of $\mathfrak{g}$

Recall that a subalgebra $\mathfrak{f}$ of Lie algebra $\mathfrak{g}$ is a regular subalgebra if $[\mathfrak{h}, \mathfrak{f}] \subset \mathfrak{f}$ for some Cartan subalgebra $\mathfrak{h}$ of $\mathfrak{g}$. Given $\mathfrak{f} \subset \mathfrak{g}$ and $\mathfrak{h}$ this program tests to see if $\mathfrak{f}$ is regular. (See Example 7.2.2)

VerifyReductive(subAlg)

- subAlg is an initialized Lie algebra.

This program calculates the radical of a given Lie algebra and then verifies that every element in the radical is semisimple. (See Example 8.1.1).

ClosedSubsystemToRegularSubalgebra(Gamma, Delta, RSD, CSA)

- Delta is the root system of a Lie algebra
- Gamma is a closed subsystem
- RSD is the root space decomposition
- CSA is the Cartan subalgebra

Given a closed subsystem Γ this program finds the subalgebra $\mathfrak{f}(\mathfrak{t}, \Gamma)$. If CSA is given then $\mathfrak{t} = CSA$. If CSA is left blank then $\mathfrak{t}$ is the subalgebra generated by h_α for $\alpha \in \Gamma \cap (-\Gamma)$. (See

Example 7.2.2)

RegularSubalgebraToRootSystem(subAlg, CSA, RSD)

- subAlg is a list of vectors which form the basis of subalgebra of a given Lie algebra
- CSA is a choice of Cartan subalgebra
- RSD is the root space decomposition

Given a regular subalgebra $\mathfrak{f}(\mathfrak{t}, \Gamma)$, this program will return the closed subsystem Γ.

WeylGroupElementToAutomorphism(elem, Prop)

- elem is an element of the Weyl group
- Prop is a record containing various properties of the Lie algebra including the Cartan subalgebra, Simple Roots, Positive Roots, and Root Space Decomposition

It is a theorem that every element of the Weyl group corresponds to a Lie algebra automorphism which fixes the Cartan subalgebra. This program return this automorphism. (See Example 2.10.1)

AutomorphismToWeylGroupElement(Phi, Prop)

- Phi is a Lie algebra automorphism which fixes the Cartan subalgebra
- Prop is a record containing various properties of the Lie algebra including the Cartan subalgebra, Simple Roots, Positive Roots, and Root Space Decomposition

Given a Lie algebra automorphism $\varphi : \mathfrak{f}_1 \to \mathfrak{f}_2$ this program will find the corresponding Weyl group element ω such that ω maps the simple roots of $\mathfrak{f}_1$ to the simple roots of $\mathfrak{f}_2$. (See Example 2.10.2)

EquivalenceOfSubsystems(Gamma1, Gamma2, Prop)

- Gamma1 is a subsystem of the root system Δ
- Gamma2 is another subsystem of root system Δ
- Prop is a record containing various properties of the Lie algebra including the Cartan subalgebra, Simple Roots, Positive Roots, and Root Space Decomposition

Given two subsystems Γ_1 and Γ_2 we want to know if there is at least one Weyl group element ω such that $\omega\Gamma_1 = \Gamma_2$. If such an ω exists this program will return ω. (See Example 2.10.3)

ExtendedDynkinDiagram(alg)

- alg is a string either "An", "Bn", "Cn", "Dn", "E6", "E7", "E8", "F4", or "G2" where n in an integer

This program will create the extended Dynkin diagram of a simple Lie algebra. (See Example 2.9.2)

AllRootSystems(ExtRts, Prop)

- ExtRts is a list containing a choice of simple roots Δ^0 and the lowest root α_0
- Prop is a record containing various properties of the Lie algebra including the Cartan subalgebra, Simple Roots, Positive Roots, and Root Space Decomposition

This program takes an extended root system and a record of properties about the Lie algebra and returns a list of all root subsystems which are not related by any Weyl group element. To do this the program first creates a list of all possible root subsystems and well as listing all Weyl group elements. Then pick the first root subsystem. Remove all other root subsystems which are related by any Weyl group element. Now pick the next root subsystem remaining and repeat the process. The end result is a list of root subsystems not conjugate under the Weyl group. (See Example 12.1.1)

References

[1] Mark Adler, Pierre van Moerbeke, and Pol Vanhaecke, *Algebraic Integrability, Painleve Geometry and Lie Algebras*, Springer, 2004.

[2] Ana Balibanu, *Geometric Structures of Semisimple Lie Algebras.*

[3] Robert N. Cahn, *Semi-simple Lie Algebras and Their Representations* (1984).

[4] A. Douglas and J. Repka, *Levi decomposable Subalgebras of the Symplectic Algebra C_2*, Journal of Mathematical Physics **56** (2015).

