PROBLEMS ON THE GUAGE THEORY OF WEAK, ELECTROMAGNETIC AND STRONG INTERACTIONS

Eleftherios G. Papantonopoulos

A Thesis Submitted for the Degree of PhD
at the
University of St Andrews

1980

PROBLEMS ON THE GAUGE THEORY OF WEAK, ELECTROMAGNETIC AND STRONG INTERACTIONS

A thesis presented by

Eleftherios G Papantonopoulos

to the University of St Andrews

in application for the degree of

Doctor of Philosophy

August 1980

PROBLEMS ON THE GAUGE THEORY OF WEAK, ELECTROMAGNETIC AND STRONG INTERACTIONS

by Eleftherios G Papantonopoulos

ABSTRACT

The aim of this thesis is to present and discuss some mathematical and physical problems in the theory of weak, electromagnetic and strong interactions. Our main concern is a parallel development of mathematical and physical concepts and when it is possible, an attempt to bridge the abstract mathematical formulations with physical ideas.

A central role in this thesis is played by a general construction scheme, which enables us to calculate explicitly all the mathematical quantities like matrix elements, Clebsch-Gordan series, Clebsch-Gordan coefficients which are necessary for a Grand Unification model construction. In this content, we have followed two basic principles: simplicity and applicability. To meet the first principle, all the construction methods developed are based on first principles and basic concepts of the Lie algebras and its representation theory, like roots and weights. Moreover, the requirement of applicability is met with the implementation of all the algorithms into computer programs.

In the physical area, we have concentrated on the problem of mass. The lepton mass spectrum us studied in a theory of weak and electromagnetic interactions, while the mass problem of the SO(10) Grand Unified theory is analysed as a direct application of our Lie group construction scheme.

STATEMENT

The accompanying thesis is my own composition. It is based on work carried out by me and no part of it has previously been presented in application for a higher degree.

CERTIFICATE

I certify that the conditions of the Ordinance and Regulations have been fulfilled.

Professor J F Cornwell

Research Supervisor

ACKNOWLEDGEMENTS

I should like to thank Professor J F Cornwell, my research supervisor, for his constant interest and encouragement, and for many stimulating conversations we had in the course of my study in the University of St Andrews.

I thank Professor R B Dingle for the facilities made available to me in the Department of Theoretical Physics.

I thank Dr A Cant for his helpful comments on many aspects of this work.

I should also like to thank the other members of the department for their interest, and the Computing Laboratory of the University of St Andrews for the computing facilities made available to me, and specially Dr J T Henderson.

I am grateful to Miss Margaret Smith for her expert typing.

CONTENTS

CHAPTER 1

INTRODUCTION

§1.1 General Remarks

A gauge theory of weak, electromagnetic and strong interactions is described by a Yang-Mills Lagrangian, which provides the structure upon which the whole physical theory is constructed. The Yang-Mills theory has an underlying Lie group structure upon which the mathematical theory of weak, electromagnetic and strong interactions is based.

In the early stages of the development, when weak and electromagnetic interactions were unified under a single gauge group, the Lie group structure was provided by a simple or semi-simple Lie group of rank two or three. To study the properties of these groups, methods have been developed in order to generate explicit matrix elements and to calculate Clebsch-Gordan coefficients. When strong interactions were introduced into the theory, it became obvious that these methods have to be generalized to meet the needs of a Grand Unified theory with high rank groups.

At the same time, the group theoretical structure of the theory became more important, because it was recognized that it is not only a convenient mathematical tool which can be used to fit in physical data, but it was proved to be a theoretical structure with the power to make predictions and with the ability to create new physics. One example in these lines is the Grand Unification prediction of an unstable proton.

In a parallel development of physical concepts, it was accepted that strong interactions are better described with a Yang-Mills theory, and quantum chromodynamics is the field theory of strong interactions. The problem of mass, the gauge hierarchies, the flavour problem are now better understood.

The aim of this thesis is to present and discuss some mathematical and physical problems in the theory of weak, electromagnetic and strong interactions. Our main concern is a parallel development of mathematical and physical concepts and when it is possible, an attempt to bridge the abstract mathematical formulations with physical ideas.

§1.2 The Structure of the Thesis

A central role in this thesis is played by a general construction scheme, which enables us to calculate explicitly all the mathematical quantities like matrix elements, Clebsch-Gordan series, Clebsch-Gordan coefficients which are necessary for a Grand Unification model construction. In this content, we have followed two basic principles: simplicity and applicability. To meet the first principle, all the construction methods developed are based on first principles and basic concepts of the Lie algebras and its representation theory, like roots and weights. Moreover, the requirement of applicability is met with the implementation of all the algorithms into computer programs.

In the physical area, we have concentrated on the problem of mass. The lepton mass spectrum is studied in a theory of weak and electromagnetic interactions, while the mass problem of the SO(10) Grand Unified theory is analysed as a direct application of our Lie group construction scheme.

In more detail, the thesis is structured as follows. In Chapter 2, the electron-muon problem is studied in connection with models of weak and electromagnetic interactions. The main emphasis is given to physical concepts, while Appendix A gives the mathematical details. The electron-muon problem provides a very good example of the limitations of the theory of weak and electromagnetic interactions and at the same time gives us an indication that possibly an enlargement of the symmetry group with a richer structure could provide an explanation of the mass problem.

In Chapter 3 we introduce the Grand Unification theories. In Part I, the physical ideas are discussed, while Part II deals with the mathematical problems of the Grand Unified theories, and at the same time provides the natural introduction to Chapter 4.

Chapter 4 is a detailed exploration of the theory of Lie algebras and its representation theory. Old algorithms are revised and new methods are developed in order to calculate root systems, weight systems, Clebsch-Gordan series, matrix elements and Clebsch-Gordan coefficients. In Appendix C we give the programs which implement the above algorithms.

Chapter 5 gives a mathematical construction of the SO(10) theory, and matrix elements and Clebsch-Gordan coefficients of the SO(10) theory are evaluated.

Chapter 6 is a generalization of the methods developed in Chapters 4 and 5. We discuss the SO(2n) groups with n = 5,7,9,..., and the exceptional groups E_6, E_7 and E_8.

Chapter 7 examines the mass problem in the SO(10) theory, and it serves as a direct application of the mathematical formalism we have developed in a specific physical problem.

Finally, in the last chapter we discuss the possible extentions of our work.

CHAPTER 2

THE ELECTRON-MUON PROBLEM

The lepton mass spectrum is one of the longstanding mysteries of theoretical physics. The size of the electron-muon mass ratio $m_e/m_\mu \sim O(a)$, where a is the fine structure constant, suggests that the electron mass is entirely electromagnetic of origin. The recently discovered new lepton, the tau (τ) [1 , 2], imposed new problems on the lepton mass spectrum, because of its large mass about 1782 MeV/c^2, 17 times the muon mass.

The first attempt to gain some understanding on the lepton mass spectrum was made in the context of quantum electrodynamics. It was based on the work of Nambu and Jona-Lasinio [3 , 4], and Goldstone [5]. The idea was that a non-linear Lagrangian may possess solutions which lack the symmetries of the original Lagrangian. Thus, if a Lagrangian with no bare lepton mass is invariant under some symmetries, the breaking of these symmetries will generate in a dynamical way, non-zero lepton masses.

Baker and Glashow [6] proposed a model based on a non-linear Lagrangian with lepton bare mass zero and an interaction term invariant under isotopic rotations and γ_5 transformations. The breaking of these symmetries led them to a system of coupled non-linear integral equations. The solution of this system, with the available techniques at that time, failed to give the right value of the electron-muon ratio proportional to a . Thus it was believed that quantum electrodynamics cannot determine this ratio.

The advent of renormalizable models of weak and electromagnetic interactions, in gauge theories, gave a new possibility of explaining the old problem. If the mass of the electron is managed to be kept zero at first order of perturbation, through some symmetries in the Yukawa interaction, then the electron might get its mass in higher orders, as a result of radiative corrections and because the theory is renormalizable, these corrections will be finite.

This mechanism of generating lepton masses and subsequently quark masses was first proposed by Weinberg [7], and applied to the electron-muon problem by Georgi and Glashow [8]. Since then, various attempts have been made to explain the electron-muon problem based on the unified theory of weak and electromagnetic interactions [9 ,10 ,11]. As we shall see in this chapter, the calculability of the ratio was achieved in principle, but at the expense of an unrealistic model of weak and electromagnetic interactions.

The advances in gauge field theories led Vinciarelli [12] and Goldman et al [13] to reconsider the Baker and Glashow solution in the electrodynamics. They assumed that the electron-muon mass difference is a non-perturbative effect as the calculation in the electrodynamics revealed, and they tried to eliminate the dependence of the solutions on the cut-off parameter introduced in [6]. But the large mass parameters of the gauge bosons is the price to be paid for a calculable electron-muon mass ratio.

The introduction of the grand unified theories linked the lepton mass spectrum with the general problem of fermion mass and the hierarchy problem (Chapter 3). In the grand unified theories, because quarks and leptons belong to the same irreducible representations, we have relations connecting the mass spectrum of quarks to the masses of leptons.

In this chapter we shall review the Georgi and Glashow work [8] on the calculability of the electron-muon mass ratio, and we shall present an attempt at calculating this ratio using a model based on the orthogonal group SO(5).

§2.1 Calculability of the Electron-Muon Mass Ratio in the SU(3) ⊗ U(1) Gauge Model of Weak and Electromagnetic Interactions

In a theory with spontaneously broken gauge symmetry, the general zeroth-order fermion mass matrix is a sum of a bare-mass term and a term coming from the Yukawa coupling proportional to the zeroth-order vacuum expectation values of the spinless meson fields. A zeroth-order mass relation is a relation among the masses in the zeroth-order mass matrix which is left unchanged by arbitrary (but small) changes in the renormalized parameters. We have four types of zeroth-order mass relations or natural symmetries as they are known [8]:

(a) mass relations determined by an unbroken subgroup of the symmetry of the Lagrangian;

(b) mass relations determined by the representation content of the spinless meson multiplet;

(c) mass relations involving accidental symmetry; and

(d) mass relations which arise due to the constraints imposed on the Lagrangian by the requirement of renormalizability.

Type (a) is the exact mass relation associated with an unbroken symmetry. Such mass relations are maintained in higher orders. The vanishing of the neutrino mass might be a type (a) mass relation.

Type (b) mass relation can occur when the Yukawa couplings are incomplete. If the fermions are transforming according to a representation $\underset{\sim}{\Gamma}$ of the symmetry group, a spinless meson multiplet can couple to fermions if it transforms according to any irreducible component of $\underset{\sim}{\Gamma} \otimes \underset{\sim}{\Gamma}^*$. If there are no spinless mesons in these irreducible representations then we do not have the most general zeroth-order mass relations.

Type (c) mass relations can result when the most general renormalizable Yukawa couplings and couplings of the spinless meson fields among themselves have a larger invariance group than the full Lagrangian. These types of mass relations have been discussed by S Weinberg [7] and Coleman and E Weinberg [14].

Finally, type (d) mass relations can occur because the Lagrangian is required to be renormalizable, so that only quadratic, cubic, and quartic couplings of the spinless meson fields are allowed.

The strategy for calculating the electron-muon mass ratio is to choose a gauge model of weak and electromagnetic interactions in which it is possible to have a zeroth-order mass relation. Then, assuming that the electron bare-mass is zero, to calculate the contributions from the radiative corrections of higher orders in perturbation theory from the diagram of Figure 2.1.

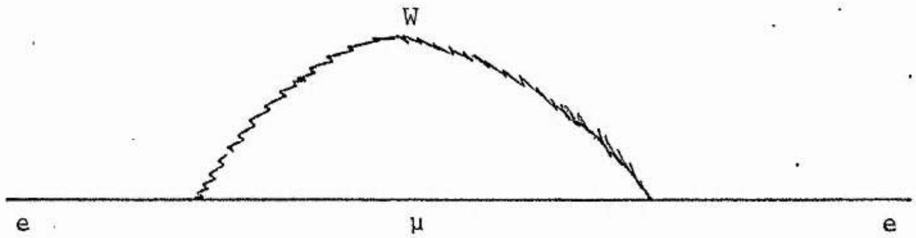


Figure 2.1: Feynman diagram which could lead to an electron mass of order am_μ

It is not always possible to find a model with zeroth-order mass relations as the example of the SU(3) model shows.

Example I: SU(3) model

Let F_j, $j = 1,\ldots,8$, be the generators of SU(3) such that

$$[F_j,F_k] = f_{jkm}F_m \tag{2.1.1}$$

where the structure contants f_{jkm} are as given by Gell-Mann [15].

Then the generators T_j, $j = 1,\ldots,8$, of reference [8] are given by

$$F_j = \frac{1}{2}T_j,\ j = 1,\ldots,8. \tag{2.1.2}$$

Then the leptonic electric charge is given by

$$Q^{\ell} = \frac{1}{2}(T_3+\sqrt{3}T_8). \tag{2.1.3}$$

Because the electron and muon must be included in the same irreducible representation, a possible choice is the Konopinski-Mahmaund triplet

$$\ell = \begin{bmatrix} \mu^+ \\ \nu \\ e^- \end{bmatrix}. \tag{2.1.4}$$

Let

$$\underline{\ell}_R = \frac{1}{2}(1-\gamma_5)\underline{\ell}, \tag{2.1.5}$$

for the right handed helicity of the lepton fields, and suppose that

$$F_p\ell_{R_j} = \sum_{k=1}^{3}\Gamma(F_p)_{k_j}\ell_{R_k} \tag{2.1.6}$$

where $\underset{\sim}{\Gamma}$ is the three-dimensional irreducible representation of SU(3) given by the Gell-Mann matrices [15].

Then, as

$$\frac{1}{2}\{\underset{\sim}{\lambda}_3+\sqrt{3}\underset{\sim}{\lambda}_8\} = \begin{pmatrix} 1 & & \\ & 0 & \\ & & -1 \end{pmatrix} \quad (\underset{\sim}{\lambda}_i \text{ are the familiar SU(3)-matrices})$$

we have after substitution to (2.1.6), using $\underset{\sim}{\Gamma}(F_j) = \frac{1}{2}\underset{\sim}{\lambda}_j$, $j = 1,\ldots,8$,

$$Q^{\ell}\mu_R^+ = +\mu_R^+,\ Q^{\ell}\nu_R = 0,\ Q^{\ell}e_R^- = -e_R^- \tag{2.1.7}$$

as required for the charge of the lepton fields.

For the left-handed helicity fields let

$$\underset{\sim}{e}'_L = \frac{1}{2}(1+\gamma_5)\begin{pmatrix} e^- \\ \nu \\ \mu^+ \end{pmatrix} \tag{2.1.8}$$

be the Konopinski-Mahmound triplet with rows 1 and 3 interchanged. Then the relations

$$F_p \ell'_{L_j} = \sum_{k=1}^{3} \overline{\Gamma(F_p)}_{k_j} \ell'_{L_k}$$

where $\overline{\underset{\sim}{\Gamma}}$ is the $\overline{3}$-dimensional representation of SU(3) give

$$Q^{\ell}e_L^- = -e_L^-,\ Q^{\ell}\nu_L = 0,\ Q^{\ell}\mu_L^+ = \mu_R^+. \tag{2.1.9}$$

In this model there is a doubly charged vector boson coupled to $e\overline{\mu^+}\gamma^{\mu}\gamma_5 e^-$, so that the diagram in Figure 2.1 exists.

The Yukawa interaction takes the form

$$G_{\phi} \sum_{jk\ell} \overline{\ell}_{Lj} \ell_{Rk} \phi_{\ell} \gamma_{jk\ell} + \text{H.C} \tag{2.1.10}$$

where ϕ_{ℓ} are the spinless mesons and the $\gamma_{jk\ell}$ are given by the Clebsch-Gordan coefficients. The Clebsh-Gordan coefficients $\gamma_{jk\ell}$ are given in Appendix A. A general method of evaluating Clebsch-Gordan coefficients will be discussed later.

The fields ϕ_{ℓ} must transform according to

$$\overline{\underset{\sim}{\Gamma}}_L \otimes \underset{\sim}{\Gamma}_R = (\overline{\overline{3}}) \otimes \underset{\sim}{3} = \underset{\sim}{3} \otimes \underset{\sim}{3} = \overline{\underset{\sim}{3}} \oplus \underset{\sim}{6}. \tag{2.1.11}$$

If both fields are included there are no zero-order mass relations; if only the $\underset{\sim}{6}$ is included the zero-order mass relation is $m_{\mu} = m_e$ as a result of the Clebsch-Gordan coefficients (Table A1) and, finally, if only the $\underset{\sim}{3}$ is included again from Table A1 we get $m_{\mu} = -m_e$.

Thus no zero-order mass relations exist giving $m_e = 0$.

Another choice of the gauge group, namely $G = SU(3) \otimes U(1)$ can provide the required zeroth-order mass relations, and in principle a calculable electron mass.

Example II: $SU(3) \otimes U(1)$ model

Let $F_1,\ldots,F_8$ be the SU(3) generators, as in Example I, and F_9 the U(1) generator. Then the leptonic charge operator (2.1.3) becomes

$$Q^{\ell} = F_3 + \sqrt{3}F_8 + F_9. \qquad (2.1.12)$$

Let

$$\underline{\psi}_R = \begin{bmatrix} \mu^+ \cos p + x^+ \sin p \\ \nu \\ e^- \end{bmatrix}_R \qquad (2.1.13)$$

where p is a parameter, (μ^+,ν,e^-) is the Konopinski-Mahmound triplet and x^+ is a heavy lepton. The multiplet $\underline{\psi}_R$ is transforming according to

$$\left.\begin{aligned} F_p\psi_{R_j} &= \sum_{k=1}^{3} (\tfrac{1}{2}\underset{\sim}{\lambda}_p)_{k_j}\psi_{R_k},\ p = 1,\ldots,8 \\ F_9\psi_{R_j} &= 0 \end{aligned}\right\} \qquad (2.1.14)$$

From (2.1.12) we get

$$Q^{\ell}\psi_{R_1} = \psi_{R_1},\ Q^{\ell}\psi_{R_2} = 0,\ Q^{\ell}\psi_{R_3} = -\psi_{R_3} \qquad (2.1.15)$$

which is consistent with the charge assignments of $\underline{\psi}_R$. Let

$$\underline{\psi}'_L = \begin{bmatrix} e^- \\ \nu \\ \mu^+ \cos\lambda - x^+ \sin\lambda \end{bmatrix}_L \qquad (2.1.16)$$

where λ is a parameter, transform according to

$$\left.\begin{aligned} F_p\psi'_{L_j} &= \sum_{k=1}^{3}(-\tfrac{1}{2}\lambda_p)_{k_j}\psi'_{L_k} \\ F_9\psi'_{L_j} &= 0 \end{aligned}\right\} \quad (2.1.17)$$

Then, again from (2.1.12), we get

$$Q^{\ell}\psi'_{L_1} = -\psi'_{L_1},\ Q^{\ell}\psi'_{L_2} = 0,\ Q^{\ell}\psi'_{L_3} = +\psi'_{L_3} \quad (2.1.18)$$

which is consistent with the charge assignments of $\underset{\sim}{\psi}'_L$.

The model also includes SU(3) singlets with transformation properties as follows:

(a) $s_R = x_R^+ \cos p - \mu_R^+ \sin p$

$$\left.\begin{aligned} F_p s_R &= 0,\ p = 1,\ldots,8 \\ F_9 s_R &= s_R \end{aligned}\right\} \quad (2.1.19)$$

(b) $s_L = x_L^+ \cos\lambda + \mu_L^+ \sin\lambda$

$$\left.\begin{aligned} F_p s_L &= 0,\ p = 1,\ldots,8 \\ F_9 s_L &= s_L \end{aligned}\right\} \quad (2.1.20)$$

From (2.1.12), we find

$$Q^{\ell}s_R = s_R,\ Q^{\ell}s_L = s_L$$

as required.

Yukawa couplings

The Yukawa couplings have the following structure. The right-handed fields transform as $\{\underset{\sim}{3} \otimes \underset{\sim}{\Gamma}^0\} \oplus \{\underset{\sim}{1} \otimes \underset{\sim}{\Gamma}^1\} = \underset{\sim}{\Gamma}_R$, where the first factor in every direct product refers to SU(3) and the second to U(1), and where $\underset{\sim}{\Gamma}^a$ is the one-dimension irreducible representation of U(1) in which F_9 has eigenvalue a. Similarly, the left-handed fields transform as $\{\underset{\sim}{\bar{3}} \otimes \underset{\sim}{\Gamma}^0\} \oplus \{\underset{\sim}{1} \otimes \underset{\sim}{\Gamma}^1\} = \underset{\sim}{\Gamma}_L$. Then

$$\bar{\Gamma}_L \otimes \Gamma_R = \overline{[\{3 \otimes \Gamma^0\} \oplus \{1 \otimes \Gamma^1\}]} \otimes [\{\bar{3} \otimes \Gamma^0\} \oplus \{1 \otimes \Gamma^1\}]$$

$$= [\{\bar{3} \otimes \Gamma^0\} \oplus \{1 \otimes \Gamma^{-1}\}] \otimes [\{\bar{3} \otimes \Gamma^0\} \oplus \{1 \otimes \Gamma^1\}], \quad \text{as } \bar{\Gamma}^1 = \Gamma^{-1}$$

$$= \{(\bar{3} \otimes \bar{3}) \otimes \Gamma^0\} \oplus \{\bar{3} \otimes \Gamma^1\} \oplus \{\bar{3} \otimes \Gamma^{-1}\} \oplus \{1 \otimes \Gamma^0\}$$

ie

$$\bar{\Gamma}_L \otimes \Gamma_R = \{\bar{3} \otimes \Gamma^0\} \oplus \{6 \otimes \Gamma^0\} \oplus \{\bar{3} \otimes \Gamma^1\} \oplus \{\bar{3} \otimes \Gamma^{-1}\} \oplus \{1 \otimes \Gamma^0\}. \tag{2.1.21}$$

Thus, if the spinless mesons transforming only as $\bar{3} \otimes \Gamma^1$ and $\bar{3} \otimes \Gamma^{-1}$ are included, then the Yukawa couplings are incomplete, and we have type (b) zeroth-order mass relations. At the same time, because there is no spinless meson coupled to $\bar{e}e$, the electron is massless in zeroth order.

Let ϕ^1_j, j = 1,2,3, be the spinless mesons fields transforming as $\bar{3} \otimes \Gamma^1$, ie

$$\left.\begin{aligned} F_p\phi^1_j &= \sum_{k=1}^{3} (-\tfrac{1}{2}\lambda_p)^*_{k_j} \phi^1_k \\ F_9\phi^1_j &= \phi^1_j \end{aligned}\right\} \tag{2.1.22}$$

and let ϕ^2_j, j = 1,2,3, be the spinless meson fields transforming as $\bar{3} \otimes \Gamma^{-1}$, ie

$$\left.\begin{aligned} F_p\phi^2_j &= \sum_{k=1}^{3} (-\tfrac{1}{2}\lambda_p)^*_{k_j} \phi^2_j \\ F_9\phi^2_9 &= -\phi^2_9 \end{aligned}\right\} \tag{2.1.23}$$

Then, from the charge operator (2.1.12), we get

$$\left.\begin{aligned} &Q^\ell\phi^1_1 = 0, \quad Q^\ell\phi^1_2 = \phi^1_2, \quad Q^\ell\phi^1_3 = 2\phi^1_3 \\ &Q^\ell\phi^2_1 = -2\phi^2_1, \quad Q^\ell\phi^2_2 = -\phi^2_2, \quad Q^\ell\phi^2_3 = 0 \end{aligned}\right\} \tag{2.1.24}$$

Thus the ϕ^1_1 and ϕ^2_3 are the neutral fields. The SU(3) ⊗ U(1) invariant Yukawa interaction Lagrangian density may then be taken to be

$$\mathcal{L}_Y = f \sum_{j=1}^{3} \bar{s}_L \phi_j^1 \psi_{R_j} + f' \sum_{j=1}^{3} \bar{\psi}^1_{L_j} \phi_j^2 s_R + \text{H.C}\,. \tag{2.1.25}$$

To preserve the electromagnetic invariance of the charge operator Q^{ℓ} only the neutral mesons fields ϕ_1^1 and ϕ_3^2 are allowed to develop non-zero vacuum expectation values, ie

$$\left.\begin{array}{l} \langle\phi_1^1\rangle_0 = a,\ \langle\phi_2^1\rangle_0 = 0,\ \langle\phi_3^1\rangle = 0 \\ \langle\phi_1^2\rangle_0 = 0,\ \langle\phi_2^2\rangle_0 = 0,\ \langle\phi_3^2\rangle_0 = b \end{array}\right\} \tag{2.1.26}$$

with $a \neq 0$, $b \neq 0$.

Substituting (2.1.26) to (2.1.25) the zero-order contribution of $\mathcal{L}_Y$ is

$$\mathcal{L}_Y^{(0)} = fa\bar{s}_L \psi_{R'} + f'b\bar{\psi}'_{L_3} s_R + \text{H.C}\,. \tag{2.1.27}$$

We can also assume that there is a mass-term associated with the singlets s_R, s_L. The only possible SU(3) ⊗ U(1) invariant term is

$$\mathcal{L}_m = m\bar{s}_L s_R + \text{H.C}\,. \tag{2.1.28}$$

No mass terms are possible for the fermion triplets ψ_R and ψ'_L as they would give a non-zero mass for the neutrinos.

Thus the zero-order lepton mass terms are

$$\begin{aligned} \mathcal{L}_m^{(0)} = \mathcal{L}_m + \mathcal{L}_Y^{(0)} &= fa\bar{s}_L \psi_{R_1} + f'b\bar{\psi}'_{L_3} s_R + m\bar{s}_L s_R + \text{H.C} \\ &= fa\{\bar{x}_L^+ \cos\lambda + \bar{\mu}_L^+ \sin\lambda\}\{\mu^+ \cos p + x^+ \sin p\} \\ &\quad + f'b\{\bar{\mu}^+ \cos\lambda - \bar{x}^+ \sin\lambda\}\{x_R^+ \cos p - \mu_R^+ \sin p\} \\ &\quad + m\{\bar{x}_L^+ \cos\lambda + \bar{\mu}_L^+ \sin\lambda\}\{x_R^+ \cos p - \mu_R^+ \sin p\} + \text{H.C} \end{aligned} \tag{2.1.29}$$

However, we require that $\mathcal{L}_m^{(0)}$ has the form

$$\mathcal{L}^0_{(m)} = m_\mu \bar{\mu}^+ \mu^+ + m_x \bar{x}^+ x^+ \tag{2.1.30}$$

with no $\overline{\mu}^+ x^+$ terms etc. Thus, comparing (2.1.29) and (2.1.30), we found the relations

$$\left.\begin{aligned} m_\mu &= fa \sin\lambda \cos p - f'b \cos\lambda \sin p - m \sin\lambda \sin p \\ m_x &= fa \cos\lambda \sin p - f'b \sin\lambda \cos p + m \cos\lambda \cos p \end{aligned}\right\} \quad (2.1.31)$$

and

$$\left.\begin{aligned} fa &= \frac{m \cos p \sin p}{1 - \sin^2\lambda - \sin^2 p} \\ f'b &= -\left\{\frac{m \cos\lambda \sin\lambda}{1 - \sin^2\lambda - \sin^2 p}\right\} \end{aligned}\right\} \quad (2.1.32)$$

From (2.1.31) and (2.1.32), we get

$$m_\mu \cos\lambda \cos p = m_x \sin\lambda \sin p. \quad (2.1.33)$$

Lepton-boson couplings

Let A_{p_μ}, $p = 1,\ldots,9$, be the vector boson fields. Then in the minimal substitutions [16,17] for lepton fields we have

$$\left.\begin{aligned} \nabla_\mu \psi_{R_j} &= \partial_\mu \psi_{R_j} - \tfrac{1}{2} i g \sum_{p=1}^{8} \sum_{k=1}^{3} A_{p_\mu} (\underset{\sim}{\lambda}_p)_{k_j} \psi_{R_k} \\ \nabla_\mu \psi'_{L_j} &= \partial_\mu \psi'_{L_j} + \tfrac{1}{2} i g \sum_{p=1}^{8} \sum_{k=1}^{3} A_{p_\mu} (\underset{\sim}{\lambda}_p)^*_{k_j} \psi'_{L_j} \\ \nabla_\mu s_R &= \partial_\mu s_R - i g' A_{9_\mu} s_R \\ \nabla_\mu s_L &= \partial_\mu s_L - i g' A_{9_\mu} s_L \end{aligned}\right\} \quad (2.1.34)$$

where g is the gauge coupling constant for the SU(3) group, and g' the coupling constant for the U(1) group. The lepton-vector meson coupling term is

$$\begin{aligned} \mathcal{L}_{\ell_A} = {} & \tfrac{1}{2} g \sum_{j=1}^{3} \sum_{k=1}^{3} \sum_{p=1}^{8} \overline{\psi}_{R_j} \gamma^\mu A_{p_\mu} (\underset{\sim}{\lambda}_p)_{k_j} \psi_{R_k} \\ & - \tfrac{1}{2} g \sum_{j=1}^{3} \sum_{k=1}^{3} \sum_{p=1}^{8} \overline{\psi}^L_{L_j} \gamma^\mu A_{p_\mu} (\underset{\sim}{\lambda}_p)^*_{k_j} \psi'_{L_k} \\ & + g' \overline{s}_R \gamma^\mu A_{9_\mu} s_R + g' \overline{s}_L \gamma^\mu A_{9_\mu} s_L . \end{aligned} \quad (2.1.35)$$

The $\mathcal{L}_{\ell_A}$ part of the Lagrangian gives rise, after some algebra, to all possible SU(3) ⊗ U(1) invariant interaction terms among the lepton fields.

Vector boson masses

In the minimal couplings for the spinless meson fields we have

$$\left.\begin{aligned} \nabla_\mu \phi_j^1 &= \partial_\mu \phi_j^1 - ig \sum_{p=1}^{8} \sum_{k=1}^{3} A_{p_\mu} (-\tfrac{1}{2}\lambda_p)^*_{k_j} \phi_k^1 - ig' A_{9_\mu} \phi_j^1 \\ \nabla_\mu \phi_j^2 &= \partial_\mu \phi_j^2 - ig \sum_{p=1}^{8} \sum_{k=1}^{3} A_{p_\mu} (-\tfrac{1}{2}\lambda_p)^*_{k_j} \phi_k^2 - ig' A_{9_\mu} \phi_j^2 \end{aligned}\right\} \qquad (2.1.36)$$

Minimal substitution in $\sum_{j=1}^{3} \{\partial_\mu \phi_j^{1+} \partial^\mu \phi_j^1 + \partial_\mu \phi_j^{2+} \partial^\mu \phi_j^2\}$ and the use of the vacuum expectation values (2.1.24) leads to the following form of the $\mathcal{L}_{\phi_A}$ Lagrangian describing the couplings of the gauge bosons.

$$\mathcal{L}_{\phi_A} = \frac{1}{2}(M_W)^2 W_\mu^+ W^{\mu^{++}} + \frac{1}{2}(M_D)^2 W_{\mu_D} W_D^{\mu+} + \frac{1}{2}(M_B)^2 W_{B_\mu} W_B^{\mu+}$$

$$+ \frac{1}{2}(M_{Z_0})^2 Z_{0_\mu} Z_0^{\mu^+} + \frac{1}{2}(M_{Z_0'})^2 Z'_{0_\mu} Z_0'^{\mu^+} \qquad (2.1.37)$$

where

$$\left.\begin{aligned} W_\mu^+ &= \frac{1}{\sqrt{2}}(A_{1\mu} + iA_{2\mu}) \\ W_{D_\mu} &= \frac{1}{\sqrt{2}}(A_{4\mu} + iA_{5\mu}) \\ W_{B_\mu} &= \frac{1}{\sqrt{2}}(A_{6\mu} + iA_{7\mu}) \\ Z_{0_\mu} &= \sqrt{\frac{3}{g^2 + 3g'^2}}\left(\frac{g}{2}A_{3\mu} + \frac{g}{2\sqrt{3}}A_{8\mu} - g'A_{9\mu}\right) \\ Z'_{0_\mu} &= \sqrt{\frac{3}{g^2 + 3g'^2}}\left(\frac{g}{\sqrt{3}}A_{8\mu} - g'A_{9\mu}\right) \end{aligned}\right\} \qquad (2.1.38)$$

and the photon field is

$$A_\mu = \frac{1}{\sqrt{g^2 + 4g'^2}} (g'A_{3\mu} + g'\sqrt{3}A_{8\mu} + gA_{9\mu}) \qquad (2.1.39)$$

and

$$\left.\begin{array}{l} M_W = ag \ , \ M_D = g\sqrt{a^2+b^2} \ , \ M_B = bg \\ M_{Z_0} = a\sqrt{\frac{2}{3}(g^2+3g'^2)} \ , \ M_{Z_0'} = b\sqrt{\frac{2}{3}(g^2+3g'^2)} \\ M_{A_\mu} = 0 \end{array}\right\} \qquad (2.1.40)$$

If we assume that $b \gg a$, then $W_{D_\mu}, W^+_{D_\mu}, W_{B_\mu}, W^+_{B_\mu}, Z'_{0_\mu}$ are all super heavy. The only non-super-heavy vector bosons are then W^+_μ, W^{++}_μ and $Z_{0\mu}$ together with the massless photon field. In the $\mathcal{L}_{\ell_A}$ part of the Lagrangian (2.1.35) the wrong-helicity current, $\bar{\nu}\gamma^\mu(1-\gamma_5)e^-$ is coupled to W_{B_μ} and is therefore suppressed. So also is the doubly charged current coupled to W_{D_μ}.

To calculate the electron mass we must combine the diagram of Figure 2.1 with the diagram of Figure 2.2.

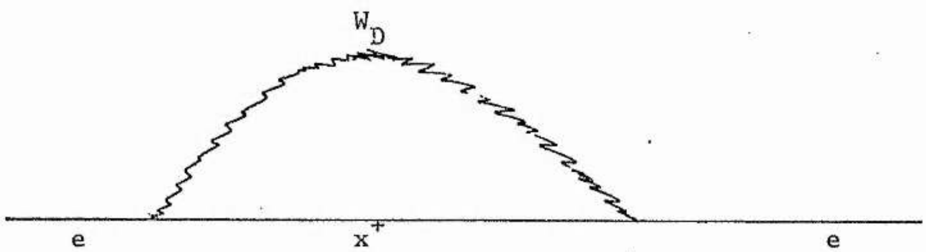


Figure 2.2: Feynman diagram for the heavy x^+ in the SU(3) ⊗ U(1) model which leads to a calculable electron mass

Because of the relation (2.1.33) the logarithmic divergences from the diagrams of Figures 2.1 and 2.2 cancel (we give the full calculations in Appendix A), and we get for the electron mass ,

$$m_e = am_\mu \cos p \cos\lambda \, \frac{3}{16\pi\cos^2\theta} \cdot \frac{m_x^2}{m_D^2 - m_x^2} \, \ell n \, \frac{m_D^2}{m_x^2} + O\left(\frac{m_\mu^2}{m_D^2}\right). \qquad (2.1.41)$$

In this model the unobserved lepton must be extremely massive. In fact, from (2.1.33) to give a reasonable value for the electron mass, m_x must be of the same order of magnitude as m_D, which is at least a few hundred GeV.

The solution (2.1.41) depends on several parameters which, while measurable in principle and conceptually acceptable especially because of the developments in the grand unification theories, are not accessible to experiments in the immediate future. This difficulty has been overcome by Barr and Zee, who proposed a class of models [11] in which the electron mass m_e can be expressed in terms of quantities already measured. The electron mass in these models has the form

$$m_e = N\frac{\alpha m_\mu}{\pi \sin^2\theta_W} + \text{(corrections)}. \qquad (2.1.42)$$

Here $\sin\theta_W = e/g$ (where g is the gauge coupling of weak interactions) can be measured in the neutral-current interactions. N is a pure number which varies from model to model depending upon the dimensions of the lepton multiplets and the choice of gauge group. The corrections referred to, Eq (2.1.42), are dependent upon unknown parameters but are down in magnitude by one power of the logarithm of a large quantity (the ratio of the mass of a heavy boson to the mass of a heavy lepton).

Their work is a generalization of the Georgi-Glashow solution in the SU(3) ⊗ U(1) model. They supposed that both the electron <u>and</u> the muon are massless at zeroth order; the muon derives its mass at the one-loop level from diagrams of the sort shown in Figure 2.3(a), and the electron derives its mass at the two-loop level from diagrams of the sort shown in Figure 2.3(b). They followed this procedure because much of the unknown parameter dependence may be expressed to be common to the expressions for m_e and m_μ, and thus to cancel out in the ratio m_e/m_μ.

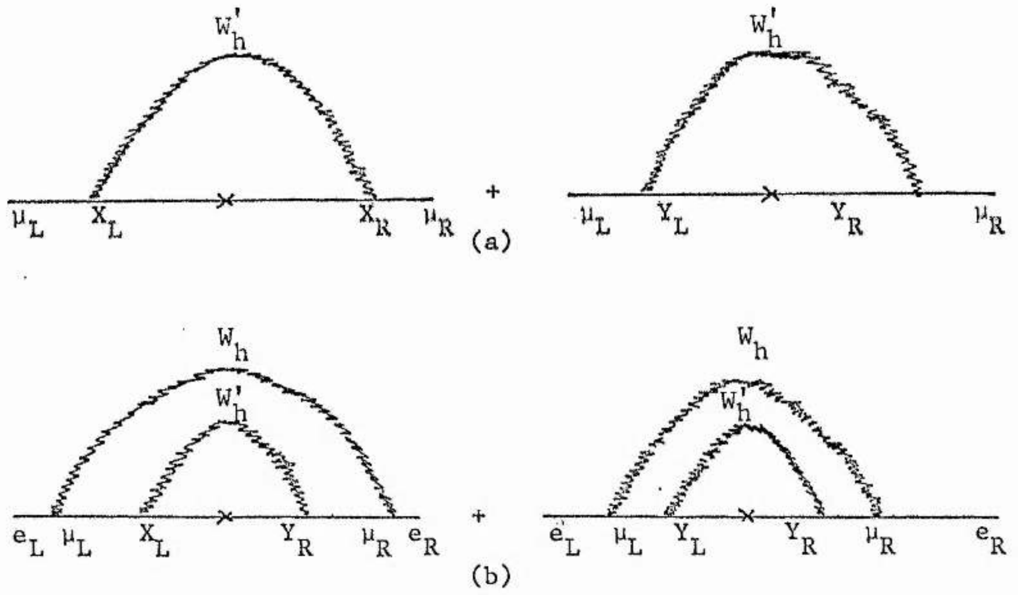


Figure 2.3: Diagrams contributing to the muon and electron masses: X^-, Y^- are heavy leptons; $W_h, W_{h'}$ are heavy gauge bosons

In spite of the simplicity of the formula (2.1.42), there is no fundamental reason why, in the true theory, the electron should be expressible in terms of the other physical constants which are currently measurable by physicists.

§2.2 An Orthogonal Model for Explaining the Electron-Muon Mass Ratio

In this section we shall study the SO(5) orthogonal model in connection with the electron-muon problem. The group SO(5) (B_2 Lie algebra) has 10 generators. We shall denote the two diagonal generators by H_i, $i = 1,2$ and the eight non-diagonal generators by $E_{\pm j}$, $j = 1,2,3,4$. The algebra B_2 has eight roots $\pm\alpha_1, \pm\alpha_2, \pm(\alpha_1+\alpha_2)$ and $\pm(\alpha_1+2\alpha_2)$ where α_1, α_2 are the two simple roots. The lowest dimensional representation has dimension four and it is self-conjugate [18]. We give an explicit four-dimensional matrix representation of the generators H,E in Appendix A. A general method of obtaining an explicit matrix realization of irreducible representations will be considered in Chapter 4.

The leptonic charge operator is taken as

$$Q^{\ell} = T_3 + \frac{1}{2}Y \tag{2.2.1}$$

where

$$\left.\begin{aligned} T_3 &= \frac{H_1}{4(6)^{1/2}} \\ Y &= \frac{H_2}{2(6)^{1/2}} \end{aligned}\right\} \tag{2.2.2}$$

Let

$$\psi_L = \begin{bmatrix} \mu^+ \\ v_\mu \\ v_e \\ e^- \end{bmatrix}_L \tag{2.2.3}$$

denote the left-handed lepton fields and let

$$\psi_R = \begin{bmatrix} \mu^+ \\ v_\mu \\ v_e \\ e^- \end{bmatrix}_R \tag{2.2.4}$$

denote the right-handed fields.

The spinless meson fields should transform according to an irreducible component of

$$\underset{\sim}{4} \otimes \overline{\underset{\sim}{4}} = \underset{\sim}{4} \otimes \underset{\sim}{4} = \underset{\sim}{10} \oplus \underset{\sim}{5} \oplus \underset{\sim}{1} \tag{2.2.5}$$

in order to have a Yukawa term invariant under SO(5) transformations.

In Figure A2 in Appendix A we give the weight diagrams of the representations included in relations (2.1.5) and in Table A2 the complete set of the Clebsch-Gordan coefficients of $\underset{\sim}{4} \otimes \underset{\sim}{4}$.

Let A_{p_μ}, $p = 1,\ldots,10$, be the vector boson fields. If with $\underset{\sim}{\Gamma}(F_p)$, $p = 1,2,\ldots,10$, we denote the four-dimensional matrices representing the generators H and E, then the part of the Lagrangian representing the lepton-vector boson couplings becomes

$$\mathcal{L}_{\ell_A} = g\sum_{p=1}^{10}\sum_{j,k=1}^{4} \overline{\psi}_{L_j}\gamma^\mu A_{p_\mu}\Gamma(F_p)_{k_j}\psi_{L_k} + g\sum_{p=1}^{10}\sum_{j,k=1}^{4} \overline{\psi}_{R_j}\gamma^\mu A_{p_\mu}\Gamma(F_p)_{k_j}\psi_{R_k} \qquad (2.2.6)$$

where g is the gauge coupling constant. Using the relations

$$\left.\begin{aligned} \overline{\psi}_{L_j}\gamma^\mu\psi_{L_k} &= \tfrac{1}{2}\overline{\psi}_j\gamma^\mu(1+\gamma_5)\psi_k \\ \overline{\psi}_{R_j}\gamma^\mu\psi_{R_k} &= \tfrac{1}{2}\overline{\psi}_j\gamma^\mu(1-\gamma_5)\psi_k \end{aligned}\right\} \qquad (2.2.7)$$

we found that with the choice of the lepton fields given by (2.2.3) and (2.2.4) we do not have the V-A structure of the weak currents.

The introduction of an abelian U(1) group gives us the required form of the weak currents. If we denote the generator of the U(1) group by F_{11} then the charge operator is taken as

$$Q^\ell = \tfrac{1}{2}F_2 - \tfrac{1}{2}F_{11}. \qquad (2.2.8)$$

The left-handed fields

$$\psi_L = \begin{bmatrix} e^- \\ \nu_e \\ \mu^- \\ \nu_\mu \end{bmatrix}_L \qquad (2.2.9)$$

transform under $SO(5) \otimes U(1)$ as

$$\left.\begin{aligned} F_p \psi_{L_j} &= \sum_{k=1}^{4} \Gamma(F_p)_{k_j} \psi_{L_k}, \; p = 1,\ldots,10 \\ F_{11} \psi_{L_j} &= 0 \end{aligned}\right\} \tag{2.2.10}$$

For the right-handed fields we have considered two singlets e_R^- and μ_R^- with transformation properties under $SO(5) \otimes U(1)$

$$\left.\begin{aligned} & F_p e_R^- = 0, \; F_p \mu_R^- = 0, \; p = 1,\ldots,10 \\ & F_{11} e_R^- = 2e_R^-, \; F_{11}\mu_R^- = 2\mu_R^- \end{aligned}\right\} \tag{2.2.11}$$

Their couplings to vector bosons are

$$\mathcal{L}_{\ell_A} = \frac{1}{2} g \sum_{p=1}^{10} \sum_{j,k=1}^{4} \overline{\psi}_j \gamma^\mu (1+\gamma_5) \psi_k \Gamma(F_p) A_{p_\mu} + \frac{1}{2} g' \overline{e}^- \gamma^\mu (1-\gamma_5) e^- F_{11} B_\mu$$
$$+ \frac{1}{2} g' \overline{\mu}^- \gamma^\mu (1-\gamma_5) \mu^- F_{11} B_\mu \tag{2.2.12}$$

where g' is the gauge coupling constant of the $U(1)$ group, and B_μ is the vector boson associated with the Abelian factor $U(1)$. If we choose two spinless mesons ϕ_i, ϕ_i', $i = 1,2,\ldots,4$, to transform as the four-dimensional representation of $SO(5)$ and have -1 as the F_{11} eigenvalue, then the Yukawa coupling becomes

$$\mathcal{L}_Y = f \overline{\psi}_L \phi e_R^- + f' \overline{\psi}_L \phi' \mu_R^- . \tag{2.2.13}$$

If we choose the vacuum expectation values of the ϕ and ϕ' fields consistently with the charge operator (2.2.8)

$$\langle \phi \rangle_0 = \begin{pmatrix} a \\ 0 \\ b \\ 0 \end{pmatrix} \quad \text{and} \quad \langle \phi' \rangle_0 = \begin{pmatrix} a' \\ 0 \\ b' \\ 0 \end{pmatrix}, \tag{2.2.14}$$

then $\mathcal{L}_Y$ ((2.2.13)) becomes

$$\mathcal{L}_Y = f(\bar{e}^-,\bar{\nu}_e^-,\bar{\mu}^-,\bar{\nu}_\mu)_L \begin{bmatrix} a \\ 0 \\ b \\ 0 \end{bmatrix} e_R^- + f'(\bar{e}^-,\bar{\nu}_e^-,\bar{\mu}^-,\bar{\nu}_\mu)_L \begin{bmatrix} a' \\ 0 \\ b' \\ 0 \end{bmatrix} \mu_R^- . \quad (2.2.15)$$

The first term in (2.2.15) gives unwanted mixings as $b\bar{\mu}_L^- e_R^-$ so if we impose on the Lagrangian the discrete symmetry

$$\left.\begin{matrix} \psi_L \to \psi_L \\ e_R \to -e_R \\ \mu_R^- \to +\mu_R^- \end{matrix}\right\} \quad (2.2.16)$$

terms like $b\bar{\mu}_L^- e_R^-$ are suppressed. The second term in (2.2.15) also gives unwanted terms like $a'\bar{e}_L^- \mu_R^-$, but we can assume $a' = 0$, hoping that the potential

$$V = -c_1(\phi'^+ \cdot \phi') - c_2(\phi'^+ \cdot \phi')^2 \quad (2.2.17)$$

when minimized will give for only b' a value different from zero.

The introduction of the discrete symmetry (2.2.16) gives us a zeroth-order mass relation, but the couplings described by the Lagrangian $\mathcal{L}_{\phi_A}$, after the introduction of the vacuum expectation values, result in a zero mass gauge boson responsible for diagram of Figure 2.1.

If we try to overcome this difficulty by introducing two multiplets

$$\psi_L = \begin{bmatrix} e^- \\ \nu_e \\ \mu^- \\ \nu_\mu \end{bmatrix}_L \quad \text{and} \quad \psi_R = \begin{bmatrix} e^- \\ \nu_e \\ \mu^- \\ \nu_\mu \end{bmatrix}_R \quad (2.2.18)$$

with transformation properties under $SO(5) \otimes U(1)$:

$$\left.\begin{aligned} F_p\psi_{L_i} &= \sum_{k=1}^{4}\Gamma(F_p)_{k_j}\psi_{L_k}, \; p = 1,\ldots,10 \\ F_{11}\psi_{L_i} &= \psi_{L_i} \end{aligned}\right\} \tag{2.2.19}$$

and

$$\left.\begin{aligned} F_p\psi_{R_i} &= \sum_{k=1}^{4}\Gamma(F_p)_{k_j}\psi_{R_k} \\ F_{11}\psi_{R_i} &= \psi_{R_i} \end{aligned}\right\} \tag{2.2.20}$$

then the Yukawa coupling has the structure

$$\begin{aligned} \bar{\underset{\sim}{\Gamma}}_L \otimes \underset{\sim}{\Gamma}_R &= \{\underset{\sim}{4} \otimes \underset{\sim}{\Gamma}^1\} \otimes \{\underset{\sim}{4} \otimes \underset{\sim}{\Gamma}^1\} = \{\bar{\underset{\sim}{4}} \otimes \underset{\sim}{\Gamma}^{-1}\} \otimes \{\underset{\sim}{4} \otimes \underset{\sim}{\Gamma}^1\} \\ &= \{\underset{\sim}{4} \otimes \underset{\sim}{4}\} \otimes \underset{\sim}{\Gamma}^0 = \{\underset{\sim}{1} \oplus \underset{\sim}{5} \oplus \underset{\sim}{10}\} \otimes \underset{\sim}{\Gamma}^0 \\ &= \{\underset{\sim}{1} \otimes \underset{\sim}{\Gamma}^0\} \oplus \{\underset{\sim}{5} \otimes \underset{\sim}{\Gamma}^0\} \oplus \{\underset{\sim}{10} \otimes \underset{\sim}{\Gamma}^0\} \end{aligned} \tag{2.2.21}$$

and the spinless mesons should transform as $\underset{\sim}{1} \otimes \underset{\sim}{\Gamma}^0, \underset{\sim}{5} \otimes \underset{\sim}{\Gamma}^0$ or $\underset{\sim}{10} \otimes \underset{\sim}{\Gamma}^0$, a structure too complicated to expect reasonable value for the electron mass. Nevertheless, using Table A2 of the Clebsch-Gordan coefficients we found that if the $\underset{\sim}{5}$ representation is included then there is no zeroth-order mass relation while if $\underset{\sim}{10}$ is included there is a mass relation but only after a specific choice of the vacuum expectation values, which choice is not necessarily the one which the complicated potential would have made.

In concluding this chapter, we would like to remark that, in spite of the unsuccessful attempts to calculate the electron-muon mass ratio, the further developments in the theory of natural symmetries [20] to which these attempts have made a considerable contribution, have helped us to gain a better understanding of the mass problem as a whole [21,22].

CHAPTER 3

GRAND UNIFIED THEORIES

It is now widely accepted that weak and electromagnetic forces are mediated by the vector bosons of a gauge invariant theory with spontaneous symmetry breaking. The SU(2) ⊗ U(1) gauge model of Glashow-Weinberg-Salam [23] proved to be the most successful among a large class of models that aim to describe the observed weak interaction phenomenology [24].

At the same time the developments in the strong interactions area, established SU(3)-colour as the gauge group of these interactions, while chromodynamics was asserted to be the field theory describing the strong forces [25]. Chromodynamics tells us that the strong interactions are mediated by an octet of neutral vector gauge gluons associated with local SU(3) symmetry. Moreover, since the strong interactions are associated with a non-Abelian theory, they are asymptotically free [26].

These two interrelated developments in the theory of weak-electromagnetic and strong interactions made it possible to advance a step further and try to unify all the forces except gravity under a single gauge group. The idea of Grand Unification is conceptually very attractive because it leads to a more symmetrical perception of the natural phenomena, and at the same time provides the hope of understanding some of the outstanding problems in gauge field theory, such as the quantization of charge, the stability of the proton, the mass problem and the problem of gauge hierarchies.

The first attempts towards a grand unification theory were made by Pati and Salam [27] who proposed an SU(4) ⊗ SU(4) theory, Fritzsch and Minkowski who studied a class of Unitary and Orthogonal groups and Georgi and Glashow, who in 1974 proposed the SU(5) group [28] as a single gauge group for grand unification. Since then, various models based on orthogonal and exceptional groups have been proposed. We shall review these models and discuss their predictions in the first part of this chapter.

Once the conceptual problems of grand unification have been solved, a model-builder is free to consider any higher order gauge group and representations of any dimensionality. However, there are two basic restrictions for the choice of the gauge group. The first and the most important is the colour restriction [29] which expresses the basic requirement that any gauge group should include the SU(3)-colour group as its subgroup, which in terms of fermion fields means that only triplets, antitriplets and singlets under colour are allowed. The second restriction is connected with the cancellation of the Adler-Bell-Jackiw anomalies [30] among the representations of the gauge group [31]. There is also a constraint on the fermion and spinless mesons content of grand unified gauge theories, imposed by the requirements of asymptotic freedom of the gauge couplings [32]. Nevertheless these restrictions are not enough to prevent a proliferation of possible gauge models of grand unification.

We believe that nature is not economical of structures but only of principles of fundamental applicability. For that reason the problem of dealing with high order groups, and large number of elementary fields, must be considered. In part II of this chapter, the problem will be set in mathematical terms, and in the following chapters a method will be developed to deal with high dimension representations of high order groups.

PART I: Models of Grand Unification

From the requirement that the unifying gauge group G should include SU(3) ⊗ SU(2) ⊗ U(1), we have that the rank of G must be at least four. There are nine rank-4 semi-simple compact Lie groups which can involve only one coupling constant: $[SU(2)]^4$, $[SO(5)]^2$, $[SU(3)]^2$, $[G_2]^2$, SO(8), SO(9), Sp(8), F_4 and SU(5).

The first two, since they do not contain SU(3), are unacceptable. To see which group we have to choose, let us study the general fermion content we want our group G to have. We classify the known fermions according to their masses as follows:

First generation

$\{u^i, d^i, e^-, \nu_e$ + their antiparticles$\}$,

Second generation

$\{c^i, s^i, \mu^-, \nu_\mu$ + their antiparticles$\}$,

Third generation

$\{t^i, b^i, \tau^-, \nu_\tau$ + their antiparticles$\}$,

where the index i represents the three colours R, W, B.

Consider the first generation. Its transformation properties under $SU(3)_C \otimes SU(2)_L$ are (the indices C and L refer to colour and left-hand helicity respectively)

$$(\underset{\sim}{1},\underset{\sim}{2}) \oplus (\underset{\sim}{1},\underset{\sim}{1}) \oplus (\underset{\sim}{3},\underset{\sim}{2}) \oplus 2(\overline{\underset{\sim}{3}},\underset{\sim}{1}),$$

ie

$$\binom{e}{\nu_e}_L \quad e_R \quad \binom{u^i}{d^i}_L \quad u^i_R, d^i_R$$

which is obviously a complex representation. The complex nature of the fermion fields representations with respect to $SU(3)_C \otimes SU(2)_L \otimes U(1)$ is a very important criterion for a good grand unification theory, because it guarantees the V-A structure of the weak currents. This fact was expressed by Georgi [33] as a 'law' for a good grand unification theory. From the remaining groups, only $[SU(3)]^2$ and SU(5) admit complex representations [34]. The group SU(3) ⊗ SU(3) is excluded. One reason is that the generator corresponding to electric charge does not admit fractional charges [22].

The SU(5) model, the only rank-4 possibility, became known as the 'standard' grand unified model, because among all the other models in which the spinless mesons are considered as elementary fields (orthodox theories) it incorporates all the basic features of the grand unification scheme and because it is the most economical model.

(a) SU(5) model

We choose two irreducible representations to describe the physics of the first generation, the $\underset{\sim}{10}$ and $\underset{\sim}{\overline{5}}$ representations. Among the lowest dimension representations of SU(5), only the combination of $\underset{\sim}{10}$ and $\underset{\sim}{\overline{5}}$ cancels out the Adler-Bell-Jackiw anomalies [28].

The fermion content of the SU(5) model is

$$\underset{\sim}{\overline{5}} = \begin{pmatrix} \overline{d}_R \\ \overline{d}_W \\ \overline{d}_B \\ e^- \\ \nu_e \end{pmatrix}_L , \quad \underset{\sim}{10} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & \overline{u}_B & -\overline{u}_W & -u_R & -d_R \\ -\overline{u}_B & 0 & \overline{u}_R & -u_W & -d_W \\ \overline{u}_W & -\overline{u}_R & 0 & -u_B & -d_B \\ u_R & u_W & u_B & 0 & -e^+ \\ d_R & d_W & d_B & e^+ & 0 \end{pmatrix}_L$$

The vector bosons are in the $\underset{\sim}{24}$ adjoint representation with transformation properties under $SU(3)_C \otimes SU(2)_L$, ie

$$\underset{\sim}{24} = (\underset{\sim}{8}^C,\underset{\sim}{1}) \oplus (\underset{\sim}{1}^C,\underset{\sim}{3}) \oplus (\underset{\sim}{1}^C,\underset{\sim}{1}) \oplus (\underset{\sim}{3}^C,\underset{\sim}{2}) \oplus (\underset{\sim}{\overline{3}}^C,\underset{\sim}{2}) \tag{1.1}$$

gluon octet $SU(3)_C$	SU(2) ⊗ U(1) theory $W^{\pm}$, Z, γ	lepton-quark, quark-quark gauge bosons.

The lepton-quark gauge bosons are responsible for the proton decay. The minimal Higgs structure is $\underset{\sim}{24} + \underset{\sim}{5}$, and the symmetry breaking proceeds in two stages

$$SU(5) \xrightarrow{\underset{\sim}{24}} SU(2) \otimes SU(3)_C \otimes U(1) \xrightarrow{\underset{\sim}{5}} SU(3)_C \otimes U(1). \qquad (1.2)$$

At the first stage 12 vector bosons acquire (very large) mass leaving the 8 gluons, $W^{\pm}$, Z, γ massless. At the second stage $W^{\pm}$, Z get masses.

