

Utah State University

DigitalCommons@USU

---

All Graduate Theses and Dissertations

Graduate Studies

---

12-2017

## Real Simple Lie Algebras: Cartan Subalgebras, Cayley Transforms, and Classification

Hannah M. Lewis  
*Utah State University*

Follow this and additional works at: <https://digitalcommons.usu.edu/etd>



Part of the [Mathematics Commons](#)

---

### Recommended Citation

Lewis, Hannah M., "Real Simple Lie Algebras: Cartan Subalgebras, Cayley Transforms, and Classification" (2017). *All Graduate Theses and Dissertations*. 6900.

<https://digitalcommons.usu.edu/etd/6900>

This Thesis is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations by an authorized administrator of DigitalCommons@USU. For more information, please contact [digitalcommons@usu.edu](mailto:digitalcommons@usu.edu).



REAL SIMPLE LIE ALGEBRAS: CARTAN SUBALGEBRAS,  
CAYLEY TRANSFORMS, AND CLASSIFICATION

by

Hannah M Lewis

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

---

Nathan Geer, Ph.D.  
Committee Member

---

Mark McLellan, Ph.D.  
Vice President for Research and  
Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY  
Logan, Utah

2017

Copyright © Hannah M Lewis 2017

All Rights Reserved

Abstract

Real Simple Lie Algebras:

Cartan Subalgebras, Cayley Transforms, and Classification

by

Hannah M. Lewis, Master of Science

Utah State University, 2017

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

The differential geometry software package in Maple has the necessary tools and commands to automate the classification process for complex simple Lie algebras. The purpose of this thesis is to automate the classification process for real simple Lie algebras. This classification is difficult because the Cartan subalgebras are not all conjugate as they are in the complex case. For the process of the real classification, one must first identify a maximally noncompact Cartan subalgebra. The process of the Cayley transform is used to find the maximally noncompact Cartan subalgebra. This Cartan subalgebra is used to find the root space decomposition and from that, the simple roots for the given real simple Lie algebra. With this information, we can then create a Satake diagram, which is unique up to isomorphism for a real simple Lie algebra. Then we match our given real Lie algebra's Satake diagram to a Satake diagram of a known real Lie algebra. The programs explained in this thesis complete this process of classification.

(241 pages)

## PUBLIC ABSTRACT

Real Simple Lie Algebras:

Cartan Subalgebras, Cayley Transforms, and Classification

Hannah M. Lewis

The differential geometry software package in Maple has the necessary tools and commands to automate the classification process for complex simple Lie algebras. The purpose of this thesis is to write the programs to complete the classification for real simple Lie algebras. This classification is difficult because the Cartan subalgebras are not all conjugate as they are in the complex case. For the process of the real classification, one must first identify a maximally noncompact Cartan subalgebra. The process of the Cayley transform is used to find this specific Cartan subalgebra. This Cartan subalgebra is used to find the simple roots for the given real simple Lie algebra. With this information, we can then create a Satake diagram. Then we match our given algebra's Satake diagram to a Satake diagram of a known algebra. The programs explained in this thesis complete this process of classification.

## Acknowledgments

This work would not have been possible without Dr. Ian Anderson. His gift of teaching math is incredible. He focused my learning and guided me to the foundational knowledge and resources I needed to complete this project.

I'm grateful to Claire Watson for her constant encouragement. I was drawn to her early in my journey here at Utah State because of her wisdom and unique understanding. Her grace when helping me through a variety of quandaries will not be forgotten.

I'm thankful to Tom Apedaile for being supportive and never making me feel inadequate. He achieved the great balance of helping when I needed help and listening when I needed a sounding board.

Finally, I am so grateful to my incredibly supportive family. My father's delight and optimism paired with his steady service to my family propelled me through. My husband, Dow, prodded me to go beyond what I thought was practical and his consistent support saw me through. No matter where life had taken me, he was always there to back me up. His reassurance and comfort were essential in this undertaking. My mother's strength and example modeled the importance of education, not settling for less, and keep on going even when it seems slow.

Hannah M Lewis

## Contents

<b>Abstract</b> . . . . .	iii
<b>Public Abstract</b> . . . . .	iv
<b>Acknowledgments</b> . . . . .	v
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 Examples . . . . .	3
1.2 Technical Details . . . . .	7
<b>2 Lie Algebras: Definitions and Concepts</b> . . . . .	<b>9</b>
2.1 Algebras: Definitions and Concepts . . . . .	9
2.2 Basic Properties of Lie Algebras . . . . .	13
2.3 Adjoint Representation and Ad . . . . .	16
2.4 Killing Form, Simple, and Semi-Simple Lie algebras . . . . .	18
2.5 Lower Central Series . . . . .	22
2.6 Normalizer . . . . .	24
<b>3 Semi Simple Lie Algebras</b> . . . . .	<b>26</b>
3.1 Cartan Subalgebras . . . . .	26
3.2 Root Space Decomposition . . . . .	33
3.2.1 Roots . . . . .	35
3.3 Cartan Decomposition and Involution . . . . .	37
<b>4 Real Simple Lie Algebras</b> . . . . .	<b>49</b>
4.1 $A_n$ : The Special Linear and Special Unitary Lie Algebras . . . . .	50
4.1.1 $SL(n)$ : Special Linear Group . . . . .	50
4.1.1.1 $sl(n, R)$ . . . . .	50
4.1.1.2 $sl(m, \mathbb{H})$ also denoted as $su^*(n)$ . . . . .	51
4.1.2 $SU(p, q)$ : Special Unitary Group . . . . .	54
4.1.2.1 $su(p, q), p > q$ . . . . .	55
4.1.2.2 $su(p, p)$ . . . . .	57
4.1.2.3 $su(n)$ . . . . .	59
4.2 $B_n$ : Odd Dimensional Special Orthogonal Lie Algebras . . . . .	60
4.2.1 $SO(p, q)$ : $p + q$ is odd: Odd Dimensional Special Orthogonal Group . . . . .	60
4.2.2 $so(p, q), p + q$ is odd . . . . .	61
4.2.3 $so(n), n$ is odd. . . . .	63
4.3 $C_n$ : Symplectic Lie Algebras . . . . .	64

4.3.1	$Sp(n, \mathbb{R})$ : Symplectic Lie Group	64
4.3.2	$sp(n, \mathbb{R})$	65
4.3.3	$Sp(p, q)$ : Symplectic Lie Group	66
4.3.4	$sp(p, q)$	67
4.3.5	$sp(p, p)$	71
4.3.6	$sp(n)$	73
4.4	$D_n$ : Even Dimensional Special Orthogonal Lie Algebras	74
4.4.1	$SO(p, q)$ : $p + q$ is even: Even Dimensional Special Orthogonal Group	74
4.4.2	$so(n, n)$	75
4.4.3	$so(p, q)$ , $p + q$ is even	78
4.4.4	$so(p - 1, p + 1)$	80
4.4.5	$so(2n)$	82
4.4.6	$so^*(2n)$ , $n$ is even	85
4.4.7	$so^*(2n)$ , $n$ is odd	87
<b>5</b>	<b>Cayley Transforms</b>	<b>90</b>
<b>6</b>	<b>Low Dimensional Isomorphisms</b>	<b>102</b>
<b>7</b>	<b>Programs Explanation and Maple Demonstrations</b>	<b>103</b>
	<b>Bibliography</b>	<b>115</b>
	<b>Appendices</b>	<b>116</b>
	<b>Maple Demonstration</b>	<b>117</b>
7.1	A	117
7.2	B	143
7.3	C	161
7.4	D	187
	<b>Computer Programs</b>	<b>207</b>



## Chapter 1

### Introduction

Sophus Lie began the study of what we would later call Lie theory in about 1870. A general classification for all Lie algebras does not exist, but mathematicians have been working on this question since the inception of Lie theory. There are three main types of Lie algebras: solvable, semi-simple, and those that are neither solvable or semi-simple[1].

The classification of semi-simple Lie algebras easily reduces to the classification of simple Lie algebras, that is, non-abelian Lie algebras whose only ideals are  $\emptyset$  and itself. The classification of complex simple Lie algebras was initiated by Wilhelm Killing in 1888-90 and completed by Élie Cartan in his Ph.D. thesis in 1894[1]. He then obtained the complete classification of real simple Lie algebras. In 1947, Eugene Dynkin provided a schematic way of summarizing the results of the complex classification which was extended to the real case by Satake in 1960.

Killing and Cartan classified simple Lie algebras into five different classes. The first four families are called the simple classical Lie algebras: the special linear algebras, the odd orthogonal algebras, the symplectic algebras, and the even orthogonal algebras. We call these  $A_n$ ,  $B_n$ ,  $C_n$ , and  $D_n$  respectively. There are five exceptional algebras,  $E_6$ ,  $E_7$ ,  $E_8$ ,  $F_4$ , and  $G_2$ .

To classify complex simple Lie algebras, we start with a Cartan subalgebra and the corresponding root space decomposition. Once the Lie algebra is decomposed into the root spaces, the expression of the positive roots in terms of the simple roots allows the construction of a Dynkin diagram. Because all Cartan subalgebras of complex simple Lie algebras are conjugate to the root space, and therefore, the structure of the Dynkin diagram does not depend on the choice of the Cartan subalgebra. The classification theorem states that for simple Lie algebras there is a one to one correspondence between

a complex simple Lie algebra and admissible Dynkin diagrams. The differential geometry software package in Maple already has the tools to automate this classification process.

The classification for real simple Lie algebras is more difficult because the Cartan subalgebras are not all conjugate. At the end of this section of the introduction, we will give some examples of inequivalent Cartan subalgebras. Because the Cartan subalgebras are not all equivalent, this means different Cartan subalgebras for the same real Lie algebra result in a different root space decomposition with different properties, e.g., the number of pure imaginary roots depends on the choice of Cartan subalgebra. Because of this we need to use a different method to classify real simple Lie algebras than is used to classify complex simple Lie algebras.

The classification of real simple Lie algebras has two steps. The first is to choose an appropriate Cartan subalgebra. We call this Cartan subalgebra maximally noncompact. The precise definition of a maximally noncompact Cartan subalgebra will be given later in chapter 3(3.3.7). A Cayley transform provides a sequence of steps by which a given Cartan subalgebra can be transformed into a maximally noncompact Cartan subalgebra. Given a maximally noncompact Cartan subalgebra, one finds the root space decomposition as before. The second step is to identify the simple roots which are pure imaginary and the simple roots which are Satake associates. The precise definition of Satake associates will be given later in chapter 3 (3.3.8). From this information we can decorate the Dynkin diagram by coloring the dots black for every imaginary root and adding red arrows between the Satake associates. We call this decorated Dynkin diagram a Satake diagram. The classification theorem for real simple Lie algebras states that there is a one to one correspondence between a real simple Lie algebra and its Satake diagram.

The purpose of this thesis is to write a computer program which automates the process of classifying real simple Lie algebras. We will start by using a Cayley transform to find the maximally noncompact Cartan subalgebra. Then this Cartan subalgebra will be used to find the root space decomposition. From that we will find the imaginary and complex roots as well as the roots that are Satake associates. This information will be used to match our given algebra to a Satake diagram of a known algebra. Then the classification is complete.

The thesis is organized as follows:

In the second chapter we review the necessary definitions for Lie algebras as well as examples that are helpful to understand the project. We also review many concepts from linear algebra that are essential to the research.

In the third chapter we review some of the basic theory of semi-simple Lie algebras. We give the necessary definitions and give examples that are helpful to understanding the project.

In the fourth chapter we look at the matrix representations of the four classical semi-simple Lie algebra classes. We start with the group operation and show the corresponding relationship for the Lie algebra.

In the fifth chapter we give the details of the Cayley transform. Using the Lie algebras  $sl(2)$  and  $so(4, 2)$ , the Cayley transform is explained with theory and examples. Then we show how the Satake diagram is constructed using the information obtained from the maximally noncompact Cartan subalgebra that is found from the Cayley transform.

In the sixth chapter we give the low dimensional isomorphisms that were confirmed with the classification program.

In chapter seven we give an explanation of the programs with a Maple demonstration.

As we remarked earlier, Cartan subalgebras are not all equivalent in the real case. Next we will give two examples of some of these nonequivalent Cartan subalgebras. Then we conclude chapter 1 by giving a technical overview of the classification problem in the real case.

Note: For this paper we will notate  $\langle x, y, z \rangle = \text{span}\{x, y, z\}$ .

## 1.1 Examples

### Example 1.1.1. $sl(2)$

Let's start with the Lie algebra  $sl(2)$  with the following multiplication table:

$sl(2)$	$e_1$	$e_2$	$e_3$
$e_1$	0	$2e_2$	$-2e_3$
$e_2$	$-2e_2$	0	$e_1$
$e_3$	$2e_3$	$-e_1$	0

When viewing  $sl(2)$  as a complex Lie algebra the following Cartan subalgebras are all equivalent:

$$\mathfrak{h}_1 = \langle e_1 \rangle, \quad (1.1)$$

$$\mathfrak{h}_2 = \langle e_2 - e_3 \rangle, \quad (1.2)$$

$$\mathfrak{h}_3 = \langle e_2 + e_3 \rangle, \quad (1.3)$$

$$\mathfrak{h}_4 = \langle -e_1 + 4e_2 - \frac{5}{4}e_3 \rangle. \quad (1.4)$$

The Cartan subalgebra  $\mathfrak{h}_1$  is equivalent to  $\mathfrak{h}_2$  by the automorphism  $\phi_1$  defined by the mapping

$$\phi_1 : (e_1, e_2, e_3) \mapsto \left( Ie_2 - Ie_3, \frac{1}{4}(e_1 - Ie_2 - Ie_3), e_1 + Ie_2 + Ie_3 \right).$$

The Cartan subalgebra  $\mathfrak{h}_1$  is equivalent to  $\mathfrak{h}_3$  by the automorphism  $\varphi_2$  defined by the mapping

$$\varphi_2 : (e_1, e_2, e_3) \mapsto \left( e_2 + e_3, \frac{1}{4}(e_1 - e_2 + e_3), e_1 + e_2 - e_3 \right).$$

The Cartan subalgebra  $\mathfrak{h}_1$  is equivalent to  $\mathfrak{h}_4$  by the automorphism  $\phi_3$  defined by the mapping

$$\phi_3 : (e_1, e_2, e_3) \mapsto \left( \frac{I}{8}(4e_1 - 16e_2 + 5e_3), \right. \\ \left. \frac{1}{8}(20e_1 - 16(1 - 2I)e_2 + 5(1 + 2I)e_3), \right. \\ \left. \frac{1}{40} \left( 5e_1 - 4(1 + 2I)e_2 + \frac{5}{4}(1 - 2I)e_3 \right) \right).$$

The Cartan subalgebra  $\mathfrak{h}_2$  is equivalent to  $\mathfrak{h}_4$  by the automorphism  $\varphi_4$  defined by the mapping

$$\varphi_4 : (e_1, e_2, e_3) \mapsto \left( \frac{4\sqrt{5}}{5}e_2 + \frac{\sqrt{5}}{4}e_3, \right. \\ \left. -\frac{1}{4}(\sqrt{5}+1)e_1 + \frac{1}{5}(5+\sqrt{5})e_2 - \frac{1}{16}(5+\sqrt{5})e_3, \right. \\ \left. -\frac{1}{4}(\sqrt{5}-1)e_1 - \frac{1}{5}(5-\sqrt{5})e_2 + \frac{1}{16}(5-\sqrt{5})e_3 \right).$$

Now, thinking of  $sl(2)$  as a real Lie algebra, we compute the restriction of the Killing form to the Cartan subalgebra for each of the four cases above:

$$\begin{aligned} k(\mathfrak{h}_1) &= [8], \\ k(\mathfrak{h}_2) &= [-8], \\ k(\mathfrak{h}_3) &= [8], \\ k(\mathfrak{h}_4) &= [-32]. \end{aligned}$$

If  $\phi : \mathfrak{g} \rightarrow \mathfrak{g}$  is an automorphism of  $\mathfrak{g}$  mapping one Cartan subalgebra  $\mathfrak{h}$  to another Cartan subalgebra  $\tilde{\mathfrak{h}}$ , then the restriction of the Killing form to  $\mathfrak{h}$  and  $\tilde{\mathfrak{h}}$  will have the same signature. In fact, in many cases the signature of the Killing form completely distinguishes all inequivalent Cartan subalgebras.

Since the signature of the Killing form for  $\mathfrak{h}_1$  and  $\mathfrak{h}_2$  are different, these are inequivalent Cartan subalgebras. Because the signature of the Killing form is equal to one for both  $\mathfrak{h}_1$  and  $\mathfrak{h}_3$ , these Cartan subalgebras are equivalent. Furthermore, we see that  $\mathfrak{h}_2$  and  $\mathfrak{h}_4$  are equivalent because the signature of the Killing form is negative one for both.

Further verifying this, we inspect the automorphisms  $\varphi_2$  and  $\varphi_4$ . These automorphisms between Cartan subalgebras that are equivalent in the real case are comprised of real numbers while the automorphisms  $\phi_1$  and  $\phi_3$  that are between inequivalent Cartan subalgebras in the real case are complex.

In the classification of real simple Lie algebras, a Cayley transform is used to move from one Cartan subalgebra to another. In this case, if we started with the Cartan

subalgebra  $\mathfrak{h}_2 = \langle e_2 - e_3 \rangle$ , then the Cayley transform would move this Cartan subalgebra to  $\mathfrak{h}_3 = \langle e_2 + e_3 \rangle$ .

**Example 1.1.2.**  $sp(4, \mathbb{R})$

Thinking of  $sp(4, \mathbb{R})$  as a complex Lie algebra the following Cartan subalgebras are all equivalent:

$$\mathfrak{h}_1 = \langle e_1, e_4 \rangle, \tag{1.5}$$

$$\mathfrak{h}_2 = \langle e_1 - e_4, e_6 - e_9 \rangle, \tag{1.6}$$

$$\mathfrak{h}_3 = \langle e_5 - e_8, e_7 - e_{10} \rangle. \tag{1.7}$$

Now, thinking of  $sp(4, \mathbb{R})$  as a real Lie algebra, we compute the restriction of the Killing form to the Cartan subalgebra for each of the four cases above:

$$k(\mathfrak{h}_1) = \begin{bmatrix} 12 & 0 \\ 0 & 12 \end{bmatrix},$$

$$k(\mathfrak{h}_2) = \begin{bmatrix} 24 & 0 \\ 0 & -24 \end{bmatrix},$$

$$k(\mathfrak{h}_3) = \begin{bmatrix} -12 & 0 \\ 0 & -12 \end{bmatrix}.$$

Because the signature of the Killing form is equal to two for  $\mathfrak{h}_1$ , zero for  $\mathfrak{h}_2$  and negative two for  $\mathfrak{h}_3$  these Cartan subalgebras are all inequivalent. Again, every Cartan subalgebra will be equivalent to one of these three Cartan subalgebras.

In the classification of real simple Lie algebras, a Cayley transform is used to move from one Cartan subalgebra to another. If we started with the Cartan subalgebra  $\mathfrak{h}_1$ , a Cayley transform would move this to a Cartan subalgebra equivalent to  $\mathfrak{h}_2$ . Similarly, if the Cayley transform were applied to  $\mathfrak{h}_2$ , the result would be equivalent to  $\mathfrak{h}_3$ .

## 1.2 Technical Details

In order to understand the structure of a real semi-simple Lie algebra, we begin with any Cartan subalgebra  $\mathfrak{h}$ . Let

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

be the corresponding root space decomposition done over the complex numbers. From the root space decomposition we construct the Cartan decomposition. This is a real decomposition. That is,  $\mathfrak{t}$  is a real subalgebra and  $\mathfrak{p}$  is a real subspace of  $\mathfrak{g}$ . The Cartan decomposition is

$$\mathfrak{g} = \mathfrak{t} \oplus \mathfrak{p}.$$

For this decomposition the following bracket relations hold:

$$[\mathfrak{t}, \mathfrak{t}] \subseteq \mathfrak{t}, \quad [\mathfrak{t}, \mathfrak{p}] \subseteq \mathfrak{p}, \quad [\mathfrak{p}, \mathfrak{p}] \subseteq \mathfrak{t}.$$

Moreover, the Killing form of  $\mathfrak{g}$  is negative definite on  $\mathfrak{t}$  and positive definite on  $\mathfrak{p}$ . Then the compact part of the Cartan decomposition,  $\mathfrak{t}$  is a subalgebra of  $\mathfrak{g}$ . The noncompact part  $\mathfrak{p}$  is not a subalgebra of  $\mathfrak{g}$ , but is a subspace of  $\mathfrak{g}$ . An important property of this construction is that the Cartan subalgebra  $\mathfrak{h}$  is aligned with the Cartan decomposition in the sense that

$$\mathfrak{h} = (\mathfrak{h} \cap \mathfrak{t}) \oplus (\mathfrak{h} \cap \mathfrak{p}).$$

A root is imaginary if every component is pure imaginary or zero. A root  $\alpha$  is noncompact if  $\mathfrak{g}_{\alpha}$  lies completely in  $\mathfrak{p}$ . If an imaginary noncompact root exists for this Cartan subalgebra, then the Cayley transform needs to be applied to find the maximally noncompact Cartan subalgebra. That imaginary noncompact root is used to construct a new Cartan subalgebra  $\tilde{\mathfrak{h}}$  whose intersection with  $\mathfrak{p}$  goes up by one dimension.

When the intersection of the Cartan subalgebra with  $\mathfrak{p}$  is of maximal dimension, then the Cartan subalgebra is said to be maximally noncompact. This maximally noncompact Cartan subalgebra will have the minimum number of pure imaginary roots. The root space decomposition can then be computed from this new Cartan subalgebra. The simple roots are then found from the root space decomposition.

The simple roots are labeled as either imaginary or complex. We then determine the Cartan matrix for the Lie algebra and transform it to its standard form. This allows us to order the simple roots in the standard order. Then the simple roots are in the order that their corresponding dots appear in the Satake diagram. For each simple root that is pure imaginary, or has all imaginary or zero entries, the dot is colored black. If the root is not pure imaginary but is complex or real, the dot is white. If a simple root is the Satake associate of another simple root, then the dots represented by these roots are connected by a red arrow.

The Satake diagram that has been constructed in this way is unique to the Lie algebra up to isomorphism. This completes the brief technical details of the algorithm for the classification of real semi simple Lie algebras.



## Chapter 2

### Lie Algebras: Definitions and Concepts

#### 2.1 Algebras: Definitions and Concepts

This chapter will review the definitions, concepts, and examples that are most helpful in discussing semi-simple Lie algebras. We will begin our discussion with a few basic definitions and concepts of Algebras. This will be followed in the next chapter by the major Lie algebra definitions needed for understanding the theory behind the classification of real simple Lie algebras.

##### **Definition 2.1.1. Algebra**

An algebra is a vector space  $A$  over a field  $F$  that has a multiplicative operator,  $*$  :  $A \times A \rightarrow A$  that satisfies the following two conditions:

1. right and left distributivity

$$(y + z) * x = y * x + z * x$$

$$x * (y + z) = x * y + x * z;$$

2. scalar multiplication

$$k \cdot (x * y) = (k \cdot x) * y = x * (k \cdot y).$$

##### **Example 2.1.1. Matrices**

Let  $M_n(\mathbb{F})$  be the set of square  $n \times n$  matrices. Let  $A_n, B_n \in M_n(\mathbb{F})$ . Then the sum of two square  $n \times n$  matrices  $A_n + B_n = C_n$ , where  $C_n \in M_n(\mathbb{F})$ . So,  $M_n$  is closed under addition. Let  $k \in \mathbb{R}$ , with  $k \cdot A_n = D_n$ . Then  $D_n \in M_n(\mathbb{F})$  and  $M_n$  is closed under scalar multiplication. So, square matrices are a vector space over a field  $F$ .

Define matrix multiplication the usual way  $(AB)_{ij} = \sum_{k=1}^n A_{ik}B_{jk}$  with  $A_n, B_n, E_n \in M_n(\mathbb{F})$ . Then  $(A_n+B_n)*E_n = A_n*E_n+B_n*E_n$  and  $E_n*(A_n+B_n) = E_n*A_n+E_n*B_n$  by the properties of matrix multiplication and the distributive property of matrices. Also we can see that  $k \cdot (A_n * B_n) = (k \cdot A_n) * B_n = A_n * (k \cdot B_n)$  by the associativity of matrices. Then the properties of an algebra are satisfied by the square  $n \times n$  matrices. Thus the square  $n \times n$  matrices are an algebra.

**Definition 2.1.2.** Lie Algebra

A Lie algebra is a vector space  $\mathfrak{g}$  over a field  $F$  with a map  $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g}$  called the Lie bracket, such that the following properties are satisfied:

1. bilinearity

$$[ax + by, z] = a[x, z] + b[y, z], \quad [z, ax + by] = a[z, x] + b[z, y];$$

2. skew symmetry

$$[x, x] = 0 \text{ which implies } [x, y] = -[y, x] \text{ for all } x, y \in \mathfrak{g};$$

3. the Jacobi identity

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

**Example 2.1.2.** Associative Algebra

Let  $(\mathfrak{g}, *)$  be an associative algebra. Let  $x, y \in \mathfrak{g}$ . Define  $[x, y] = x * y - y * x$  as the commutator, then  $(\mathfrak{g}, [\cdot, \cdot])$  is a Lie algebra.

We need to show that an associative algebra satisfies the properties of a Lie algebra under the above multiplication. First we show bilinearity. Let  $x, y, z \in \mathfrak{g}$ ,  $a, b \in \mathbb{R}$ , then

$$\begin{aligned} [ax + by, z] &= (ax + by) * z - z * (ax + by) \\ &= (ax) * z + (by) * z - z * (ax) - z * (by) \\ &= a(x * z) - a(z * x) + b(y * z) - b(z * y) \\ &= a(x * z - z * x) + b(y * z - z * y) \\ &= a[x, z] + b[y, z]. \end{aligned}$$

Similarly it can be shown that,  $[z, ax + by] = a[z, x] + b[z, y]$ .

Next, we show skew symmetry. With  $x, y \in \mathfrak{g}$ ,

$$[x, x] = x * x - x * x = 0$$

and

$$\begin{aligned} [x, y] &= x * y - y * x \\ &= -y * x + x * y \\ &= -(y * x - x * y) \\ &= -[y, x]. \end{aligned}$$

Finally, we show the Jacobi identity. With  $x, y, z \in \mathfrak{g}$

$$\begin{aligned} [x, [y, z]] + [z, [x, y]] + [y, [z, x]] &= \\ &= [x, y * z - z * y] + [z, x * y - y * x] + [y, z * x - x * z] \\ &= x * (y * z - z * y) - (y * z - z * y) * x + z * (x * y - y * x) - \\ &\quad - (x * y - y * x) * z + y * (z * x - x * z) - (z * x - x * z) * y \\ &= x * y * z - x * z * y - y * z * x + z * y * x + z * x * y - z * y * x - \\ &\quad - x * y * z + y * x * z + y * z * x - y * x * z - z * x * y + x * z * y \\ &= x * y * z - x * y * z + z * y * x - z * y * x + z * x * y - z * x * y + \\ &\quad + y * x * z - y * x * z + y * z * x - y * z * x + x * z * y - x * z * y \\ &= 0. \end{aligned}$$

So,  $\mathfrak{g}$  is a Lie algebra.

**Example 2.1.3.**  $sl(2)$

In particular, let  $\mathfrak{g}$  be the space of  $2 \times 2$  trace free matrices. We know from example 2.1.1 that the set of  $2 \times 2$  matrices form an algebra. Because matrices multiplication is associative, as long as there is closure under the Lie bracket, we know from the above

example that this is a Lie algebra. A basis for  $\mathfrak{g}$  is

$$e_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

Let  $A$  and  $B$  be matrices. For matrices the commutator  $[A, B] = AB - BA$  is the Lie bracket. We calculate

$$\begin{aligned} [e_1, e_2] &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix} \\ &= 2e_2. \end{aligned}$$

Similarly, it is calculated that  $[e_1, e_3] = -2e_3$  and  $[e_2, e_3] = e_1$ . These results can be summarized in the following table

$\mathfrak{g}$	$e_1$	$e_2$	$e_3$
$e_1$	0	$2e_2$	$-2e_3$
$e_2$	$-2e_2$	0	$e_1$
$e_3$	$2e_3$	$-e_1$	0

Thus,  $\mathfrak{g}$  is a Lie algebra.

#### Example 2.1.4. Heisenberg Algebra

The Heisenberg algebra  $\mathfrak{n}_3$  is the algebra of  $3 \times 3$  strictly upper triangular matrices

$$A = \begin{bmatrix} 0 & a & c \\ 0 & 0 & b \\ 0 & 0 & 0 \end{bmatrix}.$$

If  $A_1, A_2 \in \mathfrak{n}_3$  then,

$$\begin{aligned}
 [A_1, A_2] &= \begin{bmatrix} 0 & a_1 & c_1 \\ 0 & 0 & b_1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & a_2 & c_2 \\ 0 & 0 & b_2 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & a_2 & c_2 \\ 0 & 0 & b_2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & a_1 & c_1 \\ 0 & 0 & b_1 \\ 0 & 0 & 0 \end{bmatrix} \\
 &= \begin{bmatrix} 0 & 0 & a_1b_2 - a_2b_1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.
 \end{aligned}$$

A basis for the Heisenberg algebra is

$$e_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, e_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, e_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Then, by computing the Lie bracket, we get the multiplication table

$\mathfrak{n}_3$	$e_1$	$e_2$	$e_3$
$e_1$	0	0	$e_2$
$e_2$	0	0	0
$e_3$	$-e_2$	0	0

Thus, the Heisenberg algebra is a Lie algebra.

## 2.2 Basic Properties of Lie Algebras

### Definition 2.2.1. Lie Algebra Homomorphism

Let  $\mathfrak{g}$  and  $\mathfrak{h}$  be Lie algebras. A Lie algebra Homomorphism is a linear map  $\phi$  between  $\mathfrak{g}$  and  $\mathfrak{h}$  so that:  $\phi : \mathfrak{g} \rightarrow \mathfrak{h}$  and  $\phi([x, y]) = [\phi(x), \phi(y)]$ . In other words, it is a linear transformation which preserves the bracket relationship.

### Definition 2.2.2. Lie Algebra Isomorphism

Let  $\phi$  be a Lie algebra homomorphism. If  $\phi$  is also a bijective mapping, then  $\phi$  is a Lie algebra isomorphism.

### Example 2.2.1. $sl(2)$

Let's look at the Lie algebra from 2.1.3 with the multiplication table under the Lie bracket relationship

$sl(2)$	$e_1$	$e_2$	$e_3$
$e_1$	0	$2e_2$	$-2e_3$
$e_2$	$-2e_2$	0	$e_1$
$e_3$	$2e_3$	$-e_1$	0

Then let's look at a different Lie algebra with the following multiplication table under the Lie bracket relationship:

$sp(2)$	$f_1$	$f_2$	$f_3$
$f_1$	0	$2f_3$	$-2f_2$
$f_2$	$-2f_3$	0	$2f_1$
$f_3$	$2f_2$	$-2f_1$	0

Define  $\phi : sp(2) \rightarrow sl(2)$  by  $\phi(f_1) = -e_1$ ,  $\phi(f_2) = e_3 - e_2$ ,  $\phi(f_3) = e_2 + e_3$ .

Then we calculate

$$\begin{aligned} [\phi(f_1), \phi(f_2)] &= [-e_1, e_3 - e_2] = -[e_1, e_3] + [e_1, e_2] = 2e_3 + 2e_2 = 2(e_2 + e_3) = \phi(2f_3) \\ [\phi(f_1), \phi(f_3)] &= [-e_1, e_2 + e_3] = -[e_1, e_2] - [e_1, e_3] = -2e_2 + 2e_3 = -2(e_3 - e_2) = \phi(-2f_2) \\ [\phi(f_2), \phi(f_3)] &= [e_3 - e_2, e_2 + e_3] = [e_3, e_2 + e_3] - [e_2, e_2 + e_3] = \\ &= [e_3, e_2] + [e_3, e_3] - [e_2, e_2] - [e_2, e_3] = -e_1 - e_1 = -2e_1 = \phi(2f_1). \end{aligned}$$

So,  $\phi$  is a bijective homomorphism that preserves the bracket relationship. Then we can say that there is an isomorphism between  $sl(2)$  and  $sp(2)$ .

### Definition 2.2.3. Lie Algebra Automorphism

A Lie algebra Automorphism is a isomorphism  $H : \mathfrak{g} \rightarrow \mathfrak{g}$ .

*Remark 2.1.* The inner automorphism group of a complex Lie algebra  $\mathfrak{g}$  is the subgroup of the automorphism group of  $\mathfrak{g}$  generated by the automorphisms of the form

$$e^{\text{ad}_x} = 1 + \text{ad}_x + \frac{\text{ad}_x^2}{2!} + \dots$$

for  $x \in \mathfrak{g}$  where  $e^{\text{ad}_x}$  is a finite sum. We will talk about how to define  $\text{ad}_x$  later in this chapter. [2.3.1](#)

*Remark 2.2.* An automorphism of a group  $G$  is a one-to-one mapping in  $G$ . It represents an isomorphism of a group with itself. If the inner automorphism  $H(a) : a \rightarrow x^{-1}ax$  maps  $a$  onto itself for all  $x \in G$ , then  $a$  is called self conjugate.

**Definition 2.2.4.** Lie Subalgebra

Given a Lie algebra  $\mathfrak{g}$ , a subspace  $\mathfrak{a} \subset \mathfrak{g}$  is a Lie subalgebra if  $\mathfrak{a}$  is closed under the bracket  $[\cdot, \cdot]$ . That is, if for all  $x, y \in \mathfrak{a}$ ,  $[x, y] \in \mathfrak{a}$ , then  $\mathfrak{a}$  is a Lie subalgebra of  $\mathfrak{g}$ .

**Definition 2.2.5.** Ideal

A subspace  $\mathfrak{a}$  of a Lie algebra  $\mathfrak{g}$  is said to be an ideal if for any  $x \in \mathfrak{a}$  and for any  $y \in \mathfrak{g}$ ,  $[x, y] \in \mathfrak{a}$ .

**Properties 2.2.1.** Any ideal is also a Lie subalgebra.

**Properties 2.2.2.** Every Lie algebra  $\mathfrak{g}$  has two trivial ideals  $\{0\} \subseteq \mathfrak{g}$  and  $\mathfrak{g} \subseteq \mathfrak{g}$ .

**Example 2.2.2.**  $sl(2)$

For the Lie algebra from example [2.1.3](#), with the following multiplication table:

$sl(2)$	$e_1$	$e_2$	$e_3$
$e_1$	0	$2e_2$	$-2e_3$
$e_2$	$-2e_2$	0	$e_1$
$e_3$	$2e_3$	$-e_1$	0

Let  $\langle e_1, e_2 \rangle = \mathfrak{a}$ . Because the bracket  $[e_1, e_2] = 2e_2$  and  $2e_2 \in \mathfrak{a}$ , then  $\mathfrak{a}$  is closed under the Lie bracket and is a subalgebra of  $sl(2)$ . Since  $[e_1, e_3] = -2e_3$  and  $-2e_3 \notin \mathfrak{a}$ , then  $\mathfrak{a}$  is not an ideal.

Let  $\langle e_2, e_3 \rangle = \mathfrak{b}$ . The Lie bracket  $[e_2, e_3] = e_1$  and because  $e_1 \notin \mathfrak{b}$ ,  $\langle e_2, e_3 \rangle$  is not a subalgebra of  $sl(2)$ .

**Definition 2.2.6.** Radical

The radical of a Lie algebra is the largest solvable ideal of that Lie algebra.

**Definition 2.2.7.** Abelian Lie Algebra

A Lie algebra  $\mathfrak{g}$  is called abelian if the Lie bracket equals zero for all elements  $x, y \in \mathfrak{g}$ . That is, for all  $x, y \in \mathfrak{g}$ ,  $[x, y] = 0$ .

**Example 2.2.3.**  $so(2)$ 

For the  $2 \times 2$  skew-symmetric matrices,  $so(2)$ , there is only one possible generator up to scaling, which is

$$e_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

We calculate the Lie bracket

$$[e_1, e_1] = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

So, the Lie algebra  $so(2)$  is an abelian Lie algebra.

**2.3 Adjoint Representation and Ad****Definition 2.3.1.** Adjoint Action

Let  $\mathfrak{g}$  be a Lie algebra with  $x \in \mathfrak{g}$ . Then the adjoint representation is the linear transformation  $\text{ad}(x) : \mathfrak{g} \rightarrow \mathfrak{g}$  that is defined by:  $\text{ad}(x)(y) = [x, y] \forall x, y \in \mathfrak{g}$ . We call the map  $\text{ad}(x) : \mathfrak{g} \rightarrow \mathfrak{g}$  the adjoint action of the element  $x$  on  $\mathfrak{g}$ .

**Example 2.3.1.**  $sl(2)$ 

Let's look at the Lie algebra from 2.1.3 with the multiplication table under the bracket relationship

$sl(2)$	$e_1$	$e_2$	$e_3$
$e_1$	0	$2e_2$	$-2e_3$
$e_2$	$-2e_2$	0	$e_1$
$e_3$	$2e_3$	$-e_1$	0

Then we calculate that  $\text{ad}(e_1)(e_1) = [e_1, e_1] = 0$ ,  $\text{ad}(e_1)(e_2) = [e_1, e_2] = 2e_2$ , and  $\text{ad}(e_1)(e_3) = [e_1, e_3] = -2e_3$ . The coefficients of these make up the column vectors in the adjoint representation matrix for  $\text{ad}(e_1)$ .



This means that

$$\text{ad}(e_1) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -2 \end{bmatrix}.$$

Similarly,  $\text{ad}(e_2)(e_1) = [e_2, e_1] = -2e_2$ ,  $\text{ad}(e_2)(e_2) = [e_2, e_2] = 0$ , and  $\text{ad}(e_2)(e_3) = [e_2, e_3] = 1e_1$ . This gives the adjoint matrix for  $e_2$  to be

$$\text{ad}(e_2) = \begin{bmatrix} 0 & 0 & 1 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Finally,  $\text{ad}(e_3)(e_1) = [e_3, e_1] = 2e_3$ ,  $\text{ad}(e_3)(e_2) = [e_3, e_2] = -1e_1$ , and  $\text{ad}(e_3)(e_3) = [e_3, e_3] = 0$ . This gives the adjoint matrix for  $e_3$  to be

$$\text{ad}(e_3) = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{bmatrix}.$$

**Properties 2.3.1.** The adjoint representation satisfies the property

$$[\text{ad}(x), \text{ad}(y)](z) = \text{ad}([x, y])(z).$$

To show this, first we note that by Jacobi identity  $[x, [y, z]] + [y, [z, x]] + [z, [x, y]] = 0 \Rightarrow [x, [y, z]] - [y, [x, z]] = -[z, [x, y]]$ .

Then we calculate

$$\begin{aligned} [\text{ad}(x), \text{ad}(y)](z) &= (\text{ad}(x))(\text{ad}(y))(z) - (\text{ad}(y))(\text{ad}(x))(z) \\ &= [x, [y, z]] - [y, [x, z]] \\ &= -[z, [x, y]] \\ &= [[x, y], z] \\ &= \text{ad}([x, y])(z). \end{aligned}$$

**Definition 2.3.2.**  $\text{Ad}(x)$

For a Lie algebra  $\mathfrak{g}$ ,  $\text{Ad}$  and the adjoint action,  $\text{ad}$ , are related through the exponential

map. Specifically,  $\text{Ad}(\exp(x)) = \exp(\text{ad}_x)$  for all  $x \in \mathfrak{g}$ .

**Properties 2.3.2.** The power sum formula for  $\text{Ad}(\exp(x))$  applied on  $y$  is

$$\begin{aligned} \text{Ad}(\exp(x))(y) &= \exp(\text{ad}_x(y)) \\ &= y + [x, y] + [x, [x, y]] + \dots \\ &= \frac{y}{0!} + \frac{\text{ad}_x(y)}{1!} + \frac{\text{ad}_x^2(y)}{2!} + \frac{\text{ad}_x^3(y)}{3!} + \dots \end{aligned} \tag{2.1}$$

## 2.4 Killing Form, Simple, and Semi-Simple Lie algebras

**Definition 2.4.1.** Killing Form

Let  $\mathfrak{g}$  be a finite dimensional Lie algebra. Then the Killing form is an inner product on  $\mathfrak{g}$  defined by

$$k(X, Y) = \text{Tr}(\text{ad}(X) \text{ad}(Y)),$$

where  $\text{Tr}(A)$  is the trace of the matrix, and  $\text{ad}(X)$  is the adjoint action on  $X$ .

**Properties 2.4.1.**

$$k(\text{ad}_z(x), y) + k(x, (\text{ad}_z(y))) = 0$$

**Properties 2.4.2.** The Killing form is bilinear and symmetric.

**Example 2.4.1.**  $sl(2)$

Let's look again at the Lie algebra from example 2.1.3 with the basis

$$e_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix},$$

and the following multiplication table

$sl(2)$	$e_1$	$e_2$	$e_3$
$e_1$	0	$2e_2$	$-2e_3$
$e_2$	$-2e_2$	0	$e_1$
$e_3$	$2e_3$	$-e_1$	0

The adjoint action of these elements are represented by the matrices

$$\text{ad}(e_1) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -2 \end{bmatrix}, \quad \text{ad}(e_2) = \begin{bmatrix} 0 & 0 & 1 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \text{ad}(e_3) = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{bmatrix}.$$

The Killing form is then calculated using the definition to be

$$k(e_1, e_2) = \text{Tr}(\text{ad}(e_1) * \text{ad}(e_2)) = \text{Tr} \begin{bmatrix} 0 & 0 & 0 \\ -4 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = 0,$$

$$k(e_1, e_3) = \text{Tr}(\text{ad}(e_1) * \text{ad}(e_3)) = \text{Tr} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -4 & 0 & 0 \end{bmatrix} = 0,$$

$$k(e_2, e_3) = \text{Tr}(\text{ad}(e_2) * \text{ad}(e_3)) = \text{Tr} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} = 4.$$

Similarly  $k(e_1, e_1) = 8$ ,  $k(e_2, e_2) = 0$ ,  $k(e_3, e_3) = 0$ , etc. Therefore, the Killing form matrix is

$$\begin{bmatrix} 8 & 0 & 0 \\ 0 & 0 & 4 \\ 0 & 4 & 0 \end{bmatrix}.$$

**Example 2.4.2.**  $sl(3)$

Let's look at the Lie algebra  $sl(3)$ , of trace free  $3 \times 3$  real matrices with the following multiplication table

$\mathfrak{sl}(3)$	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$e_7$	$e_8$
$e_1$	0	0	$e_3$	$2e_4$	$-e_5$	$e_6$	$-2e_7$	$-e_8$
$e_2$	0	0	$-e_3$	$e_4$	$e_5$	$2e_6$	$-e_7$	$-2e_8$
$e_3$	$-e_3$	$e_3$	0	0	$e_1 - e_2$	$e_4$	$-e_8$	0
$e_4$	$-2e_4$	$-e_4$	0	0	$-e_6$	0	$e_1$	$e_3$
$e_5$	$e_5$	$-e_5$	$-e_1 + e_2$	$e_6$	0	0	0	$-e_7$
$e_6$	$-e_6$	$-2e_6$	$-e_4$	0	0	0	$e_5$	$e_2$
$e_7$	$2e_7$	$e_7$	$e_8$	$-e_1$	0	$-e_5$	0	0
$e_8$	$e_8$	$2e_8$	0	$-e_3$	$e_7$	$-e_2$	0	0

Then the adjoint representation of the basis elements are calculated to be

$$\text{ad}(e_1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \quad \text{ad}(e_2) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \end{bmatrix},$$

$$\text{ad}(e_3) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}, \quad \text{ad}(e_4) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\text{ad}(e_5) = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \text{ad}(e_6) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\text{ad}(e_7) = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \text{ad}(e_8) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The nonzero Killing forms are calculated to be

$$\begin{aligned} k(e_1, e_1) &= \text{Tr}(\text{ad}(e_1)\text{ad}(e_1)) = 12, & k(e_1, e_2) &= \text{Tr}(\text{ad}(e_1)\text{ad}(e_2)) = 6, \\ k(e_2, e_2) &= \text{Tr}(\text{ad}(e_2)\text{ad}(e_2)) = 12, & k(e_3, e_5) &= \text{Tr}(\text{ad}(e_3)\text{ad}(e_5)) = 6, \\ k(e_4, e_7) &= \text{Tr}(\text{ad}(e_4)\text{ad}(e_7)) = 6, & k(e_6, e_8) &= \text{Tr}(\text{ad}(e_6)\text{ad}(e_8)) = 6. \end{aligned}$$

Thus the Killing form matrix is

$$\begin{bmatrix} 12 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 12 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 \end{bmatrix}.$$

**Definition 2.4.2.** Simple Lie Algebra

A non abelian Lie algebra  $\mathfrak{g}$  is called simple if it has no non-trivial ideals. If  $\mathfrak{g}$  is simple then the solvable radical of  $\mathfrak{g}$ ,  $R(\mathfrak{g}) = \mathfrak{g}$  or  $R(\mathfrak{g}) = 0$ .

**Definition 2.4.3.** Semi-Simple Lie Algebra

A Lie algebra is called semi-simple if it is a direct sum of simple Lie algebras. A Lie algebra  $\mathfrak{g}$  is semi-simple if the solvable radical is zero.

**Theorem 2.4.1.** *Cartan Criteria*

*Let  $\mathfrak{g}$  be a finite-dimensional Lie algebra over a field of characteristic zero. Then  $\mathfrak{g}$  is semi-simple if and only if the Killing form is non-degenerate. Note: the Killing form is degenerate if  $k(X, Y) = 0$  for all  $Y \in \mathfrak{g}$  implies that  $X = 0$ .*

## 2.5 Lower Central Series

**Definition 2.5.1.** Lower Central Series

The lower central series of a Lie algebra  $\mathfrak{g}$  is the sequence of subalgebras recursively defined by

$$\mathfrak{g}_{k+1} = [\mathfrak{g}, \mathfrak{g}_k], \text{ with } \mathfrak{g}_0 = \mathfrak{g}.$$

**Properties 2.5.1.** We say that the lower central series terminates when either  $\mathfrak{g}_{k+1} = \mathfrak{g}_k$  or the subalgebra  $\mathfrak{g}_{k+1}$  is abelian, i.e  $\mathfrak{g}_{k+1} = 0$ .