[5] ______, *The Subalgebras of A_2*, Journal of Pure and Applied Algebra **220** (2016).

[6] ______, *Subalgebras of $so(4, \mathbb{C})$*, Communications in Algebra **44** (2016).

[7] Boris Doubrov, *Subalgebras of reductive Lie algebras: classification techniques and applications* (2014).

[8] Willem A. de Graaf, *Constructing semisimple subalgebras of semisimple Lie algebras*, Journal of Algebra **325** (2011).

[9] Heiko Dietrich, Paolo Faccin, and Willem A. de Graaf, *Regular Subalgebras and Nilpotent Orbits of Real Graded Lie Algebras* (2014).

[10] D. Z. Djokovic, J. Patera, P. Winternitz, and H. Zassenhaus, *Normal Forms of Elements of Classical Real and Complex Lie and Jordan Agebras^a*), Journal of Mathematical Physics **24** (1983).

[11] E.B. Dynkin, *The Maximal Subgroups of the Classical Groups*, Tr. Mosk. Mat. Obs. **1** (1952).

[12] ______, *Semisimple Subalgebras of Semisimple Lie Algebras*, Mat. Sb. (N.S.) **30(72)** (1952).

[13] Karin Erdmann and Mark J. Wildon, *Introduction to Lie Algebras*, Springer, 2006.

[14] G. Feldman, T. Fulton, and P. T. Matthews, *The semisimple subalgebras of Lie algebras*, Journal of Mathematical Physics **20** (1984).

[15] William Fulton and Joe Harris, *Representation Theory: A First Course*, Springer-Verlag, 1991.

[16] Victor Kac, *Lecture 18 - Classification of Dynkin Diagrams* (2010).

[17] Bertram Kostant, *The Principal Three-Dimensional Subgroup and the Betti Numbers of a Complex Simple Lie Group*, Americal Journal of Mathematics **81** (1959).

[18] M. Lorente and B. Gruber, *Classification of Semisimple Subalgebras of Simple Lie Algebras*, Journal of Mathematical Physics **13** (1972).

[19] Nicolas Rey-Le Lorier, *Dynkin Diagrams or Everything You Ever Wanted to Know About Lie Algebras (But Were Too Busy to Ask)* (2012).

[20] Evgeny Mayanskiy, *The Subalgebras of G_2* (2016).

[21] A. N. Minchenko, *The Semisimple Subalgebras of Exceptional Lie Algebras*, Trans. Moscow Math Soc. (2006).

[22] Peter J. Olver, *Applications of Lie Groups to Differential Equations*, Springer-Verlag, 1986.

[23] A.L. Onishchik and E.B. Vinberg, *Lie Groups and Lie Algebras III*, Springer-Verlag.

[24] Dmitri I. Panyushev, *The Dynkin Index and $\mathfrak{sl}_2$-Subalgebras of Simple Lie Algebras* (2013).

[25] Marcel Perroud, *The Maximal Solvable Subalgebras of the Real Classical Lie Algebras*, Journal of Mathematical Physics **17** (1976).

[26] J. Patera, P. Winternitz, and H. Zassenhaus, *Continuous Subgroups of the Fundamental Groups of Physics. I. General Method and the Poincare Group**., Journal of Mathematical Physics **16** (1975).

[27] ______, *Continuous Subgroups of the Fundamental Groups of Physics. II. The Similitude Group**., Journal of Mathematical Physics **16** (1975).

[28] J. Patera, R. T. Sharp, P. Winternitz, and H. Zassenhaus, *Continuous Subgroups of the Fundamental Groups of Physics. III. The de Sitter Groups*^{a})., Journal of Mathematical Physics **18** (1977).

[29] ______, *Invariants of Real Low Dimension Lie Algebras**, Journal of Mathematical Physics **17** (1976).

[30] ______, *Subgroups of the Poincare Group and Their Invariants*, Journal of Mathematical Physics **17** (1976).

[31] Miao Qian, Hu Xiao-Rui, and Chen Yong, *ONEOptimal: A Maple Package for Generating One-Dimensional Optimal System of Finite Dimensional Lie Algebra*, Commun. Theor. Phys. **61** (2014).

[32] G. Thompson and Z. Wick, *Subalgebras of gl(3, R)*, Extracta Mathematicae **27** (2012).

[33] David Towers, *Lie algebras all of whose Maximal Subalgebras have Codimension one*, Procedings of the Edinburgh Mathematical Society (1981).

[34] David A. Towers, *Maximal Subalgebras of Lie Algebras Containing Engel Subalgebras*, Journal of Pure and Applied Algebra **216** (2012).

[35] P. Winternitz, *Subalgebras of Real Three- and Four- Dimensional Lie Algebras*, Journal of Mathematical Physics (1977).