The fermions get masses from the spinless mesons which transform as one or more irreducible components of

$$\left.\begin{aligned} &\overline{\underset{\sim}{5}} \otimes \underset{\sim}{10} = \underset{\sim}{5} \oplus \underset{\sim}{45} \\ &\underset{\sim}{10} \otimes \underset{\sim}{10} = \overline{\underset{\sim}{5}} \oplus \overline{\underset{\sim}{45}} \oplus \underset{\sim}{50} \end{aligned}\right\} \qquad (1.3)$$

If $\underset{\sim}{5}$ is included, then from the coupling

$$\overline{\underset{\sim}{5}}_{(\text{fermion})} \times \underset{\sim}{5}_{(\text{meson})} \times \underset{\sim}{10}_{(\text{fermion})} \qquad (1.4)$$

we get the mass relation $m_{d,s,b} = m_{\ell^-}$, while the coupling

$$\underset{\sim}{10}_{(\text{fermion})} \times \underset{\sim}{5}_{(\text{meson})} \times \underset{\sim}{10}_{(\text{fermion})} \qquad (1.5)$$

gives masses to u,c,t [35]. If $\underset{\sim}{45}$ is included, then we get the mass relation $m_{d,s,b} = -\frac{1}{3} m_{\ell^-}$ which is unacceptable. In the symmetry limit the Weinberg angle has the value $\sin^2\theta_W = \frac{3}{8}$. Using the renormalization group equations [36], we get the renormalized values $\sin^2\theta_W \simeq 0.20$, $m_s \simeq 0.4$ GeV, $m_b = 5$ GeV, and for the mass scale where the unification occurs, we get $M = 10^{16}$ GeV.

The SU(5) model, in spite of some successful predictions (like the masses of s and b quarks), failed to give the right value for the ratio m_e/m_d. But nevertheless it was the first model which put a limit on the grand unification mass, and gave the indication that the proton might be unstable with a value of its life time $\tau_p \simeq 10^{33}$ years. At the same time, because of the mass relation $m_{d,s,b} = m_{\ell^-}$, it offered another possibility to the old lepton mass problem, that the muon-electron mass splitting might have the same origin as the SU(3) breaking [28].

(b) SO(10) model

The SO(10) model was first proposed by Fritzsch and Minkowski [37], and subsequently analysed by Chanowitz, Ellis and Gaillard [38], and further developed by Georgi and Nanopoulos [39].

The SO(10) model is free of Adler-Bell-Jackiw anomalies, a property which is shared by all orthogonal groups [31]. The SU(5) model of Georgi and Glashow is naturally included in the SO(10) theory because $SU(5) \otimes U(1)$ is one of the maximal subgroups of SO(10), under which the $\underset{\sim}{16}$ spinorial representation decomposes as

$$\underset{\sim}{16} = \underset{\sim}{\overline{5}} \oplus \underset{\sim}{10} \oplus \underset{\sim}{1}. \tag{2.1}$$

From (2.1) we can see that all the features of the SU(5) theory are included in the SO(10) model, and furthermore the Abelian factor U(1) provides the missing helicity of the SU(5) model, so the neutrino is not automatically massless. The fermions of each generation (with the right hand neutrino fields) transform according to two $\underset{\sim}{16}$ spinorial representations which are complex [32].

From the group theoretical point of view, the SO(10) is a rank-5 group. There are five rank-5 semi-simple compact Lie groups which can involve only one coupling constant, or admit a discrete symmetry and so can have a unique coupling constant. These are $[SU(2)]^5$, SO(10), SO(11), SU(6) and Sp(10). Of these possibilities, $[SU(2)]^5$ has no SU(3) subgroup, while SO(11), SU(6) and Sp(10) have no 16-dimensional complex representations suitable for the fundamental fermions.

An important feature of this model is that the weak and electromagnetic interactions are described by an ambidextrous $SU(2)_L \otimes SU(2)_R \otimes U(1)$ theory. In the $SU(2)_L \otimes SU(2)_R \otimes U(1)$ theory we have natural zeroth-order relations among the mixing angles and quark masses [40]; thus, there is a hope of recovering these successful relations in the SO(10) theory.

The symmetry breaking proceeds through the following stages [38]:

$$SO(10) \to SU(2)_L \otimes SU(2)_R \otimes SU(4) \to SU(2) \otimes U(1) \otimes SU(3) \to SU(3) \otimes U(1). \qquad (2.2)$$

The first stage is achieved with a Higgs field transforming as the $\underset{\sim}{45}$ adjoint representation, while the $W^{\pm}$,Z and the fermions are getting their masses from Higgs fields transforming as $\underset{\sim}{10}$, $\underset{\sim}{120}$ and $\underset{\sim}{126}$ representations. In the symmetry limit the most successful mass relation is

$$\frac{m_d}{m_e} = \frac{m_s}{m_\mu} = \frac{m_b}{m_\tau} \simeq 2 - 3. \qquad (2.3)$$

Because SU(5) is naturally embedded in SO(10), after renormalization we get for the Weinberg angle $\sin^2\theta w \simeq 0.20$. Georgi and Nanopoulos [39], modifying the gauge hierarchy structure in SO(10), improved the value of the Weinberg angle and they obtained $\sin^2\theta w \simeq 0.23$. The prediction for the proton lifetime is the same as in an SU(5) theory.

Fritzsch [41], working with six quark flavours, in a $SU(2)_L \otimes SU(2)_R$ theory, shows that, if the mass matrix of the up quarks takes the form

$$\overline{(u\ c\ t)}_L\, M_{q_{2/3}} \begin{pmatrix} u \\ c \\ t \end{pmatrix}_R + h.c \qquad (2.4)$$

with $M_{q_{2/3}}$ having the form

$$M_{q_{2/3}} = \begin{pmatrix} 0 & A & 0 \\ A & 0 & B \\ 0 & B & C \end{pmatrix}, \qquad (2.5)$$

and a similar form for the down quark mass matrix $M_{q_{-1/3}}$, then we can find relations giving the Cabibbo-like angles in terms of quark mass ratio. Intuitively, such a mass matrix can be interpreted to arise as follows: the masses of the heavy quarks t,b are introduced initially and subsequently the masses of the 'lighter' quarks u,d,c,s are generated by a weak interaction mixing (radiative corrections) proceeding like a cascade. First $m_c(m_s)$ is generated via the mixing described by the parameter B and after that $m_u(m_d)$ is generated via the mixing described by the parameter A.

A mass matrix of the form (2.5) has been constructed in the SO(10) theory [39,42] and the mass of the top quark is predicted to be in the range 13-14 GeV. Unfortunately, this value for the mass of the top quark has been ruled out by recent experiments.

(c) E_6 model

The E_6 model was first proposed by Gürsey, Ramond and Sikivie [43], and subsequently analyzed by Achiman and Stech [44], Ruegg and Schücker [45], Barbieri and Nanopoulos [46].

E_6 is one of the exceptional groups and has rank 6 with 78 generators, most of which have to be broken at super large mass, leaving unbroken the 12 generators of $SU(3)_C \otimes SU(2) \otimes U(1)$. The fermions are assigned to the $\underset{\sim}{27}$ representation, so the Higgs fields must transform according to one or more irreducible components of

$$\underset{\sim}{27} \otimes \underset{\sim}{27} = \underset{\sim}{27} \oplus \underset{\sim}{351} \oplus \underset{\sim}{351}'. \tag{3.1}$$

The $\underset{\sim}{27}$ representation under the maximal subgroup $SO(10) \otimes U(1)$ decomposes as follows

$$\underset{\sim}{27} = \underset{\sim}{16} + \underset{\sim}{10} + \underset{\sim}{1}. \tag{3.2}$$

From (3.2) we can see that, in addition to the fermion fields of the SO(10) model, the $\underset{\sim}{27}$ of E_6 includes more neutral fermions. In reference [46] the possibility of obtaining the SU(5) relations $m_b = m_\tau$,

$m_s = m_\mu$, $m_d = m_e$ is discussed and a prediction of the mass of top quark is given $m_t \approx 20$ GeV.

PART II: Mathematical Formulation of Models of Grand Unification

Let us suppose that a model of grand unification has as a gauge group structure the Lie group G. The physics of this model is described by a Lagrangian $\mathcal{L}$, whose interaction terms can be written as follows

$$\mathcal{L} = \mathcal{L}_{fA} + \mathcal{L}_{\phi A} + \mathcal{L}_{f\phi} + V(\phi).$$

In the above expression, $\mathcal{L}_{fA}$ represents the couplings of fermions to the vector bosons; the term $\mathcal{L}_{\phi A}$ represents the couplings of the spinless meson fields to the vector bosons, and when the symmetry breaking is considered, it gives masses to the physical gauge bosons except the gluons and the photon; the term $\mathcal{L}_{f\phi}$ is the Yukawa interaction term responsible for the fermion masses; finally the potential $V(\phi)$ after minimization determines the pattern of the symmetry breakdown.

The fermion fields f transform according to some irreducible representation of the Lie group G, while the vector bosons always transform according to the adjoint representation. To have a Yukawa term invariant under the gauge group G, the spinless meson fields must transform according to an irreducible representation appearing in the tensor product of $f \otimes \overline{f}$. The explicit form of the potential $V(\phi)$ depends upon the transformation properties of the spinless meson fields ϕ.

Thus, to be able to have a complete group description of the model, we need the knowledge of:

(a) an explicit matrix realization of the irreducible representations under which the fermion fields are transformed;

(b) a matrix realization of the adjoint representation of the gauge group G;

(c) the knowledge of the Clebsch-Gordan series of the tensor product $f \otimes \bar{f}$;

(d) the Clebsch-Gordan coefficients of the tensor product $f \otimes \bar{f}$; and

(e) a matrix representing the ϕ fields.

As we discussed in Part I, the rank of the gauge group G must be at least four. For Lie groups G with rank four or greater, the problem of constructing matrix realizations of irreducible representations and evaluating Clebsch-Gordan coefficients will be considered and analyzed in the following chapters.

In Figure 3.1 we give the way we shall approach the problem using basic concepts of the Lie algebra and its representation theory.

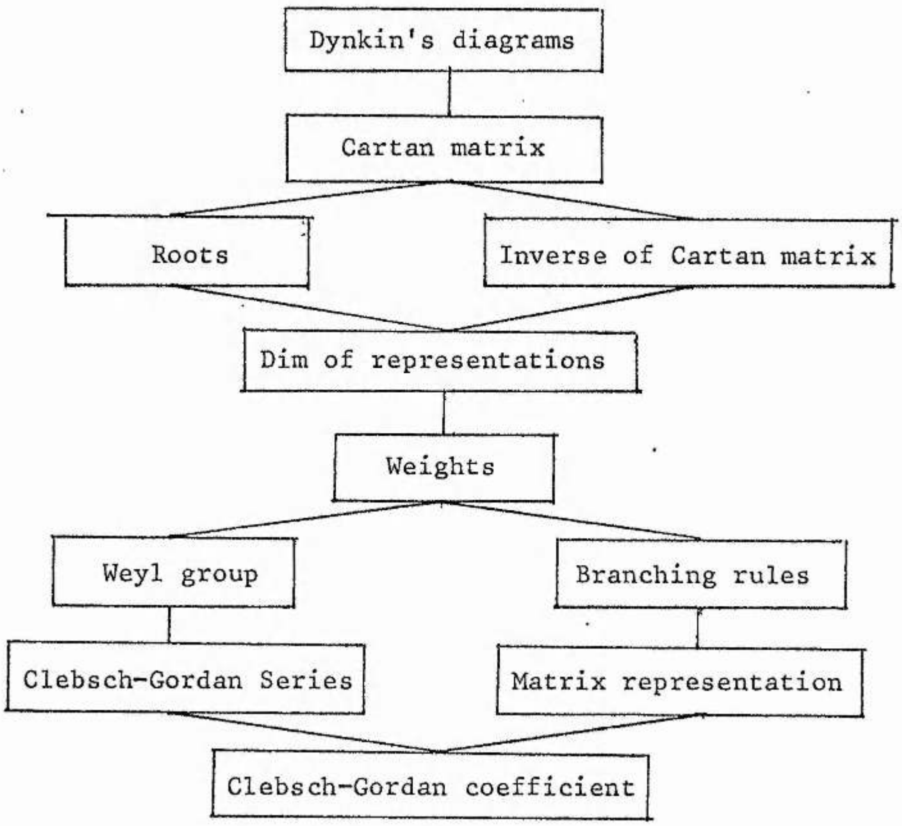


Figure 3.1

CHAPTER 4

CONSTRUCTION METHODS IN LIE GROUPS AND LIE ALGEBRAS

The purpose of this chapter is to show that, starting from the Dynkin diagrams and following the successive steps indicated in Figure 3.1, it is possible to obtain all the information needed for a group theoretical construction of a grand unified model. Moreover, it will be shown that all the construction methods can have a computer implementation.

The first explicit calculations in Lie algebras were made by hand by Konuma, Shima and Wada [47], who studied the classical Lie algebras up to rank three. A survey of Lie algebra computational methods was also given by Behrends et al [18]. With the introduction of computers in science, various people have implemented standard algorithms [48] to generate the weights [49], the Weyl group [50] and to calculate branching rules [51]. The need for the exceptional groups in the theory of grand unification resulted in a more detailed study of these groups [52,53].

The chapter is organized as follows. In §4.1 we give the basic concepts of the structure theory. The representation theory is discussed in §4.2. We analyse the methods of finding the Clebsch-Gordan series in §4.3. In §4.4 the method of an explicit matrix realization of irreducible representations is discussed, and finally in §4.5 we show how to evaluate the Clebsch-Gordan coefficients. In Appendix C all the programs are given and some technical information about them.

§4.1 Root Systems

The roots of a simple or semi-simple Lie algebra play a central role in the structure theory of the Lie algebras. The knowledge of the root system enables us not only to understand the structure of the Lie algebras, but also becomes a great help, after defining an appropriate basis of the algebra, when we want to classify them.

Instead of adopting an axiomatic approach to the construction theory and specially to the root systems, we develop the theory by carefully defining the terms we are using and giving the theorems that prove the properties of the roots. This approach has the advantage of giving us a deeper understanding of the construction theory and at the same time establishes a consistent notation, which is a serious problem in the literature of Lie algebras and representation theory. Moreover we follow this approach throughout this chapter. We are not giving the proofs of the quoted theorems, which can be found in any treatment of Lie algebras [54,55,56].

This section is structured as follows. The basic concepts of the structure theory are defined in §4.1.1. In §4.1.2 we describe the root program, and finally in §4.1.3 the root systems of all classical or exceptional simple Lie algebras are given.

§4.1.1 Basic concepts of the structure theory

To define the root system of a semi-simple complex Lie algebra $\mathcal{L}$, we need the notions of the adjoint representation of $\mathcal{L}$, the Cartan subalgebra H, and the rank ℓ of the algebra.

Definition: Adjoint representation

Let $\mathcal{L}$ be a semi-simple complex Lie algebra of dimension n, and let $a_1, a_2, \ldots, a_n$ be a basis for $\mathcal{L}$. The set of matrices $\underset{\sim}{ad}(a)$ of an n-dimensional representation of $\mathcal{L}$, defined by

$$[a,a_j] = \Sigma_{k=1}^{n} \{\underset{\sim}{ad}(a)\}_{kj} a_k \qquad (1.1.1)$$

for $j = 1,2,\ldots,n$ and any $a \in \mathcal{L}$, is called the <u>adjoint representation</u> of $\mathcal{L}$.

<u>Definition:</u> <u>Cartan subalgebra</u>

A 'Cartan subalgebra' H of a semi-simple complex Lie algebra $\mathcal{L}$ is a subalgebra of $\mathcal{L}$ with the following two properties:

(a) H is a maximal Abelian subalgebra of $\mathcal{L}$,

(b) $\underset{\sim}{ad}(h)$ is completely reducible for every $h \in H$.

<u>Definition:</u> <u>The rank of a semi-simple complex Lie algebra</u>

The 'rank' ℓ of a semi-simple complex Lie algebra $\mathcal{L}$ is defined to be the dimension of its Cartan subalgebra.

Now let $h_1,h_2,\ldots,h_\ell$ be a basis of a Cartan subalgebra H of a semi-simple complex Lie algebra $\mathcal{L}$, of rank ℓ and dimension n. Then, as H is Abelian, the irreducible representations of H are all one-dimensional. Consequently the matrices $\underset{\sim}{ad}(h_j)$ for $j = 1,2,\ldots,\ell$ must not only be diagonalizable but must be simultaneously diagonalizable. As a similarity transformation applied to $\underset{\sim}{ad}$ corresponds to a change of basis of $\mathcal{L}$, there exists a basis $h_1,h_2,\ldots,h_\ell,a_1',a_2',\ldots,a_{n-\ell}'$ of $\mathcal{L}$ such that

$$[h_j,a_k'] = a_k(h_j)a_k' , \qquad (1.1.2)$$

for $j = 1,2,\ldots,\ell$ and $k = 1,2,\ldots,n-\ell$, where $a_k(h_j)$ are a set of complex numbers. As H is Abelian

$$[h_j,h_k] = 0 , \qquad (1.1.3)$$

for $j,k = 1,2,\ldots,\ell$. Moreover, as H is a maximal Abelian subalgebra of $\mathcal{L}$, for each $k = 1,2,\ldots,n-\ell$ there must exist at least one j $(= 1,2,\ldots,\ell)$ such that $a_k(h_j) \neq 0$.

Now let $h = \Sigma_{j=1}^{\ell} \mu_j h_j$ be any element of H, and for each $k = 1,2,\ldots,n-\ell$ define a linear functional α_k on H by

$$\alpha_k(h) = \sum_{j=1}^{\ell} \mu_j \alpha_k(h_j) \ . \tag{1.1.4}$$

Here $\mu_1, \mu_2, \ldots, \mu_\ell$ are arbitrary complex numbers. Then for all $h \in H$ and for each $k = 1,2,\ldots,n-\ell$, the linear functional α_k is not identically zero (ie for some $h \in H$ $\alpha_k(h) \neq 0$) and

$$[h, a'_k] = \alpha_k(h) a'_k \ . \tag{1.1.5}$$

Each such linear functional is called a <u>non-zero root</u> of $\mathcal{L}$.

For any non-zero root α of $\mathcal{L}$ the set of elements $a_\alpha \in \mathcal{L}$ such that

$$[h, a_\alpha] = \alpha(h) a_\alpha \tag{1.1.6}$$

form a subspace of $\mathcal{L}$ which will be denoted by $\mathcal{L}_\alpha$, and will be called the <u>root subspace</u> corresponding to α. Then $\mathcal{L}$ is a vector space direct sum of H and all the root subspaces $\mathcal{L}_\alpha$ corresponding to non-zero roots. As $[h,h'] = 0$ for all $h,h' \in H$, we can make the identification $H = \mathcal{L}_0$. Then $\mathcal{L}$ can be represented by

$$\mathcal{L} = \mathcal{L}_0 \oplus \mathcal{L}_\alpha \oplus \mathcal{L}_\beta \oplus \mathcal{L}_\gamma \oplus \ldots \tag{1.1.7}$$

where $\alpha,\beta,\gamma,\ldots \in \Delta$, where Δ denotes the set of all distinct non-zero roots.

Equation (1.1.6) is an important equation in the construction theory.

We shall next give a series of theorems which demonstrate the properties of the non-zero roots.

<u>Theorem 4.1</u>

If $a_\alpha \in \mathcal{L}_\alpha$ and $a_\beta \in \mathcal{L}_\beta$, then $[a_\alpha, a_\beta] \in \mathcal{L}_{\alpha+\beta}$ if $\alpha + \beta \in \Delta$, but $[a_\alpha, a_\beta] = 0$ if $\alpha + \beta \notin \Delta$.

<u>Theorem 4.2</u>

If $\alpha \in \Delta$, then $-\alpha \in \Delta$.

Definition: The Killing form

The Killing form $B(a,b)$ corresponding to any two elements a and b of a Lie algebra $\mathcal{L}$ is defined by

$$B(a,b) = \mathrm{tr}\{\underset{\sim}{ad}(a)\underset{\sim}{ad}(b)\}. \tag{1.1.8}$$

Theorem 4.3

The Killing form is a symmetric bilinear form, ie

(a) $B(a,b) = B(b,a)$ for all $a,b \in \mathcal{L}$,

(b) $B(\alpha a,\beta b) = \alpha\beta B(a,b)$ for all $a,b \in \mathcal{L}$, α and $\beta \in \mathbb{C}$,

(c) $B(a,b+c) = B(a,b) + B(a,c)$ for all $a,b,c \in \mathcal{L}$,

(d) if ψ is any automorphism of $\mathcal{L}$, $B(\psi(a),\psi(b)) = B(a,b)$ for all $a,b \in \mathcal{L}$,

(e) $B([a,b],c) = B(a,[b,c])$ for all $a,b,c \in \mathcal{L}$, and finally

(f) if $\mathcal{L}'$ is an invariant subalgebra of $\mathcal{L}$, and $B_{\mathcal{L}'}$ denotes the Killing form of $\mathcal{L}'$ considered as a Lie algebra in its own right, then $B(a,b) = B_{\mathcal{L}'}(a,b)$ for all $a,b \in \mathcal{L}'$.

It is possible to associate with every linear functional on H, and in particular with each root $\alpha \in \Delta$, a unique element h_α of H by the definition

$$B(h_\alpha,h) = \alpha(h), \tag{1.1.9}$$

for all $h \in H$. We shall need the elements h_α to define later the canonical basis of $\mathcal{L}$. Then, from (1.1.9), we have

$$B(h_\alpha,h_\beta) = \alpha(h_\beta) = \beta(h_\alpha), \tag{1.1.10}$$

as B is symmetric (Theorem 4.3).

We introduce the notation

$$\alpha(h_\beta) = \beta(h_\alpha) = \langle\alpha,\beta\rangle. \tag{1.1.11}$$

Equation (1.1.6) can be written in the notation (1.1.10):

$$[h_\beta,a_\alpha] = \langle\alpha,\beta\rangle a_\alpha, \tag{1.1.12}$$

for all $\alpha,\beta \in \Delta$.

Theorem 4.4

If $a_\alpha \in \mathcal{L}_\alpha$ and $a_{-\alpha} \in \mathcal{L}_{-\alpha}$ then

$$[a_\alpha, a_{-\alpha}] = B(a_\alpha, a_{-\alpha})h_\alpha. \qquad (1.1.13)$$

The nature of the number $B(a_\alpha, a_{-\alpha})$ is given by the following theorem.

Theorem 4.5

For each $\alpha \in \Delta$ and every $a_\alpha \in \mathcal{L}_\alpha$ there exists an element $a_{-\alpha}$ of $\mathcal{L}_{-\alpha}$ such that $B(a_\alpha, a_{-\alpha}) \neq 0$.

Theorem 4.6

For every $\alpha, \beta \in \Delta$, the quantities $\langle\alpha,\beta\rangle$ are real and rational. Moreover for every $\alpha \in \Delta$, $\langle\alpha,\alpha\rangle$ is positive.

This theorem has a very important consequence in the classification of simple Lie algebras, because it specifies the entries of the Cartan matrix. The magnitudes of these entries will be given by Theorem 4.11.

Theorem 4.7

$\mathbb{H}$ coincides with the subspace of $\mathcal{L}$ consisting of all elements of the form $\Sigma_{\alpha\in\Delta}\mu_\alpha h_\alpha$, where the μ_α take all complex values.

This theorem implies that from the set of elements h_α ($\alpha \in \Delta$) a subset of ℓ linearly-independent elements may be selected and may be taken to form a basis for H. The elements of this set will be denoted by $h_{\beta_1}, h_{\beta_2}, \ldots, h_{\beta_\ell}$ ($\beta_1, \beta_2, \ldots, \beta_\ell \in \Delta$).

Theorem 4.8

Every non-zero root α of Δ can be written in the form

$$\alpha = \sum_{j=1}^{\ell} k_j \beta_j,$$

where the coefficients $k_1, k_2, \ldots, k_\ell$ are all real and rational.

Theorem 4.9

If $\alpha \in \Delta$, then $\dim \mathcal{L}_\alpha = 1$ and $k\alpha \in \Delta$ only if $k = 1$ or -1.

To develop an algorithm for generation of the roots of a simple Lie algebra we need the notion of a 'string' of roots.

Definition: The α-string of roots containing β

Suppose that $\alpha,\beta \in \Delta$. Then the α-string of roots containing β is the set of all roots of the form $\beta + k\alpha$, where k is an integer.

Theorem 4.10

Let $\alpha,\beta \in \Delta$. Then there exist two non-negative integers p and q (which depend on α and β) such that $\beta + k\alpha$ is in the α-string containing β for every integer k that satisfies the relation $-p \leq k \leq q$. Moreover p and q are such that

$$p - q = 2\langle\beta,\alpha\rangle/\langle\alpha,\alpha\rangle. \qquad (1.1.14)$$

Also

$$\beta - \{2\langle\beta,\alpha\rangle/\langle\alpha,\alpha\rangle\}\alpha \qquad (1.1.15)$$

is a non-zero root.

The algorithm for generation of the roots is mainly based on the equations (1.1.14) and (1.1.15).

Finally, we give the last theorem of this subsection which specifies the magnitudes of the entries in the Cartan matrix.

Theorem 4.11

For all $\alpha,\beta \in \Delta$, $2\langle\beta,\alpha\rangle/\langle\alpha,\alpha\rangle$ can take only the integral values 0, ±1, ±2 or ±3 (the quantities $2\langle\beta,\alpha\rangle/\langle\alpha,\alpha\rangle$ are called the Cartan integers).

§4.1.2 The algorithm for the generation of positive roots of any simple Lie algebra

Having defined the roots and developed the structure theory to some extent, we shall now show that the knowledge of the 'positive' roots is sufficient to specify the properties of a particular Lie algebra. Having done that, we shall describe the algorithm, for generation of these 'positive' roots.

As we have seen in Theorem 4.8, every non-zero root α of Δ can be written in the form

$$\alpha = \Sigma_{j=1}^{\ell} k_j \beta_j, \qquad (1.2.1)$$

with the coefficients $k_1, k_2, \ldots, k_\ell$ all real and rational.

Definition: Positive root

A non-zero root α of Δ is said to be positive (with respect to the basis $\beta_1, \beta_2, \ldots, \beta_\ell$) if the first non-vanishing coefficient of the set $k_1, k_2, \ldots, k_\ell$ appearing in (1.2.1) is positive.

Definition: Lexicographic ordering of roots

Let α and β be any two roots of Δ. Then, if $(\alpha-\beta) > 0$, one says that $\alpha > \beta$.

Definition: Simple root of Δ

A non-zero root α of Δ is said to be simple if α is positive but α cannot be expressed in the form $\alpha = \beta + \gamma$, where β and γ are both positive roots of Δ.

Theorem 4.12

If α and β are two simple roots of Δ, and $\alpha \neq \beta$, then

(a) $\alpha - \beta$ is not a root of Δ,

(b) $\langle\alpha,\beta\rangle \leq 0$.

Theorem 4.13

If $\mathcal{L}$ has rank ℓ, then $\mathcal{L}$ possesses precisely ℓ simple roots $\alpha_1, \ldots, \alpha_\ell$. They form a basis forthe dual spase $\dot{H}^*$. Moreover, if α is any positive root of Δ then

$$\alpha = \Sigma_{j=1}^{\ell} k_j \alpha_j,$$

where $k_1, k_2, \ldots, k_\ell$ are a set of non-negative integers.

Definition: Cartan matrix $\underset{\sim}{A}$

The Cartan matrix $\underset{\sim}{A}$ of $\mathcal{L}$ is a $\ell \times \ell$ matrix whose elements A_{jk} are defined in terms of the simple roots $\alpha_1, \alpha_2, \ldots, \alpha_\ell$ of $\mathcal{L}$ by

$$A_{jk} = \frac{2\langle\alpha_j,\alpha_k\rangle}{\langle\alpha_j,\alpha_j\rangle},\ j,k = 1,2,\ldots,\ell. \tag{1.2.3}$$

In Appendix B the Cartan matrices of all simple Lie algebras are given, and the values of $\langle\alpha_j,\alpha_k\rangle$ are listed.

Now, we are ready to give a simple algorithm, for the generation of the positive roots.

First we define the 'level' of an arbitrary positive root $\alpha = \Sigma_{j=1}^{\ell} k_j\alpha_j$ to be $\Sigma_{j=1}^{\ell} k_j$, which is a positive integer by Theorem 4.13. Then, the positive roots of level 1 are just the simple roots. Suppose that for some k ($\geq$ 1) all the positive roots of level k and below are known which is certainly true for k = 1. Suppose β is a level (k+1) root. Then, from the above theorem it can be shown that there exists a simple root α_j such that $\langle\beta,\alpha_j\rangle > 0$. Consider the α_j-string containing β. By Theorem 4.10 this consists of $\beta + r\alpha_j$ for all integers r satisfying $-p \leq r \leq q$, where $p \geq 0$ and $q \geq 0$ and

$$p - q = 2\langle\beta,\alpha_j\rangle/\langle\alpha_j,\alpha_j\rangle.$$

But with $\langle\beta,\alpha_j\rangle > 0$, this implies $p > 0$, so $\beta - \alpha_j$ must be a root of Δ. Clearly $\beta - \alpha_j$ is of level k. Thus, to determine the roots of level (k+1) from those of level k, one must decide for each root α of level k and each simple root α_j, whether $\alpha + \alpha_j$ is a root. But $\alpha + \alpha_j$ is a root if the α_j-string containing α is $\alpha + r\alpha_j$ with $-p \leq r \leq q$ and q positive. But p is known because $\alpha + r\alpha_j$ with $r \leq 0$ are all roots of level k or below, which are assumed to be known. Then, as

$$q = p - 2\langle\alpha,\alpha_j\rangle/\langle\alpha_j,\alpha_j\rangle,$$

and as $\alpha = \Sigma_{i=1}^{\ell} k_i\alpha_i$ (where the integers $k_1,k_2,\ldots,k_\ell$ are all known) and as

$$2\langle\alpha,\alpha_j\rangle/\langle\alpha_j,\alpha_j\rangle = \Sigma_{i=1}^{\ell} k_i A_{ji},$$

q can be determined.

The above algorithm can be easily implemented in the form of a computer program (Program A1(1)).

Description of Program A1(1)

The program has as input the number of positive roots; the rank of the algebra; the simple roots, and the Cartan matrix.

As an output we get the positive roots listed one below the other. Each root is represented by the integers $k_1, k_2, \ldots k_\ell$ of (1.2.1). The program starts with the simple roots of level $k = 1$. In each subsequent level, it calculates the integer q. The root that comes out from the string of roots is tested to see whether it has already been calculated previously. If not, it is then placed in the array of the positive roots. The program ends when the given number of positive roots is reached.

In Table 4.1, we list the number of positive roots for each algebra, which is needed for the above program.

Table 4.1

Type of algebra	Number of positive roots
A_ℓ	$\frac{1}{2}\ell(\ell+1)$
B_ℓ, C_ℓ	ℓ^2
D_ℓ	$\ell(\ell-1)$
E_6	36
E_7	63
E_8	120
F_4	24
G_2	6

E_6

1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1
1	1	0	0	0	0
0	1	1	0	0	0
0	0	1	1	0	0
0	0	1	0	0	1
0	0	0	1	1	0
1	1	1	0	0	0
0	1	1	1	0	0
0	1	1	0	0	1

0	0	1	1	1	0
0	0	1	1	0	1
1	1	1	1	0	0
1	1	1	0	0	1
0	1	1	1	1	0
0	1	1	1	0	1
0	0	1	1	1	1
1	1	1	1	1	0
1	1	1	1	0	1
0	1	1	1	1	1
0	1	2	1	0	1
1	1	1	1	1	1
1	1	2	1	0	1
0	1	2	1	1	1
1	1	2	1	1	1
1	2	2	1	0	1
0	1	2	2	1	1
1	2	2	1	1	1
1	1	2	2	1	1
1	2	2	2	1	1
1	2	3	2	1	1
1	2	3	2	1	2

E_7

1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	1
1	1	0	0	0	0	0
0	1	1	0	0	0	0
0	0	1	1	0	0	0

0	0	1	0	0	0	1
0	0	0	1	1	0	0
0	0	0	0	1	1	0
1	1	1	0	0	0	0
0	1	1	1	0	0	0
0	1	1	0	0	0	1
0	0	1	1	1	0	0
0	0	1	1	0	0	1
0	0	0	1	1	1	0
1	1	1	1	0	0	0
1	1	1	0	0	0	1
0	1	1	1	1	0	0
0	1	1	1	0	0	1
0	0	1	1	1	1	0
0	0	1	1	1	0	1
1	1	1	1	1	0	0
1	1	1	1	0	0	1
0	1	1	1	1	1	0
0	1	1	1	1	0	1
0	1	2	1	0	0	1
0	0	1	1	1	1	1
1	1	1	1	1	1	0
1	1	1	1	1	0	1
1	1	2	1	0	0	1
0	1	1	1	1	1	1
0	1	2	1	1	0	1
1	1	1	1	1	1	1
1	1	2	1	1	0	1
1	2	2	1	0	0	1
0	1	2	1	1	1	1

0	1	2	2	1	0	1
1	1	2	1	1	1	1
1	2	2	1	1	0	1
1	1	2	2	1	0	1
0	1	2	2	1	1	1
1	2	2	1	1	1	1
1	1	2	2	1	1	1
1	2	2	2	1	0	1
0	1	2	2	2	1	1
1	2	2	2	1	1	1
1	1	2	2	2	1	1
1	2	3	2	1	0	1
1	2	3	2	1	1	1
1	2	2	2	2	1	1

1	2	3	2	1	0	2
1	2	3	2	2	1	1
1	2	3	2	1	1	2
1	2	3	3	2	1	1
1	2	3	2	2	1	2
1	2	3	3	2	1	2
1	2	4	3	2	1	2
1	3	4	3	2	1	2
2	3	4	3	2	1	2

E_8

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	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	1	0
0	0	0	0	0	0	0	1
1	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	0	0	0	0	1
0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0
0	0	0	0	0	1	1	0
1	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0
0	1	1	0	0	0	0	1
0	0	1	1	1	0	0	0
0	0	1	1	0	0	0	1
0	0	0	1	1	1	0	0
0	0	0	0	1	1	1	0
1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	1
0	1	1	1	1	0	0	0
0	1	1	1	0	0	0	1
0	0	1	1	1	1	0	0
0	0	1	1	1	0	0	1
0	0	0	1	1	1	1	0
1	1	1	1	1	0	0	0
1	1	1	1	0	0	0	1
0	1	1	1	1	1	0	0
0	1	1	1	1	0	0	1
0	1	2	1	0	0	0	1
0	0	1	1	1	1	1	0
0	0	1	1	1	1	0	1
1	1	1	1	1	1	0	0
1	1	1	1	1	0	0	1
1	1	2	1	0	0	0	1
0	1	1	1	1	1	1	0
0	1	1	1	1	1	0	1
0	1	2	1	1	0	0	1
0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	0
1	1	1	1	1	1	0	1
1	1	2	1	1	0	0	1
1	2	2	1	0	0	0	1
0	1	1	1	1	1	1	1
0	1	2	1	1	1	0	1
0	1	2	2	1	0	0	1

1	1	1	1	1	1	1	1
1	1	2	1	1	1	0	1
1	2	2	1	1	0	0	1
1	1	2	2	1	0	0	1
0	1	2	1	1	1	1	1
0	1	2	2	1	1	0	1
1	1	2	1	1	1	1	1
1	2	2	1	1	1	0	1
1	1	2	2	1	1	0	1
1	2	2	2	1	0	0	1
0	1	2	2	1	1	1	1
0	1	2	2	2	1	0	1
1	2	2	1	1	1	1	1
1	1	2	2	1	1	1	1
1	2	2	2	1	1	0	1
1	1	2	2	2	1	0	1
1	2	3	2	1	0	0	1
0	1	2	2	2	1	1	1
1	2	2	2	1	1	1	1
1	1	2	2	2	1	1	1
1	2	3	2	1	1	0	1
1	2	2	2	2	1	0	1
1	2	3	2	1	0	0	2
0	1	2	2	2	2	1	1
1	2	3	2	1	1	1	1
1	2	2	2	2	1	1	1
1	1	2	2	2	2	1	1
1	2	3	2	2	1	0	1
1	2	3	2	1	1	0	2
1	2	3	2	2	1	1	1
1	2	3	2	1	1	1	2
1	2	2	2	2	2	1	1
1	2	3	3	2	1	0	1
1	2	3	2	2	1	0	2
1	2	3	3	2	1	1	1
1	2	3	2	2	2	1	1
1	2	3	2	2	1	1	2
1	2	3	3	2	1	0	2
1	2	3	3	2	2	1	1
1	2	3	3	2	1	1	2
1	2	3	2	2	2	1	2
1	2	4	3	2	1	0	2
1	2	3	3	3	2	1	1
1	2	3	3	2	2	1	2

1	2	[illegible]	3	2	1	1	2
1	3	4	3	2	1	0	2
1	2	3	3	3	2	1	2
1	2	4	3	2	2	1	2
1	3	4	3	2	1	1	2
2	3	4	3	2	1	0	2
1	2	4	3	3	2	1	2
1	3	4	3	2	2	1	2
2	3	4	3	2	1	1	2
1	3	4	3	3	2	1	2
1	2	4	4	3	2	1	2
2	3	4	3	2	2	1	2
2	3	4	3	3	2	1	2
1	3	4	4	3	2	1	2
2	3	4	4	3	2	1	2
1	3	5	4	3	2	1	2
2	3	5	4	3	2	1	2
1	3	5	4	3	2	1	3
2	4	5	4	3	2	1	2
2	3	5	4	3	2	1	3
2	4	5	4	3	2	1	3
2	4	6	4	3	2	1	3
2	4	6	5	3	2	1	3
2	4	6	5	4	2	1	3
2	4	6	5	4	3	1	3
2	4	6	5	4	3	2	3

F_4

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	1	0	0
0	1	1	0
0	1	2	0
0	0	1	1
1	1	1	0
1	1	2	0
0	1	1	1
0	1	2	1
0	1	2	2
1	1	1	1
1	2	2	0
1	1	2	1
1	1	2	2
1	2	2	1
1	2	2	2
1	2	3	1
1	2	3	2
1	2	4	2
1	3	4	2
2	3	4	2

G_2

1	0
0	1
1	1
1	2
1	3
2	3

Table 4.2: The root systems of the exceptional Lie algebras

§4.1.3 Root systems of exceptional and classical Lie algebras

I Exceptional Lie algebras

In Table 4.2 all the roots of the exceptional Lie algebras are listed.

II Classical Lie algebras

We shall give the general formulae for the root systems of the classical Lie algebras.

(a) A_ℓ

A_ℓ has $\frac{1}{2}\ell(\ell+1)$ positive roots, namely:

$$\Sigma_{p=j}^{k}\alpha_p$$

for all $j,k = 1,2,\ldots,\ell$ with $j \leq k$.

(b) B_ℓ

B_ℓ has ℓ^2 positive roots, namely:

$$\left.\begin{array}{ll} \Sigma_{p=j}^{\ell}\alpha_p & , \; j = 1,2,\ldots,\ell; \\ \Sigma_{p=j}^{k-1}\alpha_p + 2\Sigma_{p=k}^{\ell}\alpha_p & , \; j,k = 1,2,\ldots,\ell; \; j < k; \\ \Sigma_{p=j}^{k-1}\alpha_p & , \; j,k = 1,2,\ldots,\ell; \; j < k; \end{array}\right\}$$

(c) C_ℓ

C_ℓ has ℓ^2 positive roots, namely:

$$\left.\begin{array}{ll} \Sigma_{p=j}^{k-1}\alpha_p & , \; j,k = 1,2,\ldots,\ell; \; j < k; \\ \Sigma_{p=j}^{k-1}\alpha_p + 2\Sigma_{p=k}^{\ell-1}\alpha_p + \alpha_\ell & , \; j,k = 1,2,\ldots,\ell-1; \; j < k; \\ \Sigma_{p=j}^{\ell-1}\alpha_p + \alpha_\ell & , \; j = 1,2,\ldots,\ell-1; \\ 2\Sigma_{p=j}^{\ell-1}\alpha_p + \alpha_\ell & , \; j = 1,2,\ldots,\ell-1; \\ \alpha_\ell \, . & \end{array}\right\}$$

(d) D_ℓ

D_ℓ has $\ell(\ell-1)$ positive roots, namely:

$$\left.\begin{array}{l} \Sigma_{p=j}^{k-1}\alpha_p + 2\Sigma_{p=k}^{\ell-2}\alpha_p + \alpha_{\ell-1} + \alpha_\ell \\ \Sigma_{p=j}^{k-1}\alpha_p \end{array}\right\} j,k = 1,2,\ldots,\ell-2;\ j < k\ (\ell \geq 3)$$

$$\left.\begin{array}{l} \Sigma_{p=j}^{\ell-2}\alpha_p + \alpha_{\ell-1} + \alpha_\ell \\ \Sigma_{p=j}^{\ell-2}\alpha_p + \alpha_{\ell-1} \\ \Sigma_{p=j}^{\ell-2}\alpha_p + \alpha_\ell \\ \Sigma_{p=j}^{\ell-2}\alpha_p \end{array}\right\} j = 1,2,\ldots,\ell-2\quad (\ell \geq 3).$$

§4.2 Weight Systems

The knowledge of the weights of a particular irreducible representation is essential if we want to construct an explicit matrix realization of it. In this section we describe the method of generating the weight systems of irreducible representations.

As in §4.1, the development of the theory will be limited to the definitions of terms, and the quotation of the relevant theorems, whose statements give us the properties of the weight systems.

The section includes the introduction of some results on the representation theory (§4.2.1), the discussion of the weight algorithms (§4.2.2), and in §4.2.3 the weight systems of the representations $\underset{\sim}{126}$, $\underset{\sim}{120}$, $\underset{\sim}{16}$ and $\underset{\sim}{10}$ of D_5 and $\underset{\sim}{27}$, $\underset{\sim}{14}$ and $\underset{\sim}{7}$ of G_2.

§4.2.1 Basic concepts of representation theory

Consider a representation $\underset{\sim}{\Gamma}$ of dimension d of a semi-simple complex Lie algebra $\mathcal{L}$. Because $\underset{\sim}{\Gamma}$ provides a representation of the compact form $\mathcal{L}_C$ of $\mathcal{L}$, on $\mathcal{L}_C$ $\underset{\sim}{\Gamma}$ is equivalent to a representation by anti-Hermitian matrices. Moreover, if $h \in H$, then the matrices $\underset{\sim}{\Gamma}(h)$ may be diagonalized by the same similarity transformation. We can then assume that $\underset{\sim}{\Gamma}(h)$ is a diagonal matrix for each $h \in H$.

Consider the diagonal elements $\Gamma(h)_{jj}$ for some fixed j ($j = 1,2,\ldots,d$). As $\underset{\sim}{\Gamma}(ah+bh') = a\underset{\sim}{\Gamma}(h) + b\underset{\sim}{\Gamma}(h')$ for all $h,h' \in H$ and any complex numbers a and b, it follows that

$$\Gamma(ah+bh')_{jj} = a\Gamma(h)_{jj} + b\Gamma(h')_{jj}.$$

Thus the diagonal elements $\Gamma(h)_{jj}$ are linear functionals defined on H. These linear functionals are called the <u>weights</u> of the representation, so that a d-dimensional representation possesses d weights, some of which may be identical.

Suppose that $\psi_1,\psi_2,\ldots,\psi_d$ form a basis of the carrier space V of the representation $\underset{\sim}{\Gamma}$, and that W(a) is the operator defined for each $a \in \mathcal{L}$ by

$$W(a)\psi_j = \Sigma_{k=1}^{d}\Gamma(a)_{kj}\psi_k, \; j = 1,2,\ldots,d.$$

Then, for each $h \in H$, as $\underset{\sim}{\Gamma}(h)$ is diagonal

$$W(h)\psi_j = \Gamma(h)_{jj}\psi_j, \; j = 1,2,\ldots,d. \tag{2.1.1}$$

Thus for each $j = 1,2,\ldots,d$ and for all $h \in H$, $\Gamma(h)_{jj}$ is an eigenvalue of the operator W(h), the corresponding eigenvector being ψ_j.

Let $\lambda_j(h) = \Gamma(h)_{jj}$ define the weight $\underset{\sim}{\lambda}_j$ corresponding to the j^{th} position in the representation. Denoting the corresponding eigenvector ψ_j by $\psi(\lambda_j)$ (2.1.1) becomes

$$W(h)\psi(\lambda_j) = \lambda_j(h)\psi(\lambda_j), \; j = 1,2,\ldots,d. \tag{2.1.2}$$

If the weight λ appears $m(\lambda)$ times in the representation, $m(\lambda)$ is said to be the <u>multiplicity</u> of λ. If $m(\lambda) = 1$, then λ is called a <u>simple weight</u> of the representation. When we omit the index j then (2.1.2) can be written

$$W(h)\psi(\lambda) = \lambda(h)\psi(\lambda) \tag{2.1.3}$$

for all $h \in H$, $\psi(\lambda)$ being any eigenvector of $W(h)$ with eigenvalue $\lambda(h)$. The multiplicity $m(\lambda)$ is then the dimension of the subspace of V spanned by the eigenvectors $\psi(\lambda)$.

Theorem 4.14

If λ is a weight of a representation, then $\lambda + \alpha$ is also a weight of the same representation for each $\alpha \in \Delta$ such that $W(e_\alpha)\psi(\lambda) \neq 0$.

Note: We define the operator e_α in §4.4.1.

Theorem 4.15

For any weight λ of any representation of $\mathcal{L}$ and any root α of Δ, $2\langle\lambda,\alpha\rangle/\langle\alpha,\alpha\rangle$ is an integer.

Theorem 4.16

Every weight λ can be written in terms of the simple roots $\alpha_1, \alpha_2, \ldots, \alpha_\ell$ by

$$\lambda = \Sigma_{j=1}^{\ell} \mu_j \alpha_j \tag{2.1.4}$$

where the coefficients μ_j are all real and rational.

The concept of a 'string' can be extended from roots to weights, and produces a generalization of Theorem 4.10.

Definition: The α-string of weights containing λ

Suppose that α is a root of $\mathcal{L}$ and λ is a weight of some representation of $\mathcal{L}$. Then the α-string of weights containing λ is the set of all weights of that representation of the form $\lambda + k\alpha$, where k is an integer.

Theorem 4.17

Let α be a non-zero root of $\mathcal{L}$ and λ a weight of some representation of $\mathcal{L}$. Then there exist two non-negative integers p and q (which depend on α and λ) such that $\lambda + k\alpha$ is in the α-string containing λ for every integer k that satisfies the relations $-p \leq k \leq q$. Moreover, p and q are such that

$$p - q = 2\langle\lambda,\alpha\rangle/\langle\alpha,\alpha\rangle \qquad (2.1.5)$$

(for the proof, see Jacobson [54], p 220).

Each irreducible representation is uniquely and completely specified by its 'highest weight', all of its properties such as its dimension and the other weights being deducible from it.

Definition: Highest weight Λ of a representation

If Λ is a weight of a representation of $\mathcal{L}$ such that $\Lambda > \lambda$ for every other weight λ (the lexicograthic ordering being defined relative to the basis $\alpha_1,\ldots,\alpha_\ell$), then Λ is said to be the highest weight of the representation.

Theorem 4.18

If Λ is the highest weight of an irreducible representation of a semi-simple complex Lie algebra $\mathcal{L}$, then

(a) Λ is a simple weight (ie $m(\Lambda) = 1$);

(b) every other weight λ of the representation has the form

$$\lambda = \Lambda - \Sigma_{j=1}^{\ell} q_j\alpha_j \qquad (2.1.6)$$

where $q_1, q_2, \ldots, q_\ell$ are a set of non-negative integers.

Definition: Fundamental weights of a semi-simple complex Lie algebra

The ℓ fundamental weights $\Lambda_1,\Lambda_2,\ldots,\Lambda_\ell$ of $\mathcal{L}$, are the ℓ linear functionals on H defined by

$$\Lambda_j(h) = \Sigma_{k=1}^{\ell}(A^{-1})_{kj}\alpha_k(h) \qquad (2.1.1)$$

for all $h \in H$, where $\alpha_1,\alpha_2,\ldots,\alpha_\ell$ are the simple roots, and $\underset{\sim}{A}$ is the Cartan matrix.

Theorem 4.19

For every irreducible representation of a semi-simple complex

Lie algebra $\mathcal{L}$ the highest weight Λ can be written as

$$\Lambda = \sum_{j=1}^{\ell} n_j \Lambda_j \qquad (2.1.8)$$

where $\{n_1, n_2, \ldots, n_\ell\}$ is a set of non-negative integers and $\Lambda_1, \Lambda_2, \ldots, \Lambda_\ell$ are the fundamental weights of $\mathcal{L}$. Moreover, to every set of non-negative integers $\{n_1, n_2, \ldots, n_\ell\}$ there exists an irreducible representation of $\mathcal{L}$ with highest weight Λ given by (2.1.8), and this representation is unique up to equivalence.

The irreducible representation of $\mathcal{L}$ with highest weight Λ specified by (2.1.8) will be denoted by $D(\{n_1, n_2, \ldots, n_\ell\})$. Its dimension d is given by the following formula, known as Weyl's dimensionality formula

$$d = \Pi_{\alpha \in \Delta +} \{ \langle \Lambda + \delta, \alpha \rangle / \langle \delta, \alpha \rangle \}, \qquad (2.1.9)$$

where $\Delta^+ \subset \Delta$ denotes the set of all positive roots, and $\delta = \frac{1}{2} \Sigma_{\alpha \in \Delta +} \alpha$.

Now, we are ready to develop an algorithm for generation of all the weights, given the highest weight of a particular representation.

§4.2.2 Algorithms for generation of weights

Weyl's dimensionality formula (2.1.9) can be rewritten as

$$d = \Pi_{\alpha \in \Delta +} [\{ \Sigma_{j=1}^{\ell} n_j k_j^{\alpha} w_j / \Sigma_{j=1}^{\ell} k_j^{\alpha} w_j \} + 1], \qquad (2.2.1)$$

where $\alpha = \Sigma_{j=1}^{\ell} K_j^{\alpha} \alpha_j$, $\Lambda = \Sigma_{j=1}^{\ell} n_j \Lambda_j$, and w_j is the weighting factor of the Dynkin diagrams corresponding to α_j. (We give the Dynkin diagrams in Appendix B.) In that form, Weyl's formula can be easily implemented by a computer program (Program B1(2)).

Description of Program B1(2)

The program first calculates the positive roots (Program A1(1)). The call of the procedure PRODUCT, which represents Weyl's dimensionality formula (2.2.1), calculates the dimension d of the representations.

The input consists of:

the number of positive roots (Table 4.1); the rank of the algebra; the number of representations whose dimensions we want to find; an array of ℓ integer numbers, which represents the last representation $D\{(n_1,n_2,\ldots,n_\ell)\}$ to be calculated; the simple roots; the Cartan matrix; and finally the weighting factors of the relevant Dynkin diagram.

As output we get a list of representations of the number given as an input. The first ℓ columns are the integers $n_1,n_2,\ldots,n_\ell$ and in the $\ell + 1$ column we get the dimension d.

If the weights of an irreducible representation are simple, then Theorem 4.17 provides a simple algorithm for generating the weights. We have developed a program (Program B2(2)) for generating the weights of an irreducible representation using Theorem 4.17. The structure of this program is the same as the A1(1) program, for the generation of the roots. The program B2(2) cannot calculate the multiplicity of a weight, if this weight is not simple, but it helps us to predict the multiplicity of a weight, when the application of the Freudenthal recursive formula is 'run-time' expensive.

In the general case, when the weights are not simple, a method is required for evaluating the multiplicities. Freudenthal's recursion formula, which is stated in the next theorem, provides the necessary information.

Theorem 4.20

Consider an irreducible representation of $\mathcal{L}$ with highest weight Λ. Then the multiplicity $m(\lambda)$ of a possible weight $\lambda = \Lambda - \Sigma_{j=1}^{\ell} q_j \alpha_j$ (with $q_1, q_2, \ldots, q_\ell$ all non-negative integers) is given by

$$\{<\Lambda+\delta,\Lambda+\delta>-(\lambda+\delta,\lambda+\delta>\}m(\lambda) = 2\Sigma_{\alpha\in\Delta+}\sum_{k} m(\lambda+k\alpha)<\lambda+k\alpha,\alpha>, \qquad (2.2.2)$$

where the second sum on the right hand side is only over those values of k for which $\lambda + k\alpha$ is a weight of the representation whose level (defined in §4.1.2) is less than that of λ, and where $\delta = \frac{1}{2}\Sigma_{\alpha\in\Delta+}\alpha$. In particular, if $m(\lambda) = 0$, then λ is not a weight of the representation (for the proof see Jacobson [54], p 243).

There are two ways in which one can approach the problem of generation of the weight systems in the general case where $m(\lambda) \geq 1$ using a computer.

I Algorithm

We can use Theorem 4.17 to generate the weights without concerning ourselves with their multiplicity (Program B2(2)). If the number of weights is less than the dimension d of the representation (which means that some weights have $m(\lambda) > 1$) then we order them according to a lexicographical order (Theorem 4.19) and using Theorem 4.20 we find the missing multiplicities. Program B3(2) uses this algorithm.

II Algorithm

We can use directly the Freudenthal's recursive formula for generation of the weights. As $m(\Lambda) = 1$, the formula allows the multiplicities of the weights to be obtained first for level 1, then for level 2, and so on. For example, the weight of level 1 has the form $\lambda = \Lambda - \alpha_j$, where α_j is some simple root. The only non-zero term on the right hand side of (2.2.2) occurs with $\alpha = \alpha_j$ and $k = 1$, and is $2m(\Lambda)\langle\Lambda,\alpha_j\rangle = 2\langle\Lambda,\alpha_j\rangle$. Similarly, every level 2 weight multiplicity is given by (2.2.2) in terms of the multiplicities of weights 1 and 0, and so on. On the other hand, if the linear functional $\lambda = \Lambda - \Sigma_{j=1}^{\ell} q_j\alpha_j$ is not a weight, then (2.2.2) gives $m(\lambda) = 0$. Thus the formula provides a self-contained and exhaustive procedure for finding all the weights and their multiplicities by simply investigating every linear functional

$\Lambda - \Sigma_{j=1}^{\ell} q_j \alpha_j$ for every set of non-negative integers $\{q_1, q_2, \ldots, q_\ell\}$ for increasing values of $q' = \Sigma_{j=1}^{\ell} q_j$, stopping when the sum of the multiplicities reaches the value d. Program B4(2) uses this algorithm.

We shall next describe Programs B3(2) and B4(2), and discuss their applicability.

Description of Program B3(2)

The program consists of the following procedures:
procedure LHS which represents the left hand side of (2.2.2); procedure LEVL which finds the levels of the weights; procedure ORDER which orders the weights according to Theorem 4.19; and procedure FREUDENTHAL which finds the multiplicities.

The program starts with the calculation of the positive roots using A1(1) as a subprogram. The weights without multiplicity are next calculated using B2(2), and are ordered by call of the procedure ORDER. If the number of calculated weights is less than the dimension of the representation, then the call of the procedure LEVL specifies their level, and the call of the procedure FREUDENTHAL finds their multiplicity.

The input of the program consists of:
the dimension of the representation; the rank of the algebra; the number of positive roots; the length of the simple roots (the quantities $\langle\alpha_i, \alpha_i\rangle$, $i = 1,2,\ldots,\ell$); the simple roots; the Cartan matrix; the inverse of the Cartan matrix, and finally the highest weight of the representation in the notation $D\{(n_1, n_2, \ldots, n_\ell)\}$.

As output we get the weights listed one below the other in the form of (2.1.6).

Description of Program B4(2)

The program consists of the following procedure:
procedure LHS which, as in B3(2), calculates the left hand side of (2.2.2).

The program starts with the calculation of the positive roots using as before A1(1) as a subprogram. From the inverse of the Cartan matrix, the highest weight is calculated (2.1.7) and the value of 1 for its multiplicity is assigned to it. The program next generates the integers q_j, $j = 1,\ldots,\ell$ entering the relation $\lambda = \Lambda - \Sigma_{j=1}^{\ell} q_j\alpha_j$. For each set of values of $q_1, q_2, \ldots, q_\ell$ it evaluates the quantity $\lambda = \Lambda - \Sigma_{j=1}^{\ell} q_j\alpha_j$. If $m(\lambda) = 0$, then another set of the q_j integers is considered, while if $m(\lambda) \neq 0$ the weight λ is placed in the array of the weights, and the value of $m(\lambda)$ is assigned to its multiplicity. The program ends when the dimension of the representation is reached.

The input is the same as in Program B3(2). A new card is only needed in this program to terminate the procedure for the generating of the integers q_j.

In the output we get the weights with their multiplicity.

For some representations with very large dimensions, if we know the multiplicities of the first few non-simple weights, it is possible to predict all the other weights' multiplicities. In that case, the use of Program B4(2) can give us the information required for the first few weights.

The selective use of Programs B2(2), B3(2) and B4(2) can determine any weight system of any Lie algebra, classical or exceptional, within the capacity of a given computer (see Chapter 6).