**Example 2.5.1.**

Consider the Lie algebra with the following multiplication table

$\mathfrak{g}$	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$
$e_1$	0	$e_3$	$e_4$	0	$e_6$	0
$e_2$	$-e_3$	0	0	0	0	0
$e_3$	$-e_4$	0	0	0	0	0
$e_4$	0	0	0	0	0	0
$e_5$	$-e_6$	0	0	0	0	0
$e_6$	0	0	0	0	0	0

Then we calculate  $\mathfrak{g}_1 = [\mathfrak{g}, \mathfrak{g}]$ . This means that  $\mathfrak{g}_1$  is the result of bracketing every element in  $\mathfrak{g}$  with every other element in  $\mathfrak{g}$ . By the multiplication table we can see that  $\mathfrak{g}_1 = \langle e_3, e_4, e_6 \rangle$ .

We then find  $\mathfrak{g}_2 = [\mathfrak{g}, \mathfrak{g}_1]$ . This is the result of taking the Lie bracket of  $\mathfrak{g}$  with  $\mathfrak{g}_1$ . These Lie brackets are  $[e_1, e_3] = e_4, [e_1, e_4] = 0, [e_1, e_6] = 0, [e_2, e_3] = 0, [e_2, e_4] = 0, [e_2, e_6] = 0, [e_3, e_3] = 0, [e_3, e_4] = 0, [e_3, e_6] = 0, [e_4, e_4] = 0, [e_4, e_6] = 0, [e_5, e_6] = 0, [e_6, e_6] = 0$ . This shows that  $\mathfrak{g}_2 = \langle e_4 \rangle$ . Finally  $\mathfrak{g}_3 = [\mathfrak{g}, \mathfrak{g}_2]$ , which is the result of the Lie brackets  $[e_i, e_4] = 0$  for  $i = 1..6$ . The lower central series terminates at this step because  $\mathfrak{g}_3 = \{\}$ .

### Example 2.5.2.

Let  $\mathfrak{g}$  be the five dimensional Lie algebra with the nonzero structure equations  $[e_2, e_3] = e_1, [e_2, e_5] = e_3, [e_4, e_5] = e_4$ . Then  $\mathfrak{g}_0 = \mathfrak{g}$ ,  $\mathfrak{g}_1 = \langle e_1, e_3, e_4 \rangle$ ,  $\mathfrak{g}_2 = \langle e_1, e_4 \rangle$ ,  $\mathfrak{g}_3 = \langle e_4 \rangle$ ,  $\mathfrak{g}_4 = \langle e_4 \rangle$ . Since  $\mathfrak{g}_3 = \mathfrak{g}_4$ , we have completed the lower central series for this Lie algebra.

**Properties 2.5.2.** The next subalgebra in the sequence of the lower central series is always a subset of the previous subalgebra in the series, i.e.  $\mathfrak{g}_{k+1} \subseteq \mathfrak{g}_k$ .

**Properties 2.5.3.** All  $\mathfrak{g}_k$  are ideals of the Lie algebra  $\mathfrak{g}$ .

**Definition 2.5.2.** Nilpotent

A Lie Algebra is called nilpotent when the lower central series ends in the trivial lie algebra.

### Example 2.5.3.

Let  $\mathfrak{w}$  be the five dimensional Lie algebra with the nonzero structure equations  $[e_3, e_4] = e_2, [e_3, e_5] = e_1, [e_4, e_5] = e_3$ . We compute the lower central series as before to determine if the Lie algebra is nilpotent. The first of these is calculated to be  $\mathfrak{w}_0 = \langle e_1, e_2, e_3, e_4, e_5 \rangle$ . Then we find that  $\mathfrak{w}_1 = \langle e_1, e_2, e_3 \rangle, \mathfrak{w}_2 = \langle e_1, e_2 \rangle$ , and  $\mathfrak{w}_3 = \{ \}$ . Since the lower central series terminates with the trivial Lie algebra,  $\mathfrak{w}$  is nilpotent.

In 2.5.2, the lower central series for the Lie algebra  $\mathfrak{g}$  terminated with  $\langle e_4 \rangle$ , so  $\mathfrak{g}$  from that example is not nilpotent.

## 2.6 Normalizer

**Definition 2.6.1.** Normalizer

The normalizer of a subalgebra  $\mathfrak{b}$  of a Lie Algebra  $\mathfrak{g}$  is given by

$$N_{\mathfrak{g}}(\mathfrak{b}) = \{x \in \mathfrak{g} | [x, b] \in \mathfrak{b} \quad \forall b \in \mathfrak{b}\}.$$

**Example 2.6.1.**  $sl(2, \mathbb{C})$

Let's look at the simple Lie algebra of  $2 \times 2$  trace free matrices allowing for complex entries.

Then a basis for this space is

$$\begin{aligned} e_1 &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, & e_2 &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, & e_3 &= \begin{bmatrix} 0 & i \\ 0 & 0 \end{bmatrix}, \\ e_4 &= \begin{bmatrix} 0 & 0 \\ i & 0 \end{bmatrix}, & e_5 &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, & e_6 &= \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix}. \end{aligned}$$

The multiplication table for this algebra is



$\mathfrak{g}$	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$
$e_1$	0	$e_5$	0	$e_6$	$-2e_1$	$-2e_3$
$e_2$	$-e_5$	0	$-e_6$	0	$2e_2$	$2e_4$
$e_3$	0	$e_6$	0	$-e_5$	$-2e_3$	$2e_1$
$e_4$	$-e_6$	0	$e_5$	0	$2e_4$	$-2e_2$
$e_5$	$2e_1$	$-2e_2$	$2e_3$	$-2e_4$	0	0
$e_6$	$2e_3$	$-2e_4$	$-2e_1$	$2e_2$	0	0

Then the Lie algebra,  $sl(2)$ , from 2.1.3 is represented in this algebra by  $sl(2) = \mathfrak{h} = \langle e_1, e_2, e_5 \rangle$ , where  $\mathfrak{h}$  is a subalgebra of  $\mathfrak{g}$ . Let  $y = a^1e_1 + a^2e_2 + \cdots + a^6e_6$  be in  $\mathfrak{g}$ . We calculate the following Lie brackets to be

$$[y, e_1] = -a^2e_5 - a^4e_6 + 2a^5e_1 + 2a^6e_3,$$

$$[y, e_2] = a^1e_5 + a^3e_6 - 2a^5e_2 - 2a^6e_4,$$

$$[y, e_5] = -2a^1e_1 + 2a^2e_2 - 2a^3e_3 + 2a^4e_4.$$

From the first Lie bracket we can see that  $y$  is in the normalizer iff  $a^4 = a^6 = 0$ . Similarly, from the second Lie bracket,  $a^3 = a^6 = 0$  and the third  $a^3 = a^4 = 0$ . Thus, the normalizer  $N_{\mathfrak{g}}(\mathfrak{h}) = \langle e_1, e_2, e_5 \rangle$ .

**Definition 2.6.2.** Self Normalizing Subalgebra

A subalgebra is self normalizing if  $N_{\mathfrak{g}}(\mathfrak{h}) = \mathfrak{h}$  that is, if  $[x, y] \in \mathfrak{h} \forall x \in \mathfrak{h}$ , then  $y \in \mathfrak{h}$ .

**Example 2.6.2.**  $sl(2,)$

In the above example, because the normalizer of the subalgebra  $\mathfrak{h}$  is  $\mathfrak{h}$ , then this subalgebra  $\mathfrak{h}$  is self normalizing in  $\mathfrak{g}$ .

This completes our review of frequently used concepts. In the next chapter we will apply these in the theory and concepts of semi-simple Lie algebras.

## Chapter 3

### Semi Simple Lie Algebras

In this chapter we review some of the basic theory of semisimple Lie algebras. We begin with Cartan subalgebras. Then we continue the discussion with definitions and examples of the root space decomposition, the Cartan decomposition and Cartan involution. This will give the understanding needed for the discussion of the classification of real simple Lie algebras using the process of the Cayley transform that happens in chapter 5.

#### 3.1 Cartan Subalgebras

**Definition 3.1.1.** Cartan Subalgebra

Let  $\mathfrak{g}$  be a Lie algebra. A subalgebra  $\mathfrak{h}$  of  $\mathfrak{g}$  is called a Cartan subalgebra if it satisfies the following conditions:

1. The subalgebra  $\mathfrak{h}$  is nilpotent.
2. The subalgebra is self normalizing, i.e.  $N_{\mathfrak{g}}(\mathfrak{h}) = \mathfrak{h}$ .

**Properties 3.1.1.** A Cartan subalgebra always exists for any finite dimensional Lie algebra over  $\mathbb{R}$  or  $\mathbb{C}$ .

**Properties 3.1.2.** Cartan subalgebras in complex semisimple Lie algebras are unique up to conjugation. Because the real numbers are not algebraically closed, then unlike the complex case, Cartan subalgebras in real semisimple Lie algebras are not unique up to conjugation. We showed examples in Chapter 1 (1.1.1) of real simple Lie algebras where the Cartan subalgebras are not conjugate.

**Properties 3.1.3.** If  $\mathfrak{g}$  is a nilpotent Lie algebra, then  $\mathfrak{g}$  is its own Cartan subalgebra.

**Properties 3.1.4.** For the Lie algebra of  $n \times n$  real valued matrices  $sl(n, \mathbb{R})$ , a Cartan subalgebra is the subalgebra of all diagonal matrices.

**Example 3.1.1.**  $sl(2, \mathbb{R})$ 

Let's look at the Lie algebra from 2.1.3  $sl(2, \mathbb{R})$ . A standard representation for  $sl(2, \mathbb{R})$  are the following trace free matrices

$$e_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

Then we can see that  $\mathfrak{h}_1 = \langle e_1 \rangle = \left\langle \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \right\rangle$  is a Cartan subalgebra by the above property 3.1.4.

Another Cartan subalgebra is  $\mathfrak{h}_2 = \langle e_2 - e_3 \rangle = \left\langle \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \right\rangle$ . We can show this by demonstrating that  $\mathfrak{h}_2$  is nilpotent and self normalizing. Since the Lie bracket  $[e_2 - e_3, e_2 - e_3] = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ , then  $\mathfrak{h}_2$  is trivially nilpotent.

Next we calculate the normalizer. Let  $y = a^1 e_1 + a^2 e_2 + a^3 e_3$ . Then we calculate the Lie bracket to be

$$[y, e_2 - e_3] = [y, e_2] - [y, e_3] = 2a^1 e_2 - a^3 e_1 + 2a^1 e_3 - a^2 e_1.$$

So  $y$  is in the normalizer iff  $a^2 = a^3 = 0$ . Then  $N_{sl(2)}(\langle e_2 - e_3 \rangle) = \langle e_2, e_3 \rangle$ . This means that  $\mathfrak{h}_2$  is self normalizing. Thus  $\mathfrak{h}_2$  is a Cartan subalgebra of  $sl(2, \mathbb{R})$ .

**Example 3.1.2.**  $sl(3)$ 

Let's look at again at 2.4.2  $sl(3)$  with the following multiplication table:

$sl(3)$	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$e_7$	$e_8$
$e_1$	0	0	$e_3$	$2e_4$	$-e_5$	$e_6$	$-2e_7$	$-e_8$
$e_2$	0	0	$-e_3$	$e_4$	$e_5$	$2e_6$	$-e_7$	$-2e_8$
$e_3$	$-e_3$	$e_3$	0	0	$e_1 - e_2$	$e_4$	$-e_8$	0
$e_4$	$-2e_4$	$-e_4$	0	0	$-e_6$	0	$e_1$	$e_3$
$e_5$	$e_5$	$-e_5$	$-e_1 + e_2$	$e_6$	0	0	0	$-e_7$
$e_6$	$-e_6$	$-2e_6$	$-e_4$	0	0	0	$e_5$	$e_2$
$e_7$	$2e_7$	$e_7$	$e_8$	$-e_1$	0	$-e_5$	0	0
$e_8$	$e_8$	$2e_8$	0	$-e_3$	$e_7$	$-e_2$	0	0

We consider the subalgebra  $\mathfrak{h} = \langle e_1, e_2 \rangle$ . The lower central series of the subalgebra  $\mathfrak{h}$  is  $\mathfrak{h}_0 = \langle e_1, e_2 \rangle$ , and  $\mathfrak{h}_1 = \{ \}$ . This means that the subalgebra  $\mathfrak{h}$  is nilpotent.

Next we calculate the normalizer. Let  $y = a^1e_1 + a^2e_2 + a^3e_3 + a^4e_4 + a^5e_5 + a^6e_6 + a^7e_7 + a^8e_8$  be in  $sl(3)$ . Then we calculate the Lie bracket to be

$$[y, e_1] = -a^3e_3 - 2a^4e_4 + a^5e_5 - a^6e_6 + 2a^7e_7 + a^8e_8.$$

So  $y$  is in the normalizer iff  $a^3 = a^4 = a^5 = a^6 = a^7 = a^8 = 0$ .

Then based on this, let  $y_1 = a^1e_1 + a^2e_2$ . Then we calculate the Lie bracket to be

$$[y, e_1] = 0.$$

Then the normalizer of  $sl(3)$  with respect to  $\mathfrak{h}$  is  $N_{sl(3)}(\langle e_1, e_2 \rangle) = \langle e_1, e_2 \rangle$ . This shows that  $\mathfrak{h}$  is self normalizing. Thus,  $\mathfrak{h}$  is a Cartan subalgebra of  $sl(3)$ .

**Example 3.1.3.**  $so(4, 2)$

The Lie algebra  $so(4, 2)$  has the following multiplication table:

$so(4, 2)$	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$e_7$	$e_8$	$e_9$	$e_{10}$	$e_{11}$	$e_{12}$	$e_{13}$	$e_{14}$	$e_{15}$
$e_1$	0	$e_2$	$-e_3$	0	$e_5$	$-e_6$	$e_7$	$e_8$	0	0	$-e_{11}$	$-e_{12}$	0	0	0
$e_2$	$-e_2$	0	$e_1 - e_4$	$e_2$	0	0	0	0	$e_7$	$e_8$	$-e_{13}$	$-e_{14}$	0	0	0
$e_3$	$e_3$	$-e_1 + e_4$	0	$-e_3$	0	0	$e_9$	$e_{10}$	0	0	0	0	$-e_{11}$	$-e_{12}$	0
$e_4$	0	$-e_2$	$e_3$	0	$e_5$	$-e_6$	0	0	$e_9$	$e_{10}$	0	0	$-e_{13}$	$-e_{14}$	0
$e_5$	$-e_5$	0	0	$-e_5$	0	$-e_1 - e_4$	0	0	0	0	$e_9$	$e_{10}$	$-e_7$	$-e_8$	0
$e_6$	$e_6$	0	0	$e_6$	$e_1 + e_4$	0	$e_{13}$	$e_{14}$	$-e_{11}$	$-e_{12}$	0	0	0	0	0
$e_7$	$-e_7$	0	$-e_9$	0	0	$-e_{13}$	0	0	$e_5$	0	$-e_1$	$e_{15}$	$-e_2$	0	$-e_8$
$e_8$	$-e_8$	0	$-e_{10}$	0	0	$-e_{14}$	0	0	0	$e_5$	$-e_{15}$	$-e_1$	0	$-e_2$	$e_7$
$e_9$	0	$-e_7$	0	$-e_9$	0	$e_{11}$	$-e_5$	0	0	0	$-e_3$	0	$-e_4$	$e_{15}$	$-e_{10}$
$e_{10}$	0	$-e_8$	0	$-e_{10}$	0	$e_{12}$	0	$-e_5$	0	0	0	$-e_3$	$-e_{15}$	$-e_4$	$e_9$
$e_{11}$	$e_{11}$	$e_{13}$	0	0	$-e_9$	0	$e_1$	$e_{15}$	$e_3$	0	0	0	$e_6$	0	$-e_{12}$
$e_{12}$	$e_{12}$	$e_{14}$	0	0	$-e_{10}$	0	$-e_{15}$	$e_1$	0	$e_3$	0	0	0	$e_6$	$e_{11}$
$e_{13}$	0	0	$e_{11}$	$e_{13}$	$e_7$	0	$e_2$	0	$e_4$	$e_{15}$	$-e_6$	0	0	0	$-e_{14}$
$e_{14}$	0	0	$e_{12}$	$e_{14}$	$e_8$	0	0	$e_2$	$-e_{15}$	$e_4$	0	$-e_6$	0	0	$e_{13}$
$e_{15}$	0	0	0	0	0	0	$e_8$	$-e_7$	$e_{10}$	$-e_9$	$e_{12}$	$-e_{11}$	$e_{14}$	$-e_{13}$	0

We will show that a Cartan subalgebra of  $so(4, 2)$  is  $\mathfrak{h} = \langle e_{15}, e_1 + e_4, e_2 - e_3 \rangle$ . We start with the lower central series. Then  $\mathfrak{h}_0 = \langle e_{15}, e_1 + e_4, e_2 - e_3 \rangle$ . Then we calculate the potentially nonzero brackets of  $h_1 = [h_0, h_0]$  to be

$$\begin{aligned}
[e_{15}, e_1 + e_4] &= [e_{15}, e_1] + [e_{15}, e_4] = 0 + 0 = 0, \\
[e_{15}, e_2 - e_3] &= [e_{15}, e_2] - [e_{15}, e_3] = 0 - 0 = 0, \\
[e_1 + e_4, e_2 - e_3] &= [e_1, e_2 - e_3] + [e_4, e_2 - e_3] = \\
&= [e_1, e_2] - [e_1, e_3] + [e_4, e_2] - [e_4, e_3] = \\
&= e_2 + e_3 - e_2 - e_3 = 0.
\end{aligned}$$

Then  $\mathfrak{h}_1$  is the trivial Lie algebra and  $\mathfrak{h}$  is nilpotent.

Next we calculate the normalizer. Let  $y = a^1 e_1 + a^2 e_2 + \dots + a^{14} e_{14} + a^{15} e_{15}$  be in  $so(4, 2)$ . Then we calculate the Lie bracket to be

$$[y, e_{15}] = a^7 e_8 - a^8 e_7 + a^9 e_{10} - a^{10} e_9 + a^{11} e_{12} - a^{12} e_{11} + a^{13} e_{14} - a^{14} e_{13}.$$

So  $y$  is in the normalizer iff  $a^7 = a^8 = a^9 = a^{10} = a^{11} = a^{12} = a^{13} = a^{14} = 0$ .

Then based on this, let  $y_1 = a^1e_1 + a^2e_2 + a^3e_3 + a^4e_4 + a^5e_5 + a^6e_6 + a^{15}e_{15}$ . Then we calculate the Lie bracket to be

$$[y_1, e_1 + e_4] = [y_1, e_1] + [y_1, e_4] = -a^2e^2 + a^3e_3 - a^5e_5 + a^6e_6 + a^2e_2 - a^3e_3 - a^5e_5 + a^6e_6.$$

So  $y_1$  is in the normalizer iff  $a^5 = a^6 = 0$ .

Then based on this, let  $y_2 = a^1e_1 + a^2e_2 + a^3e_3 + a^4e_4 + a^{15}e_{15}$ . Then we calculate the Lie bracket to be

$$[y_2, e_2 - e_3] = [y_2, e_2] - [y_2, e_3] = a^1e_2 - a^3e_1 + a^3e_4 + a^4e_2 + a^1e_3 - a^2e_1 + a^2e_4 - a^4e_3.$$

So  $y_2$  is in the normalizer. Then the normalizer of  $so(4, 2)$  with respect to  $\mathfrak{h}$  is  $N_{sl(4,2)}(\langle e_{15}, e_1 + e_4, e_2 - e_3 \rangle) = \langle e_1, e_2, e_3, e_4, e_{15} \rangle$ . This shows that  $\mathfrak{h}$  is self normalizing. Thus,  $\mathfrak{h}$  is a Cartan subalgebra of  $so(4, 2)$ .

**Definition 3.1.2.** Conjugation on  $\mathfrak{g}$

A Cartan subalgebra  $\mathfrak{h}$  of a Lie algebra  $\mathfrak{g}$  for  $x \in \mathfrak{g}$ , is said to be conjugate to  $\tilde{\mathfrak{h}}$  if

$$e^{\text{ad}(x)}\mathfrak{h} = \tilde{\mathfrak{h}}.$$

**Definition 3.1.3.** Diagonalizable

In linear algebra, if a square matrix  $A$  is similar to a diagonal matrix, then  $A$  is called diagonalizable. In particular,  $A$  is diagonalizable if there exists an invertible matrix  $Q$  such that  $Q^{-1}AQ$  is a diagonal matrix.

Let  $V$  be a finite-dimensional vector space. Then a linear map  $P : V \rightarrow V$  is called diagonalizable if there exists an ordered basis of  $V$  with respect to which  $P$  is represented by a diagonal matrix.

**Theorem 3.1.1.** *If  $\mathfrak{g}$  is semi simple and  $\mathfrak{h}$  is a Cartan subalgebra, then  $ad(x)$  is diagonalizable for any  $x \in \mathfrak{h}$ . In fact, since  $\mathfrak{h}$  is abelian, we can say that if  $\langle h_1, h_2, h_3, \dots, h_m \rangle$  is a basis for the Cartan subalgebra, then  $ad(h_i)$  are simultaneously diagonalizable.*

**Example 3.1.4.**  $sl(3)$

Refer back to the multiplication table for  $sl(3)$  3.1.2. Remember that a Cartan subalgebra for this Lie algebra is  $\mathfrak{h} = \langle e_1, e_2 \rangle$ . We can see that the adjoint matrices for the two Cartan subalgebra elements are

$$\text{ad}(e_1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \quad \text{ad}(e_2) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \end{bmatrix}.$$

These are diagonal matrices.

**Example 3.1.5.**  $su(2,1)$

Let's look at the real simple Lie algebra  $su(2,1)$ . The multiplication table for this Lie algebra is

$su(2,1)$	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$e_7$	$e_8$
$e_1$	0	0	$3e_7$	$3e_8$	0	0	$-3e_3$	$-3e_4$
$e_2$	0	0	$-e_3$	$e_4$	$-2e_5$	$2e_6$	$-e_7$	$e_8$
$e_3$	$-3e_7$	$e_3$	0	$e_2$	0	$-e_8$	$2e_5$	$e_1$
$e_4$	$-3e_8$	$-e_4$	$-e_2$	0	$-e_7$	0	$e_1$	$2e_6$
$e_5$	0	$2e_5$	0	$e_7$	0	$e_2$	0	$-e_3$
$e_6$	0	$-2e_6$	$e_8$	0	$-e_2$	0	$-e_4$	0
$e_7$	$3e_3$	$e_7$	$-2e_5$	$-e_1$	0	$e_4$	0	$e_2$
$e_8$	$3e_4$	$-e_8$	$-e_1$	$-2e_6$	$e_3$	0	$-e_2$	0

A Cartan subalgebra for this Lie algebra is  $\mathfrak{h} = \langle e_1, e_2 \rangle$ . We can see that the adjoint matrices for the two Cartan subalgebra elements are

$$\text{ad}(e_1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \text{ad}(e_2) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The second of these is already a diagonal matrix. We will now show that the first,  $\text{ad}(e_1)$  is diagonalizable.

For a matrix to be diagonalizable, we know we can find a matrix  $P$  such that  $P^{-1}\text{ad}(e_1)P = D$ , where  $D$  is a diagonal matrix. For this matrix we calculate

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & I & 0 & -I & 0 & 0 & 0 & 0 \\ I & 0 & -I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$



Then we calculate that  $P^{-1}\text{ad}(e_1)P =$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -3I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3I \end{bmatrix}.$$

Note: This matrix  $P$  also keeps the matrix  $P^{-1}\text{ad}(e_2)P$  as a diagonal matrix. The matrix  $P$  can be used to diagonalize all of the adjoint matrices for the Cartan subalgebra elements of any given Lie algebra.

### 3.2 Root Space Decomposition

#### Definition 3.2.1. Root Space Decomposition

Let  $\mathfrak{g}$  be a semi-simple Lie algebra and let  $\mathfrak{h}$  be a Cartan subalgebra. Then, let  $\langle h_1, h_2, h_3, \dots, h_m \rangle$  be a basis for the Cartan subalgebra  $\mathfrak{h}$ . A vector  $x \in \mathfrak{g}$  is called a root vector if  $\text{ad}(h_i)(x) = \alpha_i x$ . The  $m$ -tuple  $\alpha = [\alpha_1, \dots, \alpha_m]$  is called the root of  $x$ . We denote the set of all roots by  $\Delta = \{\alpha_1, \alpha_2, \dots, \alpha_N\}$  are called the roots of  $\mathfrak{g}$  with respect to  $\mathfrak{h}$ . The root space decomposition of  $\mathfrak{g}$  with respect to  $\mathfrak{h}$  is

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

The root spaces are generally complex vectors so that this decomposition should be considered a decomposition of the complexification of  $\mathfrak{g}$ .

#### Example 3.2.1. $sl(3)$

Let's look at again at 2.4.2  $sl(3)$ . A Cartan subalgebra for  $sl(3)$  is  $\mathfrak{h} = \langle e_1, e_2 \rangle$ . Then we calculate  $\mathfrak{g}_{\alpha_1} = \langle e_3 \rangle$  by

$$\text{ad}(e_1)(e_3) = [e_1, e_3] = e_3,$$

$$\text{ad}(e_2)(e_3) = [e_2, e_3] = -e_3,$$

so that  $\alpha_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ .

We calculate  $\mathfrak{g}_{\alpha_2} = \langle e_4 \rangle$  to be

$$\text{ad}(e_1)(e_4) = [e_1, e_4] = 2e_4,$$

$$\text{ad}(e_2)(e_4) = [e_2, e_4] = e_4,$$

so that  $\alpha_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ .

We calculate  $\mathfrak{g}_{\alpha_3} = \langle e_5 \rangle$  to be

$$\text{ad}(e_1)(e_5) = [e_1, e_5] = -e_5,$$

$$\text{ad}(e_2)(e_5) = [e_2, e_5] = e_5,$$

so that,  $\alpha_3 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ .

We calculate  $\mathfrak{g}_{\alpha_4} = \langle e_6 \rangle$  to be

$$\text{ad}(e_1)(e_6) = [e_1, e_6] = e_6,$$

$$\text{ad}(e_2)(e_6) = [e_2, e_6] = 2e_6,$$

so that  $\alpha_4 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ .

We calculate  $\mathfrak{g}_{\alpha_5} = \langle e_7 \rangle$  to be

$$\text{ad}(e_1)(e_7) = [e_1, e_7] = -2e_7,$$

$$\text{ad}(e_2)(e_7) = [e_2, e_7] = -e_7,$$

so that  $\alpha_5 = \begin{bmatrix} -2 \\ -1 \end{bmatrix}$ .

We calculate  $\mathfrak{g}_{\alpha_6} = \langle e_8 \rangle$  to be

$$\text{ad}(e_1)(e_8) = [e_1, e_8] = -e_8,$$

$$\text{ad}(e_2)(e_8) = [e_2, e_8] = -2e_8,$$

so that  $\alpha_6 = \begin{bmatrix} -1 \\ -2 \end{bmatrix}$ .

So, the root space decomposition of

$$sl(3) = \langle e_1, e_2 \rangle \oplus \langle e_3 \rangle \oplus \langle e_4 \rangle \oplus \langle e_5 \rangle \oplus \langle e_6 \rangle \oplus \langle e_7 \rangle \oplus \langle e_8 \rangle$$

with the corresponding roots  $\alpha_i$  associated with the elements in  $\mathfrak{g}_{\alpha_i}$  as given above.

**Properties 3.2.1.** The root space decomposition satisfies the following properties:

1. If  $\alpha$  is a root, then  $-\alpha$  is a root.
2. If  $x \in \mathfrak{g}_\alpha$  and  $y \in \mathfrak{g}_\beta$ , then  $[x, y] \in \mathfrak{g}_{\alpha+\beta}$  if  $\alpha + \beta$  is a root. Otherwise,  $[x, y] = 0$ .
3. Root spaces are always one dimensional.

### 3.2.1 Roots

**Definition 3.2.2.** Positive Roots and Negative Roots

Let the  $m$ -tuples  $\Delta = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$  be the roots of  $\mathfrak{g}$  with respect to the Cartan subalgebra  $\mathfrak{h}$ . Let

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

be the root space decomposition of  $\mathfrak{g}$  with respect to  $\mathfrak{h}$ . Then we can choose the positive roots  $\alpha_+$  in many ways. A common method in choosing positive roots, and the method we will use in this paper, is to choose the roots where the first term in the root is positive. The negative roots are  $\alpha_- = -\alpha_+$ . It is always possible to write  $\Delta = \alpha_+ \cup \alpha_-$ . If  $\alpha, \beta \in \alpha_+$  and  $\alpha + \beta \in \Delta$ , then  $\alpha + \beta \in \alpha_+$ .

**Example 3.2.2.**

Let's look at the Lie algebra  $sl(3)$  2.4.2 again. One possible Cartan subalgebra  $\mathfrak{h} = \langle e_1, e_2 \rangle$ . The roots of  $sl(3)$  were calculated in the previous example to be

$$\Delta = \left\{ \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} -2 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ -2 \end{bmatrix} \right\}.$$

We choose the positive roots by choosing the root with the first term being positive. So the positive roots are

$$\alpha_+ = \left\{ \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\}.$$

The negative roots are then determined to be

$$\alpha_- = \left\{ \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} -2 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ -2 \end{bmatrix} \right\}.$$

**Definition 3.2.3.** Simple Roots

An element of the positive roots  $\alpha_+$  is called a simple root if it cannot be written as a sum of any two elements of  $\alpha_+$ .

**Example 3.2.3.**

In the previous example we can see that in the set positive of roots

$$\alpha_+ = \left\{ \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\}.$$

The only one of these that can be written as a sum of the other two is

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

We then can determine the simple roots to be

$$\alpha_s = \left\{ \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\}.$$

### 3.3 Cartan Decomposition and Involution

#### Definition 3.3.1. Cartan Involution

A Cartan involution on a Lie algebra  $\mathfrak{g}$  is a Lie algebra automorphism  $\Theta : \mathfrak{g} \rightarrow \mathfrak{g}$  such that:

1. The square of  $\Theta$  is equal to the identity, i.e.  $\Theta^2 = \text{Id}$ .
2. The Killing form  $k$  has the property that  $k_{\Theta}(X, Y) = -k(X, \Theta Y)$  is a positive definite form.
3. Two Cartan involutions  $\Theta_1$  and  $\Theta_2$  are considered equivalent if they differ only by an inner automorphism.

#### Definition 3.3.2. Cartan Decomposition

A Cartan decomposition of a Lie algebra  $\mathfrak{g}$  is a vector space decomposition  $\mathfrak{g} = \mathfrak{t} \oplus \mathfrak{p}$  satisfying the following properties:

1. The set  $\mathfrak{t}$  is subalgebra and the set  $\mathfrak{p}$  is a subspace.
2. The Lie brackets  $[\mathfrak{t}, \mathfrak{t}] \subset \mathfrak{t}$ ,  $[\mathfrak{t}, \mathfrak{p}] \subset \mathfrak{p}$  and  $[\mathfrak{p}, \mathfrak{p}] \subset \mathfrak{t}$  hold.
3. The restriction of the Killing form of  $\mathfrak{t}$  and  $\mathfrak{p}$  are negative definite and positive definite respectively.

The subalgebra  $\mathfrak{t}$  is called the compact part of the Cartan decomposition and the subspace  $\mathfrak{p}$  is called the positive part.

#### Definition 3.3.3. Symmetric Pair

Let  $\mathfrak{g}$  be a Lie algebra,  $T \subset \mathfrak{g}$  a subalgebra, and  $P \subset \mathfrak{g}$  a subspace. The subalgebra, subspace pair  $T, P$  is a symmetric pair if the following conditions hold:

1. The Lie algebra  $\mathfrak{g} = T \oplus P$ .
2. For all  $x \in T$  and  $y \in P$ ,  $[x, y] \in T$ .
3. For all  $x \in P$  and  $y \in P$ ,  $[x, y] \in T$ .

Cartan decomposition and Cartan involution are related in the following way.

**Theorem 3.3.1.** *Given a Cartan decomposition, the linear transformation which is the identity on  $\mathfrak{t}$  and the negative identity on  $\mathfrak{p}$  is a Cartan involution. Conversely, given a Cartan involution  $\Theta$ , the eigenspaces  $E_1 = \mathfrak{p}$  and  $E_{-1} = \mathfrak{t}$  define a Cartan decomposition.*

The next theorem gives an easy way of finding a Cartan decomposition of matrix Lie algebras.

**Theorem 3.3.2.** *For a Lie algebra with a matrix representation, if the set of matrices that define the basis for the representation are closed under transposition, then the decomposition into skew-Hermitian ( $\mathfrak{t}$ ) and Hermitian ( $\mathfrak{p}$ ) matrices will give a Cartan decomposition. (note: a matrix that has only real entries is Hermitian if and only if it is a symmetric matrix, i.e., if it is symmetric with respect to the main diagonal)*

Our next theorem states the relationship between two Cartan decompositions.

**Theorem 3.3.3.** *Let  $\mathfrak{g}_0$  be a semisimple Lie algebra over  $\mathbb{R}$ . Suppose*

$$\mathfrak{g}_0 = \mathfrak{t}_1 + \mathfrak{p}_1, \quad \mathfrak{g}_0 = \mathfrak{t}_2 + \mathfrak{p}_2$$

*are two cartan decompositions of  $\mathfrak{g}_0$ . Then there exists an element  $\psi \in \text{Int}(\mathfrak{g}_0)$ , where the Lie algebra of  $\text{Int}(\mathfrak{g})$  is the image of adjoint representation of  $\mathfrak{g}$ , such that*

$$\psi \cdot \mathfrak{t}_1 = \mathfrak{t}_2, \quad \psi \cdot \mathfrak{p}_1 = \mathfrak{p}_2$$

**Corollary 3.3.1.** Any two equivalent Cartan involutions  $\Theta_1$  and  $\Theta_2$  on a Lie algebra  $\mathfrak{g}$  are related by an inner automorphism  $\phi : \mathfrak{g} \rightarrow \mathfrak{g}$ , where  $\Theta_2 = \phi\Theta_1\phi^{-1}$ .

*Remark 3.1.* We can start with Cartan subalgebras that do not relate by an inner automorphism. But, the resulting Cartan decompositions will be equivalent under an inner automorphism

$$L(t) : \mathfrak{g} \rightarrow \mathfrak{g}$$

.

Finally we remark on the role that Cartan decompositions play in studying the isomorphism problem.

**Theorem 3.3.4.** [4] Suppose  $\mathfrak{g}_1$  and  $\mathfrak{g}_2$  are real forms of the same simple Lie algebra  $\mathfrak{g}$  over  $\mathbb{C}$ . Let  $\mathfrak{g}_1 = \mathfrak{t}_1 + \mathfrak{p}_1$  and  $\mathfrak{g}_2 = \mathfrak{t}_2 + \mathfrak{p}_2$  be any Cartan decompositions. Then if  $\mathfrak{t}_1$  and  $\mathfrak{t}_2$  are isomorphic,  $\mathfrak{g}_1$  and  $\mathfrak{g}_2$  are isomorphic.

**Example 3.3.1.**  $sl(3)$

Let's look again at  $sl(3)$  2.4.2. For reference, here is the multiplication table for this Lie algebra again

$sl(3)$	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$e_7$	$e_8$
$e_1$	0	0	$e_3$	$2e_4$	$-e_5$	$e_6$	$-2e_7$	$-e_8$
$e_2$	0	0	$-e_3$	$e_4$	$e_5$	$2e_6$	$-e_7$	$-2e_8$
$e_3$	$-e_3$	$e_3$	0	0	$e_1 - e_2$	$e_4$	$-e_8$	0
$e_4$	$-2e_4$	$-e_4$	0	0	$-e_6$	0	$e_1$	$e_3$
$e_5$	$e_5$	$-e_5$	$-e_1 + e_2$	$e_6$	0	0	0	$-e_7$
$e_6$	$-e_6$	$-2e_6$	$-e_4$	0	0	0	$e_5$	$e_2$
$e_7$	$2e_7$	$e_7$	$e_8$	$-e_1$	0	$-e_5$	0	0
$e_8$	$e_8$	$2e_8$	0	$-e_3$	$e_7$	$-e_2$	0	0

The standard matrix representation of  $sl(3)$  is given with the following trace free matrices

$$\begin{aligned}
 e_1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, & e_2 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, & e_3 &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & e_4 &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
 e_5 &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & e_6 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, & e_7 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, & e_8 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.
 \end{aligned}$$

Now we start to look for the symmetric matrices. The basis elements  $e_1$  and  $e_2$  are the first choices. The rest will have to be constructed.

The following symmetric matrices can be constructed through addition to get

$$e_3 + e_5 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad e_4 + e_7 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad e_6 + e_8 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Together, these symmetric matrices form

$$\mathfrak{p} = \langle e_1, e_2, e_3 + e_5, e_4 + e_7, e_6 + e_8 \rangle.$$

We then use subtraction to construct the skew-symmetric matrices to be

$$e_3 - e_5 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad e_4 - e_7 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad e_6 - e_8 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}.$$

Then we can say that

$$\mathfrak{t} = \langle e_3 - e_5, e_4 - e_7, e_6 - e_8 \rangle.$$

To check that this is correct, we need to find the Killing forms of  $\mathfrak{t}$  and  $\mathfrak{p}$ . The Killing form of  $\mathfrak{t}$  is

$$K(\mathfrak{t}) = \begin{bmatrix} -12 & 0 & 0 \\ 0 & -12 & 0 \\ 0 & 0 & -12 \end{bmatrix}$$

which is, in fact, negative definite. The Killing form of  $\mathfrak{p}$  is

$$K(\mathfrak{p}) = \begin{bmatrix} 12 & 6 & 0 & 0 & 0 \\ 6 & 12 & 0 & 0 & 0 \\ 0 & 0 & 12 & 0 & 0 \\ 0 & 0 & 0 & 12 & 0 \\ 0 & 0 & 0 & 0 & 12 \end{bmatrix}$$

which is positive definite. Now, we check that  $[\mathfrak{t}, \mathfrak{t}] \subset \mathfrak{t}$ ,  $[\mathfrak{t}, \mathfrak{p}] \subset \mathfrak{p}$  and  $[\mathfrak{p}, \mathfrak{p}] \subset \mathfrak{t}$ .

First we check that  $[\mathfrak{t}, \mathfrak{t}] \subset \mathfrak{t}$  by evaluating the following Lie brackets to be

$$[e_3 - e_5, e_3 - e_5] = 0,$$

$$[e_3 - e_5, e_4 - e_7] = e_8 - e_6,$$

$$[e_3 - e_5, e_6 - e_8] = e_4 - e_7,$$



$$[e_4 - e_7, e_3 - e_5] = e_6 - e_8,$$

$$[e_4 - e_7, e_4 - e_7] = 0,$$

$$[e_4 - e_7, e_6 - e_8] = -e_3 + e_5,$$

$$[e_6 - e_8, e_3 - e_5] = -e_4 + e_7,$$

$$[e_6 - e_8, e_4 - e_7] = -e_5 + e_3,$$

$$[e_6 - e_8, e_6 - e_8] = 0.$$

This shows that  $[\mathfrak{t}, \mathfrak{t}] \subset \mathfrak{t}$ .

Now we check that  $[\mathfrak{t}, \mathfrak{p}] \subset \mathfrak{p}$  by evaluating the following Lie brackets to be

$$[e_3 - e_5, e_1] = -e_3 - e_5,$$

$$[e_3 - e_5, e_2] = e_3 + e_5,$$

$$[e_3 - e_5, e_3 + e_5] = 2e_1 - 2e_2,$$

$$[e_3 - e_5, e_4 + e_7] = -e_8 - e_6,$$

$$[e_3 - e_5, e_6 + e_8] = e_4 + e_7,$$

$$[e_4 - e_7, e_1] = -2e_4 - 2e_7,$$

$$[e_4 - e_7, e_2] = -e_4 - e_7,$$

$$[e_4 - e_7, e_3 + e_5] = -e_6 - e_8,$$

$$[e_4 - e_7, e_4 + e_7] = 2e_1,$$

$$[e_4 - e_7, e_6 + e_8] = e_3 + e_5,$$

$$[e_6 - e_8, e_1] = -e_6 - e_8,$$

$$[e_6 - e_8, e_2] = -2e_6 - 2e_8,$$

$$[e_6 - e_8, e_3 + e_5] = -e_4 - e_7,$$

$$[e_6 - e_8, e_4 + e_7] = e_5 + e_3,$$

$$[e_6 - e_8, e_6 + e_8] = 2e_2.$$

This shows that  $[\mathfrak{t}, \mathfrak{p}] \subset \mathfrak{p}$ .

Next we check that  $[\mathfrak{p}, \mathfrak{p}] \subset \mathfrak{t}$  by evaluating the following

$$[e_1, e_1] = 0,$$

$$[e_1, e_2] = 0,$$

$$[e_1, e_3 + e_5] = e_3 - e_5,$$

$$[e_1, e_4 + e_7] = 2e_4 - 2e_7,$$

$$[e_1, e_6 + e_8] = e_6 - e_8,$$

$$[e_2, e_1] = 0,$$

$$[e_2, e_2] = 0,$$

$$[e_2, e_3 + e_5] = -e_3 + e_5,$$

$$[e_2, e_4 + e_7] = e_4 - e_7,$$

$$[e_2, e_6 + e_8] = 2e_6 - 2e_8,$$

$$[e_3 + e_5, e_1] = -e_3 + e_5,$$

$$[e_3 + e_5, e_2] = e_3 - e_5,$$

$$[e_3 + e_5, e_3 + e_5] = 0,$$

$$[e_3 + e_5, e_4 + e_7] = -e_8 + e_6,$$

$$[e_3 + e_5, e_6 + e_8] = e_4 - e_7,$$

$$[e_4 + e_7, e_1] = -2e_4 + 2e_7,$$

$$[e_4 + e_7, e_2] = -e_4 + e_7,$$

$$[e_4 + e_7, e_3 + e_5] = -e_6 + e_8,$$

$$[e_4 + e_7, e_4 + e_7] = 0,$$

$$[e_4 + e_7, e_6 + e_8] = e_3 - e_5,$$

$$[e_6 + e_8, e_1] = -e_6 + e_8,$$

$$[e_6 + e_8, e_2] = -2e_6 + 2e_8,$$

$$[e_6 + e_8, e_3 + e_5] = -e_4 + e_7,$$

$$[e_6 + e_8, e_4 + e_7] = e_5 - e_3,$$

$$[e_6 + e_8, e_6 + e_8] = 0.$$

This shows that  $[\mathfrak{t}, \mathfrak{t}] \subset \mathfrak{t}$ ,  $[\mathfrak{t}, \mathfrak{p}] \subset \mathfrak{p}$ ,  $[\mathfrak{p}, \mathfrak{p}] \subset \mathfrak{t}$  and then the Cartan decomposition of  $sl(3) = \mathfrak{t} \oplus \mathfrak{p} = \langle e_3 - e_5, e_4 - e_7, e_6 - e_8 \rangle \oplus \langle e_1, e_2, e_3 + e_5, e_4 + e_7, e_6 + e_8 \rangle$ . Because in this case, the decomposition defined by  $\mathfrak{t}, \mathfrak{p}$  defines a symmetric pair, we can easily construct the Cartan Involution to be

$$\Theta(e_1) = -e_1, \quad \Theta(e_2) = -e_2, \quad \Theta(e_3) = -e_5, \quad \Theta(e_4) = -e_7,$$

$$\Theta(e_5) = -e_3, \quad \Theta(e_6) = -e_8, \quad \Theta(e_7) = -e_4, \quad \Theta(e_8) = -e_6.$$

**Example 3.3.2.**  $sp(2, 2)$

For the next example, we look at the 10-dimensional real Lie algebra  $sp(2, 2)$ . The matrices that form the basis for the standard representation of  $sp(2, 2)$  are

$$e_1 = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 & I & 0 & 0 \\ -I & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & -I & 0 \end{bmatrix},$$

$$e_4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad e_5 = \begin{bmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad e_6 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

$$e_7 = \begin{bmatrix} 0 & 0 & 0 & I \\ 0 & 0 & I & 0 \\ 0 & -I & 0 & 0 \\ -I & 0 & 0 & 0 \end{bmatrix}, \quad e_8 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -I \end{bmatrix}, \quad e_9 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix},$$

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

Then using the theorem 3.3.2, we need to find the Hermitian and skew Hermitian matrices to find the Cartan decomposition. We start by finding the compact part  $\mathfrak{k}$  of the Cartan decomposition by finding the skew Hermitian matrices. These are

$$\begin{aligned} e_1 &= \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & e_4 &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ e_5 &= \begin{bmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & e_8 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -I \end{bmatrix}, \\ e_9 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, & e_{10} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \end{bmatrix}. \end{aligned}$$

The positive part of the Cartan decomposition  $\mathfrak{p}$  is the span of the Hermitian matrices. These are

$$e_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 & I & 0 & 0 \\ -I & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & -I & 0 \end{bmatrix},$$

$$e_6 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad e_7 = \begin{bmatrix} 0 & 0 & 0 & I \\ 0 & 0 & I & 0 \\ 0 & -I & 0 & 0 \\ -I & 0 & 0 & 0 \end{bmatrix}.$$

the Cartan decomposition defined by this representation is therefore  $sp(2, 2) = \mathfrak{t} \oplus \mathfrak{p} = \langle e_1, e_4, e_5, e_8, e_9, e_{10} \rangle \oplus \langle e_2, e_3, e_6, e_7 \rangle$ . By using Theorem 3.3.1 the corresponding Cartan Involution is found to be

$$\begin{aligned} \Theta(e_1) &= e_1, & \Theta(e_2) &= -e_2, & \Theta(e_3) &= -e_3, & \Theta(e_4) &= e_4, & \Theta(e_5) &= e_5, \\ \Theta(e_6) &= -e_6, & \Theta(e_7) &= -e_7, & \Theta(e_8) &= e_8, & \Theta(e_9) &= e_9, & \Theta(e_{10}) &= e_{10}. \end{aligned}$$

**Definition 3.3.4.**  $\Theta$  Stable Cartan Subalgebra

A Cartan subalgebra is said to be  $\Theta$ -stable if  $\Theta(\mathfrak{h}) = \mathfrak{h}$  with respect to the Cartan involution  $\Theta$ .

**Example 3.3.3.**  $sp(2, 2)$

For the previous example,  $\Theta(\langle e_1, e_8 \rangle) = \langle e_1, e_8 \rangle$ .

**Definition 3.3.5.** Noncompact Root

An imaginary root  $\beta$  is said to be noncompact if  $\mathfrak{g}_\beta \cap \mathfrak{t} = \emptyset$  and  $\mathfrak{g}_\beta \cap \mathfrak{p} \subseteq \mathfrak{p}$ .

Note: Not compact is different than noncompact.

**Definition 3.3.6.** Compact Root

An imaginary root  $\alpha$  is said to be compact if  $\mathfrak{g}_\alpha \cap \mathfrak{p} = \emptyset$  and  $\alpha \cap \mathfrak{t} \subseteq \mathfrak{t}$ .

**Example 3.3.4.**  $sp(2, 2)$

Let's look at the 10 dimensional simple Lie algebra  $sp(2, 2)$ . A Cartan subalgebra of  $sp(2, 2)$  is  $\mathfrak{h} = \langle e_1, e_8 \rangle$ . The Cartan decomposition of this Lie algebra is  $sp(2, 2) = \mathfrak{t} \oplus \mathfrak{p}$ , where

$$\mathfrak{t} = \langle e_1, e_4, e_5, e_8, e_9, e_{10} \rangle \text{ and } \mathfrak{p} = \langle e_2, e_3, e_6, e_7 \rangle.$$

One of the positive roots in the root space decomposition for this Cartan subalgebra is  $[-I, I]^T$  where  $\mathfrak{g}_{[-I, I]^T} = \langle e_2 + Ie_3 \rangle$ . The root space of this root is completely contained in the positive part  $\mathfrak{p}$  of the Cartan decomposition. Therefore, this root is noncompact.

Another Cartan subalgebra of  $sp(2, 2)$  is  $\mathfrak{h}_1 = \langle e_1 - e_8, e_6 \rangle$ . One of the positive roots in the root space decomposition for  $\mathfrak{h}_1$  is  $[2I, 0]^T$  where  $\mathfrak{g}_{[2I, 0]^T} = e_4 - Ie_5 - e_9 - Ie_{10}$ . The root space of this root is completely contained in the compact part  $\mathfrak{t}$  of the Cartan decomposition. Therefore, this root is compact.

**Example 3.3.5.**  $su(4, 2)$

Let's look at the 35 dimensional simple Lie algebra  $su(4, 2)$ . A Cartan subalgebra of  $su(4, 2)$  is  $\mathfrak{h} = \langle e_1, e_2, e_3, e_4, e_5 \rangle$ . The Cartan decomposition defined by this representation of this Lie algebra is

$$\begin{aligned} \mathfrak{t} = & \langle e_1, e_2, e_3, e_6 - e_8, e_7 + e_{13}, e_9 + e_{14}, e_{10}, e_{11} + e_{16}, e_{12} + e_{17}, e_{15} + e_{18}, e_{19} + e_{26}, \\ & e_{20} + e_{27}, e_{21} + e_{23}, e_{22} + e_{30}, e_{24} + e_{31}, e_{25}, e_{28} + e_{33}, e_{29} + e_{34}, e_{32} + e_{35} \rangle \\ \mathfrak{p} = & \langle e_4, e_5, e_6 + e_8, e_7 - e_{13}, e_9 - e_{14}, e_{11} - e_{16}, e_{12} - e_{17}, e_{15} - e_{18}, e_{19} - e_{26}, \\ & e_{20} - e_{27}, e_{21} - e_{23}, e_{22} - e_{30}, e_{24} - e_{31}, e_{28} - e_{33}, e_{29} - e_{34}, e_{32} - e_{35} \rangle. \end{aligned}$$

The positive roots  $\alpha_+$  with their corresponding root spaces  $\mathfrak{g}_{\alpha_+}$  are

$$\begin{aligned} \mathfrak{g}_{[0, 0, 0, 0, 2]^T} &= \langle e_{27} \rangle, & \mathfrak{g}_{[I, 2I, I, 0, 1]^T} &= \langle e_{14} - Ie_{31} \rangle, \\ \mathfrak{g}_{[2I, I, -I, -1, 0]^T} &= \langle e_{15} - Ie_{32} \rangle, & \mathfrak{g}_{[I, -I, 0, 1, 1]^T} &= \langle e_{13} + Ie_{30} \rangle, \\ \mathfrak{g}_{[I, 2I, -I, 0, -1]^T} &= \langle e_{12} - Ie_{29} \rangle, & \mathfrak{g}_{[I, 2I, -I, 0, 1]^T} &= \langle e_{17} - Ie_{34} \rangle, \\ \mathfrak{g}_{[I, -I, 0, -1, -1]^T} &= \langle e_7 + Ie_{22} \rangle, & \mathfrak{g}_{[2I, I, -I, 1, 0]^T} &= \langle e_{18} - Ie_{35} \rangle, \\ \mathfrak{g}_{[2I, I, I, -1, 0]^T} &= \langle e_{11} - Ie_{28} \rangle, & \mathfrak{g}_{[I, -I, 0, -1, 1]^T} &= \langle e_6 + Ie_{21} \rangle, \\ \mathfrak{g}_{[0, 0, 2I, 0, 0]^T} &= \langle e_{10} - Ie_{25} \rangle, & \mathfrak{g}_{[2I, I, I, 1, 0]^T} &= \langle e_{16} - Ie_{33} \rangle, \\ \mathfrak{g}_{[I, 2I, I, 0, -1]^T} &= \langle e_9 - Ie_{24} \rangle, & \mathfrak{g}_{[0, 0, 0, 2, 0]^T} &= \langle e_{26} \rangle, \\ & & \mathfrak{g}_{[I, -I, 0, 1, -1]^T} &= \langle e_8 - Ie_{23} \rangle. \end{aligned}$$

In the root spaces of the positive roots, the only one whose intersection with  $\mathfrak{p}$  is empty and is completely contained in  $\mathfrak{t}$  is the root space  $\mathfrak{g}_{[0, 0, 2I, 0, 0]^T} = e_{10} - Ie_{25}$ . So  $[0, 0, 2I, 0, 0]^T$  is the only compact root for  $su(4, 2)$ . None of these positive roots  $\alpha_+$  have a root space space that is completely contained in  $\mathfrak{p}$ , so there are not any noncompact roots in this root space decomposition.

**Definition 3.3.7.** Maximally Noncompact Cartan Subalgebra

Let  $\mathfrak{h}_0$  be a  $\Theta$  stable Cartan subalgebra of a Lie algebra  $\mathfrak{g}_0$ . Then we can say that  $h_0$

is a maximally noncompact Cartan subalgebra if and only if there are no noncompact imaginary roots.

**Definition 3.3.8.** Satake Associates

For a given Cartan subalgebra  $\mathfrak{h}$ , let  $\mathfrak{g} = \mathfrak{h} \oplus_{\alpha \in \Delta} \mathfrak{g}_\alpha$  be the root space decomposition for a noncompact real simple Lie algebra. Let  $\alpha_+$  be a set of positive roots,  $\alpha_c$  the compact roots,  $\alpha_s$  be the simple roots and  $\alpha_{cs}$  the compact simple roots. We choose the positive roots to be closed under complex conjugation. Then for each root  $\beta \in \alpha_s/\alpha_{cs}$ , if there is a unique root  $\beta' \in \alpha_s/\alpha_{cs}$  such that  $\bar{\beta} - \beta' \in \langle \alpha_s \rangle$ , then the root  $\beta'$  is called the Satake associate of  $\beta$ .

**Example 3.3.6.**

The simple Lie algebra  $so^*(10)$  has the simple roots:

$$\alpha_1 = \begin{bmatrix} 0 \\ 0 \\ 2I \\ 0 \\ 0 \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 1 \\ -1 \\ -I \\ -I \\ 0 \end{bmatrix}, \quad \alpha_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2I \\ 0 \end{bmatrix}, \quad \alpha_4 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -I \\ -I \end{bmatrix}, \quad \alpha_5 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -I \\ I \end{bmatrix}.$$

Of these,  $\alpha_1$  and  $\alpha_3$  are compact. Then we calculate

$$\bar{\alpha}_2 - \alpha_2 = \begin{bmatrix} 1 \\ -1 \\ I \\ I \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ -1 \\ -I \\ -I \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2I \\ 2I \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2I \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2I \\ 0 \end{bmatrix} = \alpha_1 + \alpha_3.$$

So the Satake associate of  $\alpha_2$  is itself. Next we see that

$$\bar{\alpha}_4 - \alpha_5 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ I \\ I \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 0 \\ -I \\ I \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2I \\ 0 \end{bmatrix} = \alpha_3.$$

So the Satake associate of  $\alpha_4$  is  $\alpha_5$ .

This completes our review of the basic theory and concepts of semi-simple Lie algebras. The reader should have the necessary information for understanding the process of the Cayley transform. In the next chapter, we'll discuss the seventeen different real simple Lie algebra types and their corresponding Satake diagrams.



## Chapter 4

### Real Simple Lie Algebras

In this chapter we will describe the seventeen types of classical real simple Lie algebras. For each type, we give the standard matrix representation, the Satake diagram, and an example. The seventeen real types are organized according to their classification as simple complex Lie algebras.

$A_n$	$sl(n, \mathbb{R})$ $sl(m, \mathbb{H})$ $su(p, q), p > q$ $su(p, p)^*$ $su(n)$
$B_n$	$so(p, q)$ $so(n), n$ is odd
$C_n$	$sp(n, \mathbb{R})$ $sp(p, p)$ $sp(n)$
$D_n$	$so(n, n)$ $so(p, q), p + q$ is even $so(2n)$ $so^*(2n), n$ is even $so^*(2n), n$ is odd

\*We distinguish  $su(p, q)$  from  $su(p, p)$  because the Satake diagrams have a different pattern when  $p = q$ .

## 4.1 $A_n$ : The Special Linear and Special Unitary Lie Algebras

### 4.1.1 $SL(n)$ : Special Linear Group

The special linear group,  $SL(n)$ , is the set of  $n \times n$  real-valued matrices with determinant 1. The group operations are ordinary matrix multiplication and matrix inversion. If  $A(t) \in SL(n)$  is a smooth curve with  $A(0) = I$ , then any non-singular matrix  $A$  satisfies  $AA^{-1} = A^{-1}A = I$ . Then the well known formula for the derivative of the determinant of a matrix  $A(t)$  gives [3]:  $\frac{d}{dt}|_{t=0} \det(A(t)) = \text{trace}(A'(0))$ . Therefore the special linear Lie algebra is the set of all trace-free  $n \times n$  matrices.

#### 4.1.1.1 $sl(n, R)$

The dimension of the Lie algebra  $sl(n, R)$  is  $n^2 - 1$ . A basis for  $sl(n, R)$  is a set of linearly independent  $n \times n$  trace-free matrices with real entries. The number of simple roots is  $n - 1$ . All of these simple roots will be real roots. Therefore no distinct Satake associates will occur in  $sl(n, R)$ . So, red arrows will not appear on the Satake diagrams of  $sl(n, R)$ . The Satake diagram for  $sl(n)$  will have  $n - 1$  number of white dots and will look like this:



#### Example 4.1.1. $sl(3)$ :

A basis for the standard representation for  $sl(3)$  is defined by the following trace-free matrices:

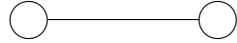
$$\begin{aligned}
 e_1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} & e_2 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} & e_3 &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & e_4 &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
 e_5 &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & e_6 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} & e_7 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} & e_8 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.
 \end{aligned}$$

The Cartan decomposition is  $sl(3) = \mathfrak{t} \oplus \mathfrak{p}$ , where  $\mathfrak{t} = \langle e_3 - e_5, e_4 - e_7, e_6 - e_8 \rangle$ , and  $\mathfrak{p} = \langle e_1, e_2, e_3 + e_5, e_4 + e_7, e_6 + e_8 \rangle$ . Therefore,  $\langle e_1, e_2 \rangle$  is a maximally noncompact Cartan

subalgebra with simple roots

$$\alpha_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Because both of these roots have real entries, the Satake diagram will have two white dots. The roots are not complex conjugates of each other, therefore there are no red arrows. So the Satake diagram will look like this:



#### 4.1.1.2 $sl(m, \mathbb{H})$ also denoted as $su^*(n)$

The Lie algebra  $sl(m, \mathbb{H})$  is the set of quaternionic trace-free  $n \times n$  matrices, where  $2m = n$ . We look at these as complex matrices. The relationship between quaternionic matrices and complex matrices is as follows [2]. Writing an element  $v \in \mathbb{H}^m$  as  $v = a + bi + cj + dk$  with  $a, b, c, d \in \mathbb{R}^n$ , we can define  $\psi : \mathbb{H}^m \rightarrow \mathbb{C}^{2n}$  by  $\psi(v) = (a + bi, c + di)$ . The map  $v \rightarrow (z_1, z_2)$  induces an  $\mathbb{R}$  linear isomorphism  $\mathbb{H}^m \rightarrow \mathbb{C}^{2n}$  that preserves the bracket as follows:

Let  $X \in sl(m, \mathbb{H})$  where  $X = Y + Ui + Wj + Vk$ . Set  $A = Y + Ui$  and  $B = W + Vi$  and  $\psi(X) = \begin{bmatrix} A & B \\ -\bar{B} & \bar{A} \end{bmatrix}$ . We will check that  $\psi$  is a Lie algebra isomorphism, i.e.  $\psi(X_1X_2 - X_2X_1) = [\psi(X_1), \psi(X_2)]$ . Then,

$$\begin{aligned} & X_1X_2 - X_2X_1 = \\ &= (Y_1 + U_1i + W_1j + V_1k)(Y_2 + U_2i + W_2j + V_2k) - (Y_2 + U_2i + W_2j + V_2k)(Y_1 + U_1i + W_1j + V_1k) = \\ &= (Y_1Y_2 - U_1U_2 - W_1W_2 - V_1V_2 - Y_2Y_1 + U_2U_1 + W_2W_1 + V_2V_1) + \\ &+ (Y_1U_2 + U_1Y_2 + W_1V_2 - V_1W_2 - Y_2U_1 - U_2Y_1 - W_2V_1 + V_2W_1)i + \\ &+ (Y_1W_2 - U_1V_2 + W_1Y_2 + V_1U_2 - Y_2W_1 + U_2V_1 - W_2Y_1 - V_2U_1)j + \\ &+ (Y_1V_2 + U_1W_2 - W_1U_2 + V_1Y_2 - Y_2V_1 - U_2W_1 + W_2U_1 - V_2Y_1)k. \end{aligned}$$

Then  $\psi(X_1X_2 - X_2X_1) = \begin{bmatrix} A' & B' \\ -\bar{B}' & \bar{A}' \end{bmatrix}$ , where

$$A' = (Y_1Y_2 - U_1U_2 - W_1W_2 - V_1V_2 - Y_2Y_1 + U_2U_1 + W_2W_1 + V_2V_1) + \\ + (Y_1U_2 + U_1Y_2 + W_1V_2 - V_1W_2 - Y_2U_1 - U_2Y_1 - W_2V_1 + V_2W_1)i,$$

$$B' = (Y_1W_2 - U_1V_2 + W_1Y_2 + V_1U_2 - Y_2W_1 + U_2V_1 - W_2Y_1 - V_2U_1) + \\ + (Y_1V_2 + U_1W_2 - W_1U_2 + V_1Y_2 - Y_2V_1 - U_2W_1 + W_2U_1 - V_2Y_1)i.$$

Then if  $A_1 = Y_1 + U_1i$ ,  $A_2 = Y_2 + U_2i$ ,  $B_1 = W_1 + V_1i$ ,  $B_2 = W_2 + V_2i$ :

$$A' = A_1A_2 - B_1\bar{B}_2 - A_2A_1 + B_2\bar{B}_1$$

$$B' = A_1B_2 + B_1\bar{A}_2 - A_2B_1 - B_2\bar{A}_1.$$

Then we can see that

$$\begin{aligned} \psi(X_1X_2 - X_2X_1) &= \begin{bmatrix} A' & B' \\ -\bar{B}' & \bar{A}' \end{bmatrix} = \\ &= \begin{bmatrix} A_1A_2 - B_1\bar{B}_2 - A_2A_1 + B_2\bar{B}_1 & A_1B_2 + B_1\bar{A}_2 - A_2B_1 - B_2\bar{A}_1 \\ -\bar{A}_1\bar{B}_2 - \bar{B}_1A_2 + \bar{A}_2\bar{B}_1 + \bar{B}_2A_1 & \bar{A}_1\bar{A}_2 - \bar{B}_1B_2 - \bar{A}_2\bar{A}_1 + \bar{B}_2B_1 \end{bmatrix} = \\ &= \begin{bmatrix} A_1 & B_1 \\ -\bar{B}_1 & \bar{A}_1 \end{bmatrix} \begin{bmatrix} A_2 & B_2 \\ -\bar{B}_2 & \bar{A}_2 \end{bmatrix} - \begin{bmatrix} A_2 & B_2 \\ -\bar{B}_2 & \bar{A}_2 \end{bmatrix} \begin{bmatrix} A_1 & B_1 \\ -\bar{B}_1 & \bar{A}_1 \end{bmatrix}. \end{aligned}$$

Thus this linear isomorphism  $\psi$  preserves the bracket. Then the following characterizes  $sl(m, \mathbb{H})$  as the set of linearly independent trace-free complex matrices of the block form:

$$sl(m, \mathbb{H}) = su^*(n) = \left\{ \left[ \begin{array}{cc} A & B \\ -\bar{B} & \bar{A} \end{array} \right] \mid \begin{array}{l} A \text{ is a } n \times n \text{ complex matrix,} \\ B \text{ is a } n \times n \text{ complex matrix,} \\ tr(A) + tr(\bar{A}) = 0 \end{array} \right\}$$

The dimension of the Lie algebra  $sl(m, H), su^*(n)$  is  $n^2 - 1$  where  $2m = n$ . The number of simple roots is  $l = 2m - 1$ . There are  $\frac{l-1}{2}$  real or complex simple roots, and  $\frac{l+1}{2}$  imaginary simple roots. Since  $n$  is even, the number of simple roots will always be odd. These simple roots are a mixture of imaginary, complex, and real and no red arrows will appear on the Satake diagrams of  $sl(m, \mathbb{H})$ .

The simple roots will start with a imaginary root and alternate with a complex or real root until ending with an imaginary root. So the Satake diagram will look like this:



**Example 4.1.2.**  $sl(2, H), su^*(4)$ :

A basis for the standard representation for  $sl(2, H), su^*(4)$  is the following trace-free matrices:

$$\begin{aligned} e_1 &= \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_2 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -I \end{bmatrix} & e_3 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} & e_4 &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ e_5 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} & e_6 &= \begin{bmatrix} 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -I \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_7 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 \end{bmatrix} & e_8 &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ e_9 &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_{10} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} & e_{11} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} & e_{12} &= \begin{bmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ -I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

$$e_{13} = \begin{bmatrix} 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad e_{14} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \end{bmatrix} \quad e_{15} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \end{bmatrix}.$$

The Cartan decomposition is  $su^*(4) = \mathfrak{t} \oplus \mathfrak{p}$  where,  $\mathfrak{t} = \langle e_1, e_2, e_4 - e_5, e_6 + e_7, e_8, e_9 + e_{10}, e_{11}, e_{12}, e_{13} + e_{14}, e_{15} \rangle$ , and  $\mathfrak{p} = \langle e_3, e_4 + e_5, e_6 - e_7, e_9 - e_{10}, e_{13} - e_{14} \rangle$ . From the basis of the standard representation above we can see that  $e_1, e_2$  and  $e_3$  are diagonal matrices. Then,  $\langle e_1, e_2, e_3 \rangle$  is a Cartan subalgebra with the simple roots:

$$\alpha_1 = \begin{bmatrix} 0 \\ 2I \\ 0 \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 2 \\ -I \\ -I \end{bmatrix}, \quad \alpha_3 = \begin{bmatrix} 0 \\ 0 \\ 2I \end{bmatrix}.$$

The root spaces of the imaginary roots are  $E_{\alpha_1} = e_{11} - Ie_{15}$  and  $E_{\alpha_3} = e_8 - Ie_{12}$  and these both lie completely in the compact part  $\mathfrak{t}$  of the Cartan decomposition. Then because there isn't an imaginary noncompact root the Cartan subalgebra is maximally noncompact.

The order of these roots does matter, and we will discuss how to get that order in Chapter 5. The first and third roots are imaginary and the middle one is complex. This means that we will have a black dot, a white dot, and then another black dot. Because the first and third roots are not complex conjugates of each other, there are no red arrows. So the Satake diagram will look like this:



#### 4.1.2 $SU(p, q)$ : Special Unitary Group

Let  $Q$  be the diagonal matrix  $\begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix}$ . Then, the group  $SU(p, q)$  is the group of  $n \times n$  complex matrices that satisfy  $MQM^\dagger = Q$  and  $\det(M) = 1$ , where  $n = p + q$  and  $\dagger$  represents the conjugate transpose of the matrix. If  $M(t) \in SU(p, q)$  is a smooth curve with  $M(0) = I$ , then for any non-singular matrix  $M$  we can take the derivative of the above relationship with respect to  $t$ . Then, evaluating for  $t = 0$  gives us the relationship for the corresponding Lie algebra.  $Q \cdot X + X^\dagger \cdot Q = 0$ , and  $\text{trace}(X) = 0$ .

Let  $X$  be the block matrix  $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ . Then,  $Q \cdot X + X^\dagger \cdot Q = 0$  gives

$$\begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} \cdot \begin{bmatrix} A & B \\ C & D \end{bmatrix} + \begin{bmatrix} A^\dagger & C^\dagger \\ B^\dagger & D^\dagger \end{bmatrix} \cdot \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} = 0$$

$$\begin{bmatrix} A & B \\ -C & -D \end{bmatrix} + \begin{bmatrix} A^\dagger & -C^\dagger \\ B^\dagger & -D^\dagger \end{bmatrix} = 0.$$

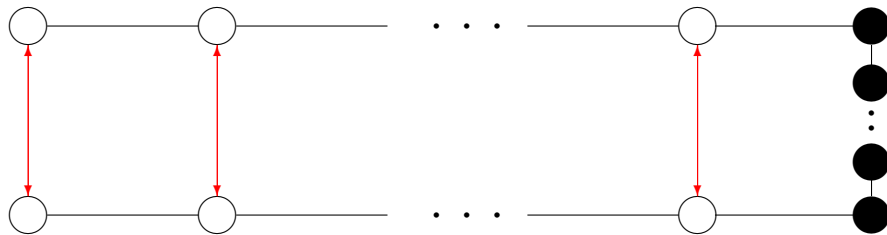
This leads to  $A = -A^\dagger, D = -D^\dagger, C = B^\dagger$ . Finally,

$$su(p, q) = \left\{ \begin{array}{l} \begin{bmatrix} A & B \\ B^\dagger & D \end{bmatrix} \mid \begin{array}{l} A \text{ is a } q \times q \text{ skew hermitian matrix,} \\ D \text{ is a } p \times p \text{ skew hermitian matrix and} \\ B \text{ is a } q \times p \text{ matrix} \\ tr(A) + tr(D) = 0 \end{array} \right\}$$

The case where  $q = 0$  is a special case of  $su(p, q)$  called  $su(n)$ .

**4.1.2.1**  $su(p, q), p > q$

The dimension of the Lie algebra  $su(p, q)$  is  $(p + q)^2 - 1$ , where  $p = l + 1 - q$  and  $l$  is the number of simple roots. The simple roots will be a mixture of imaginary and complex/real. There are  $2q$  complex simple roots, and  $l - 2q$  imaginary simple roots. In the ordering of the simple roots in the Satake diagram, the simple roots will have half the complex simple roots followed by all of the imaginary roots and ending with the other half of the complex simple roots. Every complex simple root will have a Satake associate in the set of simple roots. So the Satake diagram will look like this:



**Example 4.1.3.**  $su(3, 1)$ :

A basis for the the standard representation for  $su(3, 1)$  are the following trace-free matrices:

$$\begin{aligned}
e_1 &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_2 &= \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_3 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_4 &= \begin{bmatrix} 0 & I & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
e_5 &= \begin{bmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_6 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_7 &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} & e_8 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \\
e_9 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} & e_{10} &= \begin{bmatrix} 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -I & 0 & 0 & 0 \end{bmatrix} & e_{11} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \end{bmatrix} & e_{12} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & -I & 0 \end{bmatrix} \\
e_{13} &= \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_{14} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & -I & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_{15} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & -I \end{bmatrix}.
\end{aligned}$$

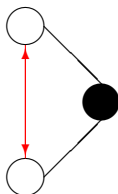
The Cartan decomposition is  $su(3, 1) = \mathfrak{t} \oplus \mathfrak{p}$  where  $\mathfrak{t} = \langle e_1, e_2, e_3, e_4, e_5, e_6, e_{13}, e_{14}, e_{15} \rangle$ , and  $\mathfrak{p} = \langle e_7, e_8, e_9, e_{10}, e_{11}, e_{12} \rangle$ . We can see that  $\langle e_1, e_9, e_{13} + 2e_{14} + e_{15} \rangle$  is a Cartan subalgebra and with simple roots:

$$\alpha_1 = \begin{bmatrix} 1 \\ -I \\ -I \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 0 \\ 0 \\ 2I \end{bmatrix}, \quad \alpha_3 = \begin{bmatrix} 1 \\ I \\ -I \end{bmatrix}.$$

The order of these roots does matter, and we will discuss how to get that order in Chapter 5. The root space of the only imaginary root is  $E_{\alpha_2} = e_4 - Ie_{13}$ . This is completely contained in the compact part  $\mathfrak{t}$  of the Cartan decomposition. Thus, our



Cartan subalgebra is maximally noncompact. The first and third roots are complex and the middle one is imaginary. This means that we will have a white dot, a black dot, and then another white dot. Since  $\alpha_2$  is a compact root and  $\overline{\alpha_1} - \alpha_3 = [1, I, I]^T - [1, I, -I]^T = [0, 0, 2I]^T = \alpha_2$  then  $\alpha_3$  is the Satake associate of  $\alpha_1$ . So the Satake diagram will look like this:

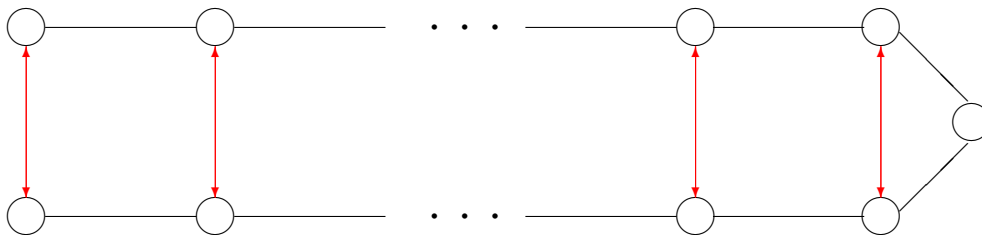


**4.1.2.2**  $su(p, p)$

A basis for  $su(p, p)$  is a list of linearly independent complex matrices of the block form:

$$su(p, p) = \left\{ \left[ \begin{array}{cc} A & B \\ B^\dagger & D \end{array} \right] \mid \begin{array}{l} A \text{ is a } p \times p \text{ skew hermitian matrix,} \\ D \text{ is a } p \times p \text{ skew hermitian matrix and} \\ B \text{ is a } p \times p \text{ matrix} \\ \text{tr}(A) + \text{tr}(D) = 0 \end{array} \right\}$$

The dimension of the Lie algebra  $su(p, p)$  is  $(2p)^2 - 1$ , where  $l = 2p - 1$  and  $l$  is the number of simple roots. All of the simple roots are real or complex. In fact, there is exactly one real root, and the rest are complex. Every complex simple root has its Satake associate in the set of simple roots. So the Satake diagram will look like this:



**Example 4.1.4.**  $su(2, 2)$ :

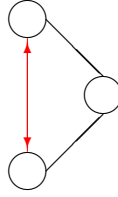
A basis for the standard representation for  $su(2, 2)$  are the following trace-free matrices:

$$\begin{aligned}
e_1 &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_2 &= \begin{bmatrix} 0 & I & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_3 &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_4 &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \\
e_5 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_6 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} & e_7 &= \begin{bmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ -I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_8 &= \begin{bmatrix} 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -I & 0 & 0 & 0 \end{bmatrix} \\
e_9 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & -I & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_{10} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \end{bmatrix} & e_{11} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} & e_{12} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & I & 0 \end{bmatrix} \\
e_{13} &= \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_{14} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & -I & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_{15} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & -I \end{bmatrix} .
\end{aligned}$$

The Cartan decomposition is:  $su(2, 2) = \mathfrak{t} \oplus \mathfrak{p}$  where,  $\mathfrak{t} = \langle e_1, e_2, e_{11}, e_{12}, e_{13}, e_{14}, e_{15} \rangle$ , and  $\mathfrak{p} = \langle e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10} \rangle$ . We see that  $\langle e_1 + e_{11}, e_3 + e_6, e_9 - e_8 \rangle$  is a Cartan subalgebra with simple roots:

$$\alpha_1 = \begin{bmatrix} 0 \\ 2 \\ 2I \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 2 \\ -2 \\ 0 \end{bmatrix}, \quad \alpha_3 = \begin{bmatrix} 0 \\ 2 \\ -2I \end{bmatrix}.$$

Since none of these roots are imaginary we can say that this Cartan subalgebra is maximally noncompact. The order of these roots does matter, and we will discuss how to get that order in Chapter 5. The first and third roots are complex and the middle one is real. This means that we will have three white dots. The simple root  $\alpha_2$  is not compact, so because  $\alpha_1$  and  $\alpha_3$  are complex conjugates,  $\alpha_3$  is the Satake associate of  $\alpha_1$ . The Satake diagram will look like this:



**4.1.2.3**  $su(n)$

A basis for  $su(n)$  is a list of linearly independent  $n \times n$  skew hermitian trace-free matrices with complex entries. The dimension of the Lie algebra  $su(n)$  is  $n^2 - 1$ , where  $l = n - 1$  and  $l$  is the number of simple roots. All of the simple roots are imaginary. In fact, because the basis for  $su(n)$  are skew hermitian matrices, the entire basis will be contained in the compact part  $\mathfrak{t}$  of the Cartan decomposition. This means that every one of the simple roots will be compact. Thus, there will not be any Satake associates for  $su(n)$ . So the Satake diagram will look like this:



**Example 4.1.5.**  $su(4)$ :

A basis for the standard representation for  $su(4)$  are the following trace-free skew hermitian matrices:

$$\begin{aligned}
 e_1 &= \begin{bmatrix} -I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I \end{bmatrix} &
 e_2 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I \end{bmatrix} &
 e_3 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} &
 e_4 &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 e_5 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} &
 e_6 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} &
 e_7 &= \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} &
 e_8 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \\
 e_9 &= \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} &
 e_{10} &= \begin{bmatrix} 0 & I & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} &
 e_{11} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} &
 e_{12} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & I & 0 \end{bmatrix}
 \end{aligned}$$

$$e_{13} = \begin{bmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad e_{14} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \end{bmatrix} \quad e_{15} = \begin{bmatrix} 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \end{bmatrix}.$$

The Cartan decomposition is  $su(4) = \mathfrak{t} \oplus \mathfrak{p}$ , where  $\mathfrak{t} = \langle e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15} \rangle$  and  $\mathfrak{p} = \langle \rangle$ . We see that  $\mathfrak{h} = \langle e_1, e_2, e_3 \rangle$  is a Cartan subalgebra with simple roots:

$$\alpha_1 = \begin{bmatrix} I \\ I \\ 2I \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 0 \\ I \\ -I \end{bmatrix}, \quad \alpha_3 = \begin{bmatrix} I \\ -I \\ 0 \end{bmatrix}.$$

Because  $\mathfrak{p}$  is empty, this Cartan subalgebra must be maximally noncompact. All of the roots are imaginary. This means that we will have three black dots. So the Satake diagram will look like this:



## 4.2 $B_n$ : Odd Dimensional Special Orthogonal Lie Algebras

### 4.2.1 $SO(p, q) : p + q$ is odd: Odd Dimensional Special Orthogonal Group

Let  $Q$  be the diagonal matrix  $\begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix}$ . Then the group  $SO(p, q)$  is the group of  $n \times n$  real-valued matrices  $M$  that satisfy  $M^T \cdot Q \cdot M = Q$  and  $\det(M) = 1$ , where  $n = p + q$  and  $T$  represents the transpose of the matrix. If  $M(t) \in SO(p, q)$  is a smooth curve with  $M(0) = I$ , then for any non-singular matrix  $M$  we can take the derivative of the above relationship with respect to  $t$ . Then, evaluating for  $t = 0$  gives us the relationship for the corresponding Lie algebra

$$M^T \cdot Q \cdot \dot{M} + \dot{M}^T \cdot Q \cdot M = 0$$

$$Q \cdot \dot{M} + \dot{M}^T \cdot Q = 0.$$

Let  $\dot{M} = X$  then,

$$Q \cdot X + X^T \cdot Q = 0.$$

Then let  $X$  be the block matrix  $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ . Then,  $Q \cdot X + X^T \cdot Q = 0$  gives:

$$\begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} \cdot \begin{bmatrix} A & B \\ C & D \end{bmatrix} + \begin{bmatrix} A^T & C^T \\ B^T & D^T \end{bmatrix} \cdot \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} = 0,$$

$$\begin{bmatrix} A & B \\ -C & -D \end{bmatrix} + \begin{bmatrix} A^T & -C^T \\ B^T & -D^T \end{bmatrix} = 0.$$

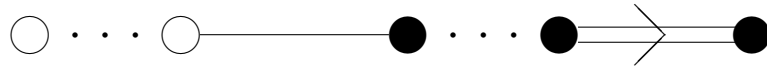
This leads to  $A = -A^T, D = -D^T, C = B^T$ . Finally,

$$so(p, q) = \left\{ \begin{array}{l} \begin{bmatrix} A & B \\ B^T & D \end{bmatrix} \mid \begin{array}{l} A \text{ is a } p \times p \text{ skew symmetric matrix,} \\ D \text{ is a } q \times q \text{ skew symmetric matrix and} \\ B \text{ is a } p \times q \text{ matrix} \end{array} \end{array} \right\}$$

#### 4.2.2 $so(p, q), p + q$ is odd

$$so(p, q) = \left\{ \begin{array}{l} \begin{bmatrix} A & B \\ B^T & D \end{bmatrix} \mid \begin{array}{l} A \text{ is a } p \times p \text{ skew symmetric matrix,} \\ D \text{ is a } q \times q \text{ skew symmetric matrix and} \\ B \text{ is a } p \times q \text{ matrix} \end{array} \end{array} \right\}$$

The dimension of the Lie algebra  $so(p, q)$  is  $\frac{n(n-1)}{2}$ , where  $l = \frac{p+q-1}{2}$  and  $l$  is the number of simple roots. There are  $p$  real/complex simple roots and  $l - p$  imaginary simple roots. Then the Satake diagram will look like this:



**Example 4.2.1.** Let's look at  $so(4, 1)$ .

A basis for the standard representation for  $so(4, 1)$  are the following trace-free matrices:

$$\begin{aligned}
e_1 &= \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_2 &= \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_3 &= \begin{bmatrix} 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
e_4 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_5 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_6 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
e_7 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} & e_8 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} & e_9 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \\
e_{10} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} .
\end{aligned}$$

The Cartan decomposition is  $so(4, 1) = \mathfrak{t} \oplus \mathfrak{p}$ , where  $\mathfrak{t} = \langle e_1, e_2, e_3, e_4, e_5, e_6 \rangle$ , and  $\mathfrak{p} = \langle e_7, e_8, e_9, e_{10} \rangle$ . We see that  $\langle e_6, e_7 \rangle$  is a Cartan subalgebra with simple roots:

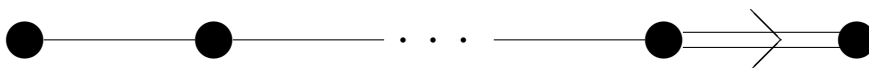
$$\alpha_1 = \begin{bmatrix} 1 \\ -I \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 0 \\ I \end{bmatrix}.$$

The root space of the imaginary root is  $E_{\alpha_2} = e_4 - Ie_5$ , which is completely contained in the compact part  $\mathfrak{t}$  of the Cartan decomposition. Then, our Cartan subalgebra is maximally noncompact. The first simple root is complex and the second is imaginary. This means that we will have one white dot and then one black dot. So the Satake diagram will look like this:



### 4.2.3 $so(n), n$ is odd.

A basis for  $so(n)$  is a list of linearly independent  $n \times n$  skew symmetric trace-free matrices with real entries. The dimension of the Lie algebra  $so(n)$  is  $\frac{n(n-1)}{2}$ , where  $l = \frac{n-1}{2}$  and  $l$  is the number of simple roots. Since all of the matrices that make up the basis for  $so(n)$  are skew symmetric, the Cartan decomposition will only have elements in  $\mathfrak{t}$ , the compact part. All of the simple roots will be imaginary and compact and therefore there will not be any Satake associates. Then the Satake diagram will look like this:



**Example 4.2.2.** Let's look at  $so(5)$ .

A basis for the standard representation for  $so(5)$  are the following trace-free skew symmetric matrices with real entries:

$$\begin{aligned}
 e_1 &= \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_2 &= \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_3 &= \begin{bmatrix} 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 e_4 &= \begin{bmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} & e_5 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_6 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 e_7 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} & e_8 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_9 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}
 \end{aligned}$$

$$e_{10} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

The Cartan decomposition is  $\mathfrak{so}(5) = \mathfrak{t} \oplus \mathfrak{p}$ , where  $\mathfrak{t} = \langle e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10} \rangle$  and  $\mathfrak{p} = \langle \rangle$ . Because the positive part of the Cartan decomposition  $\mathfrak{p}$  is empty,  $\mathfrak{h} = \langle e_1, e_8 \rangle$  is a maximally noncompact Cartan subalgebra with simple roots

$$\alpha_1 = \begin{bmatrix} I \\ -I \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 0 \\ I \end{bmatrix}.$$

They are both imaginary. This means that we will have two black dots. So the Satake diagram will look like this:



### 4.3 $C_n$ : Symplectic Lie Algebras

#### 4.3.1 $Sp(n, \mathbb{R})$ : Symplectic Lie Group

Let  $J$  be the block matrix,  $J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$  where  $I$  is the  $n \times n$  identity matrix.

Then the group  $Sp(n, \mathbb{R})$  is the group of  $n \times n$  matrices that satisfy  $M^T J M = J$ , where  $n = p + q$  and  $T$  represents the transpose of the matrix. If  $M(t) \in Sp(p, q)$  is a smooth curve with  $M(0) = I$ , then for any non-singular matrix  $M$  we can take the derivative of the above relationship with respect to  $t$ . Then, evaluating for  $t = 0$  gives us the relationship for the corresponding Lie algebra.

$$M^T \cdot J \cdot \dot{M} + \dot{M}^T \cdot J \cdot M = 0$$

$$J \cdot \dot{M} + \dot{M}^T \cdot J = 0.$$

$$\text{Let } \dot{M} = X.$$

$$\text{Then, } J \cdot X + X^T \cdot J = 0.$$



**4.3.2**  $sp(n, R)$

Let  $X$  be the block matrix  $X = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ . Then,  $J \cdot X + X^T \cdot J = 0$  gives

$$\begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix} \cdot \begin{bmatrix} A & B \\ C & D \end{bmatrix} + \begin{bmatrix} A^T & C^T \\ B^T & D^T \end{bmatrix} \cdot \begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix} = 0$$

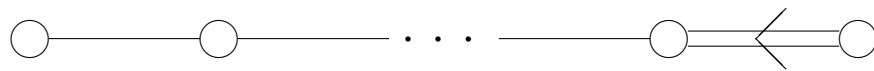
$$\begin{bmatrix} C & D \\ -A & -B \end{bmatrix} + \begin{bmatrix} -C^T & A^T \\ -D^T & B^T \end{bmatrix} = 0.$$

This leads to  $D = -A^T, B = B^T, C = C^T$ .

Finally, a basis for  $sp(n, R)$  are real-valued matrices such that:

$$sp(n, R) = \left\{ \begin{bmatrix} A & B \\ C & -A^T \end{bmatrix} \mid \begin{array}{l} A \text{ is a } n \times n \text{ matrix,} \\ B \text{ is a } n \times n \text{ symmetric matrix and} \\ C \text{ is a } n \times n \text{ symmetric matrix} \end{array} \right\}$$

The dimension of the Lie algebra  $sp(n, R)$  is  $\frac{n(n+1)}{2}$ , where  $l = n/2$  and  $l$  is the number of simple roots. All of the simple roots are real. Then, there will not be any Satake associates for  $sp(n, R)$ . So the Satake diagram will look like this:



**Example 4.3.1.**  $sp(4, R)$ :

A basis for the standard representation for  $sp(4, R)$  are the following trace-free matrices:

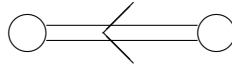
$$e_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad e_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad e_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$\begin{aligned}
e_5 &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_6 &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_7 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_8 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
e_9 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} & e_{10} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.
\end{aligned}$$

The Cartan decomposition is  $sp(4, R) = \mathfrak{t} \oplus \mathfrak{p}$  where,  $\mathfrak{t} = \langle e_2 - e_3, e_5 - e_8, e_6 - e_9, e_7 - e_{10} \rangle$  and  $\mathfrak{p} = \langle e_1, e_2 + e_3, e_4, e_5 + e_8, e_6 + e_9, e_7 + e_{10} \rangle$ . We can see that  $\langle e_1, e_4 \rangle$  is a Cartan subalgebra. Because this is completely contained in the positive and noncompact part  $\mathfrak{p}$  of the Cartan decomposition, this is a maximally noncompact Cartan subalgebra with simple roots

$$\alpha_1 = \begin{bmatrix} I \\ -I \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}.$$

Both of these roots are real. This means that we will have two white dots. So the Satake diagram will look like this:



### 4.3.3 $Sp(p, q)$ : Symplectic Lie Group

Let  $J$  be the block matrix,  $J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$  where  $I$  is the  $n \times n$  identity matrix.

Then the group  $Sp(p, q)$  is the group of  $n \times n$  complex matrices that satisfy  $M^T J M = J$ , where  $n = p + q$  and  $T$  represents the transpose of the matrix. If  $M(t) \in Sp(p, q)$  is a smooth curve with  $M(0) = I$ , then for any non-singular matrix  $M$  we can take the derivative of the above relationship with respect to  $t$ . Then, evaluating for  $t = 0$  gives us the relationship for the corresponding Lie algebra.

$$M^T \cdot J \cdot \dot{M} + \dot{M}^T \cdot J \cdot M = 0$$

$$J \cdot \dot{M} + \dot{M}^T \cdot J = 0.$$

$$\text{Let } \dot{M} = X.$$

$$\text{Then, } J \cdot X + X^T \cdot J = 0.$$

#### 4.3.4 $sp(p, q)$

A basis for  $sp(p, q)$  are complex valued matrices satisfying  $XJ + JX^T = 0$  and  $XQ + QX^\dagger = 0$ . The first of these gives

$$X = \begin{bmatrix} A_1 & A_2 & B_1 & B_2 \\ A_3 & A_4 & B_2^T & B_4 \\ C_1 & C_2 & -A_1^T & -A_3^T \\ C_2^T & C_4 & -A_2^T & -A_4^T \end{bmatrix}$$

where  $B_1, B_4, C_1, C_4$  are symmetric as shown previously. The second, with

$$Q = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & -I \end{bmatrix}$$

gives:

$$\begin{bmatrix} A_1 & A_2 & B_1 & B_2 \\ A_3 & A_4 & B_2^T & B_4 \\ C_1 & C_2 & -A_1^T & -A_3^T \\ C_2^T & C_4 & -A_2^T & -A_4^T \end{bmatrix} \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & -I \end{bmatrix} +$$

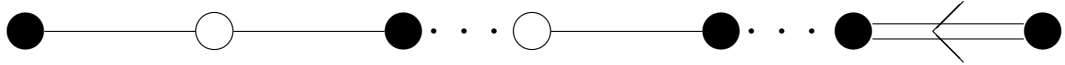
$$\begin{aligned}
& + \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & -I \end{bmatrix} \begin{bmatrix} A_1 & A_2 & B_1 & B_2 \\ A_3 & A_4 & B_2^T & B_4 \\ C_1 & C_2 & -A_1^T & -A_3^T \\ C_2^T & C_4 & -A_2^T & -A_4^T \end{bmatrix}^\dagger = 0 \\
& \begin{bmatrix} A_1 & A_2 & B_1 & B_2 \\ A_3 & A_4 & B_2^T & B_4 \\ C_1 & C_2 & -A_1^T & -A_3^T \\ C_2^T & C_4 & -A_2^T & -A_4^T \end{bmatrix} \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & -I \end{bmatrix} + \\
& + \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & -I \end{bmatrix} \begin{bmatrix} A_1^\dagger & A_3^\dagger & C_1^\dagger & (C_2^t)^\dagger \\ A_2^\dagger & A_4^\dagger & C_2^\dagger & C_4^\dagger \\ B_1^\dagger & (B_2^T)^\dagger & -(A_1^T)^\dagger & -(A_2^T)^\dagger \\ B_2^\dagger & B_4^\dagger & -A_3^\dagger & -(A_4^T)^\dagger \end{bmatrix} = 0 \\
& \begin{bmatrix} A_1 & -A_2 & B_1 & -B_2 \\ A_3 & -A_4 & B_2^T & -B_4 \\ C_1 & -C_2 & -A_1^T & A_3^T \\ C_2^T & -C_4 & -A_2^T & A_4^T \end{bmatrix} + \begin{bmatrix} A_1^\dagger & A_3^\dagger & C_1^\dagger & (C_2^t)^\dagger \\ -A_2^\dagger & -A_4^\dagger & -C_2^\dagger & -C_4^\dagger \\ B_1^\dagger & (B_2^T)^\dagger & -(A_1^T)^\dagger & -(A_2^T)^\dagger \\ -B_2^\dagger & -B_4^\dagger & A_3^\dagger & (A_4^T)^\dagger \end{bmatrix} = 0 \\
& \begin{bmatrix} A_1 & -A_2 & B_1 & -B_2 \\ A_3 & -A_4 & B_2^T & -B_4 \\ C_1 & -C_2 & -A_1^T & A_3^T \\ C_2^T & -C_4 & -A_2^T & A_4^T \end{bmatrix} + \begin{bmatrix} A_1^\dagger & A_3^\dagger & C_1^\dagger & \bar{C}_2 \\ -A_2^\dagger & -A_4^\dagger & -C_2^\dagger & -C_4^\dagger \\ B_1^\dagger & \bar{B}_2 & -\bar{A}_1 & -\bar{A}_2 \\ -B_2^\dagger & -B_4^\dagger & A_3^\dagger & \bar{A}_4 \end{bmatrix} = 0.
\end{aligned}$$

So,  $A_1 = -A_1^\dagger, \bar{A}_1 = -A_1^T, A_3 = A_3^\dagger = \bar{A}_2^T, -A_4 = A_4^\dagger = \bar{A}_4, C_1 = -B_1^\dagger = -\bar{B}_1, C_2 = \bar{B}_2, C_4 = -B_4^\dagger = -\bar{B}_4$ , where  $A_1, A_4$  are skew Hermitian and  $B_1, B_4$  are symmetric.