§4.2.3 Weight systems of the representations $\underset{\sim}{126}$, $\underset{\sim}{120}$, $\underset{\sim}{16}$ and $\underset{\sim}{10}$ of D_5 and $\underset{\sim}{27}$, $\underset{\sim}{14}$ and $\underset{\sim}{7}$ of G_2

In Table 4.3 the positive weights of the D_5 representations are listed, while Table 4.4 gives the positive weights of G_2 representations.

126-DIM REPRESENTATION

1	2	3	1.5	2.5	MULT=1
1	2	3	1.5	1.5	MULT=1
1	2	3	1.5	0.5	MULT=1
1	2	2	1.5	1.5	MULT=1
1	2	2	1.5	0.5	MULT=1
1	2	2	0.5	1.5	MULT=1
1	2	2	0.5	0.5	MULT=1
1	2	1	1.5	0.5	MULT=1
1	2	1	0.5	0.5	MULT=1
1	2	1	-0.5	0.5	MULT=1
1	1	2	1.5	1.5	MULT=1
1	1	2	1.5	0.5	MULT=1
1	1	2	0.5	1.5	MULT=1
1	1	2	0.5	0.5	MULT=1
1	1	1	1.5	0.5	MULT=1
1	1	1	0.5	1.5	MULT=1
1	1	1	0.5	0.5	MULT=3
1	1	1	0.5	-0.5	MULT=1
1	1	1	-0.5	0.5	MULT=1
1	1	0	0.5	0.5	MULT=1
1	1	0	0.5	-0.5	MULT=1
1	1	0	-0.5	0.5	MULT=1
1	1	0	-0.5	-0.5	MULT=1
1	0	1	1.5	0.5	MULT=1
1	0	1	0.5	0.5	MULT=1
1	0	1	-0.5	0.5	MULT=1
1	0	0	0.5	0.5	MULT=1
1	0	0	0.5	-0.5	MULT=1
1	0	0	-0.5	0.5	MULT=1
1	0	0	-0.5	-0.5	MULT=1
1	0	-1	-0.5	0.5	MULT=1
1	0	-1	-0.5	-0.5	MULT=1
1	0	-1	-0.5	-1.5	MULT=1
0	1	2	1.5	1.5	MULT=1
0	1	2	1.5	0.5	MULT=1
0	1	2	0.5	1.5	MULT=1
0	1	2	0.5	0.5	MULT=1
0	1	1	1.5	0.5	MULT=1
0	1	1	0.5	1.5	MULT=1
0	1	1	0.5	0.5	MULT=3
0	1	1	0.5	-0.5	MULT=1
0	1	1	-0.5	0.5	MULT=1
0	1	0	0.5	0.5	MULT=1
0	1	0	0.5	-0.5	MULT=1
0	1	0	-0.5	0.5	MULT=1
0	1	0	-0.5	-0.5	MULT=1
0	0	1	1.5	0.5	MULT=1
0	0	1	0.5	1.5	MULT=1
0	0	1	0.5	0.5	MULT=3
0	0	1	0.5	-0.5	MULT=1
0	0	1	-0.5	0.5	MULT=1
0	0	0	0.5	0.5	MULT=3
0	0	0	0.5	-0.5	MULT=3

120-DIM REPRESENTATION

1	2	3	1.5	1.5	MULT=1
1	2	2	1.5	1.5	MULT=1
1	2	2	1.5	0.5	MULT=1
1	2	2	0.5	1.5	MULT=1
1	2	2	0.5	0.5	MULT=1
1	2	1	0.5	1.5	MULT=1
1	1	2	1.5	1.5	MULT=1
1	1	2	1.5	0.5	MULT=1
1	1	2	0.5	1.5	MULT=1
1	1	2	0.5	0.5	MULT=1
1	1	1	1.5	1.5	MULT=1
1	1	1	0.5	1.5	MULT=1
1	1	1	0.5	0.5	MULT=4
1	1	1	0.5	-0.5	MULT=1
1	1	1	-0.5	0.5	MULT=1
1	1	0	0.5	0.5	MULT=1
1	1	0	0.5	-0.5	MULT=1
1	1	0	-0.5	0.5	MULT=1
1	1	0	-0.5	-0.5	MULT=1
1	0	1	0.5	0.5	MULT=1
1	0	0	0.5	0.5	MULT=1
1	0	0	0.5	-0.5	MULT=1
1	0	0	-0.5	0.5	MULT=1
1	0	0	-0.5	-0.5	MULT=1
1	0	-1	-0.5	-0.5	MULT=1
0	1	2	1.5	1.5	MULT=1
0	1	2	1.5	0.5	MULT=1
0	1	2	0.5	1.5	MULT=1
0	1	2	0.5	0.5	MULT=1
0	1	1	1.5	0.5	MULT=1
0	1	1	0.5	1.5	MULT=1
0	1	1	0.5	0.5	MULT=4
0	1	1	0.5	-0.5	MULT=1
0	1	1	-0.5	0.5	MULT=1
0	1	0	0.5	0.5	MULT=1
0	1	0	0.5	-0.5	MULT=1
0	1	0	-0.5	0.5	MULT=1
0	1	0	-0.5	-0.5	MULT=1
0	0	1	1.5	0.5	MULT=1
0	0	1	0.5	1.5	MULT=1
0	0	1	0.5	0.5	MULT=4
0	0	1	0.5	-0.5	MULT=1
0	0	1	-0.5	0.5	MULT=1
0	0	0	0.5	0.5	MULT=4
0	0	0	0.5	-0.5	MULT=4

16-DIM REPRESENTATION

0.5	1	1.5	0.75	1.25	MULT=1
0.5	1	1.5	0.75	0.25	MULT=1
0.5	1	0.5	0.75	0.25	MULT=1
0.5	1	0.5	-0.25	0.25	MULT=1
0.5	0	0.5	0.75	0.25	MULT=1
0.5	0	0.5	-0.25	0.25	MULT=1
0.5	0	-0.5	-0.25	0.25	MULT=1
0.5	0	-0.5	-1.25	-0.75	MULT=1

16-DIM REPRESENTATION

1	1	1	0.5	0.5	MULT=1
0	1	1	0.5	0.5	MULT=1
0	0	1	0.5	0.5	MULT=1
0	0	0	0.5	0.5	MULT=1
0	0	0	0.5	-0.5	MULT=1

Table. 4.2

27-DIM REPR.

2	4	MULT=1
2	3	MULT=1
2	2	MULT=1
1	3	MULT=1
1	2	MULT=2
1	1	MULT=2
1	0	MULT=1
0	2	MULT=1
0	1	MULT=2
0	0	MULT=3

14-DIM REPR.

2	3	MULT=1
1	3	MULT=1
1	2	MULT=1
1	1	MULT=1
1	0	MULT=1
0	1	MULT=1
0	0	MULT=2

7-DIM REPR.

1	2	MULT=1
1	1	MULT=1
0	1	MULT=1
0	0	MULT=1

Table 4.3

§4.3 Clebsch-Gordan Series of Classical and Exceptional Lie Algebras

Let us suppose that we have two irreducible representations $\underset{\sim}{\Gamma}$ and $\underset{\sim}{\Gamma}'$. We saw in §4.2 that each irreducible representation is uniquely and completely specified by its highest weight. The representations $\underset{\sim}{\Gamma}$ and $\underset{\sim}{\Gamma}'$ can then be denoted by $\underset{\sim}{\Gamma}(\Lambda)$ and $\underset{\sim}{\Gamma}(\Lambda')$ where Λ and Λ' are the highest weights of the two representations. From these two representations we can construct the tensor product $\underset{\sim}{\Gamma}(\Lambda) \otimes \underset{\sim}{\Gamma}(\Lambda')$.

Definition: The Clebsch-Gordan Series

If we have two irreducible representations, then we call Clebsch-Gordan Series the series in which their tensor product is decomposed into irreducible representations,

$$\underset{\sim}{\Gamma}(\Lambda) \oplus \underset{\sim}{\Gamma}(\Lambda') = m_{\Lambda_1} \underset{\sim}{\Gamma}(\Lambda_1) \oplus m_{\Lambda_2} \underset{\sim}{\Gamma}(\Lambda_2) \oplus \ldots$$

where m_{Λ_i} are the multiplicities of each irreducible representation.

The Clebsch-Gordan Series is also known by the name Kronecker decomposition.

The value of the knowledge of the Clebsch-Gordan Series was emphasized in Chapter 3. In the mathematical literature, there are various methods of calculating the decomposition of two or more irreducible representations of a simple or semi-simple Lie algebra. In this section we shall review the existing methods, namely, the Young tableau technique (§4.3.1), the Konstant-Steinberg formula (§4.3.2), and the method using the higher order indices of a simple or semi-simple Lie algebra (§4.3.3).

From the analysis that follows, it should become obvious that the third method is the easiest one to implement by a computer program. The programs for the algebras A_ℓ, B_ℓ, C_ℓ, D_ℓ, G_2, F_4, E_6, E_7, E_8 are given in §4.3.4 and some results for the algebra D_ℓ (SO(10), SO(14), SO(18) groups) are listed in §4.3.5.

§4.3.1 Young tableau technique

Let us assume that a doublet of SU(2) is represented by the two states

$$\psi_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \psi_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3.1.1}$$

With these functions we denote the two states of a single particle of spin $\frac{1}{2}$. Another notation is by means of a single-box Young tableau $\square$. We make the identification

$$\psi_1 = \boxed{1}\ ,\ \psi_2 = \boxed{2}\ . \tag{3.1.2}$$

The single-box tableau without a number stands for both members of the doublet. Now, suppose we have a two-particle state. For that multiplet there are two possibilities: the state must be either symmetric, corresponding to the Young tableau $\square\square$, or antisymmetric corresponding to the tableau $\begin{array}{|c|}\hline \ \\ \hline \ \\ \hline\end{array}$. First, consider the symmetric state. If both particles are in the state ψ_1 the corresponding tableau is $\boxed{1\,|\,1}$ whereas if both particles are in the state ψ_2 the tableau is $\boxed{2\,|\,2}$. Now, it is obvious that the tableau $\boxed{1\,|\,2}$ will represent the situation when one particle is in the state ψ_1 and the other in the state ψ_2, and the arrangement $\boxed{2\,|\,1}$ is obviously the same as the arrangement $\boxed{1\,|\,2}$, since the state is symmetric. Thus, for the symmetric state we have the following arrangements:

$$\boxed{1\,|\,1} \qquad \boxed{2\,|\,2} \qquad \boxed{1\,|\,2}\ . \tag{3.1.3}$$

These tableaux represent a triplet. The only antisymmetric state is $\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline\end{array}$. Thus, the product representation of two SU(2) doublets decomposes into a triplet and a singlet. Symbolically with the help of Young tableaux

$$\square \times \square = \square\square + \begin{array}{|c|}\hline \ \\ \hline \ \\ \hline\end{array}\ . \tag{3.1.4}$$

In the standard notation, (3.1.4) can be written

$$\underset{\sim}{2} \otimes \underset{\sim}{2} = \underset{\sim}{3} \oplus \underset{\sim}{1}, \tag{3.1.5}$$

and in terms of states, (3.1.6) can be represented by the functions

$$\psi_1\psi_1,\ (\psi_1\psi_2+\psi_2\psi_1)/\sqrt{2},\ \psi_2\psi_2,\ (\psi_1\psi_2-\psi_2\psi_1)/\sqrt{2}. \tag{3.1.6}$$

These examples illustrate the fact that a Young tableau can be used to denote any multiplet of SU(2). The individual members of the multiplet are denoted by the different arrangements and the multiplicity by the total number of arrangements. An analogous result holds for SU(n) with the numbers in each box restricted to be 1,2,...,n.

We shall now give the general rules for reducing the Kronecker product of two representations by means of Young tableaux in order to obtain the representations of the Clebsch-Gordan series. We draw the two Young tableaux of the representations (for the general rules of constructing the Young tableau of a given representation, see [57]), marking each box of the second diagram with the number of the row to which it belongs. We then attach the boxes of the second tableau in all possible ways to the first tableau, subject to the following rules of the combined tableaux:

(1) each tableau should be a proper tableau ; that is, no row is longer than any row above it;

(2) no tableau should have a column with more than n boxes if the group is SU(n);

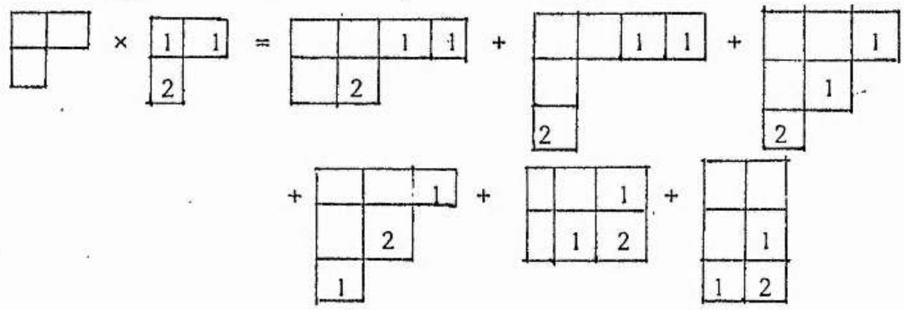
(3) we can make a path by counting each row from the right, starting with the top row; at each point of the path the number of boxes encountered with the number i must be less than or equal to the number of boxes with i-1;

(4) the numbers must not decrease in going from left to right across a row;

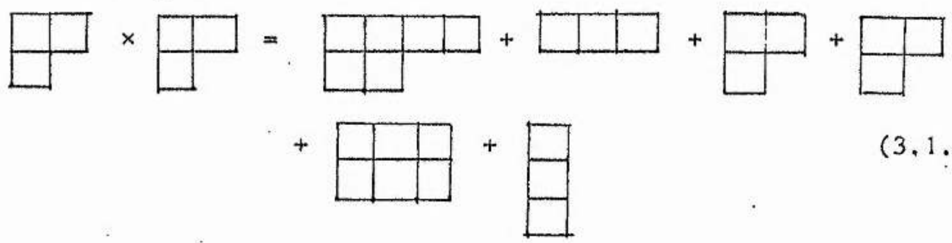
(5) the numbers must increase in going from top to bottom in a column.

As an example, we find the irreducible representations in the Kronecker product of the representations $\underset{\sim}{8}$ (D(1,1)) and $\underset{\sim}{8}$ of SU(3).

Following the above rules, we obtain [57]



To simplify the above picture, we can remove columns with three boxes and we get



(3.1.7)

or, in the standard notation

$$\underset{\sim}{8} \otimes \underset{\sim}{8} = \underset{\sim}{27} \oplus \underset{\sim}{10} \oplus \underset{\sim}{8} \oplus \underset{\sim}{8} \oplus \overline{\underset{\sim}{10}} \oplus \underset{\sim}{1}. \tag{3.1.8}$$

The Young Tableau Technique can be applied to any simple or semi-simple Lie algebra, but as we go to higher dimensional representations and to higher rank algebras the application of the method becomes very difficult.

§4.3.2 Kostant-Steinberg formula

The Kostant-Steinberg formula gives the multiplicity of an irreducible representation appearing in the decomposition of two irreducible representations. To introduce the Kostant-Steinberg formula we need to define the Weyl group.

For any linear functional β defined on H and for any non-zero root $\alpha \in \Delta$, define the linear functional $S_\alpha \beta$ on H by

$$(S_\alpha\beta)(h) = \beta(h) - \{2<\beta,\alpha>/<\alpha,\alpha>\}\alpha(h) \qquad (3.2.1)$$

for all $h \in H$. This defines an operator S_α that acts on linear functionals. The following properties are valid for any $\alpha \in \Delta$:

(a) $S_\alpha\alpha = -\alpha$,

(b) $S_\alpha(S_\alpha\beta) = \beta$ for any linear functional β on H,

(c) for any linear functionals β and γ on H,

$$<S_\alpha\beta, S_\alpha\gamma> = <\beta,\gamma>.$$

Definition: The Weyl group

The set consisting of the operators defined by the relation (3.2.1), the identity operator E defined by $E\alpha = \alpha$ (for any linear functional α on H), and all products of operators defined by $S_\alpha S_\beta\gamma = S_\alpha(S_\beta\gamma)$ (for any linear functional γ on H) is called the Weyl group and will be denoted by W.

It can be shown [54] that every element of W can be expressed as a product of the operators S_{α_j} associated with the simple roots. The construction of the Weyl group W is helped by the observations that if $S,T \in W$ then $S = T$ if and only if $S\alpha_j = T\alpha_j$ for every simple root α_j and that the relation (3.2.1) can be written (by (1.2.3)) as

$$S\alpha_j = \alpha_j - A_{kj}\alpha_k \qquad (3.2.2)$$

for $j,k = 1,2,\ldots,\ell$.

Theorem 4.21

If α is a positive root and $\alpha \neq \alpha_j$, then $S_{\alpha_j}\alpha > 0$.

Theorem 4.22

If $S_{\alpha_j}\alpha = S_{\alpha_j}\beta$ for some $j = 1,2,\ldots,\ell$, then $\alpha = \beta$.

The Kostant-Steinberg formula is given by the following theorem.

Theorem 4.23

The multiplicity of an irreducible representation with highest weight Λ appearing in the reduction of two irreducible representations with highest weights Λ^1 and Λ^2 respectively is given by

$$m_\Lambda = \sum_{S,T\in W} \det(ST)P\{S(\Lambda^1+\delta)+T(\Lambda^2+\delta)-\Lambda+2\delta)\}, \tag{3.2.3}$$

where W is the Weyl group, δ is one half of the sum over all the positive roots. P is the partition function defined as follows:

P(M) is the number of solutions $k_\alpha, k_\beta, \ldots, k_\omega$ of

$k_\alpha\alpha + k_\beta\beta + \ldots + k_\omega\omega = M$,

where all the $k_\alpha, k_\beta, \ldots, k_\omega$ are non-negative integers and $(\alpha, \beta, \ldots, \omega)$ is the set of all the positive roots (for the proof, see Jacobson [54], p 259).

The formula (3.2.3) is a very important result of the representation theory, because it solves the reduction problem of two irreducible representations completely. However, in practice its application is difficult. To see the difficulties involved, we shall apply it to the case of $\underset{\sim}{4} \otimes \underset{\sim}{4}$ of the algebra B_2.

Example

We are looking for the Clebsch-Gordan series of $\underset{\sim}{4} \otimes \underset{\sim}{4}$ of B_2. The roots of B_2 are

$$\alpha_1,\ \alpha_2,\ \alpha_1 + \alpha_2,\ \alpha_1 + 2\alpha_2. \tag{3.2.4}$$

The highest weights of the representations involved are

$$\Lambda^1 = \Lambda^2 = \tfrac{1}{2}\alpha_1 + \alpha_2.$$

Thus

$$\Lambda = m_1\Lambda^1 + m_2\Lambda^2 = (m_1+\tfrac{1}{2}m_2)\alpha_1 + (m_1+m_2)\alpha_2. \tag{3.2.5}$$

The value of $\delta = \frac{1}{2}\Sigma_{\alpha\in\Delta+}\alpha$ is

$$\delta = \frac{1}{2}(\alpha_1+\alpha_2+\alpha_1+\alpha_2+\alpha_1+2\alpha_2) = \frac{1}{2}(3\alpha_1+4\alpha_2). \tag{3.2.6}$$

Substituting (3.2.5) and (3.2.6) in (3.2.3), we get

$$m_\Lambda = \sum_{S,T\in W} \det(ST)P\{(S+T)(2\alpha_1+3\alpha_2) - [(m_1+\tfrac{1}{2}m_2)\alpha_1 + (m_1+m_2)\alpha_2] - (3\alpha_1+4\alpha_2)\}. \tag{3.2.7}$$

The first thing we can see from (3.2.7) is the double sum over the Weyl group in the right hand side of (3.2.7). In this particular example, it is easy to find how the term $2\alpha_1 + 3\alpha_2$ is transformed under the Weyl group. But for higher rank algebras the Weyl group becomes very large, so the calculation of the right hand side of (3.2.7) becomes very difficult (see Table 4.9).

In the case of B_2, where the Weyl group is

$$E,\ S_{\alpha_1},\ S_{\alpha_2},\ S_{\alpha_1}S_{\alpha_2},\ S_{\alpha_2}S_{\alpha_1},\ S_{\alpha_1}S_{\alpha_2}S_{\alpha_1},\ S_{\alpha_2}S_{\alpha_1}S_{\alpha_2},\ (S_{\alpha_1}S_{\alpha_2})^2, \tag{3.2.8}$$

Table 4.5 shows the results of the calculation of how the term $2\alpha_1 + 3\alpha_2$ transforms under (3.2.8). Substituting the different terms of Table 4.5 to (3.2.7), we find three possible pairs (m1,m2) which can express the argument of P as a linear combination of α_1 and α_2 by non-negative integers. These are

$$(0,0),\ (1,0),\ (0,2). \tag{3.2.9}$$

Using the Weyl dimensionality formula (2.2.1) we find

$$d_1^{(0,0)} = 1,\ d_2^{(1,0)} = 5,\ d_3^{(0,2)} = 10. \tag{3.2.10}$$

To find the multiplicities of the representations (3.2.10), we substitute the values of (m_1,m_2) (3.2.9) to (3.2.7) and evaluate the partition function P.

<u>(a) $d_1^{(0,0)}$</u>

For this representation (3.2.7) becomes

$$m_\Lambda = \sum_{S,T\in W} \det(ST)P[(S+T)(2\alpha_1+3\alpha_2) - (3\alpha_1+4\alpha_2)]. \qquad (3.2.11)$$

Table 4.6 shows the terms of the argument of P of the right hand side of (3.2.11) which are not negative (the negative terms in the argument of P cannot give any contribution to the partition function). From Table 4.6 we get

$$\begin{aligned} m_\Lambda &= \det(EE)P(\alpha_1+2\alpha_2) + \det(ES_{\alpha_1})P(2\alpha_2) + \det(ES_{\alpha_2})P(\alpha_1) \\ &+ \det(S_{\alpha_1}E)P(2\alpha_2) + \det(S_{\alpha_1}S_{\alpha_1})P(-\alpha_1+2\alpha_2) + \det(S_{\alpha_1}S_{\alpha_2})P(0) \\ &+ \det(S_{\alpha_2}E)P(\alpha_1) + \det(S_{\alpha_2}S_{\alpha_1})P(0) + \det(S_{\alpha_2}S_{\alpha_2})P(\alpha_1-2\alpha_2). \end{aligned} \qquad (3.2.12)$$

The values of $\det(S)$, $S \in W$, are [54]

$$\det S = -1,\ S \in W - \{E\},\ \det E = 1. \qquad (3.2.13)$$

Equation (3.2.12), after substitution of (3.2.13), becomes

$$\begin{aligned} m_\Lambda &= P(\alpha_1+2\alpha_2) - 2P(2\alpha_2) - 2P(\alpha_1) + P(-\alpha_1+2\alpha_2) + 2P(0) \\ &+ P(\alpha_1-2\alpha_2). \end{aligned} \qquad (3.2.14)$$

The argument $\alpha_1 + 2\alpha_2$ of the partition function P of the first term of the right hand side of (3.2.14) can be expressed in three different ways as a linear combination of the positive roots (3.2.4) by non-negative integers.

$$2(\alpha_1) + 2(\alpha_2),\ 1(\alpha_1+\alpha_2) + 1(\alpha_2),\ 1(\alpha_1+2\alpha_2) + 0(\alpha) \quad (\alpha \in \Delta).$$

Thus

$$P(\alpha_1+2\alpha_2) = 3. \qquad (3.2.15)$$

Similarly, we get

$$P(2\alpha_2) = 3,\ P(\alpha_1) = 3,\ P(-\alpha_1+2\alpha_2) = P(\alpha_1-2\alpha_2) = 0. \qquad (3.2.16)$$

Finally,

Table 4.5: Transformation properties of $2\alpha_1 + 3\alpha_2$

S \ T	E	S_{α_1}	S_{α_2}	$S_{\alpha_1}S_{\alpha_2}$	$S_{\alpha_2}S_{\alpha_1}$	$S_{\alpha_1}S_{\alpha_2}S_{\alpha_1}$	$S_{\alpha_2}S_{\alpha_1}S_{\alpha_2}$	$(S_{\alpha_1}S_{\alpha_2})^2$
E	$4\alpha_1 + 6\alpha_2$							
S_{α_1}	$3\alpha_1 + 6\alpha_2$	$2\alpha_1 + 6\alpha_2$						
S_{α_2}	$4\alpha_1 + 4\alpha_2$	$3\alpha_1 + 4\alpha_2$	$4\alpha_1 + 2\alpha_2$					
$S_{\alpha_1}S_{\alpha_2}$	$4\alpha_1 + \alpha_1$	$4\alpha_2$	$\alpha_1 + 2\alpha_2$	$2\alpha_2 - 2\alpha_1$				
$S_{\alpha_2}S_{\alpha_1}$	$3\alpha_1 + 2\alpha_2$	$2\alpha_1 + 2\alpha_2$	α_1	0	$2\alpha_1 - 2\alpha_2$			
$S_{\alpha_1}S_{\alpha_2}S_{\alpha_1}$	$2\alpha_2$	$-\alpha_1 + 2\alpha_2$	0	$-\alpha_1$	$-\alpha_1 - 2\alpha_2$	$-4\alpha_1 - 2\alpha_2$		
$S_{\alpha_2}S_{\alpha_1}S_{\alpha_2}$	α_1	0	$\alpha_1 - 2\alpha_2$	$-2\alpha_1 - 2\alpha_2$	$-4\alpha_2$	$-3\alpha_1 - 4\alpha_2$	$-2\alpha_1 - 6\alpha_2$	
$(S_{\alpha_1}S_{\alpha_2})^2$	0	$-\alpha_1$	$-2\alpha_2$	$-3\alpha_1 - 2\alpha_2$	$-4\alpha_2 - \alpha_1$	$-4\alpha_1 - 4\alpha_2$	$-3\alpha_1 - 6\alpha_2$	$-4\alpha_1 - 6\alpha_2$

$$P(0) = 1 \qquad (3.2.17)$$

by definition [54].

Substituting (3.2.15), (3.2.16) and (3.2.17) to (3.2.14), we get

$$m_\Lambda = 3 - 2 - 2 + 2 = 1.$$

<u>(b) $d_2^{(1,0)}$</u>

Equation (3.2.7) becomes

$$m_\Lambda = \sum_{S,T \in W} \det(ST) P[(S+T)(2\alpha_1 + 3\alpha_2) - 4\alpha_1 - 5\alpha_2].$$

Table 4.7 gives $P(\alpha_2) = 1$. Thus, $m_\Lambda = 1$.

<u>(c) $d_3^{(0,2)}$</u>

Equation (3.2.7) becomes

$$m_\Lambda = \sum_{S,T \in W} \det(ST) P[(S+T)(2\alpha_1 + 3\alpha_2) - 4\alpha_1 - 6\alpha_2].$$

Table 4.8 gives $P(0) = 1$. Thus, $m_\Lambda = 1$.

The final result is

$$\underset{\sim}{4} \otimes \underset{\sim}{4} = \underset{\sim}{1} \oplus \underset{\sim}{5} \oplus \underset{\sim}{10}. \qquad (3.2.18)$$

As we saw in the above example, the application of the Kostant-Steinberg formula is in practice difficult. We must know the Weyl group in order to apply (3.2.3). In Table 4.9 we give the order of the Weyl group for all the classical and exceptional algebras.

We have developed a program (Program C1(3)) to generate the Weyl group.

<u>Description of Program C1(3)</u>

First the program calculates the effect of the identity operator E on the simple roots, and stores the transformed simple roots under the array <u>TRANS</u>. Next, it examines if the product of operators $S_{\alpha_i} S_{\alpha_j} \ldots S_{\alpha_k}$ with $i, j, \ldots, k = 1, \ldots, \ell$ when applied on the simple roots according to (3.2.1) gives one of the non-zero roots. If it does, the

Table 4.6: Transformation properties of $2\alpha_1 + 3\alpha_2$ of the representation D(0,0)

S \ T	E	S_{α_1}	S_{α_2}
E	$\alpha_1 + 2\alpha_2$	$2\alpha_2$	α_1
S_{α_1}	$2\alpha_2$	$-\alpha_1 + 2\alpha_2$	0
S_{α_2}	α_1	0	$\alpha_1 - 2\alpha_2$

Table 4.7: D(1,0) representation

S \ T	E
E	α_2

Table 4.8: D(0,2) representation

S \ T	E
S	0

Table 4.9

Type of algebra	Order of Weyl group
A_ℓ	$(\ell+1)!$
B_ℓ, C_ℓ	$2^\ell.\ell!$
D_ℓ	$2^{\ell-1}.\ell!$
E_6	$2^7.3^4.5$
E_7	$2^{10}.3^4.5.7$
E_8	$2^{14}.3^5.5^2.7$
F_4	$2^7.3^2$
G_2	$2^2.3$

program tests whether the new element $S = S_{\alpha_i} S_{\alpha_j} \ldots S_{\alpha_k}$ has already been calculated. If it is not, it is placed in the array WEYL-GROUP and the program continues until it reaches the order of the Weyl group.

The input consists of:
the rank of the algebra; the number of positive roots; the order of the Weyl group (Table 4.9); the Weyl group of the simple roots; the positive roots, and finally the Cartan matrix.

In the output we get the generated Weyl group in the notation $S_{\alpha_i} S_{\alpha_j} \ldots S_{\alpha_k}$, $i,j,k = 1,\ldots,\ell$.

Using Program C1(3), we have generated the Weyl group of the classical Lie algebras up to rank six.

Using the Weyl group, we have implemented the Kostant-Steinberg formula to a computer program (Program C2(3)).

Description of Program C2(3)

This program includes the following procedures:
procedure REFLECTION, procedure DELTA, procedure PARTITION. The program starts with the generation of the Weyl group, using C1(3) as a sub-program. The call of the procedure DELTA calculates the number δ. The procedure REFLECTION, when it is called, finds all the terms like the ones shown in Tables 4.5 to 4.8. The knowledge of these terms determines the representations $D(m_1,m_2,\ldots,m_\ell)$. The call of Program B1(2) calculates the dimensions of the representations for each value of the integer numbers $m_1,m_2,\ldots,m_\ell$. Then the program proceeds recursively and calculates the number of times each of the above representations appears in the reduction.

The input of the program consists of:
the rank of the algebra; the number of positive roots; the order of the Weyl group; the dimensions of the two representations under reduction; the positive roots, and the Cartan matrix.

In the output we get the dimensions of the representations and their multiplicity.

Program C2(3) can be used for the classical algebras up to rank five.

§4.3.3 The method of higher order indices

Having analysed the common methods of tensor decomposition, ie the Young tableau technique and the Kostant-Steinberg formula, we come to the third method of the higher indices, which can be easily implemented as a computer program. In this section we shall describe the method and in §4.3.4 we shall give the programs.

The method was first introduced by Patera, Sharp and Winternitz

[58,59]. It lacks the conceptual simplicity of the Kostant-Steinberg formula, but it is very powerful from the computational point of view. It is a guess like method, in the sense that the knowledge of the values of the higher indices allows us to guess the Clebsch-Gordan decomposition.

Dynkin [60] defined the index of an irreducible representation of a simple Lie group by the formula

$$J = d(k^2-R^2)/r. \tag{3.3.1}$$

d is the dimension of the irreducible representation, r is the order of the group, and $R^2 = \underline{R}.\underline{R}$, where R is half the sum of the positive weights of the adjoint representation. k is defined as

$$k = \Lambda + R, \tag{3.3.2}$$

where Λ is the highest weight of the irreducible representation.

Dynkin showed that this index has additivity properties similar to those of the dimension under reduction of the direct product of two irreducible representations. If the direct product of irreducible representations 1 and 2 decomposes into irreducible representations, then

$$d_2 J_1 + d_1 J_2 = \Sigma J. \tag{3.3.3}$$

Patera et al [58] generalized Dynkin's index by defining the n^{th} order index, n a non-negative even integer, as the sum of n^{th} powers of the magnitudes of the weights of the irreducible representation

$$I^{(n)} = \sum_{\lambda} \lambda^n. \tag{3.3.4}$$

The sum is over all weights λ belonging to the irreducible representation, each occurring a number of times equal to its multiplicity.

The zeroth order index is just the dimension of the representation. The second order index is Dynkin's index (3.3.1) multiplied by the rank of the group

$$I^{(2)} = \ell d(k^2-R^2)/r. \tag{3.3.5}$$

We shall give the explicit algebraic forms of the $I^{(2)}$, $I^{(4)}$ indices for each Lie algebra.

A Special unitary groups

The indices of the special unitary group SU(n) (= A_{n-1}) are:

$$I^{(2)} = d[n(n+1)]^{-1} \sum_{i<j} [(\ell_i-\ell_j)^2 - (\ell_i^0-\ell_j^0)^2]. \tag{3.3.6}$$

$$\begin{aligned} I^{(4)} = d[[p_4(\ell)-p_4(\ell^0)\ \frac{(n-1)(n^2+7n-6)}{n^2(n+1)(n+2)(n+3)} + [p_4(\ell)-p_1(\ell)p_3(\ell)- \\ -p_4(\ell^0)+p_1(\ell^0)p_3(\ell^0)]\frac{n^2+7n-6}{n^2(n+1)(n+2)} \\ +\{3[p_2(\ell)]^2 - 3[p_1(\ell)]^2 p_2(\ell) + [p_1(\ell)]^4 - p_4(\ell) - 3[p_2(\ell^0)]^2 \\ +3[p_1(\ell^0)]^2 p_2(\ell^0) - [p_1(\ell^0)]^4 + p_4(\ell^0)\}.\frac{1}{n^2} \\ +\{[p_2(\ell)]^2 + p_1(\ell)p_3(\ell) - p_4(\ell) - [p_1(\ell)]^2 p_2(\ell) - [p_2(\ell^0)]^2 \\ -p_1(\ell^0)p_3(\ell^0) + p_4(\ell^0) + [p_1(\ell^0)]^2 p_2(\ell^0)\}.\frac{n-3}{n^2(n+1)} \\ -\frac{1}{6}\sum_{i<j}\{(\ell_i-\ell_j)^2 - (\ell_i^0-\ell_j^0)^2\}]. \end{aligned} \tag{3.3.7}$$

Here, the ℓ_j are given by

$$\ell_j = \sum_{k=j}^{n-1} \lambda_k + n - j, \quad \ell_n = 0. \tag{3.3.8}$$

The λ_k are given by

$$\lambda_k = 2\Lambda_j.\alpha_k/\alpha_k^2, \tag{3.3.9}$$

where α_k are the simple roots. The ℓ_j^0 are not defined in [58], but in [60] (page 356, g_k in Dynkin's notation) an explicit form of ℓ_j^0 is given. For the unitary groups we have

$$\ell_j^0 = n - j. \tag{3.3.10}$$

The functions $p_i(\epsilon)$, $i = 0,\ldots,4$ are defined in [58] as follows:

$$p_0(\epsilon) = 1,$$

$$p_1(\epsilon) = \sum_i \epsilon_i,$$

$$p_2(\epsilon) = \sum_i \epsilon_i^2 + \sum_{i>j} \epsilon_i \epsilon_j,$$

$$p_3(\epsilon) = \sum_i \epsilon_i^3 + \sum_{i\neq j} \epsilon_i^2 \epsilon_j + \sum_{i>j>k} \epsilon_i \epsilon_j \epsilon_k,$$

$$p_4(\epsilon) = \sum_i \epsilon_i^4 + \sum_{i\neq j} \epsilon_i^2 \epsilon_j + \sum_{i>j} \epsilon_i^2 \epsilon_j^2 + \sum_{i\neq j,k} \sum_{j>k} \epsilon_i^2 \epsilon_j \epsilon_k$$
$$+ \sum_{i>j>k>\ell} \epsilon_i \epsilon_j \epsilon_k \epsilon_\ell. \qquad (3.3.11)$$

Formula (3.3.7) is valid for all SU(n) groups except SU(2) and SU(3). For these groups the formulae for the exceptional groups are applied.

B Orthogonal groups

I) $SO(2n+1)$ $(= B_n)$

$$I^{(2)} = \frac{2d}{2(2n+1)} \sum_j [\ell_j^2 - (\ell_j^0)^2]. \qquad (3.3.12)$$

(In the case of SO(3), $I^{(2)}$ must be multiplied by a factor of 2.)

$$I^{(4)} = 4d\left(\frac{(n+5)[p_2(\ell^2)-p_2((\ell^0)^2)]}{4(n+1)(2n+1)(2n+3)} + \frac{\sum_{i>j}[(\ell_i\ell_j)^2 - (\ell_i^0\ell_j^0)^2]}{4(2n-1)(2n+1)}\right.$$
$$\left. - \frac{(n+2)\Sigma_i(\ell_i^0)^2 \Sigma_j[\ell_j^2-(\ell_j^2)^2]}{2n(2n+1)^2}\right). \qquad (3.3.13)$$

(In the case of SO(3), $I^{(4)}$ must be multiplied by 4.) Here,

$$\ell_j = \sum_{k=j}^{n-1} \lambda_k + \frac{1}{2}\lambda_n + n - j + \frac{1}{2},$$

$$\ell_j^0 = n - j + \frac{1}{2} \quad \text{(ref [60])}. \qquad (3.3.14)$$

The λ_k are as in (3.3.9), and $p_i(\epsilon)$ as in (3.3.11).

II) $SO(2n)$ $(= D_n)$

$$I^{(2)} = (2n-1)^{-1} d \sum_j (\ell_j - \ell_j^0)(\ell_j - \ell_j^0) \qquad (3.3.15)$$

$$I^{(4)} = d[\frac{(n+5)[p_2(\ell^2) - p_2((\ell^0)^2)]}{(n+1)(2n-1)(2n+1)} + \frac{\Sigma_{i>j}[(\ell_i\ell_j)^2 - (\ell_i^0\ell_j^0)^2]}{(2n-1)(2n-3)}$$

$$- \frac{2(n+2)\Sigma_i(\ell_i^0)\Sigma_j[\ell_j^2 - (\ell_j^0)^2]}{n(2n-1)^2}]. \qquad (3.3.16)$$

Here,

$$\left.\begin{aligned} \ell_j &= \sum_{k=j}^{n-2}\lambda_k + \frac{1}{2}(\lambda_{n-1}-\lambda_n) + n - j, \ 1 \le j \le n-1, \\ \ell_\eta &= \frac{1}{2}(\lambda_{n-1}-\lambda_n). \end{aligned}\right\} \qquad (3.3.17)$$

$$\left.\begin{aligned} \ell_j^0 &= n - j, \ 1 \le j \le n-1, \\ \ell_\eta^0 &= 0. \end{aligned}\right\} \qquad (3.3.18)$$

λ_k and $p_i(\epsilon)$ are as in (3.3.9) and (3.3.11) respectively.

Formula (3.3.15) is valid for all SO(2n), while (3.3.16) is valid for SO(2n) with $\eta \geq 3$.

C Symplectic groups

For the symplectic group Sp(2n) (= C_n), we have

$$I^{(2)} = \frac{d}{2(2n+1)} \sum_j [\ell_j^2 - (\ell_j^0)^2]. \qquad (3.3.19)$$

$$I^{(4)} = d(\frac{(n+5)[p_2(\ell^2) - p_2((\ell^0)^2)]}{4(n+1)(2n+1)(2n+3)} + \frac{\Sigma_{i>j}[(\ell_i\ell_j)^2 - (\ell_i^0\ell_j^0)^2]}{4(2n-1)(2n+1)}$$

$$- \frac{(n+2)\Sigma_i(\ell_i^0)^2\Sigma_j[\ell_j^2-(\ell_j^0)^2]}{2n(2n+1)^2}). \qquad (3.3.20)$$

Here,

$$\left.\begin{aligned} \ell_j &= \sum_{k=j}^{n}\lambda_k + n - j + 1, \\ \ell_j^0 &= n - j + 1. \end{aligned}\right\} \qquad (3.3.21)$$

λ_k and $p_i(\epsilon)$ are as in (3.3.9) and (3.3.11) respectively.

D Exceptional groups

$$I^{(2)} = \ell d(k^2-R^2)/r. \qquad (3.3.22)$$

$$I^{(4)} = \frac{\ell+2}{d\ell}\{I^{(2)}\}^2 - \frac{d}{120R^4}\{k^4-R^4\}\Sigma_{\alpha}^{+}\alpha^4. \qquad (3.3.23)$$

Equation (3.3.23) is valid for all five exceptional groups G_2, F_4, E_6, E_7, E_8 and for SU(2) and SU(3). The values of $\Sigma_{\alpha}^{+}\alpha^4$ for these seven groups are: SU(2): 4; SU(3): 12; G_2: 40/3; F_4: 60; E_6: 144; E_7: 252; E_8: 480.

§4.3.4 Programs for the higher order indices

The algebraic form of the formulae (3.3.6) to (3.3.23) allows us to develop them into computer programs. Programs C3(3) (for the algebra A_ℓ), C4(3) (for the algebra B_ℓ), C5(3) (for the algebra C_ℓ), C6(3) (for the algebra D_ℓ) have similar structure, while Program C7(3) (for the exceptional algebras) is different.

We shall describe Programs C3(3) and C7(3).

Description of Program C3(3)

The program is structured as follows:
the procedures SECONDORDER, FOURTHORDER calculate the indices $I^{(2)}$, $I^{(4)}$ respectively using the formulae (3.3.6) and (3.3.7); the procedures PA1, PA2, PA3, PA4 calculate the functions $p_i(\epsilon)$, $i = 1,\ldots,4$, of Equations (3.3.11); the procedures LABEL, LABEL2 are the formulae (3.3.8) and (3.3.10); the procedure INDEX stands for the formula (3.3.9). The program includes a subprogram for calculating the positive roots (Program A1(1)), and the dimension of a representation (Program B1(2)).

The input of the program is:
the rank of the algebra; the number of representations whose indices we want to determine; in the notation $D(\{n_1,n_2,\ldots,n_\ell\})$, the numbers $\min(n_1,n_2,\ldots,n_\ell)$ and $\max(n_1,n_2,\ldots,n_\ell)$; the simple roots; the Cartan

matrix; the inverse of the Cartan matrix; the weighting factors of the Dynkin diagram.

As output we get:

the numbers $n_1, n_2, \ldots, n_\ell$, specifying the representation $D(\{n_1, n_2, \ldots, n_\ell\})$ in the first ℓ columns; the dimension of the representation in the ℓ^{th} column; the indices $I^{(2)}$, $I^{(4)}$ in the $\ell + 2$, $\ell + 3$ columns.

Description of Program C7(3)

The structure of the program is quite the same as Program C3(3) with the only difference being that the procedures PA1, PA2, PA3, PA4 are no longer needed.

As an extra input, we have the number of positive roots; the order of the group; the values of the $\Sigma_{\alpha}^{+}\alpha^4$, and the adjoint representation with its dimension.

The output is the same as for Program C3(3).

§4.3.5 Applications and results

Let us suppose that we want to know the Kronecker decomposition of the two lowest dimensional spinorial representations of the group SO(10). Running Program C6(3), we find the following values of the $I^{(2)}$ and $I^{(4)}$ indices, for the few lowest dimensional representations.

Table 4.11: Higher indices of the group SO(10)

Representation	Dim	$I^{(2)}$	$I^{(4)}$
D(0,0,0,0,0)	1	0	0
D(0,0,0,0,1)	16	20	25
D(0,0,0,0,2)	126	350	1150
D(0,0,0,1,0)	16	20	25
D(0,0,0,1,1)	210	560	1760
D(0,0,0,2,0)	126	350	1150
D(0,0,1,0,0)	120	280	760
D(0,1,0,0,0)	45	80	160
D(0,1,0,1,0)	560	1820	7195
D(0,2,0,0,0)	770	3080	15360
D(1,0,0,0,0)	10	10	10
D(1,0,0,0,1)	144	340	445
D(1,0,1,0,0)	945	3360	14720
D(1,1,0,0,0)	320	960	3520
D(2,0,0,0,0)	54	120	320
D(2,0,0,0,1)	720	2660	12245

The SO(10) group has two lowest dimensional spinorial representations $\underset{\sim}{16}$ and $\underset{\sim}{16}'$. For the $\underset{\sim}{16} \otimes \underset{\sim}{16}$ decomposition, we have

$$I^{(2)}_{\underset{\sim}{16}} = 20, \; I^{(4)}_{\underset{\sim}{16}} = 25, \tag{3.5.1}$$

and if we assume that $\underset{\sim}{16} \otimes \underset{\sim}{16}$ decomposes according to

$$\underset{\sim}{16} \otimes \underset{\sim}{16} = f_1 + f_2 + \ldots + f_k, \tag{3.5.2}$$

then, using the additivity properties of the indices (§4.3.3), we have for the indices of the right hand side of (3.5.2)

$$\sum_k I^{(2)}_k = N_2 I^{(2)}_1 + N_1 I^{(2)}_2, \tag{3.5.3}$$

$$\Sigma I_k^{(4)} = N_2 I_1^{(4)} + N_1 I_2^{(4)} + [2(\ell+2)/\ell] I_1^{(2)} I_2^{(2)}, \tag{3.5.4}$$

$$\Sigma N_k = N_1 N_2. \tag{3.5.5}$$

Substituting the different values of N_1, N_2, $I^{(2)}$, $I^{(4)}$, we find

$\underset{\sim}{16} \otimes \underset{\sim}{16}$	$N = 256$	$\Sigma I^{(2)} = 640$	$\Sigma I^{(4)} = 1920$	
N	$\underset{\sim}{126}$ $\otimes$	$\underset{\sim}{120}$ $\otimes$	$\underset{\sim}{10}$	$\Sigma N = 256$
$I^{(2)}$	350	280	10	$\Sigma I^{(2)} = 640$
$I^{(4)}$	1150	760	10	$\Sigma I^{(4)} = 1920$

and, for the case $\underset{\sim}{16} \otimes \underset{\sim}{16}'$, we have

$\underset{\sim}{16} \otimes \underset{\sim}{16}'$	$N = 256$	$\Sigma I^{(2)} = 640$	$\Sigma I^{(4)} = 1920$	
N	$\underset{\sim}{210}$ $\oplus$	$\underset{\sim}{45}$ $\oplus$	$\underset{\sim}{1}$	$\Sigma N = 256$
$I^{(2)}$	560	80	0	$\Sigma I^{(2)} = 640$
$I^{(4)}$	1760	160	0	$\Sigma I^{(4)} = 1920$

$\underline{D_7}$ (= SO(14))

Table 4.12: Higher indices of the group SO(14)

Representation	Dim	$I^{(2)}$	$I^{(4)}$
D(0,0,0,0,0,0,0)	1	0	0
D(0,0,0,0,0,0,1)	64	112	196
D(0,0,0,0,0,0,2)	1716	6468	27636
D(0,0,0,0,0,1,1)	3003	11088	46368
D(0,0,0,0,1,0,0)	2002	6930	27090
D(0,0,0,1,0,0,0)	1001	3080	10640
D(0,0,1,0,0,0,0)	364	924	2604
D(0,1,0,0,0,0,0)	91	168	336
D(0,1,0,0,0,0,1)	4928	18480	80052
D(0,2,0,0,0,0,0)	3080	12320	58352
D(1,0,0,0,0,0,0)	14	14	14
D(2,0,0,0,0,0,0)	104	224	560

For the $\underset{\sim}{64} \otimes \underset{\sim}{64}$ and $\underset{\sim}{64} \otimes \underset{\sim}{64}'$ decompositions we have

$\underset{\sim}{64} \otimes \underset{\sim}{64}$	N = 4096 $\Sigma I^{(2)}$ = 14,336 $\Sigma I^{(4)}$ = 57,344				
N	$\underset{\sim}{1716}$ ⊕	$\underset{\sim}{2002}$ ⊕	$\underset{\sim}{364}$ ⊕	$\underset{\sim}{14}$	ΣN = 4096
$I^{(2)}$	6468	6930	924	14	$\Sigma I^{(2)}$ = 14336
$I^{(4)}$	27636	27090	2604	14	$\Sigma I^{(4)}$ = 57344
$\underset{\sim}{64} \otimes \underset{\sim}{64}'$	N = 4096 $\Sigma I^{(2)}$ = 14,336 $\Sigma I^{(4)}$ = 57,344				
N	$\underset{\sim}{3003}$ ⊕	$\underset{\sim}{91}$ ⊕	$\underset{\sim}{1001}$ ⊕	$\underset{\sim}{1}$	ΣN = 4096
$I^{(2)}$	11088	168	3080	0	$\Sigma I^{(2)}$ = 14336
$I^{(4)}$	46368	336	10640	0	$\Sigma I^{(4)}$ = 57344

$\underline{D_9}$ (= SO(18))

Table 4.13: Higher order indices of the group SO(18)

Representation	Dim	$I^{(2)}$	$I^{(4)}$
D(0,0,0,0,0,0,0,0,0)	1	0	0
D(0,0,0,0,0,0,0,0,1)	256	576	1296
D(0,0,0,0,0,0,0,1,0)	256	576	1296
D(0,0,0,0,0,0,0,1,1)	43758	205920	1070784
D(0,0,0,0,0,0,1,0,0)	31824	144144	720720
D(0,0,0,0,0,1,0,0,0)	18564	78624	366912
D(0,0,0,0,1,0,0,0,0)	8568	32760	137592
D(0,0,0,1,0,0,0,0,0)	3060	10080	36288
D(0,0,1,0,0,0,0,0,0)	816	2160	6192
D(0,1,0,0,0,0,0,0,0)	153	288	576
D(0,1,0,0,0,0,0,0,1)	34560	146880	697968
D(1,0,0,0,0,0,0,0,0)	18	18	18
D(1,0,0,0,0,0,0,0,1)	4352	14400	51984
D(0,0,0,0,0,0,0,0,2)	24310	115830	610038
D(2,0,0,0,0,0,0,0,0)	170	360	864

For the decomposition we have

$\underset{\sim}{256} \otimes \underset{\sim}{256}$	$\Sigma N = 65536$ $\Sigma I^{(2)} = 294912$ $\Sigma I^{(4)} = 1474560$					
N	$\underset{\sim}{24310}$ ⊕	$\underset{\sim}{31824}$ ⊕	$\underset{\sim}{8568}$ ⊕	$\underset{\sim}{816}$ ⊕	$\underset{\sim}{18}$	$\Sigma N = 65536$
$I^{(2)}$	115830	144144	32760	2160	18	$\Sigma I^{(2)} = 294912$
$I^{(4)}$	610038	720720	137592	6192	18	$\Sigma I^{(4)} = 1474560$

256 ⊗ 256'	$\Sigma N = 65536$ $\Sigma I^{(2)} = 294912$ $\Sigma I^{(4)} = 1474560$					
N	153 ⊕	43758 ⊕	18564 ⊕	3060 ⊕	1	$\Sigma N = 65536$
$I^{(2)}$	288	205920	78624	10080	0	$\Sigma I^{(2)} = 294912$
$I^{(4)}$	576	1070784	366912	36288	0	$\Sigma I^{(4)} = 1474560$

Finally, we summarize:

Table 4.14: Clebsch-Gordan Series of the lowest dim representations of SO(10), SO(14), SO(18) groups

Group	Clebsch-Gordan Series
SO(10)	16 ⊗ 16 = 126 ⊕ 120 ⊕ 10 16 ⊗ 16' = 210 ⊕ 45 ⊕ 1 10 ⊗ 10 = 54 ⊕ 45 ⊕ 1
SO(14)	64 ⊗ 64 = 2002 ⊕ 1716 ⊕ 364 ⊕ 14 64 ⊗ 64' = 3003 ⊕ 1001 ⊕ 91 ⊕ 1 14 ⊗ 14 = 104 ⊕ 91 ⊕ 1
SO(18)	256 ⊗ 256 = 31824 ⊕ 24310 ⊕ 8568 ⊕ 816 ⊕ 18 256 ⊗ 256' = 43758 ⊕ 18564 ⊕ 3060 ⊕ 1 18 ⊗ 18 = 120 ⊕ 153 ⊕ 1

§4.4 Matrix Representation

As we discussed in Chapter 3, an explicit matrix realization of an irreducible representation of a simple complex Lie algebra enables us to carry out detailed calculations of the physical quantities involved. In the mathematical literature various methods exist [18,61] for the construction of explicit matrix representations. Nevertheless, none of these methods provide a simple and understandable approach. Moreover they require a mathematical background which in most cases is a privilege only of the specialists. We believe that the rapidly developing theory of grand unification needs a simple and tractable method for finding a matrix realization of a given irreducible representation.

Our approach is based upon a simple idea. The knowledge of the A_1-subalgebra content of an irreducible representation (in other words, how the A_1-subalgebra is embedded to a representation of an algebra $\mathcal{L}$) is sufficient to specify its matrix representation up to a phase factor. On the other hand, the structure of any representation of A_1 Lie algebra is well known (it is the familiar SU(2) angular momentum theory).

An important by-product of this method is a procedure for evaluating the Clebsch-Gordan coefficients. As we shall explain in §4.5, the unambiguous specification of the states of a representation fixes the Clebsch-Gordan coefficients of the representations entering the Clebsch-Gordan series.

The mathematical background that will be needed is explored in §4.4.1. The discussion of the method is in §4.4.2, and finally in §4.4.3 we give the programs.

§4.4.1 A matrix realization of a simple Lie algebra

In §4.2.1 we showed that it is possible to associate with every linear functional α on H, and in particular with each root $\alpha \in \Delta$, a unique element h_α of H by the definition

$$B(h_\alpha, h) = \alpha(h), \qquad (4.1.1)$$

for all $h \in H$. Then, from (4.1.1), it follows that

$$h_{\alpha+\beta} = h_\alpha + h_\beta. \qquad (4.1.2)$$

The following theorem gives us the information we need for the construction of the Weyl canonical basis.

Theorem 4.23

H coincides with the subspace of $\mathcal{L}$ consisting of all elements of the form $\sum_{\alpha\in\Delta}\mu_\alpha h_\alpha$, where μ_α takes all complex values.

This theorem implies that from the set of elements h_α ($\alpha \in \Delta$) a subset of ℓ linearly-independent elements may be selected and may be taken to form a basis for H. Let H_R denote the real vector space with basis $h_{\alpha_1}, h_{\alpha_2}, \ldots, h_{\alpha_\ell}$. Theorem 4.8 implies that, for any $\alpha \in \Delta$, $h_\alpha = \Sigma_{j=1}^{\ell} k_j h_{\alpha_j}$, with $k_1, k_2, \ldots, k_\ell$ real and rational, so $h_\alpha \in H_R$ for all $\alpha \in \Delta$. Thus, H_R is actually independent of the choice of the basis $h_{\alpha_1}, h_{\alpha_2}, \ldots, h_{\alpha_\ell}$ of H.

Now, for each pair of roots α and $-\alpha$ of Δ, there is a three-dimensional simple subalgebra of $\mathcal{L}$ which can be constructed in the following way. Define H_α ($\in$ H) by

$$H_\alpha = \{2/\langle\alpha,\alpha\rangle\}h_\alpha, \tag{4.1.3}$$

and let E_α, $E_{-\alpha}$ be elements of $\mathcal{L}_\alpha$, $\mathcal{L}_{-\alpha}$ respectively such that

$$B(E_\alpha, E_{-\alpha}) = 2/\langle\alpha,\alpha\rangle. \tag{4.1.4}$$

Then, from (1.1.6), (1.1.11), (1.1.13), (4.1.3) and (4.1.4), we get

$$\left.\begin{aligned} [H_\alpha, E_\alpha] &= 2E_\alpha, \\ [H_\alpha, E_{-\alpha}] &= -2E_{-\alpha}, \\ [E_\alpha, E_{-\alpha}] &= H_\alpha. \end{aligned}\right\} \tag{4.1.5}$$

We shall make an extensive use of the commutation relations (4.1.5) in the discussion of matrix representation in §4.4.2.

The operators $E_\alpha, E_{-\alpha}$ can be identified with the familiar raising and lowering operators from the angular momentum theory. The correspondence is

$$H \leftrightarrow H_\alpha,\ E_+ \leftrightarrow E_\alpha,\ E_- \leftrightarrow E_{-\alpha}. \quad (4.1.6)$$

Suppose that α,β and $\alpha+\beta \in \Delta$, and let e_α, e_β and $e_{\alpha+\beta}$ be basis elements of $\mathcal{L}_\alpha, \mathcal{L}_\beta$, and $\mathcal{L}_{\alpha+\beta}$ respectively. Theorem 4.1 then implies that there exists a complex number $N_{\alpha,\beta}$ such that

$$[e_\alpha, e_\beta] = N_{\alpha,\beta} e_{\alpha+\beta}. \quad (4.1.7)$$

The properties of the $N_{\alpha,\beta}$ are given in the following theorems.

Theorem 4.24

If α,β and $\alpha+\beta \in \Delta$, then $N_{\alpha,\beta} \neq 0$.