Then  $sp(p, q)$  are complex valued matrices in the following block form:

$$sp(p, q) = \left\{ \left[ \begin{array}{cccc} A_1 & A_2 & B_1 & B_2 \\ \overline{A_2^T} & A_4 & B_2^T & B_4 \\ -\overline{B_1} & \overline{B_2} & \overline{A_1} & -\overline{A_2} \\ \overline{B_2^T} & -\overline{B_4} & -A_2^T & \overline{A_4} \end{array} \right] \left| \begin{array}{l} A_1, B_1 \text{ are } p \times p \text{ matrices} \\ A_1, A_4 \text{ are skew Hermitian matrices} \\ A_2, B_2 \text{ are } p \times q \text{ matrices} \\ B_1, B_4 \text{ are symmetric matrices} \end{array} \right. \right\}$$

There are  $p$  real/complex simple roots and  $l - p$  imaginary simple roots, where  $l$  is the number of simple roots. There will not be any Satake associates for  $sp(p, q)$ . So the Satake diagram will look like this:



**Example 4.3.2.** Let's look at  $sp(4, 2)$ .

A basis for the standard representation for  $sp(4, 2)$  are the following matrices:

$$e_1 = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad e_3 = \begin{bmatrix} 0 & I & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -I & 0 \\ 0 & 0 & 0 & -I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$e_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad e_5 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \quad e_6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

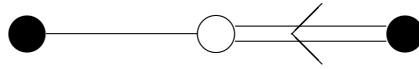
$$\begin{aligned}
e_7 &= \begin{bmatrix} 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -I & 0 & 0 \end{bmatrix} & e_8 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & -I & 0 \end{bmatrix} & e_9 &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
e_{10} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{11} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{12} &= \begin{bmatrix} 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
e_{13} &= \begin{bmatrix} 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{14} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{15} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
e_{16} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{17} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & -I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -I & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{18} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 & 0 & 0 \end{bmatrix} \\
e_{19} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -I \end{bmatrix} & e_{20} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} & e_{21} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 \end{bmatrix}.
\end{aligned}$$

The Cartan decomposition is  $sp(4, 2) = \mathfrak{t} \oplus \mathfrak{p}$ , where  $\mathfrak{t} = \langle e_1, e_2, e_3, e_4, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{19}, e_{20}, e_{21} \rangle$  and  $\mathfrak{p} = \langle e_5, e_6, e_7, e_8, e_{15}, e_{16}, e_{17}, e_{18} \rangle$ . We can see that  $\langle e_1 -$

$e_{19}, e_2 + e_4 - e_{19}, e_{15} - e_{18}$  is a maximally noncompact Cartan subalgebra with simple roots:

$$\alpha_1 = \begin{bmatrix} 0 \\ 2I \\ 0 \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} \sqrt{2} \\ -2I \\ -2I \end{bmatrix}, \quad \alpha_3 = \begin{bmatrix} 0 \\ 2I \\ 4I \end{bmatrix}.$$

The corresponding root spaces of the imaginary roots are :  $E_{\alpha_1} = \langle e_9 + Ie_{10} - e_{11} + Ie_{12} - e_{13} - Ie_{14} - 2e_{20} + 2Ie_{21} \rangle, E_{\alpha_3} = \langle e_9 + Ie_{10} - e_{11} - Ie_{12} + e_{13} + Ie_{14} \rangle$ . These are completely contained in the compact part  $\mathfrak{t}$  of the Cartan decomposition. Then because there are not any noncompact imaginary roots, this Cartan subalgebra is maximally noncompact. The first and third simple roots are imaginary and the second is complex. This means that we will have one black dot, then one white dot and then one black dot. There are not any Satake associates, so the Satake diagram will look like this:



**4.3.5**  $sp(p, p)$

The basis for  $sp(p, p)$  are complex matrices in the following block form.

$$sp(p, p) = \left\{ \left[ \begin{array}{cccc} A_1 & A_2 & B_1 & B_2 \\ \overline{A_2^T} & A_4 & B_2^T & B_4 \\ -\overline{B_1} & \overline{B_2} & \overline{A_1} & -\overline{A_2} \\ \overline{B_2^T} & -\overline{B_4} & -A_2^T & \overline{A_4} \end{array} \right] \middle| \begin{array}{l} A_1, B_1 \text{ are } p \times p \text{ matrices} \\ A_1, A_4 \text{ are skew Hermitian matrices} \\ A_2, B_2 \text{ are } p \times p \text{ matrices} \\ B_1, B_4 \text{ are symmetric matrices} \end{array} \right\}$$

The Lie algebra  $sp(p, p)$  has  $2p$  simple roots where  $p$  of them are real/complex, and  $p$  of them are imaginary. They alternate starting with an imaginary simple root. So the Satake diagram will look like this:



**Example 4.3.3.**  $sp(2, 2)$ :

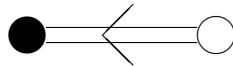
A basis for the standard representation for  $sp(2, 2)$  are the following matrices:

$$\begin{aligned}
e_1 &= \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_2 &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix} & e_3 &= \begin{bmatrix} 0 & I & 0 & 0 \\ -I & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & -I & 0 \end{bmatrix} \\
e_4 &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_5 &= \begin{bmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_6 &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \\
e_7 &= \begin{bmatrix} 0 & 0 & 0 & I \\ 0 & 0 & I & 0 \\ 0 & -I & 0 & 0 \\ -I & 0 & 0 & 0 \end{bmatrix} & e_8 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -I \end{bmatrix} & e_9 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \\
e_{10} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \end{bmatrix}.
\end{aligned}$$

The Cartan decomposition is  $sp(2, 2) = \mathfrak{t} \oplus \mathfrak{p}$ , where  $\mathfrak{t} = \langle e_1, e_4, e_5, e_8, e_9, e_{10} \rangle$  and  $\mathfrak{p} = \langle e_2, e_3, e_6, e_7 \rangle$ . We can see that  $\mathfrak{h} = \langle e_6, e_8 - e_1 \rangle$  is a Cartan subalgebra with simple roots:

$$\alpha_1 = \begin{bmatrix} 0 \\ 2I \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 2 \\ -2I \end{bmatrix}.$$

The corresponding root space of the imaginary root is:  $E_{\alpha_1} = \langle e_4 + Ie_5 - e_9 + Ie_{10} \rangle$ . This is completely contained in the compact part  $\mathfrak{t}$  of the Cartan decomposition. Because there are not any noncompact imaginary roots, our cartan subalgebra is maximally noncompact. The first simple root is imaginary and the second is complex. So the Satake diagram will look like this:





### 4.3.6 $sp(n)$

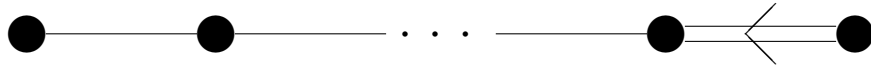
$sp(n)$  is a special case of  $sp(p, q)$  with  $q = 0$ . This is not to be confused with  $sp(n, \mathbb{R})$ . A basis for  $sp(n)$  are complex valued matrices that are of the form  $X = \begin{bmatrix} A & B \\ C & -A^T \end{bmatrix}$  where  $B, C$  are symmetric matrices and that satisfy the condition that  $X + X^\dagger = 0$ . Then,

$$\begin{aligned} \begin{bmatrix} A & B \\ C & -A^T \end{bmatrix} + \begin{bmatrix} A & B \\ C & -A^T \end{bmatrix}^\dagger &= \\ = \begin{bmatrix} A & B \\ C & -A^T \end{bmatrix} + \begin{bmatrix} A^\dagger & C^\dagger \\ B^\dagger & -\bar{A} \end{bmatrix} &= 0. \end{aligned}$$

This leads to  $A = -A^\dagger, C = -B^\dagger$ . Finally,

$$sp(n) = \left\{ \begin{bmatrix} A & B \\ -B^\dagger & \bar{A} \end{bmatrix} \mid \begin{array}{l} A \text{ is a } n/2 \times n/2 \text{ skew Hermitian matrix,} \\ B \text{ is a } n/2 \times n/2 \text{ symmetric matrix} \end{array} \right\}$$

The dimension of the Lie algebra  $sp(n)$  is  $\frac{n(n+1)}{2}$ , where  $n$  is the number of simple roots. All of the simple roots are imaginary. Then, there will not be any Satake associates for  $sp(n)$ . So the Satake diagram will look like this:



**Example 4.3.4.**  $sp(4)$ :

A basis for the standard representation for  $sp(4)$  are the following complex valued matrices:

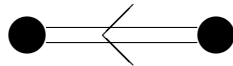
$$e_1 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad e_3 = \begin{bmatrix} 0 & I & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & -I \\ 0 & 0 & -I & 0 \end{bmatrix}$$

$$\begin{aligned}
e_4 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -I \end{bmatrix} & e_5 &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_6 &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \\
e_7 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} & e_8 &= \begin{bmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & e_9 &= \begin{bmatrix} 0 & 0 & 0 & I \\ 0 & 0 & I & 0 \\ 0 & I & 0 & 0 \\ I & 0 & 0 & 0 \end{bmatrix} \\
e_{10} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \end{bmatrix}.
\end{aligned}$$

The Cartan decomposition is  $sp(4) = \mathfrak{t} \oplus \mathfrak{p}$ , where  $\mathfrak{t} = \langle e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10} \rangle$  and  $\mathfrak{p} = \langle \rangle$ . We can see that  $\mathfrak{h} = \langle e_1, e_2 + e_4 \rangle$  is a Cartan subalgebra with simple roots:

$$\alpha_1 = \begin{bmatrix} 2I \\ -2I \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 0 \\ 2I \end{bmatrix}.$$

Both of these roots are imaginary. The corresponding root space of these simple roots are  $E_{\alpha_1} = \langle e_5 + Ie_6 - e_7 - Ie_8 + e_9 + Ie_{10} \rangle, E_{\alpha_2} = \langle e_2 - Ie_3 - e_4 \rangle$ . Both of these are completely contained in  $\mathfrak{t}$ , the compact part of the Cartan decomposition. Because there are not any noncompact imaginary roots, this Cartan subalgebra is maximally noncompact. This means that we will have two black dots that correspond to the two imaginary roots. So the Satake diagram will look like this:



#### 4.4 $D_n$ : Even Dimensional Special Orthogonal Lie Algebras

##### 4.4.1 $SO(p, q) : p + q$ is even: Even Dimensional Special Orthogonal Group

Let  $Q$  be the diagonal matrix  $Q = \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix}$ . Then the group  $SO(p, q)$  is the group of  $n \times n$  real valued matrices that satisfy  $M^T \cdot Q \cdot M = Q$  and  $\det(M) = 1$ , where  $n = p + q$  and  $t$  represents the transpose of the matrix. If  $M(t) \in SO(p, q)$  is a smooth

curve with  $M(0) = I$ , then for any non-singular matrix  $M$  we can take the derivative of the above relationship with respect to  $t$ . Then, evaluating for  $t = 0$  gives us the relationship for the corresponding Lie algebra.

$$M^T \cdot Q \cdot M = Q$$

$$M^T \cdot Q \cdot \dot{M} + \dot{M}^T \cdot Q \cdot M = 0$$

$$Q \cdot \dot{M} + \dot{M}^T \cdot Q = 0.$$

$$\text{Let } \dot{M} = X.$$

$$\text{Then, } Q \cdot X + X^T \cdot Q = 0.$$

Let  $X$  be the block matrix  $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ . Then,  $Q \cdot X + X^T \cdot Q = 0$ ,

$$\begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} \cdot \begin{bmatrix} A & B \\ C & D \end{bmatrix} + \begin{bmatrix} A^T & C^T \\ B^T & D^T \end{bmatrix} \cdot \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} = 0$$

$$\begin{bmatrix} A & B \\ -C & -D \end{bmatrix} + \begin{bmatrix} A^T & -C^T \\ B^T & -D^T \end{bmatrix} = 0.$$

This leads to  $A = -A^T, D = -D^T, C = B^T$ . Finally,

$$so(p, q) = \left\{ \begin{bmatrix} A & B \\ B^T & D \end{bmatrix} \mid \begin{array}{l} A \text{ is a } p \times p \text{ skew symmetric matrix,} \\ D \text{ is a } q \times q \text{ skew symmetric matrix and} \\ B \text{ is a } p \times q \text{ matrix} \end{array} \right\}$$

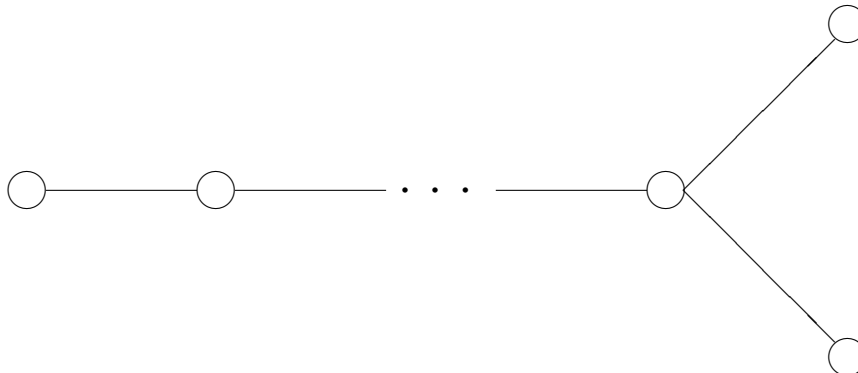
#### 4.4.2 $so(n, n)$

A basis for  $so(n, n)$  are trace-free matrices of the form:

$$so(n, n) = \left\{ \begin{bmatrix} A & B \\ B^T & D \end{bmatrix} \mid \begin{array}{l} A \text{ is a } n \times n \text{ skew symmetric matrix,} \\ B \text{ is a } n \times n \text{ matrix,} \\ D \text{ is a } n \times n \text{ skew symmetric matrix} \end{array} \right\}$$

The dimension of the Lie algebra  $so(n, n)$  is  $2n^2 - n$ , where  $l = n$  and  $l$  is the number of simple roots. All of the simple roots are real. Then the Satake diagram will look like

this:



**Example 4.4.1.** Let's look at  $so(3,3)$ .

A basis for the standard representation for  $so(3,3)$  are the following trace-free matrices with real entries:

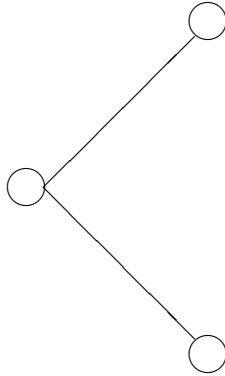
$$\begin{aligned}
 e_1 &= \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} &
 e_2 &= \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} &
 e_3 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 e_4 &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} &
 e_5 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} &
 e_6 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 e_7 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} &
 e_8 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} &
 e_9 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}
 \end{aligned}$$

$$\begin{aligned}
e_{10} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{11} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{12} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \\
e_{13} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{14} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} & e_{15} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} .
\end{aligned}$$

The Cartan decomposition is  $so(3,3) = \mathfrak{t} \oplus \mathfrak{p}$  where  $\mathfrak{t} = \langle e_1, e_2, e_3, e_{13}, e_{14}, e_{15} \rangle$  and  $\mathfrak{p} = \langle e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12} \rangle$ . We can see that  $\mathfrak{h} = \langle e_5, e_9, e_{10} \rangle$  is a Cartan subalgebra with simple roots:

$$\alpha_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, \quad \alpha_3 = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}.$$

Because all of these simple roots are real, there are not any noncompact imaginary roots and this Cartan subalgebra is maximally noncompact. We will have three white dots, so the Satake diagram will look like this:

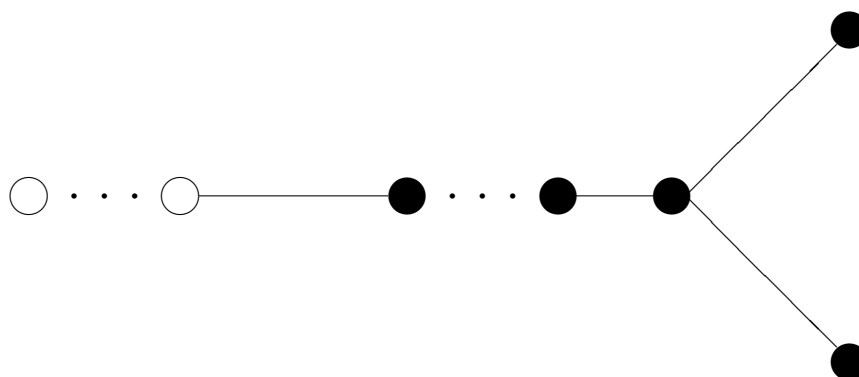


### 4.4.3 $so(p, q), p + q$ is even

A basis for  $so(p, q)$  are trace-free matrices of the form:

$$so(p, q) = \left\{ \left[ \begin{array}{cc} A & B \\ B^T & D \end{array} \right] \mid \begin{array}{l} A \text{ is a } p \times p \text{ skew symmetric matrix,} \\ D \text{ is a } q \times q \text{ skew symmetric matrix and} \\ B \text{ is a } p \times q \text{ matrix, and } \text{tr}(A) + \text{tr}(D) = 0 \end{array} \right\}$$

The dimension of the Lie algebra  $so(p, q)$  is  $\frac{p(p-1)}{2} + \frac{q(q-1)}{2} + pq$ , where  $l = \frac{p+q}{2}$  and  $l$  is the number of simple roots. There are  $p$  complex/real simple roots and  $l - p$  imaginary simple roots. Then the Satake diagram will look like this:



**Example 4.4.2.** Let's look at  $so(5, 1)$ .

A basis for the standard representation for  $so(5, 1)$  are the following trace-free matrices with real entries:

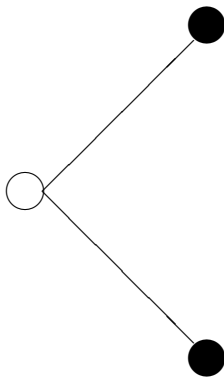
$$e_1 = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad e_3 = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned}
e_4 &= \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_5 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_6 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
e_7 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_8 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_9 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
e_{10} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{11} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 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 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{12} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \\
e_{13} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} & e_{14} &= \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 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} & e_{15} &= \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 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.
\end{aligned}$$

The Cartan decomposition is  $so(5, 1) = \mathfrak{t} \oplus \mathfrak{p}$  where  $\mathfrak{t} = \langle e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10} \rangle$  and  $\mathfrak{p} = \langle e_{11}, e_{12}, e_{13}, e_{14}, e_{15} \rangle$ . We can see that  $\langle e_1, e_8, e_{15} \rangle$  is a Cartan subalgebra with simple roots:

$$\alpha_1 = \begin{bmatrix} 1 \\ -I \\ 0 \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 0 \\ I \\ -I \end{bmatrix}, \quad \alpha_3 = \begin{bmatrix} 0 \\ I \\ I \end{bmatrix}.$$

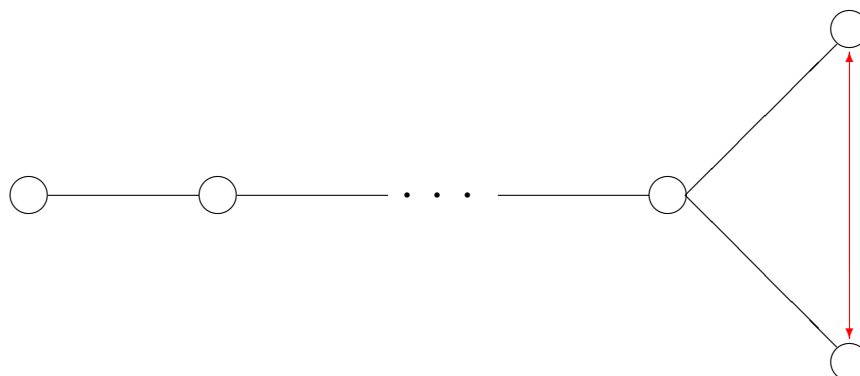
The corresponding root space of the imaginary roots is:  $E_{e_{\alpha_2}} = \langle e_2 - Ie_3 + Ie_5 + e_6 \rangle, E_{\alpha_3} = \langle e_2 - Ie_3 - Ie_5 - e_6 \rangle$ . These are completely contained in  $\mathfrak{t}$ , the compact part of the Cartan decomposition. Therefore, our Cartan subalgebra is maximally noncompact. The first simple root is complex and the second two are imaginary. Then we will have a white dot followed by two black dots. So the Satake diagram will look like this:



**4.4.4**  $so(p - 1, p + 1)$

$$so(p - 1, p + 1) = \left\{ \left[ \begin{array}{cc} A & B \\ B^T & D \end{array} \right] \middle| \begin{array}{l} A \text{ is a } p + 1 \times p + 1 \text{ skew symmetric matrix,} \\ B \text{ is a } p - 1 \times p + 1 \text{ matrix,} \\ D \text{ is a } p - 1 \times p - 1 \text{ skew symmetric matrix} \end{array} \right\}$$

The dimension of the Lie algebra  $so(p + 1, p - 1)$  is  $\frac{(p+1)(p)}{2} + \frac{(p-1)(p-2)}{2} + (p + 1)(p - 1)$ , where  $l = p$  and  $l$  is the number of simple roots. All of the simple roots are complex/real. So there will not be any compact simple roots. The last and second to last simple roots will be complex conjugates, and therefore Satake associates, because there aren't any compact roots. So the Satake diagram will look like this:



**Example 4.4.3.** Let's look at  $so(4, 2)$ .



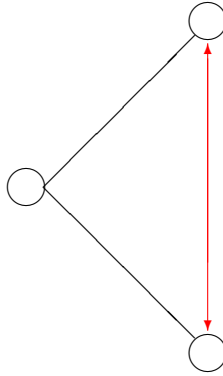


$$e_{13} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad e_{14} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad e_{15} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

The Cartan decomposition is  $so(4, 2) = \mathfrak{t} \oplus \mathfrak{p}$ , where  $\mathfrak{t} = \langle e_1, e_2, e_3, e_4, e_5, e_6, e_{15} \rangle$  and  $\mathfrak{p} = \langle e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14} \rangle$ . We can see that  $\mathfrak{h} = \langle e_1, e_{11}, e_{14} \rangle$  is a Cartan subalgebra with simple roots:

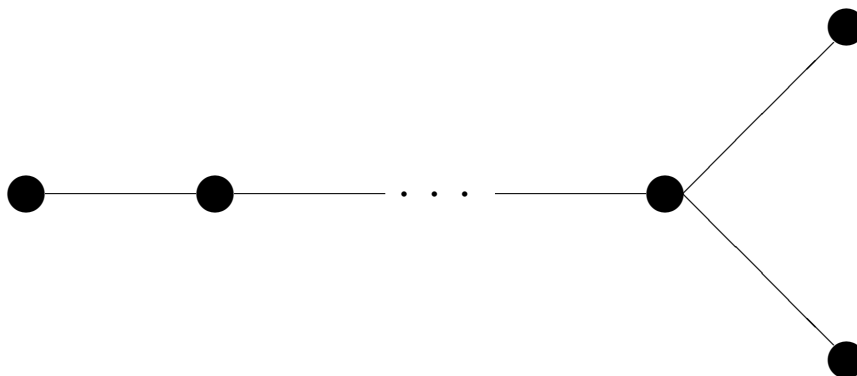
$$\alpha_1 = \begin{bmatrix} 1 \\ -I \\ 0 \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 0 \\ 1 \\ -I \end{bmatrix}, \quad \alpha_3 = \begin{bmatrix} 0 \\ 1 \\ -I \end{bmatrix}.$$

There are not any imaginary roots, so our Cartan subalgebra is maximally noncompact. All three simple roots are complex and  $\overline{\alpha_2} = \alpha_3$ . Then  $\alpha_2$  is the Satake associate of  $\alpha_3$ . So the Satake diagram will look like this:



#### 4.4.5 $so(2n)$

A basis for  $so(2n)$  is a list of linearly independent  $2n \times 2n$  skew symmetric trace-free matrices with real entries. The dimension of the Lie algebra  $so(2n)$  is  $n(2n - 1)$ , where  $n$  is the number of simple roots. All of the roots are imaginary. Since the Cartan decomposition for  $so(2n)$  will only have elements in  $\mathfrak{t}$ , the compact part, then all of the imaginary simple roots will be compact. So, there will not be any Satake associates. Then the Satake diagram will look like this:



**Example 4.4.4.** Let's look at  $so(6)$ .

A basis for the standard representation for  $so(6)$  are the following trace-free matrices with real entries:

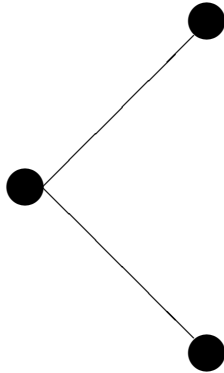
$$\begin{aligned}
 e_1 &= \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_2 &= \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_3 &= \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 e_4 &= \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_5 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_6 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 e_7 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_8 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_9 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}
 \end{aligned}$$

$$\begin{aligned}
e_{10} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{11} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{12} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \\
e_{13} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{14} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} & e_{15} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.
\end{aligned}$$

The Cartan decomposition is  $so(6) = \mathfrak{t} \oplus \mathfrak{p}$ , where  $\mathfrak{t} = \langle e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15} \rangle$  and  $\mathfrak{p} = \langle \rangle$ . We can see that  $\mathfrak{h} = \langle e_1, e_{10}, e_{15} \rangle$  is a Cartan subalgebra with simple roots:

$$\alpha_1 = \begin{bmatrix} I \\ -I \\ 0 \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 0 \\ I \\ -I \end{bmatrix}, \quad \alpha_3 = \begin{bmatrix} 0 \\ I \\ I \end{bmatrix}.$$

These imaginary roots have the root spaces:  $E_{\alpha_1} = e_{11} - Ie_{12} + Ie_{13} + e_{14}$ ,  $E_{\alpha_2} = e_2 - Ie_3 + Ie_6 + e_7$ ,  $E_{\alpha_3} = e_2 - Ie_3 - Ie_6 - e_7$ . These are all completely contained in the compact part  $\mathfrak{t}$  of the Cartan decomposition. Because we don't have any imaginary noncompact roots, our Cartan subalgebra is maximally noncompact. All three simple roots are complex imaginary roots and there are no Satake associates. So the Satake diagram will have three black dots and look like this:



**4.4.6**  $so^*(2n), n$  is even

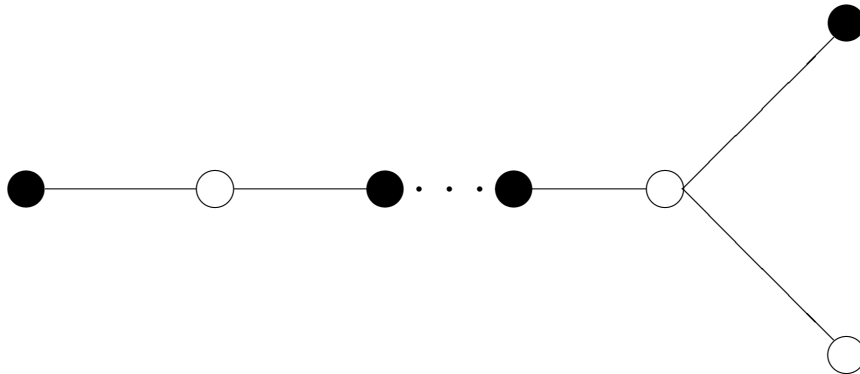
A basis for  $so^*(2n)$  are complex valued matrices that preserve two conditions:  $M + M^T = 0$  and  $QM + M^\dagger Q = 0$ . The first gives  $M = \begin{bmatrix} A & B \\ -B^T & D \end{bmatrix}$  where  $A$  and  $D$  are skew symmetric as shown previously. The second, with  $Q = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$  gives,

$$\begin{aligned} \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} \cdot \begin{bmatrix} A & B \\ B^T & D \end{bmatrix} + \begin{bmatrix} A & B \\ B^T & D \end{bmatrix}^\dagger \cdot \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} &= 0 \\ \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} \cdot \begin{bmatrix} A & B \\ B^T & D \end{bmatrix} + \begin{bmatrix} A^\dagger & \bar{B} \\ B^\dagger & D^\dagger \end{bmatrix} \cdot \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} &= 0 \\ \begin{bmatrix} B^T & D \\ -A & -B \end{bmatrix} + \begin{bmatrix} -\bar{B} & A^\dagger \\ -D^\dagger & B^\dagger \end{bmatrix} &= 0. \end{aligned}$$

This leads to  $B^T = \bar{B}, B = B^\dagger, D = -A^\dagger = \bar{A}$ . Then

$$so^*(2n) = \left\{ \begin{bmatrix} A & B \\ -\bar{B} & \bar{A} \end{bmatrix} \mid \begin{array}{l} A \text{ is a } n \times n \text{ skew symmetric complex matrix,} \\ B \text{ is a } n \times n \text{ Hermitian matrix} \end{array} \right\}$$

The dimension of the Lie algebra  $so^*(2n)$  is  $2n^2 - n$ , where  $n$  is the number of simple roots. The simple roots will be a mixture of imaginary and real/complex. There will not be any Satake associates for  $so^*(2n)$  where  $n$  is even. So the Satake diagram will look like this:



**Example 4.4.5.** Let's look at  $so^*(4)$ .

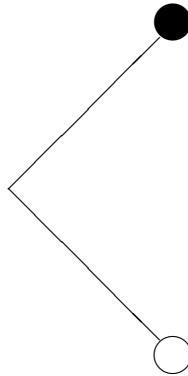
A basis for the standard representation for  $so^*(4)$  are the following complex valued matrices:

$$\begin{aligned}
 e_1 &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} & e_2 &= \begin{bmatrix} 0 & -I & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & -I & 0 \end{bmatrix} & e_3 &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 e_4 &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} & e_5 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} & e_6 &= \begin{bmatrix} 0 & 0 & 0 & -I \\ 0 & 0 & I & 0 \\ 0 & -I & 0 & 0 \\ I & 0 & 0 & 0 \end{bmatrix}.
 \end{aligned}$$

The Cartan decomposition is  $so^*(4) = \mathfrak{t} \oplus \mathfrak{p}$ , where  $\mathfrak{t} = \langle e_1, e_3, e_4, e_5 \rangle$  and  $\mathfrak{p} = \langle e_2, e_6 \rangle$ . We can see that  $\langle e_1, e_2 \rangle$  is a Cartan subalgebra with simple roots:

$$\alpha_1 = \begin{bmatrix} 0 \\ 2I \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}.$$

The imaginary root has the corresponding root space of  $E_{\alpha_1} = \langle e_3 - Ie_4 - e_5 \rangle$ . Because this is completely contained in the compact part  $\mathfrak{t}$  of the Cartan decomposition, our Cartan subalgebra is maximally noncompact. The first simple root is imaginary and the second is real, then we will have a black dot and a white dot. So the Satake diagram will look like this:

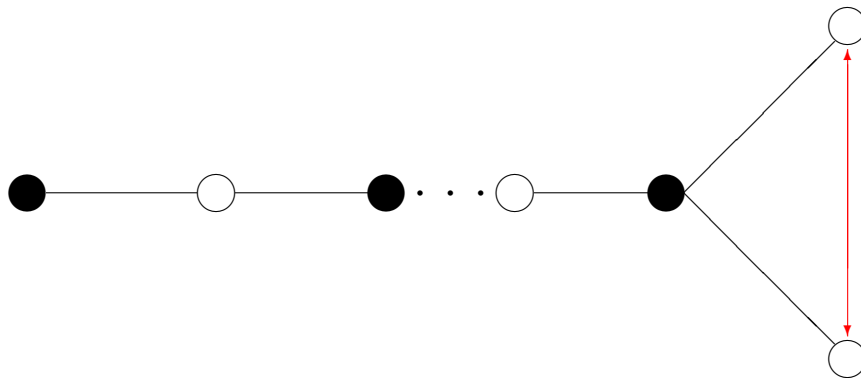


#### 4.4.7 $so^*(2n), n$ is odd

Remember that

$$so^*(2n) = \left\{ \left[ \begin{array}{cc} A & B \\ -\overline{B} & \overline{A} \end{array} \right] \mid \begin{array}{l} A \text{ is a } n \times n \text{ skew symmetric complex matrix,} \\ B \text{ is a } n \times n \text{ Hermitian matrix} \end{array} \right\}$$

The dimension of the Lie algebra  $so^*(2n)$  is  $2n^2 - n$ , where  $n$  is the number of simple roots. The simple roots will be a mixture of imaginary and real/complex. There will be one Satake associate pair for  $so^*(2n)$  where  $n$  is odd. So the Satake diagram will look like this:



**Example 4.4.6.** Let's look at  $so^*(6)$ .

A basis for the standard representation for  $so^*(6)$  are the following complex valued matrices:

$$e_1 = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad e_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

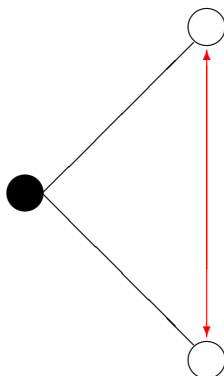
$$\begin{aligned}
e_4 &= \begin{bmatrix} 0 & -I & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & -I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_5 &= \begin{bmatrix} 0 & 0 & -I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \end{bmatrix} & e_6 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & -I & 0 \end{bmatrix} \\
e_7 &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_8 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_9 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
e_{10} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{11} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{12} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \\
e_{13} &= \begin{bmatrix} 0 & 0 & 0 & 0 & -I & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -I & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & e_{14} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -I \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & -I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 1 & 0 & 0 \end{bmatrix} & e_{15} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -I \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 \end{bmatrix}.
\end{aligned}$$

The Cartan decomposition is  $so^*(6) = \mathfrak{t} \oplus \mathfrak{p}$ , where  $\mathfrak{t} = \langle e_1, e_2, e_3, e_7, e_8, e_9, e_{10}, e_{11}, e_{12} \rangle$  and  $\mathfrak{p} = \langle e_4, e_5, e_6, e_{13}, e_{14}, e_{15} \rangle$ . We can see that  $\mathfrak{h} = \langle e_1, e_4, e_{12} \rangle$  is a Cartan subalgebra with simple roots:

$$\alpha_1 = \begin{bmatrix} 1 \\ I \\ -I \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 0 \\ 0 \\ 2I \end{bmatrix}, \quad \alpha_3 = \begin{bmatrix} 1 \\ -I \\ -I \end{bmatrix}.$$



The imaginary root has the root space:  $E_{\alpha_2} = e_7 - Ie_8 - e_{10}$ . Because this is completely contained in the compact part  $\mathfrak{t}$  of the Cartan decomposition, we don't have any noncompact imaginary roots and therefore our Cartan subalgebra is maximally noncompact. The first and third simple roots are complex and the second is imaginary. This means we will have a white dot, black dot and another white dot. Since  $\alpha_2$  is a compact root and  $\overline{\alpha_1} - \alpha_3 = [1, -I, I]^T - [1, -I, -I]^T = [0, 0, 2I]^T = \alpha_2$ . Then  $\alpha_3$  is the Satake associate of  $\alpha_1$ . So the Satake diagram will look like this:



This concludes the overview of the matrix representations of the four classical semi simple Lie algebra classes. The details of the exceptional cases  $E_6, E_7, E_8, F_4$ , and  $G_2$  will not be covered in this thesis.

## Chapter 5

### Cayley Transforms

The classification of real semi-simple Lie algebras has two steps. The first is to choose an appropriate Cartan subalgebra, namely one which is maximally noncompact. This is defined in chapter 3 (3.3.7). A Cayley transform is a conjugation of the Lie algebra mapping one  $\Theta$  stable Cartan subalgebra to another  $\Theta$  stable Cartan subalgebra. Through understanding the process of a Cayley transform, we can provide a sequence of steps by which a given Cartan subalgebra can be transformed into a maximally noncompact Cartan subalgebra. Given a maximally noncompact Cartan subalgebra, one finds the root space decomposition.

The second step is to identify the simple roots. From the simple roots one calculates the Cartan matrix. If necessary, one permutes the simple roots so that the resulting Cartan matrix is in standard form. This puts the simple roots into their proper order for the Satake diagram.

Then we determine which are pure imaginary and the simple roots which are Satake associates. From this information we can decorate the Dynkin diagram by coloring the dots and adding red arrows. We call this decorated Dynkin diagram a Satake diagram. The classification theorem for real simple Lie algebras states that there is a one to one correspondence between real simple Lie algebras and their Satake diagrams. In this chapter we will look at the details for this process of classification.

A Cayley transform is a conjugation of  $\mathfrak{g}_{\mathbb{C}}$  and will map one  $\Theta$  stable Cartan subalgebra of  $\mathfrak{g}$  to another  $\Theta$  stable Cartan subalgebra of  $\mathfrak{g}$ . There are two kinds of Cayley transforms. One generates a sequence of Cartan subalgebras resulting in a maximally compact Cartan subalgebra and the other generates a sequence of Cartan subalgebras resulting in a maximally noncompact Cartan subalgebra. I will focus on the second of these.

The basic steps for defining a Cayley transform are as follows[5]: First, find a noncompact imaginary root  $\beta$  (See the definition from chapter 3 3.3.6) from the positive roots of the given Lie algebra and Cartan subalgebra. Second, let  $E_\beta$  be a nonzero root vector. Because  $\beta$  is purely imaginary,  $\overline{E_\beta} \in \mathfrak{g}_{-\beta}$ . Because  $\beta$  is a noncompact root, we can say that  $E_\beta$  lies entirely in the positive part  $\mathfrak{p}$  of the Cartan decomposition. Therefore,  $\Theta(\overline{E_\beta}) = -E_\beta$ .

A Cartan involution has the property that  $B_\Theta(X, Y) = -B(X, \Theta Y)$  is positive definite, where  $B$  is the Killing form. Then, since  $\beta$  is noncompact, we have

$$0 < B_\Theta(E_\beta, E_\beta) = -B(E_\beta, \Theta \overline{E_\beta}) = B(E_\beta, \overline{E_\beta}).$$

Then we are allowed to normalize  $E_\beta$  to make  $B(E_\beta, \overline{E_\beta})$  any positive constant. Next, we find a unique  $H_\beta$  by scaling to satisfy the following bracket relationships:

$$[H_\beta, E_\beta] = 2E_\beta, \quad [H_\beta, \overline{E_\beta}] = -2\overline{E_\beta}, \quad [E_\beta, \overline{E_\beta}] = H_\beta. \quad (5.1)$$

Note: the vectors  $E_\beta, \overline{E_\beta}, H_\beta$  define the three dimensional Lie algebra  $sl(2)$ .

The elements  $E_\beta + \overline{E_\beta}, i(E_\beta - \overline{E_\beta}) \in \mathfrak{g}$  because they are fixed by complex conjugation.

The Cayley transform for the noncompact imaginary root  $\beta$  is the automorphism:

$$c_\beta = \exp(\text{ad}(\frac{\pi}{4}(\overline{E_\beta} - E_\beta))) \quad (5.2)$$

where  $\exp$  is the exponential of the adjoint matrix. Our new Cartan subalgebra  $\tilde{\mathfrak{h}}$  is:

$$\tilde{\mathfrak{h}} = \ker(\beta|_{\mathfrak{h}}) \oplus \mathbb{R}(E_\beta + \overline{E_\beta}). \quad (5.3)$$

Explicitly, let the Cartan subalgebra  $\mathfrak{h} = \langle H_1, H_2, \dots, H_r \rangle$ . The corresponding simple roots are  $\Delta_s = \langle \alpha_1, \alpha_2, \dots, \alpha_r \rangle$  and the noncompact imaginary root is  $\beta = \alpha_m =$

$[p_1, p_2, \dots, p_r]^T$ . After the Cayley transform, the new Cartan subalgebra is

$$\tilde{\mathfrak{h}} = \{a^i H_i, E_\beta + \overline{E_\beta} \mid a^i p_i = 0\},$$

where  $E_\beta + \overline{E_\beta} \in \mathfrak{p}$  and  $a^i H_i \mid a^i p_i = 0$  has one less element in  $\mathfrak{t}$  than  $\mathfrak{h}$ . Note:  $E_\beta$  is not uniquely determined and these formulas depend on the choice of  $\beta$ .

The new Cartan subalgebra  $\tilde{\mathfrak{h}}$  has the following properties:

1. The Cartan subalgebra  $\tilde{\mathfrak{h}}$  is a real subalgebra of  $\mathfrak{g}$ .
2. The intersection  $\tilde{\mathfrak{h}} \cap \mathfrak{p}$  is one dimension greater compared to  $\mathfrak{h} \cap \mathfrak{p}$ .

To see that 5.3 is valid, we can use infinite series to calculate that[5]:

$$c_\beta(H_\beta) = E_\beta + \overline{E_\beta} \tag{5.4}$$

$$c_\beta(E_\beta - \overline{E_\beta}) = E_\beta - \overline{E_\beta} \tag{5.5}$$

$$c_\beta(E_\beta + \overline{E_\beta}) = -H_\beta. \tag{5.6}$$

Let's check the first of these. Note that:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots, \tag{5.7}$$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots, \tag{5.8}$$

$$\begin{aligned} \text{Ad}(\exp(x))(y) &= \exp(\text{ad}_x(y)) \\ &= y + [x, y] + [x, [x, y]] + \dots \\ &= \frac{y}{0!} + \frac{\text{ad}_x(y)}{1!} + \frac{\text{ad}_x^2(y)}{2!} + \frac{\text{ad}_x^3(y)}{3!} + \dots \end{aligned} \tag{5.9}$$

Then,

$$\begin{aligned} c_\beta(H_\beta) &= \text{Ad}(\exp \frac{\pi}{4}(\overline{E_\beta} - E_\beta))(H_\beta) \\ &= \frac{H_\beta}{0!} + \frac{[\pi/4(\overline{E_\beta} - E_\beta), H_\beta]}{1!} + \frac{[\pi/4(\overline{E_\beta} - E_\beta), [\pi/4(\overline{E_\beta} - E_\beta), H_\beta]]}{2!} + \\ &+ \frac{[\pi/4(\overline{E_\beta} - E_\beta), [\pi/4(\overline{E_\beta} - E_\beta), [\pi/4(\overline{E_\beta} - E_\beta), H_\beta]]]}{3!} + \dots \end{aligned}$$

$$= a_0 + a_1 + a_2 + a_3 + \cdots.$$

Let's calculate  $a_0 + a_1 + a_2 + a_3 + \cdots$  separately using the bracket relations from. We calculate that 5.1

$$\begin{aligned} a_0 &= \frac{H_\beta}{0!}, \\ a_1 &= \frac{[\pi/4(\overline{E}_\beta - E_\beta), H_\beta]}{1!} = \frac{\pi/4([\overline{E}_\beta, H_\beta] - [E_\beta, H_\beta])}{1!} = \frac{\pi(\overline{E}_\beta + E_\beta)}{2 \cdot 1!}, \\ a_2 &= \frac{[\pi/4(\overline{E}_\beta - E_\beta), [\pi/4(\overline{E}_\beta - E_\beta), H_\beta]]}{2!} = \frac{[\pi/4(\overline{E}_\beta - E_\beta), \pi/2(\overline{E}_\beta + E_\beta)]}{2!} \\ &= \left(\frac{\pi}{2}\right)^2 \frac{[\overline{E}_\beta, E_\beta] - [E_\beta, \overline{E}_\beta]}{2 \cdot 2!} = -\left(\frac{\pi}{2}\right)^2 H_\beta, \\ a_3 &= \frac{[\pi/4(\overline{E}_\beta - E_\beta), -(\pi/2)^2 H_\beta]}{3!} = -\left(\frac{\pi}{2}\right)^3 \frac{\overline{E}_\beta + E_\beta}{3!}, \\ a_4 &= \left(\frac{\pi}{2}\right)^4 \frac{H_\beta}{4!}, \\ a_5 &= \left(\frac{\pi}{2}\right)^5 \frac{\overline{E}_\beta + E_\beta}{5!}. \end{aligned}$$

Then we can see that

$$\begin{aligned} c_\beta(H_\beta) &= (\overline{E}_\beta + E_\beta) \left( \frac{\pi}{2} - \left(\frac{\pi}{2}\right)^3 \frac{1}{3!} + \left(\frac{\pi}{2}\right)^5 \frac{1}{5!} - \left(\frac{\pi}{2}\right)^7 \frac{1}{7!} + \cdots \right) + \\ &\quad + (H_\beta) \left( 1 - \left(\frac{\pi}{2}\right)^2 \frac{1}{2!} + \left(\frac{\pi}{2}\right)^4 \frac{1}{4!} - \left(\frac{\pi}{2}\right)^6 \frac{1}{6!} + \cdots \right) \\ &= (\overline{E}_\beta + E_\beta) \sin(\pi/2) + H_\beta \cos(\pi/2) \\ &= \overline{E}_\beta + E_\beta. \end{aligned}$$

**Example 5.0.1.**  $sl(2)$

Let's follow this process through with a few examples. We will start with the real simple Lie algebra  $sl(2)$ .