Theorem 4.25

Let $N_{\alpha,\beta}$ be the structure constant defined in (4.1.7) and let $B(e_\alpha, e_{-\alpha}) = B_\alpha$ $(\alpha \in \Delta)$. Then

(i) $$N_{\beta,\alpha} = -N_{\alpha,\beta}, \quad (4.1.8)$$

(ii) if $\alpha,\beta,\gamma \in \Delta$ and $\alpha+\beta+\gamma = 0$, then

$$N_{\alpha,\beta}B_\gamma = N_{\beta,\gamma}B_\alpha = N_{\gamma,\alpha}B_\beta, \quad (4.1.9)$$

(iii) if $\alpha,\beta,\gamma,\delta \in \Delta$ are such that the sum of any two of them is zero, and if $\alpha+\beta+\gamma+\delta = 0$, then

$$N_{\alpha,\beta}N_{\gamma,\delta}B_{\alpha+\beta} + N_{\beta,\gamma}N_{\alpha,\delta}B_{\beta+\gamma} + N_{\gamma,a}N_{\beta,\delta}B_{a+\gamma} = 0, \quad (4.1.10)$$

(iv) for any $\alpha,\beta \in \Delta$

$$N_{\alpha,\beta}N_{-\alpha,-\beta} = -\tfrac{1}{2}\langle\alpha,\alpha\rangle\{B_\alpha B_\beta / B_{\alpha+\beta}\}q(p+1), \quad (4.1.11)$$

where p and q are such that the α-string containing β is $\beta-p\alpha,\ldots,\beta,\ldots,\beta+q\alpha$.

Theorem 4.26

With $B(e_\alpha, e_{-\alpha})$ taking any assigned value B_α for each pair of roots α and $-\alpha$ of Δ, the basis elements of $\mathcal{L}$ may be chosen so that either $N_{\alpha,\beta} = N_{-\alpha,-\beta}$ for all $\alpha,\beta \in \Delta$ or $N_{\alpha,\beta} = -N_{-\alpha,-\beta}$ for all $\alpha,\beta \in \Delta$.

Both of the choices $N_{\alpha,\beta} = N_{-\alpha,-\beta}$ or $N_{\alpha,\beta} = -N_{-\alpha,-\beta}$ are allowed for any arbitrary chosen set of values for the quantities $B_\alpha = B(e_\alpha, e_{-\alpha})$. If $N_{\alpha,\beta} = N_{-\alpha,-\beta}$ then (4.1.11) gives

$$\{N_{\alpha,\beta}\}^2 = -\frac{1}{2}\langle\alpha,\alpha\rangle\{B_\alpha B_\beta / B_{\alpha+\beta}\} q(p+1), \tag{4.1.12}$$

whereas with $N_{\alpha,\beta} = -N_{-\alpha,-\beta}$ (4.1.11) gives

$$\{N_{\alpha,\beta}\}^2 = +\frac{1}{2}\langle\alpha,\alpha\rangle\{B_\alpha B_\beta / B_{\alpha+\beta}\} q(p+1). \tag{4.1.13}$$

Many different choices of $N_{\alpha,\beta}$ and B_α are made in the mathematical literature [47,54]. The most commonly used, which has the advantage of keeping $N_{\alpha,\beta}$ real, is to take B_α to be

$$B_\alpha = B(e_\alpha, e_{-\alpha}) = -1, \tag{4.1.14}$$

for all pairs α and $-\alpha$ of Δ, and for all $\alpha,\beta \in \Delta$ we take

$$N_{-\alpha,-\beta} = N_{\alpha,\beta}. \tag{4.1.15}$$

With this convention, the $N_{\alpha,\beta}$ are all real and (1.1.13) gives

$$[e_\alpha, e_{-\alpha}] = -h_\alpha, \tag{4.1.16}$$

(4.1.9) becomes

$$N_{\alpha,\beta} = N_{\beta,\gamma} = N_{\gamma,\alpha}, \tag{4.1.17}$$

(4.1.10) becomes

$$N_{\alpha,\beta}N_{\gamma,\delta} + N_{\beta,\gamma}N_{\alpha,\delta} + N_{\gamma,\alpha}N_{\beta,\delta} = 0, \tag{4.1.18}$$

and finally (4.1.11) takes the form

$$\{N_{\alpha,\beta}\}^2 = \frac{1}{2}\langle\alpha,\alpha\rangle q(p+1). \tag{4.1.19}$$

Previously we derived the commutation relations (4.1.5) of the A_1 algebra, defining a basis $H_\alpha, E_\alpha, E_{-\alpha}$ (relations (4.1.3), (4.1.4)). In the general case of a Lie algebra $\mathcal{L}$, the relation between the basis elements $e_\alpha, e_{-\alpha}$ of $\mathcal{L}_\alpha$ and $\mathcal{L}_{-\alpha}$ satisfying (4.1.14), and to the basis elements of $\mathcal{L}_\alpha, \mathcal{L}_{-\alpha}$ denoted by $E_\alpha, E_{-\alpha}$ satisfying (4.1.4), is given by the set of equations

$$\left.\begin{aligned} E_\alpha &= \{2/\langle\alpha,\alpha\rangle\}^{1/2} e_\alpha, \\ E_{-\alpha} &= -\{2/\langle\alpha,\alpha\rangle\}^{1/2} e_{-\alpha}. \end{aligned}\right\} \tag{4.1.20}$$

This new basis $H_\alpha, E_\alpha, E_{-\alpha}$ has the advantage of carrying the A_1-subalgebra structure to any Lie algebra $\mathcal{L}$.

The elements $\underset{\sim}{h}_\alpha$ of H, in the case of matrices, may be taken to be diagonal Hermitean matrices, while for each pair α and $-\alpha$ of Δ the matrices $\underset{\sim}{e}_\alpha, \underset{\sim}{e}_{-\alpha}$ may be chosen so that

$$\underset{\sim}{e}_{-\alpha} = -\underset{\sim}{e}_\alpha^{+}, \tag{4.1.21}$$

and, correspondingly,

$$\underset{\sim}{E}_{-\alpha} = \underset{\sim}{E}_\alpha^{+}. \tag{4.1.22}$$

§4.4.2 A method of constructing matrix elements of irreducible representations of a simple Lie algebra

A matrix representation of a simple Lie algebra can be completely specified if we know the matrices representing the elements of the Cartan subalgebra H_α, and the matrices representing the elements $E_\alpha, E_{-\alpha}$ of the $\mathcal{L}_\alpha, \mathcal{L}_{-\alpha}$ respectively, for every simple root α. Then, from (4.1.7), all the other matrices $\underset{\sim}{\Gamma}(E_\alpha), \underset{\sim}{\Gamma}(E_{-\alpha})$, $\alpha \in \Delta$, can be constructed using Theorem 4.25 (with the set of our conventions (4.1.14) and (4.1.15)).

There is a straightforward method for the construction of diagonal matrices representing H_α, when the weight system is known. We defined the weights (§4.2.1) as the eigenvalues of the operator $W(h)$, $h \in H$, of the basis $\psi_1, \psi_2, \ldots, \psi_d$. This implies that the diagonal elements are given by

$$\Gamma(h)_{jj} = \lambda_j(h), \tag{4.2.1}$$

where $h \in H$.

In the basis defined in §4.4.1 the equation (4.2.1) (if we omit the position index j) becomes

$$\underset{\sim}{\Gamma}(H_\alpha) = \lambda(H_\alpha) = \{2/\langle\alpha,\alpha\rangle\}\lambda(h_\alpha) = 2\langle\lambda,\alpha\rangle/\langle\alpha,\alpha\rangle. \qquad (4.2.2)$$

In determining the $\underset{\sim}{\Gamma}(e_\alpha)$, a simple observation saves us from a lot of work. From (1.1.6) we have

$$[h,e_\alpha] = \alpha(h)e_\alpha$$

or

$$[\underset{\sim}{\Gamma}(h),\underset{\sim}{\Gamma}(e_\alpha)] = \alpha(h)\underset{\sim}{\Gamma}(e_\alpha).$$

In particular, the pq element is

$$\{\Gamma(h)_{pp} - \Gamma(h)_{qq} - \alpha(h)\}(\Gamma(e_\alpha))_{pq} = 0. \qquad (4.2.3)$$

The relation (4.2.3) tells us that $(\Gamma(e_\alpha))_{pq} \neq 0$ only if

$$\Gamma(h)_{pp} - \Gamma(h)_{qq} = \alpha(h), \qquad (4.2.4)$$

ie $\underset{\sim}{\Gamma}(e_\alpha) \neq 0$ if the difference between the p^{th} weight and the q^{th} weight is $\alpha(h)$.

Having constructed the matrices representing the elements of the Cartan subalgebra, using (4.2.2), we can partition these matrices into blocks according to their A_1-subalgebra content. If all the weights are simple, then there is a unique block form of these diagonal matrices. In general, however, some of the weights of an irreducible representation have multiplicity greater than one. In this case if we try to bring then to an A_1-subalgebra block form, there is the difficulty of deciding which element belongs to each A_1-subalgebra, due to the weight multiplicity. A specific example will elucidate our discussion.

Example

Let us consider the $\underset{\sim}{7}$ and $\underset{\sim}{27}$ dimensional representations of G_2. From Table 4.4, we observe that the multiplicity of all the weights of the $\underset{\sim}{7}$ is equal to one, while in the $\underset{\sim}{27}$ dimensional representation some of the weights have multiplicity greater than one. Using (4.2.2) we find (in the basis h_α)

$$\left.\begin{aligned} h^7_{\alpha_1} &= \text{diag}(0,3/2,-3/2,0,3/2,-3/2,0), \\ h^7_{\alpha_2} &= \text{diag}(1/2,-1/2,1,0,-1,1/2,-1/2). \end{aligned}\right\} \quad (4.2.5)$$

$$\left.\begin{aligned} h^{27}_{\alpha_1} &= \text{diag}(0,3/2,3,-3/2,0,0,3/2,3,-3,-3/2,0,0,0,3/2, \\ &\qquad 3/2,3,-3,-3/2,-3/2,0,0,3/2,-3,-3/2,0), \\ h^{27}_{\alpha_2} &= \text{diag}(1,0,-1,3/2,1/2,1/2,-1/2,-1/2,-3/2,2,1,1,0, \\ &\qquad 0,0,-1,-1,-2,-3/2,1/2,1/2,-1/2,-1/2,-3/2,1,0,-1). \end{aligned}\right\} \quad (4.2.6)$$

To find their A_1-subalgebra content we consider the H_α basis given by the relation (4.1.3)

$$H_\alpha = \frac{2}{\langle\alpha,\alpha\rangle} h_\alpha .$$

As for the normalization of the roots, we take the following:

$$\langle\alpha_1,\alpha_1\rangle = 3, \quad \langle\alpha_2,\alpha_2\rangle = 1. \qquad (4.2.7)$$

The connection between the ζ_3 generator of the angular momentum and H_α of the SU(2) algebra is given by

$$\frac{1}{2} H_\alpha = \zeta_3 . \qquad (4.2.8)$$

The generators H_α become after substituting (4.2.7), (4.2.8), (4.2.5) and (4.2.6) to (4.1.3)

$$\left.\begin{aligned} \Gamma^7(H_{\alpha_1}) &= \text{diag}(0,1/2,-1/2,0,1/2,-1/2,0), \\ \Gamma^7(H_{\alpha_2}) &= \text{diag}(1/2,-1/2,1,0,-1,1/2,-1/2). \end{aligned}\right\} \quad (4.2.9)$$

$$\left.\begin{aligned}\Gamma^{27}(H_{\alpha_1}) &= \mathrm{diag}(0,1/2,1,-1/2,0,0,1/2,1/2,1,-1,-1,\\ &\qquad -1/2,-1/2,0,0,0,1/2,1/2,-1,-1,-1/2,-1/2,\\ &\qquad 0,0,1/2,-1,-1/2,0),\\ \Gamma^{27}(H_{\alpha_2}) &= \mathrm{diag}(1,0,-1,3/2,1/2,1/2,-1/2,-1/2,-3/2,\\ &\qquad 2,1,1,0,0,0,-1,-1,-2,3/2,1/2,-1/2,-3/2,\\ &\qquad 1,0,-1).\end{aligned}\right\}\qquad (4.2.10)$$

In the case of the $\underset{\sim}{7}$ dimensional representation, the diagonal generators $\Gamma(H_{\alpha_1}),\Gamma(H_{\alpha_2})$, with the help of (4.2.3), can be written in their A_1-subalgebra content as follows:

$$\left.\begin{aligned}\Gamma^{7}(H_{\alpha_1}) &= \mathrm{diag}(0\,;\underbrace{1/2,-1/2}_{A_1\text{-doublet}};0\,;\underbrace{1/2,-1/2}_{A_1\text{-doublet}};0),\\ \Gamma^{7}(H_{\alpha_2}) &= \mathrm{diag}(\underbrace{1/2,-1/2}_{A_1\text{-doublet}}\,;\underbrace{1\,,0\,,1}_{A_1\text{-triplet}}\,;\underbrace{1/2,-1/2}_{A_1\text{-doublet}}).\end{aligned}\right\}\qquad (4.2.11)$$

In the case of the $\underset{\sim}{27}$ dimensional representation, $\Gamma^{27}(H_{\alpha_2})$, for example, can be blocked as follows:

$$\left.\begin{aligned}\Gamma^{27}(H_{\alpha_2}) &= \mathrm{diag}(\underbrace{1,0,-1}_{A_1\text{-triplet}}\,;\underbrace{3/2,1/2,1/2,-1/2,-1/2,-3/2}_{\substack{A_1\text{-tetraplet}\\ A_1\text{-doublet}}}\,;\\ &\qquad \underbrace{2,1,1,0,0,0,-1,-1,2}_{\substack{A_1\text{-pentaplet}\\ A_1\text{-triplet}\\ A_1\text{-singlet}}}\,;\\ &\qquad \underbrace{3/2,1/2,1/2,\ \ldots\ -1/2,-1/2,-3/2}_{\substack{A_1\text{-tetraplet}\\ A_1\text{-doublet}}}\,;\underbrace{1,0,-1}_{A_1\text{-triplet}}).\end{aligned}\right\}\qquad (4.2.12)$$

The lines in (4.2.12) indicate the ambiguity of assigning the eigenvalues to a particular multiplet according to (4.2.3).

Note

In the SU(2) theory we know that every representation D(j) is specified by the eigenvalue j, and each state of the multiplet is characterized by the eigenvalue m, taking values in the range $-j \leq m \leq +j$, ie $(2j+1)$ values in all. On the other hand, each state of the multiplet is obtained from the state ψ_{jm} by the application of the raising and lowering operators

$$\zeta_{\pm}\psi_{jm} = \pm\sqrt{(j\mp m)(j\pm m+1)}\ \psi_{jm\pm 1}. \tag{4.2.13}$$

To construct the matrices representing $\underset{\sim}{\Gamma}(e_\alpha)$ we shall use again the information from their A_1-subalgebra content. From the difference of the weights we can find the non-zero elements of $\underset{\sim}{\Gamma}(e_\alpha)$, while from their A_1-subalgebra content their magnitudes. The method works very well when the multiplicity is equal to one. For multiplicity greater than one, the magnitudes of the non-zero elements can not be fixed unambigously, because we do not know which state belongs to each A_1-subalgebra.

Let us return to the above example. The matrix $\underset{\sim}{\Gamma}^7(E_{\alpha_2})$, after the change of basis, can be written, according to its A_1-subalgebras:

$$\underset{\sim}{\Gamma}^7(E_{\alpha_2}) = \begin{pmatrix} \boxed{A_1\text{-doublet}} & & \\ & \boxed{A_1\text{-triplet}} & \\ & & \boxed{A_1\text{-doublet}} \end{pmatrix} . \tag{4.2.14}$$

From the known matrices representing A_1-doublet, and A_1-triplet (4.2.14) becomes

$$\underset{\sim}{\Gamma}^7(E_{\alpha_2}) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \qquad (4.2.15)$$

Similarly for $\underset{\sim}{\Gamma}(E_{\alpha_1})$

$$\underset{\sim}{\Gamma}^7(E_{\alpha_1}) = \begin{bmatrix} 0 & & & & \\ & \boxed{A_1\text{-doublet}} & & & \\ & & 0 & & \\ & & & \boxed{A_1\text{-doublet}} & \\ & & & & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \qquad (4.2.16)$$

For $\underset{\sim}{\Gamma}^{27}(E_{\alpha_2})$ we have

$$\underset{\sim}{\Gamma}^{27}(E_{\alpha_2}) = \begin{pmatrix} \boxed{A_1\text{-triplet}} & & & & \\ & \boxed{\begin{matrix} A_1\text{-tetraplet} \\ A_1\text{-triplet} \end{matrix}} & & & \\ & & \boxed{\begin{matrix} A_1\text{-pentaplet} \\ A_1\text{-triplet} \\ A_1\text{-singlet} \end{matrix}} & & \\ & & & \boxed{\begin{matrix} A_1\text{-tetraplet} \\ A_1\text{-triplet} \end{matrix}} & \\ & & & & \boxed{A_1\text{-triplet}} \end{pmatrix} \tag{4.2.17}$$

To resolve the multiplicity problem, let us recall how this problem is solved in the case of the mesonoctet (the eight-fold way [15], SU(3) Lie algebra). The weight diagram for the $\underset{\sim}{8}$ dimensional representation of A_2 is given in Figure 4.1.

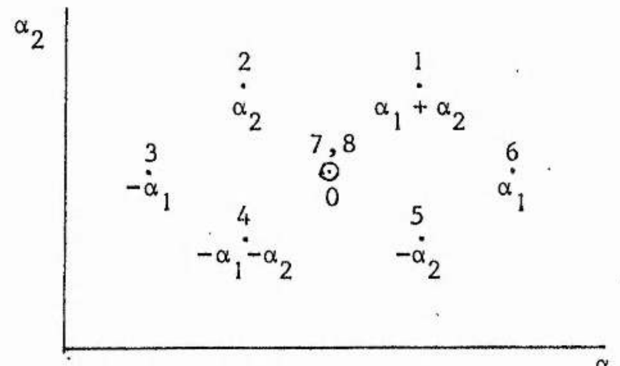


Figure 4.1: The weight diagram of the $\underset{\sim}{8}$ of A_2

There are various A_1-subalgebras corresponding to U,V,I spin. Let us suppose that the state ψ_7 belongs to an α_2-triplet, ie (ψ_2,ψ_7,ψ_5), and ψ_8 is an α_2-singlet. Thus

$$\left.\begin{aligned} E_{\alpha_2}\psi_7 &= \sqrt{2}\psi_2 \quad \text{(from (4.2.13))} \\ E_{\alpha_2}\psi_8 &= 0. \end{aligned}\right\} \tag{4.2.18}$$

As $E_{\alpha_1}\psi_7$ and $E_{\alpha_1}\psi_8$ are both proportional to ψ_6, let

$$\left.\begin{aligned} E_{\alpha_1}\psi_7 &= a\psi_6 \\ E_{\alpha_1}\psi_8 &= b\psi_6. \end{aligned}\right\} \tag{4.2.19}$$

As the representation of A_2 forms a representation of the compact real form SU(3), and as this may be integrated to form a representation of SU(3) [54], which may be taken to be a unitary representation, the matrices representing $E_\alpha, E_{-\alpha}$ may be chosen correspondingly so that ((4.1.22))

$$\underset{\sim}{\Gamma}(E_\alpha) = \underset{\sim}{\Gamma}(E_{-\alpha})^+. \tag{4.2.20}$$

Furthermore, they can be chosen to be real, so that

$$\underset{\sim}{\Gamma}(E_\alpha) = \underset{\sim}{\tilde{\Gamma}}(E_{-\alpha}). \tag{4.2.21}$$

In this particular example the following result would be valid:

$$\left.\begin{aligned} \Gamma(E_{\alpha_1})_{67} &= a,\ \Gamma(E_{\alpha_1})_{j7} = 0,\ j \neq 6 \\ \Gamma(E_{\alpha_1})_{68} &= b,\ \Gamma(E_{\alpha_1})_{j8} = 0,\ j \neq 6. \end{aligned}\right\} \tag{4.2.22}$$

There would be complex numbers λ and μ such that

$$E_{-\alpha_1}\psi_6 = \mu\psi_7 + \lambda\psi_8. \tag{4.2.23}$$

This implies that, for the $\underset{\sim}{8}$,

$$\left.\begin{aligned} \Gamma(E_{-\alpha_1})_{76} &= \mu,\ \Gamma(E_{-\alpha_1})_{j6} = 0,\ j \neq 7,8 \\ \Gamma(E_{-\alpha_1})_{86} &= \lambda, \end{aligned}\right\} \tag{4.2.24}$$

Thus, by (4.2.21), (4.2.22) and (4.2.24), give

$$a = \mu,\ b = \lambda. \tag{4.2.25}$$

Hence, (4.2.23) becomes

$$E_{-\alpha_1}\psi_6 = a\psi_7 + b\psi_8. \tag{4.2.26}$$

Now, acting on (4.2.26) with E_{α_2}, as $[E_{-\alpha_1}, E_{\alpha_2}] = 0$, from

$$E_{\alpha_2}E_{-\alpha_1}\psi_6 = aE_{\alpha_2}\psi_7 + bE_{\alpha_2}\psi_8,$$

we get

$$E_{-\alpha_1}E_{\alpha_2}\psi_6 = aE_{\alpha_2}\psi_7 + bE_{\alpha_2}\psi_8.$$

By (4.2.18) and the fact that $E_{\alpha_2}\psi_6 = \psi_1$, $E_{-\alpha_1}\psi_1 = \psi_2$,

$$\psi_2 = a\sqrt{2}\psi_2.$$

So,

$$a = 1/\sqrt{2}. \tag{4.2.27}$$

To calculate the other coefficient b we use the identity $[E_{\alpha_1}, E_{-\alpha_1}] = H_{\alpha_1}$ acting on ψ_6,

$$E_{\alpha_1}E_{-\alpha_1}\psi_6 - E_{-\alpha_1}E_{\alpha_1}\psi_6 = H_{\alpha_1}\psi_6. \tag{4.2.28}$$

But $E_{\alpha_1}\psi_6 = 0$, and $H_{\alpha_1}\psi_6 = \frac{2<\alpha_1,\alpha_1>}{<\alpha_1,\alpha_1>}\psi_6 = 2\psi_6$ (from (4.2.2)).

Substituting (4.2.19) and (4.2.26) to (4.2.28), we get

$$b = \pm\sqrt{\frac{3}{2}}. \tag{4.2.29}$$

A generalization of this method will remove the ambiguities in (4.2.17).

In the weight diagram (Figure 4.2) we have enumerated the weights according to their lexicographical order. The vertical lines join pairs of eigenvectors ψ_λ and $\psi_{\lambda'}$ such that $(\psi_\lambda, E_{-\alpha_2}\psi_{\lambda'}) \neq 0$, while the horizontal lines join pairs $\psi_\lambda, \psi_{\lambda'}$ of eigenvectors such that $(\psi_\lambda, E_{-\alpha_1}\psi_{\lambda'}) \neq 0$. The direction of a loop inside the diagram indicates the way we apply the commutation relation $[E_{-\alpha}, E_{+\beta}] = 0$, $\alpha \neq \beta$ simple roots.

As in the case of the $\underset{\sim}{8}$ of A_2, we choose the multiplets in the α_2-direction. We have chosen the states ψ_5, ψ_7 to belong to the tetraplet $(\psi_4, \psi_5, \psi_7, \psi_9)$, while the states ψ_6, ψ_8 are chosen such as to form a α_2-doublet (ψ_6, ψ_8); the states $\psi_{10}, \psi_{11}, \psi_{13}, \psi_{16}, \psi_{18}$ form a pentaplet; the states $\psi_{12}, \psi_{14}, \psi_{17}$ form a triplet; the state ψ_{15} is a singlet; the states $\psi_{19}, \psi_{20}, \psi_{22}, \psi_{24}$ form a tetraplet, and finally the states ψ_{21}, ψ_{23} form a doublet. This choice of the multiplets fixes the matrix representation $\Gamma^{27}(E_{\alpha_2})_{ij}$.

To fix the matrix elements of the generator E_{α_1}, we must consider all the possible loops of Figure 4.2(a). We shall give here the calculation of the first loop $(2,3,5,4)_{\text{loop}}$, while the results of the calculation of the $\Gamma^{27}(E_{\alpha_1}), \Gamma^{27}(E_{\alpha_2})$ will be given in §4.3.1 as an output of a computer program implementation of the above method.

As in the case of the $\underset{\sim}{8}$ dimensional representation of A_2, we start with the relation

$$E_{-\alpha_1}\psi_3 = a\psi_5 + b\psi_6. \tag{4.2.30}$$

The action of E_{α_2} on (4.2.30), as $[E_{-\alpha_1}, E_{\alpha_2}] = 0$, will give

$$\begin{aligned} E_{\alpha_2}E_{-\alpha_1}\psi_3 &= aE_{\alpha_2}\psi_5 + bE_{\alpha_2}\psi_6 \\ &= \sqrt{3}a\psi_5 \quad \text{(from our previous choice and (4.2.13))} \end{aligned}$$

$$\begin{aligned} E_{-\alpha_1}E_{\alpha_2}\psi_3 &= E_{-\alpha_1}(\sqrt{2}\psi_2) \\ &= \sqrt{2}\psi_4 \quad \text{(because } E_{-\alpha_1}\psi_2 = \psi_4 \text{ from (4.2.13)).} \end{aligned}$$

Thus

$$a = \sqrt{\frac{2}{3}}. \tag{4.2.31}$$

From $[E_{\alpha_1}, E_{-\alpha_1}] = H_{\alpha_1}$, we have, when applied on ψ_3,

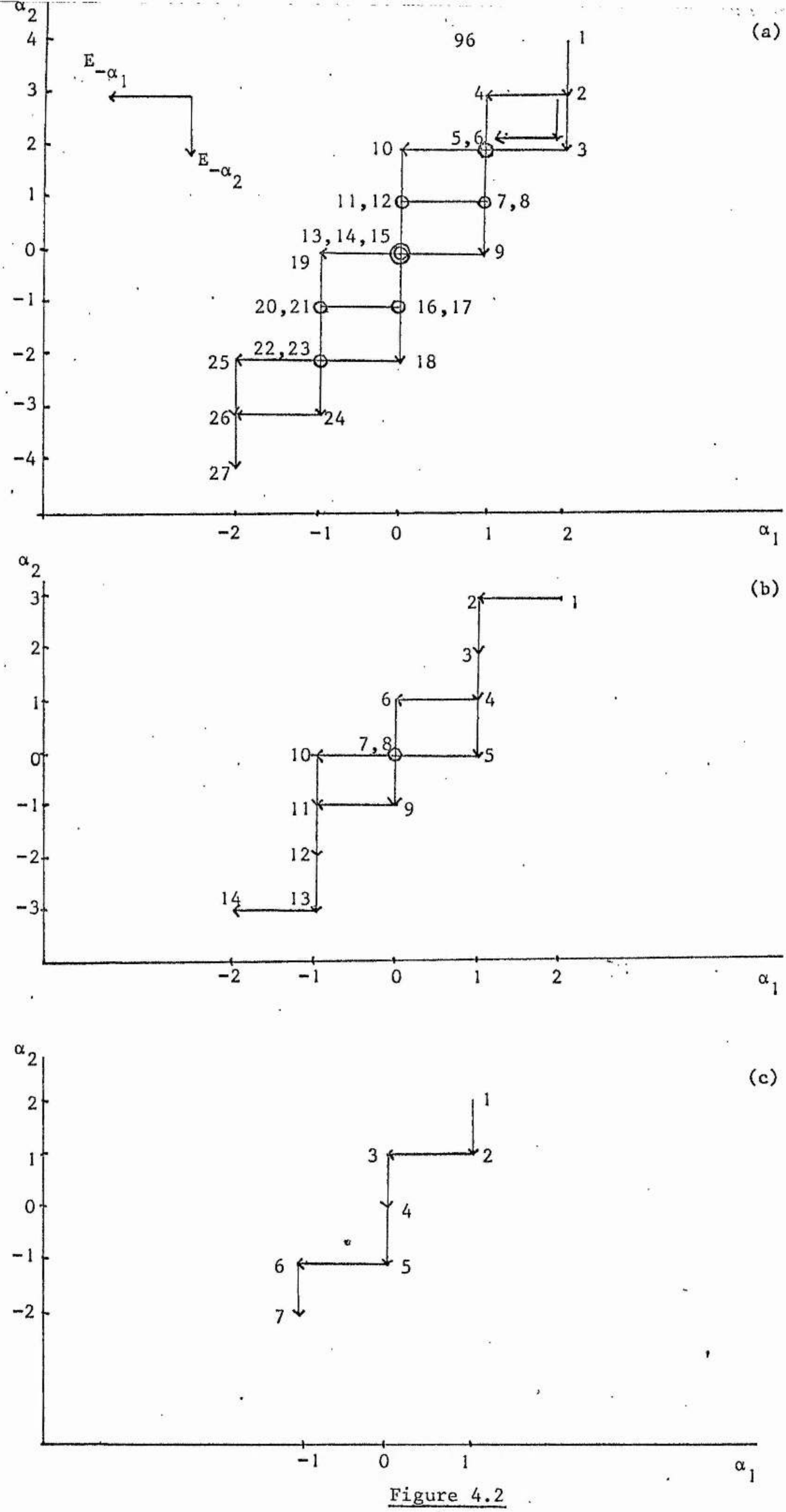

(a)
(b)
(c)
α_2
α_1
$E_{-\alpha_1}$
$E_{-\alpha_2}$
5,6
7,8
11,12
13,14,15
16,17
20,21
22,23

Figure 4.2

$$E_{\alpha_1} E_{-\alpha_1} \psi_3 - E_{-\alpha_1} E_{\alpha_1} \psi_3 = H_{\alpha_1} \psi_3$$

$$\Rightarrow E_{\alpha_1} (a\psi_5 + b\psi_6) = 2\frac{(2\alpha_1 + 2\alpha_2, \alpha_1)}{(\alpha_1, \alpha_1)} \psi_3$$

$$\Rightarrow (a^2+b^2)\psi_3 = [2[\frac{2(\alpha_1,\alpha_1)}{(\alpha_1,\alpha_1)}] + 2[\frac{2(\alpha_2,\alpha_1)}{(\alpha_1,\alpha_1)}]]\psi_3$$

$$\Rightarrow (a^2+b^2)\psi_3 = [4 + 2(-1)]\psi_3$$ (from the Cartan matrix of G_2 we have $A_{11} = 2$, $A_{21} = -1$)

$$\Rightarrow a^2 + b^2 = 2. \tag{4.2.32}$$

From (4.2.31), finally, we get the value

$$b = \frac{2}{\sqrt{3}} . \tag{4.2.33}$$

As we go to higher rank algebras, the weight system becomes more complicated, but nevertheless a careful consideration of all of the loops can fix the matrix elements. The method can be translated to an algorithm for generation of the matrix elements of a representation of any simple Lie algebra by a computer. We describe a pilot program for the case of G_2 in the next section (§4.4.3). A more complicated case is considered in Chapter 5, where the matrix elements of the $\underset{\sim}{126}$, $\underset{\sim}{120}$, $\underset{\sim}{16}$ of SO(10) are calculated.

§4.4.3 Programs

An explicit matrix representation $\underset{\sim}{\Gamma}(E_\alpha)$ of the three-dimensional A_1 Lie algebra can be easily constructed. Each irreducible representation is characterised by the value of j. We have developed a simple program (Program D1(4)) for generating the matrix elements of any representation of the A_1 Lie algebra.

Description of Program D1(4)

Program D1(4) is based upon the formula (4.2.13). It includes a recursive method for determining the eigenvalue m, and comes to an end when m takes the maximum value $2j + 1$. The input of the program is the value of j, and the output the matrix representing D(j) in a (2j+1,2j+1) array.

The next program (Program D2(4)) generates the diagonal generators of $\mathcal{L}$. There are two versions of that program: D2(4) which includes B3(2) as a subprogram, and D3(4) where the weights are given as input.

Description of Program D2(4)

It is based upon the formula (4.2.2). The roots and the weights are generated and consequently used for the diagonal matrices. The input is as in Program B3(2), and in the output we get the diagonal generators $H_{\alpha_1}, H_{\alpha_2}, \ldots, H_{\alpha_\ell}$.

Description of Program D3(4)

Using the various symmetries of the weight diagrams (rotation, reflection), we reduce the space required to store the weights. After this reduction we use the weights as input.

Any program for calculating the matrix elements of a generator E_α of $\mathcal{L}_\alpha$ should be constructed in such a way that the storage problem is minimized. There is a simple technique in computing science called the 'sparse array technique' which reduces the storage considerably. If an (n,n) array has most of its elements zero, then we can store only the non-zero elements and their position coordinates. Then, the array is reduced to an (n',3) array, where n' is the number of rows where a non-zero element exists. We employed this technique when developing Program D4(4) which generates the matrix elements of $\underset{\sim}{\Gamma}^n(E_{\alpha_1}), \underset{\sim}{\Gamma}^n(E_{\alpha_2})$ of G_2, for n = 27, 14, 7.

Description of Program D4(4)

This program implements the method discussed in §4.4.2. The program mainly consists of three procedures, procedure VERTICAL, procedure HORIZONTAL and procedure LOOP. The call of the procedure VERTICAL fixes $\underset{\sim}{\Gamma}(E_{\alpha_2})$ by an arbitrary choice of the α_2-multiplets. With the call of the procedure HORIZONTAL the differences of the weights are calculated. If these differences are equal to the simple root α_1, then an A_1-multiplet is formed. If the multiplicity of a state belonging to that multiplet is equal to one, then the procedure LOWER (which is the lowering operator formula (4.2.9)) calculates the corresponding matrix element. If the multiplicity is greater than one (the multiplicity of a state is indicated by the variable MULT), then the procedure LOOP is called and performs similar calculations as in (4.2.14) - (4.2.16), taking into consideration the orthogonality relations of the A_1-subalgebras. The program ends when the value of the variable LAST (which controls the number of the states, belonging to the representation N) reaches the dimension d of the representation.

The input of the program is:
the rank of the algebra; the dimension of the representation; the maximum weights multiplicity of the representation increased by one; the weights; the simple roots.

The output consists of an array with three columns. In the first two columns the coordinates of a non-zero matrix element are given, while in the third its value is stated.

In Table 4.15 are the results from the calculation of the $\underset{\sim}{\Gamma}^{27}(E_{\alpha_i}), \underset{\sim}{\Gamma}^{14}(E_{\alpha_i}), \underset{\sim}{\Gamma}^{7}(E_{\alpha_i})$, i = 1,2 of the algebra G_2 using Program D4(4).

Program D4(4) is a pilot program for a more complicated and more sophisticated program which will be developed in Chapter 5, for the case of D_5.

To make sure that Program D4(4) generates correctly the required matrix elements, a test program (Program D6(4)) has been

a) 27

THE GENERATOR EA1 IS

****

2	4	1.00
3	5	0.82
5	10	0.82
3	6	1.15
6	10	1.15
7	11	0.82
7	12	0.58
8	11	0.58
8	12	-0.82
9	13	0.58
13	19	0.58
9	14	0.71
14	19	0.71
9	15	1.08
15	19	1.08
16	20	0.82
16	21	0.58
17	20	0.58
17	21	-0.82
18	22	0.82
22	25	0.82
18	23	1.15
23	25	1.15
24	26	1.00

THE GENERATOR EA2 IS

****

1	2	1.41
2	3	1.41
4	5	1.73
5	7	2.00
6	8	1.00
7	9	1.73
10	11	2.00
11	13	2.45
12	14	1.41
13	16	2.45
14	17	1.41
16	18	2.00
19	20	1.73
20	22	2.00
21	23	1.00
22	24	1.73
25	26	1.00

b) 14

THE GENERATOR EA1 IS

****

1	2	1.00
4	6	1.00
5	7	1.22
7	10	1.22
5	8	0.71
8	10	0.71
9	11	1.00
13	14	1.00

THE GENERATOR EA2 IS

****

2	3	1.73
3	4	2.00
4	5	1.73
6	7	1.41
7	9	1.41
10	11	1.73
11	12	2.00
12	13	1.73

c) 7

THE GENERATOR EA1 IS

****

2	3	1.00
5	6	1.00

THE GENERATOR EA2 IS

****

1	2	1.00
3	4	1.41
4	5	1.41
6	7	1.00

Table 4.15

developed which verifies the commutation relations of the relevant algebra. Using this program, we verified the commutation relations of the G_2 algebra for the above representations.

§4.5 Clebsch-Gordan Coefficients

Having constructed an explicit matrix realization of a simple complex Lie algebra $\mathcal{L}$, we are now in a position to attack the Clebsch-Gordan coefficients problem.

In the general case to distinguish the multiplicity of the weights from the multiplicity of the irreducible representations entering the Clebsch-Gordan series, we use the term 'internal multiplicity' for the weights multiplicity, while we keep the unqualified term 'multiplicity' for the Clebsch-Gordan series.

The Clebsch-Gordan coefficients theory for the SU(2) group is well developed. For the SU(3) group detailed calculations are given by De Swart in the context of the octet model [62]. The group SU(2) differs from the other ones in that its irreducible representations appear in the tensor product of two irreducible representations with multiplicity not exceeding one. If the multiplicity of a representation is greater than one, then considerable complications arise in the Clebsch-Gordan coefficients theory [63,64]. In the course of our study, we shall discuss how this problem can be solved.

This section includes the definition of the terms we use (§4.5.1), the method of evaluating the Clebsch-Gordan coefficients (§4.5.2), and the computer implementation of the method in the case of the tensor product $\underset{\sim}{7} \otimes \underset{\sim}{7}$ of G_2 (§4.5.3).

§4.5.1 The theory of Clebsch-Gordan coefficients

Suppose we have the basis functions of two unitary irreducible representations of a simple compact Lie group. We denote these functions by $\psi_{\mu}^{(a)}$ and $\psi_{v}^{(b)}$, where a and b stand for all numbers necessary to specify the first and second representations, and μ and v stand for all numbers which differentiate among the different states within these representations. Following our previous development, we can say that μ and v represent two things: the weight systems of the two representations, and to differentiate the states with the same weight, when the internal multiplicity of the representation is greater than one, the lexicographical ordering of the weights. The indices a and b can represent the dimensions of the representations.

If we take the direct product of these two basis functions, we obtain the basis tensors of the direct product space, given by $\psi_{\mu}^{(a)} \otimes \psi_{v}^{(b)}$.

In general, these product basis tensors are not the basis tensors of an irreducible representation.

Theorem 4.26

The basis tensors of any irreducible representation contained in the direct product can be written as a linear combination of the product tensors. If we denote the irreducible tensors contained in the direct product by $\psi_{m}^{(j)}$, then we have

$$\psi_{m}^{(j)} = \sum_{\mu v} \left(\begin{matrix} a & b \\ \mu & v \end{matrix} \middle| \begin{matrix} j \\ m \end{matrix}\right) \psi_{\mu}^{(a)} \otimes \psi_{v}^{(b)} . \tag{5.1.1}$$

Following (5.1.1), we define

Definition: Clebsch-Gordan coefficients

The coefficients entering the sum in (5.1.1) are called Clebsch-Gordan coefficients.

However, in general, the direct product space is not simply reducible, so that another index γ must enter the relation (5.1.1) to differentiate the representations with multiplicity greater than one.

Then (5.1.1) can be written

$$\psi_m^{(j\gamma)} = \sum_{\mu\nu} \left(\begin{matrix} a & b \\ \mu & \nu \end{matrix} \middle| \begin{matrix} j,\gamma \\ m \end{matrix}\right) \psi_\mu^{(a)} \psi_\nu^{(b)}. \quad (5.1.2)$$

Theorem 4.26 also implies that the product bases functions can be written as linear combinations of the basis functions of the irreducible representations

$$\psi_\mu^{(a)} \psi_\nu^{(b)} = \sum_{\mu\nu} \left(\begin{matrix} j,\gamma \\ m \end{matrix} \middle| \begin{matrix} a & b \\ \mu & \nu \end{matrix}\right) \psi_m^{(j\gamma)}. \quad (5.1.3)$$

The following theorem gives the basic properties of the Clebsch-Gordan coefficients.

<u>Theorem 4.27</u>

If $\left(\begin{matrix} a & b \\ \mu & \nu \end{matrix} \middle| \begin{matrix} j,\gamma \\ m \end{matrix}\right)$ are the Clebsch-Gordan coefficients defined in Theorem 4.26, then they may be chosen so that

(a) $$\left(\begin{matrix} a & b \\ \mu & \nu \end{matrix} \middle| \begin{matrix} j,\gamma \\ m \end{matrix}\right) = \left(\begin{matrix} j,\gamma \\ m \end{matrix} \middle| \begin{matrix} a & b \\ \mu & \nu \end{matrix}\right)^*, \quad (5.1.4)$$

(b) $$\sum_{\mu\nu} \left(\begin{matrix} a & b \\ \mu & \nu \end{matrix} \middle| \begin{matrix} j,\gamma \\ m \end{matrix}\right) \left(\begin{matrix} j,\gamma' \\ m' \end{matrix} \middle| \begin{matrix} a & b \\ \mu & \nu \end{matrix}\right) = \delta_{jj'} \delta_{mm'} \delta_{\gamma\gamma'}, \quad (5.1.5)$$

(c) $$\sum_{jm\gamma} \left(\begin{matrix} j,\gamma \\ m \end{matrix} \middle| \begin{matrix} a & b \\ \mu & \nu \end{matrix}\right) \left(\begin{matrix} a & b \\ \mu' & \nu' \end{matrix} \middle| \begin{matrix} j,\gamma \\ m \end{matrix}\right) = \delta_{\mu\mu'} \delta_{\nu\nu'}. \quad (5.1.6)$$

Without loss of generality, we can choose the phases in (5.1.4) so that the Clebsch-Gordan coefficients are all real.

To illustrate the difficulty of defining the index γ, let us consider an example of two eight-dimensional representations of SU(3) which has the Clebsch-Gordan series

$$\underset{\sim}{8} \otimes \underset{\sim}{8} = \underset{\sim}{1} \oplus \underset{\sim}{8} \oplus \underset{\sim}{8} \oplus \underset{\sim}{10} \oplus \overline{\underset{\sim}{10}} \oplus \underset{\sim}{27}.$$

There is no group theoretical method to distinguish the two eight-dimensional representations $\underset{\sim}{\Gamma}^{8,1}$ and $\underset{\sim}{\Gamma}^{8,2}$. The Clebsch-Gordan coefficients will depend upon the basis functions of these two representations.

§4.5.2 <u>The method of calculating the Clebsch-Gordan coefficients</u>

We shall consider first the case of multiplicity one. Using one of the methods of Section 3, or running Program C7(3) for the case G_2, we find that the direct product of the lowest dimension representation of G_2 with itself, has the following Clebsch-Gordan series:

$$\underset{\sim}{7} \otimes \underset{\sim}{7} = \underset{\sim}{27} \oplus \underset{\sim}{14} \oplus \underset{\sim}{7} \oplus \underset{\sim}{1}. \tag{5.2.1}$$

We shall denote the basis functions of the representations in (5.2.1) by $\psi^7_{i,\nu}$ (i = 1,2,...,7), $\psi^{7'}_{j,\mu}$ (j = 1,2,...,7), $\psi^{27}_{k,\pi}$ (k = 1,2,..., 27), $\psi^{14}_{\ell,\lambda}$ (ℓ = 1,2,...,14), $\psi^7_{m,\omega}$ (m = 1,2,...,7) and $\psi^1_{1,(0,0)}$. The Greek indices indicate the weight systems, while the Latin indices i, j, k, ℓ, m specify the position of a state in the weight diagrams (Table 4.15). In terms of the basis functions (5.2.1) can be written

$$\psi^7_{i,\nu} \otimes \psi^{7'}_{j,\mu} = \begin{pmatrix} 7 & 7' \\ i,\nu & j,\mu \end{pmatrix} \Big| \begin{matrix} 27 \\ k,\pi \end{pmatrix} \psi^{27}_{k,\pi} + \begin{pmatrix} 7 & 7' \\ i,\nu & j,\mu \end{pmatrix} \Big| \begin{matrix} 14 \\ \ell,\lambda \end{pmatrix} \psi^{14}_{\ell,\lambda}$$

$$+ \begin{pmatrix} 7 & 7' \\ i,\nu & j,\mu \end{pmatrix} \Big| \begin{matrix} 7 \\ m,\omega \end{pmatrix} \psi^7_{m,\omega} + \begin{pmatrix} 7 & 7' \\ i,\nu & j,\mu \end{pmatrix} \Big| \begin{matrix} 1 \\ 1,(0,0) \end{pmatrix} \psi^1_{1,(0,0)}. \tag{5.2.2}$$

To calculate the Clebsch-Gordan coefficients $\left(\begin{matrix} 7 & 7' \\ i,\nu & j,\mu \end{matrix} \Big| \begin{matrix} 27 \\ k,\pi \end{matrix}\right)$, we start with the highest weight of the $\underset{\sim}{27}$ representation. In terms of the basis functions of the two 7's we have

$$\psi^{27}_{1,(2,4)} = 1 \cdot \psi^7_{1,(1,2)} \psi^{7'}_{1,(1,2)}. \tag{5.2.3}$$

From Table 4.17 the application of the operator $E^{27}_{-\alpha_2}$ on the state $\psi^{27}_{1,(2,4)}$ gives

$$E^{27}_{-\alpha_2} \psi^{27}_{1,(2,4)} = \sqrt{2}\ \psi^{27}_{2,(2,3)}. \tag{5.2.4}$$

The application of the operator $E_{-\alpha_2}$ on the right hand side of (5.2.3) gives

$$E_{-\alpha_2}(\psi^7_{1,(1,2)} \psi^{7'}_{1,(1,2)}) = (E^7_{-\alpha_2} \psi^7_{1,(1,2)}) \psi^7_{1(1,2)}$$

$$+ \psi^7_{1,(1,2)} (E^7_{-\alpha_2} \psi^{7'}_{1,(1,2)}). \tag{5.2.5}$$

Again from Table 4.17, (5.2.5) becomes

$$E_{-\alpha_2}(\psi^{7}_{1,(1,2)}\psi^{7'}_{1,(1,2)}) = \psi^{7}_{2,(1,1)}\psi^{7'}_{1,(1,2)} + \psi^{7}_{1,(1,2)}\psi^{7'}_{2,(1,1)}. \tag{5.2.6}$$

Equating the right hand sides of (5.2.4) and (5.2.6), we get

$$\psi^{27}_{2,(2,3)} = \frac{1}{\sqrt{2}}(\psi^{7}_{2,(1,1)}\psi^{7'}_{1,(1,2)}+\psi^{7}_{1,(1,2)}\psi^{7'}_{2,(1,1)}). \tag{5.2.7}$$

Following the above procedure, the successive application of the lowering and raising operators on the states of the $\underset{\sim}{27}$ and $\underset{\sim}{7},\underset{\sim}{7}'$ representations, according to Table 4.17, will result in a complete determination of the Clebsch-Gordan coefficients $\begin{pmatrix}7 & 7' & | & 27\\ i,\nu & j,\mu & | & k,\pi\end{pmatrix}$.

Applying this method, we must be careful when we encounter a state with internal multiplicity greater than one. In Figure 4.3 we have isolated the first state with internal multiplicity two of the $\underset{\sim}{27}$ dimension representation.

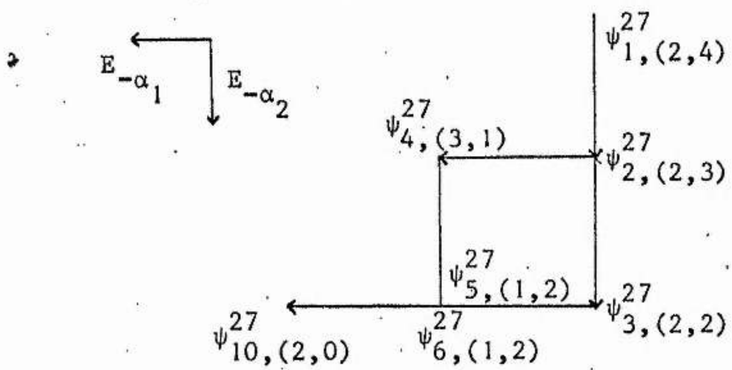


Figure 4.3

The application of $E_{-\alpha_1}$ to the state $\psi^{27}_{3,(2,2)}$ gives

$$E_{-\alpha_1}\psi^{27}_{3,(2,2)} = \sqrt{\frac{2}{3}}\,\psi^{27}_{5,(1,2)} + \frac{2}{\sqrt{3}}\,\psi^{27}_{6,(1,2)}. \tag{5.2.8}$$

The result of applying $E_{-\alpha_2}$ on $\psi^{27}_{2,(2,3)}$ is

$$\psi^{27}_{3,(2,2)} = \psi^{7}_{2,(1,1)}\psi^{7'}_{2,(1,1)}. \tag{5.2.9}$$

Then, the left hand side of (5.2.8) becomes

$$E_{-\alpha_1}\psi^{7}_{2,(1,1)}\psi^{7'}_{2,(1,1)} = \psi^{7}_{3,(0,1)}\psi^{7'}_{2,(1,1)} + \psi^{7}_{2,(1,1)}\psi^{7}_{3,(0,1)}. \tag{5.2.10}$$

Because we have chosen the state 5 to belong to an E_{α_2}-triplet, when $E_{-\alpha_2}$ is applied on the basis function $\psi^{27}_{4,(3,1)}$ we find

$$\psi^{27}_{5,(1,2)} = \frac{1}{\sqrt{3}}\psi^{7}_{4,(0,0)}\psi^{7'}_{1,(1,2)} + \frac{1}{\sqrt{6}}\psi^{7}_{3,(0,1)}\psi^{7'}_{2,(1,1)} + \frac{1}{\sqrt{6}}\psi^{7}_{2,(1,1)}\psi^{7'}_{3,(0,1)} + \frac{1}{\sqrt{3}}\psi^{7}_{1,(1,2)}\psi^{7'}_{4,(0,0)}. \tag{5.2.11}$$

Now, if we substitute the states $\psi^{27}_{3,(2,2)}$ and $\psi^{27}_{5,(1,2)}$ and relation (5.2.10) to (5.2.8), we get

$$\begin{aligned}\psi^{7}_{3,(0,1)}\psi^{7'}_{2,(1,1)} &+ \psi^{7}_{2,(1,1)}\psi^{7}_{3,(0,1)} \\ &= \sqrt{\tfrac{2}{3}}\Big(\frac{1}{\sqrt{3}}\psi^{7}_{4,(0,0)}\psi^{7'}_{1,(1,2)} + \frac{1}{\sqrt{6}}\psi^{7}_{3,(0,1)}\psi^{7}_{2,(1,1)} \\ &\quad + \frac{1}{\sqrt{6}}\psi^{7}_{2,(1,1)}\psi^{7'}_{3,(0,1)} + \frac{1}{\sqrt{3}}\psi^{7}_{1,(1,2)}\psi^{7'}_{4,(0,0)}\Big) \\ &\quad + \frac{2}{\sqrt{3}}\psi^{27}_{6,(1,2)},\end{aligned} \tag{5.2.12}$$

from which the state $\psi^{27}_{6,(1,2)}$ is determined in terms of its $\underset{\sim}{7}$ and $\underset{\sim}{7}'$ components. The result is

$$\psi^{27}_{6,(1,2)} = \frac{1}{\sqrt{3}}\psi^{7}_{3,(0,1)}\psi^{7'}_{2,(1,1)} + \frac{1}{\sqrt{3}}\psi^{7}_{2,(1,1)}\psi^{7'}_{3,(0,1)} - \frac{1}{\sqrt{6}}\psi^{7}_{4,(0,0)}\psi^{7'}_{1,(1,2)} - \frac{1}{\sqrt{6}}\psi^{7}_{1,(1,2)}\psi^{7'}_{4,(0,0)}. \tag{5.2.13}$$

To evaluate the Clebsch-Gordan coefficients $\left(\begin{smallmatrix}7 & 7' \\ i,\nu & j,\mu\end{smallmatrix}\middle|\begin{smallmatrix}14 \\ \ell,\lambda\end{smallmatrix}\right)$, we start again with the highest weight of the $\underset{\sim}{14}$ representation (2,3). However, there is another state with the same weight, belonging to the $\underset{\sim}{27}$ representation, given by

$$\psi^{27}_{2,(2,3)} = \frac{1}{\sqrt{2}}\left(\psi^{7}_{2,(1,1)}\psi^{7'}_{1,(1,2)} + \psi^{7}_{1,(1,2)}\psi^{7'}_{2,(1,1)}\right).$$

To define the state $\psi^{14}_{1,(2,3)}$ we choose the orthogonal combination to above state

$$\psi^{14}_{1,(2,3)} = \frac{1}{\sqrt{2}}(\psi^{7}_{2,(1,1)}\psi^{7'}_{1,(1,2)}-\psi^{7}_{1,(1,2)}\psi^{7'}_{2,(1,1)}), \qquad (5.2.14)$$

and we repeat exactly the same procedure. The $\left(\begin{smallmatrix}7 & 7' \\ i,\nu & j,\mu\end{smallmatrix}\middle|\begin{smallmatrix}7\\ m,\omega\end{smallmatrix}\right)$ coefficients will be the result of the same manipulations, starting with the state

$$\psi^{7}_{1,(1,2)} = \frac{1}{\sqrt{6}}\psi^{7}_{4,(0,0)}\psi^{7'}_{1,(1,2)} - \frac{1}{\sqrt{3}}\psi^{7}_{3,(0,1)}\psi^{7'}_{2,(1,1)}$$

$$+ \frac{1}{\sqrt{3}}\psi^{7}_{2,(1,1)}\psi^{7'}_{3,(0,1)} - \frac{1}{\sqrt{6}}\psi^{7}_{1,(1,2)}\psi^{7'}_{4,(0,0)}, \qquad (5.2.15)$$

which is orthogonal to the states $\psi^{27}_{5,(1,2)},\psi^{27}_{6,(1,2)},\psi^{14}_{3,(1,2)}$. Finally the state $\psi^{1}_{1,(0,0)}$ is given by

$$\psi^{1}_{1,(0,0)} = \frac{1}{2}(-\psi^{7}_{6,(-1,-1)}\psi^{7'}_{2,(1,1)}+\psi^{7}_{2,(1,1)}\psi^{7'}_{6,(-1,-1)}$$

$$-\psi^{7}_{1,(1,2)}\psi^{7'}_{7,(-1,-2)}+\psi^{7}_{7,(-1,-2)}\psi^{7'}_{1,(1,2)}), \qquad (5.2.16)$$

which is orthogonal to the states $\psi^{27}_{13,(0,0)},\psi^{27}_{14,(0,0)},\psi^{27}_{15,(0,0)}$, $\psi^{14}_{7,(0,0)}$ and $\psi^{14}_{8,(0,0)}$.

The orthogonality properties of the above Clebsch-Gordan coefficients (relations (5.1.4)-(5.1.6)) are automatically satisfied.

We shall give the complete set of the Clebsch-Gordan coefficients of the $\underset{\sim}{7} \otimes \underset{\sim}{7}$ tensor product in §4.5.3. The above method is of a general nature and can be applied to any classical or exceptional simple complex Lie algebra.

In the case of multiplicity greater than one, the above method is also applicable. The only difference is that the basis function corresponding to the highest weight of the representation $\underset{\sim}{\Gamma}^{a,\gamma_i}$ must be an orthogonal combination of the bases functions corresponding to the highest weights of the representations $\underset{\sim}{\Gamma}^{a,\gamma_j}$ with $j < i$.

§4.5.3 Computer implementation

We have developed a pilot program (Program E1(5)) to deal with the evaluation of the Clebsch-Gordan coefficients of the tensor product $\underset{\sim}{7} \otimes \underset{\sim}{7}$ of G_2. As for the matrix representation program, we found that Algol-W is the best programming language to be used for such a job.

The main structure of Program EI(5) consists of a loop in which the control variable I takes the values from one to four. Each value of the control variable corresponds to a representation which enters the Clebsch-Gordan series. When the value of one is assigned to I, then two three-dimensional arrays E and MAT are declared. The first array E is filled in with the matrix elements of the $\underset{\sim}{27}$ representation after the call of the procedure INPUT. As the program is executed the array MAT stores the calculated Clebsch-Gordan coefficients.

The evaluation of the Clebsch-Gordan coefficients starts with the application of the lowering operators to the first state of the 27-dimensional (which corresponds to the highest weight of the $\underset{\sim}{27}$ representation). The Clebsch-Gordan coefficients of this state are read from the input cards. The internal multiplicity of the representation is controlled by the variable COUNT. If the variable COUNT has the value of one, then the lowering operators are again applied, and the call of the procedure FINDING-MULTIPLET calculates the Clebsch-Gordan coefficients. If COUNT takes a value greater than one, the procedure LOOP is called and performs similar calculations to the ones in §4.5.2. Again, the call of the procedure FINDING-MULTIPLET fixes the Clebsch-Gordan coefficients.

Before the control variable I takes the value of two, the Clebsch-Gordan coefficients of the $\underset{\sim}{27}$ representation are printed out. This allows us to choose the state of the 14-dimensional representation corresponding to the highest weight of this representation, and consequently the Clebsch-Gordan coefficients of this state. This procedure is continued until all the representations of the Clebsh-Gordan series are exhausted.

The input of the program consists of: the rank of the algebra; the number of representations entering the Clebsch-Gordan series; the maximum internal multiplicity of the representations of the Clebsch-Gordan series; the dimension of one of the representations of the tensor product; the dimensions of the representations of the Clebsch-Gordan series; the Clebsch-Gordan coefficients of the state corresponding to the highest weights of the representations, and finally the matrix elements of the representations of the Clebsch-Gordan series.