A standard representation for  $sl(2, R)$  are the following trace free matrices

$$e_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} e_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} e_3 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

Given the Cartan subalgebra  $\mathfrak{h} = \langle e_2 - e_3 \rangle$ , the Cartan decomposition,

$$\mathfrak{t} = \langle e_2 - e_3 \rangle, \mathfrak{p} = \langle e_1, e_2 + e_3 \rangle, \quad (5.10)$$

and the Cartan involution,  $\Theta = e_1 \rightarrow -e_1, e_2 \rightarrow -e_3, e_3 \rightarrow -e_2$ , we can see that the given Cartan subalgebra is entirely contained in the compact part  $\mathfrak{t}$  of the Cartan decomposition.

The root space decomposition with respect to this Cartan subalgebra is  $\mathfrak{g} = \mathfrak{h} \oplus \mathfrak{g}_{2I} \oplus \mathfrak{g}_{-2I}$  where

$$\begin{aligned} \mathfrak{g}_{[2I]} &= \langle e_1 + Ie_2 + Ie_3 \rangle, \\ \mathfrak{g}_{[-2I]} &= \langle e_1 - Ie_2 - Ie_3 \rangle. \end{aligned}$$

There is one positive root and therefore one simple root,  $\beta = [2I]$ . This root is imaginary, so we need to check to see if it is noncompact. Since  $\mathfrak{g}_\beta = \langle e_1 + Ie_2 + Ie_3 \rangle$  which is entirely contained in  $\mathfrak{p}$ , the positive noncompact part of the Cartan decomposition,  $\beta$  is a noncompact imaginary root and we can use it to construct the Cayley transform,  $c_\beta$  of the Cartan subalgebra  $\mathfrak{h} = \langle e_2 - e_3 \rangle$ .

After scaling  $\mathfrak{g}_\beta$  and finding  $H_\beta$  to satisfy the bracket relations 5.1 :

$$\begin{aligned} E_\beta &= \frac{I}{2}e_1 - \frac{1}{2}e_2 - \frac{1}{2}e_3, \\ \overline{E}_\beta &= -\frac{I}{2}e_1 - \frac{1}{2}e_2 - \frac{1}{2}e_3, \\ H_\beta &= -Ie_2 + Ie_3. \end{aligned}$$

Then, the new element of the Cartan subalgebra is

$$E_\beta + \overline{E}_\beta = \frac{I}{2}e_1 - \frac{1}{2}e_2 - \frac{1}{2}e_3 - \frac{I}{2}e_1 - \frac{1}{2}e_2 - \frac{1}{2}e_3 = -e_2 - e_3.$$

Since there is only one element to our Cartan subalgebra, this means that the new Cartan subalgebra is  $\tilde{\mathfrak{h}} = \langle e_2 + e_3 \rangle$ . Looking back at the Cartan decomposition 5.10, we

can see that the new Cartan subalgebra has “moved” over to the positive part  $\mathfrak{p}$  of the Cartan decomposition.

Following the formula 5.2 for this example we find that

$$X = \overline{E_\beta} - E_\beta = -\frac{I}{2}e_1 - \frac{1}{2}e_2 - \frac{1}{2}e_3 - \frac{I}{2}e_1 + \frac{1}{2}e_2 + \frac{1}{2}e_3 = -Ie_1.$$

Then we find the adjoint matrix of  $\exp X \frac{\pi}{4}$  which is

$$c_\beta = \exp(\text{ad}(\frac{\pi}{4}(\overline{E_\beta} - E_\beta))) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -I & 0 \\ 0 & 0 & I \end{bmatrix}.$$

Using this matrix to form a Lie algebra homomorphism  $e_i \rightarrow c_\beta(e_i)$  gives the following:

$$e_1 \rightarrow e_1, \quad e_2 \rightarrow -Ie_2, \quad e_3 \rightarrow Ie_3.$$

Let's use this map to verify the equations from above 5.4 to be

$$c_\beta(-Ie_2 + Ie_3) = -e_2 - e_3, \tag{5.11}$$

$$c_\beta(Ie_1) = Ie_1, \tag{5.12}$$

$$c_\beta(e_2 - e_3) = -Ie_2 - Ie_3. \tag{5.13}$$

Then we can also see that  $c_\beta$  sends the original Cartan subalgebra  $\mathfrak{h} = \langle e_2 - e_3 \rangle$  to our new maximally noncompact Cartan subalgebra  $\tilde{\mathfrak{h}} = \langle I(-e_2 - e_3) \rangle$ . Note: we can simplify this to be  $\tilde{\mathfrak{h}} = \langle e_2 + e_3 \rangle$ . The corresponding root space decomposition using this new Cartan subalgebra is  $\mathfrak{g} = \mathfrak{h} \oplus \mathfrak{g}_{[-2]} \oplus \mathfrak{g}_{[2]}$ , where

$$\mathfrak{g}_{[-2]} = \langle e_1 + e_2 - e_3 \rangle,$$

$$\mathfrak{g}_{[2]} = \langle e_1 - e_2 + e_3 \rangle.$$

The real root [2] is the positive simple root. There aren't any imaginary noncompact

roots, so this Cartan subalgebra is maximally noncompact and can be used to draw the Satake diagram and classify the Lie algebra. Since we have just one simple root, and that root is real, we simply have one white dot for the Satake diagram. It will look like this:



**Example 5.0.2.**  $so(4, 2)$

Let's look at this process with another example. Consider the Lie algebra  $so(4, 2)$ .

It is 15 dimensional with the following multiplication table:

$so(4, 2)$	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$e_7$	$e_8$	$e_9$	$e_{10}$	$e_{11}$	$e_{12}$	$e_{13}$	$e_{14}$	$e_{15}$
$e_1$	0	$e_2$	$-e_3$	0	$e_5$	$-e_6$	$e_7$	$e_8$	0	0	$-e_{11}$	$-e_{12}$	0	0	0
$e_2$	$-e_2$	0	$e_1 - e_4$	$e_2$	0	0	0	0	$e_7$	$e_8$	$-e_{13}$	$-e_{14}$	0	0	0
$e_3$	$e_3$	$-e_1 + e_4$	0	$-e_3$	0	0	$e_9$	$e_{10}$	0	0	0	0	$-e_{11}$	$-e_{12}$	0
$e_4$	0	$-e_2$	$e_3$	0	$e_5$	$-e_6$	0	0	$e_9$	$e_{10}$	0	0	$-e_{13}$	$-e_{14}$	0
$e_5$	$-e_5$	0	0	$-e_5$	0	$-e_1 - e_4$	0	0	0	0	$e_9$	$e_{10}$	$-e_7$	$-e_8$	0
$e_6$	$e_6$	0	0	$e_6$	$e_1 + e_4$	0	$e_{13}$	$e_{14}$	$-e_{11}$	$-e_{12}$	0	0	0	0	0
$e_7$	$-e_7$	0	$-e_9$	0	0	$-e_{13}$	0	0	$e_5$	0	$-e_1$	$e_{15}$	$-e_2$	0	$-e_8$
$e_8$	$-e_8$	0	$-e_{10}$	0	0	$-e_{14}$	0	0	0	$e_5$	$-e_{15}$	$-e_1$	0	$-e_2$	$e_7$
$e_9$	0	$-e_7$	0	$-e_9$	0	$e_{11}$	$-e_5$	0	0	0	$-e_3$	0	$-e_4$	$e_{15}$	$-e_{10}$
$e_{10}$	0	$-e_8$	0	$-e_{10}$	0	$e_{12}$	0	$-e_5$	0	0	0	$-e_3$	$-e_{15}$	$-e_4$	$e_9$
$e_{11}$	$e_{11}$	$e_{13}$	0	0	$-e_9$	0	$e_1$	$e_{15}$	$e_3$	0	0	0	$e_6$	0	$-e_{12}$
$e_{12}$	$e_{12}$	$e_{14}$	0	0	$-e_{10}$	0	$-e_{15}$	$e_1$	0	$e_3$	0	0	0	$e_6$	$e_{11}$
$e_{13}$	0	0	$e_{11}$	$e_{13}$	$e_7$	0	$e_2$	0	$e_4$	$e_{15}$	$-e_6$	0	0	0	$-e_{14}$
$e_{14}$	0	0	$e_{12}$	$e_{14}$	$e_8$	0	0	$e_2$	$-e_{15}$	$e_4$	0	$-e_6$	0	0	$e_{13}$
$e_{15}$	0	0	0	0	0	0	$e_8$	$-e_7$	$e_{10}$	$-e_9$	$e_{12}$	$-e_{11}$	$e_{14}$	$-e_{13}$	0

A Cartan subalgebra of  $so(4, 2)$  is  $\mathfrak{h} = [e_{15}, e_1 + e_4, e_2 - e_3]$ .

The root space decomposition for this Cartan subalgebra is  $\mathfrak{g} = \mathfrak{h} \oplus_{i=1}^{12} \mathfrak{g}_\beta$  where

$$\begin{aligned}
\mathfrak{g}_{[-I, -1, -I]^T} &= \langle e_{11} + Ie_{12} - Ie_{13} + e_{14} \rangle, & \mathfrak{g}_{[I, -1, -I]^T} &= \langle e_{11} - Ie_{12} - Ie_{13} - e_{14} \rangle, \\
\mathfrak{g}_{[0, 0, -2I]^T} &= \langle e_1 - Ie_2 - Ie_3 - e_4 \rangle, & \mathfrak{g}_{[0, -2, 0]^T} &= \langle e_6 \rangle, \\
\mathfrak{g}_{[I, -1, I]^T} &= \langle e_{11} - Ie_{12} + Ie_{13} + e_{14} \rangle, & \mathfrak{g}_{[-I, 1, I]^T} &= \langle e_7 + Ie_8 + Ie_9 - e_{10} \rangle, \\
\mathfrak{g}_{[I, 1, -I]^T} &= \langle e_7 - Ie_8 - Ie_9 - e_{10} \rangle, & \mathfrak{g}_{[-I, -1, I]^T} &= \langle e_{11} + Ie_{12} + Ie_{13} - e_{14} \rangle, \\
\mathfrak{g}_{[0, 2, 0]^T} &= \langle e_5 \rangle, & \mathfrak{g}_{[-I, 1, -I]^T} &= \langle e_7 + Ie_8 - Ie_9 + e_{10} \rangle, \\
\mathfrak{g}_{[I, 1, I]^T} &= \langle e_7 - Ie_8 + Ie_9 + e_{10} \rangle, & \mathfrak{g}_{[0, 0, 2I]^T} &= \langle e_1 + Ie_2 + Ie_3 - e_4 \rangle.
\end{aligned}$$



Then from this root space decomposition we choose the positive roots to be

$$\alpha_+ : \begin{bmatrix} I \\ -1 \\ I \end{bmatrix}, \begin{bmatrix} I \\ 1 \\ -I \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}, \begin{bmatrix} I \\ 1 \\ I \end{bmatrix}, \begin{bmatrix} I \\ -1 \\ -I \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 2I \end{bmatrix}.$$

From the positive roots the simple roots are determined to be

$$\alpha_s : \begin{bmatrix} I \\ -1 \\ -I \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 2I \end{bmatrix}.$$

The root space decomposition and therefore the positive and simple roots are determined based on the choice of Cartan subalgebra. So as the Cartan subalgebra changes, these will also change. However, the Cartan decomposition and involution are not dependent on the choice of Cartan subalgebra.

The Cartan Decomposition of  $so(4, 2)$  is determined to be

$$\mathfrak{t} = \langle e_2 - e_3, e_5 + 4e_6, e_7 + 2e_{11}, e_8 + 2e_{12}, e_9 + 2e_{13}, e_{10} + 2e_{14}, e_{15} \rangle,$$

which is the compact part, and

$$\mathfrak{p} = \langle e_1, e_2 + e_3, e_4, e_5 - 4e_6, e_7 - 2e_{11}, e_8 - 2e_{12}, e_9 - 2e_{13}, e_{10} - 2e_{14} \rangle,$$

which is the positive part.

Visually inspecting the Cartan subalgebra tells us that the intersection of  $\mathfrak{h}$  and  $\mathfrak{t}$  is two dimensional,  $\mathfrak{h} \cap \mathfrak{t} = \langle e_2 - e_3, e_{15} \rangle$ . Also, we can see that the intersection of  $\mathfrak{h}$  and  $\mathfrak{p}$  is one dimension,  $\mathfrak{h} \cap \mathfrak{p} = \langle e_1 + e_4 \rangle$ .

There are three simple roots. For this Cartan subalgebra, one is real, one is complex, and one is imaginary. If the imaginary root is noncompact, we can use it to construct a new Cartan subalgebra whose intersection with  $\mathfrak{p}$  goes up by one dimension.

We call the imaginary root  $\beta = [0, 0, 2I]^T$ . To make sure that  $\beta$  is noncompact, we use the root space decomposition  $\mathfrak{g} = \mathfrak{h} \oplus_{i=1}^{12} \mathfrak{g}_\beta$  to check that the root space  $\mathfrak{g}_\beta = \langle e_1 + Ie_2 + Ie_3 - e_4 \rangle$  is completely contained in  $\mathfrak{p}$ . By inspection, we can see that  $R_\beta$  lies entirely in  $\mathfrak{p}$ . Then  $\overline{\mathfrak{g}}_\beta = \langle e_1 - Ie_2 - Ie_3 - e_4 \rangle$ . We use the structure equations

$$[H_\beta, E_\beta] = 2E_\beta, [H_\beta, \overline{E}_\beta] = -2\overline{E}_\beta, [E_\beta, \overline{E}_\beta] = H_\beta$$

to scale the given basis for  $E_\beta$  and  $\overline{E}_\beta$  to get  $\mathfrak{g}_\beta$  to  $E_\beta$  and  $\overline{R}_\beta$  to  $\overline{E}_\beta$  as well as uniquely determine  $H_\beta$ . Then after scaling,

$$E_\beta = 1/2e_1 + I/2e_2 + I/2e_3 - 1/2e_4$$

$$\overline{E}_\beta = 1/2e_1 - I/2e_2 - I/2e_3 - 1/2e_4.$$

We find the vector  $H_\beta \in \mathfrak{h}$ , where  $\mathfrak{h}$  is the Cartan subalgebra, such that, for  $E_\beta, \overline{E}_\beta \in R_\beta$  and  $H_\beta \in \mathfrak{h}$  the structure equations above are satisfied. Thus, in this example our  $H_\beta = -Ie_2 + Ie_3$ .

Then the new Cartan subalgebra element is  $E_\beta + \overline{E}_\beta = 1/2e_1 + I/2e_2 + I/2e_3 - 1/2e_4 + 1/2e_1 - I/2e_2 - I/2e_3 - 1/2e_4 = e_1 - e_4$ . The other elements of the new

Cartan subalgebra are found using the nullspace of  $\beta$ . In this case, because  $\beta = \begin{bmatrix} 0 \\ 0 \\ 2I \end{bmatrix}$ ,

then the null space  $NS = \left\{ \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right\}$ . The original Cartan subalgebra was  $\mathfrak{h} = \langle e_{15}, e_1 + e_4, e_2 - e_3 \rangle$ . Therefore, the other elements of the new Cartan subalgebra are,  $e_1 + e_4$  and  $e_{15}$ . Thus, the new Cartan subalgebra is  $\tilde{\mathfrak{h}} = \langle e_1 + e_4, e_{15}, e_1 - e_4 \rangle$ . The final result is  $\tilde{\mathfrak{h}} = \langle e_1, e_4, e_{15} \rangle$ .

Following the formula in 5.2 we find the homomorphism  $e_i \rightarrow c_\beta(e_i)$  is given by

$$e_1 \rightarrow \frac{1}{2}e_1 + \frac{I}{2}e_2 - \frac{I}{2}e_3 + \frac{1}{2}e_4, \quad e_2 \rightarrow \frac{I}{2}e_1 + \frac{1}{2}e_2 + \frac{1}{2}e_3 - \frac{I}{2}e_4,$$

$$e_3 \rightarrow -\frac{I}{2}e_1 + \frac{1}{2}e_2 + \frac{1}{2}e_3 + \frac{I}{2}e_4, \quad e_4 \rightarrow \frac{1}{2}e_1 - \frac{I}{2}e_2 + \frac{I}{2}e_3 + \frac{1}{2}e_4, \quad e_5 \rightarrow e_5, \quad e_6 \rightarrow e_6,$$

$$\begin{aligned}
e_7 &\rightarrow \frac{\sqrt{2}}{2}e_7 - \frac{I\sqrt{2}}{2}e_9, & e_8 &\rightarrow \frac{\sqrt{2}}{2}e_8 - \frac{I\sqrt{2}}{2}e_{10}, & e_9 &\rightarrow -\frac{I\sqrt{2}}{2}e_7 + \frac{\sqrt{2}}{2}e_9, \\
e_{10} &\rightarrow -\frac{I\sqrt{2}}{2}e_8 + \frac{\sqrt{2}}{2}e_{10}, & e_{11} &\rightarrow \frac{\sqrt{2}}{2}e_{11} + \frac{I\sqrt{2}}{2}e_{13}, & e_{12} &\rightarrow \frac{\sqrt{2}}{2}e_{12} + \frac{I\sqrt{2}}{2}e_{14}, \\
e_{13} &\rightarrow \frac{I\sqrt{2}}{2}e_{11} + \frac{\sqrt{2}}{2}e_{13}, & e_{14} &\rightarrow \frac{I\sqrt{2}}{2}e_{12} + \frac{\sqrt{2}}{2}e_{14}, & e_{15} &\rightarrow e_{15}.
\end{aligned}$$

Let's use this map to verify the equations from above 5.4 as

$$c_\beta(-Ie_2 + Ie_3) = e_1 - e_4, \quad (5.14)$$

$$c_\beta(Ie_2 + Ie_3) = Ie_2 + Ie_3, \quad (5.15)$$

$$c_\beta(e_1 - e_4) = Ie_2 - Ie_3. \quad (5.16)$$

Then we can also see that  $c_\beta$  sends the original Cartan subalgebra  $\mathfrak{h} = \langle e_{15}, e_1 + e_4, e_2 - e_3 \rangle$  to  $\tilde{\mathfrak{h}} = \langle e_{15}, e_1 + e_4, Ie_1 - Ie_4 \rangle$ .

Visually inspecting the new Cartan subalgebra tells us that the intersection of  $\tilde{\mathfrak{h}}$  and  $\mathfrak{t}$  is one dimension,  $\tilde{\mathfrak{h}} \cap \mathfrak{t} = \langle e_{15} \rangle$ . Also, we can see that the intersection of  $\tilde{\mathfrak{h}}$  and  $\mathfrak{p}$  is two dimensions,  $\tilde{\mathfrak{h}} \cap \mathfrak{p} = \langle e_1, e_4 \rangle$ .

The root space decomposition for the new Cartan subalgebra is  $\mathfrak{g} = \tilde{\mathfrak{h}} \oplus_{i=1}^{12} R_\alpha$ , where

$$\begin{aligned}
\mathfrak{g}_{[1,0,-I]^T} &= \langle e_7 + Ie_8 \rangle, & \mathfrak{g}_{[-1,-1,0]^T} &= \langle e_6 \rangle, \\
\mathfrak{g}_{[0,1,I]^T} &= \langle e_9 - Ie_{10} \rangle, & \mathfrak{g}_{[-1,1,0]^T} &= \langle e_3 \rangle, \\
\mathfrak{g}_{[-1,0,-I]^T} &= \langle e_{11} - Ie_{12} \rangle, & \mathfrak{g}_{[0,-1,I]^T} &= \langle e_{13} + Ie_{14} \rangle, \\
\mathfrak{g}_{[1,1,0]^T} &= \langle e_5 \rangle, & \mathfrak{g}_{[0,1,-I]^T} &= \langle e_9 + Ie_{10} \rangle, \\
\mathfrak{g}_{[-1,0,I]^T} &= \langle e_{11} - Ie_{12} \rangle, & \mathfrak{g}_{[1,-1,0]^T} &= \langle e_2 \rangle, \\
\mathfrak{g}_{[1,0,I]^T} &= \langle e_7 - Ie_8 \rangle, & \mathfrak{g}_{[0,-1,-I]^T} &= \langle e_{13} + Ie_{14} \rangle.
\end{aligned}$$

Then from this root space decomposition we choose the positive roots to be

$$\alpha_+ : \begin{bmatrix} 1 \\ 0 \\ -I \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ I \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ I \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -I \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}.$$

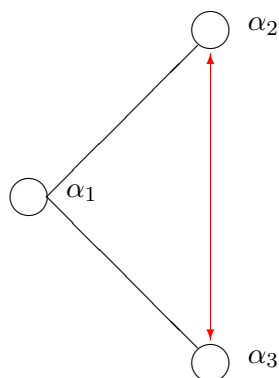
From the positive roots the simple roots are determined to be

$$\alpha_s : \alpha_1 = \begin{bmatrix} I \\ -1 \\ 0 \end{bmatrix}, \alpha_2 = \begin{bmatrix} 0 \\ 1 \\ I \end{bmatrix}, \alpha_3 = \begin{bmatrix} 0 \\ 1 \\ -I \end{bmatrix}.$$

These simple roots are in the correct order. If there are not any imaginary noncompact simple roots, then the Cartan subalgebra is in its maximally noncompact form. Because none of these roots are imaginary, then there are not any noncompact imaginary roots, and we can say that our Cartan subalgebra is maximally noncompact.

The next step is to identify the simple roots which are imaginary and the roots which are Satake associates. In this case the second and third simple roots are complex conjugates, and because there aren't any compact roots, are therefore Satake associates. Any pure imaginary roots are colored black. A red arrow is added connecting the Satake associates. We can decorate the Dynkin diagram by coloring the dots and adding red arrows. We call this decorated Dynkin diagram a Satake diagram. The classification theorem for real simple Lie algebras states that there is a one to one correspondence between a real simple Lie algebra and its Satake diagram.

Because this Cartan subalgebra does not have any noncompact imaginary roots associated with it, this Cartan subalgebra is maximally noncompact, and can be used to classify the algebra as described above. Since all of the simple roots are complex and none are purely imaginary, all of our dots will remain white.  $\overline{\alpha_2} = \alpha_3$ , so a red arrow will connect the two. So the Satake diagram for  $so(4, 2)$  will look like this:



In this chapter, we have detailed the theory behind the Cayley transform. We have

shown how this process allows us to classify real simple Lie algebras. Then, we showed two examples giving the details of the classification using the Cayley transform. This completes the discussion of the theory for this thesis.

## Chapter 6

### Low Dimensional Isomorphisms

In Chapter 10 of Helgason [4] the following low dimensional isomorphisms of real simple Lie algebras are stated as:

$$sl(4) \sim so(3, 3)$$

$$sl(2) \sim so(2, 1) \sim sp(2, \mathbb{R})$$

$$su^*(4) \sim so(5, 1)$$

$$su(2) \sim so(3) \sim sp(2)$$

$$su(3, 1) \sim so^*(6)$$

$$su(2, 2) \sim so(4, 2)$$

$$su(4) \sim so(6)$$

$$so(3, 2) \sim sp(4, \mathbb{R})$$

$$so(4, 1) \sim sp(2, 2)$$

$$so(5) \sim sp(4)$$

$$so(6, 2) \sim so^*(8).$$

I confirmed all of these low dimensional isomorphisms with my program. The following trivial isomorphism of  $so^*(2)$  was also found to be

$$su^*(2) \sim su(2) \sim so(3) \sim sp(2).$$

## Chapter 7

### Programs Explanation and Maple Demonstrations

The programs that I wrote are:

```

RootSort(PR,T,P,RSD),
NewCSAImRoot(CSA,T0,root),
NewCSARealRoot(CSA,T0,root),
MaximalCartanSubalgebra(CSA,T,P,noncompact),
CalculateSatakeDataForRealSimpleLieAlgebra(LA,CSA0,RSD0,PR0,SR0),
SatakeArrows(DotsSRTypeDimCartan),
PossibleSemiSimpleLieAlgebraGivenComplexClassAndDimCartan(AlgType, DimCartan),
SatakeInfo(alg),
ClassifyRealSemiSimpleLieAlgebra(alg,CSA0,RSD0),
    sln(n),
    sup(n),
    supq(n,m),
    supp(n),
    slm(n),
    sop(n,m),
    so2l(n,m),
    sop(n),
    sopq(n,m),
    soleo(n,m),
    soll(n,m),
    sopq2(n,m),
    so2le(n),
    so2lo(n),
    spl(n),

```

sppp(n),  
 sppq(n,m),  
 spp(n,m).

We will give a description of these with a running example. Before we start, we need to get some information about the Lie algebra we will be considering through this example,  $so(4, 2)$ .

```
>LD1 := SimpleLieAlgebraData("so(4,2)",so42);
>DGsetup(LD1);
```

*Liealgebra : so42*

This Maple command initializes the Lie algebra  $so(4, 2)$ . Now we find the matrices that represent the standard representation.

```
> M := StandardRepresentation(so42);
```

$$M = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$



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

The Cartan decomposition is not based on the choice of Cartan subalgebra. It is given here by the Maple command.

```
>T,P := CartanDecomposition(M,so42):
```

$$T, P := [e2 - e3, e5 + e6, e7 + e11, e8 + e12, e9 + e13, e10 + e14, e15], \\
[e1, e2 + e3, e4, e5 - e6, e7 - e11, e8 - e12, e9 - e13, e10 - e14]$$

This is our choice of Cartan subalgebra.

```
>CSA := evalDG([e15, e1+e4, e2-e3]);
```

$$CSA := [e15, e1 + e4, e2 - e3]$$

From the Cartan subalgebra we calculate the root space decomposition. It is a table where the indices are a list of the roots and the corresponding root spaces.

>RSD := RootSpaceDecomposition(CSA):

$RSD := \text{table}([ [I, -1, -I] = e11 - Ie12 - Ie13 - e14, [I, -1, I] = e11 - Ie12 + Ie13 + e14, [-I, 1, -I] = e7 + Ie8 - Ie9 + e10, [-I, -1, -I] = e11 + Ie12 - Ie13 + e14, [0, 0, 2I] = e1 + Ie2 + Ie3 - e4, [-I, -1, I] = e11 + Ie12 + Ie13 - e14, [I, 1, -I] = e7 - Ie8 - Ie9 - e10, [0, 2, 0] = e5, [I, 1, I] = e7 - Ie8 + Ie9 + e10, [-I, 1, I] = e7 + Ie8 + Ie9 - e10, [0, 0, -2I] = e1 - Ie2 - Ie3 - e4, [0, -2, 0] = e6)$

We calculate the positive roots as outlined in Chapter 3 (3.2.2) using the Maple command.

>PR := PositiveRoots(RSD);

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

This is all the information we need to run the first program.

**RootSort(PR, T, P, RSD)**

**Inputs:**

PR- This is a list of vectors specifying a choice of positive roots for the root space decomposition.

T - This is a list of vectors in a Lie algebra defining a subalgebra on which the Killing form is negative-definite.

P - This is a list of vectors in a Lie algebra defining a subspace on which the Killing form is positive-definite.

RSD - This is a table specifying the root space decomposition of the Lie algebra with respect to the Cartan subalgebra.

**Outputs:**

Real Roots- This is a list of vectors in a Lie algebra defining the real roots for the root space decomposition.

Imaginary Roots- This is a list of vectors in a Lie algebra defining the imaginary roots for the root space decomposition.

**Description:**

This program sorts the list of positive roots into two lists. The first is a list of real roots and the second is a list of the imaginary roots. The complex roots are discarded. We

need the imaginary roots to complete the Cayley transform and increase the intersection of the Cartan subalgebra with the  $\mathfrak{p}$  part of the Cartan decomposition by one. We need the real roots to complete the Cayley transform to increase the intersection of the Cartan subalgebra with the  $\mathfrak{t}$  part of the Cartan decomposition by one.

**Example:**

`>Real,Imag := RootSort(PR,T,P,RSD);`

$$Real, Imag := \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 2I \end{bmatrix}$$

**NewCSAImRoot(CSA,T,P,ImRoot)**

**Inputs:**

CSA - This is a list of vectors defining a Cartan subalgebra of a Lie algebra.

T - This is a list of vectors in a Lie algebra defining a subalgebra on which the Killing form is negative-definite.

P - This is a list of vectors in a Lie algebra defining a subspace on which the Killing form is positive-definite.

ImRoot - This is a vector in a Lie algebra from the list of imaginary roots for the root space decomposition.

**Output:**

CSA- This is a Cartan subalgebra that has an increase in the intersection with  $\mathfrak{p}$  by 1 dimension.

$\beta$  - This is the root  $\beta$  used to complete the Cayley transform.

**Description:**

This program completes the Cayley transform using a noncompact imaginary root  $\beta$ . If  $\beta$  is a noncompact imaginary root, then the Cayley transform is completed to find the new Cartan subalgebra. This is done by finding the root space  $E_\beta$  and the complex conjugate of this  $\overline{E_\beta}$  and the root to Cartan subalgebra element  $H_\beta$ . The program checks to make sure these have the appropriate coefficients to satisfy the structure equations

$$[H_\beta, E_\beta] = 2E_\beta, [H_\beta, \overline{E_\beta}] = -2\overline{E_\beta}, [E_\beta, \overline{E_\beta}] = H_\beta.$$

The program then uses the null space of the noncompact imaginary root  $\beta$  to get the

portion of the given Cartan subalgebra that will be in the new Cartan subalgebra. The output of the program is a Cartan subalgebra that has an increase in the intersection with  $\mathfrak{p}$  by 1 along with the root  $\beta$  used to get there.

**Example:**

`>NewCSAImRoot(CSA,T,P,Imag[1]);`

$$[e1, e4, e15], \begin{bmatrix} 0 \\ 0 \\ 2I \end{bmatrix}$$

**NewCSARealRoot(CSA,T,P,RealRoot)**

**Inputs:**

CSA - This is a list of vectors defining a Cartan subalgebra of a Lie algebra.

T - This is a list of vectors in a Lie algebra defining a subalgebra on which the Killing form is negative-definite.

P - This is a list of vectors in a Lie algebra defining a subspace on which the Killing form is positive-definite.

RealRoot - This is a vector in a Lie algebra from the list of real roots for the root space decomposition.

**Output:**

CSA- This is a Cartan subalgebra that has an increase in the intersection with  $\mathfrak{p}$  by 1.

$\beta$  - This is the root  $\beta$  used to complete the Cayley transform.

**Description:**

This program completes the Cayley transform using a real root  $\alpha$ . This is done by finding the root space  $E_\alpha$ . Then we use the Cartan involution  $\Theta$  to find  $\Theta(E_\alpha)$  and  $H_\alpha$  is the root to Cartan subalgebra element  $H$ . The program checks to make sure these have the appropriate coefficients to satisfy the structure equations

$$[H_\alpha, E_\alpha] = 2E_\alpha, [H_\alpha, \Theta(E_\alpha)] = -2\Theta(E_\alpha), [E_\alpha, \Theta(E_\alpha)] = -H_\alpha.$$

The program then uses the null space of the noncompact imaginary root  $\alpha$  to get the portion of the given Cartan subalgebra that will be in the new Cartan subalgebra. The output of the program is a Cartan subalgebra that has an increase in the intersection with  $\mathfrak{t}$  by 1 dimension along with the root  $\alpha$  used to get there.

**Example:**

>NewCSARealRoot(CSA,T,P,Real[1]);

$$[e_2 - e_3, e_5 + e_6, e_{15}], \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$$

**MaximalCartanSubalgebra(CSA,T,P,Type)**

**Inputs:**

CSA - This is a list of vectors defining a Cartan subalgebra of a Lie algebra.

T - This is a list of vectors in a Lie algebra defining a subalgebra on which the Killing form is negative-definite.

P - This is a list of vectors in a Lie algebra defining a subspace on which the Killing form is positive-definite.

Type - This is a string giving the type of maximal Cartan subalgebra the user desires, either compact or noncompact.

**Output:**

CSA- This is a maximally compact or noncompact Cartan subalgebra.

$\beta$  - This is the last root  $\beta$  used to complete the Cayley transform.

**Description:**

The program runs either one of the previous two programs as many times as is necessary to obtain either a maximally compact or maximally noncompact Cartan subalgebra along with the last root used to get there.

**Example:**

>MaximalCartanSubalgebra(CSA,T,P, "compact");

$$[e_2 - e_3, e_5 + e_6, e_{15}], \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$$

>MaximalCartanSubalgebra(CSA,T,P, "noncompact");

$$[e_1, e_4, e_{15}], \begin{bmatrix} 0 \\ 0 \\ 2I \end{bmatrix}$$

### CalculateSatakeDataForRealSimpleLieAlgebra(alg,CSA,RSD,PR,SR)

#### Inputs:

alg - This is a string that is the name of an initialized Lie algebra.

CSA - This is a list of vectors defining a Cartan subalgebra of a Lie algebra.

RSD - This is a table, specifying the root space decomposition of the Lie algebra with respect to the Cartan subalgebra.

PR - This is a list of vectors, specifying a choice of positive roots for the root space decomposition.

#### Output:

ColorArray- This is an array of the colors of the simple roots in the proper order.

SR- This is a list of vectors, specifying the simple roots for the root space decomposition.

$\beta$  - This is the last root  $\beta$  used to complete the Cayley transform.

Type- This is a string representing the class of the given simple Lie algebra, either A, B, C or D.

DimCartan - This is the dimension of the Cartan subalgebra for the given Lie algebra.

#### Description:

This program finds the simple roots for the given simple Lie algebra using the maximally noncompact Cartan subalgebra. The correct order of the simple roots is determined. Then the simple roots are assigned colors. A black dot is assigned for imaginary roots, and a white dot for a real or complex root. The program returns an array of these colors in the proper order as well as the simple roots, the Lie algebra type and the dimension of the Cartan subalgebra.

#### Example:

>A :=

CalculateSatakeDataForRealSimpleLieAlgebra(so42,CSA,RSD,PR,SR);

$$A := ["white", "white", "white"], \begin{bmatrix} 0 \\ 1 \\ I \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -I \end{bmatrix}, "A", 3$$

### SatakeArrows(ColorSRRootTypeDimCartan)

#### Inputs:

A[1] = ColorArray - This is an array of the colors of the simple roots in the proper order.

A[2] = SR - This is a list of vectors, specifying the simple roots for the root space

decomposition.  $A[3] = \text{Root}$  - This is the last root  $\beta$  used to complete the Cayley transform.

$A[4] = \text{Type}$  - This is a string representing the class of the given simple Lie algebra, either A, B, C or D.

$A[5] = \text{DimCartan}$  - This is the dimension of the Cartan subalgebra for the given Lie algebra.

**Output:**

SatakeAssociates- This is a list of the form  $[[a,b] .. [c,d]]$ , where a is the Satake associate of b, and c is the Satake associate of d.

**Description:**

The Satake arrows program takes as it's input all of the data from the previous program. The program goes through the list of simple roots and determines if these roots have a Satake associate. The program returns a list of these associates.

**Example:**

`>SatakeArrows(A);`

`[[1, 3]]`

**PossibleSemiSimpleLieAlgebraGivenComplexClassAndDimCartan(Type,DimCartan)**

**Inputs:**

Type- This is a string representing the class of the given simple Lie algebra, either A, B, C or D.

DimCartan - This is the dimension of the Cartan subalgebra for the given Lie algebra.

**Output:**

Algebras- This is a list of the form  $["alg(1)" .. "alg(3)"]$ , where each alg is a possible Lie algebra given the type and the dimension of the Cartan subalgebra.

**Description:**

Given the Lie algebra type, A, B, C or D, and the dimension of the Cartan subalgebra, this program gives a list of all possible real simple Lie algebras that fit these restrictions.

**Example:**

`>PossibleSemiSimpleLieAlgebraGivenComplexClassAndDimCartan(A[3],A[4]);`

`["sl(4)" "su(4)" "su*(4)" "su(3,1)" "su(2,2)"]`

**SatakeInfo("alg")**

**Inputs:**

alg- This is a string representing a Lie algebra.

**Output:**

Dots -This is a list of the form ["white" .. "black"], where each color is the color of the simple root for the given Lie algebra.

Arrows - This is a list of the form [[a,b] .. [c,d]], where a is the Satake associate of b, and c is the Satake associate of d, etc. for the given Lie algebra.

**Description:**

This program calls one of a series of small programs that get the color list of the dots and the list of Satake associates for a given real simple Lie algebras.

**Example:**

```
>SatakeInfo("sl(4)");
```

```
["white", "white", "white"], []
```

```
>SatakeInfo("su(4)");
```

```
["black", "black", "black"], []
```

```
>SatakeInfo("su*(4)");
```

```
["black", "white", "black"], []
```

```
>SatakeInfo("su(3,1)");
```

```
["white", "black", "white"], [[1, 3]]
```

```
>SatakeInfo("su(2,2)");
```

```
["white", "white", "white"], [[1, 3]]
```

```
>SatakeInfo("so(4,2)");
```

```
["white", "white", "white"], [[2, 3]]
```



### **ClassifyRealSemiSimpleLieAlgebra(alg,CSA)**

#### **Inputs:**

alg - This is a string giving the name of an initialized Lie algebra.

CSA - This an optional input of a list of vectors, defining a Cartan subalgebra of a Lie algebra.

#### **Output:**

ClassifiedLieAlgebra - A list of isomorphic Lie algebras fitting the structure of the input Lie algebra.

#### **Description:**

The program `ClassifyRealSemiSimpleLieAlgebra` takes a given initialized Lie algebra as it's input with the option to give a Cartan subalgebra as the second input. The program then matches up the given algebras dot colors and the Satake associates with the list of possibles using all the other programs described above. The output is a list of isomorphic Lie algebras fitting the structure of the input Lie algebra.

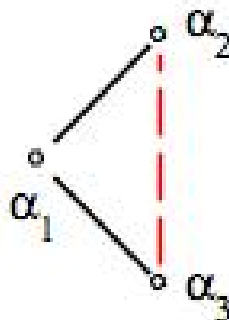
#### **Example:**

```
>ClassifyRealSemiSimpleLieAlgebra(so42,evalDG([e15,e1+e4, e2-e3]));
```

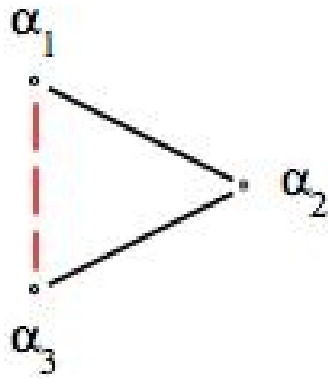
```
[ "so(4,2)", "su(2,2)" ]
```

This has two simple Lie algebras as an output because there exists an isomorphism between them. To show this we give the Satake diagrams for both of these Lie algebras using the Maple command.

```
>SatakeDiagram("so(4,2)");
```



```
>SatakeDiagram("su(2,2)");
```



This completes the Maple demonstration and explanation of the programs that I wrote to classify real simple Lie algebras using the Cayley transform.

## Bibliography

- [1] Luis Boza, Eugenio Fedriani, Juan Nunez, and Angel Tenorio. A historical review of the classifications of Lie algebras. *Revista De La Unio'n Matema'tica Argentina*, 54(2):75–99, 2013.
- [2] Andreas Cap and Jan Slovák. *Parabolic Geometries I: Background and General Theory*. American Mathematical Society, 2009.
- [3] Qizhen He. Lie algebras and Lie brackets of Lie groups-matrix groups.
- [4] Sigurdur Helgason. *Differential Geometry, Lie Groups, and Symmetric Spaces*. Academic Press, INC, 1978.
- [5] Anthony Knapp. *Lie Groups: Beyond an Introduction*. Birkhauser, 1996.

APPENDICES

## Maple Demonstration

The following four worksheets give low dimensional examples of the classification program for the seventeen (17) types from among the four (4) real simple classical Lie algebras. The classification program takes an initialized real simple Lie algebra as an input with a Cartan subalgebras as an optional input. The output will display a list of real simple Lie algebras which are isomorphic to the input real simple Lie algebra.

### 7.1 A

In this worksheet we give the low dimensional examples of the classification program for the real simple classical Lie algebras of type A.

Each of the examples are organized as follows:

1. First we demonstrate that the classification program works for the standard library. This will give us a controlled result to compare against.
2. Next, we find a basis for an isomorphic Lie algebra. We test the program to confirm that the classification remains the same.
3. To properly classify any real simple Lie algebra using a Satake diagram, a maximally noncompact Cartan subalgebra must be used. Because of this, a third test was performed to check the functionality of all the crucial parts of the classification program. To make sure this testing occurs, I used the `MaximalCartanSubalgebra` program to find a maximally compact Cartan subalgebra for the given Lie algebra. Using this maximally compact Cartan subalgebra forces the classification program to execute a Cayley transform the maximum number of times that would be possible for each real simple Lie algebra.

## Introduction

```

[
[> with(DifferentialGeometry) :
[> with(LieAlgebras) :
[> with(LinearAlgebra) :
[> read "/Users/Desktop/research files/Hannah.txt";
(1.1)

```

## A

### $sl(L + 1, R)$

#### $sl(5)$

We start by loading the structure constants for Lie algebra  $sl(5)$  from the standard Maple library.

```

[> LD1 := SimpleLieAlgebraData("sl(5)", s15) :
[> DGsetup(LD1) :
[> ClassifyRealSemiSimpleLieAlgebra(s15) ;
[> "sl(5)"
(2.1.1.1)

```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```

[> ChangeFrame(s15) :
[> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e8,
[> e9, e15, e11, e12, e10, e14, e13, e16, e17,
[> e18, e19, e20, e21, e22, e23, e24]) :
[> LD2 := LieAlgebraData(B, s15n) :
[> DGsetup(LD2) :
[> ClassifyRealSemiSimpleLieAlgebra(s15n) ;
[> "sl(5)"
(2.1.1.2)

```

The output is the expected result, namely that it matches the control.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```

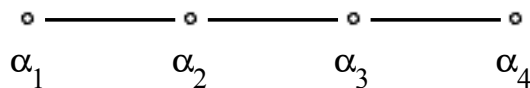
[> ChangeFrame(s15n) :
[> CSA := CartanSubalgebra(s15) ;
[> CSA := [e1, e2, e3, e4]
(2.1.1.3)

```

```

> M := StandardRepresentation(sl5) :
> T, P := CartanDecomposition(M, sl5) :
  T;
  P;
[e5 - e9, e6 - e13, e7 - e17, e8 - e21, e10 - e14, e11 - e18,
  e12 - e22, e15 - e19, e16 - e23, e20 - e24]
[e1, e2, e3, e4, e5 + e9, e6 + e13, e7 + e17, e8 + e21, e10      (2.1.1.4)
  + e14, e11 + e18, e12 + e22, e15 + e19, e16 + e23, e20
  + e24]
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
  "compact");
NewCSA, V := [e1 + e4, e2 + e3, e7 - e17, e10 - e14], [ ] (2.1.1.5)
> ClassifyRealSemiSimpleLieAlgebra(sl5, evalDG
  (NewCSA));
                                     "sl(5)" (2.1.1.6)
> SatakeDiagram("sl(5)");

```



### $sl(4) \cong so(3,3)$

We load the structure constants for Lie algebra  $sl(4)$  from the standard Maple library. Because there is an isomorphism between  $sl(4)$  and  $so(3,3)$ , we expect the program `ClassifyRealSemiSimpleLieAlgebras` to display both of these as the output.

We begin by initializing  $sl(4)$  using the standard library.

```

> LD1 := SimpleLieAlgebraData("sl(4)", sl4) :
> DGsetup(LD1) :
First we check that the program works for the known initialized Lie
algebra to establish our controlled test.
> ClassifyRealSemiSimpleLieAlgebra(sl4) ;
["sl(4)", "so(3,3)"] (2.1.2.1)

```

Now we create a Lie algebra which is isomorphic to the algebra we just tested.

```

> ChangeFrame(sl4) :
> B := evalDG([e3, e1, e2, e4, e5, e6, e7, e8,
  e9, e15, e11, e12, e10, e14, e13]) :
> LD2 := LieAlgebraData(B, sl4n) :

```

```
> DGsetup(LD2) :
```

We check the program on the new Lie algebra and confirm that the classification returns the same results as the control.

```
> ClassifyRealSemiSimpleLieAlgebra(sl4n);
      ["sl(4)", "so(3,3)"] (2.1.2.2)
```

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(sl4n) :
> CSA := CartanSubalgebra(sl4) ;
      CSA := [e1, e2, e3] (2.1.2.3)
```

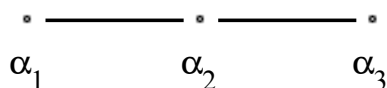
```
> M := StandardRepresentation(sl4) :
> T, P := CartanDecomposition(M, sl4) :
      T;
      P;
      [e4 - e7, e5 - e10, e6 - e13, e8 - e11, e9 - e14, e12 - e15]
      [e1, e2, e3, e4 + e7, e5 + e10, e6 + e13, e8 + e11, e9 + e14, e12
      + e15] (2.1.2.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
      "compact");
      NewCSA, V := [e1 - e2 - e3, e6 - e13, e8 - e11], [ ] (2.1.2.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra(sl4, evalDG
      (NewCSA));
      ["sl(4)", "so(3,3)"] (2.1.2.6)
```

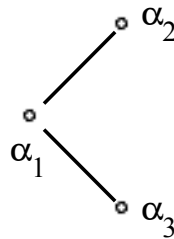
Here are the Satake diagrams that illustrate the isomorphism between  $\mathfrak{sl}(4)$  and  $\mathfrak{so}(3,3)$ .

```
> SatakeDiagram("sl(4)");
```



```
> SatakeDiagram("so(3,3)");
```





$sl(3)$

We load the structure constants for Lie algebra  $sl(3)$  from the standard Maple library.

```
> LD1 := SimpleLieAlgebraData("sl(3)", s13);
> DGsetup(LD1);
```

First we check that the program works for the known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(s13);
"sl(3)" (2.1.3.1)
```

Now we create a Lie algebra which is isomorphic to the algebra we just tested.

```
> ChangeFrame(s13);
> B := evalDG([e3, e1, e2, e4, e5, e6, e8, e7]);
> LD2 := LieAlgebraData(B, s13n);
> DGsetup(LD2);
> ClassifyRealSemiSimpleLieAlgebra(s13n);
"sl(3)" (2.1.3.2)
```

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(s13n);
> CSA := CartanSubalgebra(s13);
CSA := [e1, e2] (2.1.3.3)
```

```
> M := StandardRepresentation(s13);
> T, P := CartanDecomposition(M, s13);
T;
P;
[e3 - e5, e4 - e7, e6 - e8]
[e1, e2, e3 + e5, e4 + e7, e6 + e8] (2.1.3.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
NewCSA, V := [e1 - (1/2)e2, e6 - e8], [] (2.1.3.5)
```

```

> ClassifyRealSemiSimpleLieAlgebra(s13, evalDG
(NewCSA));
      "sl(3)"
(2.1.3.6)
> SatakeDiagram("sl(3)");

```



$sl(2) \cong so(2,1) \cong sp(2,R)$

We load the structure constants for Lie algebra  $sl(2)$  from the standard Maple library. There is an isomorphism between  $sl(2)$ ,  $so(2,1)$ , and  $sp(2,R)$ , so all three of these Lie algebras will be the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```

> LD1 := SimpleLieAlgebraData("sl(2)", s12);
> DGsetup(LD1);

```

First we check that the program works for the known initialized Lie algebra.

```

> ClassifyRealSemiSimpleLieAlgebra(s12);
["sl(2)", "so(2,1)", "sp(2,R)"]
(2.1.4.1)

```

Now we change the basis of the Lie algebra and run the program again.

```

> ChangeFrame(s12);
> B := evalDG([e3-e2, e1+e2, e2]);
> LD2 := LieAlgebraData(B, s12n);
> DGsetup(LD2);
> ClassifyRealSemiSimpleLieAlgebra(s12n);
["sl(2)", "so(2,1)", "sp(2,R)"]
(2.1.4.2)

```

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```

> ChangeFrame(s12n);
> CSA := CartanSubalgebra(s12);
      CSA := [e1]
(2.1.4.3)

```

```

> M := StandardRepresentation(s12);
> T, P := CartanDecomposition(M, s12);
      T;
      P;
      [e2 - e3]
      [e1, e2 + e3]
(2.1.4.4)

```

```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
    "compact");
    NewCSA, V := [e2 - e3], [ 0 ]           (2.1.4.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra(sl2, evalDG
    (NewCSA));
    ["sl(2)", "so(2,1)", "sp(2,R)"]       (2.1.4.6)
```

Here are the Satake diagrams that show the isomorphisms between  $\mathfrak{sl}(2)$ ,  $\mathfrak{so}(2,1)$ , and  $\mathfrak{sp}(2, \mathbb{R})$ .

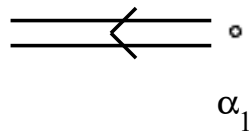
```
> SatakeDiagram("sl(2)");
```

$$\begin{array}{c} \circ \\ \alpha_1 \end{array}$$

```
> SatakeDiagram("so(2,1)");
```

$$\begin{array}{c} \Rightarrow \circ \\ \alpha_1 \end{array}$$

```
> SatakeDiagram("sp(2, R)");
```



### ▼ $\mathfrak{sl}(m, \mathbb{H})$

#### ▼ $\mathfrak{su}^*(4) \cong \mathfrak{so}(5, 1)$

We load the structure constants for Lie algebra  $\mathfrak{su}^*(4)$  from the standard Maple library. There is an isomorphism between  $\mathfrak{su}^*(4)$  and  $\mathfrak{so}(5, 1)$  so these two Lie algebras will display as the output of `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("su*(4)", su4) :
```

```
> DGsetup(LD1) :
```

First we check that the program works for the well known initialized Lie algebra  $\mathfrak{su}^*(4)$ .

```
> ClassifyRealSemiSimpleLieAlgebra(su4) ;
      ["so(5,1)", "su*(4)"] (2.2.1.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```

> ChangeFrame (su4) :
> B := evalDG([e3, e1, e2, e4, e5, e6, e7, e8,
e9, e15, e11, e12, e10, e14, e13]):
> LD2 := LieAlgebraData (B, su4n) :
> DGsetup (LD2) :
> ClassifyRealSemiSimpleLieAlgebra (su4n) ;
["so(5,1)", "su*(4)"] (2.2.1.2)

```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```

> ChangeFrame (su4n) :
> CSA := CartanSubalgebra (su4) ;
CSA := [e1, e2, e3] (2.2.1.3)

```

```

> M := StandardRepresentation (su4) :
> T, P := CartanDecomposition (M, su4) :
T;
P;
[e1, e2, e4 - e5, e6 + e7, e8, e9 + e10, e11, e12, e13 + e14,
e15]
[e3, e4 + e5, e6 - e7, e9 - e10, e13 - e14] (2.2.1.4)

```

```

> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
"compact") ;
NewCSA, V := [e1, e2, e3], [ ] (2.2.1.5)

```

```

> ClassifyRealSemiSimpleLieAlgebra (su4, evalDG
(NewCSA)) ;
["so(5,1)", "su*(4)"] (2.2.1.6)

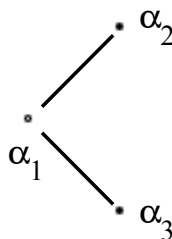
```

Here are the Satake diagrams that illustrate the isomorphism between  $\mathfrak{su}^*(4)$  and  $\mathfrak{so}(5,1)$ .

```

> SatakeDiagram ("so (5,1) ") ;

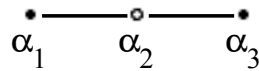
```



```

> SatakeDiagram ("su* (4) ") ;

```



$su^*(2) \cong su(2) \cong so(3) \cong sp(2)$

We load the structure constants for Lie algebra  $su^*(2)$  from the standard Maple library. There is an isomorphism between  $su^*(2)$ ,  $su(2)$ ,  $so(3)$ , and  $sp(2)$  so we have these two Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("su*(2)", su2);
> DGsetup(LD1):
```

First we check that the program works for the well known initialized Lie algebra to establish the control.

```
> ClassifyRealSemiSimpleLieAlgebra(su2);
["su(2)", "so(3)", "sp(2)", "su*(2)"] (2.2.2.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(su2):
> B := evalDG([e3, e1, e2]):
> LD2 := LieAlgebraData(B, su2n):
> DGsetup(LD2):
> ClassifyRealSemiSimpleLieAlgebra(su2n);
["su(2)", "so(3)", "sp(2)", "su*(2)"] (2.2.2.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(su2n):
> CSA := CartanSubalgebra(su2);
CSA := [e1] (2.2.2.3)
```

```
> M := StandardRepresentation(su2):
> T, P := CartanDecomposition(M, su2):
T;
P;
[e1, e2, e3]
[] (2.2.2.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
    "compact");
    NewCSA, V := [eI], [ ] (2.2.2.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra(su2, evalDG
    (NewCSA));
    ["su(2)", "so(3)", "sp(2)", "su*(2)"] (2.2.2.6)
```

Here are the Satake diagrams that illustrate the isomorphisms between  $su(2)$ ,  $so(3)$ ,  $sp(2)$ , and  $su^*(2)$ .

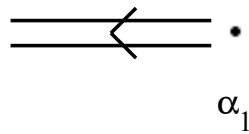
```
> SatakeDiagram("su(2)");
```

$$\alpha_1$$

```
> SatakeDiagram("so(3)");
```

$$\begin{array}{c} \Rightarrow \\ \alpha_1 \end{array}$$

```
> SatakeDiagram("sp(2)");
```



```
> SatakeDiagram("su*(2)");
```

$\alpha_1$

### ▼ $su(p, L + 1 - p)$

#### ▼ $su(3, 1) \cong so^*(6)$

We load the structure constants for Lie algebra  $su(3, 1)$  from the standard Maple library. There is an isomorphism between  $su(3, 1)$  and  $so^*(6)$  so we have these two Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("su(3,1)", su31,
```



```
'version=2') :
> DGsetup(LD1) :
```

First we check that the program works for the well known initialized Lie algebra to establish the control.

```
> ClassifyRealSemiSimpleLieAlgebra(su31) ;
      ["su(3,1)", "so*(6)"]          (2.3.1.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(su31) :
> B := evalDG([e14, e1, e2, e4, e5, e6, e7, e8,
              e9, e15, e11, e12, e10, e3, e13]) :
> LD2 := LieAlgebraData(B, su31n) :
> DGsetup(LD2) :
> ClassifyRealSemiSimpleLieAlgebra(su31n) ;
      ["su(3,1)", "so*(6)"]          (2.3.1.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(su31n) :
> CSA := CartanSubalgebra(su31) ;
      CSA := [e1, e9, e13 + 2 e14 + e15]          (2.3.1.3)
```

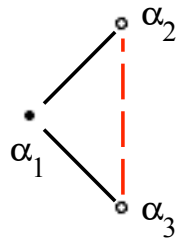
```
> M := StandardRepresentation(su31) :
> T, P := CartanDecomposition(M, su31) :
      T ;
      P ;
      [e1, e2, e3, e4, e5, e6, e13, e14, e15]
      [e7, e8, e9, e10, e11, e12]          (2.3.1.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
      "compact") ;
      NewCSA, V := [e1, e13 + 2 e14, e15], [ ]          (2.3.1.5)
```

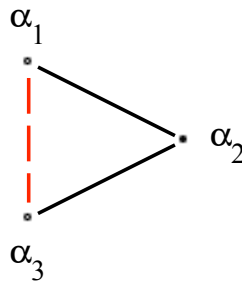
```
> ClassifyRealSemiSimpleLieAlgebra(su31, evalDG
      (NewCSA)) ;
      ["su(3,1)", "so*(6)"]          (2.3.1.6)
```

Here are the Satake diagrams that illustrate the isomorphisms between  $so^*(6)$  and  $su(3,1)$ .

```
> SatakeDiagram("so*(6) ") ;
```



```
> SatakeDiagram("su(3,1)");
```



### $su(3,2)$

We load the structure constants for Lie algebra  $su(3,2)$  from the standard Maple library.

```
> LD1 := SimpleLieAlgebraData("su(3,2)", su32);
> DGsetup(LD1);
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(su32);
"su(3,2)" (2.3.2.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(su32);
> B := evalDG([e3, e1, e2, e4, e5, e6, e7, e8,
e9, e15, e11, e12, e10, e14, e13, e16, e17,
e18, e19, e20, e21, e22, e23, e24]);
> LD2 := LieAlgebraData(B, su32n);
> DGsetup(LD2);
> ClassifyRealSemiSimpleLieAlgebra(su32n);
"su(3,2)" (2.3.2.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra

using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

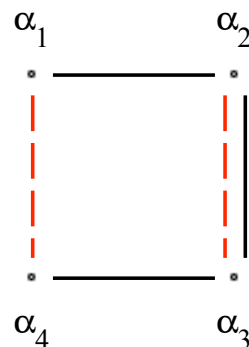
```
> ChangeFrame (su32n) :
> CSA := CartanSubalgebra (su32) ;
      CSA := [e1, e2, e3, e4] (2.3.2.3)
```

```
> M := StandardRepresentation (su32) :
> T, P := CartanDecomposition (M, su32) :
      T;
      P;
[e1, e2, e5 - e7, e6 + e10, e8 + e11, e9 + e12, e13 + e19, e14
 + e20, e15 + e17, e16 + e22, e18 + e23, e21 + e24]
[e3, e4, e5 + e7, e6 - e10, e8 - e11, e9 - e12, e13 - e19, e14 (2.3.2.4)
 - e20, e15 - e17, e16 - e22, e18 - e23, e21 - e24]
```

```
> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
      "compact");
      NewCSA, V := [e1, e2, e13 + e19, e14 + e20], [ ] (2.3.2.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra (su32, evalDG
      (NewCSA));
      "su(3, 2)" (2.3.2.6)
```

```
> SatakeDiagram ("su(3, 2)");
```



### $su(4,1)$

We load the structure constants for Lie algebra  $su(4,1)$  from the standard Maple library.

```
> LD1 := SimpleLieAlgebraData ("su(4, 1)", su41) :
> DGsetup (LD1) :
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra (su41) ; (2.3.3.1)
```

```
"su(4, 1)" (2.3.3.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame (su41) :
> B := evalDG([e14, e1, e2, e4, e5, e6, e7, e8,
e9, e15, e11, e12, e10, e3, e13, e16, e17, e18,
e19, e21, e20, e22, e23, e24]) :
> LD2 := LieAlgebraData (B, su41n) :
> DGsetup (LD2) :
> ClassifyRealSemiSimpleLieAlgebra (su41n) ;
"su(4, 1)" (2.3.3.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

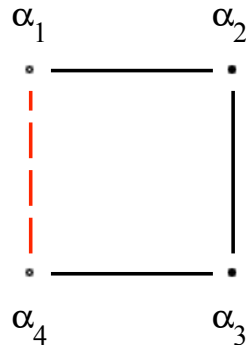
```
> ChangeFrame (su41n) :
> CSA := CartanSubalgebra (su41) ;
CSA := [e1, e2, e3, e4] (2.3.3.3)
```

```
> M := StandardRepresentation (su41) :
> T, P := CartanDecomposition (M, su41) :
T;
P;
[e1, e2, e3, e5 + e8, e6, e7, e9 + e11, e10, e12 + e13, e14
+ e15, e16 + e19, e17, e18, e20 + e22, e21, e23 + e24]
[e4, e5 - e8, e9 - e11, e12 - e13, e14 - e15, e16 - e19, e20
- e22, e23 - e24] (2.3.3.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
"compact") ;
NewCSA, V := [e1, e2, e3, e14 + e15], [ ] (2.3.3.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra (su41, evalDG
(NewCSA)) ;
"su(4, 1)" (2.3.3.6)
```

```
> SatakeDiagram ("su(4, 1)");
```



$su(4,2)$

We load the structure constants for Lie algebra  $su(4,2)$  from the standard Maple library.

```
> LD1 := SimpleLieAlgebraData("su(4,2)", su42);
> DGsetup(LD1);
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(su42);
"su(4,2)" (2.3.4.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(su42);
> B := evalDG([e14, e1, e2, e4, e5, e6, e7, e8,
e9, e15, e11, e12, e10, e3, e13, e16, e17, e18,
e19, e21, e20, e22, e23, e24, e25, e26, e27,
e28, e29, e30, e31, e32, e33, e34, e35]);
> LD2 := LieAlgebraData(B, su42n);
> DGsetup(LD2);
> ClassifyRealSemiSimpleLieAlgebra(su42n);
"su(4,2)" (2.3.4.2)
```

The output matches the control, as expected.

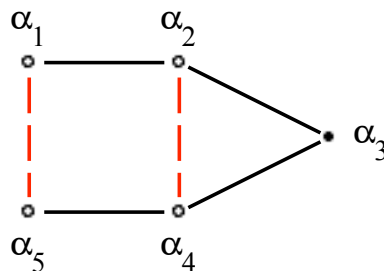
We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(su42n);
> CSA := CartanSubalgebra(su42);
CSA := [e1, e2, e3, e4, e5] (2.3.4.3)
```

```

> M := StandardRepresentation(su42) :
> T,P := CartanDecomposition(M,su42) :
  T;
  P;
[e1, e2, e3, e6 - e8, e7 + e13, e9 + e14, e10, e11 + e16, e12
 + e17, e15 + e18, e19 + e26, e20 + e27, e21 + e23, e22
 + e30, e24 + e31, e25, e28 + e33, e29 + e34, e32 + e35]
[e4, e5, e6 + e8, e7 - e13, e9 - e14, e11 - e16, e12 - e17, e15 (2.3.4.4)
 - e18, e19 - e26, e20 - e27, e21 - e23, e22 - e30, e24
 - e31, e28 - e33, e29 - e34, e32 - e35]
> NewCSA, V := MaximalCartanSubalgebra(CSA,T,P,
"compact");
  NewCSA, V := [e1, e2, e3, e19 + e26, e20 + e27], [ ] (2.3.4.5)
> ClassifyRealSemiSimpleLieAlgebra(su42, evalDG
(NewCSA));
  "su(4, 2)" (2.3.4.6)
> SatakeDiagram("su(4, 2)");

```



## su(p,p)

$su(2,2) \cong so(4,2)$

We load the structure constants for Lie algebra  $su(2,2)$  from the standard Maple library. There is an isomorphism between  $su(2,2)$  and  $so(4,2)$  so we have these two Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```

> LD1 := SimpleLieAlgebraData("su(2,2)", su22,
'Version=2');
> DGsetup(LD1);
First we check that the program works for the well known initialized
Lie algebra.
> ClassifyRealSemiSimpleLieAlgebra(su22);
  ["so(4,2)", "su(2,2)"] (2.4.1.1)

```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame (su22) :
> B := evalDG([e3, e2, e4, e1, e5, e6, e7, e8,
e9, e10, e11, e12, e13, e14, e15]):
> LD2 := LieAlgebraData (B, su22n) :
> DGsetup (LD2) :
> ClassifyRealSemiSimpleLieAlgebra (su22n) ;
      ["so(4,2)", "su(2,2)"] (2.4.1.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame (su22n) :
> CSA := CartanSubalgebra (su22) ;
      CSA := [e1, e11, e13 + 2 e14 + e15] (2.4.1.3)
```

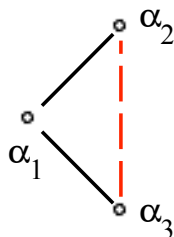
```
> M := StandardRepresentation (su22) :
> T, P := CartanDecomposition (M, su22) :
      T;
      P;
      [e1, e2, e11, e12, e13, e14, e15]
      [e3, e4, e5, e6, e7, e8, e9, e10] (2.4.1.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
"compact");
      NewCSA, V := [e1, e11, e13 + 2 e14 + e15], [ ] (2.4.1.5)
```

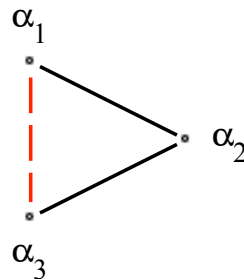
```
> ClassifyRealSemiSimpleLieAlgebra (su22, evalDG
(NewCSA));
      ["so(4,2)", "su(2,2)"] (2.4.1.6)
```

Here are the Satake diagrams that illustrate the isomorphisms between  $\mathfrak{so}(4,2)$  and  $\mathfrak{su}(2,2)$ .

```
> SatakeDiagram ("so (4, 2) ") ;
```



```
> SatakeDiagram("su(2,2)");
```



**$su(3,3)$**

We load the structure constants for Lie algebra  $su(3,3)$  from the standard Maple library.

```
> LD1 := SimpleLieAlgebraData("su(3,3)", su33);
> DGsetup(LD1);
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(su33);
"su(3,3)" (2.4.2.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(su33);
> B := evalDG([e3, e2, e6, e1, e5, e4, e7, e8,
e9, e10, e11, e12, e13, e14, e15, e16, e17,
e18, e19, e20, e21, e22, e23, e24, e25, e26,
e27, e28, e29, e30, e31, e32, e33, e34, e35]);
> LD2 := LieAlgebraData(B, su33n);
> DGsetup(LD2);
> ClassifyRealSemiSimpleLieAlgebra(su33n);
"su(3,3)" (2.4.2.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

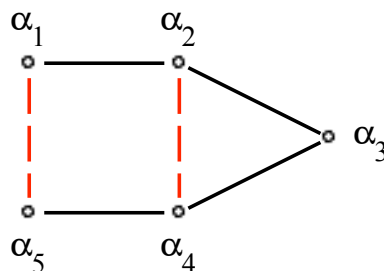
```
> ChangeFrame(su33n);
> CSA := CartanSubalgebra(su33);
(2.4.2.3)
```



```

                                CSA := [e1, e2, e3, e4, e5]                                (2.4.2.3)
> M := StandardRepresentation(su33) :
> T, P := CartanDecomposition(M, su33) :
  T;
  P;
[e1, e2, e6 - e14, e7 + e15, e8 + e16, e9 - e12, e10 - e13, e11
 + e17, e18 + e30, e19 + e31, e20 + e32, e21 + e29, e22
 + e33, e23 + e34, e24 + e27, e25 + e28, e26 + e35]
[e3, e4, e5, e6 + e14, e7 - e15, e8 - e16, e9 + e12, e10 + e13, (2.4.2.4)
 e11 - e17, e18 - e30, e19 - e31, e20 - e32, e21 - e29, e22
 - e33, e23 - e34, e24 - e27, e25 - e28, e26 - e35]
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
  "compact");
NewCSA, V := [e1, e2, e18 + e30, e19 + e31, e20 + e32], [ ] (2.4.2.5)
> ClassifyRealSemiSimpleLieAlgebra(su33, evalDG
  (NewCSA));
                                "su(3, 3)"                                (2.4.2.6)
> SatakeDiagram("su(3, 3)");

```



## ▼ su(L + 1)

▼  $su(2) \cong so(3) \cong sp(2) \cong su^*(2)$

We load the structure constants for Lie algebra  $su(2)$  from the standard Maple library. There is an isomorphism between  $su^*(2)$ ,  $su(2)$ ,  $so(3)$ , and  $sp(2)$  so we have these two Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```

[> LD1 := SimpleLieAlgebraData("su(2)", su2) :
[> DGsetup(LD1) :

```

First we check that the program works for the well known initialized Lie algebra.

```

[> ClassifyRealSemiSimpleLieAlgebra(su2);
(2.5.1.1)

```

```
["su(2)", "so(3)", "sp(2)", "su*(2)"] (2.5.1.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame (su2) :
> B := evalDG([e1-e2, e2+e1, e3]) :
> LD2 := LieAlgebraData (B, su2n) :
> DGsetup (LD2) :
> ClassifyRealSemiSimpleLieAlgebra (su2n) ;
["su(2)", "so(3)", "sp(2)", "su*(2)"] (2.5.1.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame (su2n) :
> CSA := CartanSubalgebra (su2) ;
CSA := [e1] (2.5.1.3)
```

```
> M := StandardRepresentation (su2) :
> T, P := CartanDecomposition (M, su2) :
T;
P;
[e1, e2, e3]
[] (2.5.1.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
"compact");
NewCSA, V := [e1], [] (2.5.1.5)
```

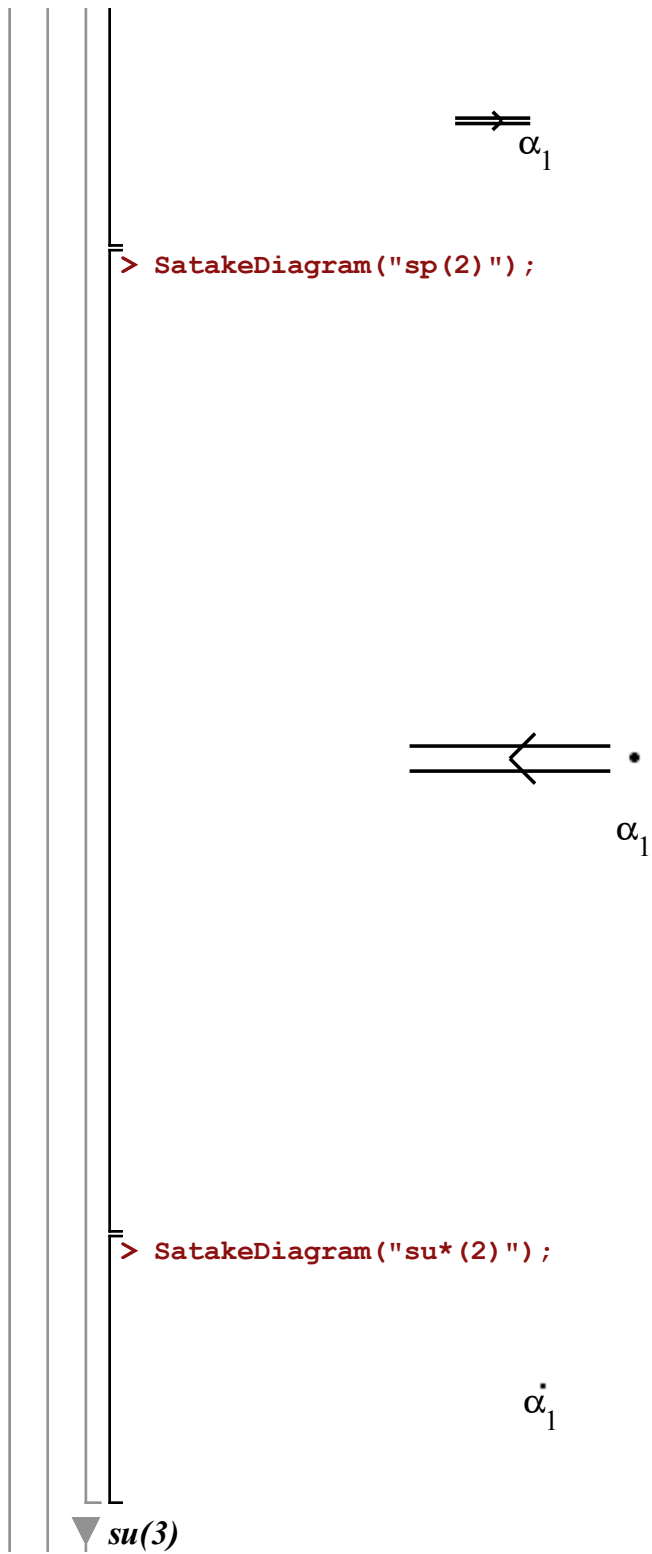
```
> ClassifyRealSemiSimpleLieAlgebra (su2, evalDG
(NewCSA)) ;
["su(2)", "so(3)", "sp(2)", "su*(2)"] (2.5.1.6)
```

Here are the Satake diagrams that illustrate the isomorphisms between  $\mathfrak{su}(2)$ ,  $\mathfrak{so}(3)$ ,  $\mathfrak{sp}(2)$ , and  $\mathfrak{su}^*(2)$ .

```
> SatakeDiagram ("su (2) ") ;
```

$$\alpha_1$$

```
> SatakeDiagram ("so (3) ") ;
```



We load the structure constants for Lie algebra  $su(3,3)$  from the standard Maple library.

```
> LD1 := SimpleLieAlgebraData("su(3)", su3);
> DGsetup(LD1):
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(su3);
"su(3)" (2.5.2.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(su3):
> B := evalDG([e3, e2, e1, e4, e8, e6, e7, e5]):
> LD2 := LieAlgebraData(B, su3n):
> DGsetup(LD2):
> ClassifyRealSemiSimpleLieAlgebra(su3n);
"su(3)" (2.5.2.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(su3n):
> CSA := CartanSubalgebra(su3);
CSA := [e1, e2] (2.5.2.3)
```

```
> M := StandardRepresentation(su3):
> T, P := CartanDecomposition(M, su3):
T;
P;
[e1, e2, e3, e4, e5, e6, e7, e8]
[] (2.5.2.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
NewCSA, V := [e1, e2], [] (2.5.2.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra(su3, evalDG
(NewCSA));
"su(3)" (2.5.2.6)
```

```
> SatakeDiagram("su(3)");
```

$$\begin{array}{c} \cdot \text{-----} \cdot \\ \alpha_1 \qquad \qquad \alpha_2 \end{array}$$

$su(4) \cong so(6)$

We load the structure constants for Lie algebra  $su(4)$  from the standard Maple library. There is an isomorphism between  $su(4)$  and  $so(6)$  so we have these two Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("su(4)", su4,
'version=2');
> DGsetup(LD1):
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(su4);
["su(4)", "so(6)"] (2.5.3.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(su4):
> B := evalDG([e3, e5, e1, e4, e2, e6, e7, e8,
e9, e10, e11, e12, e13, e15, e14]):
> LD2 := LieAlgebraData(B, su4n):
> DGsetup(LD2):
> ClassifyRealSemiSimpleLieAlgebra(su4n);
["su(4)", "so(6)"] (2.5.3.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(su4n):
> CSA := CartanSubalgebra(su4);
CSA := [e1, e2, e3] (2.5.3.3)
> M := StandardRepresentation(su4):
> T, P := CartanDecomposition(M, su4):
T;
P;
[e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12, e13, e14, e15]
```

```
[ ] (2.5.3.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,  
"compact");
```

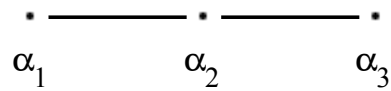
```
NewCSA, V := [e1, e2, e3], [ ] (2.5.3.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra(su4, evalDG  
(NewCSA));
```

```
["su(4)", "so(6)"] (2.5.3.6)
```

Here are the Satake diagrams that illustrate the isomorphisms between su(4) and so(6).

```
> SatakeDiagram("su(4)");
```



## 7.2 B

In this worksheet we give the low dimensional examples of the classification program for the real simple classical Lie algebras of type B.

Each of the examples are organized as follows:

1. First we demonstrate that the classification program works for the standard library. This will give us a controlled result to compare against.
2. Next, we find a basis for an isomorphic Lie algebra. We test the program to confirm that the classification remains the same.
3. To properly classify any real simple Lie algebra using a Satake diagram, a maximally noncompact Cartan subalgebra must be used. Because of this, a third test was performed to check the functionality of all the crucial parts of the classification program. To make sure this testing occurs, I used the MaximalCartanSubalgebra program to find a maximally compact Cartan subalgebra for the given Lie algebra. Using this maximally compact Cartan subalgebra forces the classification program to execute a Cayley transform the maximum number of times that would be possible for each real simple Lie algebra.

## B

## so(p,2L + 1 - p)

$so(2,1) \cong sl(2) \cong sp(2,R)$

We load the structure constants for Lie algebra  $so(2,1)$  from the standard Maple library. There is an isomorphism between  $sl(2)$ ,  $so(2,1)$ , and  $sp(2,R)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("so(2,1)", so21,
  'version=2'):
> DGsetup(LD1):
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so21);
["sl(2)", "so(2,1)", "sp(2,R)"] (1.1.1.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so21):
> B := evalDG([e2, e1, e3]):
> LD2 := LieAlgebraData(B, so21n):
> DGsetup(LD2):
> ClassifyRealSemiSimpleLieAlgebra(so21n);
["sl(2)", "so(2,1)", "sp(2,R)"] (1.1.1.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(so21n):
> CSA := CartanSubalgebra(so21);
CSA := [e1] (1.1.1.3)
```

```
> M := StandardRepresentation(so21):
> T, P := CartanDecomposition(M, so21):
T;
P;
[e1]
[e2, e3] (1.1.1.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
```



```
"compact");
      NewCSA, V:= [eI], [ ]           (1.1.1.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra(so21, evalDG
      (NewCSA) );
      ["sl(2)", "so(2,1)", "sp(2,R)"]   (1.1.1.6)
```

Here are the Satake diagrams that illustrate the isomorphisms between  $sl(2)$ ,  $so(2,1)$ , and  $sp(2, \mathbb{R})$ .

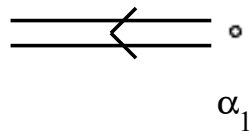
```
> SatakeDiagram("sl(2)");
```

$$\begin{array}{c} \circ \\ \alpha_1 \end{array}$$

```
> SatakeDiagram("so(2,1)");
```

$$\begin{array}{c} \Rightarrow \circ \\ \alpha_1 \end{array}$$

```
> SatakeDiagram("sp(2,R)");
```



▼  $so(3,2) \cong sp(4,R)$

We load the structure constants for Lie algebra  $so(3,2)$  from the standard Maple library. There is an isomorphism between  $so(3,2)$  and  $sp(4,R)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("so(3,2)", so32,
    'version=2');
> DGsetup(LD1):
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so32);
    ["so(3,2)", "sp(4,R)"] (1.1.2.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so32):
```

```

> B := evalDG([e2, e1, e3, e4, e6, e5, e7, e8,
e9, e10]);
> LD2 := LieAlgebraData(B, so32n) :
> DGsetup(LD2) :
> ClassifyRealSemiSimpleLieAlgebra(so32n) ;
      ["so(3,2)", "sp(4,R)"] (1.1.2.2)

```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```

> ChangeFrame(so32n) :
> CSA := CartanSubalgebra(so32) ;
      CSA := [e1, e8] (1.1.2.3)

```

```

> M := StandardRepresentation(so32) :
> T, P := CartanDecomposition(M, so32) :
T;
P;
      [e1, e2, e3, e10]
      [e4, e5, e6, e7, e8, e9] (1.1.2.4)

```

```

> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
      NewCSA, V := [e1, e10], [ ] (1.1.2.5)

```

```

> ClassifyRealSemiSimpleLieAlgebra(so32, evalDG
(NewCSA) );
      ["so(3,2)", "sp(4,R)"] (1.1.2.6)

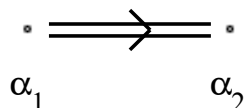
```

Here are the Satake diagrams that illustrate the isomorphisms between  $so(3,2)$ , and  $sp(4, \mathbb{R})$ .

```

> SatakeDiagram("so(3,2)");

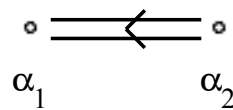
```



```

> SatakeDiagram("sp(4,R)");

```



▼  $so(4,1) \cong sp(2,2)$

We load the structure constants for Lie algebra  $so(4,1)$  from the standard Maple library. There is an isomorphism between  $so(4,1)$  and  $sp(2,2)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("so(4,1)", so41,
    'version=2');
> DGsetup(LD1):
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so41);
    ["so(4,1)", "sp(2,2)"]
```

(1.1.3.1)

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so41):
```

```

> B := evalDG([e2, e1, e3, e5, e4, e6, e7, e8,
e9, e10]):
> LD2 := LieAlgebraData(B,so41n):
> DGsetup(LD2):
> ClassifyRealSemiSimpleLieAlgebra(so41n);
      ["so(4,1)", "sp(2,2)"]           (1.1.3.2)

```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```

> ChangeFrame(so41n):
> CSA := CartanSubalgebra(so41);
      CSA := [e1, e6]                 (1.1.3.3)

```

```

> M := StandardRepresentation(so41):
> T,P := CartanDecomposition(M,so41):
      T;
      P;
      [e1, e2, e3, e4, e5, e6]
      [e7, e8, e9, e10]              (1.1.3.4)

```

```

> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
      NewCSA, V := [e1, e6], [ ]     (1.1.3.5)

```

```

> ClassifyRealSemiSimpleLieAlgebra(so41, evalDG
(NewCSA));
      ["so(4,1)", "sp(2,2)"]       (1.1.3.6)

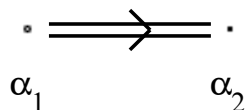
```

Here are the Satake diagrams that illustrate the isomorphisms between  $\mathfrak{so}(4,1)$ , and  $\mathfrak{sp}(2,2)$ .

```

> SatakeDiagram("so(4,1)");

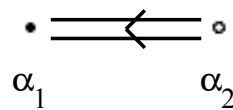
```



```

> SatakeDiagram("sp(2,2)");

```



### ▼ $so(4,3)$

We load the structure constants for Lie algebra  $so(4,3)$  from the standard Maple library.

```
> LD1 := SimpleLieAlgebraData("so(4,3)", so43) :
> DGsetup(LD1) :
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so43) ;
"so(4,3)" (1.1.4.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so43) :
> B := evalDG([e3, e2, e1, e4, e5, e6, e8, e7,
e9, e10, e11, e12, e13, e14, e15, e16, e17,
e18, e19, e21, e20]) :
> LD2 := LieAlgebraData(B, so43n) :
```

```
> DGsetup(LD2) :
> ClassifyRealSemiSimpleLieAlgebra(so43n) ;
      "so(4, 3)" (1.1.4.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

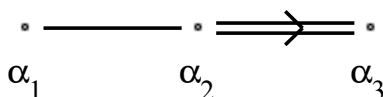
```
> ChangeFrame(so43n) :
> CSA := CartanSubalgebra(so43) ;
      CSA := [e1, e5, e9] (1.1.4.3)
```

```
> M := StandardRepresentation(so43) :
> T, P := CartanDecomposition(M, so43) :
      T;
      P;
[e2 - e4, e3 - e7, e6 - e8, e10 + e13, e11 + e14, e12 + e15,
 e16 + e19, e17 + e20, e18 + e21]
[e1, e2 + e4, e3 + e7, e5, e6 + e8, e9, e10 - e13, e11 - e14, e12 - e15,
 e16 - e19, e17 - e20, e18 - e21] (1.1.4.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
      "compact");
      NewCSA, V := [e6 - e8, e12 + e15, e16 + e19], [ ] (1.1.4.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra(so43, evalDG
      (NewCSA));
      "so(4, 3)" (1.1.4.6)
```

```
> SatakeDiagram("so(4, 3)");
```



▼  $so(5, 0) \cong sp(4)$

We load the structure constants for Lie algebra  $so(5, 0)$  from the standard Maple library. There is an isomorphism between  $so(5, 0)$  and  $sp(4)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("so(5, 0)", so50) :
> DGsetup(LD1) :
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so50);
      ["so(5)", "sp(4)"] (1.1.5.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so50):
> B := evalDG([e3, e2, e1, e4, e5, e6, e8, e7,
              e9, e10]):
> LD2 := LieAlgebraData(B, so50n):
> DGsetup(LD2):
> ClassifyRealSemiSimpleLieAlgebra(so50n);
      ["so(5)", "sp(4)"] (1.1.5.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(so50n):
> CSA := CartanSubalgebra(so50);
      CSA := [e1, e8] (1.1.5.3)
```

```
> M := StandardRepresentation(so50):
> T, P := CartanDecomposition(M, so50):
      T;
      P;
      [e1, e2, e3, e4, e5, e6, e7, e8, e9, e10]
      [ ] (1.1.5.4)
```

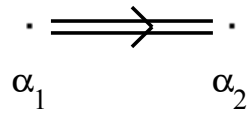
```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
      "compact");
      NewCSA, V := [e1, e8], [ ] (1.1.5.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra(so50, evalDG
      (NewCSA));
      ["so(5)", "sp(4)"] (1.1.5.6)
```

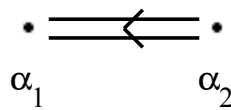
Here are the Satake diagrams that illustrate the isomorphisms between  $so(5,0)$ , and  $sp(4)$ .

```
> SatakeDiagram("so(5)");
```





```
> SatakeDiagram("sp(4)");
```



▼ *so(5,2)*

We load the structure constants for Lie algebra  $so(5,2)$  from the standard Maple library.

```
> LD1 := SimpleLieAlgebraData("so(5,2)", so52);
> DGsetup(LD1);
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so52);
"so(5,2)"
```

(1.1.6.1)

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame (so52) :
> B := evalDG([e3, e2, e1, e4, e5, e6, e8, e7,
e9, e10, e11, e12, e13, e14, e15, e16, e17,
e18, e19, e21, e20]) :
> LD2 := LieAlgebraData (B, so52n) :
> DGsetup (LD2) :
> ClassifyRealSemiSimpleLieAlgebra (so52n) ;
"so(5, 2)" (1.1.6.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

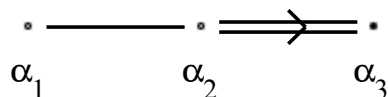
```
> ChangeFrame (so52n) :
> CSA := CartanSubalgebra (so52) ;
CSA := [e1, e4, e19] (1.1.6.3)
```

```
> M := StandardRepresentation (so52) :
> T, P := CartanDecomposition (M, so52) :
T;
P;
[e2 - e3, e5 + e6, e7 + e13, e8 + e14, e9 + e15, e10 + e16, e11
+ e17, e12 + e18, e19, e20, e21]
[e1, e2 + e3, e4, e5 - e6, e7 - e13, e8 - e14, e9 - e15, e10
- e16, e11 - e17, e12 - e18] (1.1.6.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
"compact") ;
NewCSA, V := [e2 - e3 + e5 + e6, e9 + e15, e19], [ ] (1.1.6.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra (so52, evalDG
(NewCSA)) ;
"so(5, 2)" (1.1.6.6)
```

```
> SatakeDiagram ("so (5, 2) ") ;
```



## so(2L+ 1)

$so(3) \cong su(2) \cong sp(2) \cong su^*(2)$

We load the structure constants for Lie algebra  $so(3)$  from the standard Maple library. There is an isomorphism between  $su^*(2)$ ,  $su(2)$ ,  $so(3)$ , and  $sp(2)$  so we have these two Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("so(3)", so3);
> DGsetup(LD1):
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so3);
["su(2)", "so(3)", "sp(2)", "su*(2)"] (1.2.1.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so3):
> B := evalDG([e3, e2, e1]):
> LD2 := LieAlgebraData(B, so3n):
> DGsetup(LD2):
> ClassifyRealSemiSimpleLieAlgebra(so3n);
["su(2)", "so(3)", "sp(2)", "su*(2)"] (1.2.1.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(so3n):
> CSA := CartanSubalgebra(so3);
CSA := [e1] (1.2.1.3)
```

```
> M := StandardRepresentation(so3):
> T, P := CartanDecomposition(M, so3):
T;
P;
[e1, e2, e3]
[] (1.2.1.4)
```

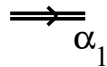
```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
NewCSA, V := [e1], [] (1.2.1.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra(so3, evalDG
```

```
(NewCSA) ;
["su(2)", "so(3)", "sp(2)", "su*(2)"] (1.2.1.6)
```

Here are the Satake diagrams that illustrate the isomorphisms between  $\mathfrak{so}(3)$ ,  $\mathfrak{su}(2)$ ,  $\mathfrak{sp}(2)$ , and  $\mathfrak{su}^*(2)$ .

```
> SatakeDiagram("so(3)");
```

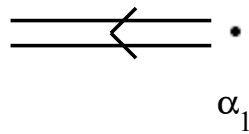


$\xrightarrow{=}$   
 $\alpha_1$

```
> SatakeDiagram("su(2)");
```

$\alpha_1$

```
> SatakeDiagram("sp(2)");
```



```
> SatakeDiagram("su*(2)");
```

$\dot{\alpha}_1$

▼  $so(5) \cong sp(4)$

We load the structure constants for Lie algebra  $so(5)$  from the standard Maple library. There is an isomorphism between  $so(5)$  and  $sp(4)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
[> LD1 := SimpleLieAlgebraData("so(5)", so5):
```

```
[> DGsetup(LD1):
```

```
[ First we check that the program works for the well known initialized Lie algebra.
```

```
> ClassifyRealSemiSimpleLieAlgebra(so5);
      ["so(5)", "sp(4)"] (1.2.2.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so5):
> B := evalDG([e2, e1, e3, e4, e6, e5, e7, e8,
              e9, e10]):
> LD2 := LieAlgebraData(B, so5n):
> DGsetup(LD2):
> ClassifyRealSemiSimpleLieAlgebra(so5n);
      ["so(5)", "sp(4)"] (1.2.2.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(so5n):
> CSA := CartanSubalgebra(so5);
              CSA := [e1, e8] (1.2.2.3)
```

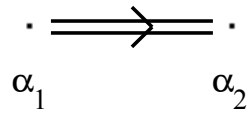
```
> M := StandardRepresentation(so5):
> T, P := CartanDecomposition(M, so5):
      T;
      P;
              [e1, e2, e3, e4, e5, e6, e7, e8, e9, e10]
              [ ] (1.2.2.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
      "compact");
              NewCSA, V := [e1, e8], [ ] (1.2.2.5)
```

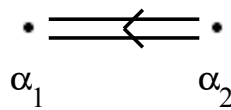
```
> ClassifyRealSemiSimpleLieAlgebra(so5, evalDG
      (NewCSA));
      ["so(5)", "sp(4)"] (1.2.2.6)
```

Here are the Satake diagrams that illustrate the isomorphisms between  $so(5)$  and  $sp(4)$ .

```
> SatakeDiagram("so(5)");
```



```
> SatakeDiagram("sp(4)");
```



▼ *so(7)*

We load the structure constants for Lie algebra  $so(5)$  from the standard Maple library.

```
> LD1 := SimpleLieAlgebraData("so(7)", so7);
> DGsetup(LD1);
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so7);
"so(7)"
```

(1.2.3.1)

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so7) :
> B := evalDG([e2, e1, e3, e4, e6, e5, e7, e8,
e9, e10, e13, e12, e11, e14, e15, e16, e17,
e18, e19, e20, e21]) :
> LD2 := LieAlgebraData(B,so7n) :
> DGsetup(LD2) :
> ClassifyRealSemiSimpleLieAlgebra(so7n) ;
"so(7)" (1.2.3.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

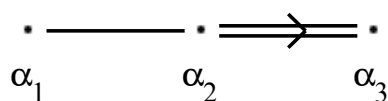
```
> ChangeFrame(so7n) :
> CSA := CartanSubalgebra(so7) ;
CSA := [e1, e12, e19] (1.2.3.3)
```

```
> M := StandardRepresentation(so7) :
> T,P := CartanDecomposition(M,so7) :
T;
P;
[e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12, e13, e14, e15,
e16, e17, e18, e19, e20, e21]
[ ] (1.2.3.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
NewCSA, V := [e1, e12, e19], [ ] (1.2.3.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra(so7, evalDG
(NewCSA)) ;
"so(7)" (1.2.3.6)
```

```
> SatakeDiagram("so(7)");
```





### 7.3 C

In this next worksheet we give the low dimensional examples of the classification program for the real simple classical Lie algebras of type C.

Each of the examples are organized as follows:

1. First we demonstrate that the classification program works for the standard library. This will give us a controlled result to compare against.
2. Next, we find a basis for an isomorphic Lie algebra. We test the program to confirm that the classification remains the same.
3. To properly classify any real simple Lie algebra using a Satake diagram, a maximally noncompact Cartan subalgebra must be used. Because of this, a third test was performed to check the functionality of all the crucial parts of the classification program. To make sure this testing occurs, I used the `MaximalCartanSubalgebra` program to find a maximally compact Cartan subalgebra for the given Lie algebra. Using this maximally compact Cartan subalgebra forces the classification program to execute a Cayley transform the maximum number of times that would be possible for each real simple Lie algebra.

## C

## sp(2L, R)

$sp(2, R) \cong sl(2) \cong so(2, 1)$

We load the structure constants for Lie algebra  $sp(2, R)$  from the standard Maple library. There is an isomorphism between  $sl(2)$ ,  $so(2, 1)$ , and  $sp(2, R)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData ("sp(2,R)", sp2R) :
> DGsetup(LD1) :
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra (sp2R) ;
["sl(2)", "so(2,1)", "sp(2,R)"] (1.1.1.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame (sp2R) :
> B := evalDG([e2, e1, e3]) :
> LD2 := LieAlgebraData (B, sp2Rn) :
> DGsetup(LD2) :
> ClassifyRealSemiSimpleLieAlgebra (sp2Rn) ;
["sl(2)", "so(2,1)", "sp(2,R)"] (1.1.1.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame (sp2Rn) :
> CSA := CartanSubalgebra (sp2R) ;
CSA := [e1] (1.1.1.3)
```

```
> M := StandardRepresentation (sp2R) :
> T, P := CartanDecomposition (M, sp2R) :
T;
P;
[e2 - e3]
[e1, e2 + e3] (1.1.1.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
"compact") ;
```

$NewCSA, V := [e_2 - e_3], [0]$  (1.1.1.5)

```
> ClassifyRealSemiSimpleLieAlgebra(sp2R, evalDG
(NewCSA));
["sl(2)", "so(2,1)", "sp(2,R)"] (1.1.1.6)
```

Here are the Satake diagrams that illustrate the isomorphisms between  $sp(2, \mathbb{R})$ ,  $sl(2)$ , and  $so(2, 1)$ .