In Table 4.16 the Clebsch-Gordan coefficients of the above Clebsch-Gordan series are listed. In that table the first two columns represent the states of the $\underset{\sim}{7}$ and $\underset{\sim}{7}'$ representations respectively, while in the third column the Clebsch-Gordan coefficients are stated.

```
*******************************

THE STATE NUM I=1
-----------------
  1    1   1.00

THE STATE NUM I=2
-----------------
  1    2   0.71
  2    1   0.71

THE STATE NUM I=3
-----------------
  2    2   1.00

THE STATE NUM I=4
-----------------
  1    3   0.71
  3    1   0.71

THE STATE NUM I=5
-----------------
  1    4   0.58
  2    3   0.41
  3    2   0.41
  4    1   0.58

THE STATE NUM I=6
-----------------
  1    4  -0.41
  2    3   0.58
  3    2   0.58
  4    1  -0.41

THE STATE NUM I=7
-----------------
  1    5   0.41
  2    4   0.58
  4    2   0.58
  5    1   0.41

THE STATE NUM I=8
-----------------
  1    5  -0.58
  2    4   0.41
  4    2   0.41
  5    1  -0.58

THE STATE NUM I=9
-----------------
  2    5   0.71
  5    2   0.71

THE STATE NUM I=10
-----------------
  3    3   1.00

THE STATE NUM I=11
-----------------
  3    4   0.71
  4    3   0.71

THE STATE NUM I=12
-----------------
  1    6   0.71
  6    1   0.71

THE STATE NUM I=13
-----------------
  3    5   0.41
  4    4   0.82
  5    3   0.41

THE STATE NUM I=14
-----------------
  1    7   0.50
  2    6   0.50
  6    2   0.50
  7    1   0.50

THE STATE NUM I=15
-----------------
  2    6   0.33
  3    5   0.44
  4    4  -0.44
  5    3   0.44

  6    2   0.33
  7    1  -0.33

THE STATE NUM I=16
-----------------
  4    5   0.71
  5    4   0.71

THE STATE NUM I=17
-----------------
  2    7   0.71
  7    2  -0.71

THE STATE NUM I=18
-----------------
  5    5   1.00

THE STATE NUM I=19
-----------------
  3    6   0.71
  6    3   0.71

THE STATE NUM I=20
-----------------
  3    7   0.41
  4    6   0.58
  6    4   0.58
  7    3   0.41

THE STATE NUM I=21
-----------------
  3    7  -0.58
  4    6   0.41
  6    4   0.41
  7    3  -0.58

THE STATE NUM I=22
-----------------
  5    6   0.41
  6    5   0.41
  7    4   0.58

THE STATE NUM I=23
-----------------
  4    7  -0.41
  5    6   0.58
  6    5   0.58
  7    4  -0.41

THE STATE NUM I=24
-----------------
  5    7   0.71
  7    5   0.71

THE STATE NUM I=25
-----------------
  6    6   1.00

THE STATE NUM I=26
-----------------
  6    7   0.71
  7    6   0.71

THE STATE NUM I=27
-----------------
  7    7   1.00
```

THE C.G.COEF OF THE REPR=14 ARE
****

THE STATE NUM I=1

1	2	-0.71
2	1	0.71

THE STATE NUM I=2

1	3	-0.71
3	1	0.71

THE STATE NUM I=3

1	4	-0.58
2	3	-0.41
3	2	0.41
4	1	0.58

THE STATE NUM I=4

1	5	-0.41
2	4	-0.58
4	2	0.58
5	1	0.41

THE STATE NUM I=5

2	5	-0.71
5	2	0.71

THE STATE NUM I=6

1	6	-0.41
3	4	-0.58
4	3	0.58
6	1	0.41

THE STATE NUM I=7

1	7	-0.29
2	6	-0.29
3	5	-0.58
5	3	0.58
6	2	0.29
7	1	0.29

THE STATE NUM I=8

1	7	0.50
2	6	-0.50
6	2	0.50
7	1	-0.50

THE STATE NUM I=9

2	7	-0.41
4	5	-0.58
5	4	0.58
7	2	0.41

THE STATE NUM I=10

3	6	-0.71
6	3	0.71

THE STATE NUM I=11

3	7	-0.41
4	6	-0.58
6	4	0.58
7	3	0.41

THE STATE NUM I=12

4	7	-0.58
5	6	-0.41
6	5	0.41
7	4	0.58

THE STATE NUM I=13

5	7	-0.71
7	5	0.71

THE STATE NUM I=14

6	7	-0.71
7	6	0.71

THE C.G.COEF OF THE REPR=7 ARE
****

THE STATE NUM I=1

1	4	-0.40
2	3	0.58
3	2	-0.58
4	1	0.40

THE STATE NUM I=2

1	5	-0.57
2	4	0.41
4	2	-0.41
5	1	0.57

THE STATE NUM I=3

1	6	-0.57
3	4	0.41
4	3	-0.41
6	1	0.57

THE STATE NUM I=4

1	7	-0.40
2	6	-0.40
3	5	0.41
5	3	-0.41
6	2	0.40
7	1	-0.40

THE STATE NUM I=5

2	7	-0.57
4	5	0.41
5	4	-0.41
7	2	0.57

THE STATE NUM I=6

3	7	-0.57
4	6	0.41
6	4	-0.41
7	3	0.57

THE STATE NUM I=7

4	7	-0.40
5	6	0.58
6	5	-0.58
7	4	0.40

THE C.G.COEF OF THE REPR=1 ARE
****

THE STATE NUM I=1

1	7	-0.50
2	6	0.50
6	2	-0.50
7	1	0.50

Table 4.16

CHAPTER 5

SO(10) MODEL: MATRIX REALIZATION AND CLEBSCH-GORDAN COEFFICIENTS

The mathematical structure of the successful SO(10) model will be explored in this chapter. The study based on the results of Chapter 4 will be directed to the construction of an explicit matrix realization of the irreducible representations $\underset{\sim}{126}$, $\underset{\sim}{120}$, $\underset{\sim}{16}$ and $\underset{\sim}{10}$, and the derivation of the Clebsch-Gordan coefficients of the Clebsch-Gordan series $\underset{\sim}{16} \otimes \underset{\sim}{16} = \underset{\sim}{126} \oplus \underset{\sim}{120} \oplus \underset{\sim}{10}$.

The SO(10) theory has been studied in detail for its physical implications (Chapter 3). However, little has been said about its Lie structure. In reference [29], the structure of the SO(10) model is investigated in connection with other models obeying the colour restriction [29]. In a series of papers [65], the SO(10) model is studied using the Clifford algebra [66]. However, nothing has been done on specifying unambiguously the states of the SO(10) theory, and on deriving a complete set of Clebsch-Gordan coefficients.

Our analysis of the SO(10) theory is of a general nature, and it can be applied to other SO(2n) models with n = 7,9,..., and to models based on the exceptional Lie algebras. We shall investigate these models in the next chapter.

The chapter is organized as follows. In §5.1 we construct the matrix realization of the SO(10) model. The Clebsch-Gordan coefficients are given in §5.2.

§5.1 Matrix Realization of the SO(10) Model

The algebra D_5 has rank 5 and the number of positive roots is 20. The adjoint representation D(0,1,0,0,0) has dimension D = 45. The algebra is generated by the following 45 generators:

H_{α_i}, $i = 1,2,\ldots,5$ (diagonal generators)

$E_{\pm\alpha}, E_{\pm\beta}, \ldots, E_{\pm\omega}$, $(\alpha,\beta,\ldots,\omega) \in \Delta+$ (non-diagonal generators)

The algebra has two inequivalent spinorial representations $D(0,0,0,0,1)$ and $D(0,0,0,1,0)$ of dimension $d = 16$. Their Kronecker product is

$$D^{16}(0,0,0,0,1) \otimes D^{16}(0,0,0,0,1)$$
$$= D^{10}(1,0,0,0,0) \oplus D^{120}(0,0,1,0,0) \oplus D^{126}(0,0,0,0,2)$$

and

$$D^{16}(0,0,0,0,1) \otimes D^{16'}(0,0,0,1,0)$$
$$= D^{1}(0,0,0,0,0) \oplus D^{45}(0,1,0,0,0) \oplus D^{210}(0,0,0,1,1).$$

The weights of the $\underset{\sim}{16}$, $\underset{\sim}{16}'$ and $\underset{\sim}{10}$ representations are all simple, while the non-simple weights with their multiplicity of the $\underset{\sim}{120}$ and $\underset{\sim}{126}$ representations are

$$\left.\begin{matrix} 1 & 1 & 1 & 0.5 & 0.5 \\ 0 & 1 & 1 & 0.5 & 0.5 \\ 0 & 0 & 1 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0.5 & -0.5 \end{matrix}\right\} \text{mult} = 4, \quad \left.\begin{matrix} 1 & 1 & 1 & 0.5 & 0.5 \\ 0 & 1 & 1 & 0.5 & 0.5 \\ 0 & 0 & 1 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0.5 & -0.5 \end{matrix}\right\} \text{mult} = 3.$$

As before, we represent each weight $\lambda = \sum_{j=1}^{\ell} \mu_j \alpha_j$ with the coefficients μ_j, $j = 1,2,\ldots,5$.

We observe that the non-simple weights of the $\underset{\sim}{120}$ and $\underset{\sim}{126}$ representations come with a specific pattern and only their multiplicities differ. In this particular case, the above listed weights are the weights of the 10-dimensional representation. This observation plays an important role in finding the weights of higher rank algebras D_ℓ.

In this section the matrix realization of the diagonal generators H_{α_i}, $i = 1,2,\ldots,\ell$ (§5.1.1), and the matrix realization of the $E_{\pm\alpha}$, $\alpha \in \Delta+$ (§5.1.2) are constructed.

§5.1.1 Diagonal generators

The diagonal generators are easily constructed from the equation (4.2.2) (Chapter 4). Because the weighting factors for D_5 are all equal to 1, the relation $H_{\alpha_i} = \{2/\langle\alpha_i,\alpha_i\rangle\}h_{\alpha_i}$ takes the form (using (4.2.8) of Chapter 4 and the normalization of $\langle\alpha_i,\alpha_i\rangle$ of Appendix B)

$$H_{\alpha_i} = 2(\ell-1)h_{\alpha_i},\ i = 1,2,\ldots,5. \tag{1.1.1}$$

Using Program D2(4) we have tabulated the diagonal generators (1.1.1) (Tables 5.1, 5.2, 5.3 and 5.4) of the representations $\underset{\sim}{126}$, $\underset{\sim}{120}$, $\underset{\sim}{16}$ and $\underset{\sim}{10}$. In the above tables the factor $2(\ell-1)$ of (1.1.1) is understood.

§5.1.2 Non-diagonal generators

For the 10-dimensional representation, because all the weights are simple, the generators E_{α_i}, $i = 1,2,\ldots,\ell$, can easily be constructed from the A_1-subalgebra content of this representation.

The $\underset{\sim}{126}$ representation has a more complicated weight diagram which is given in Figure 5.1. In this figure the weights are listed according to their lexicographical order. The various coloured lines join pairs of eigenvectors ψ_λ and $\psi_{\lambda'}$ such that $(\psi_\lambda, E_{-\alpha_i}\psi_{\lambda'}) \neq 0$ for $i = 1,2,\ldots,5$. The calculation of the matrix elements $\Gamma(\underset{\sim}{E}_{\alpha_i})$, $i = 1,2,\ldots,5$, is tedious, because of the high rank of the algebra D_5 and the difficulty of representing a five dimensional space weight diagram. Here we shall outline the calculation procedure which has been implemented as a computer program (Program D5(4)), and give the full calculations of the first conjunction of weights with multiplicity three.

126 -DIMENSIONAL REPRESENTATION

```
THE DIAGONAL GENERATOR HA1 IS
HA1 =DIAG(
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5
 0.5  0.5  0.5  0.5  0.5  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0 -0.5 -0.5 -0.5 -0.5 -0.5
-0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.5  0.5  0.5  0.5
 0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0
-1.0 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0 )
------------------------------------------
THE DIAGONAL GENERATOR HA2 IS
HA2 =DIAG(
 0.0  0.0  0.0  0.5  0.5  0.5  0.5  1.0  1.0  1.0 -0.5 -0.5 -0.5 -0.5  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.5  0.5  0.5  0.5 -1.0 -1.0 -1.0 -0.5 -0.5 -0.5 -0.5  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.5
 0.5  0.5  0.5  0.5  0.5  0.5  1.0  1.0  1.0  1.0 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.5  0.5  0.5  0.5  0.5  0.5  0.5 -1.0 -1.0 -1.0 -1.0
-0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.5  0.5  0.5  0.5  1.0  1.0
 1.0 -0.5 -0.5 -0.5 -0.5  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.5  0.5  0.5  0.5 -1.0 -1.0 -1.0 -0.5
-0.5 -0.5 -0.5  0.0  0.0  0.0 )
------------------------------------------
THE DIAGONAL GENERATOR HA3 IS
HA3 =DIAG(
 0.0  0.5  1.0 -0.5  0.0  0.0  0.5 -1.0 -0.5  0.0  0.0  0.5  0.5  1.0 -0.5 -0.5  0.0  0.0  0.0  0.5
 0.5 -1.0 -0.5 -0.5  0.0  0.0  0.5  1.0 -0.5  0.0  0.0  0.5 -1.0 -0.5  0.0  0.0  0.5  0.5  1.0 -0.5
-0.5  0.0  0.0  0.0  0.5  0.5 -1.0 -0.5 -0.5  0.0  0.0  0.0  0.5  0.5  0.5  1.0  1.0 -0.5 -0.5 -0.5
 0.0  0.0  0.0  0.0  0.0  0.0  0.5  0.5  0.5 -1.0 -1.0 -0.5 -0.5 -0.5  0.0  0.0  0.0  0.5  0.5  1.0
-0.5 -0.5  0.0  0.0  0.0  0.5  0.5 -1.0 -0.5 -0.5  0.0  0.0  0.5  1.0 -0.5  0.0  0.0  0.5 -1.0 -0.5
 0.0  0.0  0.5  0.5  1.0 -0.5 -0.5  0.0  0.0  0.0  0.5  0.5 -1.0 -0.5 -0.5  0.0  0.0  0.5  1.0 -0.5
 0.0  0.0  0.5 -1.0 -0.5  0.0 )
```

```
THE DIAGONAL GENERATOR HA4 IS
HA4 =DIAG(
 0.0  0.0  0.0  0.5  0.5 -0.5 -0.5  1.0  0.0 -1.0  0.5  0.5 -0.5 -0.5  1.0  0.0  0.0  0.0  0.0  0.0
-1.0  0.5  0.5 -0.5 -0.5  1.0  0.0 -1.0  0.5  0.5 -0.5 -0.5  0.0  0.0  0.0  0.5  0.5 -0.5 -0.5  1.0
 0.0  0.0  0.0  0.0  0.0 -1.0  0.5  0.5 -0.5 -0.5  1.0  0.0  0.0  0.0  0.0  0.0 -1.0  0.5  0.5  0.5
 0.5  0.5  0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5  1.0  0.0  0.0  0.0  0.0  0.0 -1.0  0.5  0.5 -0.5 -0.5
 1.0  0.0  0.0  0.0  0.0  0.0 -1.0  0.5  0.5 -0.5 -0.5  1.0  0.0 -1.0  0.5  0.5 -0.5 -0.5  0.0  0.0
 0.0  0.5  0.5 -0.5 -0.5 -1.0  0.0  0.0  0.0  0.0  0.0 -1.0  0.5  0.5 -0.5 -0.5  0.0  0.0  0.0  0.5
 0.5 -0.5 -0.5  1.0  0.0 -1.0 )
------------------------------------
THE DIAGONAL GENERATOR HA5 IS
HA5 =DIAG(
 1.0  0.0 -1.0  0.5 -0.5  0.5 -0.5  0.0  0.0  0.0  0.5 -0.5  0.5 -0.5  0.0  1.0  0.0  0.0  0.0 -1.0
 0.0  0.5 -0.5  0.5 -0.5  0.0  0.0  0.0  0.5 -0.5  0.5 -0.5  1.0  0.0 -1.0  0.5 -0.5  0.5 -0.5  0.0
 1.0  0.0  0.0  0.0 -1.0  0.0  0.5 -0.5  0.5 -0.5  0.0  1.0  0.0  0.0  0.0 -1.0  0.0  0.5  0.5  0.5
-0.5 -0.5 -0.5  0.5  0.5  0.5 -0.5 -0.5 -0.5  0.0  1.0  0.0  0.0  0.0 -1.0  0.0  0.5 -0.5  0.5 -0.5
 0.0  1.0  0.0  0.0  0.0 -1.0  0.0  0.5 -0.5  0.5 -0.5  0.0  0.0  0.0  0.5 -0.5  0.5 -0.5  1.0  0.0
-1.0  0.5 -0.5  0.5 -0.5  0.0  1.0  0.0  0.0  0.0 -1.0  0.0  0.5 -0.5  0.5 -0.5  1.0  0.0 -1.0  0.5
-0.5  0.5 -0.5  0.0  0.0  0.0 )
```

Table 5.1

120 -DIMENSIONAL REPRESENTATION

```
THE DIAGONAL GENERATOR HA1 IS
HA1 =DIAG(
 0.0  0.0  0.0  0.0  0.0  0.0  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5
 0.5  0.5  1.0  1.0  1.0  1.0  1.0  1.0 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5
-0.5 -0.5 -0.5 -0.5  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.5  0.5  0.5  0.5
 0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -0.5 -0.5
-0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5  0.0  0.0  0.0  0.0  0.0  0.0 )
--------------------------------
THE DIAGONAL GENERATOR HA2 IS
HA2 =DIAG(
 0.0  0.5  0.5  0.5  0.5  1.0 -0.5 -0.5 -0.5 -0.5  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.5  0.5
 0.5  0.5 -1.0 -0.5 -0.5 -0.5 -0.5  0.0  0.0  0.0  0.0  0.0  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5
 1.0  1.0  1.0  1.0 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5  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  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5 -1.0 -1.0 -1.0 -1.0
-0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5  0.0  0.0  0.0  0.0  0.0  0.5  0.5  0.5  0.5  1.0 -0.5 -0.5
-0.5 -0.5  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.5  0.5  0.5  0.5 -1.0 -0.5 -0.5 -0.5 -0.5  0.0 )
--------------------------------
THE DIAGONAL GENERATOR HA3 IS
HA3 =DIAG(
 0.5 -0.5  0.0  0.0  0.5 -0.5  0.0  0.5  0.5  1.0 -0.5 -0.5  0.0  0.0  0.0  0.0  0.5  0.5 -1.0 -0.5
-0.5  0.0  0.5 -0.5  0.0  0.0  0.5 -0.5  0.0  0.5  0.5  1.0 -0.5 -0.5  0.0  0.0  0.0  0.0  0.5  0.5
-1.0 -0.5 -0.5  0.0  0.0  0.0  0.5  0.5  0.5  0.5  1.0  1.0 -0.5 -0.5 -0.5 -0.5  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.5  0.5  0.5  0.5 -1.0 -1.0 -0.5 -0.5 -0.5 -0.5  0.0  0.0  0.0  0.5  0.5  1.0
-0.5 -0.5  0.0  0.0  0.0  0.0  0.5  0.5 -1.0 -0.5 -0.5  0.0  0.5 -0.5  0.0  0.0  0.5 -0.5  0.0  0.5
-0.5  1.0 -0.5 -0.5  0.0  0.0  0.0  0.0  0.5  0.5 -1.0 -0.5 -0.5  0.0  0.5 -0.5  0.0  0.0  0.5 -0.5 )
--------------------------------
THE DIAGONAL GENERATOR HA4 IS
HA4 =DIAG(
 0.0  0.5  0.5 -0.5 -0.5  0.0  0.5  0.5 -0.5 -0.5  1.0  0.0  0.0  0.0  0.0  0.0  0.0 -1.0  0.5  0.5
-0.5 -0.5  0.0  0.5  0.5 -0.5 -0.5  0.0  0.5  0.5 -0.5 -0.5  1.0  0.0  0.0  0.0  0.0  0.0  0.0 -1.0
 0.5  0.5 -0.5 -0.5  0.0  0.0  0.0  0.0  0.0  0.0  0.0 -1.0  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5
-0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5 -0.5  1.0  0.0  0.0  0.0  0.0  0.0  0.0 -1.0  0.5  0.5 -0.5 -0.5
 1.0  0.0  0.0  0.0  0.0  0.0  0.0 -1.0  0.5  0.5 -0.5 -0.5  0.0  0.5  0.5 -0.5 -0.5  0.0  0.5  0.5
-0.5 -0.5  1.0  0.0  0.0  0.0  0.0  0.0  0.0 -1.0  0.5  0.5 -0.5 -0.5  0.0  0.5  0.5 -0.5 -0.5  0.0 )
--------------------------------
```

```
THE DIAGONAL GENERATOR HAS IS
HAS =DIAG(
 0.0  0.5 -0.5  0.5 -0.5  0.0  0.5 -0.5  0.5 -0.5  0.0  1.0  0.0  0.0  0.0  0.0 -1.0  0.0  0.5 -0.5
 0.5 -0.5  0.0  0.5 -0.5  0.5 -0.5  0.0  0.5 -0.5  0.5 -0.5  0.0  1.0  0.0  0.0  0.0  0.0 -1.0  0.0
 0.5 -0.5  0.5 -0.5  0.0  1.0  0.0  0.0  0.0  0.0 -1.0  0.0  0.5  0.5  0.5  0.5 -0.5 -0.5 -0.5 -0.5
 0.5  0.5  0.5  0.5 -0.5 -0.5 -0.5 -0.5  0.0  1.0  0.0  0.0  0.0  0.0 -1.0  0.0  0.5 -0.5  0.5 -0.5
 0.0  1.0  0.0  0.0  0.0  0.0 -1.0  0.0  0.5 -0.5  0.5 -0.5  0.0  0.5 -0.5  0.5 -0.5  0.0  0.5 -0.5
 0.5 -0.5  0.0  1.0  0.0  0.0  0.0  0.0 -1.0  0.0  0.5 -0.5  0.5 -0.5  0.0  0.5 -0.5  0.5 -0.5  0.0 )
```

Table 5.2

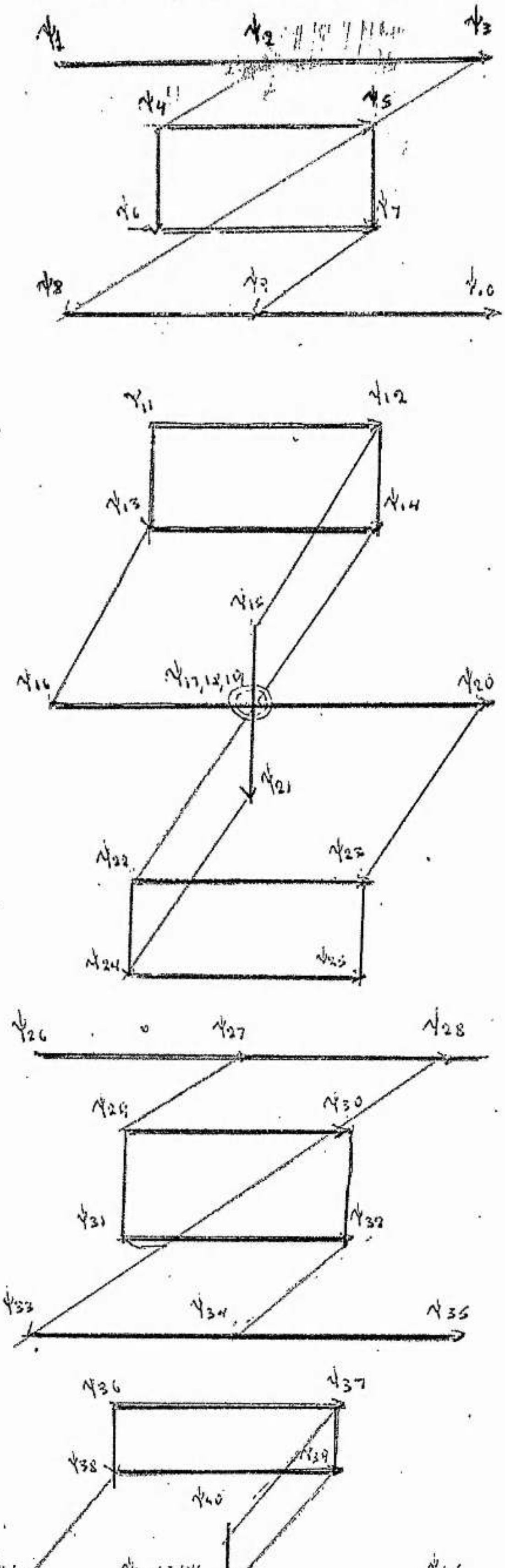
```
16 -DIMENSIONAL REPRESENTATION
******************************

THE DIAGONAL GENERATOR HA1 IS
HA1 =DIAG(
 0.0  0.0  0.0  0.0  0.5  0.5  0.5  0.5 -0.5 -0.5 -0.5 -0.5  0.0  0.0  0.0  0.0 )
--------------------------------------
THE DIAGONAL GENERATOR HA2 IS
HA2 =DIAG(
 0.0  0.0  0.5  0.5 -0.5 -0.5  0.0  0.0  0.0  0.0  0.5  0.5 -0.5 -0.5  0.0  0.0 )
--------------------------------------
THE DIAGONAL GENERATOR HA3 IS
HA3 =DIAG(
 0.0  0.5 -0.5  0.0  0.0  0.5 -0.5  0.0  0.0  0.5 -0.5  0.0  0.0  0.5 -0.5  0.0 )
--------------------------------------
THE DIAGONAL GENERATOR HA4 IS
HA4 =DIAG(
 0.0  0.0  0.5 -0.5  0.5 -0.5  0.0  0.0  0.5 -0.5  0.0  0.0  0.0  0.0  0.5 -0.5 )
--------------------------------------
THE DIAGONAL GENERATOR HA5 IS
HA5 =DIAG(
 0.5 -0.5  0.0  0.0  0.0  0.0  0.5 -0.5  0.0  0.0  0.5 -0.5  0.5 -0.5  0.0  0.0 )
```

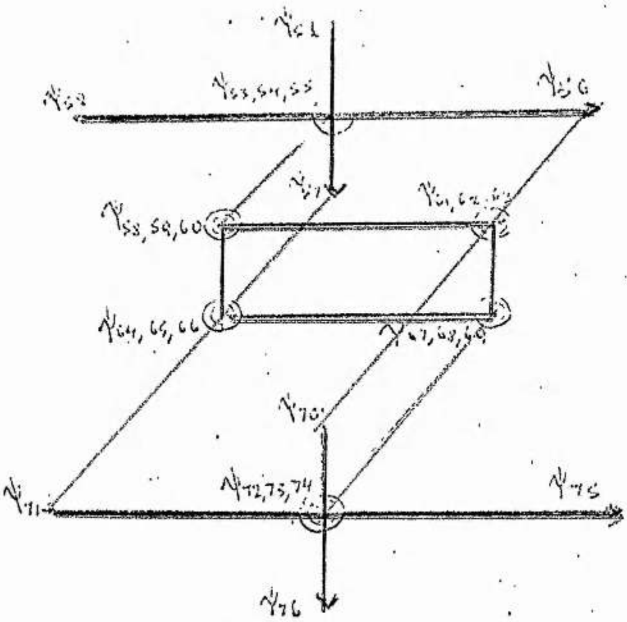
Table 5.3

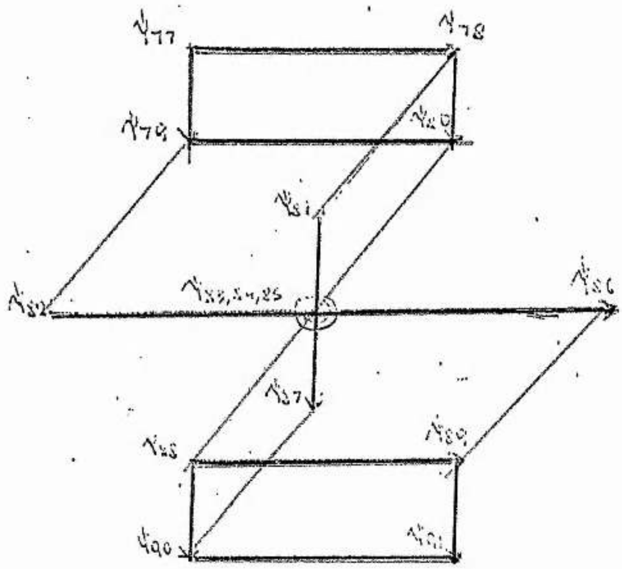
```
10 -DIMENSIONAL REPRESENTATION
******************************

THE DIAGONAL GENERATOR HA1 IS
HA1 =DIAG(
 0.5 -0.5  0.0  0.0  0.0  0.0  0.0  0.0  0.5 -0.5 )
------------------------------------------------
THE DIAGONAL GENERATOR HA2 IS
HA2 =DIAG(
 0.0  0.5 -0.5  0.0  0.0  0.0  0.0  0.5 -0.5  0.0 )
------------------------------------------------
THE DIAGONAL GENERATOR HA3 IS
HA3 =DIAG(
 0.0  0.0  0.5 -0.5  0.0  0.0  0.5 -0.5  0.0  0.0 )
------------------------------------------------
THE DIAGONAL GENERATOR HA4 IS
HA4 =DIAG(
 0.0  0.0  0.0  0.5  0.5 -0.5 -0.5  0.0  0.0  0.0 )
------------------------------------------------
THE DIAGONAL GENERATOR HA5 IS
HA5 =DIAG(
 0.0  0.0  0.0  0.5 -0.5  0.5 -0.5  0.0  0.0  0.0 )
```

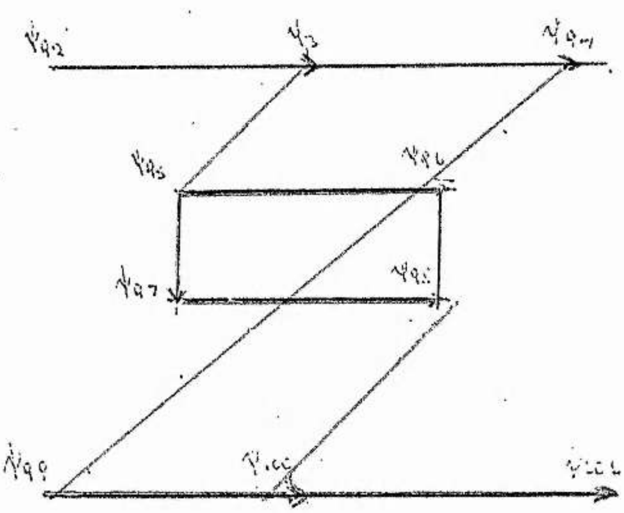
Table 5.4

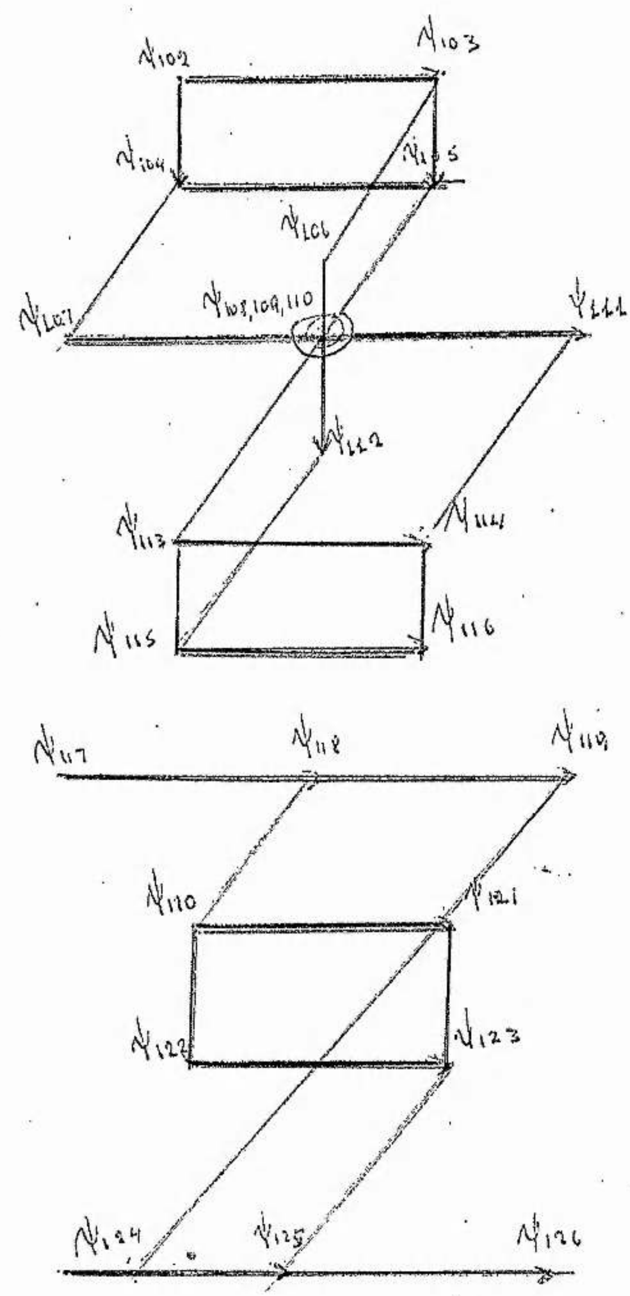
$Ed\alpha_5$, $Ed\alpha_4$, $Ed\alpha_3$







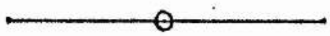


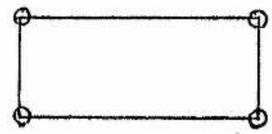


ψ_{102}
ψ_{103}
ψ_{104}
ψ_{105}
ψ_{106}
ψ_{107}
$\psi_{108,109,110}$
ψ_{111}
ψ_{112}
ψ_{113}
ψ_{114}
ψ_{115}
ψ_{116}
ψ_{117}
ψ_{118}
ψ_{119}
ψ_{110}
ψ_{121}
ψ_{122}
ψ_{123}
ψ_{124}
ψ_{125}
ψ_{126}

Figure 5.1

In the weight diagram there are two different arrangements of non-simple weights.





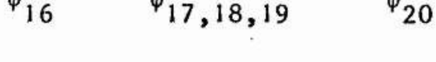
The first arrangement appears six times, while the second appears once.

We start by an arbitrary choice of the α_5-multiplets (red lines). Our choice is

$(\psi_{16},\psi_{17},\psi_{20})$-triplet; $(\psi_{41},\psi_{42},\psi_{45})$-triplet; $(\psi_{52},\psi_{53},\psi_{56})$-triplet; (ψ_{58},ψ_{61})-doublet; (ψ_{59},ψ_{62})-doublet; (ψ_{60},ψ_{63})-doublet; (ψ_{64},ψ_{67})-doublet; (ψ_{65},ψ_{68})-doublet; (ψ_{66},ψ_{69})-doublet; $(\psi_{82},\psi_{83},\psi_{86})$-triplet; $(\psi_{99},\psi_{100},\psi_{101})$-triplet; $(\psi_{102},\psi_{108},\psi_{111})$-triplet; ψ_{18},ψ_{19}; ψ_{43},ψ_{44}; ψ_{54},ψ_{55}; ψ_{73},ψ_{74}; ψ_{84},ψ_{85}; ψ_{109},ψ_{110}-singlets. (2.1.1)

This choice fixes the matrix $\underset{\sim}{\Gamma}(E_{\alpha_5})$. For each of the other α_i-multiplets (i = 4,3,2,1) we consider all the loops (α_i,α_j) with $i > j$ and $i,j = 5,4,3,2,1$. If the loop (α_i,α_j) cannot specify the states of the α_j-multiplet, we again make an arbitrary choice of the states belonging to the α_j-multiplet.

For example, for the first arrangement



we have

<u>(1) α_5-direction</u>

We have chosen ((1.2.1)) the states as follows: $(\psi_{16},\psi_{17},\psi_{20})$ as a triplet, and ψ_{18},ψ_{19} as singlets.

<u>(2) α_4-direction</u>

Let us suppose

$$E_{-\alpha_4}\psi_{15} = a\psi_{17} + b\psi_{18} + c\psi_{19}. \tag{2.1.2}$$

We consider the (α_5,α_4) loop, for which we have

$$E_{\alpha_5}E_{-\alpha_4}\psi_{15} = E_{\alpha_5}(a\psi_{17}+b\psi_{18}+c\psi_{19})$$

$$= a\sqrt{2}\psi_{16} \text{ (from our previous choice).}$$

and (as $E_{\alpha_5}\psi_{15} = 0$)

$$E_{-\alpha_4}E_{\alpha_5}\psi_{15} = 0.$$

Thus $\alpha = 0$ and (2.1.2) becomes

$$E_{-\alpha_4}\psi_{15} = b\psi_{18} + c\psi_{19}.$$

From $[E_{\alpha_4},E_{-\alpha_4}] = H_{\alpha_4}$ we have, when applied to ψ_{15},

$$E_{\alpha_4}E_{-\alpha_4}\psi_{15} - E_{-\alpha_4}E_{\alpha_4}\psi_{15} = H_{\alpha_4}\psi_{15}.$$

As $E_{\alpha_4}\psi_{15} = 0$ we have

$$E_{\alpha_4}(b\psi_{18}+c\psi_{19}) = \frac{2(\alpha_1+\alpha_2+\alpha_3+1.5\alpha_4+0.5\alpha_5,\alpha_4)}{(\alpha_4,\alpha_4)}\psi_{15}$$

or $(b^2+c^2)\psi_{15} = 2\psi_{15}$, from which we get

$$b^2 + c^2 = 2. \tag{2.1.3}$$

From (2.1.3) we have the freedom of choosing ψ_{18} or ψ_{19} to belong to a triplet. We shall choose $(\psi_{15},\psi_{18},\psi_{21})$ to be a triplet, and ψ_{19} to be a singlet.

<u>(3) α_3-direction</u>

Let

$$E_{-\alpha_3}\psi_{14} = d\psi_{17} + e\psi_{18} + f\psi_{19}. \tag{2.1.4}$$

Then we consider the following loops:

(a) (α_5,α_3) loop

$$E_{\alpha_5}E_{-\alpha_3}\psi_{14} = E_{\alpha_5}(d\psi_{17}+e\psi_{18}+f\psi_{19})$$

$$= \sqrt{2}d\psi_{16} \text{ (from our choice in the } \alpha_5\text{-direction)}$$

and

$$E_{-\alpha_3}E_{\alpha_5}\psi_{14} = E_{-\alpha_3}(\psi_{13}) \text{ (using (4.2.13) of Chapter 4)}$$

$$= \psi_{16} \text{ (using (4.2.13) of Chapter 4)}$$

Thus $d = 1/\sqrt{2}$.

(b) (α_4,α_3) loop

We have as before

$$E_{\alpha_4}E_{-\alpha_3}\psi_{14} = E_{\alpha_4}(d\psi_{17}+e\psi_{18}+f\psi_{19}) \text{ (from (2.1.4))}$$

$$= \sqrt{2}e\psi_{15} \text{ (from our choice in the } \alpha_4\text{-direction)}$$

and

$$E_{-\alpha_3}E_{\alpha_4}\psi_{14} = E_{-\alpha_3}(\psi_{12}) \text{ (from (4.2.13) of Chapter 4)}$$

$$= \psi_{15} \text{ (from (4.2.13) of Chapter 4)}$$

Thus the value of $e = 1/\sqrt{2}$. Now the identity $[E_{+\alpha_3},E_{-\alpha_3}] = H_{\alpha_3}$ will determine the coefficient f. We have as before

$$E_{\alpha_3}E_{-\alpha_3}\psi_{14} - E_{-\alpha_3}E_{\alpha_3}\psi_{14} = H_{\alpha_3}\psi_{14}$$

$$E_{\alpha_3}(d\psi_{17}+e\psi_{18}+f\psi_{19}) = \frac{2(\alpha_1+\alpha_2+3\alpha_3+0.5\alpha_4+0.5\alpha_5,\alpha_3)}{(\alpha_3,\alpha_3)}\psi_{14}$$

from which we get

$$(d^2+e^2+f^2)\psi_{14} = 2\psi_{14}.$$

Thus

$$d^2 + e^2 + f^2 = 2.$$

Substituting the values of d and e we have $f^2 = 1$, and we shall choose $f = 1$.

(4) α_2-direction

Let $E_{-\alpha_2}\psi_9 = g\psi_{17} + h\psi_{18} + i\psi_{19}$. (2.1.5)

(a) (α_5,α_2) loop

$$E_{\alpha_5}E_{-\alpha_2}\psi_9 = E_{\alpha_5}(g\psi_{17}+h\psi_{18}+i\psi_{19}) \text{ (from (2.1.5))}$$

$$= g\sqrt{2}\psi_{16} \text{ (from the choice in the } \alpha_5\text{-direction)}$$

As $E_{\alpha_5}\psi_9 = 0$, we have $E_{-\alpha_2}E_{\alpha_5}\psi_9 = 0$. Thus $g = 0$.

(b) (α_4,α_2) loop

$$E_{\alpha_4}E_{-\alpha_2}\psi_9 = E_{\alpha_4}(h\psi_{18}+i\psi_{19}) \text{ (because } g = 0)$$

$$= \sqrt{2}h\psi_{15} \text{ (using (4.2.13) of Chapter 4)}$$

and

$$E_{-\alpha_2}E_{\alpha_4}\psi_9 = E_{-\alpha_2}(\sqrt{2}\psi_8)$$

$$= 2\psi_{15} \text{ (using (4.2.13) of Chapter 4).}$$

Thus $h = \sqrt{2}$.

(c) (α_3,α_2) loop

$$E_{\alpha_3}E_{-\alpha_2}\psi_9 = E_{\alpha_3}(h\psi_{18}+i\psi_{19}) \text{ (because } g = 0)$$

$$= \frac{1}{\sqrt{2}}h\psi_{14} + i\psi_{14} \text{ (using (4.2.13) of Chapter 4)}$$

$$= (\frac{1}{\sqrt{2}}h+i)\psi_{14}$$

and

$$E_{-\alpha_2}E_{\alpha_3}\psi_9 = E_{-\alpha_2}(\psi_7)$$

$$= \psi_{14} \text{ (using (4.2.13) of Chapter 4).}$$

Thus $\frac{1}{\sqrt{2}}h + i = 1$, and, substituting the value of h, we get $i = 0$.

Exactly the same manipulations are repeated for each of the other arrangements.

The same analysis is applied to the $\underset{\sim}{120}$ representation. The weight diagram has the same structure as the weight diagram of the $\underset{\sim}{126}$ representation, with the only difference being that the non-simple weights have multiplicity four. Program D5(4) with some changes can be used to generate the matrix elements $\underset{\sim}{\Gamma}(E_{\alpha_i})$, $i = 1,2,\ldots,5$, of the $\underset{\sim}{120}$ representation.

In Tables 5.5 - 5.7, we give the matrix elements of the generators E_{α_i}, $i = 1,2,\ldots,5$, for the representations $\underset{\sim}{126}$, $\underset{\sim}{120}$ and $\underset{\sim}{10}$. Using these matrix elements as the input of the test Program D6(4), we have verified the commutation relations for the above representations.

Description of Program D5(4)

Program D5(4) is a generalization of Program D4(4). It is more complicated than Program D4(4), because it performs calculations in a five dimensional space, and because, having employed the sparse array technique to save space in the computer memory, we are forced to use various techniques to perform the basic operations of the sparse arrays like multiplication or division. For example, the multiplication of two sparse arrays cannot be performed row by column, but only coordinate by coordinate, and for these coordinates where the corresponding entry of the matrix is non-zero. This makes it difficult for the reader to read and understand the program.

The program starts reading the weights and the simple roots from the data cards. The procedure <u>VERTICAL</u> fixes the matrix $\underset{\sim}{\Gamma}(E_{\alpha_5})$. When it is called, first it evaluates the difference of the weights. If this difference is equal to α_5, then it proceeds recursively and fixes the state in each multiplet.

The control variable <u>K</u> which takes integer values in the range $5 < i \leq 1$ in decreasing order, controls the calculations of all the other matrices $\underset{\sim}{\Gamma}(\alpha_i)$, $i = 4,3,2,1$. When, for example, the control

THE GENERATOR EA1 IS
****

11	36	1.00
12	37	1.00
13	38	1.00
14	39	1.00
15	40	1.00
16	41	1.00
17	42	1.00
18	43	1.00
19	44	1.00
20	45	1.00
21	46	1.00
22	47	1.00
23	48	1.00
24	49	1.00
25	50	1.00
26	51	1.41
51	92	1.41
27	54	1.41
54	93	1.41
28	57	1.41
57	94	1.41
29	59	1.41
59	95	1.41
30	62	1.41
62	96	1.41
31	65	1.41
65	97	1.41
32	68	1.41
68	98	1.41
33	71	1.41
71	99	1.41
34	72	1.41
72	100	1.41
35	75	1.41
75	101	1.41
77	102	1.00
78	103	1.00
79	104	1.00
80	105	1.00
81	106	1.00
82	107	1.00
83	108	1.00
84	109	1.00
85	110	1.00
86	111	1.00
87	112	1.00
88	113	1.00
89	114	1.00
90	115	1.00
91	116	1.00

THE GENERATOR EA2 IS
****

4	11	1.00
5	12	1.00
6	13	1.00
7	14	1.00
8	15	1.41
15	26	1.41
9	18	1.41
18	27	1.41
10	21	1.41
21	28	1.41
22	29	1.00
23	30	1.00
24	31	1.00
25	32	1.00
40	51	1.00
41	52	1.00
42	53	1.00
43	54	1.00
44	55	1.00
45	56	1.00
46	57	1.00
47	58	0.71
58	77	0.71
47	59	0.71
59	77	0.71
47	60	1.00
60	77	1.00
48	61	0.71
61	78	0.71
48	62	0.71
62	78	0.71
48	63	1.00
63	78	1.00
49	64	0.71
64	79	0.71
49	65	0.71
65	79	0.71
49	66	1.00
66	79	1.00
50	67	0.71
67	80	0.71
50	68	0.71
68	80	0.71
50	69	1.00
69	80	1.00
70	81	1.00
71	82	1.00
72	83	1.00
73	84	1.00
74	85	1.00
75	86	1.00
76	87	1.00
95	102	1.00
96	103	1.00
97	104	1.00
98	105	1.00
99	107	1.41
107	117	1.41
100	108	1.41
108	118	1.41
101	111	1.41
111	119	1.41
113	120	1.00
114	121	1.00
115	122	1.00
116	123	1.00

THE GENERATOR EA3 IS
****

2	4	1.00
3	5	1.41
5	8	1.41
7	9	1.00
12	15	1.00
13	16	1.00
14	17	0.71
17	22	0.71
14	18	0.71
18	22	0.71
14	19	1.00
19	22	1.00
20	23	1.00
21	24	1.00
27	29	1.00
28	31	1.41
31	33	1.41
32	34	1.00
37	40	1.00
38	41	1.00
39	42	0.71
42	47	0.71
39	43	0.71
43	47	0.71
39	44	1.00
44	47	1.00
45	48	1.00
46	49	1.00
53	58	1.00
54	59	1.00
55	60	1.00
56	61	1.41
61	70	1.41
57	65	1.41
65	71	1.41
67	73	1.00
68	72	1.00
69	74	1.00
78	81	1.00
79	82	1.00
80	83	0.71
83	88	0.71
80	84	0.71
84	88	0.71
80	85	1.00
85	88	1.00
86	89	1.00
87	90	1.00
93	95	1.00
94	97	1.41
97	99	1.41
98	100	1.00

103	106	1.00
104	107	1.00
105	108	0.71
108	113	0.71
105	109	0.71
109	113	0.71
105	110	1.00
110	113	1.00
111	114	1.00
112	115	1.00
118	120	1.00
119	121	1.41
121	124	1.41
123	125	1.00

THE GENERATOR EA4 IS
****

4	6	1.00
5	7	1.00
8	9	1.41
9	10	1.41
11	13	1.00
12	14	1.00
15	18	1.41
18	21	1.41
22	24	1.00
23	25	1.00
26	27	1.41
27	28	1.41
29	31	1.00
30	32	1.00
36	38	1.00
37	39	1.00
40	43	1.41
43	46	1.41
47	49	1.00
48	50	1.00
51	54	1.41
54	57	1.41
58	64	1.00
59	65	1.00
60	66	1.00
61	67	1.00
62	68	1.00
63	69	1.00
70	73	1.41
73	76	1.41
77	79	1.00
78	80	1.00
81	84	1.41
84	87	1.41
88	90	1.00
89	91	1.00
92	93	1.41
93	94	1.41
95	97	1.00
96	98	1.00
102	104	1.00
103	105	1.00
106	109	1.41
109	112	1.41
113	115	1.00
114	116	1.00
120	122	1.00
121	123	1.00
124	125	1.41
125	126	1.41

THE GENERATOR EA5 IS
****

1	2	1.41
2	3	1.41
4	5	1.00
6	7	1.00
11	12	1.00
13	14	1.00
16	17	1.41
17	20	1.41
22	23	1.00
24	25	1.00
29	30	1.00
31	32	1.00
33	34	1.41
34	35	1.41
36	37	1.00
38	39	1.00
41	42	1.41
42	45	1.41
47	48	1.00
49	50	1.00
52	53	1.41
53	56	1.41
58	61	1.00
59	62	1.00
60	63	1.00
64	67	1.00
65	68	1.00
66	69	1.00
71	72	1.41
72	75	1.41
77	78	1.00
79	80	1.00
82	83	1.41
83	86	1.41
88	89	1.00
90	91	1.00
95	96	1.00
97	98	1.00
99	100	1.41
100	101	1.41
102	103	1.00
104	105	1.00
107	108	1.41
108	111	1.41
113	114	1.00
115	116	1.00
117	118	1.41
118	119	1.41
120	121	1.00
122	123	1.00

Table 5.5: Matrix elements of the non-diagonal generators of the 126 representation

THE GENERATOR EA1 IS
#####################

7	29	1.00
8	30	1.00
9	31	1.00
10	32	1.00
11	33	1.00
12	34	1.00
13	35	1.00
14	36	1.00
15	37	1.00
16	38	1.00
17	39	1.00
18	40	1.00
19	41	1.00
20	42	1.00
21	43	1.00
22	44	1.00
23	49	1.41
49	93	1.41
24	55	1.41
55	94	1.41
25	59	1.41
59	95	1.41
26	63	1.41
63	96	1.41
27	67	1.41
67	97	1.41
28	73	1.41
73	98	1.41
77	99	1.00
78	100	1.00
79	101	1.00
80	102	1.00
81	103	1.00
82	104	1.00
83	105	1.00
84	106	1.00
85	107	1.00
86	108	1.00
87	109	1.00
88	110	1.00
89	111	1.00
90	112	1.00
91	113	1.00
92	114	1.00

THE GENERATOR EA2 IS
#####################

2	7	1.00
3	8	1.00
4	9	1.00
5	10	1.00
6	15	1.41
15	23	1.41
19	24	1.00
20	25	1.00
21	26	1.00
22	27	1.00
33	45	1.00
34	46	1.00
35	47	1.00
36	48	1.00
37	49	1.00
38	50	1.00
39	51	1.00
40	52	1.00
41	53	0.71
53	77	0.71
41	54	0.71
54	77	0.71
41	55	0.71
55	77	0.71
41	56	0.71
56	77	0.71
42	57	0.71
57	78	0.71
42	58	0.71
58	78	0.71
42	59	0.71
59	78	0.71
42	60	0.71
60	78	0.71
43	61	0.71
61	79	0.71
43	62	0.71
62	79	0.71
43	63	0.71
63	79	0.71
43	64	0.71
64	79	0.71
44	65	0.71
65	80	0.71
44	66	0.71
66	80	0.71
44	67	0.71
67	80	0.71
44	68	0.71
68	80	0.71
69	81	1.00
70	82	1.00
71	83	1.00
72	84	1.00
73	85	1.00
74	86	1.00
75	87	1.00
76	88	1.00
94	99	1.00
95	100	1.00
96	101	1.00
97	102	1.00
98	107	1.41
107	115	1.41
111	116	1.00
112	117	1.00
113	118	1.00
114	119	1.00

THE GENERATOR EA3 IS
#####################

1	2	1.00
5	6	1.00
8	11	1.00
9	12	1.00
10	13	0.71
13	19	0.71
10	14	0.71
14	19	0.71
10	15	0.71
15	19	0.71
10	16	0.71
16	19	0.71
17	20	1.00
18	21	1.00
23	24	1.00
27	28	1.00
30	33	1.00
31	34	1.00
32	35	0.71
35	41	0.71
32	36	0.71
36	41	0.71
32	37	0.71
37	41	0.71
32	38	0.71
38	41	0.71
39	42	1.00
40	43	1.00
47	53	1.00
48	54	1.00
49	55	1.00
50	56	1.00
51	57	1.41
57	69	1.41
52	62	1.41
62	70	1.41
65	72	1.00
66	71	1.00
67	73	1.00
68	74	1.00
78	81	1.00
79	82	1.00
80	83	0.71
83	89	0.71
80	84	0.71
84	89	0.71
80	85	0.71
85	89	0.71
80	86	0.71
86	89	0.71
87	90	1.00
88	91	1.00

93	94	1.00
97	98	1.00
100	103	1.00
101	104	1.00
102	105	0.71
105	111	0.71
102	106	0.71
106	111	0.71
102	107	0.71
107	111	0.71
102	108	0.71
108	111	0.71
109	112	1.00
110	113	1.00
115	116	1.00
119	120	1.00

THE GENERATOR EA4 IS
####################

2	4	1.00
3	5	1.00
7	9	1.00
8	10	1.00
11	14	1.41
14	18	1.41
19	21	1.00
20	22	1.00
24	26	1.00
25	27	1.00
29	31	1.00
30	32	1.00
33	36	1.41
36	40	1.41
41	43	1.00
42	44	1.00
45	48	1.41
48	52	1.41
53	61	1.00
54	62	1.00
55	63	1.00
56	64	1.00
57	65	1.00
58	66	1.00
59	67	1.00

60	68	1.00
69	72	1.41
72	76	1.41
77	79	1.00
78	80	1.00
81	84	1.41
84	88	1.41
89	91	1.00
90	92	1.00
94	96	1.00
95	97	1.00
99	101	1.00
100	102	1.00
103	106	1.41
106	110	1.41
111	113	1.00
112	114	1.00
116	118	1.00
117	119	1.00

THE GENERATOR EA5 IS
####################

2	3	1.00
4	5	1.00
7	8	1.00
9	10	1.00
12	13	1.41
13	17	1.41
19	20	1.00
21	22	1.00
24	25	1.00
26	27	1.00
29	30	1.00
31	32	1.00
34	35	1.41
35	39	1.41
41	42	1.00
43	44	1.00
46	47	1.41
47	51	1.41
53	57	1.00
54	58	1.00
55	59	1.00
56	60	1.00

61	65	1.00
62	66	1.00
63	67	1.00
64	68	1.00
70	71	1.41
71	75	1.41
77	78	1.00
79	80	1.00
82	83	1.41
83	87	1.41
89	90	1.00
91	92	1.00
94	95	1.00
96	97	1.00
99	100	1.00
101	102	1.00
104	105	1.41
105	109	1.41
111	112	1.00
113	114	1.00
116	117	1.00
118	119	1.00

Table 5.6: Matrix elements of the non-diagonal generators of the 120 representation

THE GENERATOR EAI IS

4	5	1.00
6	7	1.00

THE GENERATOR EA2 IS

4	6	1.00
5	7	1.00

THE GENERATOR EA3 IS

3	4	1.00
7	8	1.00

THE GENERATOR EA2 IS

2	3	1.00
8	9	1.00

THE GENERATOR EAI IS

1	2	1.00
9	10	1.00

Table 5.7: Matrix elements of the non-diagonal generators of the 10 representation

variable K takes the value K = i, then the procedure HORIZONTAL is called and again finds the difference of the weights. If this difference is equal to α_i, then the length of the multiplet is evaluated. If a state in that multiplet has multiplicity greater than one, the procedure LOOP is called and loop calculations are performed for all the possible loops (α_j, α_i) with $j > i$. If all the states of a multiplet have multiplicity one, then the procedure LOWER is called, and the magnitudes of the corresponding matrix elements are calculated, while the procedure FILLING fixes the matrix elements throughout the whole multiplet.

If the procedure LOOP cannot fix the matrix elements of a multiplet, then again the procedure VERTICAL is called and makes a choice of the states.

The input of the program is:
the rank of the algebra; the dimension of the representation; the maximum weights multiplicity of the representation increased by one; the weights; the simple roots.

The output consists of an array with three columns. In the first two columns the coordinates of a non-zero matrix element are given, while in the third its value is stated.