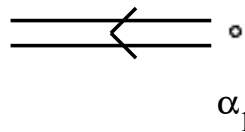
```
> SatakeDiagram("sl(2)");
```

◦  
 $\alpha_1$

```
> SatakeDiagram("so(2,1)");
```

◦  
→  
 $\alpha_1$

```
> SatakeDiagram("sp(2,R)");
```



▼  $sp(4, R) \cong so(3, 2)$

We load the structure constants for Lie algebra  $sp(4, R)$  from the standard Maple library. There is an isomorphism between  $sp(4, R)$  and  $so(3, 2)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("sp(4,R)", sp4R) :
> DGsetup(LD1) :
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(sp4R) ;
["so(3,2)", "sp(4,R)"] (1.1.2.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(sp4R) :
> B := evalDG([e2, e1, e3, e4, e6, e5, e7, e8,
```

```

e9, e10]):
> LD2 := LieAlgebraData (B, sp4Rn) :
> DGsetup(LD2) :
> ClassifyRealSemiSimpleLieAlgebra (sp4Rn) ;
    ["so(3,2)", "sp(4,R)"] (1.1.2.2)

```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```

> ChangeFrame (sp4Rn) :
> CSA := CartanSubalgebra (sp4R) ;
    CSA := [e1, e4] (1.1.2.3)

```

```

> M := StandardRepresentation (sp4R) :
> T, P := CartanDecomposition (M, sp4R) :
    T;
    P;
    [e2 - e3, e5 - e8, e6 - e9, e7 - e10]
    [e1, e2 + e3, e4, e5 + e8, e6 + e9, e7 + e10] (1.1.2.4)

```

```

> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
    "compact");
    NewCSA, V := [e5 - e8, e7 - e10], [ ] (1.1.2.5)

```

```

> ClassifyRealSemiSimpleLieAlgebra (sp4R, evalDG
    (NewCSA)) ;
    ["so(3,2)", "sp(4,R)"] (1.1.2.6)

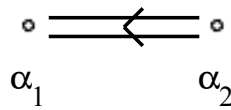
```

Here are the Satake diagrams that illustrate the isomorphism between  $\mathfrak{sp}(2, \mathbb{R})$  and  $\mathfrak{so}(3, 2)$ .

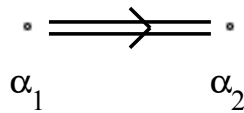
```

> SatakeDiagram ("sp (4, R) ") ;

```



```
> SatakeDiagram("so(3,2)");
```



▼  $\mathfrak{sp}(6, \mathbb{R})$

We load the structure constants for Lie algebra  $\mathfrak{sp}(6, \mathbb{R})$  from the standard Maple library.

```
> LD1 := SimpleLieAlgebraData("sp(6,R)", sp6R);
> DGsetup(LD1);
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(sp6R);
"sp(6, R)"
```

(1.1.3.1)

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame (sp6R) :
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e9,
e8, e10, e11, e12, e13, e14, e15, e16, e17,
e18, e19, e20, e21]):
> LD2 := LieAlgebraData (B, sp6Rn) :
> DGsetup (LD2) :
> ClassifyRealSemiSimpleLieAlgebra (sp6Rn) ;
      "sp(6, R)" (1.1.3.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

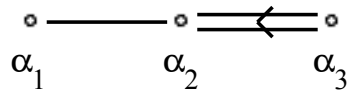
```
> ChangeFrame (sp6Rn) :
> CSA := CartanSubalgebra (sp6R) ;
      CSA := [e1, e5, e9] (1.1.3.3)
```

```
> M := StandardRepresentation (sp6R) :
> T, P := CartanDecomposition (M, sp6R) :
      T;
      P;
[e2 - e4, e3 - e7, e6 - e8, e10 - e16, e11 - e17, e12 - e18,
e13 - e19, e14 - e20, e15 - e21]
[e1, e2 + e4, e3 + e7, e5, e6 + e8, e9, e10 + e16, e11 + e17, e12(1.1.3.4)
+ e18, e13 + e19, e14 + e20, e15 + e21]
```

```
> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
"compact");
      NewCSA, V := [e10 - e16, e13 - e19, e15 - e21], [ ] (1.1.3.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra (sp6R, evalDG
(NewCSA));
      "sp(6, R)" (1.1.3.6)
```

```
> SatakeDiagram ("sp (6, R) ") ;
```



### ▼ $sp(8, R)$

We load the structure constants for Lie algebra  $sp(8, R)$  from the standard Maple library.

```
> LD1 := SimpleLieAlgebraData("sp(8,R)", sp8R) :
> DGsetup(LD1) :
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(sp8R) ;
"sp(8, R)" (1.1.4.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(sp8R) :
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e9,
e8, e10, e11, e12, e13, e14, e15, e16, e17,
e18, e19, e20, e21, e22, e23, e24, e25, e26,
e27, e28, e29, e30, e31, e32, e33, e34, e35,
```



```

e36] ) :
> LD2 := LieAlgebraData (B, sp8Rn) :
> DGsetup (LD2) :
> ClassifyRealSemiSimpleLieAlgebra (sp8Rn) ;
      "sp(8, R)" (1.1.4.2)

```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```

> ChangeFrame (sp8Rn) :
> CSA := CartanSubalgebra (sp8R) ;
      CSA := [e1, e6, e11, e16] (1.1.4.3)

```

```

> M := StandardRepresentation (sp8R) :
> T, P := CartanDecomposition (M, sp8R) :
  T;
  P;
[e2 - e5, e3 - e9, e4 - e13, e7 - e10, e8 - e14, e12 - e15, e17
 - e27, e18 - e28, e19 - e29, e20 - e30, e21 - e31, e22
 - e32, e23 - e33, e24 - e34, e25 - e35, e26 - e36]
[e1, e2 + e5, e3 + e9, e4 + e13, e6, e7 + e10, e8 + e14, e11, e12(1.1.4.4)
 + e15, e16, e17 + e27, e18 + e28, e19 + e29, e20 + e30,
 e21 + e31, e22 + e32, e23 + e33, e24 + e34, e25 + e35, e26
 + e36]

```

```

> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
  "compact") ;
NewCSA, V := [e8 - e14, e17 - e27, e21 + e26 - e31 - e36, (1.1.4.5)
  e24 - e34], [ ]

```

```

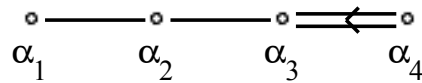
> ClassifyRealSemiSimpleLieAlgebra (sp8R, evalDG
  (NewCSA)) ;
      "sp(8, R)" (1.1.4.6)

```

```

> SatakeDiagram ("sp(8, R) ") ;

```



### ▼ $sp(p, L - p)$

#### ▼ $sp(2,2) \cong so(4,1)$

We load the structure constants for Lie algebra  $sp(2,2)$  from the standard Maple library. There is an isomorphism between  $sp(2,2)$  and  $so(4,1)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
[> LD1 := SimpleLieAlgebraData("sp(2,2)", sp22) :
[> DGsetup(LD1) :
```

First we check that the program works for the well known initialized Lie algebra.

```
[> ClassifyRealSemiSimpleLieAlgebra(sp22) ;
["so(4,1)", "sp(2,2)"] (1.2.1.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```

> ChangeFrame(sp22) :
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e9,
e8, e10]):
> LD2 := LieAlgebraData(B, sp22n) :
> DGsetup(LD2) :
> ClassifyRealSemiSimpleLieAlgebra(sp22n) ;
      ["so(4,1)", "sp(2,2)"]           (1.2.1.2)

```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```

> ChangeFrame(sp22n) :
> CSA := CartanSubalgebra(sp22) ;
      CSA := [e1, e8]                 (1.2.1.3)

```

```

> M := StandardRepresentation(sp22) :
> T, P := CartanDecomposition(M, sp22) :
T;
P;
      [e1, e4, e5, e8, e9, e10]
      [e2, e3, e6, e7]               (1.2.1.4)

```

```

> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
      NewCSA, V := [e1, e8], [ ]     (1.2.1.5)

```

```

> ClassifyRealSemiSimpleLieAlgebra(sp22, evalDG
(NewCSA)) ;
      ["so(4,1)", "sp(2,2)"]       (1.2.1.6)

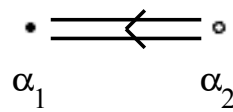
```

Here are the Satake diagrams that illustrate the isomorphism between  $sp(2,2)$  and  $so(4,1)$ .

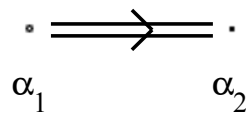
```

> SatakeDiagram("sp(2,2)");

```



```
> SatakeDiagram("so(4,1)");
```



▼ *sp(4,2)*

We load the structure constants for Lie algebra  $\mathfrak{sp}(4,2)$  from the standard Maple library.

```
> LD1 := SimpleLieAlgebraData("sp(4,2)", sp42);
> DGsetup(LD1);
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(sp42);
"sp(4,2)"
```

(1.2.2.1)

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```

> ChangeFrame (sp42) :
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e9,
e8, e10, e11, e12, e13, e14, e15, e16, e17,
e18, e19, e20, e21]):
> LD2 := LieAlgebraData (B, sp42n) :
> DGsetup (LD2) :
> ClassifyRealSemiSimpleLieAlgebra (sp42n) ;
      "sp(4, 2)" (1.2.2.2)

```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```

> ChangeFrame (sp42n) :
> CSA := CartanSubalgebra (sp42) ;
      CSA := [e1, e2 + e4, e19] (1.2.2.3)

```

```

> M := StandardRepresentation (sp42) :
> T, P := CartanDecomposition (M, sp42) :
T;
P;
[e1, e2, e3, e4, e9, e10, e11, e12, e13, e14, e19, e20, e21]
[e5, e6, e7, e8, e15, e16, e17, e18] (1.2.2.4)

```

```

> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
"compact");
      NewCSA, V := [e1, e2 + e4, e19], [ ] (1.2.2.5)

```

```

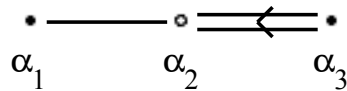
> ClassifyRealSemiSimpleLieAlgebra (sp42, evalDG
(NewCSA));
      "sp(4, 2)" (1.2.2.6)

```

```

> SatakeDiagram ("sp(4, 2)");

```



### ▼ $sp(p,p)$

#### ▼ $sp(2,2) \cong so(4,1)$

We load the structure constants for Lie algebra  $sp(2,2)$  from the standard Maple library. There is an isomorphism between  $sp(2,2)$  and  $so(4,1)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("sp(2,2)", sp22,
  'version=2');
> DGsetup(LD1);
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(sp22);
      ["so(4,1)", "sp(2,2)"]           (1.3.1.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```

> ChangeFrame(sp22) :
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e9,
e8, e10]) :
> LD2 := LieAlgebraData(B, sp22n) :
> DGsetup(LD2) :
> ClassifyRealSemiSimpleLieAlgebra(sp22n) ;
      ["so(4,1)", "sp(2,2)"]          (1.3.1.2)

```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```

> ChangeFrame(sp22n) :
> CSA := CartanSubalgebra(sp22) ;
      CSA := [e1, e8]                (1.3.1.3)

```

```

> M := StandardRepresentation(sp22) :
> T, P := CartanDecomposition(M, sp22) :
T;
P;
      [e1, e4, e5, e8, e9, e10]
      [e2, e3, e6, e7]              (1.3.1.4)

```

```

> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact") ;
      NewCSA, V := [e1, e8], [ ]    (1.3.1.5)

```

```

> ClassifyRealSemiSimpleLieAlgebra(sp22, evalDG
(NewCSA)) ;
      ["so(4,1)", "sp(2,2)"]       (1.3.1.6)

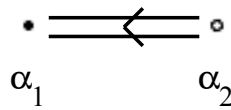
```

Here are the Satake diagrams that illustrate the isomorphism between  $sp(2,2)$  and  $so(4,1)$ .

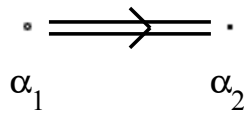
```

> SatakeDiagram("sp(2,2)");

```



```
> SatakeDiagram("so(4,1)");
```



▼  $sp(4,4)$

We load the structure constants for Lie algebra  $\mathfrak{sp}(4,4)$  from the standard Maple library.

```
> LD1 := SimpleLieAlgebraData("sp(4,4)", sp44);
> DGsetup(LD1);
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(sp44);
"sp(4,4)"
```

(1.3.2.1)



Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame (sp44) :
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e9,
e8, e10, e11, e12, e13, e14, e15, e16, e17,
e18, e19, e20, e21, e22, e23, e24, e25, e26,
e27, e28, e29, e30, e31, e32, e33, e34, e35,
e36]) :
> LD2 := LieAlgebraData (B, sp44n) :
> DGsetup (LD2) :
> ClassifyRealSemiSimpleLieAlgebra (sp44n) ;
"sp(4, 4)" (1.3.2.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

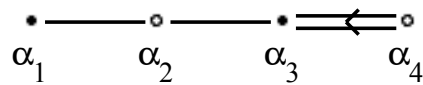
```
> ChangeFrame (sp44n) :
> CSA := CartanSubalgebra (sp44) ;
CSA := [e1, e2 + e4, e27, e28 + e30] (1.3.2.3)
```

```
> M := StandardRepresentation (sp44) :
> T, P := CartanDecomposition (M, sp44) :
T;
P;
[e1, e2, e3, e4, e13, e14, e15, e16, e17, e18, e27, e28, e29, e30,
e31, e32, e33, e34, e35, e36]
[e5, e6, e7, e8, e9, e10, e11, e12, e19, e20, e21, e22, e23, e24,
e25, e26] (1.3.2.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
"compact") ;
NewCSA, V := [e1, e2 + e4, e27, e28 + e30], [ ] (1.3.2.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra (sp44, evalDG
(NewCSA)) ;
"sp(4, 4)" (1.3.2.6)
```

```
> SatakeDiagram ("sp(4, 4) ") ;
```



### ▼ $sp(L)$

▼  $sp(2) \cong su(2) \cong so(3) \cong su^*(2)$

We load the structure constants for Lie algebra  $su^*(2)$  from the standard Maple library. There is an isomorphism between  $su^*(2)$ ,  $su(2)$ ,  $so(3)$ , and  $sp(2)$  so we have these two Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
[> LD1 := SimpleLieAlgebraData("sp(2)", sp2) :
```

```
[> DGsetup(LD1) :
```

First we check that the program works for the well known initialized Lie algebra.

```
[> ClassifyRealSemiSimpleLieAlgebra(sp2) ;
      ["su(2)", "so(3)", "sp(2)", "su*(2)"] (1.4.1.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```

> ChangeFrame (sp2) :
> B := evalDG([e2, e1, e3]):
> LD2 := LieAlgebraData (B, sp2n) :
> DGsetup (LD2) :
> ClassifyRealSemiSimpleLieAlgebra (sp2n) ;
      ["su(2)", "so(3)", "sp(2)", "su*(2)"]          (1.4.1.2)

```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```

> ChangeFrame (sp2n) :
> CSA := CartanSubalgebra (sp2) ;
      CSA := [e1]                                     (1.4.1.3)

```

```

> M := StandardRepresentation (sp2) :
> T, P := CartanDecomposition (M, sp2) :
      T;
      P;
      [e1, e2, e3]
      [ ]                                             (1.4.1.4)

```

```

> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
      "compact");
      NewCSA, V := [e1], [ ]                         (1.4.1.5)

```

```

> ClassifyRealSemiSimpleLieAlgebra (sp2, evalDG
      (NewCSA));
      ["su(2)", "so(3)", "sp(2)", "su*(2)"]          (1.4.1.6)

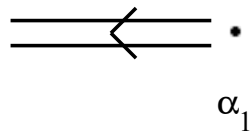
```

Here are the Satake diagrams that illustrate the isomorphisms between  $\mathfrak{sp}(2)$ ,  $\mathfrak{su}(2)$ ,  $\mathfrak{so}(3)$ , and  $\mathfrak{su}^*(2)$ .

```

> SatakeDiagram ("sp (2) ") ;

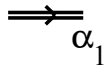
```



```
> SatakeDiagram("su(2)");
```

$\alpha_1$

```
> SatakeDiagram("so(3)");
```



```
> SatakeDiagram("su*(2)");
```

$$\dot{\alpha}_1$$

**$sp(4) \cong so(5)$**

We load the structure constants for Lie algebra  $sp(4)$  from the standard Maple library. There is an isomorphism between  $sp(4)$  and  $so(5)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("sp(4)", sp4,
    "version=2");
> DGsetup(LD1);
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(sp4);
    ["so(5)", "sp(4)"] (1.4.2.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(sp4);
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e9,
    e8, e10]);
> LD2 := LieAlgebraData(B, sp4n);
> DGsetup(LD2);
> ClassifyRealSemiSimpleLieAlgebra(sp4n);
    ["so(5)", "sp(4)"] (1.4.2.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(sp4n);
> CSA := CartanSubalgebra(sp4);
    CSA := [e1, e2 + e4] (1.4.2.3)
> M := StandardRepresentation(sp4);
> T, P := CartanDecomposition(M, sp4);
    T;
    P;
    [e1, e2, e3, e4, e5, e6, e7, e8, e9, e10]
```

```
[ ] (1.4.2.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
```

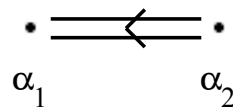
```
NewCSA, V := [e1, e2 + e4], [ ] (1.4.2.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra(sp4, evalDG
(NewCSA));
```

```
["so(5)", "sp(4)"] (1.4.2.6)
```

Here are the Satake diagrams that illustrate the isomorphism between  $\mathfrak{sp}(4)$  and  $\mathfrak{so}(5)$ .

```
> SatakeDiagram("sp(4)");
```



```
> SatakeDiagram("so(5)");
```

$$\begin{array}{ccc} \cdot & \xrightarrow{\quad} & \cdot \\ \alpha_1 & & \alpha_2 \end{array}$$

### *sp(6)*

We load the structure constants for Lie algebra  $sp(6)$  from the standard Maple library.

```
> LD1 := SimpleLieAlgebraData("sp(6)", sp6);
> DGsetup(LD1);
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(sp6);
      "sp(6)" (1.4.3.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(sp6);
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e9,
e8, e10, e11, e12, e13, e14, e15, e16, e17,
e18, e19, e20, e21]);
> LD2 := LieAlgebraData(B, sp6n);
> DGsetup(LD2);
> ClassifyRealSemiSimpleLieAlgebra(sp6n);
      "sp(6)" (1.4.3.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(sp6n);
> CSA := CartanSubalgebra(sp6);
      CSA := [e1, e4 + e7, e9] (1.4.3.3)
```

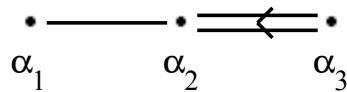
```
> M := StandardRepresentation(sp6);
> T, P := CartanDecomposition(M, sp6);
      T;
      P;
[e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12, e13, e14, e15,
e16, e17, e18, e19, e20, e21]
```

(1.4.3.4)

```

[ ] (1.4.3.4)
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
NewCSA, V := [e1, e4 + e7, e9], [ ] (1.4.3.5)
> ClassifyRealSemiSimpleLieAlgebra(sp6, evalDG
(NewCSA));
"sp(6)" (1.4.3.6)
> SatakeDiagram("sp(6)");

```



### $sp(8)$

We load the structure constants for Lie algebra  $sp(8)$  from the standard Maple library.

```

> LD1 := SimpleLieAlgebraData("sp(8)", sp8);
> DGsetup(LD1);

```

First we check that the program works for the well known initialized Lie algebra.

```

> ClassifyRealSemiSimpleLieAlgebra(sp8);

```

(1.4.4.1)



"sp(8)" (1.4.4.1)

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame (sp8) :
> B := evalDG([e2, e1, e3, e4, e5, e10, e7, e9,
e8, e6, e11, e12, e13, e14, e15, e18, e17, e16,
e19, e20, e21, e22, e23, e24, e25, e26, e27,
e28, e29, e30, e31, e32, e33, e34, e35, e36]);
> LD2 := LieAlgebraData (B, sp8n) :
> DGsetup (LD2) :
> ClassifyRealSemiSimpleLieAlgebra (sp8n) ;
"sp(8)" (1.4.4.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

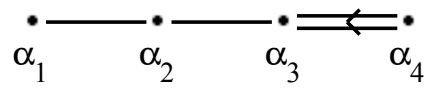
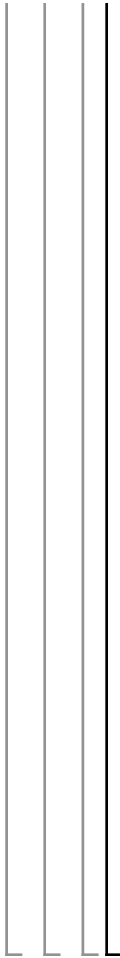
```
> ChangeFrame (sp8n) :
> CSA := CartanSubalgebra (sp8) ;
CSA := [e1, e6, e7 + e11, e14 + e16] (1.4.4.3)
```

```
> M := StandardRepresentation (sp8) :
> T, P := CartanDecomposition (M, sp8) :
T;
P;
[e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12, e13, e14, e15,
e16, e17, e18, e19, e20, e21, e22, e23, e24, e25, e26, e27,
e28, e29, e30, e31, e32, e33, e34, e35, e36]
[ ] (1.4.4.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
"compact");
NewCSA, V := [e1, e6, e7 + e11, e14 + e16], [ ] (1.4.4.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra (sp8, evalDG
(NewCSA));
"sp(8)" (1.4.4.6)
```

```
> SatakeDiagram ("sp (8) ") ;
```



## 7.4 D

In the following worksheet we give the low dimensional examples of the classification program for the real simple classical Lie algebras of type D.

Each of the examples are organized as follows:

1. First we demonstrate that the classification program works for the standard library. This will give us a controlled result to compare against.
2. Next, we find a basis for an isomorphic Lie algebra. We test the program to confirm that the classification remains the same.
3. To properly classify any real simple Lie algebra using a Satake diagram, a maximally noncompact Cartan subalgebra must be used. Because of this, a third test was performed to check the functionality of all the crucial parts of the classification program. To make sure this testing occurs, I used the `MaximalCartanSubalgebra` program to find a maximally compact Cartan subalgebra for the given Lie algebra. Using this maximally compact Cartan subalgebra forces the classification program to execute a Cayley transform the maximum number of times that would be possible for each real simple Lie algebra.

## D

## so(L,L)

so(3,3)  $\cong$  sl(4)

We load the structure constants for Lie algebra so(3,3) from the standard Maple library. There is an isomorphism between so(3,3) and sl(4), so we have all three of these Lie algebras as the output of the classifying program, ClassifyRealSemiSimpleLieAlgebras.

```
> LD1 := SimpleLieAlgebraData("so(3,3)", so33,
  'version=2'):
> DGsetup(LD1):
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so33);
      [sl(4), "so(3,3)"] (1.1.1.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so33):
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e8,
  e9, e10, e11, e12, e14, e13, e15]):
> LD2 := LieAlgebraData(B, so33n):
> DGsetup(LD2):
> ClassifyRealSemiSimpleLieAlgebra(so33n);
      [sl(4), "so(3,3)"] (1.1.1.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(so33n):
> CSA := CartanSubalgebra(so33);
      CSA := [e1, e10, e15] (1.1.1.3)
```

```
> M := StandardRepresentation(so33):
> T, P := CartanDecomposition(M, so33):
  T;
  P;
      [e1, e2, e3, e13, e14, e15]
      [e4, e5, e6, e7, e8, e9, e10, e11, e12] (1.1.1.4)
```

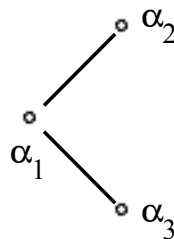
```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
```

```
"compact");
NewCSA, V := [e1, e10, e15], [ ] (1.1.1.5)
```

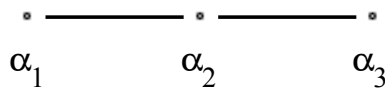
```
> ClassifyRealSemiSimpleLieAlgebra(so33, evalDG
(NewCSA));
["sl(4)", "so(3,3)"] (1.1.1.6)
```

Here are the Satake diagrams that illustrate the isomorphism between  $so(3,3)$  and  $sl(4)$ .

```
> SatakeDiagram("so(3,3)");
```



```
> SatakeDiagram("sl(4)");
```



#### so(4,4)

We load the structure constants for Lie algebra  $so(4,4)$  from the standard Maple library.

```
> LD1 := SimpleLieAlgebraData("so(4,4)", so44);
> DGsetup(LD1);
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so44);
"so(4,4)" (1.1.2.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so44);
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e8,
e9, e10, e11, e12, e14, e13, e15, e16, e17,
e18, e19, e20, e21, e22, e23, e24, e25, e26,
e27, e28]);
> LD2 := LieAlgebraData(B, so44n);
> DGsetup(LD2);
> ClassifyRealSemiSimpleLieAlgebra(so44n);
"so(4,4)" (1.1.2.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(so44n) :
> CSA := CartanSubalgebra(so44) ;
      CSA := [e1, e6, e11, e16] (1.1.2.3)
```

```
> M := StandardRepresentation(so44) :
> T, P := CartanDecomposition(M, so44) :
      T;
      P;
[e2 - e5, e3 - e9, e4 - e13, e7 - e10, e8 - e14, e12 - e15, e17
 + e23, e18 + e24, e19 + e25, e20 + e26, e21 + e27, e22
 + e28]
[e1, e2 + e5, e3 + e9, e4 + e13, e6, e7 + e10, e8 + e14, e11, e12(1.1.2.4)
 + e15, e16, e17 - e23, e18 - e24, e19 - e25, e20 - e26,
 e21 - e27, e22 - e28]
```

```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
NewCSA, V := [e3 - e9, e8 - e14, e18 + e24, e21 + e27], [ ] (1.1.2.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra(so44, evalDG
(NewCSA));
"so(4, 4)" (1.1.2.6)
```

## ▼ $so(p, 2L - p)$

### ▼ $so(5, 1) \cong su^*(4)$

We load the structure constants for Lie algebra  $so(5, 1)$  from the standard Maple library. There is an isomorphism between  $so(5, 1)$  and  $su^*(4)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("so(5, 1)", so51,
'Version=2'):
> DGsetup(LD1) :
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so51);
["so(5, 1)", "su*(4)"] (1.2.1.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```

> ChangeFrame(so51) :
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e9,
e8, e10, e11, e12, e13, e14, e15]):
> LD2 := LieAlgebraData(B,so51n) :
> DGsetup(LD2) :
> ClassifyRealSemiSimpleLieAlgebra(so51n) ;
["so(5,1)", "su*(4)"] (1.2.1.2)

```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```

> ChangeFrame(so51n) :
> CSA := CartanSubalgebra(so51) ;
CSA := [e1, e8, e15] (1.2.1.3)

```

```

> M := StandardRepresentation(so51) :
> T, P := CartanDecomposition(M,so51) :
T;
P;
[e1, e2, e3, e4, e5, e6, e7, e8, e9, e10]
[e11, e12, e13, e14, e15] (1.2.1.4)

```

```

> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
NewCSA, V := [e1, e8, e15], [ ] (1.2.1.5)

```

```

> ClassifyRealSemiSimpleLieAlgebra(so51, evalDG
(NewCSA)) ;
["so(5,1)", "su*(4)"] (1.2.1.6)

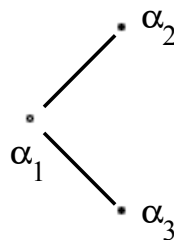
```

Here are the Satake diagrams that illustrate the isomorphism between  $\mathfrak{so}(5,1)$  and  $\mathfrak{su}^*(4)$ .

```

> SatakeDiagram("so(5,1)");

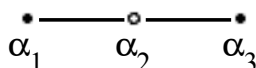
```



```

> SatakeDiagram("su*(4)");

```



$so(6,0) \cong su(4)$

We load the structure constants for Lie algebra  $so(6,0)$  from the standard Maple library. There is an isomorphism between  $so(6,0)$  and  $su(4)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("so(6,0)",so60):
```

```
> DGsetup(LD1):
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so60);
      ["su(4)", "so(6)"] (1.2.2.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so60):
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e9,
              e8, e10, e11, e12, e13, e14, e15]):
> LD2 := LieAlgebraData(B,so60n):
> DGsetup(LD2):
> ClassifyRealSemiSimpleLieAlgebra(so60n);
      ["su(4)", "so(6)"] (1.2.2.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(so60n):
> CSA := CartanSubalgebra(so60);
      CSA := [e1, e10, e15] (1.2.2.3)
```

```
> M := StandardRepresentation(so60):
> T,P := CartanDecomposition(M,so60):
      T;
      P;
      [e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12, e13, e14, e15]
```

(1.2.2.4)



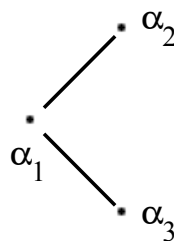
```

[ ] (1.2.2.4)
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
NewCSA, V := [e1, e10, e15], [ ] (1.2.2.5)
> ClassifyRealSemiSimpleLieAlgebra(so60, evalDG
(NewCSA));
["su(4)", "so(6)"] (1.2.2.6)

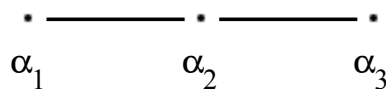
```

Here are the Satake diagrams that illustrate the isomorphism between  $so(6,0)$  and  $su(4)$ .

```
> SatakeDiagram("so(6,0)");
```



```
> SatakeDiagram("su(4)");
```



$so(6,2) \cong so^*(8)$

We load the structure constants for Lie algebra  $so(6,2)$  from the standard Maple library. There is an isomorphism between  $so(6,2)$  and  $so^*(8)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("so(6,2)", so62,
'version=2');
> DGsetup(LD1);
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so62);
["so*(8)", "so(6,2)"] (1.2.3.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so62);
> B := evalDG([e2, e1, e3, e4, e8, e6, e7, e9,
e5, e10, e11, e12, e13, e14, e15, e16, e17,
e18, e19, e20, e21, e22, e23, e24, e25, e26,
```

```

e27, e28] ):
> LD2 := LieAlgebraData (B, so62n) :
> DGsetup (LD2) :
> ClassifyRealSemiSimpleLieAlgebra (so62n) ;
      ["so*(8)", "so(6,2)"]           (1.2.3.2)

```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```

> ChangeFrame (so62n) :
> CSA := CartanSubalgebra (so62) ;
      CSA := [e1, e10, e15, e28]     (1.2.3.3)

```

```

> M := StandardRepresentation (so62) :
> T, P := CartanDecomposition (M, so62) :
T;
P;
[e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12, e13, e14, e15,
e28]
[e16, e17, e18, e19, e20, e21, e22, e23, e24, e25, e26, e27] (1.2.3.4)

```

```

> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
"compact");
      NewCSA, V := [e1, e10, e15, e28], [ ] (1.2.3.5)

```

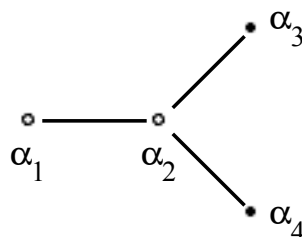
```

> ClassifyRealSemiSimpleLieAlgebra (so62, evalDG
(NewCSA) );
      ["so*(8)", "so(6,2)"]           (1.2.3.6)

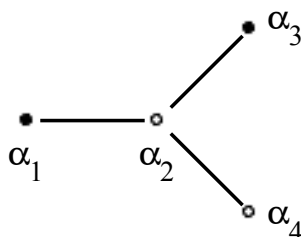
```

Here are the Satake diagrams that illustrate the isomorphism between  $\mathfrak{so}(6,2)$  and  $\mathfrak{so}^*(8)$ .

```
> SatakeDiagram ("so (6, 2) ") ;
```



```
> SatakeDiagram ("so*(8) ") ;
```



### so(L - 1, L + 1)

$so(4,2) \cong su(2,2)$

We load the structure constants for Lie algebra  $so(4,2)$  from the standard Maple library. There is an isomorphism between  $so(4,2)$  and  $su(2,2)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("so(4,2)", so42,
'version=2'):
> DGsetup(LD1):
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so42);
["so(4,2)", "su(2,2)"] (1.3.1.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so42):
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e9,
e8, e10, e11, e12, e13, e14, e15]):
> LD2 := LieAlgebraData(B, so42n):
> DGsetup(LD2):
> ClassifyRealSemiSimpleLieAlgebra(so42n);
["so(4,2)", "su(2,2)"] (1.3.1.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(so42n):
> CSA := CartanSubalgebra(so42);
CSA := [e1, e6, e15] (1.3.1.3)
> M := StandardRepresentation(so42):
```

```

> T, P := CartanDecomposition(M, so42) :
T;
P;
      [e1, e2, e3, e4, e5, e6, e15]
      [e7, e8, e9, e10, e11, e12, e13, e14]      (1.3.1.4)

```

---

```

> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
      NewCSA, V := [e1, e6, e15], [ ]      (1.3.1.5)

```

---

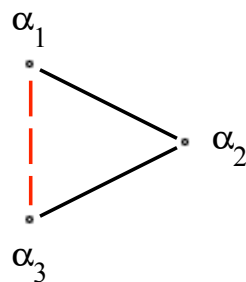
```

> ClassifyRealSemiSimpleLieAlgebra(so42, evalDG
(NewCSA));
      ["so(4,2)", "su(2,2)"]      (1.3.1.6)

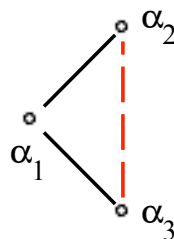
```

Here are the Satake diagrams that illustrate the isomorphism between  $so(4,2)$  and  $su(2,2)$ .

```
> SatakeDiagram("su(2,2)");
```



```
> SatakeDiagram("so(4,2)");
```



### so(5,3)

We load the structure constants for Lie algebra  $so(5,3)$  from the standard Maple library

```

> LD1 := SimpleLieAlgebraData("so(5,3)", so53) :
> DGsetup(LD1) :

```

First we check that the program works for the well known initialized Lie algebra.

```

> ClassifyRealSemiSimpleLieAlgebra(so53);
      "so(5,3)"      (1.3.2.1)

```

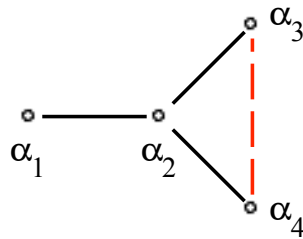
Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame (so53) :
> B := evalDG([e2, e1, e3, e5, e6, e4, e7, e8,
e9, e10, e11, e12, e13, e14, e15, e16, e17,
e18, e19, e20, e21, e22, e23, e24, e25, e26,
e27, e28]) :
> LD2 := LieAlgebraData (B, so53n) :
> DGsetup (LD2) :
> ClassifyRealSemiSimpleLieAlgebra (so53n) ;
"so(5,3)" (1.3.2.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame (so53n) :
> CSA := CartanSubalgebra (so53) ;
CSA := [e1, e5, e9, e28] (1.3.2.3)
> M := StandardRepresentation (so53) :
> T, P := CartanDecomposition (M, so53) :
T;
P;
[e2 - e4, e3 - e7, e6 - e8, e10 + e13, e11 + e14, e12 + e15,
e16 + e22, e17 + e23, e18 + e24, e19 + e25, e20 + e26, e21
+ e27, e28]
[e1, e2 + e4, e3 + e7, e5, e6 + e8, e9, e10 - e13, e11 - e14, e12(1.3.2.4)
- e15, e16 - e22, e17 - e23, e18 - e24, e19 - e25, e20
- e26, e21 - e27]
> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
"compact") ;
NewCSA, V := [e2 - e4, e9, e10 + e13, e28], [ ] (1.3.2.5)
> ClassifyRealSemiSimpleLieAlgebra (so53, evalDG
(NewCSA)) ;
"so(5,3)" (1.3.2.6)
> SatakeDiagram ("so (5, 3) ") ;
```



I came up with two potential arrows [1,3] and [3,4]. This shows that either is fine. Switching  $a_1$  and  $a_4$  changes the arrow from the above satake diagram from [a3,a4] to [a1,a3]

```
> M := Matrix(4, 4, {(1, 1) = 2, (1, 2) = 0, (1, 3) = -1, (1, 4) = 0, (2, 1) = 0, (2, 2) = 2, (2, 3) = -1, (2, 4) = 0, (3, 1) = -1, (3, 2) = -1, (3, 3) = 2, (3, 4) = -1, (4, 1) = 0, (4, 2) = 0, (4, 3) = -1, (4, 4) = 2});
```

$$M := \begin{bmatrix} 2 & 0 & -1 & 0 \\ 0 & 2 & -1 & 0 \\ -1 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \quad (1.3.2.7)$$

```
> P14 := Matrix(4, 4, {(1, 4)=1, (2, 2)=1, (3, 3)=1, (4, 1)=1});
```

$$P14 := \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (1.3.2.8)$$

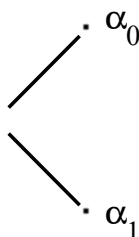
```
> P14.M.P14-M;
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (1.3.2.9)$$

## so(2L)

### so(2) Abelian

```
> LD1 := SimpleLieAlgebraData("so(2)", so2);
      LD1 := [e1, e1] = 0 \quad (1.4.1.1)
> SatakeDiagram("so(2)");
```



$so(6) \cong su(4)$

We load the structure constants for Lie algebra  $so(6)$  from the standard Maple library. There is an isomorphism between  $so(6)$  and  $su(4)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("so(6)", so6);
> DGsetup(LD1):
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so6);
      ["su(4)", "so(6)"] (1.4.2.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so6):
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e9,
              e8, e10, e11, e12, e13, e14, e15]):
> LD2 := LieAlgebraData(B, so6n):
> DGsetup(LD2):
> ClassifyRealSemiSimpleLieAlgebra(so6n);
      ["su(4)", "so(6)"] (1.4.2.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(so6n):
> CSA := CartanSubalgebra(so6);
              CSA := [e1, e10, e15] (1.4.2.3)
> M := StandardRepresentation(so6):
> T, P := CartanDecomposition(M, so6):
      T;
      P;
      [e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12, e13, e14, e15]
```

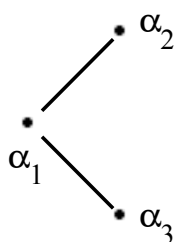
```

[ ] (1.4.2.4)
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
NewCSA, V := [e1, e10, e15], [ ] (1.4.2.5)
> ClassifyRealSemiSimpleLieAlgebra(so6, evalDG
(NewCSA));
["su(4)", "so(6)"] (1.4.2.6)

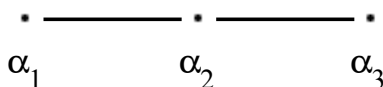
```

Here are the Satake diagrams that illustrate the isomorphism between  $\mathfrak{so}(6)$  and  $\mathfrak{su}(4)$ .

```
> SatakeDiagram("so(6)");
```



```
> SatakeDiagram("su(4)");
```



### $\mathfrak{so}(8)$

We load the structure constants for Lie algebra  $\mathfrak{so}(8)$  from the standard Maple library

```
> LD1 := SimpleLieAlgebraData("so(8)", so8);
> DGsetup(LD1);
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so8);
"so(8)" (1.4.3.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so8);
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e9,
e8, e10, e11, e12, e13, e14, e15, e16, e17,
e18, e19, e20, e21, e22, e23, e24, e25, e26,
e27, e28]);
> LD2 := LieAlgebraData(B, so8n);
> DGsetup(LD2);
```



```
> ClassifyRealSemiSimpleLieAlgebra (so8n) ;
      "so(8)" (1.4.3.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

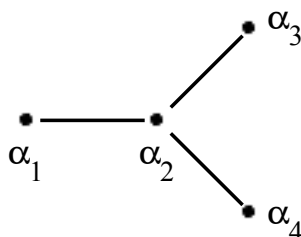
```
> ChangeFrame (so8n) :
> CSA := CartanSubalgebra (so8) ;
      CSA := [e1, e14, e23, e28] (1.4.3.3)
```

```
> M := StandardRepresentation (so8) :
> T, P := CartanDecomposition (M, so8) :
      T;
      P;
      [e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12, e13, e14, e15,
        e16, e17, e18, e19, e20, e21, e22, e23, e24, e25, e26, e27, e28]
      [ ] (1.4.3.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra (CSA, T, P,
      "compact");
      NewCSA, V := [e1, e14, e23, e28], [ ] (1.4.3.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra (so8, evalDG
      (NewCSA));
      "so(8)" (1.4.3.6)
```

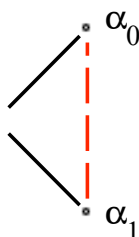
```
> SatakeDiagram ("so (8) ") ;
```



## ▼ $so^*(2L)$ where $L$ is even

### ▼ $so^*(2)$ Abelian

```
> LD1 := SimpleLieAlgebraData ("so* (2) ", so2) ;
      LD1 := [e1, e1] = 0 (1.5.1.1)
> SatakeDiagram ("so* (2) ") ;
```



$so^*(4)$

We load the structure constants for Lie algebra  $so^*(4)$  from the standard Maple library

```
> LD1 := SimpleLieAlgebraData("so*(4)", so4);
> DGsetup(LD1);
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so4);
"so*(4)" (1.5.2.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so4);
> B := evalDG([e2, e1, e3, e4, e5, e6]);
> LD2 := LieAlgebraData(B, so4n);
> DGsetup(LD2);
> ClassifyRealSemiSimpleLieAlgebra(so4n);
"so*(4)" (1.5.2.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(so4n);
> CSA := CartanSubalgebra(so4);
CSA := [e1, e2] (1.5.2.3)
```

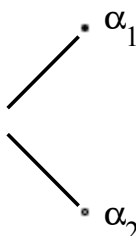
```
> M := StandardRepresentation(so4);
> T, P := CartanDecomposition(M, so4);
T;
P;
[e1, e3, e4, e5]
[e2, e6] (1.5.2.4)
```

```
> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
```

```
"compact");
NewCSA, V := [e1, e3 + e5], [ ] (1.5.2.5)
```

```
> ClassifyRealSemiSimpleLieAlgebra(so4, evalDG
(NewCSA));
"so*(4)" (1.5.2.6)
```

```
so53 > SatakeDiagram("so*(4)");
```



The following shows that using either the dot combination of [white, black] or [black, white] is the same algebra by switching  $a_1$  with  $a_2$ .

```
so62 > M := Matrix(2, 2, {(1, 1) = 2, (1, 2)
= 0, (2, 1) = 0, (2, 2) = 2});
M :=  $\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$  (1.5.2.7)
```

```
> P12 := Matrix(2, 2, {(1, 2)=1, (2, 1)=1});
P12 :=  $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$  (1.5.2.8)
```

```
> P12.M.P12-M;
 $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$  (1.5.2.9)
```

▼  $so^*(8) \cong so(6,2)$

We load the structure constants for Lie algebra  $so^*(8)$  from the standard Maple library. There is an isomorphism between  $so^*(8)$  and  $so(6,2)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("so*(8)", so8);
> DGsetup(LD1);
First we check that the program works for the well known initialized
Lie algebra.
> ClassifyRealSemiSimpleLieAlgebra(so8);
["so*(8)", "so(6,2)"] (1.5.3.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so8);
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e9,
```

```

e8, e10, e11, e12, e13, e14, e15, e16, e17,
e18, e19, e20, e21, e22, e23, e24, e25, e26,
e27, e28]):
> LD2 := LieAlgebraData(B, so8n) :
> DGsetup(LD2) :
> ClassifyRealSemiSimpleLieAlgebra(so8n) ;
      ["so*(8)", "so(6,2)"]           (1.5.3.2)

```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program MaximalCartanSubalgebra. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```

> ChangeFrame(so8n) :
> CSA := CartanSubalgebra(so8) ;
      CSA := [e1, e6, e7, e12]       (1.5.3.3)

```

```

> M := StandardRepresentation(so8) :
> T, P := CartanDecomposition(M, so8) :
T;
P;
[e1, e2, e3, e4, e5, e6, e13, e14, e15, e16, e17, e18, e19, e20, e21,
e22]
[e7, e8, e9, e10, e11, e12, e23, e24, e25, e26, e27, e28]   (1.5.3.4)

```

```

> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
      NewCSA, V := [e1, e6, e13 + e17, e20 + e22], [ ]   (1.5.3.5)

```

```

> ClassifyRealSemiSimpleLieAlgebra(so8, evalDG
(NewCSA)) ;
      ["so*(8)", "so(6,2)"]           (1.5.3.6)

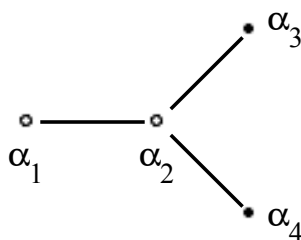
```

Here are the Satake diagrams that illustrate the isomorphism between  $\mathfrak{so}^*(8)$  and  $\mathfrak{so}(6,2)$ .

```

> SatakeDiagram("so(6,2)");

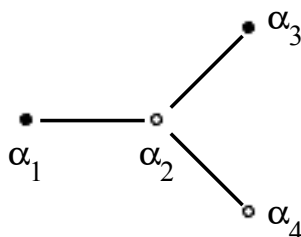
```



```

> SatakeDiagram("so*(8)");

```



### ▼ $so^*(2L)$ where $L$ is odd

▼  $so^*(6) \cong su(3,1)$

We load the structure constants for Lie algebra  $so^*(6)$  from the standard Maple library. There is an isomorphism between  $so^*(6)$  and  $su(3,1)$ , so we have all three of these Lie algebras as the output of the classifying program, `ClassifyRealSemiSimpleLieAlgebras`.

```
> LD1 := SimpleLieAlgebraData("so*(6)", so6);
> DGsetup(LD1):
```

First we check that the program works for the well known initialized Lie algebra.

```
> ClassifyRealSemiSimpleLieAlgebra(so6);
      ["su(3,1)", "so*(6)"] (1.6.1.1)
```

Now we change the basis of the Lie algebra to produce an isomorphic Lie algebra and run the program again to compare against the control.

```
> ChangeFrame(so6):
> B := evalDG([e2, e1, e3, e4, e5, e6, e7, e9,
              e8, e10, e11, e12, e13, e14, e15]):
> LD2 := LieAlgebraData(B, so6n):
> DGsetup(LD2):
> ClassifyRealSemiSimpleLieAlgebra(so6n);
      ["su(3,1)", "so*(6)"] (1.6.1.2)
```

The output matches the control, as expected.