§5.2 Clebsch-Gordan Coefficients

From Chapter 4 we have the following Clebsch-Gordan series

$$\underset{\sim}{16} \otimes \underset{\sim}{16} = \underset{\sim}{126} \oplus \underset{\sim}{120} \oplus \underset{\sim}{10}. \tag{2.1.1}$$

In terms of the basic functions (2.1.1) can be written

$$\psi^{16}_{i,\nu} \otimes \psi^{16'}_{j,\mu} = \begin{pmatrix} 16 & 16' \\ i,\nu & j,\mu \end{pmatrix}\!\Big|\, \begin{matrix} 126 \\ k,\pi \end{matrix}\Big) \psi^{126}_{k,\pi} + \begin{pmatrix} 16 & 16' \\ i,\nu & j,\mu \end{pmatrix}\!\Big|\, \begin{matrix} 120 \\ \ell,\lambda \end{matrix}\Big) \psi^{120}_{\ell,\lambda}$$

$$+ \begin{pmatrix} 16 & 16' \\ i,\nu & j,\mu \end{pmatrix}\!\Big|\, \begin{matrix} 10 \\ m,\omega \end{matrix}\Big) \psi^{10}_{m,\omega}, \tag{2.1.2}$$

in a notation which is consistent with our previous development.

The procedure of evaluating the above Clebsch-Gordan coefficients will not be given here, because it is exactly the same as in the case of G_2. Program E2(5), which is given in Appendix C, generates the Clebsch-Gordan coefficients of (2.1.2) in a similar way to Program E1(5). In Tables 5.8 - 5.10, the set of the Clebsch-Gordan coefficients is given.

THE C.G.COEF OF THE REPR=126 ARE
****

THE STATE NUM I=1

1	1	1.00

THE STATE NUM I=2

1	2	0.71
2	1	0.71

THE STATE NUM I=3

2	2	1.00

THE STATE NUM I=4

1	3	0.71
3	1	0.71

THE STATE NUM I=5

2	3	0.71
3	2	0.71

THE STATE NUM I=6

1	4	0.71
4	1	0.71

THE STATE NUM I=7

2	4	0.71
4	2	0.71

THE STATE NUM I=8

3	3	1.00

THE STATE NUM I=9

3	4	0.71
4	3	0.71

THE STATE NUM I=10

4	4	1.00

THE STATE NUM I=11

1	5	0.71
5	1	0.71

THE STATE NUM I=12

2	5	0.71
5	2	0.71

THE STATE NUM I=13

1	6	0.71
6	1	0.71

THE STATE NUM I=14

2	6	0.71
6	2	0.71

THE STATE NUM I=15

3	5	0.71
5	3	0.71

THE STATE NUM I=16

1	7	0.71
7	1	0.71

THE STATE NUM I=17

1	8	0.50
2	7	0.50
7	2	0.50
8	1	0.50

THE STATE NUM I=18

3	6	0.50
4	5	0.50
5	4	0.50
6	3	0.50

THE STATE NUM I=19

1	8	-0.35
2	7	0.35
3	6	0.35
4	5	-0.35
5	4	-0.35
6	3	0.35
7	2	0.35
8	1	-0.35

THE STATE NUM I=20

2	8	0.71
8	2	0.71

THE STATE NUM I=21

4	6	0.71
6	4	0.71

THE STATE NUM I=22

3	7	0.71
7	3	0.71

THE STATE NUM I=23

3	8	0.71
8	3	0.71

THE STATE NUM I=24

4	7	0.71
7	4	0.71

THE STATE NUM I=25

4	8	0.71
8	4	0.71

THE STATE NUM I=26

5	5	1.00

THE STATE NUM I=27

5	6	0.71
6	5	0.71

THE STATE NUM I=28

6	6	1.00

THE STATE NUM I=29

5	7	0.71
7	5	0.71

THE STATE NUM I=30

5	8	0.71
8	5	0.71

THE STATE NUM I=31

6	7	0.71
7	6	0.71

THE STATE NUM I=32

6	8	0.71
8	6	0.71

THE STATE NUM I=33

7	7	1.00

THE STATE NUM I=34

7	8	0.71
8	7	0.71

THE STATE NUM I=35

8	8	1.00

THE STATE NUM I=36

1	9	0.71
9	1	0.71

THE STATE NUM I=37

2	9	0.71
9	2	0.71

THE STATE NUM I=38

1	10	0.71
10	1	0.71

THE STATE NUM I=39

2	10	0.71
10	2	0.71

THE STATE NUM I=40

3	9	0.71
9	3	0.71

THE STATE NUM I=41

1	11	0.71
11	1	0.71

THE STATE NUM I=42

1	12	0.50
2	11	0.50
11	2	0.50
12	1	0.50

THE STATE NUM I=43

3	10	0.50
4	9	0.50
9	4	0.50
10	3	0.50

THE STATE NUM I=44

1	12	-0.35
2	11	0.35
3	10	0.35
4	9	-0.35
9	4	-0.35
10	3	0.35
11	2	0.35
12	1	-0.35

THE STATE NUM I=45

2	12	0.71
12	2	0.71

THE STATE NUM I=46

4	10	0.71
10	4	0.71

THE STATE NUM I=47

3	11	0.71
11	3	0.71

THE STATE NUM I=48

3	12	0.71
12	3	0.71

THE STATE NUM I=49

4	11	0.71
11	4	0.71

THE STATE NUM I=50

4	12	0.71
12	4	0.71

THE STATE NUM I=51

5	9	0.71
9	5	0.71

THE STATE NUM I=52

1	13	0.71
13	1	0.71

THE STATE NUM I=53

1	14	0.50
2	13	0.50
13	2	0.50
14	1	0.50

THE STATE NUM I=54

5	10	0.50
6	9	0.50
9	6	0.50
10	5	0.50

THE STATE NUM I=55

1	14	-0.35
2	13	0.35
5	10	0.35
6	9	-0.35
9	6	-0.35
10	5	0.35
13	2	0.35
14	1	-0.35

THE STATE NUM I=56

2	14	0.71
14	2	0.71

THE STATE NUM I=57

6	10	0.71
10	6	0.71

THE STATE NUM I=58

1	15	0.50
3	13	0.50
13	3	0.50
15	1	0.50

THE STATE NUM I=59

5	11	0.50
7	9	0.50
9	7	0.50
11	5	0.50

THE STATE NUM I=60

1	15	-0.35
3	13	0.35
5	11	0.35
7	9	-0.35
9	7	-0.35
11	5	0.35
13	3	0.35
15	1	-0.35

THE STATE NUM I=61

2	15	0.50
3	14	0.50
14	3	0.50
15	2	0.50

THE STATE NUM I=62

5	12	0.50
8	9	0.50
9	8	0.50
12	5	0.50

THE STATE NUM I=63

2	15	-0.35
3	14	0.35
5	12	0.35
8	9	-0.35
9	8	-0.35
12	5	0.35
14	3	0.35
15	2	-0.35

THE STATE NUM I=64

1	16	0.50
4	13	0.50
13	4	0.50
16	1	0.50

THE STATE NUM I=65

6	11	0.50
7	10	0.50
10	7	0.50
11	6	0.50

THE STATE NUM I=66

1	16	-0.35
4	13	0.35
6	11	0.35
7	10	-0.35
10	7	-0.35
11	6	0.35
13	4	0.35
16	1	-0.35

THE STATE NUM I=67

2	16	0.50
4	14	0.50
14	4	0.50
16	2	0.50

THE STATE NUM I=68

6	12	0.50
8	10	0.50
10	8	0.50
12	6	0.50

THE STATE NUM I=69

2	16	-0.35
4	14	0.35
6	12	0.35
8	10	-0.35
10	8	-0.35
12	6	0.35
14	4	0.35
16	2	-0.35

THE STATE NUM I=70

3	15	0.71
15	3	0.71

THE STATE NUM I=71

7	11	0.71
11	7	0.71

THE STATE NUM I=72

7	12	0.50
8	11	0.50
11	8	0.50
12	7	0.50

THE STATE NUM I=73

3	16	0.50
4	15	0.50
15	4	0.50
16	3	0.50

THE STATE NUM I=74

3	16	-0.35
4	15	0.35
7	12	0.35
8	11	-0.35
11	8	-0.35
12	7	0.35
15	4	0.35
16	3	-0.35

THE STATE NUM I=75

8	12	0.71
12	8	0.71

THE STATE NUM I=76

4	16	0.71
16	4	0.71

THE STATE NUM I=77

5	13	0.71
13	5	0.71

THE STATE NUM I=78

5	14	0.71
14	5	0.71

THE STATE NUM I=79

6	13	0.71
13	6	0.71

THE STATE NUM I=80

6	14	0.71
14	6	0.71

THE STATE NUM I=81

5	15	0.71
15	5	0.71

THE STATE NUM I=82

7	13	0.71
13	7	0.71

THE STATE NUM I=83

7	14	0.50
8	13	0.50
13	8	0.50
14	7	0.50

THE STATE NUM I=84

5	16	0.50
6	15	0.50
15	6	0.50
16	5	0.50

THE STATE NUM I=85

5	16	-0.35
6	15	0.35
7	14	0.35
8	13	-0.35
13	8	-0.35
14	7	0.35
15	6	0.35
16	5	-0.35

THE STATE NUM I=86

8	14	0.71
14	8	0.71

THE STATE NUM I=87

6	16	0.71
16	6	0.71

THE STATE NUM I=88

7	15	0.71
15	7	0.71

THE STATE NUM I=89

8	15	0.71
15	8	0.71

THE STATE NUM I=90

7	16	0.71
16	7	0.71

THE STATE NUM I=91

8	16	0.71
16	8	0.71

THE STATE NUM I=92

9	9	1.00

THE STATE NUM I=93

9	10	0.71
10	9	0.71

THE STATE NUM I=94

10	10	1.00

THE STATE NUM I=95

9	11	0.71
11	9	0.71

THE STATE NUM I=96

9	12	0.71
12	9	0.71

THE STATE NUM I=97

10	11	0.71
11	10	0.71

THE STATE NUM I=98

10	12	0.71
12	10	0.71

THE STATE NUM I=99

11	11	1.00

THE STATE NUM I=100

11	12	0.71
12	11	0.71

THE STATE NUM I=101

12	12	1.00

THE STATE NUM I=102

9	13	0.71
13	9	0.71

THE STATE NUM I=103

9	14	0.71
14	9	0.71

THE STATE NUM I=104

10	13	0.71
13	10	0.71

THE STATE NUM I=105

10	14	0.71
14	10	0.71

THE STATE NUM I=106

9	15	0.71
15	9	0.71

THE STATE NUM I=107

11	13	0.71
13	11	0.71

THE STATE NUM I=108

11	14	0.50
12	13	0.50
13	12	0.50
14	11	0.50

THE STATE NUM I=109

9	16	0.50
10	15	0.50
15	10	0.50
16	9	0.50

THE STATE NUM I=110

9	16	-0.35
10	15	0.35
11	14	0.35
12	13	-0.35
13	12	-0.35
14	11	0.35
15	10	0.35
16	9	-0.35

THE STATE NUM I=111

12	14	0.71
14	12	0.71

THE STATE NUM I=112

10	16	0.71
16	10	0.71

THE STATE NUM I=113

11	15	0.71
15	11	0.71

THE STATE NUM I=114

12	15	0.71
15	12	0.71

THE STATE NUM I=115

11	16	0.71
16	11	0.71

THE STATE NUM I=116

12	16	0.71
16	12	0.71

THE STATE NUM I=117

13	13	1.00

THE STATE NUM I=118

13	14	0.71
14	13	0.71

THE STATE NUM I=119

14	14	1.00

THE STATE NUM I=120

13	15	0.71
15	13	0.71

THE STATE NUM I=121

14	15	0.71
15	14	0.71

THE STATE NUM I=122

13	16	0.71
16	13	0.71

THE STATE NUM I=123

14	16	0.71
16	14	0.71

THE STATE NUM I=124

15	15	1.00

THE STATE NUM I=125

15	16	0.71
16	15	0.71

THE STATE NUM I=126

16	16	1.00

Table 5.8

THE C.G.COEF OF THE REPR=120 ARE
****

THE STATE NUM I=1

1	2	0.71
2	1	-0.71

THE STATE NUM I=2

1	3	0.71
3	1	-0.71

THE STATE NUM I=3

2	3	0.71
3	2	-0.71

THE STATE NUM I=4

1	4	0.71
4	1	-0.71

THE STATE NUM I=5

2	4	0.71
4	2	-0.71

THE STATE NUM I=6

3	4	0.71
4	3	-0.71

THE STATE NUM I=7

1	5	0.71
5	1	-0.71

THE STATE NUM I=8

2	5	0.71
5	2	-0.71

THE STATE NUM I=9

1	6	0.71
6	1	-0.71

THE STATE NUM I=10

2	6	0.71
6	2	-0.71

THE STATE NUM I=11

3	5	0.71
5	3	-0.71

THE STATE NUM I=12

1	7	0.71
7	1	-0.71

THE STATE NUM I=13

1	8	0.50
2	7	0.50
7	2	-0.50
8	1	-0.50

THE STATE NUM I=14

3	6	0.50
4	5	0.50
5	4	-0.50
6	3	-0.50

THE STATE NUM I=15

3	6	0.50
4	5	-0.50
5	4	0.50
6	3	-0.50

THE STATE NUM I=16

1	8	-0.50
2	7	0.50
7	2	-0.50
8	1	0.50

THE STATE NUM I=17

2	8	0.71
8	2	-0.71

THE STATE NUM I=18

4	6	0.71
6	4	-0.71

THE STATE NUM I=19

3	7	0.71
7	3	-0.71

THE STATE NUM I=20

3	8	0.71
8	3	-0.71

THE STATE NUM I=21

4	7	0.71
7	4	-0.71

THE STATE NUM I=22

4	8	0.71
8	4	-0.71

THE STATE NUM I=23

5	6	0.71
6	5	-0.71

THE STATE NUM I=24

5	7	0.71
7	5	-0.71

THE STATE NUM I=25

5	8	0.71
8	5	-0.71

THE STATE NUM I=26

6	7	0.71
7	6	-0.71

THE STATE NUM I=27

6	8	0.71
8	6	-0.71

THE STATE NUM I=28

7	8	0.71
8	7	-0.71

THE STATE NUM I=29

1	9	0.71
9	1	-0.71

THE STATE NUM I=30

2	9	0.71
9	2	-0.71

THE STATE NUM I=31

1	10	0.71
10	1	-0.71

THE STATE NUM I=32

2	10	0.71
10	2	-0.71

THE STATE NUM I=33

3	9	0.71
9	3	-0.71

THE STATE NUM I=34

1	11	0.71
11	1	-0.71

THE STATE NUM I=35

1	12	0.50
2	11	0.50
11	2	-0.50
12	1	-0.50

THE STATE NUM I=36

3	10	0.50
4	9	0.50
9	4	-0.50
10	3	-0.50

THE STATE NUM I=37

3	10	0.50
4	9	-0.50
9	4	0.50
10	3	-0.50

THE STATE NUM I=38

1	12	-0.50
2	11	0.50
11	2	-0.50
12	1	0.50

THE STATE NUM I=39

2	12	0.71
12	2	-0.71

THE STATE NUM I=40

4	10	0.71
10	4	-0.71

THE STATE NUM I=41

3	11	0.71
11	3	-0.71

THE STATE NUM I=42

3	12	0.71
12	3	-0.71

THE STATE NUM I=43

4	11	0.71
11	4	-0.71

THE STATE NUM I=44

4	12	0.71
12	4	-0.71

THE STATE NUM I=45

5	9	0.71
9	5	-0.71

THE STATE NUM I=46

1	13	0.71
13	1	-0.71

THE STATE NUM I=47

1	14	0.50
2	13	0.50
13	2	-0.50
14	1	-0.50

THE STATE NUM I=48

5	10	0.50
6	9	0.50
9	6	-0.50
10	5	-0.50

THE STATE NUM I=49

5	10	0.50
6	9	-0.50
9	6	0.50
10	5	-0.50

THE STATE NUM I=50

1	14	-0.50
2	13	0.50
13	2	-0.50
14	1	0.50

THE STATE NUM I=51

2	14	0.71
14	2	-0.71

THE STATE NUM I=52

6	10	0.71
10	6	-0.71

THE STATE NUM I=53

1	15	0.50
3	13	0.50
13	3	-0.50
15	1	-0.50

THE STATE NUM I=54

5	11	0.50
7	9	0.50
9	7	-0.50
11	5	-0.50

THE STATE NUM I=55

5	11	0.50
7	9	-0.50
9	7	0.50
11	5	-0.50

THE STATE NUM I=56

1	15	-0.50
3	13	0.50
13	3	-0.50
15	1	0.50

THE STATE NUM I=57

2	15	0.50
3	14	0.50
14	3	-0.50
15	2	-0.50

THE STATE NUM I=58

5	12	0.50
8	9	0.50
9	8	-0.50
12	5	-0.50

THE STATE NUM I=59

5	12	0.50
8	9	-0.50
9	8	0.50
12	5	-0.50

THE STATE NUM I=60

2	15	-0.50
3	14	0.50
14	3	-0.50
15	2	0.50

THE STATE NUM I=61

1	16	0.50
4	13	-0.50
13	4	-0.50
16	1	-0.50

THE STATE NUM I=62

6	11	0.50
7	10	0.50
10	7	-0.50
11	6	-0.50

THE STATE NUM I=63

6	11	0.50
7	10	-0.50
10	7	0.50
11	6	-0.50

THE STATE NUM I=64

1	16	-0.50
4	13	0.50
13	4	-0.50
16	1	0.50

THE STATE NUM I=65

2	16	0.50
4	14	0.50
14	4	-0.50
16	2	-0.50

THE STATE NUM I=66

6	12	0.50
8	10	0.50
10	8	-0.50
12	6	-0.50

THE STATE NUM I=67

6	12	0.50
8	10	-0.50
10	8	0.50
12	6	-0.50

THE STATE NUM I=68

2	16	-0.50
4	14	0.50
14	4	-0.50
16	2	0.50

THE STATE NUM I=69

3	15	0.71
15	3	-0.71

THE STATE NUM I=70

7	11	0.71
11	7	-0.71

THE STATE NUM I=71

7	12	0.50
8	11	0.50
11	8	-0.50
12	7	-0.50

THE STATE NUM I=72

3	16	0.50
4	15	0.50
15	4	-0.50
16	3	-0.50

THE STATE NUM I=73

7	12	0.50
8	11	-0.50
11	8	0.50
12	7	-0.50

THE STATE NUM I=74

3	16	-0.50
4	15	0.50
15	4	-0.50
16	3	0.50

THE STATE NUM I=75

8	12	0.71
12	8	-0.71

THE STATE NUM I=76

4	16	0.71
16	4	-0.71

THE STATE NUM I=77

5	13	0.71
13	5	-0.71

THE STATE NUM I=78

5	14	0.71
14	5	-0.71

THE STATE NUM I=79

6	13	0.71
13	6	-0.71

THE STATE NUM I=80

6	14	0.71
14	6	-0.71

THE STATE NUM I=81

5	15	0.71
15	5	-0.71

THE STATE NUM I=82

7	13	0.71
13	7	-0.71

THE STATE NUM I=83

7	14	0.50
8	13	0.50
13	8	-0.50
14	7	-0.50

THE STATE NUM I=84

5	16	0.50
6	15	0.50
15	6	-0.50
16	5	-0.50

THE STATE NUM I=85

7	14	0.50
8	13	-0.50
13	8	0.50
14	7	-0.50

THE STATE NUM I=86

5	16	-0.50
6	15	0.50
15	6	-0.50
16	5	0.50

THE STATE NUM I=87

8	14	0.71
14	8	-0.71

THE STATE NUM I=88

6	16	0.71
16	6	-0.71

THE STATE NUM I=89

7	15	0.71
15	7	-0.71

THE STATE NUM I=90

8	15	0.71
15	8	-0.71

THE STATE NUM I=91

7	16	0.71
16	7	-0.71

THE STATE NUM I=92

8	16	0.71
16	8	-0.71

THE STATE NUM I=93

9	10	0.71
10	9	-0.71

THE STATE NUM I=94

9	11	0.71
11	9	-0.71

THE STATE NUM I=95

9	12	0.71
12	9	-0.71

THE STATE NUM I=96

10	11	0.71
11	10	-0.71

THE STATE NUM I=97

10	12	0.71
12	10	-0.71

THE STATE NUM I=98

11	12	0.71
12	11	-0.71

THE STATE NUM I=99

9	13	0.71
13	9	-0.71

THE STATE NUM I=100

9	14	0.71
14	9	-0.71

THE STATE NUM I=101

10	13	0.71
13	10	-0.71

THE STATE NUM I=102

10	14	0.71
14	10	-0.71

THE STATE NUM I=103

9	15	0.71
15	9	-0.71

THE STATE NUM I=104

11	13	0.71
13	11	-0.71

THE STATE NUM I=105

11	14	0.50
12	13	0.50
13	12	-0.50
14	11	-0.50

THE STATE NUM I=106

9	16	0.50
10	15	0.50
15	10	-0.50
16	9	-0.50

THE STATE NUM I=107

11	14	0.50
12	13	-0.50
13	12	0.50
14	11	-0.50

THE STATE NUM I=108

9	16	-0.50
10	15	0.50
15	10	-0.50
16	9	0.50

THE STATE NUM I=109

12	14	0.71
14	12	-0.71

THE STATE NUM I=110

10	16	0.71
16	10	-0.71

THE STATE NUM I=111

11	15	0.71
15	11	-0.71

THE STATE NUM I=112

12	15	0.71
15	12	-0.71

THE STATE NUM I=113

11	16	0.71
16	11	-0.71

THE STATE NUM I=114

12	16	0.71
16	12	-0.71

THE STATE NUM I=115

13	14	0.71
14	13	-0.71

THE STATE NUM I=116

13	15	0.71
15	13	-0.71

THE STATE NUM I=117

14	15	0.71
15	14	-0.71

THE STATE NUM I=118

13	16	0.71
16	13	-0.71

THE STATE NUM I=119

14	16	0.71
16	14	-0.71

THE STATE NUM I=120

15	16	0.71
16	15	-0.71

Table 5.9

THE C,G,COEF OF THE REPR=10 ARE
****

THE STATE NUM I=1

1	8	0.35
2	7	-0.35
3	6	0.35
4	5	-0.35
5	4	-0.35
6	3	0.35
7	2	-0.35
8	1	0.35

THE STATE NUM I=2

1	12	0.35
2	11	-0.35
3	10	0.35
4	9	-0.35
9	4	-0.35
10	3	0.35
11	2	-0.35
12	1	0.35

THE STATE NUM I=3

1	14	0.35
2	13	-0.35
5	10	0.35
6	9	-0.35
9	6	-0.35
10	5	0.35
13	2	-0.35
14	1	0.35

THE STATE NUM I=4

1	15	0.35
3	13	-0.35
5	11	0.35
7	9	-0.35
9	7	-0.35
11	5	0.35
13	3	-0.35
15	1	0.35

THE STATE NUM I=5

2	15	0.35
3	14	-0.35
5	12	0.35
8	9	-0.35
9	8	-0.35
12	5	0.35
14	3	-0.35
15	2	0.35

THE STATE NUM I=6

1	16	0.35
4	13	-0.35
6	11	0.35
7	10	-0.35
10	7	-0.35
11	6	0.35
13	4	-0.35
16	1	0.35

THE STATE NUM I=7

2	16	0.35
4	14	-0.35
6	12	0.35
8	10	-0.35
10	8	-0.35
12	6	0.35
14	4	-0.35
16	2	0.35

THE STATE NUM I=8

3	16	0.35
4	15	-0.35
7	12	0.35
8	11	-0.35
11	8	-0.35
12	7	0.35
15	4	-0.35
16	3	0.35

THE STATE NUM I=9

5	16	0.35
6	15	-0.35
7	14	0.35
8	13	-0.35
13	8	-0.35
14	7	0.35
15	6	-0.35
16	5	0.35

THE STATE NUM I=10

9	16	0.35
10	15	-0.35
11	14	0.35
12	13	-0.35
13	12	-0.35
14	11	0.35
15	10	-0.35
16	9	0.35

Table 5.10

CHAPTER 6

THE MATHEMATICAL STRUCTURE OF MODELS BASED ON ORTHOGONAL (SO(2n), n = 7,9,11,...) AND EXCEPTIONAL GROUPS (E_6,E_7,E_8)

We shall develop our construction methods a step further, aiming at an understanding of the mathematical structure of possible models based on orthogonal and exceptional groups [67,68]. The main problem in the representation theory of higher rank algebras with large dimension representations is the generation of the weight systems. We successfully solved this problem at least for the physically significant representations, giving simple rules upon which the weight systems can be constructed. We shall also discuss the possibility of constructing an explicit matrix representation and evaluating Clebsch-Gordan coefficients of these groups. The orthogonal groups will be examined in §6.1, and the exceptional in §6.2.

§6.1 Orthogonal Groups

We shall study the following problems:

(a) Is it possible to establish a formula for the Clebsch-Gordan series of the two lowest dimensional spinorial representations for the orthogonal groups SO(2n) with n = 5,7,9,...?

(b) What can we say about the weight systems of higher dimensional representations?

(c) Is it practical to speak about an explicit matrix realization of an irreducible representation with dimension of 10^4 or more, and about Clebsch-Gordan coefficients of such representations?

§6.1.1 Clebsch-Gordan series formulae for the D_ℓ (ℓ = odd) algebras

Table 4.14 of Chapter 4 (§4.3.5) using the notation $D(\{n_1,n_2,\ldots,n_\ell\})$ can be represented as follows:

Table 6.1

Group	Clebsch-Gordan Series
SO(10)	$D(1_5) \otimes D(1_5) = D(1_1) \oplus D(1_3) \oplus D(2_5)$ $D(1_5) \otimes D(1_4) = D(1_0) \oplus D(1_2) \oplus D(1_4,1_5)$
SO(14)	$D(1_7) \otimes D(1_7) = D(1_1) \oplus D(1_3) \oplus D(1_5) \oplus D(2_7)$ $D(1_7) \otimes D(1_6) = D(1_0) \oplus D(1_2) \oplus D(1_4) \oplus D(1_6,1_7)$
SO(18)	$D(1_9) \otimes D(1_9) = D(1_1) \oplus D(1_3) \oplus D(1_5) \oplus D(1_7) \oplus D(2_9)$ $D(1_9) \otimes D(1_8) = D(1_0) \oplus D(1_2) \oplus D(1_4) \oplus D(1_6) \oplus D(1_8,1_9)$

In the above table the symbol $D(1_i)$ means that in the i position of the $D(\{n_1,n_2,\ldots,n_i,\ldots,n_\ell\})$ representation the n_i has the value of 1, while all the other n's have the value of 0.

From Table 6.1, it is easy to deduce the following general formulae:

Formula A

$$D(1_\ell) \otimes D(1_\ell) = \sum_{n=0}^{\frac{1}{2}(\ell-3)} \oplus D(1_{2n+1}) \oplus D(2_\ell) \qquad (1.1.1)$$

Formula B

$$D(1_\ell) \otimes D(1_{\ell-1}) = \sum_{n=0}^{\frac{1}{2}(\ell-3)} \oplus D(1_{2n}) \oplus D(1_{\ell-1},1_\ell) \qquad (1.1.2)$$

The decomposition formulae A and B can give us the Clebsch-Gordan series of any ℓ of the algebra D_ℓ, and Program B1(2) the dimensionality of the representations of the right hand side of (1.1.1) and (1.1.2).

§6.1.2 Weight systems of SO(2n) orthogonal groups

From the observation that all the weights of the $\underset{\sim}{126}$ and $\underset{\sim}{120}$ representations of D_5 entering Formula A are simple, except the ones which belong to the natural ten-dimensional representation, we have found the following results (Table 6.2) for the weights multiplicities of the representations in Formula A.

Table 6.2: Weights multiplicity of the representations entering Formula A

D_9 (= SO(18))		D_7 (= SO(14))		D_5 (= SO(10))	
$D(1_1)$	all simple	$D(1_1)$	all simple	$D(1_1)$	all simple
$D(1_3)$	whts[$D(1_1)$] × 8 other simple	$D(1_3)$	whts[$D(1_1)$] × 6 other simple	$D(1_3)$	whts[$D(1_1)$] × 4 other simple
$D(1_5)$	whts[$D(1_1)$] × 28 whts[$D(1_3)$] × 6 other simple	$D(1_5)$	whts[$D(1_1)$] × 15 whts[$D(1_3)$] × 4 other simple	$D(2_5)$	whts[$D(1_1)$] × 3 other simple
$D(1_7)$	whts[$D(1_1)$] × 56 whts[$D(1_3)$] × 15 whts[$D(1_5)$] × 4 other simple	$D(2_7)$	whts[$D(1_1)$] × 10 whts[$D(1_3)$] × 3 other simple		
$D(2_9)$	whts[$D(1_1)$] × 35 whts[$D(1_3)$] × 10 whts[$D(1_5)$] × 3 other simple				

In the above table the symbol whts$[D(1_i)] \times m$ means that all the weights of the representation $D(1_i)$ have multiplicity m.

We have found the results given in the above table, running Program B4(2) for a limited number of combinations of the numbers q_i entering the level q of a weight (see §4.2.2).

The information we can get from Table 6.2 simplifies considerably the weights computational problem, because we only need the knowledge of the weights without multiplicity. Using integer arithmetic, we have further developed Program B2(2) as to generate the weights without multiplicity of representations up to 10^5 dimensions in a very short time. For example, Program B2(2) needed 9 minutes to generate the weights of the $D(2_9)$ representation which has dimension d = 24310.

In the case of $D(1_i)$ and $D(2_i)$ representations, with $i = 1,2,\ldots,\ell$, we found some very interesting results which are summarized in the following two claims.

Claim I

For the case of $D(1_5)$ representation of the algebra D_7, the linear functional $\lambda = \Lambda - \sum_{i=1}^{7} q_i\alpha_i$ is a weight

(a) with multiplicity $m(\lambda) = 1$ if and only if the left hand side of Freudenthal's formula (formula (2.2.2) of Chapter 4) is equal to the level of this linear functional λ,

(b) with multiplicity $m(\lambda) = 4$ (see Table 6.2) if and only if the left hand side of the above formula is equal to the level of λ plus one,

(c) with multiplicity $m(\lambda) = 15$ if and only if the left hand side of Freudenthal's formula is equal to the level of λ plus two.

Claim II

For the case of $D(2_7)$ representation of D_7, the linear functional $\lambda = \Lambda - \sum_{i=1}^{7} q_i \alpha_i$ is a weight

(a) with multiplicity $m(\lambda) = 1$, if and only if the left hand side of (2.2.2) is equal to the level λ in the case when every q_i of the level $q = \sum_{i=1}^{7} q_i$ is a multiple of 2. In the case when at least one of the q_i is not a multiple of 2, then λ is a weight with $m(\lambda) = 1$ if and only if the left hand side of (2.2.2) is equal to the level of λ plus one,

(b) with multiplicity $m(\lambda) = 3$ if and only if the left hand side of (2.2.2) is equal to the level of λ plus 2,

(c) with multiplicity $m(\lambda) = 10$ if and only if the left hand side of (2.2.2) is equal to the level of λ plus 3.

Note

The above two claims have been stated in the case of D_7, but they can be similarly expressed for any D_ℓ algebras (according to Table 6.2).

We have numerically proved Claims I and II for the D_5 and D_7 algebras, but, considering the similar structure of the D_ℓ algebras, we believe that they will be true of every ℓ.

The Freudenthal's formula can be written as follows

$$[\sum_{j=1}^{\ell} q_i w_j \{(n_j+1) - \frac{1}{2} \sum_{i=1}^{\ell} q_i A_{ji}\}] m(\lambda)$$

$$= \sum_{\alpha\in\Delta+} \sum_k m(\lambda+k\alpha) [\sum_{j=1}^{\ell} k_j^\alpha w_j \{n_j + \sum_{i=1}^{\ell} A_{ji} (kk_i^\alpha - q_i)\}] \qquad (1.2.1)$$

where $\Lambda = \sum_{j=1}^{\ell} n_j \Lambda_j$, $\lambda = \Lambda - \sum_{j=1}^{\ell} q_j \alpha_j$, $\alpha = \sum_{j=1}^{\ell} k_j^\alpha \alpha_j$ and w_j are the weighting factors.

In this form the Freudenthal's formula can be used to verify the above two claims.

Example

Let us consider the $D(1_5)$ representations of D_9 ($d = 8568$). The left hand side of (1.2.1) for this representation becomes

$$\text{LHS} = \sum_{i=1}^{\ell} q_i - \sum_{i=1}^{\ell} q_i^2 + \sum_{i<j}^{8} q_i q_j + q_7 q_8 + q_5. \tag{1.2.2}$$

Using Program B4(2) for the values of q_i's (0,0,0,0,1,2,2,1,1) we found a weight with multiplicity 1. Substituting the above values of q_i's to (1.2.2) we get 7 which is equal to the level $q = \Sigma q_i = 7$ of that weight.

For the values (0,0,0,0,0,1,2,2,1) Program B4(2) tells us that the linear functional $\lambda = (\alpha_1+2\alpha_2+3\alpha_3+4\alpha_4+5\alpha_5+3\alpha_6+3\alpha_7+1.5\alpha_8+1.5\alpha_9)$ is not a weight. On the other hand, from (1.2.2) we find $4 \neq q = \Sigma q_i = 6$.

In the $D(1_5)$ representation of D_9 the non-simple weights have multiplicity 6 and 28. In the case of the weights with multiplicity 6 (1.2.2) always gives the value of level + 1, while for the case of multiplicity 28, it gives the value of level + 2.

Claims I and II simplify further the weights computational problem, because the weight systems can be trivially generated and at the same time the weights storage problem is eliminated.

§6.1.3 Matrix realization and Clebsch-Gordan coefficients

The method of constructing an explicit matrix realization of irreducible representations contained in Formula A is in principle quite the same with the method described in Chapter 4. The main problem is how to store the weights of such a large dimension representation in the computer memory. The problem is simplified if we remember that Program D5(4), for example, uses only the differences of the weights and their multiplicities and only these differences which are equal to one of the simple roots are relevant.

The evaluation of the Clebsch-Gordan coefficients depends upon the knowledge of the matrix elements of the representation. Again the storage problem can be solved because, using the sparse arrays technique, we store only the non-zero elements of the matrix realizing the representation.

§6.2 Exceptional Groups

As for the case of the orthogonal groups, we shall study the Clebsch-Gordan series of the tensor products of the lowest dimensional representations of E_6, E_7 and E_8. The problem of a matrix realization of the representations entering the above Clebsch-Gordan series will be considered, and the evaluation of the Clebsch-Gordan coefficients will be discussed.

§6.2.1 Clebsch-Gordan Series

Using Program C7(3), we found the following values of the $I^{(2)}$ and $I^{(4)}$ indices for the few lowest dimensional representations of

(a) E_6

Table 6.3: Higher indices for the E_6 group

Representation	Dim	$I^{(2)}$	$I^{(4)}$
D(0,0,0,0,0,0)	1	0	0
D(0,0,0,0,0,1)	78	144	288
D(0,0,0,0,0,2)	2430	9720	46656
D(0,0,0,0,1,0)	27	36	48
D(0,0,0,0,1,1)	1728	5760	22656
D(0,0,0,0,2,0)	351	1008	3360
D(0,0,0,1,0,0)	351	900	2640
D(0,0,1,0,0,0)	2925	10800	47520
D(0,1,0,0,0,0)	351	900	2640
D(1,0,0,0,0,0)	27	36	48
D(1,0,0,0,1,0)	650	1800	5760
D(2,0,0,0,0,0)	351	1008	3360

We have the following decompositions (see Chapter 4):

$\underset{\sim}{27} \otimes \underset{\sim}{27}$	$\Sigma N = 729$	$\Sigma I^{(2)} = 1944$	$\Sigma I^{(4)} = 6048$	
N	$\underset{\sim}{351}$ ⊕	$\underset{\sim}{351}'$ ⊕	$\underset{\sim}{27}$	$\Sigma N = 729$
$I^{(2)}$	1008	900	36	$\Sigma I^{(2)} = 1944$
$I^{(4)}$	3360	2640	48	$\Sigma I^{(4)} = 6048$
$\underset{\sim}{27} \otimes \underset{\sim}{27}'$	$\Sigma N = 729$	$\Sigma I^{(2)} = 1944$	$\Sigma I^{(4)} = 6048$	
N	$\underset{\sim}{650}$ ⊕	$\underset{\sim}{78}$ ⊕	$\underset{\sim}{1}$	$\Sigma N = 729$
$I^{(2)}$	1800	144	0	$\Sigma I^{(2)} = 1944$
$I^{(4)}$	5760	288	0	$\Sigma I^{(4)} = 6048$
$\underset{\sim}{78} \otimes \underset{\sim}{27}$	$\Sigma N = 2106$	$\Sigma I^{(2)} = 6696$	$\Sigma I^{(4)} = 25344$	
N	$\underset{\sim}{1728}$ ⊕	$\underset{\sim}{351}'$ ⊕	$\underset{\sim}{27}$	$\Sigma N = 2106$
$I^{(2)}$	5760	900	36	$\Sigma I^{(2)} = 6696$
$I^{(4)}$	22256	2640	48	$\Sigma I^{(4)} = 25344$

(b) E_7

Table 6.4: Higher indices of the group E_7

Representation	Dim	$I^{(2)}$	$I^{(4)}$
D(0,0,0,0,0,0,0)	1	0	0
D(0,0,0,0,0,0,1)	912	2520	7812
D(0,0,0,0,0,0,2)	1413	4620	16632
D(0,0,0,0,0,1,0)	56	84	126
D(0,0,0,0,0,1,1)	40755	180180	936936
D(0,0,0,0,1,0,0)	1539	4536	15120
D(0,0,0,1,0,0,0)	27664	120120	612612
D(0,1,0,0,0,0,0)	8645	32760	144144
D(1,0,0,0,0,0,0)	133	252	504
D(1,0,0,0,0,0,1)	86184	419580	2419578
D(1,0,0,0,0,1,0)	6480	22680	91476
D(2,0,0,0,0,0,0)	7371	29484	137592
D(0,0,0,0,1,1,0)	51072	237888	1308384

For the Clebsch-Gordan series we have

$\underset{\sim}{56} \otimes \underset{\sim}{56}$	$\Sigma N = 3136$	$\Sigma I^{(2)} = 9408$	$\Sigma I^{(4)} = 32256$		
N	$\underset{\sim}{1463}$ $\oplus$	$\underset{\sim}{1539}$ $\oplus$	$\underset{\sim}{133}$ $\oplus$	$\underset{\sim}{1}$	$\Sigma N = 3136$
$I^{(2)}$	4620	4536	252	0	$\Sigma I^{(2)} = 9408$
$I^{(4)}$	16632	15120	504	0	$\Sigma I^{(4)} = 32256$
$\underset{\sim}{133} \otimes \underset{\sim}{56}$	$\Sigma N = 7448$	$\Sigma I^{(2)} = 25284$	$\Sigma I^{(4)} = 99414$		
N	$\underset{\sim}{6480}$ $\oplus$	$\underset{\sim}{912}$ $\oplus$	$\underset{\sim}{56}$		$\Sigma N = 7448$
$I^{(2)}$	22680	2520	84		$\Sigma I^{(2)} = 25284$
$I^{(4)}$	91476	7812	126		$\Sigma I^{(4)} = 99414$

(c) E_8

Table 6.5: Higher indices of the group E_8

Representation	Dim	$I^{(2)}$	$I^{(4)}$
D(0,0,0,0,0,0,0,0)	1	0	0
D(0,0,0,0,0,0,1,0)	248	480	960
D(0,0,0,0,0,0,2,0)	27000	108000	492480
D(0,0,0,0,0,1,0,0)	30380	177600	517440
D(1,0,0,0,0,0,0,0)	3875	12000	41280

We have the following Clebsch-Gordan series:

$\underset{\sim}{248} \otimes \underset{\sim}{248}$	$\Sigma N = 61504$ $\Sigma I^{(2)} = 238080$ $\Sigma I^{(4)} = 1052160$					
N	$\underset{\sim}{27000}$ ⊕	$\underset{\sim}{30380}$ ⊕	$\underset{\sim}{3875}$ ⊕	$\underset{\sim}{248}$ ⊕	$\underset{\sim}{1}$	$\Sigma N = 61504$
$I^{(2)}$	108000	117600	12000	480	0	$\Sigma I^{(2)} = 238080$
$I^{(4)}$	492480	517440	41280	960	0	$\Sigma I^{(4)} = 1052160$

To summarize, the tensor products of the lowest dimensional representations of the groups E_6, E_7, E_8 have the following Clebsch-Gordan series:

Table 6.6: Clebsch-Gordan series of the groups E_6, E_7 and E_8

Group	Clebsch-Gordan series
E_6	$D(1_5) \otimes D(1_5) = D(2_5) \oplus D(1_4) \oplus D(1_1)$
	$D(1_5) \otimes D(1_1) = D(1_1,1_5) \oplus D(1_6) \oplus D(1_0)$
	$D(1_5) \otimes D(1_6) = D(1_4) \oplus D(1_5,1_6) \oplus D(1_1)$
E_7	$D(1_6) \otimes D(1_6) = D(2_6) \oplus D(1_5) \oplus D(1_1) \oplus D(1_0)$
	$D(1_1) \otimes D(1_6) = D(1_1,1_6) \oplus D(1_7) \oplus D(1_6)$
E_8	$D(1_7) \otimes D(1_7) = D(2_7) \oplus D(1_6) \oplus D(1_1) \oplus D(1_7) \oplus D(1_0)$

§6.2.2 Matrix realization and Clebsch-Gordan coefficients

Employing the same techniques as in §6.1.2, we found the following weights multiplicities:

Table 6.7: Weights multiplicity of E_6 and E_7

	E_7		E_6
$D(1_0)$	all simple	$D(1_5)$	all simple
$D(1_1)$	whts[$D(1_0)$] × 7 other simple	$D(1_4)$	whts[$D(1_5)$] × 5 other simple
$D(1_5)$	whts[$D(1_1)$] × 6 whts[$D(1_0)$] × 27 other simple	$D(2_5)$	whts[$D(1_5)$] × 4 other simple
$D(2_6)$	whts[$D(1_1)$] × 5 whts[$D(1_0)$] × 21 other simple		⋮

Claims I and II of the orthogonal groups are also valid for the exceptional groups. A matrix realization and the Clebsch-Gordan coefficients of the E_6 theory can be easily constructed following the methods of Chapter 4, while for the E_7 and E_8 theories the remarks of §6.1.3 are applied.

CHAPTER 7

MASS RELATIONS IN THE SYMMETRY LIMIT OF THE SO(10) MODEL

Having studied the group theoretical structure of the SO(10) theory (Chapter 5), we are now in a position to analyse in detail the Yukawa interaction term of the Lagrangian of the SO(10) theory. This analysis, which will be carried out in this chapter, has the advantage of giving us a better understanding of the mass relations in the SO(10) model (Chapter 3). On the other hand, it is of a general nature, so that it can be used to analyse Yukawa couplings of any other grand unification scheme, provided of course that the methods of Chapter 4 are used for the generation of matrix elements and the evaluation of Clebsch-Gordan coefficients.

The general group theoretical structure of the Yukawa interaction term will be formulated in §7.1, while the mass relations will be derived in §7.2.

§7.1 Yukawa Couplings

§7.1.1 General formulation

Consider an interaction Lagrangian of the Yukawa type

$$\mathcal{L}_{int} = -g\psi^{+}\psi\phi, \tag{1.1.1}$$

where ψ represents the fermion fields and ϕ the spinless meson fields. In a gauge field theory with spontaneously broken symmetries the spinless meson can exist in states ϕ_i, where ϕ_i are the basis functions of an irreducible representation of the gauge group G. The fermions can exist in states ψ_i, where ψ_i are the basis functions of an irreducible representation of G. Then (1.1.1) can be written

$$\mathcal{L}_{int} = -g_{ijk}\psi_i^*\psi_j\phi_k. \qquad (1.1.2)$$

$\mathcal{L}_{int}$ should be an invariant of G, ie it should transform as the irreducible representation $\underset{\sim}{1}$.

Suppose that the ϕ_k transform as the irreducible representation $\underset{\sim}{P}$. Then we must have $\theta_k = g_{ijk}\psi_i^*\psi_j$ transforming as $\overline{\underset{\sim}{P}}$ in order that $\mathcal{L}_{int}$ be an invariant.

Now, suppose that ψ_j transforms as the irreducible representation $\underset{\sim}{Q}$ so that ψ_i^* transforms as $\overline{\underset{\sim}{Q}}$. The functions $\psi_i^*\psi_j$ provide the basis functions for $\overline{\underset{\sim}{Q}} \otimes \underset{\sim}{Q}$ which is in general reducible as we saw in Chapter 4 (§4.3 and Theorem 4.2.6). Thus, if the Clebsch-Gordan series for $\overline{\underset{\sim}{Q}} \otimes \underset{\sim}{Q}$ contains the irreducible representation T then

$$\sum_{i,j}\left(\begin{matrix}\overline{Q} & Q \\ i & j\end{matrix}\middle|\begin{matrix}T \\ k\end{matrix}\right)\psi_i^*\psi_j \qquad (1.1.3)$$

transforms according to the kth row of $\underset{\sim}{T}$, where

$$\left(\begin{matrix}\overline{Q} & Q \\ i & j\end{matrix}\middle|\begin{matrix}T \\ k\end{matrix}\right)$$

is a Clebsch-Gordan coefficient.

Hence, $g_{ijk}\psi_i^*\psi_j$ transforms as $\overline{P}$ if [62]

(a) $$g_{ijk} = \left(\begin{matrix}\overline{Q} & Q \\ i & j\end{matrix}\middle|\begin{matrix}\overline{P} \\ k\end{matrix}\right)g \text{ (g is a constant [62])} \qquad (1.1.4)$$

if $\overline{\underset{\sim}{P}}$ appears only once in $\overline{\underset{\sim}{Q}} \otimes \underset{\sim}{Q}$, or

(b) $$g_{ijk} = g_1\left(\begin{matrix}\overline{Q} & Q \\ i & j\end{matrix}\middle|\begin{matrix}P_1 \\ k\end{matrix}\right) + g_2\left(\begin{matrix}\overline{Q} & Q \\ i & j\end{matrix}\middle|\begin{matrix}P_2 \\ k\end{matrix}\right) + \ldots + g_n\left(\begin{matrix}\overline{Q} & Q \\ i & j\end{matrix}\middle|\begin{matrix}\overline{P}_n \\ k\end{matrix}\right) \qquad (1.1.5)$$

if $\overline{\underset{\sim}{P}}$ appears n times in $\overline{\underset{\sim}{Q}} \otimes \underset{\sim}{Q}$.

§7.1.2 Yukawa interaction term in the SO(10) model

Each generation of the fermion fields in the SO(10) theory is described by two 16-dimensional spinorial representations. If the left-handed fields

$$\psi_L = (U^{R,W,B}, D^{R,W,B};\ L,N;\ \overline{U}^{R,W,B}, \overline{D}^{R,W,B};\ \overline{L},\overline{N}) \qquad (1.2.1)$$

where $Q(U) = 2/3$, $Q(D) = -1/3$, $Q(L) = -1$, $Q(N) = 0$, transform according to $D(1_5)$ of SO(10), then the right-handed fields

$$\psi_R = \psi_L^C \tag{1.2.2}$$

transform according to $D(1_4)$. But the two spinorial representations $D(1_5)$ and $D(1_4)$ of D_5 are connected by the following relation:

$$\overline{D(1_4)} = D(1_5). \tag{1.2.3}$$

Thus, the Yukawa term (1.1.1) using the relation

$$\overline{\psi}_R \otimes \psi_L = D(1_5) \otimes D(1_5) \equiv \underset{\sim}{16} \otimes \underset{\sim}{16} = \underset{\sim}{126} \oplus \underset{\sim}{120} \oplus \underset{\sim}{10}$$

takes the form

$$\begin{aligned}\mathcal{L}_{int} &= -g_{ijk}\psi_i^*\psi_j\phi_k \\ &= \sum_k^{126}\sum_{i,j}^{16}\begin{pmatrix}\overline{126} & 126 \\ \pi & -\pi\end{pmatrix}\Big|\begin{matrix}1\\0\end{matrix}\Big)\begin{pmatrix}16 & 16 \\ i & j\end{pmatrix}\Big|\begin{matrix}126\\k\end{matrix}\Big)\psi_i\psi_j\psi^{126}_{k,\pi} \\ &+ \sum_\ell^{120}\sum_{i,j}^{16}\begin{pmatrix}120 & 120 \\ \lambda & -\lambda\end{pmatrix}\Big|\begin{matrix}1\\0\end{matrix}\Big)\begin{pmatrix}16 & 16 \\ i & j\end{pmatrix}\Big|\begin{matrix}120\\\ell\end{matrix}\Big)\psi_i\psi_j\phi^{120}_{\ell,\lambda} \\ &+ \sum_m^{10}\sum_{i,j}^{16}\begin{pmatrix}10 & 10 \\ \omega & -\omega\end{pmatrix}\Big|\begin{matrix}1\\0\end{matrix}\Big)\begin{pmatrix}16 & 16 \\ i & j\end{pmatrix}\Big|\begin{matrix}10\\m\end{matrix}\Big)\psi_i\psi_j\phi^{10}_{m,\omega}\end{aligned} \tag{1.2.4}$$

where π,λ,ω are the weight systems of $\underset{\sim}{126}$, $\underset{\sim}{120}$ and $\underset{\sim}{10}$ representations respectively (compare with (2.1.2) of Chapter 5). Note that because $\underset{\sim}{10}$ and $\underset{\sim}{120}$ are real representations we have $\overline{\underset{\sim}{10}} = \underset{\sim}{10}$ and $\overline{\underset{\sim}{120}} = \underset{\sim}{120}$. The coefficients

$$\begin{pmatrix}\overline{126} & 126 \\ \pi & -\pi\end{pmatrix}\Big|\begin{matrix}1\\0\end{matrix}\Big),\ \begin{pmatrix}120 & 120 \\ \lambda & -\lambda\end{pmatrix}\Big|\begin{matrix}1\\0\end{matrix}\Big),\ \begin{pmatrix}10 & 10 \\ \omega & -\omega\end{pmatrix}\Big|\begin{matrix}1\\0\end{matrix}\Big) \tag{1.2.5}$$

express the fact that we are interested only for those linear combinations of meson fields that transform as singlets under SO(10).

In the above expression for the Yukawa interaction term (1.2.4), the coefficients $\begin{pmatrix}16 & 16 \\ i & j\end{pmatrix}\Big|\, a\Big)$, where $a = 126$, 120 and 10, are the Clebsch-Gordan coefficients of Tables 5.8 – 5.10 of Chapter 5. To evaluate the coefficients (1.2.5) we have to apply the operators $E^a_{\alpha_i}$, $i = 1,2,\ldots,5$ to the product of basis functions $\phi^{\bar{a}}_i\phi^a_j$, where a = 126,120,10. The action of these operators to the meson states is known and it is given in Tables 5.5 - 5.7 .

The calculation of the above coefficients is simplified if we observe that only colour singlets and neutral spinless meson fields are allowed to develop vacuum expectation values.

Thus, we are faced with the problem of defining the charge operator and finding the colour quantum numbers of the meson fields. Because the spinless meson fields are coupled in an invariant way with the fermion fields through the Clebsch-Gordan coefficients, it is sufficient to work with the 16-dimensional fermion representation.

One of the maximal subgroups of SO(10) is $SU(4) \otimes SU(2)_L \otimes SU(2)_R$. Knowing the generators of SO(10) we shall find the generators of the subgroups SU(4), $SU(2)_L$, $SU(2)_R$. The knowledge of these generators will allow us to define the charge operator and to find the transformation properties of the fermion fields under the various colour and flavour operators.

§7.2 Mass Relations in the SO(10) Model

§7.2.1 The reduction problem

(a) D_5

The Lie algebra of SO(10) is the real Lie algebra of all 10 × 10 real anti-symmetric matrices. It is convenient to introduce 10 × 10 real anti-symmetric matrices $\underset{\sim}{M}_{pq}$ defined by

$$(\underset{\sim}{M}_{pq})_{jk} = \delta_{pj}\delta_{pk} - \delta_{pk}\delta_{qj}, \; j,k,p,q = 1,2,\ldots,10 \qquad (2.1.1)$$

We may take the matrices

$$\underset{\sim}{M}_{1,2}, \underset{\sim}{M}_{3,4}, \underset{\sim}{M}_{5,6}, \underset{\sim}{M}_{7,8}, \underset{\sim}{M}_{9,10}$$

as the basis of the Cartan subalgebra. The canonical basis for the Cartan subalgebra is connected to the above matrices by the following relations:

$$\left.\begin{aligned} h_{\alpha_1} &= -i/16(\underset{\sim}{M}_{1,2}-\underset{\sim}{M}_{3,4}) \\ h_{\alpha_2} &= -i/16(\underset{\sim}{M}_{3,4}-\underset{\sim}{M}_{5,6}) \\ h_{\alpha_3} &= -i/16(\underset{\sim}{M}_{5,6}-\underset{\sim}{M}_{7,8}) \\ h_{\alpha_4} &= -i/16(\underset{\sim}{M}_{7,8}-\underset{\sim}{M}_{9,10}) \\ h_{\alpha_5} &= -i/16(\underset{\sim}{M}_{7,8}+\underset{\sim}{M}_{9,10}) \end{aligned}\right\} \qquad (2.1.2)$$

(b) D_3

The corresponding relations to (2.1.2) are

$$\left.\begin{aligned} h_{\alpha_1} &= -i/8(\underset{\sim}{M}_{1,2}-\underset{\sim}{M}_{3,4}) \\ h_{\alpha_2} &= -i/8(\underset{\sim}{M}_{3,4}-\underset{\sim}{M}_{5,6}) \\ h_{\alpha_3} &= -i/8(\underset{\sim}{M}_{3,4}+\underset{\sim}{M}_{5,6}) \end{aligned}\right\} \qquad (2.1.3)$$

(c) D_2

For D_2 we have

$$h_{\alpha_1} = -i/4(\underset{\sim}{M}_{1,2}-\underset{\sim}{M}_{3,4})$$

$$h_{\alpha_2} = -i/4(\underset{\sim}{M}_{1,2}+\underset{\sim}{M}_{3,4})$$

The $\underset{\sim}{M}_{pq}$ matrices of D_5 can be blocked as follows.

$$\underset{\sim}{M}_{pq} = \left[\begin{array}{c|c} SO(6) & \\ \hline & SO(4) \end{array}\right]$$

Then, the generators of SO(6) and SO(4) can be identified.

From the relations (2.1.1), (2.1.2) and (2.1.3) we have

$\underline{D}_3$

$$\left.\begin{aligned} h_{\alpha_1}^{D_5} &= \tfrac{1}{2} h_{\alpha_1}^{D_3} \\ h_{\alpha_2}^{D_5} &= \tfrac{1}{2} h_{\alpha_2}^{D_3} \\ h_{\alpha_2}^{D_5} + 2h_{\alpha_3}^{D_5} + h_{\alpha_4}^{D_5} + h_{\alpha_5}^{D_5} &= \tfrac{1}{2} h_{\alpha_3}^{D_3} \end{aligned}\right\} \qquad (2.1.4)$$

A change to the H_α basis, using the relation $H_\alpha = \{2/\langle\alpha,\alpha\rangle\}h_\alpha$ (§4.1.3 of Chapter 4) and the appropriate values for the quantities $\langle\alpha,\alpha\rangle$ (see Appendix B, §B.4) gives

$$\left.\begin{aligned} h_{\alpha_\ell}^{D_5} &= H_{\alpha_\ell}^{D_5}/16 \\ h_{\alpha_\ell}^{D_3} &= H_{\alpha_\ell}^{D_3}/8 \end{aligned}\right\}. \qquad (2.1.5)$$

The relations (2.1.4) become

$$\left.\begin{aligned} H_{\alpha_1}^{D_3} &= H_{\alpha_1}^{D_5} \\ H_{\alpha_2}^{D_3} &= H_{\alpha_2}^{D_5} \\ H_{\alpha_3}^{D_3} &= H_{\alpha_2}^{D_5} + 2H_{\alpha_3}^{D_5} + H_{\alpha_4}^{D_5} + H_{\alpha_5}^{D_5} \end{aligned}\right\}. \qquad (2.1.6)$$

Using the isomorphism $SO(6) \simeq SU(4)$, we then can speak about the SU(4) generators instead of SO(6).