We find a maximally compact Cartan subalgebra for this Lie algebra using my program `MaximalCartanSubalgebra`. For our classification using a Satake diagram, we need a maximally noncompact Cartan subalgebra. By giving the classification program this maximally compact Cartan subalgebra, the program is forced to complete the Cayley transform process.

```
> ChangeFrame(so6n):
> CSA := CartanSubalgebra(so6);
      CSA := [e1, e4, e12] (1.6.1.3)
> M := StandardRepresentation(so6):
> T,P := CartanDecomposition(M, so6):
```

```

T;
P;
[e1, e2, e3, e7, e8, e9, e10, e11, e12]
[e4, e5, e6, e13, e14, e15] (1.6.1.4)

```

```

> NewCSA, V := MaximalCartanSubalgebra(CSA, T, P,
"compact");
NewCSA, V := [e1, e7 + e10, e12], [ ] (1.6.1.5)

```

```

> ClassifyRealSemiSimpleLieAlgebra(so6, evalDG
(NewCSA));
["su(3,1)", "so*(6)"] (1.6.1.6)

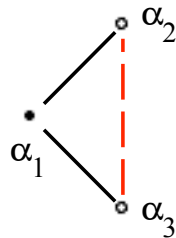
```

Here are the Satake diagrams that illustrate the isomorphism between  $\mathfrak{so}^*(6)$  and  $\mathfrak{su}(3,1)$ .

```

> SatakeDiagram("so*(6)");

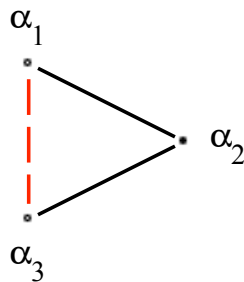
```



```

> SatakeDiagram("su(3,1)");

```



## Computer Programs

---

```

RootSort:=proc(proots, T, P, RSD)
description ' This program checks the elements of each root to determine if the
root is real, complex or imaginary. A list of the real and a list of imaginary
roots.
The complex roots are discarded. ' ;
    local dim, Relist, Imlist, Reindex, Imindex, testvector, Rebool, Imbool,
        ind, dims, Colist, Coindex, tvector, ImList;

#Relist is list of the real roots.
#Imlist is a list of the imaginary roots.

dim := ArrayNumElems(proots[1]);
Relist := Array(1 .. nops(proots));
Imlist := Array(1 .. nops(proots));
Reindex := 0;
Imindex := 0;

for testvector in roots do
    ind := 1;
    Rebool := false;
    Imbool := false;
    while (ind <= dim) and not(Rebool and Imbool) do
        if testvector[ind] > 0 then
            if evalb(Re(testvector[ind])=testvector[ind]) then
                Rebool := true;
            else
                Imbool := true;
            fi;
        fi;
        ind:=ind + 1;
    od;
    if Rebool and not Imbool then
        Reindex := Reindex + 1;
        Relist[Reindex] := testvector;
    elif not Rebool and Imbool then
        Imindex := Imindex + 1;
        Imlist[Imindex] := testvector;
    fi;
end do;
end proc;

```

```

        fi;
    od;

    ImList := convert(Imlist[1 .. Imindex],list);

    dims := nops(ImList);
    Colist := Array(1 .. dims);
    Coindex := 0;

    for tvector in ImList do
        if nops(GetComponents(RSD[convert(tvector, list)], T)) = 0
        and not nops(GetComponents(RSD[convert(tvector, list)], P)) = 0
        then
            Coindex := Coindex + 1;
            Colist[Coindex] := tvector;
        fi;
    od;
    return convert(Relist[1 .. Reindex], list), convert(Imlist[1 .. Imindex], list);
end proc;

```

---

```

NewCSAImRoot:=proc(CSA,T0,{root:=[]})
description ' Using a noncompact imaginary root, this program completes
the Cayley transform and returns a new Cartan subalgebra which has its
intersection with the p part of the Cartan subalgebra up by one ' ;
    local RSD, PR, Re, Im, beta, Ebeta, Fbeta, Hbeta, NewCSAelement,
    NS, NewCSA, V, EQ, Sol, dim, CSAlist, CSAindex, i, C, ind, T, P,
    Theta, Ebeta1, Fbeta1, Hbeta1, x, y, c, ImNonCompact;

    if T0 :: list then
        T := T0;
        P := args[3];
        Theta := CartanInvolution(T,P);
    else Theta := T0;
        T, P := CartanDecomposition(Theta);
    fi;
    RSD := RootSpaceDecomposition(CSA);

    PR := PositiveRoots(RSD);

    if root <> [] then
        beta := root;
    else
        Re,Im := RootSort(PR, T, P, RSD);

        if Im=[] then
            return CSA,[];
        fi;
    fi;

```



```

ImNonCompact := Array[];
userinfo(3,ClassifyRealSemiSimpleLieAlgebra,'NoName',"1.41 Finding a
noncompact root ");
i := 1;
  for x in Im do
    y := RSD[convert(x, list)];
      if P = [] then
        C := [];
      else
        C := GetComponents(y, P);
      fi;
      if C <> [] then
#this means that y is contained in P so y is noncompact
        ImNonCompact[i] := x;

        i := i + 1;
      fi;
    od;
  fi;
if nops(ImNonCompact) = 0 then
  return CSA, [];
fi;
userinfo(3,ClassifyRealSemiSimpleLieAlgebra,'NoName'," 1.42 Using the
root to complete the Cayley transform ");
#Here we are scaling E_beta and the conjugate of E_beta to satisfy the
structure equations for sl(2)

beta := ImNonCompact[1];
Ebeta1 := evalDG((lambda) *RSD[convert(beta, list)]);
Hbeta1 := evalDG(RootToCartanSubalgebraElementH(beta, RSD));
Fbeta1 := evalDG((lambda)*RSD[convert(-beta, list)]);
V := evalDG(LieBracket(Ebeta1, Fbeta1) - evalDG(Hbeta1));
EQ := DGinformation(evalDG(V), "CoefficientSet");
Sol := solve(EQ, lambda, explicit = true);
Ebeta := subs(Sol[1], Ebeta1);
Fbeta := subs(Sol[1], Fbeta1);
Hbeta := subs(Sol[1], Hbeta1);
#CSAlist is a list of the elements in the new Cartan subalgebra that are
coming from the given Cartan subalgebra.

if
  DGequal( LieBracket(Hbeta, Ebeta), evalDG(2 * Ebeta) ) and
  DGequal( LieBracket(Hbeta, Fbeta), evalDG(-2 * Fbeta) ) and
  LieBracket(Ebeta, Fbeta) - evalDG(Hbeta) = 0 then
  # matching page 333 knapp
  NewCSAelement := evalDG(Ebeta &plus; DGconjugate(Ebeta));
  NS := LinearAlgebra :- NullSpace(beta^+);

```

```

        dim := 1 + nops(NS);
        CSAList := Array(1 .. dim);
        CSAindex := 0;

    for i from 1 to nops(NS) do
        CSAindex := CSAindex + 1;
        CSAList[CSAindex] := DGzip(NS[i], CSA, "plus");
    od;

    CSAindex := CSAindex + 1;
    CSAList[CSAindex] := NewCSAelement;

return CanonicalBasis(convert(CSAList[1 .. CSAindex], list)), beta;

else return "error";
fi:

end proc:

```

---

```

NewCSARealRoot:=proc(CSA,T0, P0, {root:=[]})
description ' Using a real root, this program completes the Cayley transform
and returns a new Cartan subalgebra which has its intersection with the t
part of the Cartan subalgebra up by one ' ;
    local RSD, PR, Re, Im, alpha, Ealpha, Falpha, Halpha, NewCSAelement,
    NS, NewCSA, V, EQ, Sol, dim, CSAList, CSAindex, i, ind, T, P, Theta,
    Ealpha1, Falpha1, Halpha1, RSD1, PR1, Re1, Im1;

if T0 :: list then
    T := T0;
    P := P0;
    Theta := CartanInvolution(T, P);

    else Theta := T0;

        T,P := CartanDecomposition(Theta);
        fi;

RSD := RootSpaceDecomposition(CSA);

PR := PositiveRoots(RSD);

if root = [] then
    Re, Im := RootSort(PR, T, P, RSD);

    if Re = [] then

```

```

        return CSA, [], [];
    fi;
    alpha := Re[1];
else
    alpha := root;
fi;
#Here we are scaling E_beta and the conjugate of E_beta to satisfy
the structure equations below
Ealpha1 := evalDG((lambda) * (RSD[convert(alpha, list)]));

Halpha1 := evalDG(RootToCartanSubalgebraElementH(alpha, RSD));

Falpha1 := evalDG((ApplyLinearTransformation(Theta, Ealpha1)));
V := evalDG(LieBracket(Ealpha1, Falpha1) + Halpha1);
EQ := DGinformation(evalDG(V), "CoefficientSet");
Sol := solve(EQ, lambda, explicit = true);
Ealpha := evalDG(subs(Sol[1], Ealpha1));
Falpha := evalDG(subs(Sol[1], Falpha1));
Halpha := evalDG(subs(Sol[1], Halpha1));

if
    DGequal( LieBracket(Halpha, Ealpha), evalDG(2 * Ealpha) ) and
    DGequal( LieBracket(Halpha, Falpha), evalDG(-2 * Falpha) ) and
    LieBracket(Ealpha, Falpha) - evalDG(-Halpha) = 0 then
        NewCSAelement := evalDG(Ealpha &plus Falpha);

        NS := LinearAlgebra :- NullSpace(alpha^+);
        dim := 1 + nops(NS);
        CSAlist := Array(1 .. dim);
        CSAindex := 0;

        for i from 1 to nops(NS) do
            CSAindex := CSAindex + 1;
            CSAlist[CSAindex] := DGzip(NS[i], CSA, "plus");
        od;

        CSAindex := CSAindex + 1;
        CSAlist[CSAindex] := NewCSAelement;

NewCSA := CanonicalBasis(convert(CSAlist[1 .. CSAindex], list));

RSD1 := RootSpaceDecomposition(NewCSA);

PR1 := PositiveRoots(RSD1);

Re1, Im1 := RootSort(PR1, T, P, RSD1);

return NewCSA , alpha, Re1;

```

```

else return [-1],[],[];
fi:

```

```

end proc:

```

---

```

MaximalCartanSubalgebra := proc(CSA,TO,PO,kind,{output:=
"CartanSubalgebra",root:=[]})
description ' This program runs the NewCSARealRoot program
or the NewCSAImRoot program as many times as is necessary
to achieve either a maximally noncompact Cartan subalgebra or
a maximally compact Cartan subalgebra respectively ' ;
    local dim, boolcom, CompactIndex, MaxCompactCSA,
    MaxRootArray, NONCOMPACTCSA, ROOT, REALLIST ;
    if not type(kind, string) then
        error "Fourth argument must be a string"
    fi;
    dim := nops(CSA);
    MaxCompactCSA := Array(1 .. dim);
    MaxRootArray:= Array(1 .. dim);
    CompactIndex := 1;
    boolcom := True;
    NONCOMPACTCSA := CSA;
    ROOT := root;
    while boolcom = True and CompactIndex <= dim do;
        if _params['PO'] = NULL then
            if kind[1] = "c" or kind[1] = "C" then
                NONCOMPACTCSA, ROOT, REALLIST :=
                NewCSARealRoot(NONCOMPACTCSA, TO,
                PO, root = ROOT);
                while REALLIST <> [] do;
                    NONCOMPACTCSA, ROOT, REALLIST :=
                    NewCSARealRoot(NONCOMPACTCSA,
                    TO, PO, root = ROOT);
                od;
                if ROOT = [] then
                    return NONCOMPACTCSA, ROOT;
                fi;
            elif kind[1] = "n" or kind[1] = "N" then
                NONCOMPACTCSA, ROOT :=
                NewCSAImRoot(NONCOMPACTCSA, TO, root = ROOT);
                if ROOT = [] then
                    return NONCOMPACTCSA, ROOT;
                fi;
            else
                error "Kind must be either compact or noncompact";
            fi;
        else
            if kind[1]="c" or kind[1]="C" then

```

```

NONCOMPACTCSA, ROOT, REALLIST :=
NewCSARealRoot(NONCOMPACTCSA, TO, PO);
    while REALLIST <> [] do;
        NONCOMPACTCSA,
        ROOT, REALLIST :=
        NewCSARealRoot(NONCOMPACTCSA,
            TO, PO);
    od;
    if ROOT = [] then
return NONCOMPACTCSA, ROOT;
        fi;

    elif kind[1]="n" or kind[1]="N" then
        NONCOMPACTCSA, ROOT :=
        NewCSAImRoot(NONCOMPACTCSA, TO, PO);
        if ROOT = [] then
            return NONCOMPACTCSA, ROOT;
        fi;
    else
        error "Kind must be either compact or noncompact";
    fi;
fi;

if ROOT = [] then
    boolcom := false;
else
    MaxCompactCSA(CompactIndex) := NONCOMPACTCSA;
    MaxRootArray(CompactIndex) := ROOT;

    ROOT:=[];
    CompactIndex := CompactIndex +1;
fi;

od;

if output="All" then
    return MaxCompactCSA(1 .. CompactIndex - 1) ,
    MaxRootArray(1 .. CompactIndex - 1);
else
    return MaxCompactCSA(CompactIndex - 1), Array(1..1);
fi:

end proc:

```

---



---

```

CalculateSatakeDataForRealSimpleLieAlgebra := proc(LA,CSAO, RSDO, PRO, SRO)
description 'This program uses the maximally noncompact Cartan subalgebra to find

```

the root space decomposition and then the simple roots for this Cartan subalgebra. Then the colors white and black are assigned to the real/complex and imaginary roots respectively.‘;

```

local CSAN, RSD, PR, T, P, CSA1, SR, CM, CMS, SRM, colorArray, indx,
testvector, ind, Rebool, x, y, newcsa, SRM1, x1, y1, CSAlist, i;
if nargs>1 then
    CSAN := CSA0;
else
    CSAN := CartanSubalgebra(LA);
fi;
if nargs > 2 then
    RSD := RSD0;
else

userinfo(2, ClassifyRealSemiSimpleLieAlgebra, 'NoName', "1.1 Calculating root
space decomposition for the initial cartan subalgebra");
    RSD := RootSpaceDecomposition(CSAN);
fi;
userinfo(2, ClassifyRealSemiSimpleLieAlgebra, 'NoName', "1.2 Calculating positive
and simple roots for the initial cartan subalgebra");

    if nargs > 3 then
        PR := PRO;
    else
        PR := PositiveRoots(RSD);
    fi;
    if nargs > 4 then
        SR := SRO;
    else
        SR := SimpleRoots(PR);
    fi;

userinfo(2, ClassifyRealSemiSimpleLieAlgebra, 'NoName', "1.3 Calculating cartan
decomposition for the initial cartan subalgebra");

    T, P := CartanDecomposition(CSAN, RSD, PR);
userinfo(2, ClassifyRealSemiSimpleLieAlgebra, 'NoName', "1.4 Calculating new
cartan subalgebra");

    x1 := IntersectSubspaces([CSAN, P]);
    y1 := IntersectSubspaces([CSAN, T]);

    CSA1 := MaximalCartanSubalgebra(CSAN, T, P, "noncompact");
    x := IntersectSubspaces([CSA1[1], P]);
    y := IntersectSubspaces([CSA1[1], T]);
    CSAlist := array[1 .. nops(CSAN)];
    CSAlist[1] := CSA1[1];

```

```

i := 2;

while nops(x1) <> nops(x) do;
    x1 := IntersectSubspaces([CSAlist[i - 1], P]);
    y1 := IntersectSubspaces([CSAlist[i - 1], T]);
    CSAlist[i] := MaximalCartanSubalgebra(CSAlist[i - 1],
        T, P, "noncompact")[1];
    x := IntersectSubspaces([CSAlist[i], P]);
    y := IntersectSubspaces([CSAlist[i], T]);
    i := i + 1;
od;
newcsa := [op(x), op(y)];
userinfo(2, ClassifyRealSemiSimpleLieAlgebra, 'NoName', "1.5 Calculating root
space decomposition for the new Cartan subalgebra");

RSD := RootSpaceDecomposition(newcsa);

PR := PositiveRoots(RSD);
SR := SimpleRoots(PR);
CM := CartanMatrix(SR, RSD);

CMS := CartanMatrixToStandardForm(CM);
SRM := convert(SR, Matrix);
userinfo(2, ClassifyRealSemiSimpleLieAlgebra, 'NoName', "1.6 Assigning the
colors white or black to the corresponding simple roots ");

SRM1 := (SRM.CMS[2]);
SR := [LinearAlgebra :- Column(SRM1, 1 .. nops(SR))];
colorArray := [seq("black", k = 1 .. nops(SR))];
indx := 0;
for indx from 1 to nops(SR) do
    testvector := SR[indx];
    ind := 1;
    Rebool := false;
    while (ind <= nops(SR)) and not(Rebool) do
        if testvector[ind] <> 0 then
            if evalb(Re(testvector[ind]) = testvector[ind])
            then
                Rebool := true;
                colorArray[indx] := "white";
# Color the dot white in the case of a nonimmigrant root. This means the root
is either complex or real.
                fi:
            fi:
            ind := ind + 1;
        od:
    od:
od:

```

```

        return colorArray, SR, CMS[3][1], nops(newcsa);
end proc:

```

---

```

SatakeArrows:= proc(CRS1,CRS2)
description 'Given the dot colors and simple roots, find the Satake associates
of a real simple Lie algebra';
#CRS1 is a list of the simple root dot colors [white , , black].
#CRS2 is a list of the simple roots.
    local i, H, A, SA, lst, k, dim;
    dim := ceil((nops(CRS1)/2)) + 1;
#SA will be a list [ [a,b] ] where a and b are the Satake associates.
    SA := Array(1 .. dim);
    i := 1;
    if dim = 2 then
        lst := [];
        return lst;
    fi;
    while (i <= dim) do;
        if CRS1[i] = "white" then;
            H := SatakeAssociate(CRS2[i], CRS2);
            A := GetComponents(H, CRS2);
            member(1, A, 'k');
            if i < k then;
                SA[i] := [i, k];
            fi;
        fi;
        i:=i + 1;
    od;
    lst := [];
    for k from 1 to dim do
        if is(SA[k] <> 0) then
            lst := [op(lst), SA[k]];
        fi;
    od;
    return lst;
    return SA(1 .. i);
end proc:

```

---

```

PossibleSemiSimpleLieAlgebraGivenComplexClassAndDimCartan := module()
    local ModuleApply, ClassifyRealSemiSimpleLieAlgebraProc,
        SatakeDataProc;
    ModuleApply := proc(ComplexClass, DimCartan)
description 'gives a list of the possible real simple Lie algebras given the
complex class and the dimension of the Cartan subalgebra ';
local algType, alg, ind, p, q, i, alist, arraysize;

#alist is a list of the potential Lie algebras.
convert(ComplexClass, string);

```



```

SatakeDataProc[ComplexClass](DimCartan);
end;

SatakeDataProc["A"] := proc(DimCartan)
    local algType, alg, ind, p, q, i, alist, arraysize, Dim;
    Dim := DimCartan + 1;
    arraysize := trunc(Dim / 2) + 3;
    alist := Array(1 .. arraysize);
    alist[1] := cat("sl(", Dim, ")");
    alist[2] := cat("su(", Dim, ")");
    alist[3] := cat("su*(", Dim, ")");

    p := Dim;
    q := 1;
    for i from 1 to Dim do
        ind := 4;
        while (p > q + 1) do
            p := Dim - ind + 3;
            q := ind - 3;

            alist[ind] := cat("su(", p, ", ", q, ")");
            ind := ind + 1;
        od;

        od:

        alist;
    end;

SatakeDataProc["B"] := proc(DimCartan)
    local algType, alg, ind, p, q, i, alist, arraysize, Dim;
    Dim := 2 * DimCartan + 1;
    arraysize := ceil(Dim / 2);
    alist := Array(1 .. arraysize);
    alist[1] := cat("so(", Dim, ")");

    p := Dim;
    q := 1;
    for i from 1 to Dim do
        ind := 2;
        while (p > q + 2) do
            p := Dim - ind + 1;
            q := ind - 1;

            alist[ind] := cat("so(", p, ", ", q, ")");
            ind := ind + 1;
        od;

        od:

        alist;
    end;

SatakeDataProc["C"] := proc(DimCartan)

```

```

local algType, alg, ind, p, q, i, alist, arraysize, Dim, blist;
Dim := DimCartan * 2;
arraysize := trunc(Dim / 2) + 2;
alist := Array(1 .. arraysize);
alist[1] := cat("sp(", Dim, ", R", ")");
alist[2] := cat("sp(", Dim, ")");

p := Dim;
q := 1;
for i from 1 to Dim do
  ind := 3;

  while (p > q + 1) do
    p := Dim - ind + 2;
    q := ind - 2;
    if type(p, even) then
      alist[ind] := cat("sp(", p, ", ", " ", q, ")");
    fi;

    ind := ind + 1;
  od;

  od:
  blist := Array(1 .. ArrayNumElems(alist, NonZero));
  ind := 1;
  for i from 1 to numelems(alist) do
    if alist[i] <> 0 then
      blist[ind] := alist[i];
      ind := ind + 1;
    fi;
  od;
  blist;

end;

SatakeDataProc["D"] := proc(DimCartan)
  local algType, alg, ind, p, q, i, alist, arraysize, Dim;
  Dim := 2*DimCartan ;

  if Dim = 8 then
    arraysize := trunc(Dim / 2) + 3;
    alist := Array(1 .. arraysize);
    alist[1] := cat("so(", Dim, ")");
    alist[2] := cat("so*(", Dim, ")");
  fi;
end;

```

```

alist[3] := cat("hannah(", Dim, ")");

p := Dim;
q := 1;
  for i from 1 to Dim do
    ind := 4;
      while (p > q + 1) do
        p := Dim - ind + 3;
        q := ind - 3;

        alist[ind] := cat("so(", p, ", ", ", q, ")");
        ind := ind + 1;
      od;

    od;

elif type(Dim, even) then
arraysize := trunc(Dim / 2) + 3;
alist := Array(1 .. arraysize);
alist[1] := cat("so(", Dim, ")");
alist[2] := cat("so*(", Dim, ")");
alist[3] := cat("so(", Dim - 1, ", ", ", Dim - 1, ")");

p := Dim;
q := 1;
  for i from 1 to Dim do
    ind := 4;
      while (p > q + 1) do
        p := Dim - ind + 3;
        q := ind - 3;

        alist[ind] := cat("so(", p, ", ", ", q, ")");
        ind := ind + 1;
      od;

    od;

else;
arraysize := trunc(Dim / 2) + 2;
alist := Array(1 .. arraysize);
alist[1] := cat("so(", Dim, ")");
alist[2] := cat("so(", Dim - 1, ", ", ", Dim - 1, ")");
  p := Dim;
  q := 1;
  for i from 1 to Dim - 1 do
    ind := 3;
      while (p > q + 1) do

```

```

        p := Dim - ind + 2;
        q := ind - 2;

        alist[ind] := cat("so(", p, ", ", ", q, ")");
        ind := ind + 1;
    od;

    od:

    fi;

    return alist;
end;

end module:

```

---

```

SatakeInfo := module()
local ModuleApply;

ModuleApply := proc(alg)
description 'give the color of the dots and the Satake associates of a real simple
Lie algebra';
#option trace;
local a, b, c, n, m, algType, A, B, ans, ColorsSRTypeDimCartan, X, Arrow, Dot, d,
    i, ind, Arr;
if alg = 0 then
    return alg
fi;
#This program pulls all the important pieces from a given Lie algebra and then
uses that to find the right program to give the dots and arrows for the given
Lie algebra.

m := StringTools:-FirstFromLeft("(", alg);
algType := alg[1 .. m-1];
a := StringTools:-FirstFromLeft("(", alg);
b := StringTools:-FirstFromLeft(",", alg);
c := StringTools:-FirstFromLeft(")", alg);

if b = 0 then
    A := StringTools:-SubString(alg, a + 1 .. c - 1);
    n := sscanf(A, "%d")[1];
else
    A := StringTools:-SubString(alg, a + 1 .. b - 1);
    B := StringTools:-SubString(alg, b + 1 .. c - 1);
    n := sscanf(A, "%d")[1];
    m := sscanf(B, "%d");
    if m = [] then
        m := sscanf(B, "%s");
    fi;
fi;

```

```

    if m = [] then
        error("invalid first argument");
    fi;
    m := m[1];
fi;

#####
#      A sl(n), sl(n, R)
#####
if algType = "sl" then
    if b = 0 then
        Dot := sln["sln"](n)[1];
        Arrow := sln["sln"](n)[2];

    else

        if m = "R" then
            Dot := sln["sln"](n)[1];
            Arrow := sln["sln"](n)[2];

        fi;

    fi;

#####
#      A su(n), su(p, q), su(n, *)
#####
elif algType = "su" then
    if b = 0 then
        Dot := sup["sun", 1](n)[1];
        Arrow := sup["sun", 1](n)[2];

    else

        if m::integer then
            if n - m > 2 then
                Dot := supq["supq", 1](n, m)[1];
                Arrow := supq["supq", 1](n, m)[2];

            elif n - m = 2 then
                Dot := supq["supq", 1](n, m)[1];
                Arrow := supq["supq", 1](n, m)[2];
                #ans := supq["supq", 2](n, m)

            elif n - m = 1 then
                Dot := supq["supq", 1](n, m)[1];
                Arrow := supq["supq", 1](n, m)[2];
                #ans := supq["supq", 3](n, m)

            elif n - m = 0 then
                Dot := supq["supq", 1](n, m)[1];
                Arrow := supq["supq", 1](n, m)[2];
                #ans := supq["supq", 4](n, m)

            fi;

        fi;

fi;

```

```

fi;
#####
#      A su*(n)
#####
elif algType = "su*" and n:: even then
  if not n :: even then
    error("expected n to be an even integer for algebra
          type \"su*(n)\". Received %1", n)
  fi;
  Dot := slm["spn", 1](n)[1];
  Arrow := slm["spn", 1](n)[2];
  #ans := slm["su*n"](n)
#####
#      B,D so(n, R), so(p, q)
#####
elif algType = "so" then
  if b = 0 then
    if n :: odd then
      Dot := sop["sopqB", 1](n, m)[1];
      Arrow := sop["sopqB", 1](n, m)[2];
      #ans := sop["sopqB", 1](n);
    else
      Dot := so2l["sopqD", 1](n, m)[1];
      Arrow := so2l["sopqD", 1](n, m)[2];
      #ans := so2l["sopqD", 1](n);
    fi;
  else
    if m :: integer then
      if (n + m) :: odd then
        if m = 0 then
          Dot := sop["sopqB", 1](n)[1];
          Arrow := sop["sopqB", 1](n)[2];
          #ans := sop["sopqB", 1](n)
        else
          Dot := sopq["sopqB", 1](n, m)[1];
          Arrow := sopq["sopqB", 1](n, m)[2];
          #ans := sopq["sopqB", 2](n, m)
        fi;
      elif n - m = 2 then
        Dot := soleo["sopqD", 1](n, m)[1];
        Arrow := soleo["sopqD", 1](n, m)[2];
        #ans := soleo["sopqD", 3](n, m)
      elif n = m then
        Dot := soll["sopqD", 1](n, m)[1];
        Arrow := soll["sopqD", 1](n, m)[2];
        #ans := soll["sopqD", 4](n)
      else

```

```

Dot := sopq2["sopqD", 1](n, m)[1];
Arrow := sopq2["sopqD", 1](n, m)[2];
#ans := sopq2["sopqD", 2](n, m)

fi;
elif m = "R" then
  if n :: odd then
    #ans := SDiagram["sopqB", 1](n);
  else
    #ans := SDiagram["sopqD", 1](n);
  fi;
fi;

fi;

#####
#      D so*(n)
#####
elif algType = "so*" then
  if not n :: even then
    error("expected n to be an even integer for algebra
          type \"so*(n)\". Received %1", n)
  fi;
  if (n/2) :: even then
    Dot := so21e["spn", 1](n)[1];
    Arrow := so21e["spn", 1](n)[2];
    #ans := so21e["so*n", 1](n)
  else
    Dot := so21o["spn", 1](n)[1];
    Arrow := so21o["spn", 1](n)[2];
    #ans := so21o["so*n", 2](n)
  fi;

#####
#      hannah(8) triality isomorphism
#####
elif algType = "hannah" then

  if n = 8 then
    Dot := ["black","white","white","black"];
    Arrow := [];

  fi;

#####
#      C sp(n), sp(n, R), sp(p, q)
#####
elif algType = "sp" then
  if b = 0 then
    if not n :: even then
      error("expected n to be an even integer for algebra

```

```

        type \"sp(n)\". Received %1\", n)
    fi;
    Dot := spl[\"spn\", 1](n)[1];
    Arrow := spl[\"spn\", 1](n)[2];
    #ans := spl[\"spn\"](n);
else
    if m::integer then
        if not n :: even then
            error(\"expected n to be an even integer for
                the algebra type \"sp(n, m)\". Received %1\", n)
        elif not m :: even then
            error(\"expected n to be an even integer for
                the algebra type \"sp(n, m)\". Received %1\", n)
        fi;
        if m = 0 then
            Dot := spl[\"spn\", 1](n)[1];
            Arrow := spl[\"spn\", 1](n)[2];
            #ans := spl[\"spn\"](n);
        elif m = n then
            Dot := sPPP[\"sppq\", 1](n)[1];
            Arrow := sPPP[\"sppq\", 1](n)[2];
            #ans := sPPP[\"sppq\", 2](n);
        else
            Dot := sppq[\"sppq\", 1](n, m)[1];
            Arrow := sppq[\"sppq\", 1](n, m)[2];
            #ans := sppq[\"sppq\", 1](n, m);
        fi;
    elif m = \"R\" then
        if not n :: even then
            error(\"expected n to be an even integer for
                algebra type \"sp(n, R)\". Received %1\", n)
        fi;
        Dot := spp[\"spnR\", 1](n, m)[1];
        Arrow := spp[\"spnR\", 1](n, m)[2];
        #ans := spp[\"spnR\"](n);
    fi;
fi;

Dot, Arrow;
end:
end:

```

---

```

ClassifierUserInfo := proc(m,n, message)
    local tab0, tab;
    tab0 := \" \";
    if n=0 then

```



```

        tab := "";
    else
        tab := cat(tab0$n);
    fi;

end:

ClassifyRealSemiSimpleLieAlgebra := proc(alg,CSA0, RSD0)
    local DotsSRTypeDimCartan, PossibleAlg, MyArrows, MyDots,A,
        Dots, Arrows,finalanswer;

    userinfo(1,ClassifyRealSemiSimpleLieAlgebra,'NoName',"1. Calculating the
        Satake Data");
        DotsSRTypeDimCartan :=
            CalculateSatakeDataForRealSimpleLieAlgebra(args);

    userinfo(1,ClassifyRealSemiSimpleLieAlgebra,'NoName',"2. Calculating the
        Possible Algebras given the complex class and the dimension of the Cartan
        subalgebra");
        PossibleAlg := PossibleSemiSimpleLieAlgebraGivenComplexClass-
            AndDimCartan(DotsSRTypeDimCartan[3], DotsSRTypeDimCartan[4]);

    userinfo(1,ClassifyRealSemiSimpleLieAlgebra,'NoName',"3. Calculating the given
        algebras arrows");
        MyArrows := SatakeArrows(DotsSRTypeDimCartan);
        MyDots := DotsSRTypeDimCartan[1];

    userinfo(1,ClassifyRealSemiSimpleLieAlgebra,'NoName',"4. Matching the
        given algebras dots and arrows to the list of possible algebras");
    for A in PossibleAlg do;
        Dots,Arrows := SatakeInfo(A);
        if MyDots = Dots and MyArrows = Arrows then
            finalanswer := A;
            if finalanswer = "sl(2)" or finalanswer = "so(2,1)" or
                finalanswer = "sp(2,R)" then
                finalanswer := ["sl(2)", "so(2,1)", "sp(2,R)"];
            fi;
            if finalanswer = "sl(4)" or finalanswer = "so(3,3)" then
                finalanswer := ["sl(4)", "so(3,3)"];
            fi;
            if finalanswer = "su(2)" or finalanswer = "so(3)" or
                finalanswer = "sp(2)" or finalanswer = "su*(2)" then
                finalanswer := ["su(2)", "so(3)", "sp(2)", "su*(2)"];
            fi;
            if finalanswer = "su(3, 1)" or finalanswer = "so*(6,0)" or
                finalanswer = "so*(6)" then

```

```

        finalanswer := ["su(3,1)", "so*(6)"];
    fi;
    if finalanswer = "su(4)" or finalanswer = "so(6, 0)" or
    finalanswer = "so(6)" then
        finalanswer := ["su(4)", "so(6)"];
    fi;
    if finalanswer = "so*(6)" or finalanswer = "su(3, 1)" then
        finalanswer := ["so*(6)", "su(3,1)"];
    fi;
    if finalanswer = "so*(8)" or finalanswer = "so(6, 2)" or
    finalanswer = "hannah(8)" then
        finalanswer := ["so*(8)", "so(6,2)"];
    fi;
    if finalanswer = "so(3, 2)" or finalanswer = "sp(4, R)" then
        finalanswer := ["so(3,2)", "sp(4,R)"];
    fi;
    if finalanswer = "so(4, 1)" or finalanswer = "sp(2, 2)" then
        finalanswer := ["so(4,1)", "sp(2,2)"];
    fi;
    if finalanswer = "so(5)" or finalanswer = "sp(4)" then
        finalanswer := ["so(5)", "sp(4)"];
    fi;
    if finalanswer = "so(5, 1)" or finalanswer = "su*(4)" then
        finalanswer := ["so(5,1)", "su*(4)"];
    fi;
    if finalanswer = "so(4, 2)" or finalanswer = "su(2, 2)" then
        finalanswer := ["so(4,2)", "su(2,2)"];
    fi;

    return finalanswer;
fi;
od;

if DotsSRTypeDimCartan[4] = 4 and DotsSRTypeDimCartan[3] = "D" and MyDots =
["white", "white", "white", "white"] then
    finalanswer := "so(5,3)";
fi;

if DotsSRTypeDimCartan[4] = 2 and DotsSRTypeDimCartan[3] = "A" and MyDots =
["white", "black"] or MyDots = ["black", "white"] then
    finalanswer := "so*(4)";

fi;

return finalanswer;
end:


---




---


sln := proc(n)

```

```

local rank, BlackWhite, i, SatakeAssociates;
description 'This gives the Satake diagram data for  $sl(1 + 1, R)$ ';
#see Cap Slovak page 612 entry 1

rank := n - 1;
BlackWhite := Array(1 .. rank);
for i from 1 to rank do
    BlackWhite[i] := "white"
od;

BlackWhite := convert(BlackWhite, list);
BlackWhite, [];

end:

```

---

```

sup := proc(p)
    local rank, BlackWhite, i, SatakeAssociates;
    description 'This gives the Satake diagram data for  $su(1 + 1)$ ';
    #see Cap Slovak page 612 entry 5
    rank := p - 1;
    BlackWhite := Array(1 .. rank);
    for i from 1 to p - 1 do
        BlackWhite[i] := "black"
    od;

    BlackWhite := convert(BlackWhite, list);

    BlackWhite, [];

end:

```

---

```

supq := proc(q, p)
    local rank, BlackWhite, i, SatakeAssociates;
    description 'This gives the Satake diagram data for  $sl(p, 1 + 1 - p)$ ';
    #see Cap Slovak page 612 entry 3
    rank := p + q - 1;
    BlackWhite := Array(1 .. rank);
    for i from 1 to p do
        BlackWhite[i] := "white"
    od;
    for i from p + 1 to rank - p do
        BlackWhite[i] := "black"
    od;
    for i from rank - p + 1 to rank do
        BlackWhite[i] := "white"
    od;
    BlackWhite := convert(BlackWhite, list);
    SatakeAssociates := [seq([i, q + p - i], i = 1 .. p)];

```

```
BlackWhite,SatakeAssociates;
```

```
end:
```

---

```
supp := proc(p)
  local rank, BlackWhite, i, SatakeAssociates;
  description 'This gives the Satake diagram data for su(p,p)';
  #see Cap Slovak page 612 entry 4

  rank := 2 * p - 1;

  BlackWhite := Array(1 .. rank);
  for i from 1 to rank do
    BlackWhite[i] := "white"
  od;

  BlackWhite := convert(BlackWhite, list);
  SatakeAssociates := [seq([i, 2 * p - i], i = 1 .. p - 1)];
  BlackWhite,SatakeAssociates;
```

```
end:
```

---

```
slm := proc(p)
  local rank, BlackWhite, i, SatakeAssociates,m;
  description 'This gives the Satake diagram data for sl(m,H)';
  #see Cap Slovak page 612 entry 2
  m := p/2;
  rank := 2*m - 1;
  BlackWhite := Array(1 .. rank);
  for i from 1 to rank do
    if type(i,'odd') then
      BlackWhite[i] := "black"
    else
      BlackWhite[i] := "white"
    fi;
  od;
  BlackWhite := convert(BlackWhite, list);
  BlackWhite,[];
```

```
end:
```

---

```
sop := proc(p)
  local rank, BlackWhite, i, SatakeAssociates;
  description 'This gives the Satake diagram data for so(2l + 1)';
  #see Cap Slovak page 612 entry 7

  rank := (p - 1)/2;
```

```

BlackWhite := Array(1 .. rank);
for i from 1 to rank do
    BlackWhite[i] := "black"
od;

BlackWhite := convert(BlackWhite, list);

BlackWhite,[];

end:

```

---

```

so2l := proc(p)
    local rank, BlackWhite, i, SatakeAssociates;
    description 'This gives the Satake diagram data for  $sl(1 - 1, 1 + 1)$ ';
    #see Cap Slovak page 613 entry 3

    rank := (p)/2;
    BlackWhite := Array(1 .. rank);
    for i from 1 to rank do
        BlackWhite[i] := "black"
    od;

    BlackWhite := convert(BlackWhite, list);
    BlackWhite,[];

end:

```

---

```

sop := proc(p)
    local rank, BlackWhite, i, SatakeAssociates;
    description 'This gives the Satake diagram data for  $so(2l + 1)$ ';
    #see Cap Slovak page 612 entry 7

    rank := (p - 1)/2;

    BlackWhite := Array(1 .. rank);
    for i from 1 to rank do
        BlackWhite[i] := "black"
    od;

    BlackWhite := convert(BlackWhite, list);

    BlackWhite,[];

end:

```

---

```

sopq := proc(q, p)
    local rank, BlackWhite, i, SatakeAssociates;

```

```

description 'This gives the Satake diagram data for  $sl(p, 2l + 1 - p)$ ';
#see Cap Slovak page 612 entry 6
rank := (q + p - 1)/2;
BlackWhite := Array(1 .. rank);
for i from 1 to p do
    BlackWhite[i] := "white"
od;
for i from p + 1 to rank do
    BlackWhite[i] := "black"
od;

BlackWhite := convert(BlackWhite, list);

BlackWhite, [];

end:

```

---

```

soleo := proc(q,p)
    local rank, BlackWhite, i, SatakeAssociates;
    description 'This gives the Satake diagram data for  $so(l - 1, l + 1)$ ';
    #see Cap Slovak page 613 entry 3
    if p+q=8 then
        rank := p + 1;

        BlackWhite := Array(1 .. rank);

        for i from 1 to rank do
            BlackWhite[i] := "white"
        od;

        BlackWhite := convert(BlackWhite, list);
        SatakeAssociates := [[3, 4], [1, 4], [1, 3]];

    else
        rank := p + 1;

        BlackWhite := Array(1 .. rank);
        for i from 1 to rank do
            BlackWhite[i] := "white"
        od;

        BlackWhite := convert(BlackWhite, list);
        SatakeAssociates := [rank - 1, rank ];
    fi;
    BlackWhite, SatakeAssociates;
end:

```

---

---

```

soll := proc(l)
    local rank, BlackWhite, i, SatakeAssociates;
    description 'This gives the Satake diagram data for sl(1,1)';
    #see Cap Slovak page 613 entry 1

    rank := 1 ;
    BlackWhite := Array(1 .. rank);
    for i from 1 to rank do

        BlackWhite[i] := "white"
    od;

    BlackWhite := convert(BlackWhite, list);
    BlackWhite, [];

end:

```

---

```

sopq2 := proc(q, p)
    local rank, BlackWhite, i, SatakeAssociates;
    description 'This gives the Satake diagram data for so(p,2l - p)';
    #see Cap Slovak page 613 entry 2

    rank := (q + p)/2;
    BlackWhite := Array(1 .. rank);
    for i from 1 to p do
        BlackWhite[i] := "white"
    od;
    for i from p+1 to rank do

        BlackWhite[i] := "black"
    od;

    BlackWhite := convert(BlackWhite, list);
    BlackWhite, [];

end:

```

---

```

so2le := proc(p)
    local rank, BlackWhite, i, SatakeAssociates;
    description 'This gives the Satake diagram data for so*(2l), l is even';
    #see Cap Slovak page 613 entry 5

    rank := p/2 ;
    BlackWhite := Array(1 .. rank);
    for i from 1 to rank do
        if type(i,odd) then

```

```

        BlackWhite[i] := "black"
      else
        BlackWhite[i] := "white"
      fi;
    od;

    BlackWhite := convert(BlackWhite, list);
    BlackWhite,[];

end:

```

---

```

so2lo := proc(p)
  local rank, BlackWhite, i, SatakeAssociates;
  description 'This gives the Satake diagram data for so*(2l), l is odd';
  #see Cap Slovak page 613 entry 6

  rank := p/2;

  BlackWhite := Array(1 .. rank);
  for i from 1 to rank - 2 do
    if type(i,odd) then
      BlackWhite[i] := "black"
    else
      BlackWhite[i] := "white"
    fi;
  od;
  for i from rank -1 to rank do
    BlackWhite[i] := "white"
  od;

  BlackWhite := convert(BlackWhite, list);
  SatakeAssociates := [rank - 1, rank ];
  BlackWhite,SatakeAssociates;

end:

```

---

```

spl := proc(l)
  local rank, BlackWhite, i, SatakeAssociates;
  description 'This gives the Satake diagram data for so(l,l)';
  #see Cap Slovak page 612 entry 11

  rank := l/2 ;
  BlackWhite := Array(1 .. rank);
  for i from 1 to rank do

    BlackWhite[i] := "black"

```



```

od;

BlackWhite := convert(BlackWhite, list);
BlackWhite, [];

end:

```

---

```

sppp := proc(p)
  local rank, BlackWhite, i, SatakeAssociates;
  description 'This gives the Satake diagram data for sp(p,p)';
  #see Cap Slovak page 612 entry 10

  rank := p ;
  BlackWhite := Array(1 .. rank);
  for i from 1 to p do
    if type(i,odd) then
      BlackWhite[i] := "black"
    else
      BlackWhite[i] := "white"
    fi;
  od;

  BlackWhite := convert(BlackWhite, list);
  BlackWhite, [];

end:

```

---

```

sppq := proc(q, p)
  local rank, BlackWhite, i, SatakeAssociates, q1, p1;
  description 'This gives the Satake diagram data for sl(p,1 - p)';
  #see Cap Slovak page 612 entry 9
  q1 := q/2;
  p1 := p/2;
  rank := p1 + q1 ;
  BlackWhite := Array(1 .. rank);
  for i from 1 to 2*p1 do
    if type(i,odd) then
      BlackWhite[i] := "black"
    else
      BlackWhite[i] := "white"
    fi;
  od;
  for i from 2*p1 + 1 to rank do
    BlackWhite[i] := "black"
  od;

  BlackWhite := convert(BlackWhite, list);

```

```
BlackWhite, [];  
  
end:  


---



---

  
spp := proc(p)  
  local rank, BlackWhite, i, SatakeAssociates;  
  description 'This gives the Satake diagram data for sp(21,R)';  
  #see Cap Slovak page 612 entry 8  
  
  rank := p/2;  
  
  BlackWhite := Array(1 .. rank);  
  for i from 1 to rank do  
    BlackWhite[i] := "white"  
  od;  
  
  BlackWhite := convert(BlackWhite, list);  
  
  BlackWhite, [];  
  
end:  


---


```