$\underline{D}_2$

$$\left.\begin{aligned} h_{\alpha_1}^{D_2} &= 4h_{\alpha_4}^{D_5} \\ h_{\alpha_2}^{D_2} &= 4h_{\alpha_5}^{D_5} \end{aligned}\right\} \qquad (2.1.7)$$

Because SO(4) ≈ SO(3) ⊗ SO(3) ≈ SU(2) ⊗ SU(2) we can speak about A_1-generators, and again using the relation $H_\alpha = \{2/\langle\alpha,\alpha\rangle\}h_\alpha$ with the corresponding values of the quantities $\langle\alpha,\alpha\rangle$ from Appendix B, §B.1, we have

$$\left.\begin{aligned} H_{\alpha_1}^{A_1} &= H_{\alpha_5}^{D_5} \\ H_{\alpha_1}^{A_1'} &= H_{\alpha_4}^{D_5} \end{aligned}\right\} \tag{2.1.8}$$

and we can identify

$$\left.\begin{aligned} I_3^L &\equiv H_{\alpha_1}^{A_1} = H_{\alpha_5}^{D_5} \\ I_3^R &\equiv H_{\alpha_2}^{A_1'} = H_{\alpha_4}^{D_5} \end{aligned}\right\} \tag{2.1.9}$$

Now, if we become more precise, and work with the first generation of fermion fields, (1.2.1) becomes

$$\psi_L = (u^i, d^i, e^-, \nu_e; \bar{u}^i, \bar{d}^i, e^+, \bar{\nu}_e) \tag{2.1.10}$$

with i = 1,2,3 corresponding to the three colours. From the diagonal generators of the 16-dimensional representation (Table 5.3) and for the $(\not{u}, \nu_e)$ system the SU(4) generators of relations (2.1.6) become

$$H_{\alpha_1}^{A_3} = \begin{bmatrix} 0 & & & \underset{\sim}{0} \\ & 1/2 & & \\ & & -1/2 & \\ \underset{\sim}{0} & & & 0 \end{bmatrix}, \quad H_{\alpha_2}^{A_3} = \begin{bmatrix} 0 & & & \underset{\sim}{0} \\ & 0 & & \\ & & 1/2 & \\ \underset{\sim}{0} & & & -1/2 \end{bmatrix}, \quad H_{\alpha_3}^{A_3} = \begin{bmatrix} 1/2 & & & \underset{\sim}{0} \\ & -1/2 & & \\ & & 0 & \\ \underset{\sim}{0} & & & 0 \end{bmatrix} \tag{2.1.11}$$

To pick up the SU(3) generators we block the A_3-generators according to

$$H_{\alpha_\ell}^{A_3} = \left[\begin{array}{c|c} 0 & * \\ \hline & SU(3) \end{array}\right]$$

Then, the SU(3) generators are

$$H_{\alpha_1}^{A_2} = \begin{bmatrix} 1/2 & & \underset{\sim}{0} \\ & -1/2 & \\ \underset{\sim}{0} & & 0 \end{bmatrix}, \quad H_{\alpha_2}^{A_2} = \begin{bmatrix} 0 & & \underset{\sim}{0} \\ & 1/2 & \\ \underset{\sim}{0} & & -1/2 \end{bmatrix}$$

and in terms of the D_5 generators we have

$$\left.\begin{aligned} H_{\alpha_1}^{A_2} &= H_{\alpha_1}^{D_5} \\ H_{\alpha_2}^{A_2} &= H_{\alpha_2}^{D_5} \end{aligned}\right\} \tag{2.1.12}$$

We define a 'physical' SU(3) basis as follows:

$$\left.\begin{aligned} I_3^{SU(3)} &= \begin{bmatrix} 1/2 & & \underset{\sim}{0} \\ & -1/2 & \\ \underset{\sim}{0} & & 0 \end{bmatrix} \equiv H_{\alpha_1}^{A_2'} = H_{\alpha_1}^{D_5} \\ Y^{SU(3)} &= \begin{bmatrix} 1/3 & & \underset{\sim}{0} \\ & 1/3 & \\ \underset{\sim}{0} & & -2/3 \end{bmatrix} = H_{\alpha_2}^{A_2'} = \frac{2}{3}(H_{\alpha_1}^{A_2}+2H_{\alpha_2}^{A_2}) \\ &\equiv \frac{2}{3}(H_{\alpha_1}^{D_5}+2H_{\alpha_2}^{D_5}) \end{aligned}\right\} \tag{2.1.13}$$

It would be helpful to define also a 'physical' basis for the SU(4) as follows:

$$\left.\begin{aligned}
I_3^{SU(4)} &= \begin{bmatrix} 0 & & & \underset{\sim}{0} \\ & 1/2 & & \\ & & -1/2 & \\ \underset{\sim}{0} & & & 0 \end{bmatrix} = H_{\alpha_1}^{D_5} \\
Y^{SU(4)} &= \begin{bmatrix} 0 & & & \underset{\sim}{0} \\ & 1/3 & & \\ & & 1/3 & \\ \underset{\sim}{0} & & & -2/3 \end{bmatrix} = \frac{2}{3}(H_{\alpha_1}^{D_5}+2H_{\alpha_2}^{D_5}) \\
X^{SU(4)} &= \begin{bmatrix} -3/2 & & & \underset{\sim}{0} \\ & 1/2 & & \\ & & 1/2 & \\ \underset{\sim}{0} & & & 1/2 \end{bmatrix} = -(2H_{\alpha_1}^{A_3}+H_{\alpha_2}^{A_3}+3H_{\alpha_3}^{A_3}) \\
&= -(2H_{\alpha_1}^{D_5}+4H_{\alpha_2}^{D_5}+6H_{\alpha_3}^{D_5}+3H_{\alpha_4}^{D_5}+3H_{\alpha_5}^{D_5})
\end{aligned}\right\} \qquad (2.1.14)$$

Finally, we define the charge operator by

$$Q = I_{3L} + I_{3R} + \frac{1}{3}X. \qquad (2.1.15)$$

We summarize the results from the above reduction procedure.

<u>I SU(4)</u>

$$X^{SU(4)} = -(2H_{\alpha_1}^{D_5}+4H_{\alpha_2}^{D_5}+6H_{\alpha_3}^{D_5}+3H_{\alpha_4}^{D_5}+3H_{\alpha_5}^{D_5})$$

<u>II SU(3)</u>

$$I_3^{SU(3)} = H_{\alpha_1}^{D_5}$$

$$Y^{SU(3)} = \frac{2}{3}(H_{\alpha_1}^{D_5}+2H_{\alpha_2}^{D_5})$$

<u>III $SU(2)_L$</u>

$$I_{3L}^{SU(2)} = H_{\alpha_5}^{D_5}$$

<u>IV $SU(2)_R$</u>

$$I_{3R}^{SU(2)} = H_{\alpha_4}^{D_5}$$

V Charge

$$Q = I_{3L} + I_{3R} + \frac{1}{3}X.$$

Now we are ready to assign the fermion fields of the first generation to the 16-dimensional representations.

§7.2.2 Assignment

(a) $D(1_5)$ representation

Table 7.1

N	H_{α_1}	H_{α_2}	H_{α_3}	H_{α_4}	H_{α_5}	I_{3L}	I_{3R}	$I_3^{SU(3)}$	$Y^{SU(3)}$	$X^{SU(4)}$	Q	Ass
1	0	0	0	0	1/2	1/2	0	0	0	-3/2	0	v_e
2	0	0	1/2	0	-1/2	-1/2	0	0	0	-3/2	-1	e^-
3	0	1/2	-1/2	1/2	0	0	1/2	0	2/3	-1/2	1/3	$\bar{d}_3$
4	0	1/2	0	-1/2	0	0	-1/2	0	2/3	-1/2	-2/3	$\bar{u}_3$
5	1/2	-1/2	0	1/2	0	0	1/2	1/2	-1/3	-1/2	1/3	$\bar{d}_2$
6	1/2	-1/2	1/2	-1/2	0	0	-1/2	1/2	-1/3	-1/2	-2/3	$\bar{u}_2$
7	1/2	0	-1/2	0	1/2	1/2	0	1/2	1/3	1/2	2/3	u_1
8	1/2	0	0	0	-1/2	-1/2	0	1/2	1/3	1/2	-1/3	d_1
9	-1/2	0	0	1/2	0	0	1/2	-1/2	-1/3	-1/2	1/3	$\bar{d}_1$
10	-1/2	0	1/2	-1/2	0	0	-1/2	-1/2	-1/3	-1/2	-2/3	$\bar{u}_1$
11	-1/2	1/2	-1/2	0	1/2	1/2	0	-1/2	1/3	1/2	2/3	u_2
12	-1/2	1/2	0	0	-1/2	-1/2	0	-1/2	1/3	1/2	-1/3	d_2
13	0	-1/2	0	0	1/2	1/2	0	0	-2/3	1/2	2/3	u_3
14	0	-1/2	1/2	0	-1/2	-1/2	0	0	-2/3	1/2	-1/3	d_3
15	0	0	-1/2	1/2	0	0	1/2	0	0	3/2	1	e^+
16	0	0	0	-1/2	0	0	-1/2	0	0	3/2	0	$\bar{v}_e$

(b) $D(1_4)$ representation

The $D(1_4)$ diagonal generators are the same with the diagonal generators of $D(1_5)$, the only exception being that H_{α_4} and H_{α_5} of $D(1_4)$ are equal to

$$H_{\alpha_4}^{D(1_4)} = H_{\alpha_5}^{D(1_5)}, \quad H_{\alpha_5}^{D(1_4)} = H_{\alpha_4}^{D(1_5)}.$$

Consequently, the right-handed fermions have the same quantum numbers with the left-handed fields except for an interchange of $I_{3L} \leftrightarrow I_{3R}$.

The meson fields which have zero charge and zero eigenvalues of the operators $I_3^{SU(3)}, Y^{SU(3)}$ in the $\underset{\sim}{10}$, $\underset{\sim}{120}$ and $\underset{\sim}{126}$ dimensional representations are given in Tables 7.2 - 7.4.

Table 7.2: Mesons in the $\underset{\sim}{10}$ representation

N	H_{α_1}	H_{α_2}	H_{α_3}	H_{α_4}	H_{α_5}	I_{3L}	I_{3R}	$I_3^{SU(3)}$	$Y^{SU(3)}$	$X^{SU(4)}$	Q	Charge neutral
1	1/2	0	0	0	0	0	0	1/2	1/3	-1	-1/3	
2	-1/2	1/2	0	0	0	0	0	-1/2	1/3	-1	-1/3	
3	0	-1/2	1/2	0	0	0	0	0	-2/3	-1	-1/3	
4	0	0	-1/2	1/2	1/2	1/2	1/2	0	0	0	1	
5	0	0	0	1/2	-1/2	-1/2	1/2	0	0	0	0	ϕ_5
6	0	0	0	-1/2	1/2	1/2	-1/2	0	0	0	0	ϕ_6
7	0	0	1/2	-1/2	-1/2	-1/2	-1/2	0	0	0	-1	
8	0	1/2	-1/2	0	0	0	0	0	2/3	1	1/3	
9	1/2	-1/2	0	0	0	0	0	-1/2	-1/3	1	1/3	
10	-1/2	0	0	0	0	0	0	1/2	-1/3	1	1/3	

Table 7.3: Mesons in the 120 representation

N	H_{α_1}	H_{α_2}	H_{α_3}	H_{α_4}	H_{α_5}	I_{3L}	I_{3R}	$I_3^{SU(3)}$	$Y^{SU(3)}$	$X^{SU(4)}$	Q	Charge neutral
53	0	0	−1/2	1/2	1/2	1/2	1/2	0	0	0	1	
54	0	0	−1/2	1/2	1/2	1/2	1/2	0	0	0	1	
55	0	0	−1/2	1/2	1/2	1/2	1/2	0	0	0	1	
56	0	0	−1/2	1/2	1/2	1/2	1/2	0	0	0	1	
57	0	0	0	1/2	−1/2	−1/2	1/2	0	0	0	0	ϕ_{57}
58	0	0	0	1/2	−1/2	−1/2	1/2	0	0	0	0	ϕ_{58}
59	0	0	0	1/2	−1/2	−1/2	1/2	0	0	0	0	ϕ_{59}
60	0	0	0	1/2	−1/2	−1/2	1/2	0	0	0	0	ϕ_{60}
61	0	0	0	−1/2	1/2	1/2	−1/2	0	0	0	0	ϕ_{61}
62	0	0	0	−1/2	1/2	1/2	−1/2	0	0	0	0	ϕ_{62}
63	0	0	0	−1/2	1/2	1/2	−1/2	0	0	0	0	ϕ_{63}
64	0	0	0	−1/2	1/2	1/2	−1/2	0	0	0	0	ϕ_{64}
65	0	0	1/2	−1/2	−1/2	−1/2	−1/2	0	0	0	−1	
66	0	0	1/2	−1/2	−1/2	−1/2	−1/2	0	0	0	−1	
67	0	0	1/2	−1/2	−1/2	−1/2	−1/2	0	0	0	−1	
68	0	0	1/2	−1/2	−1/2	−1/2	−1/2	0	0	0	−1	

Table 7.4: Mesons in the 126 representation

N	H_{α_1}	H_{α_2}	H_{α_3}	H_{α_4}	H_{α_5}	I_{3L}	I_{3R}	$I_3^{SU(3)}$	$Y^{SU(3)}$	$X^{SU(4)}$	Q	Charge neutral
58	0	0	−1/2	1/2	1/2	1/2	1/2	0	0	0	1	
59	0	0	−1/2	1/2	1/2	1/2	1/2	0	0	0	1	
60	0	0	−1/2	1/2	1/2	1/2	1/2	0	0	0	1	
61	0	0	0	1/2	−1/2	−1/2	1/2	0	0	0	0	ϕ_{61}
62	0	0	0	1/2	−1/2	−1/2	1/2	0	0	0	0	ϕ_{62}
63	0	0	0	1/2	−1/2	−1/2	1/2	0	0	0	0	ϕ_{63}
64	0	0	0	−1/2	1/2	1/2	−1/2	0	0	0	0	ϕ_{64}
65	0	0	0	−1/2	1/2	1/2	−1/2	0	0	0	0	ϕ_{65}
66	0	0	0	−1/2	1/2	1/2	−1/2	0	0	0	0	ϕ_{66}
67	0	0	1/2	−1/2	−1/2	−1/2	−1/2	0	0	0	−1	
68	0	0	1/2	−1/2	−1/2	−1/2	−1/2	0	0	0	−1	
69	0	0	1/2	−1/2	−1/2	−1/2	−1/2	0	0	0	−1	

§7.2.3 Calculation of the coefficients in the Yukawa term

Here we shall give the full calculations of the coefficients $\begin{pmatrix} 10 & 10 & | & 1 \\ \omega & -\omega & | & 0 \end{pmatrix}$ and we shall indicate the calculation procedure of the 120 and 126 representations.

(a) $\underset{\sim}{10}$ representation

Let us consider a linear combination of those states in the 10-dimensional representation which have opposite weights.

$$\sum_{i=1}^{10} a_i \phi_i \phi_{11-i} = a_1\phi_1\phi_{10} + a_2\phi_2\phi_9 + a_3\phi_3\phi_8 + a_4\phi_4\phi_7 + a_5\phi_5\phi_6$$

$$+ a_6\phi_6\phi_5 + a_7\phi_7\phi_4 + a_8\phi_8\phi_3 + a_9\phi_9\phi_2 + a_{10}\phi_{10}\phi_1 \qquad (2.3.1)$$

The action of the operators $E_{-\alpha_i}$, $i = 1,2,\ldots,5$ (Table 5.4) on the linear combination is

(i) $E_{-\alpha_1}$

$$a_1\phi_2\phi_{10} + a_2\phi_2\phi_{10} + a_9\phi_{10}\phi_2 + a_{10}\phi_{10}\phi_2 = 0$$

$$\Rightarrow a_1 + a_2 = a_9 + a_{10} = 0$$

(ii) $E_{-\alpha_2}$

$$a_2\phi_3\phi_9 + a_3\phi_3\phi_9 + a_8\phi_9\phi_3 + a_9\phi_9\phi_3 = 0$$

$$\Rightarrow a_2 + a_3 = a_8 + a_9 = 0$$

(iii) $E_{-\alpha_3}$

$$a_3\phi_4\phi_8 + a_4\phi_4\phi_8 + a_7\phi_8\phi_4 + a_8\phi_8\phi_4 = 0$$

$$\Rightarrow a_3 + a_4 = a_7 + a_8 = 0$$

(iv) $E_{-\alpha_4}$

$$a_4\phi_6\phi_7 + a_7\phi_7\phi_6 + a_6\phi_6\phi_7 + a_7\phi_7\phi_6 = 0$$

$$\Rightarrow a_4 + a_6 = a_5 + a_7 = 0$$

(v) $E_{-\alpha_5}$

$$a_4\phi_5\phi_7 + a_5\phi_5\phi_7 + a_6\phi_7\phi_5 + a_7\phi_7\phi_5 = 0$$

$$\Rightarrow a_4 + a_5 = a_6 + a_7 = 0.$$

From Table 7.2, we are interested only in the states ϕ_5, ϕ_6. From the above calculations we have

$$a_5 = a_6. \qquad (2.3.2)$$

On the other hand, for reasons which will become obvious later, we are not interested in the values of the a's (which values can be fixed from the normalization of the states in (2.3.1)) but only in the relative signs between them.

(b) 120 representation

For this representation and for charge neutral meson fields we find the following signs:

Table 7.5

Coefficients	a_{57}	a_{58}	a_{59}	a_{60}	a_{61}	a_{62}	a_{63}	a_{64}
Signs	−	−	−	−	+	+	+	+

(c) 126 representation

For this representation we have a slight complication, because the coefficients $\left(\begin{array}{cc|c}\overline{126} & 126 & 1\\ \pi & -\pi & 0\end{array}\right)$ involve the $\overline{126}$ representation. The $\overline{126}$ differs from the 126 in the fourth and fifth components of their weights systems. If $\lambda = \alpha_1\lambda_1 + \alpha_2\lambda_2 + \alpha_3\lambda_3 + \alpha_4\lambda_4 + \alpha_5\lambda_5$ is a weight of the 126 representation then $\lambda' = \alpha_1\lambda_1 + \alpha_2\lambda_2 + \alpha_3\lambda_3 + \alpha_4\lambda_5 + \alpha_5\lambda_4$ is the corresponding weight of the $\overline{126}$ representation. With this observation it is easy to construct the weight diagram of the $\overline{126}$ representation, and using the methods of Chapter 4, its matrix representation. For the charge neutral meson fields we find the following signs.

Table 7.6

Coefficients	a_{61}	a_{62}	a_{63}	a_{64}	a_{65}	a_{66}
Signs	−	−	−	+	+	+

§7.2.4 Specification of the colour singlets meson states

A meson state ϕ would be a colour singlet state if the following conditions are satisfied.

$$\left.\begin{aligned} H^c_{\alpha_1}\phi &= 0 \\ H^c_{\alpha_2}\phi &= 0 \\ E^c_{\alpha_1}\phi &= 0 \\ E^c_{\alpha_2}\phi &= 0 \end{aligned}\right\} \qquad (2.4.1)$$

where c indicates the colour. In §2.1 we have identified the two colour diagonal generator as (relations (2.1.13))

$$H^c_{\alpha_1} = I_3^{SU(3)} = H^{D_5}_{\alpha_1}$$

$$H^c_{\alpha_2} = Y^{SU(3)} = \frac{2}{3}(H^{D_5}_{\alpha_1}+2H^{D_5}_{\alpha_2}).$$

To identify the $E^c_{\alpha_1}$ and $E^c_{\alpha_2}$ generators of SU(3), we must follow the reduction procedure $D_5 \to D_3 \equiv A_3 \to A_2$, but this time for the case of the non-diagonal generators E_{α_i}, $i = 1,2,\ldots,5$.

For the D_ℓ algebras the following relations hold between the canonical generators e_{α_i}, $i = 1,2,\ldots,5$, and the 10×10 antisymmetric matrices (see §2.1)

$$\left.\begin{aligned} \underset{\sim}{e}_{\epsilon_j+\epsilon_k} &= \{1/16(\ell-1)\}^{1/2}\{\underset{\sim}{M}_{2j,2k} - i\underset{\sim}{M}_{2j,2k-1} - i\underset{\sim}{M}_{2j-1,2k} - \underset{\sim}{M}_{2j-1,2k-1}\} \\ \underset{\sim}{e}_{\epsilon_j-\epsilon_k} &= \{1/16(\ell-1)\}^{1/2}\{\underset{\sim}{M}_{2j,2k} + i\underset{\sim}{M}_{2j,2k-1} - i\underset{\sim}{M}_{2j-1,2k} + \underset{\sim}{M}_{2j-1,2k-1}\} \\ & \qquad\qquad\qquad\qquad j,k = 1,2,\ldots,5 \end{aligned}\right\}(2.4.2)$$

where $\epsilon_j \pm \epsilon_k$ is a convenient way of denoting the positive roots. In the case of D_5 the relation between the positive roots and the quantities $\epsilon_j \pm \epsilon_k$, $j,k = 1,2,\ldots,5$ is given in Table 7.7.

Table 7.7

Positive Roots	$\epsilon_j \pm \epsilon_k$
α_1	$\epsilon_1 - \epsilon_2$
α_2	$\epsilon_2 - \epsilon_3$
α_3	$\epsilon_3 - \epsilon_4$
α_4	$\epsilon_4 - \epsilon_5$
α_5	$\epsilon_4 + \epsilon_5$
$\alpha_1 + \alpha_2$	$\epsilon_1 - \epsilon_3$
$\alpha_2 + \alpha_3$	$\epsilon_2 - \epsilon_4$
$\alpha_3 + \alpha_4$	$\epsilon_3 - \epsilon_5$
$\alpha_3 + \alpha_5$	$\epsilon_3 + \epsilon_5$
$\alpha_1 + \alpha_2 + \alpha_3$	$\epsilon_1 - \epsilon_4$
$\alpha_2 + \alpha_3 + \alpha_4$	$\epsilon_2 - \epsilon_5$
$\alpha_3 + \alpha_4 + \alpha_5$	$\epsilon_3 + \epsilon_4$
$\alpha_2 + \alpha_3 + \alpha_5$	$\epsilon_2 + \epsilon_5$
$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4$	$\epsilon_1 - \epsilon_5$
$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_5$	$\epsilon_1 + \epsilon_5$
$\alpha_2 + \alpha_3 + \alpha_4 + \alpha_5$	$\epsilon_2 + \epsilon_4$
$\alpha_2 + 2\alpha_3 + \alpha_4 + \alpha_5$	$\epsilon_2 + \epsilon_3$
$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5$	$\epsilon_1 + \epsilon_4$
$\alpha_1 + 2\alpha_2 + 2\alpha_3 + \alpha_4 + \alpha_5$	$\epsilon_1 + \epsilon_2$
$\alpha_1 + \alpha_2 + 2\alpha_3 + \alpha_4 + \alpha_5$	$\epsilon_1 + \epsilon_3$

In more detail we have

(a) D_5

$$\left.\begin{aligned}
e_{\alpha_1} &\equiv e_{\epsilon_1-\epsilon_2} = \sqrt{\tfrac{1}{16.4}}(\underset{\sim}{M}_{2,4}+i\underset{\sim}{M}_{2,3}-i\underset{\sim}{M}_{1,4}+\underset{\sim}{M}_{1,3}) \\
e_{\alpha_2} &\equiv e_{\epsilon_2-\epsilon_3} = \sqrt{\tfrac{1}{16.4}}(\underset{\sim}{M}_{4,6}+i\underset{\sim}{M}_{4,5}-i\underset{\sim}{M}_{3,6}+\underset{\sim}{M}_{3,5}) \\
e_{\alpha_3} &\equiv e_{\epsilon_3-\epsilon_4} = \sqrt{\tfrac{1}{16.4}}(\underset{\sim}{M}_{6,8}+i\underset{\sim}{M}_{6,7}-i\underset{\sim}{M}_{5,8}+\underset{\sim}{M}_{5,7}) \\
e_{\alpha_4} &\equiv e_{\epsilon_4-\epsilon_5} = \sqrt{\tfrac{1}{16.4}}(\underset{\sim}{M}_{8,10}+i\underset{\sim}{M}_{8,9}-i\underset{\sim}{M}_{7,10}+\underset{\sim}{M}_{7,9}) \\
e_{\alpha_5} &\equiv e_{\epsilon_4+\epsilon_5} = \sqrt{\tfrac{1}{16.4}}(\underset{\sim}{M}_{8,10}-i\underset{\sim}{M}_{8,9}-i\underset{\sim}{M}_{7,10}-\underset{\sim}{M}_{7,9})
\end{aligned}\right\} \tag{2.4.3}$$

(b) D_3

$$\left.\begin{aligned}
e_{\alpha_1} &= e_{\epsilon_1-\epsilon_2} = \sqrt{\tfrac{1}{16.2}}(\underset{\sim}{M}_{2,4}+i\underset{\sim}{M}_{2,3}-i\underset{\sim}{M}_{1,4}+\underset{\sim}{M}_{1,3}) \\
e_{\alpha_2} &= e_{\epsilon_2-\epsilon_3} = \sqrt{\tfrac{1}{16.2}}(\underset{\sim}{M}_{4,6}+i\underset{\sim}{M}_{4,5}-i\underset{\sim}{M}_{3,6}+\underset{\sim}{M}_{3,5}) \\
e_{\alpha_3} &= e_{\epsilon_2+\epsilon_3} = \sqrt{\tfrac{1}{16.2}}(\underset{\sim}{M}_{4,6}-i\underset{\sim}{M}_{4,5}-i\underset{\sim}{M}_{3,6}-\underset{\sim}{M}_{3,5})
\end{aligned}\right\} \tag{2.4.4}$$

From (2.4.3), (2.4.4) and Table 7.7 we have

$$\left.\begin{aligned}
e^{D_5}_{\alpha_1} &= \tfrac{1}{\sqrt{2}}e^{D_3}_{\alpha_1} \\
e^{D_5}_{\alpha_2} &= \tfrac{1}{\sqrt{2}}e^{D_3}_{\alpha_1} \\
e^{D_5}_{\alpha_2+2\alpha_3+\alpha_4+\alpha_5} &= \tfrac{1}{\sqrt{2}}e^{D_3}_{\alpha_3}
\end{aligned}\right\} \tag{2.4.5}$$

Changing to the E_{α_i}, $i = 1,2,\ldots,$, basis using the relation $E_\alpha = \{2/\langle\alpha,\alpha\rangle\}^{1/2}e_\alpha$ ((4.1.20) of Chapter 4) and the values of the quantities $\langle\alpha,\alpha\rangle$ from Appendix B, §B.4, we get

$$\left.\begin{aligned} E^{D_3}_{\alpha_1} &= E^{D_5}_{\alpha_1} \\ E^{D_3}_{\alpha_2} &= E^{D_5}_{\alpha_2} \\ E^{D_3}_{\alpha_3} &= E^{D_5}_{\alpha_2+2\alpha_3+\alpha_4+\alpha_5} \end{aligned}\right\} \qquad (2.4.6)$$

Using the fact that $D_3 \simeq A_3$, we finally have

$$\left.\begin{aligned} E^{c}_{\alpha_1} &\equiv E^{A_2}_{\alpha_1} = E^{D_5}_{\alpha_1} \\ E^{c}_{\alpha_2} &\equiv E^{A_2}_{\alpha_2} = E^{D_5}_{\alpha_2} \end{aligned}\right\} \qquad (2.4.7)$$

Now, we are ready to study the mass relations in the SO(10) theory.

§7.2.5 Mass relations

In Figure 7.1 we give the weight diagram of the $D(1_5)$ representation with the phases of the fermion fields we have used in Chapter 5.

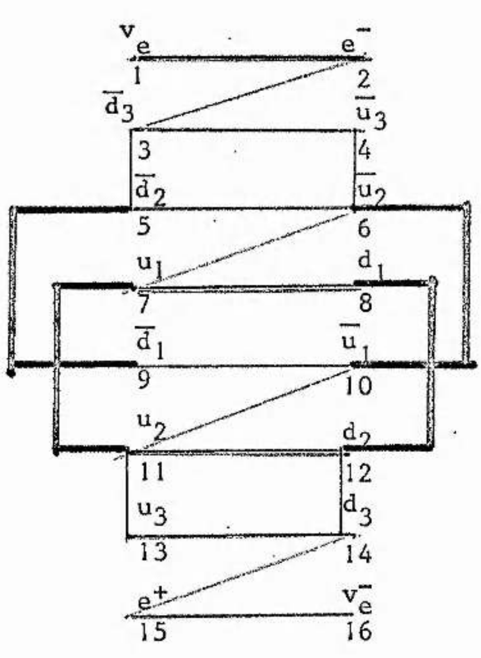


Figure 7.1: Phases for the fermion fields in the $D(1_5)$ representation

For the $D(1_4)$ representation, following reference [62], the generators of the $D(1_4)$ representation are the negatives of the $D(1_5)$ generators, so we have

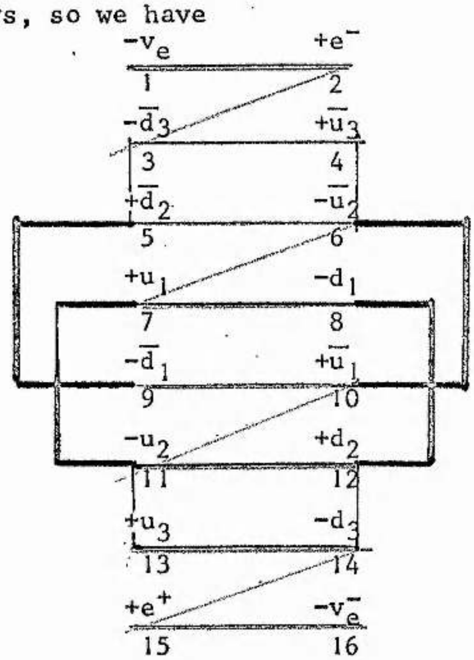


Figure 7.2: Phases for the fermion fields in the $D(1_4)$ representation

(a) If the spinless meson fields are transforming as the 10-dimensional representation, then we have the following couplings:

state ϕ_5

$$\frac{1}{2\sqrt{2}}(\psi_2\psi_{15}-\psi_3\psi_{14}+\psi_5\psi_{12}-\psi_8\psi_9-\psi_9\psi_8+\psi_{12}\psi_5-\psi_{14}\psi_3+\psi_{15}\psi_2)$$

state ϕ_6

$$\frac{1}{2\sqrt{2}}(\psi_1\psi_{16}-\psi_4\psi_{13}+\psi_6\psi_{11}-\psi_7\psi_{10}-\psi_{10}\psi_7+\psi_{11}\psi_6-\psi_{13}\psi_4+\psi_{16}\psi_1)$$

Using the phases from Figures 7.1 and 7.2, and (2.3.2), we have (after a rearrangement of the terms)

state ϕ_5

$$\frac{1}{2\sqrt{2}}(e^+_R e^+_L+\overline{d}_{3R}\overline{d}_{3L}+\overline{d}_{2R}\overline{d}_{2L}+\overline{d}_{1R}\overline{d}_{1L}+d_{1R}d_{1L}+d_{2R}d_{2L}+d_{3R}d_{3L}+\overline{e}_R\overline{e}_L)$$

state ϕ_6

$$-\frac{1}{2\sqrt{2}}(\overline{v}_{e_R}\overline{v}_{e_L}+\overline{u}_{3R}\overline{u}_{3L}+\overline{u}_{2R}\overline{u}_{2L}+\overline{u}_{1R}\overline{u}_{1L}+u_{1R}u_{1L}+u_{2R}u_{2L}+u_{3R}u_{3L}+v_{e_R}v_{e_R})$$

Thus, if $<\phi_5>_0 = <\phi_6>_0 = a \neq 0$, we get

$$-(<(U_R,V_{e_R})|(U_L,V_{e_L})>+<(\overline{U}_R,\overline{V}_{e_R})|(\overline{U}_L,\overline{V}_{e_L})>)a$$

and

$$+(<(D_R,e_R^-)|(D_L,e_L^-)>+<(D_R,e^+)|(D_L,e_L^+)>)a$$

from which we deduce

$$\left.\begin{array}{l} m_{e^-} = m_d \\ m_{v_e} = m_u \end{array}\right\} \qquad (2.5.1)$$

Note

The conditions (2.4.1), namely,

$$\left.\begin{array}{l} H^c_{\alpha_1}\phi_i = 0 \\ H^c_{\alpha_2}\phi_i = 0 \\ E^c_{\alpha_1}\phi_i = 0 \\ E^c_{\alpha_2}\phi_i = 0 \end{array}\right\} \quad i = 5,6$$

are trivially satisfied.

(b) If the spinless meson fields are transforming as the 126-dimensional representation, then we have the following couplings:

I. (d,e^-) system

state 61

$$\frac{1}{2}(\psi_2\psi_{15}+\psi_3\psi_{14}+\psi_{14}\psi_3+\psi_{15}\psi_2)$$

state 62

$$\frac{1}{2}(\psi_5\psi_{12}+\psi_8\psi_9+\psi_9\psi_8+\psi_{12}\psi_5)$$

state 63

$$\frac{1}{2\sqrt{2}}(-\psi_2\psi_{15}+\psi_3\psi_{14}+\psi_5\psi_{12}-\psi_8\psi_9-\psi_9\psi_8+\psi_{12}\psi_5+\psi_{14}\psi_3-\psi_{15}\psi_2).$$

Using the phases of Figures 7.1 and 7.2, we have

state 61

$$\frac{1}{2}(e^+_R e^+_L-\overline{d}_{3R}\overline{d}_{3L}-d_{3R}d_{3L}+e^-_R e^-_L).$$

state 63

$$\frac{1}{2\sqrt{2}}(-e^+_R e^+_L-\overline{d}_{3R}\overline{d}_{3L}+\overline{d}_{2R}\overline{d}_{2L}+\overline{d}_{1R}\overline{d}_{1L}+d_{1R}d_{1L}+d_{2R}d_{2L}-d_{3R}d_{3L}-e^-_R e^-_L).$$

We consider the following linear combination of meson states which have the same weights

$$\phi = a\phi_{61} + b\phi_{62} + c\phi_{63}.$$

The conditions $H^c_{\alpha_1}\phi = 0$ and $H^c_{\alpha_2}\phi = 0$ are satisfied (see Table 7.4). For the $E^c_{\alpha_1}$ we have

$$E^c_{\alpha_1}\phi \equiv E^{D_5}_{\alpha_1}\phi = E^{D_5}_{\alpha_1}(a\phi_{61}+b\phi_{62}+c\phi_{63})$$

$$= b\frac{1}{\sqrt{2}}\phi_{30} \quad \text{(from Table 5.5)}$$

and if we demand

$$E_{\alpha_1}\phi = 0,$$

then b = 0. For the $E^c_{\alpha_2}$ operator we have

$$E^c_{\alpha_2}\phi = E_{\alpha_2}\phi = E_{\alpha_2}(a\phi_{61}+c\phi_{63})$$

$$= \frac{1}{\sqrt{2}}a\phi_{48} + c\phi_{48} \quad \text{(from Table 5.5)}.$$

If we again demand

$$E_{\alpha_2}\phi = 0 \Rightarrow a = -\sqrt{2}c.$$

Thus,

$$\phi = -\sqrt{2}c\phi_{61} + c\phi_{63} = c(-\sqrt{2}\phi_{61}+\phi_{63})$$

and in terms of the fermion fields we get

$$\phi = c(-\frac{\sqrt{2}}{2}e^+_R e^+_L - \frac{1}{2\sqrt{2}}e^+_R e^+_L + \frac{\sqrt{2}}{2}\bar{d}_{3R}\bar{d}_{3L} - \frac{1}{2\sqrt{2}}\bar{d}_{3R}\bar{d}_{3L} + \frac{1}{2\sqrt{2}}\bar{d}_{2R}\bar{d}_{2L} + \frac{1}{2\sqrt{2}}\bar{d}_{1R}\bar{d}_{1L} + h.c)$$

or

$$\phi = \frac{c}{2\sqrt{2}}(-3e^+_R e^+_L + \bar{d}_{3R}\bar{d}_{3L} + \bar{d}_{2R}\bar{d}_{2L} + \bar{d}_{1R}\bar{d}_{1L} + h.c).$$

From this relation, and after allowing the meson fields to develop vacuum expectation values, we easily get that

$$3m_d = m_{e^-} .$$

<u>II. (u,ν_e) system</u>

The same arguments and Table 7.6 lead to the following relation:

$$3m_u = m_{\nu_e} .$$

(c) If the spinless meson fields are transforming as the 120-dimensional representation, then we have the following couplings:

<u>I. (d,e^-) system</u>

<u>state ϕ_{57}</u>

$$\frac{1}{2}(\psi_2\psi_{15}+\psi_3\psi_{14}-\psi_{14}\psi_3-\psi_{15}\psi_2)$$

<u>state ϕ_{58}</u>

$$\frac{1}{2}(\psi_5\psi_{12}+\psi_8\psi_9-\psi_9\psi_8-\psi_{12}\psi_5)$$

<u>state ϕ_{59}</u>

$$\frac{1}{2}(\psi_5\psi_{12}-\psi_8\psi_9+\psi_9\psi_8-\psi_{12}\psi_5)$$

<u>state ϕ_{60}</u>

$$\frac{1}{2}(-\psi_2\psi_{15}+\psi_3\psi_{14}-\psi_{14}\psi_3+\psi_{15}\psi_2)$$

and in terms of the fermion fields we have

<u>state ϕ_{57}</u>

$$\frac{1}{2}(e^+_R e^+_L + \bar{d}_{3R}\bar{d}_{3L} - d_{3R}d_{3L} - e^-_R e^-_L)$$

<u>state ϕ_{58}</u>

$$\frac{1}{2}(-\bar{d}_{2R}\bar{d}_{2L}-\bar{d}_{1R}\bar{d}_{2L}+d_{1R}d_{2L}+d_{2R}d_{2L})$$

<u>state ϕ_{59}</u>

$$\frac{1}{2}(e^{+}_{R}e^{+}_{L}-\bar{d}_{3R}\bar{d}_{3L}+d_{3R}d_{3L}-e^{-}_{R}e^{-}_{L})$$

<u>state ϕ_{60}</u>

$$\frac{1}{2}(-e^{+}_{R}e^{+}_{L}+\bar{d}_{3R}\bar{d}_{3L}-d_{3R}d_{3L}+e^{-}_{R}e^{-}_{L}).$$

From these couplings we can immediately see that there are no mass relations between the fermion fields because a typical coupling is of the form

$$\left.\begin{matrix} \bar{d}_{iR}\bar{d}_{iL} - d_{iR}d_{iL} \\ e^{+}_{R}e^{+}_{L} - e^{-}_{R}e^{-}_{L} \end{matrix}\right\} \qquad (2.5.2)$$

<u>II. (u,v_e) system</u>

Again, the states $\phi_{61},\ldots,\phi_{64}$ exhibit couplings like the ones in (2.5.2), so there are no mass relations.

CHAPTER 8

DISCUSSION

In this thesis we have restricted ourselves to the generation of matrix elements and the evaluation of the Clebsch-Gordan coefficients of the SO(10) theory. As we noticed in the introduction of Chapter 5 and in §6.2.2 of Chapter 6, the same analysis can be carried out in the case of the E_6 model. The E_6 model (paragraph c, part I. of Chapter 3) is in the focus of intense investigation and so far there are no concrete results available. A group theoretical analysis of this model, in line with our construction methods of Chapter 4, could be valuable. Moreover, the masses of the fermion fields participating in the E_6 model can be investigated in a similar way as for the case of the SO(10) model of Chapter 7.

In Chapter 6 we found a simple connection between the weights multiplicities and the level of a weight. Some results were stated for the fundamental representations of D_ℓ and exceptional algebras. Because of the importance of these results for generating weight systems, more investigation is needed to study what happens with the other algebras and for representations different from the fundamental ones.

The algorithms developed in Chapter 4 and 5 for generating matrix elements and Clebsch-Gordan coefficients were limited to the G_2 and D_5 algebras and to weights multiplicities not exceeding four. An interesting problem would have been to investigate what happens for the other algebras and for representations with multiplicities greater than four. This investigation could result in a mathematical construction of models based on the SO(14), SO(18), ... orthogonal groups and E_7,E_8 exceptional groups.

Finally, concluding this chapter, we believe that a complete Lie algebra computer package can be developed in order to solve practical computational problems of the Lie algebras and its representation theory.

APPENDIX A

This is an appendix of Chapter 2, so all the relations quoted here refer to this chapter.

§A.1 . Calculation of the Feynman Diagrams of Figures 2.1 and 2.2 in the case of the SU(3) ⊗ U(1) Model

The contribution to the electron mass from the diagrams in Figures 2.1 and 2.2 is

$$I_r = \int \frac{d_4k}{(2\pi)^4} \mathrm{Tr}\,\{(-ig'\gamma^\mu)\frac{i\not k + m_\mu}{k^2 - m_\mu^2}(-ig''\gamma^\nu)\frac{i}{k^2 - m_D^2}\} + \begin{Bmatrix} m_\mu \to m_x \\ g' \to \hat{g}' \\ g'' \to \hat{g}'' \end{Bmatrix} \tag{A.1.1}$$

where $g' = g\cos p$, $\hat{g}' = g\sin p$, $g'' = g\cos\lambda$, $\hat{g}'' = g\sin\lambda$.

After evaluating the tr of (A.1.1) and using (2.1.33) we get

$$I_r = \frac{3}{i}g^2 m_\mu \cos\lambda \cos p \int \frac{d_4k}{(2\pi)^4}\left[\frac{1}{(k^2-m_D^2)(k^2-m_\mu^2)} - \frac{1}{(k^2-m_D^2)(k^2-m_x^2)}\right] \tag{A.1.2}$$

With the use of the Feynman variable x given by the formula

$$\frac{1}{ab} = \int_0^1 dx \frac{1}{[ax + b(1-x)]^2} \tag{A.1.3}$$

equation (A.1.2) becomes

$$I_r = \frac{3}{i}m_\mu g^2 \cos\lambda \cos p \int \frac{d_4k}{(2\pi)^4}\int_0^1 dx\left\{\frac{1}{(k^2-m_D^2(1-x)-m_\mu^2 x)^2} - \frac{1}{(k^2-m_D^2(1-x)-m_x^2 x)^2}\right\} \tag{A.1.4}$$

The integral over k gives [19]:

$$I = i\pi^2\{\int_0^1 dx\ \ell n\,\{m_D^2(1-x)+m_\mu^2 x\} - \int_0^1 dx\ \ell n\,\{m_D^2(1-x)+m_x^2 x\}\}. \qquad (A.1.5)$$

We shall calculate first the integral

$$I_1 = \int_0^1 dx\ \ell n\,\{m_D^2(1-x)+m_\mu^2 x\}. \qquad (A.1.6)$$

Using the identity

$$\int_0^1 dx\ \ell n\,(1+ax) = \frac{1+a}{a}\ \ell n\ (1+a) - 1,$$

I_1 is written as

$$I_1 = \int_0^1 dx\ \ell n\,\{m_D^2 + (m_\mu^2 - m_D^2)x\} = \int_0^1 dx\ \ell n\, m_D^2\{1 + \frac{(m_\mu^2-m_D^2)}{m_D^2}x\}$$

$$= \int_0^1 dx\ \ell n\, m_D^2 + \int_0^1 dx\{1 + \frac{(m_\mu^2-m_D^2)}{m_D^2}x\}$$

$$= \ell n m_D^2 + \frac{1+\frac{m_\mu^2-m_D^2}{m_D^2}}{\frac{m_\mu^2-m_D^2}{m_D^2}}\ \ell n(1+\frac{m_\mu^2-m_D^2}{m_D^2}) - 1.$$

Thus

$$I_1 = \ell n m_D^2 + \frac{m^2}{m_\mu^2 - m_D^2}\ \ell n\ \frac{m_\mu^2}{m_D^2} - 1. \qquad (A.1.7)$$

From the integral

$$I_2 = \int_0^1 dx\ \ell n\,\{m_D^2(1-x)+m_x^2 x\} \qquad (A.1.8)$$

after similar calculations we get

$$I_2 = -\ln m_D^2 + \frac{m_x^2}{m_x^2 - m_D^2} \ln \frac{m_x^2}{m_D^2} + 1. \qquad (A.1.9)$$

Substituting the expressions of I_1 and I_2 to I_r (relation (A.1.2)), we have

$$m_e = m_\mu \cos\lambda \cos p \frac{3g^2}{16\pi^2} \left\{\frac{m_\mu^2}{m_\mu^2 - m_D^2} \ln \frac{m_\mu^2}{m_D^2} + \frac{m_x^2}{m_x^2 - m_D^2} \ln \frac{m_x^2}{m_D^2}\right\}. \qquad (A.1.10)$$

Assuming $m_D \gg m_\mu$, (A.1.10) can be written

$$m_e = m_\mu \cos\lambda \cos p \frac{3g^2}{16\pi^2} \frac{m_x^2}{m_D^2 - m_x^2} \ln \frac{m_D^2}{m_x^2} + O\left(\frac{m_\mu^2}{m_D^2}\right).$$

§A.2 Four-Dimensional Matrices Representing the Generators of the SO(5) Group

$$H_1 = 2(6)^{1/2} \begin{bmatrix} 1 & & & \underset{\sim}{0} \\ & -1 & & \\ & & 1 & \\ \underset{\sim}{0} & & & -1 \end{bmatrix} \qquad H_2 = 2(6)^{1/2} \begin{bmatrix} 1 & & & \underset{\sim}{0} \\ & 1 & & \\ & & -1 & \\ \underset{\sim}{0} & & & -1 \end{bmatrix}$$

$$E_1 = 2(3)^{1/2} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad E_2 = (6)^{-1/2} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$E_3 = 2(3)^{1/2} \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad E_4 = (6)^{-1/2} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$E_{-i} = -\tilde{E}_i, \; i = 1,2,3,4.$$

§A.3 Clebsch-Gordan Coefficients

In Table A1 we give the Clebsch-Gordan coefficients of the tensor product $\underset{\sim}{3} \otimes \underset{\sim}{3} = \underset{\sim}{\overline{3}} \oplus \underset{\sim}{6}$ of SU(3), while the Clebsch-Gordan coefficients of $\underset{\sim}{4} \otimes \underset{\sim}{4} = \underset{\sim}{10} \oplus \underset{\sim}{5} \oplus \underset{\sim}{1}$ of SO(5) are given in Table A2.

We use the following notation. If $\psi_\mu^{(a)}$ and $\psi_\nu^{(b)}$ are the basis functions of the irreducible representations entering the tensor product, then

$$\psi_\mu^{(a)}\psi_\nu^{(b)} = \left(\begin{matrix} a & b \\ \psi & \nu \end{matrix}\,\middle|\,\begin{matrix} j \\ m \end{matrix}\right)\psi_m^j \qquad (A.3.1)$$

where $\Psi_m^{(j)}$ are the irreducible tensors contained in the tensor product $\psi_\mu^{(a)} \otimes \psi_\nu^{(b)}$, and a,b,j are numbers which specify the dimensions of the representations, while μ,ν,m represent the quantum numbers which distinguish the different states within a multiplet. Here μ,ν,m are identified with $\mu = (I_1,Y_1)$, $\nu = (I_2,Y_2)$, $m = (I,Y)$. Thus, a typical table of Clebsch-Gordan coefficients reads

Values of I,Y		Dimension of representations
I_1,Y_1	I_2,Y_2	
Values of I_1,Y_1	Values of I_2,Y_2	Clebsch-Gordan coefficients

Figure A1: Analysis of $\underset{\sim}{3} \otimes \underset{\sim}{3} = \overline{\underset{\sim}{3}} \oplus \underset{\sim}{6}$ for SU(3)

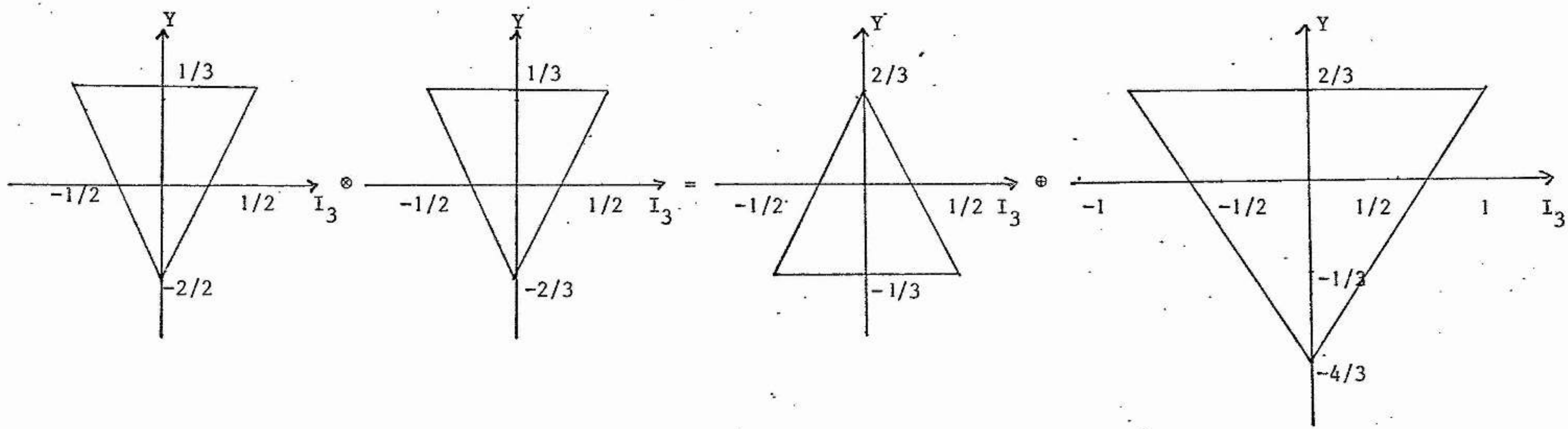


Figure A2: The analysis of $\underset{\sim}{4} \otimes \underset{\sim}{4} = \underset{\sim}{10} \oplus \underset{\sim}{5} \oplus \underset{\sim}{1}$ of SO(5)

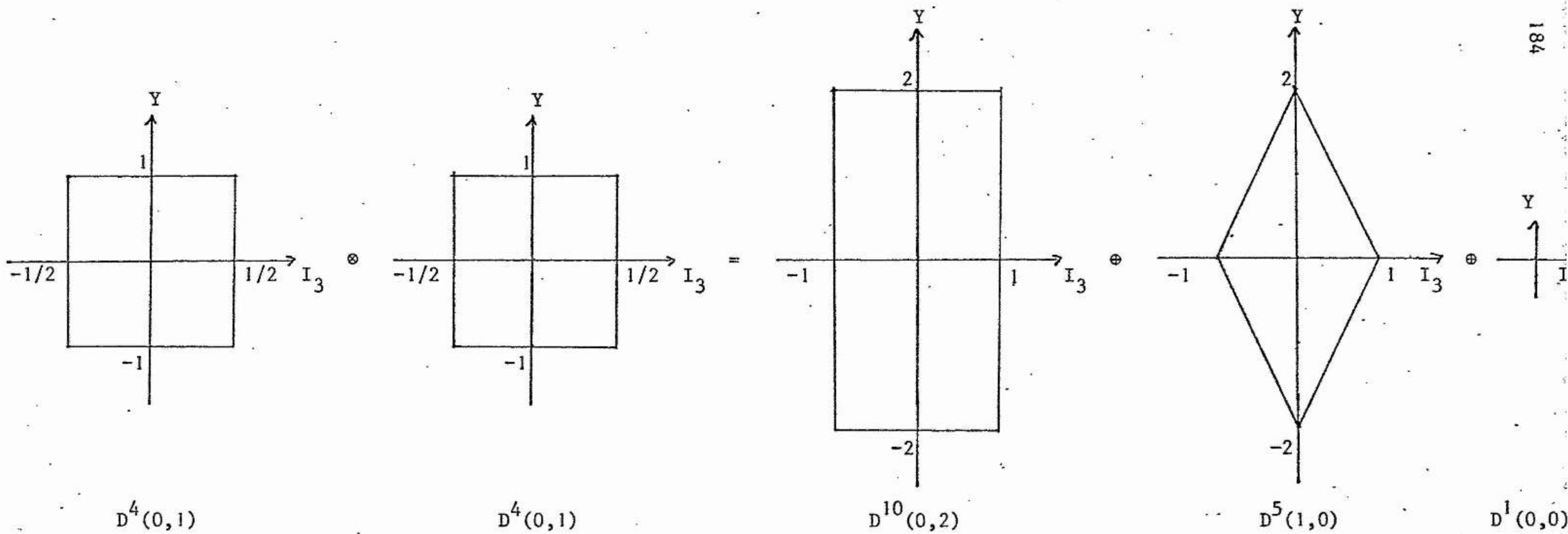


Table A1: Clebsch-Gordan coefficients of the tensor product

$\underset{\sim}{3} \otimes \underset{\sim}{3} = \underset{\sim}{\overline{3}} \oplus \underset{\sim}{6}$ for SU(3)

I = 1/2 , Y = -1/3

I_1	Y_1	I_2	Y_2	$\underset{\sim}{\overline{3}}$	$\underset{\sim}{6}$
1/2	1/3	0	-2/3	$1/\sqrt{2}$	$1/\sqrt{2}$
0	-2/3	1/2	1/3	$-1/\sqrt{2}$	$1/\sqrt{2}$
-1/2	1/3	0	-2/3	$1/\sqrt{2}$	$1/\sqrt{2}$
0	-2/3	-1/2	1/3	$-1/\sqrt{2}$	$1/\sqrt{2}$

I = 1 , Y = 2/3

I_1	Y_1	I_2	Y_2	$\underset{\sim}{6}$
1/2	1/3	1/2	1/3	1
1/2	1/3	-1/2	1/3	$1/\sqrt{2}$
-1/2	1/3	1/2	1/3	$1/\sqrt{2}$
-1/2	1/3	-1/2	1/3	1

I = 0 , Y = 2/3

I_1	Y_1	I_2	Y_2	$\underset{\sim}{\overline{3}}$
1/2	1/3	-1/2	1/3	$1/\sqrt{2}$
-1/2	1/3	1/2	1/3	$-1/\sqrt{2}$

I = 0 , Y = -4/3

I_1	Y_1	I_2	Y_2	$\underset{\sim}{6}$
0	-2/3	0	-2/3	1

Table A.2: The Clebsch-Gordan coefficients of $\underset{\sim}{4} \otimes \underset{\sim}{4} = \underset{\sim}{1} \oplus \underset{\sim}{5} \oplus \underset{\sim}{10}$ of SO(5)

$I = 0 \ , \ Y = 0$

I_1	Y_1	I_2	Y_2	$\underset{\sim}{10}$	$\underset{\sim}{1}$
1/2	1	-1/2	-1	1/2	1/2
-1/2	1	1/2	-1	-1/2	-1/2
1/2	-1	-1/2	1	-1/2	1/2
-1/2	-1	1/2	1	1/2	-1/2

$I = 0 \ , \ Y = 2$

I_1	Y_1	I_2	Y_2	$\underset{\sim}{5}$
1/2	1	-1/2	1	$1/\sqrt{2}$
-1/2	1	1/2	1	$-1/\sqrt{2}$

$I = 0 \ , \ Y = -2$

I_1	Y_1	I_2	Y_2	$\underset{\sim}{5}$
1/2	-1	-1/2	-1	$1/\sqrt{2}$
-1/2	-1	1/2	-1	$-1/\sqrt{2}$

$I = 1 \ , \ Y = 2$

I_1	Y_1	I_2	Y_2	$\underset{\sim}{10}$
1/2	1	1/2	1	-1
-1/2	1	-1/2	1	-1
1/2	1	-1/2	1	$-1/\sqrt{2}$
-1/2	1	1/2	1	$-1/\sqrt{2}$

$I = 1 \ , \ Y = 0$

I_1	Y_1	I_2	Y_2	$\underset{\sim}{5}$	$\underset{\sim}{10}$
1/2	1	1/2	-1	$1/\sqrt{2}$	$1/\sqrt{2}$
1/2	-1	1/2	1	$1/\sqrt{2}$	$1/\sqrt{2}$
-1/2	1	-1/2	-1	$1/\sqrt{2}$	$1/\sqrt{2}$
-1/2	-1	-1/2	1	$-1/\sqrt{2}$	$1/\sqrt{2}$
1/2	1	-1/2	-1	1/2	1/2
-1/2	1	1/2	-1	1/2	1/2
1/2	-1	-1/2	1	-1/2	1/2
-1/2	-1	1/2	1	-1/2	1/2

$I = 1 \ , \ Y = -2$

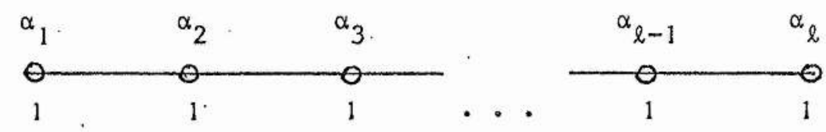
I_1	Y_1	I_2	Y_2	$\underset{\sim}{10}$
1/2	-1	1/2	-1	1
-1/2	-1	-1/2	-1	-1
1/2	-1	-1/2	-1	$1/\sqrt{2}$
-1/2	-1	1/2	-1	$1/\sqrt{2}$

APPENDIX B

DYNKIN DIAGRAMS, CARTAN MATRICES AND THEIR INVERSES, VALUES OF THE QUANTITIES $<\alpha_j,\alpha_k>$ FOR ALL SIMPLE LIE ALGEBRAS

B.1 A_ℓ

(a) Dynkin diagram



(b) Cartan matrix

$$\underset{\sim}{A} = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 2 \end{pmatrix}$$

(c) Inverse of the Cartan matrix

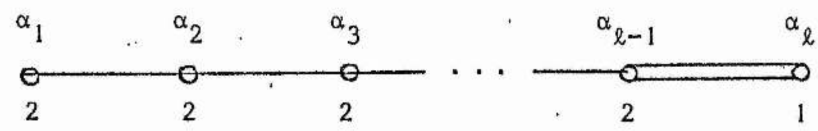
$$\underset{\sim}{A}^{-1} = \frac{1}{(\ell+1)} \begin{pmatrix} \ell & (\ell-1) & (\ell-2) & (\ell-3) & \cdots & 3 & 2 & 1 \\ (\ell-1) & 2(\ell-1) & 2(\ell-2) & 2(\ell-3) & \cdots & 6 & 4 & 2 \\ (\ell-2) & 2(\ell-2) & 3(\ell-2) & 3(\ell-3) & \cdots & 9 & 6 & 3 \\ (\ell-3) & 2(\ell-3) & 3(\ell-3) & 4(\ell-3) & \cdots & 12 & 8 & 4 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 3 & 6 & 9 & 12 & & 3(\ell-2) & 2(\ell-2) & (\ell-2) \\ 2 & 4 & 6 & 8 & & 2(\ell-2) & 2(\ell-1) & (\ell-1) \\ 1 & 2 & 3 & 4 & & (\ell-2) & (\ell-1) & \ell \end{pmatrix}$$

(d)

$$<\alpha_j,\alpha_k> = \begin{cases} 1/(\ell+1) & , j = k \ (j = 1,2,\ldots,\ell) \\ -1/\{2(\ell+1)\} & , j = k \pm 1 \ (j,k = 1,2,\ldots,\ell) \\ 0 & , \text{all other values of } j,k \ (j,k = 1,2,\ldots,\ell) \end{cases}$$

B.2 B_ℓ

(a) Dynkin diagram



(b) Cartan matrix

$$\underset{\sim}{A} = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & 0 & \cdots & 0 & -2 & 0 \end{pmatrix}$$

(c) Inverse of the Cartan matrix

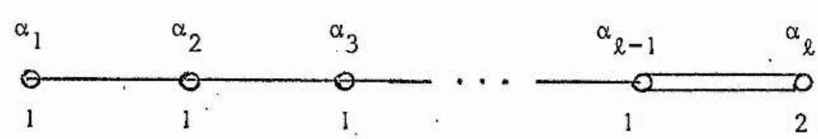
$$\underset{\sim}{A}^{-1} = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1/2 \\ 1 & 2 & 2 & 2 & \cdots & 2 & 2 & 1 \\ 1 & 2 & 3 & 3 & \cdots & 3 & 3 & 3/2 \\ 1 & 2 & 3 & 4 & \cdots & 4 & 4 & 2 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 1 & 2 & 3 & 4 & \cdots & (\ell-2) & (\ell-2) & \frac{1}{2}(\ell-2) \\ 1 & 2 & 3 & 4 & \cdots & (\ell-2) & (\ell-1) & \frac{1}{2}(\ell-1) \\ 1 & 2 & 3 & 4 & \cdots & (\ell-2) & (\ell-1) & \frac{1}{2}\ell \end{pmatrix}$$

(d)

$$\langle\alpha_j,\alpha_k\rangle = \begin{cases} 1/(2\ell-1) & , j = k \ (j = 1,2,\ldots,\ell-1) \\ 1/\{2(2\ell-1)\} & , j = k = \ell \\ -1/\{2(2\ell-1)\} & , j = k \pm 1 \ (j,k = 1,2,\ldots,\ell) \\ 0 & , \text{all other values of } j,k \ (j,k = 1,2,\ldots,\ell) \end{cases}$$

B.3 C_ℓ

(a) Dynkin diagram



(b) Cartan matrix

$$\underset{\sim}{A} = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 2 & -2 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 2 \end{bmatrix}$$

(c) Inverse of the Cartan matrix

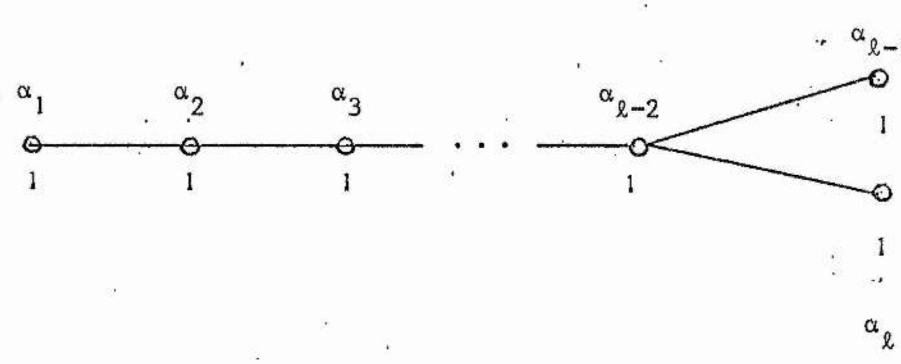
$$\underset{\sim}{A}^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & \cdots & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 & \cdots & 3 & 3 & 3 \\ 1 & 2 & 3 & 4 & \cdots & 4 & 4 & 4 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 1 & 2 & 3 & 4 & \cdots & (\ell-2) & (\ell-2) & (\ell-2) \\ 1 & 2 & 3 & 4 & \cdots & (\ell-2) & (\ell-1) & (\ell-1) \\ 1/2 & 1 & 3/2 & 2 & \cdots & (\ell-2) & \frac{1}{2}(\ell-1) & \frac{1}{2}\ell \end{bmatrix}$$

(d)

$$\langle\alpha_j,\alpha_k\rangle = \begin{cases} 1/\{2(\ell+1)\} & , j = k \ (j = 1,2,\ldots,\ell-1) \\ 1/(\ell+1) & , j = k = \ell \\ -1/\{4(\ell+1)\} & , j = k \pm 1 \ (j,k = 1,2,\ldots,\ell-1) \\ -1/\{2(\ell+1)\} & , j = \ell - 1,\ k = \ell \text{ and } j = \ell,\ k = \ell - 1 \\ 0 & , \text{all other values of } j,k \ (j,k = 1,2,\ldots,\ell) \end{cases}$$

B.4 D_ℓ

(a) Dynkin diagram



(b) Cartan matrix

$$\underset{\sim}{A} = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 2 & -1 & -1 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 2 \end{pmatrix}$$

(c) Inverse of the Cartan matrix

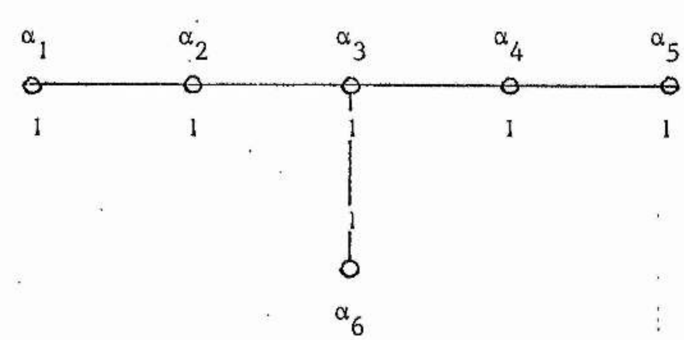
$$\underset{\sim}{A}^{-1} = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 1/2 & 1/2 \\ 1 & 2 & 2 & 2 & \cdots & 2 & 1 & 1 \\ 1 & 2 & 3 & 3 & \cdots & 3 & 3/2 & 3/2 \\ 1 & 2 & 3 & 4 & \cdots & 4 & 2 & 2 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 1 & 2 & 3 & 4 & \cdots & (\ell-2) & \frac{1}{2}(\ell-2) & \frac{1}{2}(\ell-2) \\ 1/2 & 1 & 3/2 & 2 & \cdots & \frac{1}{2}(\ell-2) & \frac{1}{4}\ell & \frac{1}{4}(\ell-2) \\ 1/2 & 1 & 3/2 & 2 & \cdots & \frac{1}{2}(\ell-2) & \frac{1}{4}(\ell-2) & \frac{1}{4}\ell \end{pmatrix}$$

(d)

$$\langle\alpha_j,\alpha_k\rangle = \begin{cases} 1/\{2(\ell-1)\} & , j = k \ (j = 1,2,\ldots,\ell) \\ -1/\{4(\ell-1)\} & , j = k \pm 1 \ (j,k = 1,2,\ldots,\ell-3); \\ & \quad j = \ell - 2 \text{ with } k = \ell - 1,\ \ell; \text{ and} \\ & \quad k = \ell - 2 \text{ with } j = \ell - 1,\ \ell \text{ (all for} \\ & \quad \ell \geq 3) \\ 0 & , \text{all other values of } j,k \ (j,k = 1,2,\ldots,\ell) \end{cases}$$

B.5 E_6

(a) Dynkin diagram



(b) Cartan matrix

$$\underset{\sim}{A} = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & -1 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & 0 & 2 \end{pmatrix}$$

(c) Inverse of the Cartan matrix

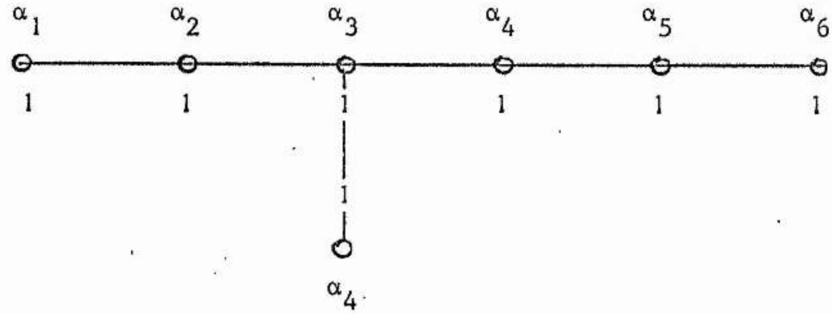
$$\underset{\sim}{A}^{-1} = \frac{1}{3} \begin{pmatrix} 4 & 5 & 6 & 4 & 2 & 3 \\ 5 & 10 & 12 & 8 & 4 & 6 \\ 6 & 12 & 18 & 12 & 6 & 9 \\ 4 & 8 & 12 & 10 & 5 & 6 \\ 2 & 4 & 6 & 5 & 4 & 3 \\ 3 & 6 & 9 & 6 & 3 & 6 \end{pmatrix}$$

(d)

$$\langle \alpha_j, \alpha_k \rangle = \begin{cases} 1/12 & , j = k \ (j = 1,2,\ldots,6) \\ -1/24 & , j \neq k, \ (j,k) = (1,2), (2,1), (2,3), (3,2), \\ & \quad (3,4), (4,3), (3,6), (6,3), (4,5), (5,4) \\ 0 & , \text{for all other pairs } (j,k) \ (j,k = 1,2,\ldots,6) \\ & \quad \text{with } j \neq k \end{cases}$$

<u>B.6 E_7</u>

(a) Dynkin diagram



(b) Cartan matrix

$$\underset{\sim}{A} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 2 \end{bmatrix}$$

(c) Inverse of the Cartan matrix

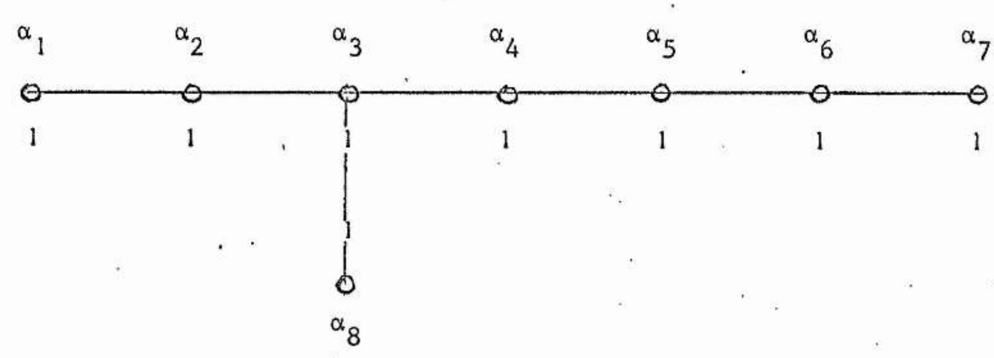
$$\underset{\sim}{A}^{-1} = \begin{bmatrix} 2 & 3 & 4 & 3 & 2 & 1 & 2 \\ 3 & 6 & 8 & 6 & 4 & 2 & 4 \\ 4 & 8 & 12 & 9 & 6 & 3 & 6 \\ 3 & 6 & 9 & 15/2 & 5 & 5/2 & 9/2 \\ 2 & 4 & 6 & 5 & 4 & 2 & 3 \\ 1 & 2 & 3 & 5/2 & 2 & 3/2 & 3/2 \\ 2 & 4 & 6 & 9/2 & 3 & 3/2 & 7/2 \end{bmatrix}$$

(d)

$$\langle\alpha_j,\alpha_k\rangle = \begin{cases} 1/18 , & j = k \ (j = 1,2,\ldots,7) \\ -1/36 , & j \neq k , \ (j,k) = (1,2),\ (2,1),\ (2,3),\ (3,2) \\ & (3,4),\ (4,3),\ (3,7),\ (7,3),\ (4,5),\ (5,4), \\ & (5,6),\ (6,5) \\ 0 , & \text{for all other pairs } (j,k) \ (j,k = 1,2,\ldots,7) \\ & \text{with } j \neq k \end{cases}$$

B.7 E_8

(a) Dynkin diagram



(b) Cartan matrix

$$\underset{\sim}{A} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

(c) Inverse of the Cartan matrix

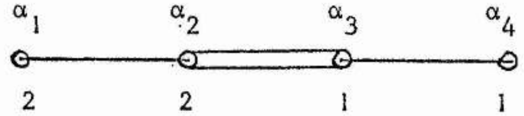
$$\underset{\sim}{A}^{-1} = \begin{bmatrix} 4 & 7 & 10 & 8 & 6 & 4 & 2 & 5 \\ 7 & 14 & 20 & 16 & 12 & 8 & 4 & 10 \\ 10 & 20 & 30 & 24 & 18 & 12 & 6 & 15 \\ 8 & 16 & 24 & 20 & 15 & 10 & 5 & 12 \\ 6 & 12 & 18 & 15 & 12 & 8 & 4 & 9 \\ 4 & 8 & 12 & 10 & 8 & 6 & 3 & 6 \\ 2 & 4 & 6 & 5 & 4 & 3 & 2 & 3 \\ 5 & 10 & 15 & 12 & 9 & 6 & 3 & 8 \end{bmatrix}$$

(d)

$$\langle \alpha_j, \alpha_k \rangle = \begin{cases} 1/30 & , j = k \ (j = 1,2,\ldots,8) \\ -1/60 & , j \neq k , (j,k) = (1,2), (2,1), (2,3), (3,2) \\ & \quad (3,4), (4,3), (3,8), (8,3), (4,5), (5,4), \\ & \quad (5,6), (6,5), (6,7), (7,6) \\ 0 & , \text{for all other pairs } (j,k) \ (j,k = 1,2,\ldots,8) \\ & \quad \text{with } j \neq k \end{cases}$$

<u>B.8 F_4</u>

(a) Dynkin diagram



(b) Cartan matrix

$$\underset{\sim}{A} = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -2 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

(c) Inverse of the Cartan matrix

$$\underset{\sim}{A}^{-1} = \begin{pmatrix} 2 & 3 & 2 & 1 \\ 3 & 6 & 4 & 2 \\ 4 & 8 & 6 & 3 \\ 2 & 4 & 3 & 2 \end{pmatrix}$$

(d)

$$\langle\alpha_1,\alpha_1\rangle = \langle\alpha_2,\alpha_2\rangle = 1/9, \langle\alpha_3,\alpha_3\rangle = \langle\alpha_4,\alpha_4\rangle = 1/18,$$

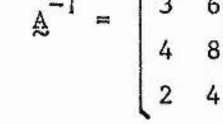
$$\langle\alpha_1,\alpha_2\rangle = \langle\alpha_2,\alpha_1\rangle = \langle\alpha_2,\alpha_3\rangle = \langle\alpha_3,\alpha_2\rangle = -1/18,$$

$$\langle\alpha_3,\alpha_4\rangle = \langle\alpha_4,\alpha_3\rangle = -1/36$$

with $\langle\alpha_j,\alpha_k\rangle = 0$ for all other pairs j,k ($j,k = 1,2,3,4$).

<u>B.9 G_2</u>

(a) Dynkin diagram



(b) Cartan matrix

$$\underset{\sim}{A} = \begin{pmatrix} 2 & -1 \\ -3 & 2 \end{pmatrix}$$

(c) Inverse of the Cartan matrix

$$\underset{\sim}{A}^{-1} = \begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix}$$

(d)

$$\langle\alpha_1,\alpha_1\rangle = 1/4,\ \langle\alpha_1,\alpha_2\rangle = \langle\alpha_2,\alpha_1\rangle = -1/8,\ \langle\alpha_2,\alpha_2\rangle = 1/12$$

APPENDIX C

PROGRAMS

In this appendix all the programs which have been developed for Chapters 4, 5 and 6 are given. These programs are naturally divided into five groups A, B, C, D and E. Each of these groups corresponds to each section of Chapter 4. Therefore each group includes those programs which implement the algorithms discussed in the corresponding section of Chapter 4 according to Table C1 (see over).

Each program has the general structure

I Declarations

II Procedures

III Main program.

In I, all the variables and arrays of different types (integer, real, long real) are declared. In II, the procedures which are called in the main program are stated. This part of the program always includes procedures which control the input and output of the program. Finally, part III is the main program.

As a programming language ALGOL-W was used. ALGOL-W is an advanced version of ALGOL-60. The basic characteristic of the ALGOL programming languages is their block structure. The block structure of higher level programming language allows the computer implementation of very complicated mathematical algorithms.

The programs were run in the IBM/360 computer of the Computing Laboratory of the University of St Andrews. After the installation of a new computer, VAX-VMS/11, the programs of group B were executed in this new system, after being translated in FORTRAN IV.

Table C1

Section of Chapter 4	Group		Content of the Group
§4.1 Root Systems	A	A1(1)	Calculation of the positive roots
§4.2 Weight Systems	B	B1(2)	Dimensionality of representations
		B2(2)	Weights without multiplicity
		B3(2)	Weights algorithm 1
		B4(2)	Weights algorithm 2
§4.3 Clebsch-Gordan Series	C	C1(3)	Weyl group
		C2(3)	Kostant-Steinberg formula
		C3(3)	Higher order indices
		C4(3)	Higher order indices
		C5(3)	Higher order indices
		C6(3)	Higher order indices
		C7(3)	Higher order indices
§4.4 Matrix Representation	D	D1(4)	Matrix representation of A_1
		D2(4)	Diagonal generators, version 1
		D3(4)	Diagonal generators, version 2
		D4(4)	Non-diagonal generators of G_2
		D5(4)	Non-diagonal generators of D_5
		D6(4)	Test program
§4.5 Clebsch-Gordan Coefficients	E	E1(5)	Clebsch-Gordan coefficients at G_2
		E2(5)	Clebsch-Gordan coefficients at D_5

§C.1 Group A

```
BEGIN COMMENT CALCULATION OF THE POSITIVE ROOTS,PROGRAM A1(1);
INTEGER N,R;READON(N,R);I_W:=1;S_W:=1;
WRITE("THE POSITIVE ROOT SYSTEM OF  THE ALGEBRA ");
WRITE("THE NUMBER OF POSITIVE ROOTS IS=",N);
WRITE("THE RANK OF THE ALGEBRA IS=",R);
BEGIN
INTEGER ARRAY ROT(1::N ,1::R);INTEGER ARRAY CAR(1::R,1::R);
INTEGER ARRAY NROT(1::R);
INTEGER S1,LAST,B,W,A;


PROCEDURE INPUTI(INTEGER ARRAY A(*,*));
BEGIN
      FOR I:=1 UNTIL R DO
      BEGIN
             IOCONTROL(2);
            FOR J:=1 UNTIL R DO
            BEGIN
                  READON(A(I,J));
                  WRITEON(A(I,J))
            END;
            WRITE(" ")
      END
END;


PROCEDURE OUTPUTI(INTEGER ARRAY A(*,*));
BEGIN
      FOR I:=1 UNTIL N DO
      BEGIN
            IOCONTROL(2);
            FOR J:=1 UNTIL R DO
            WRITEON(A(I,J));
      END
END;

I_W:=2;S_W:=1;

COMMENT 1 CALCULATION OF THE ROOTS;
WRITE("SIMPLE ROOTS");INPUTI(ROT);
WRITE("CARTAN MATRIX");INPUTI(CAR);
LAST:=R;W:=1;
WHILE W<=LAST DO
BEGIN
      FOR I:=1 UNTIL R DO
      BEGIN
            S1:=0;
            FOR J:=1 UNTIL R DO
            S1:=S1+ROT(W,J)*CAR(I,J);
            IF S1<=0 THEN A:=-1
                      ELSE A:=+1;
            FOR Q:=1 UNTIL ABS(S1) DO
            BEGIN
                  B:=0;
```

```
                FOR K:=1 UNTIL R DO
                BEGIN
                    NROT(K):=ROT(W,K)-A*Q*ROT(I,K);
                    IF NROT(K)<0 THEN GOTO L;
                    B:=B+NROT(K)
                END;
                IF B=0 THEN GOTO L;
                FOR M:=1 UNTIL LAST DO
                BEGIN
                    FOR K:=1 UNTIL R DO
                    IF ABS(ROT(M,K)-NROT(K))>0 THEN GOTO L1;
                    GOTO L;
                L1:END;
                LAST:=LAST+1;
                FOR K:=1 UNTIL R DO
                ROT(LAST,K):=NROT(K);
                IF LAST=N THEN GOTO EXIT;
            L:END;
        END;
        W:=W+1
END;
EXIT:WRITE("THE POSITIVE ROOTS ARE");OUTPUTI(ROT)
END
END.
```

§C.2   Group B

I   Program B1(2)

```
BEGIN COMMENT CALCULATION OF THE DIMENSIONALITY,PROGRAM B1(2);
INTEGER N,R,N1,N2,N3; READON(N,R,N1,N2,N3);
BEGIN
INTEGER ARRAY ROT(1::N,1::R);LONG REAL ARRAY L(1::1,1::R+1);
INTEGER AF [illegible] C [illegible] .::R,1:: [illegible];
INTEGER A [illegible] Y N [illegible] ,T,K,M1(1::R
LONG REAL [illegible] RA [illegible] A(1::R);
INTEGER A,LAST [illegible] ,B,P1,P2,S1,G; LONG REAL M;


PROCEDURE INPUTI(INTEGER ARRAY A(*,*));
BEGIN
    FOR I:=1 UNTIL R DO
    BEGIN
        IOCONTROL(2);
        FOR J:=1 UNTIL R DO
        BEGIN
            I_W:=2;S_W:=1;
            READON(A(I,J));
            WRITEON(A(I,J))
        END;
        WRITE(" ")
    END
END;
```

```
PROCEDURE OUTPUTR(LONG REAL ARRAY A(*,*);INTEGER VALUE M,L);
BEGIN
     FOR I:=1 UNTIL M DO
     BEGIN
          IOCONTROL(2);
          FOR J:=1 UNTIL L-1 DO
          BEGIN
               I_W:=2;S_W:=1;
               WRITEON(TRUNCATE(A(I,J)))
          END;
          WRITEON(A(I,L));
     END
END;

PROCEDURE PRODUCT(INTEGER ARRAY C(*);INTEGER VALUE I;
                    INTEGER VALUE RESULT P);
BEGIN
  FOR J:=1 UNTIL R DO
  P:=P+(C(J)+1)*T(J)*ROT(I,J)
END;

COMMENT 1 CALCULATION OF THE POSITIVE ROOTS;
COMMENT CALL PROGRAM A1(1);
------------------------------------------
COMMENT 2 CULCULATION OF THE DIMENTIONALITY;
FOR I:=1 UNTIL R DO
BEGIN
     READON(T(I));
     WRITEON(T(I))
END;
WRITE("THE DIM OF THE REPR ARE ");
LAST:=0;
FOR I:=1 UNTIL R-1 DO
M1(I):=N2;M1(R):=N2-1;Q:=R;
WHILE 1=1 DO
BEGIN
     IF M1(Q)=N3 THEN
     WHILE M1(Q)=N3 DO
     IF Q>1 THEN Q:=Q-1
     ELSE GOTO FIN;
     M1(Q):=M1(Q)+1;
     FOR I:=Q+1 UNTIL R DO
     M1(I):=N2;Q:=R;
     M:=1;
     FOR I:=1 UNTIL N DO
     BEGIN
          P1:=P2:=0;
          FOR I:=1 UNTIL R DO
          K(I):=M1(I);
          COMMENT 2 THE PROCEDURE  PRODUCT REPRESENTS THE WEYL'S
          DIMENTIONAL FORMULA;
          PRODUCT(K,I,P1);
          FOR I:=1 UNTIL R DO  K(I):=0;
          PRODUCT(K,I,P2);
          M:=M*(P1/P2)
     END;
```

```
      LAST:=LAST+1;
      IF LAST=N1+1 THEN GOTO FIN;
      FOR I:=1 UNTIL R DO
      L(1,I):=M1(I);L(1,R+1):=M;
      OUTPUTR(L,1,R+1)
 END;
 FIN:END
 END.
```

II   Program B2(2)

```
BEGIN COMMENT CALCULATION OF THE WEIGHTS WITHOUT MULT,PROGRAM B2(2);
INTEGER N,R;READON(N,R);I_W:=1;S_W:=2;
WRITE("WE ARE CALCULATING THE WEIGHTS WITHOUT MULT");
WRITE("OF THE REPR. DIM=",N);
BEGIN
INTEGER ARRAY ROT(1::N1,1::R);INTEGER ARRAY CAR(1::R,1::R);
LONG REAL ARRAY INVCAR(1::R,1::R);
LONG REAL ARRAY WEIGHTS(1::N,1::R);
LONG REAL ARRAY GH(1::R);LONG REAL ARRAY NWEIGHT(1::R);
INTEGER ARRAY L(1::1,1::R);
INTEGER W,LAST,A;LONG REAL S1,S2;

PROCEDURE INPUTI(INTEGER ARRAY A(*,*); INTEGER VALUE M,L);
BEGIN
     FOR I:=1 UNTIL M DO
     FOR J:=1 UNTIL L DO
     READON(A(I,J))
END;

PROCEDURE INPUTR(LONG REAL ARRAY A(*,*);INTEGER VALUE M,L);
BEGIN
     FOR I:=1 UNTIL M DO
     FOR J:=1 UNTIL L DO
     READON(A(I,J))
END;

PROCEDURE OUTPUTR(LONG REAL ARRAY A(*,*);INTEGER VALUE M,L);
BEGIN
  FOR I:=1 UNTIL M DO
  BEGIN
    R_W:=8;    R_FORMAT:="A";R_D:=5;S_W:=1;
    IOCONTROL(2);
    FOR J:=1 UNTIL L  DO
    WRITEON(A(I,J));
  END
END;
```

```
PROCEDURE ORDER(INTEGER VALUE LAST);
BEGIN
INTEGER T,K,O;LONG REAL ARRAY  WEIGHT1,WEIGHT2(1::1,1::R);
T:=1;
FOR I:=2 UNTIL LAST DO
BEGIN
     FOR J:=1 UNTIL R DO
     BEGIN
          IF (WEIGHTS(T,J)-WEIGHTS(I,J))>0 THEN  T:=T+1
                                   ELSE IF (WEIGHTS(T,J)
                                           -WEIGHTS(I,J))<0 THEN
            BEGIN
                 FOR F:=1 UNTIL R DO
                 BEGIN
                      WEIGHT1(1,F):=WEIGHTS(T,F);
                      WEIGHT2(1,F):=WEIGHTS(I,F);
                      WEIGHTS(T+1,F):=WEIGHT1(1,F);
                      WEIGHTS(T,F):=WEIGHT2(1,F)
                 END;
               K:=I-2;O:=T;
               WHILE K>0 DO
               BEGIN
                    FOR F:=1 UNTIL R DO
                    BEGIN
                         IF (WEIGHTS(K,F)-WEIGHTS(O,F))>0 THEN
                                BEGIN
                                     T:=T+1;GOTO U
                                END
                                     ELSE IF (WEIGHTS(K,F)
                                             -WEIGHTS(O,F))<0 THE
                              BEGIN
                                   FOR G:=1 UNTIL R DO
                                   BEGIN
                                        WEIGHT1(1,G):=WEIGHTS(K,G)
                                      WEIGHT2(1,G):=WEIGHTS(O,G);
                                        WEIGHTS(O,G):=WEIGHT1(1,G)
                                        WEIGHTS(K,G):=WEIGHT2(1,G)
                                   END;
                                   K:=K-1;O:=O-1;GOTO U1
                              END
                    END;
                 U1:END
            END
     END;
U:END
END;


COMMENT 1 BASIC INPUT;
INPUTI(ROT,R,R);INPUTI(CAR,R,N);
INPUTR(INVCAR,R,R);INPUTI(L,1,R);
```

```
COMMENT 2 WEIGHTS WITHOUT MULT;
FOR I:=1 UNTIL R DO
BEGIN
      GH(I):=0;
      FOR J:=1 UNTIL R DO
      GH(I):=GH(I)+INVCAR(J,I)*L(1,J)
END;
FOR I:=1 UNTIL R DO
WEIGHTS(1,I):=GH(I);
LAST:=1;W:=LAST;
WHILE W<=LAST DO
BEGIN
      FOR I:=1 UNTIL R DO
      BEGIN
            S1:=0;
            FOR J:=1 UNTIL R DO
            S1:=S1+WEIGHTS(W,J)*CAR(J,I);
            IF ABS(S1)<1'-3 THEN S1:=0;
            IF S1<=0 THEN A:=-1
                     ELSE A:=1;
            FOR Q:=1 UNTIL  ROUND(ABS(S1)) DO
            BEGIN
                  FOR K:=1 UNTIL R DO
                  NWEIGHT(K)  :=WEIGHTS(W,K)-A*Q*ROT(I,K);
                  FOR M:=1 UNTIL LAST DO
                  BEGIN
                        FOR K:=1 UNTIL R DO
                        BEGIN
                              S2:=WEIGHTS(M,K)-NWEIGHT(K);
                              IF ABS(S2)<1'-3 THEN S2:=0;
                              IF ABS(S2)>0 THEN GOTO L1
                        END;
                        GOTO LA;
               L1:END;
                  LAST:=LAST+1;
                  FOR K:=1 UNTIL R DO
                  WEIGHTS(LAST,K):=NWEIGHT(K);
            LA:END
      END;
      W:=W+1
END;
WRITE("THE NUMBER OF CALCULATED WEIGHTS IS=",LAST);
WRITE("THE-",LAST,"WEIGHTS ORDERED ARE.");
COMMENT 3 REODERING THE WEIGHTS;
ORDER(LAST);
OUTPUTR(WEIGHTS,LAST,R);
END.
```

III Program B3(2)

```
BEGIN COMMENT CALCULATION OF THE WEIGHTS,ALGORITHM 1,PROGRAM B3(2);
INTEGER N,R,N1,LON,SOR;READON(N,R,N1,LON,SOR);I_W:=1;S_W:=2;
WRITE("WE ARE CALCULATING THE WEIGHTS OF THE REPR. DIM=",N );
WRITE("THE RANK OF THE ALGEBRA IS=",R,"THE NUMBER OF POSITIVE ROOTS");
WRITE(" IS=",N1,"THE LEN. OF THE ROOTS IS=",LON,",",SOR);
BEGIN
INTEGER ARRAY ROT(1::N1,1::R);INTEGER ARRAY CAR(1::R,1::R);
LONG REAL ARRAY SCALAR,INVCAR(1::R,1::R);
LONG REAL ARRAY WEIGHTS(1::N,1::R+2);
LONG REAL ARRAY GH,D,X,NWEIGHT(1::R);
INTEGER ARRAY L(1::1,1::R);INTEGER ARRAY M,NROT(1::R);
INTEGER S,LAST,B,W,A,Y,LEVEL;
LONG REAL S1,S2,I1,I2,F,Z;


PROCEDURE INPUTI(INTEGER ARRAY A(*,*); INTEGER VALUE M,L);
BEGIN
      FOR I:=1 UNTIL M DO
      FOR J:=1 UNTIL L DO
      READON(A(I,J))
END;


PROCEDURE INPUTR(LONG REAL ARRAY A(*,*);INTEGER VALUE M,L);
BEGIN
      FOR I:=1 UNTIL M DO
      FOR J:=1 UNTIL L DO
      READON(A(I,J))
END;


PROCEDURE SCA;
BEGIN
      INTEGER X;
      FOR I:=1 UNTIL R DO
      FOR J:=1 UNTIL R DO
      BEGIN
            IF J\=R THEN X:=LON ELSE X:=SOR;
            SCALAR(I,J):=IF I\=J THEN (CAR(I,J)*X)/2 ELSE X
      END
END;


PROCEDURE OUTPUTR(LONG REAL ARRAY A(*,*);INTEGER VALUE M,F);
BEGIN
  FOR I:=F UNTIL M DO
  BEGIN
    R_W:=8;    R_FORMAT:="A";R_D:=5;S_W:=1;
    IOCONTROL(2);
    FOR J:=1 UNTIL R  DO
    WRITEON(A(I,J));
    WRITEON("MULT=",ROUND(A(I,R+1)));
  END
END;
```

```
PROCEDURE OUTPUTI(INTEGER ARRAY A(*,*); INTEGER VALUE M,L,C);
BEGIN
  FOR I:=C UNTIL M DO
  BEGIN
    IOCONTROL(2);
    FOR J:=1 UNTIL L DO
    WRITEON(A(I,J));
    WRITE(" ")
  END
END;

PROCEDURE LHS (LONG REAL VALUE RESULT F);
BEGIN
     FOR I:=1 UNTIL R DO
     FOR J:=1 UNTIL R DO
    F:=F+(GH(I)-M(I)+D(I))*(GH(J)-M(J)+D(J))*SCALAR(I,J)
END;

PROCEDURE LEVL (INTEGER VALUE T);
BEGIN
     FOR I:=1 UNTIL T DO
     BEGIN
          LEVEL:=0;
          FOR J:=1 UNTIL R DO
          LEVEL:=LEVEL+ENTIER(WEIGHTS(1,J)-WEIGHTS(I,J));
          WEIGHTS(I,R+2):=LEVEL
     END
END;




PROCEDURE FREUDENTHAL(INTEGER VALUE T);
BEGIN
     LONG REAL ARRAY P(-1::R);
     WEIGHTS(1,R+1):=1;SCA;
     FOR I:=1 UNTIL R DO
     BEGIN
          D(I):=0;
          FOR J:=1 UNTIL N1 DO
          D(I):=D(I)+ROT(J,I);
          D(I):=D(I)/2
     END;
     I1:=0;FOR F:=1 UNTIL R DO M(F):=0;
     LHS(I1);WRITE("I1=",I1);
     FOR I:=2 UNTIL T DO
     BEGIN
          FOR J:=1 UNTIL R DO
          M(J):=ENTIER(WEIGHTS(1,J)-WEIGHTS(I,J));
          I2:=0;LHS(I2);F:=0;WRITE("I2=",I2,"M(I)=");
          FOR J:=1 UNTIL R DO WRITEON(M(J));
          FOR K:=1 UNTIL I-1 DO
          BEGIN
```

```
                IF WEIGHTS(K,R+2)<WEIGHTS(I,R+2) THEN
                BEGIN
                     FOR J:=1 UNTIL R DO
                     M(J):=ENTIER(WEIGHTS(K,J)-WEIGHTS(I,J));
                     FOR C:=1 UNTIL N1 DO
                     BEGIN
                          Y:=0;
                          FOR J:=1 UNTIL R DO
                          BEGIN
                               IF ROT(C,J)\=0 THEN
                               BEGIN
                                     X(J):=M(J)/ROT(C,J);
                                     Y:=Y+1;P(Y):=X(J);
                               END
                                                        ELSE
                               IF M(J)\=0 THEN GOTO U;
                               P(0):=P(-1):=P(Y);
                               IF P(Y)\=P(Y-1) THEN GOTO U
                          END;
                          S1:=0;
                          FOR Q:=1 UNTIL R DO
                          FOR G:=1 UNTIL R DO
                          S1:=S1+ROT(C,G)*(WEIGHTS(I,Q)+P(Y)*ROT(C,Q))
                                                  *SCALAR(Q,G);
                          F:=F+WEIGHTS(K,R+1)*S1;
                          GOTO U1;
                     U:END;
                END;
          U1:END;
          Z:=2*F/(I1-I2);
          IF Z<1'-3 THEN Z:=0;
          WEIGHTS(I,R+1):=Z;
          OUTPUTR(WEIGHTS,I,I)
     END
END;


COMMENT 1 BASIC INPUT;
INPUTI(ROT,R,R);INPUTI(CAR,R,R);
INPUTR(INVCAR,R,R);INPUTI(L,1,R);


COMMENT 2 CALCULATION OF THE ROOTS;
COMMENT CALL PROGRAM A1(1);
-----------------------------------------


COMMENT 3 WEIGHTS WITHOUT MULT;
COMMENT CALL PROGRAM B2(1);
-----------------------------------------


COMMENT 4 REODERING THE WEIGHTS;
ORDER(LAST);
COMMENT 5 FINDING THE LEVELS OF THE WEIGHTS;
LEVL(LAST);
```

```
COMMENT 6 IF THE NUMBER OF WEIGHTS  IS LESS THAN THE DIM OF
          THE REPR THEN WE CALL THE FREUDENTHAL PROCEDURE
           ELSE WE PRINT OUT THE RESULTS;

IF LAST<N THEN FREUDENTHAL(LAST)
        ELSE
BEGIN
      FOR I:=1 UNTIL LAST DO
      WEIGHTS(I,R+1):=1;OUTPUTR(WEIGHTS,N,1)
END;
END
END.
```

IV Program B4(2)

```
BEGIN COMMENT CALCULATION OF THE WEIGHTS,ALGORITHM 2,PROGRAM B4(2);
INTEGER N,R,N1,N2,LON,SOR;READON(N,R,N1,N2,LON,SOR);I_W:=1;S_W:=1;
WRITE("THE WEIGHTS OF THE REPR DIM=",N,"ARE.");
BEGIN
INTEGER ARRAY ROT(1::N1,1::R);LONG REAL ARRAY X(1::R+1);
INTEGER ARRAY CAR(1::R,1::R);LONG REAL ARRAY T(-1::R);
LONG REAL ARRAY SCALAR,INVCAR(1::R,1::R);
LONG REAL ARRAY WEIGHTS(1:: N,1::R+2);
LONG REAL ARRAY GH,D,K(1::R);INTEGER ARRAY N3(1::R);
INTEGER ARRAY L(1::1,1::R);INTEGER ARRAY M,NROT(1::R);
INTEGER S1,LAST,W,B,A,Q,NUM,Y;LONG REAL I1,S;
INTEGER LEVEL;LONG REAL I2,F,Z;

PROCEDURE INPUTI(INTEGER ARRAY A(*,*); INTEGER VALUE M,L);
BEGIN
      FOR I:=1 UNTIL M DO
      FOR J:=1 UNTIL L DO
      READON(A(I,J))
END;

PROCEDURE INPUTR(LONG REAL ARRAY A(*,*);INTEGER VALUE M,L);
BEGIN
      FOR I:=1 UNTIL M DO
      FOR J:=1 UNTIL L DO
      READON(A(I,J))
END;

PROCEDURE SCA;
BEGIN
      INTEGER X;
      FOR I:=1 UNTIL R DO
      FOR J:=1 UNTIL R DO
      BEGIN
            IF J\=R THEN X:=LON ELSE X:=SOR;
            SCALAR(I,J):=IF I\=J THEN (CAR(I,J)*X)/2 ELSE X
      END
END;
```

```
PROCEDURE OUTPUTR(LONG REAL ARRAY A(*,*);INTEGER VALUE M,L,F);
BEGIN
      FOR I:=F UNTIL M DO
      BEGIN
            IOCONTROL(2);
            FOR J:=1 UNTIL L-1 DO
            BEGIN
            R_W:=9;S_W:=1;R_FORMAT:="A";R_D:=5;WRITEON(A(I,J))
            END;
            I_W:=1;S_W:=1;WRITEON("MULT=",TRUNCATE(A(I,L)))
      END
END;

PROCEDURE OUTPUTI(INTEGER ARRAY A(*,*); INTEGER VALUE M,L);
BEGIN
      FOR I:=1 UNTIL M DO
      BEGIN
            IOCONTROL(2);
            FOR J:=1 UNTIL L DO
            BEGIN
                  I_W:=2;S_W:=1;WRITEON(A(I,J))
            END;
            WRITE(" ")
      END
END;

PROCEDURE LHS (LONG REAL VALUE RESULT F);
BEGIN
      FOR I:=1 UNTIL R DO
      FOR J:=1 UNTIL R DO
      F:=F+(GH(I)-M(I)+D(I))*(GH(J)-M(J)+D(J))*SCALAR(I,J)
END;


COMMENT 1 BASIC INPUT;
FOR I:=1 UNTIL R DO READON(N3(I));
INPUTI(ROT,R,R);INPUTI(CAR,R,R);
INPUTR(INVCAR,R,R);INPUTI(L,1,R);

COMMENT 2 CALCULATION OF THE ROOTS;
COMMENT CALL PROGRAM A1(1);
---------------------------


FOR I:=1 UNTIL R DO
BEGIN
      GH(I):=0;
      FOR J:=1 UNTIL R DO
      GH(I):=GH(I)+INVCAR(J,I)*L(1,J)
END;
FOR I:=1 UNTIL R DO
```

```
BEGIN
     D(I):=0;
     FOR J:=1 UNTIL N1 DO
     D(I):=D(I)+ROT(J,I);
     D(I):=D(I)/2
END;
COMMENT 3 APPLICATION OF THE FREUDENTHAL'S FORMULA;
SCA;I1:=0;FOR F:=1 UNTIL R DO M(F):=0;
LHS(I1);LAST:=1;NUM:=1;WRITE("I1=",I1);
FOR I:=1 UNTIL R DO
WEIGHTS(LAST,I):=GH(I);WEIGHTS(LAST,R+1):=1;WEIGHTS(LAST,R+2):=0;
OUTPUTR(WEIGHTS,LAST,R+1,LAST);
COMMENT 4 GENERATION OF THE INTEGERS Q1,Q2,...,QL;
FOR I:=1 UNTIL R-1 DO
M(I):=N2;M(R):=N2-1;Q:=R;
WHILE 1=1 DO
BEGIN
     IF M(Q)=N3(Q) THEN
     WHILE M(Q)=N3(Q) DO
     IF Q>1 THEN Q:=Q-1
     ELSE GOTO EXIT;
     M(Q):=M(Q)+1;
     FOR I:=Q+1 UNTIL R DO
     M(I):=N2;Q:=R;
          I2;=0;F:=0;LHS(I2);WRITE("I2=",I2,"M=");
          LEVEL:=0;
          FOR I:=1 UNTIL R DO
          BEGIN
               X(I):=GH(I)-M(I);LEVEL:=LEVEL+M(I);WRITEON(M(I))
          END;
          X(R+1):=LEVEL;WRITEON("LEVEL=",LEVEL);
          IF I1<=I2 THEN GOTO LOP;
         IF I1>I2 THEN
          BEGIN
               COMMENT 5 WE CALULATE THE RIGHT HAND SIDE OF
               THE FREUDENTHAL'S FORMULA;
                FOR I:=1 UNTIL LAST DO
                BEGIN
                     IF WEIGHTS(I,R+2)<X(R+1) THEN
                     BEGIN
                          FOR C:=1 UNTIL N1 DO
                          BEGIN
                               Y:=0;
                               FOR J:=1 UNTIL R DO
                               BEGIN
                                    IF ROT(C,J)\=0 THEN
                                    BEGIN
                                         K(J):=(WEIGHTS(I,J)-X(J))/
                                              ROT(C,J);
                                         Y:=Y+1;T(Y):=K(J)
                                    END
                                                   ELSE
                                    IF X(J)\=WEIGHTS(I,J) THEN GOTO
                                    T(0):=T(-1):=T(Y);
                                    IF T(Y)\=T(Y-1) THEN GOTO U
                               END;
```

```
                                        S:=0;
                                        FOR V:=1 UNTIL R DO
                                        FOR G:=1 UNTIL R DO
                                        S:=S+ROT(C,G)*(X(V)+T(Y)*ROT(C,V))
                                               *SCALAR(V,G);
                                        F:=F+WEIGHTS(I,R+1)*S;GOTO U1;
                              U:END;
                         END;
                    U1:END;
                    Z:=2*F/(I1-I2);IF Z<0.001 THEN Z:=0;
                    IF Z>0 THEN
                    BEGIN
                         LAST:=LAST+1;
                         FOR I:=1 UNTIL R DO
                         WEIGHTS(LAST,I):=X(I);
                         WEIGHTS(LAST,R+1):=Z;WEIGHTS(LAST,R+2):=LEVEL;
                         OUTPUTR(WEIGHTS,LAST,R+1,LAST);
                         NUM:=NUM+ENTIER(Z);
                         IF NUM=N             THEN GOTO EXIT
                    END
               END;
LOP:END;
EXIT:END
END.
```

§C.3 Group C

I Program C1(3)

```
BEGIN COMMENT THE WEYL GROUP OF THE ALGEBRA AL,PROGRAM C1(3);
INTEGER R,N,M;I_W:=1;S_W:=1;READON(R,N,M);
WRITE("THE WEYL GROUP OF ORDER",M,"OF THE ALGEBRA A",R,"IS.");
WRITE("IN OUTPUT WE GET IN THE FIRST LINE THE WEYL GROUP IN THE NOT");
WRITEON("ATION S=SAJ*SAI*...*SAL;IN THE SECOND LINE THE TRANSFORMED ");
WRITEON("SIMPLE ROOTS.");
WRITE("************************************************************");
BEGIN
INTEGER ARRAY WEYL_GROUP(1::M,1::N);
INTEGER ARRAY CAP(1::R,1::R);
INTEGER ARRAY ROT(1::2*N,1::R);
INTEGER ARRAY TRANS(1::M,1::R,1::R);
INTEGER ARRAY NARRAY(1::R,1::R);
INTEGER ARRAY A(1::R);
INTEGER LAST,NUMBER,LIMIT,C,B;
```

```
PROCEDURE INPUTI(INTEGER ARRAY A(*,*);INTEGER VALUE B,C);
BEGIN
      FOR I:=1 UNTIL B DO
      FOR J:=1 UNTIL C DO
      READON(A(I,J))
END;

PROCEDURE OUTPUTI(INTEGER ARRAY A(*,*,*);INTEGER VALUE B,C,D,F);
BEGIN
      FOR I:=F UNTIL B DO
      BEGIN
            IOCONTROL(2);
            FOR J:=1 UNTIL C DO
            BEGIN
                  FOR K:=1 UNTIL D DO
                  WRITEON(A(I,J,K));
                  WRITEON("____")
            END;
            WRITE(" ")
      END
END;

PROCEDURE OUTPUT(INTEGER ARRAY A(*,*);INTEGER VALUE B,C,D);
BEGIN
      FOR I:=D UNTIL B DO
      BEGIN
            IOCONTROL(2);
            FOR J:=1 UNTIL C DO
            WRITEON(A(I,J));
            WRITE(" ");
      END
END;
E_W:=1;S_W:=1;


INPUTI(WEYL_GROUP,R+1,N);
INPUTI(ROT N,R);
INPUTI(CAR,R,R);
FOR I:=1 UNTIL N DO
FOR J:=1 UNTIL R DO
ROT(N+I,J):=-ROT(N-I+1,J);


COMMENT 1 THE EFFECT OF E ON THE SIMPLE ROOTS(LEVEL 0);
FOR I:=1 UNTIL R DO
FOR J:=1 UNTIL R DO
IF J=I THEN TRANS(1,I,J):=1
       ELSE TRANS(1,I,J):=0;
OUTPUT(WEYL_GROUP,1,N,1);
OUTPUTI(TRANS,1,R,R,1);
WRITE("______________________________________________");
```

```
COMMENT 2 THE WEYL GROUP AND THE TRANS. PROP. OF THE SIMPLE
          ROOTS IN LEVEL 1;
LAST:=1;
FOR I:=1 UNTIL R DO
BEGIN
     FOR J:=1 UNTIL R DO
     BEGIN FOR K:=1 UNTIL R DO
           TRANS(I+1,J,K):=ROT(J,K)-CAR(I,J)*ROT(I,K)
     END;
     OUTPUT(WEYL_GROUP,I+1,N,I+1);
     OUTPUTI(TRANS,I+1,R,R,I+1);
     WRITE("______________________________________________");
END;
COMMENT 3 THE WEYL GROUP IN EACH LEVEL;

NUMBER:=R;LIMIT:=0;LAST:=R+1;
WHILE NUMBER>0 DO
BEGIN
     FOR I:=1 UNTIL NUMBER DO
     BEGIN
          FOR J:=1 UNTIL R DO
          BEGIN
               IF J\=WEYL_GROUP((LAST-NUMBER)+I,1) THEN
               BEGIN
                    FOR X:=1 UNTIL R DO NARRAY(1,X):=0;
                    FOR K:=1 UNTIL R DO
                    BEGIN
                         C:=TRANS((LAST-NUMBER)+I,1,K);
                         IF C\=0 THEN
                         FOR F:=1 UNTIL R DO
                         NARRAY(1,F):=NARRAY(1,F)+C*TRANS(J+1,K,F)
                    END;
                COMMENT 4 WE TEST IF THE NARRAY IS ONE OF THE
                          NON ZERO ROOTS;
                    FOR X:=1 UNTIL 2*N DO
                    BEGIN
                         FOR Y:=1 UNTIL R DO
                         IF (ROT(X,Y)-NARRAY(1,Y))\=0 THEN GOTO L1;
                         FOR Y:=2 UNTIL R DO
                         BEGIN
                         FOR W:=1 UNTIL R DO NARRAY(Y,W):=0;
                         FOR K:=1 UNTIL R DO
                         BEGIN
                              C:=TRANS((LAST-NUMBER)+I,Y,K);
                              IF C\=0 THEN
                              FOR F:=1 UNTIL R DO
                              NARRAY(Y,F):=NARRAY(Y,F)+C*TRANS(J+1,K,F
                         END
                    END;
                         COMMENT 5 WE TEST IF THE NEW ELEMENT OF
                                   THE WEYL GROUP HAS ALREADY BEEN
                                   CALCULATED;
                              FOR Z:=1 UNTIL LAST+LIMIT DO
                              BEGIN
                              FOR F:=1 UNTIL R DO
```

```
                              FOR Y:=1  UNTIL R DO
                              IF (TRANS(Z,F,Y)-NARRAY(F,Y))\=0
                                       THEN GOTO L2;
                              GOTO L;
                          L2: END;
                     LIMIT:=LIMIT+1;
                          WEYL_GROUP(LAST+LIMIT,1):=J;
                          FOR K:=1 UNTIL N-1 DO WEYL_GROUP(LAST+LIMIT,
                            K+1):=WEYL_GROUP((LAST-NUMBER)+I,K);
                        OUTPUT(WEYL_GROUP,LAST+LIMIT,N,LAST+LIMIT);
                                 FOR F:=1 UNTIL R DO
                                 FOR W:=1 UNTIL R DO
                                 TRANS(LAST+LIMIT,F,W):=NARRAY(F,W);
                 OUTPUTI(TRANS,LAST+LIMIT,R,R,LAST+LIMIT);
WRITE("--------------------------------------------------------------");
                                 GOTO L;
                    L1:END
                    END;
            L:END
END;
    NUMBER:=LIMIT; LAST:=LAST+LIMIT;
    LIMIT:=0
END;
COMMENT 6 THE PROGRAM ENDS WHEN WE REACH LEVEL R+1;
END
END.
```

II Program C2(3)

```
BEGIN COMMENT KOSTANT-STEINBERG FORMULA,PROGRAM C2(3);
INTEGER R,N,M,N3,N4;I_W:=1;S_W:=1;
READON(R,N,M,N3,N4);
BEGIN
REAL ARRAY D,REFL,P(1::R);
REAL ARRAY INVCAR,H,W(1::R,1::R);
INTEGER ARRAY ROT(1::2*N,1::R);INTEGER ARRAY CAR,NWEIGHT(1::R,1::R);
INTEGER ARRAY WEYL_GROUP(1::M,1::N);
INTEGER ARRAY TENSOR(1::06,1::R+1);INTEGER ARRAY M1,K,WW(1::R);
INTEGER ARRAY ML(1::06);INTEGER X,O,O1;
INTEGER N1,N2,N,Z,LAST,S,B,P1,P2,LIMIT;REAL MM,Y,A;
INTEGER ARRAY TRANS(1::M,1::R,1::R);
INTEGER NUMBER,C,DIM,Q;INTEGER ARRAY NARRAY(1:4R,1::R);


PROCEDURE INPUTR(REAL ARRAY A(*,*);INTEGER VALUE N,L);
BEGIN
  FOR I:=1 UNTIL N DO
  BEGIN
  IOCONTROL(2);
  FOR J:=1 UNTIL L DO
  BEGIN
    READON(A(I,J));
    WRITEON(A(I,J))
```

```
   END;
   WRITE(" ")
 END
 END;


PROCEDURE INPUTI(INTEGER ARRAY A(*,*); INTEGER VALUE K,L);
BEGIN
FOR I:=1 UNTIL K DO
BEGIN
IOCONTROL(2);
FOR J:=1 UNTIL L DO
BEGIN
READON(A(I,J));
WRITEON(A(I,J))
END;
WRITE(" ")
END
END;


PROCEDURE OUTPUT (INTEGER ARRAY A(*,*); INTEGER VALUE N,L,F);
BEGIN
  FOR I:=F UNTIL N DO
  BEGIN
    IOCONTROL(2);
    FOR J:=1 UNTIL L DO
    BEGIN
      WRITEON(A(I,J))
    END;
  WRITE(" ")
  END
END;


PROCEDURE REFLECTION(REAL ARRAY A(*,*);INTEGER VALUE L,Q);
BEGIN
      FOR I:=1 UNTIL R DO
      FOR J:=1 UNTIL R DO
      REFL(J):=REFL(J)+A(Q,I)*TRANS(L,I,J)
END;


PROCEDURE DELTA;
BEGIN
      FOR I:=1 UNTIL R DO
      BEGIN
           D(I):=0;
           FOR J:=1 UNTIL N DO
            D(I):=D(I)+ROT(J,I);
            D(I):=D(I)/2
      END
END;
```

```
PROCEDURE SUM;
BEGIN
      FOR I:=1 UNTIL R DO
      FOR J:=1 UNTIL R DO
      W(I,J):=W(I,J)+D(J)
END;

PROCEDURE PARTITION(REAL ARRAY B(*);REAL VALUE Y;INTEGER
                                          VALUE RESULT X);
BEGIN
  INTEGER ARRAY S,M(1::N   );INTEGER L,Q;
  FOR I:=1 UNTIL N-1 DO M(I):=0;
  M(N):=-1;Q:=N;
  WHILE 1=1 DO
  BEGIN
       IF M(Q)=R+1 THEN
       WHILE M(Q)=R+1 DO
       IF Q>1 THEN Q:=Q-1
               ELSE GOTO EXIT;
       M(Q):=M(Q)+1;
       FOR I:=Q+1 UNTIL N DO
       M(I):=0;Q:=N;
    L:=0;
    FOR I:=1 UNTIL N DO
    BEGIN
      S(I):=0;
      FOR J:=1 UNTIL R DO
        S(I):=S(I)+M(I)*ROT(I,J);
      L:=L+S(I)
    END;
    IF L=Y THEN
    BEGIN
      INTEGER ARRAY D(1::R);
      FOR I:=1 UNTIL R DO
      BEGIN
        D(I):=0;
        FOR J:=1 UNTIL N DO
        D(I):=D(I)+M(J)*ROT(J,I)
      END;
      FOR F:=1 UNTIL R DO
      IF D(F)\=B(F) THEN GOTO L1;X:=X+1;
    END;
L1:END;
EXIT:END;

COMMENT 1 BASIC INPUT OF THE PROGRAM ;
FOR I:=1 UNTIL 3 DO
FOR J:=1 UNTIL R DO
READON(ROT(I,J));
WRITE("CAR");INPUTI(CAR,R,R);
WRITE("INVCAR");INPUTR(INVCAR,R,R);
FOR I:=1 UNTIL 3 DO
```

```
FOR J:=1 UNTIL R DO
ROT(3+I,J):=-ROT(3-I+1,J);
INPUTR(H,R,R,DELTA;SUM;
WRITE("WEYL GROUP");
INPUTI(WEYL_GROUP,R+1,3);
FOR I:=1 UNTIL R DO READON(WW(I));
COMMENT 2 GENERATION OF THE WEYL GROUP;
COMMENT CALL PROGRAM C1(3);
----------------------------------------
COMMENT 3 FINDING THE NONEGATIVE INTEGERS EXPRESSING THE ARGUMENT
        AS LINEAR COMBINATION OF POSITIVE ROOTS;
READON(N1,N2);
N:=N1*N2;Z:=0;
LAST:=0;
FOR I:=1 UNTIL R-1 DO M1(I):=N3;
M1(R):=N3-1;Q:=R;
WHILE 1=1 DO
BEGIN
  IF M1(Q)=N4 THEN
  WHILE M1(Q)=N4 DO
  IF Q>1 THEN Q:=Q-1
        ELSE GOTO EXIT;
  M1(Q):=M1(Q)+1;
  FOR I:=Q+1 UNTIL R DO M1(I):=N3;Q:=R;
  FOR I:=1 UNTIL M DO
  FOR J:=1 UNTIL M DO
  BEGIN
      FOR F:=1 UNTIL R DO REFL(F):=0;
          REFLECTION(W,I,1);REFLECTION(W,J,2);
                FOR K:=1 UNTIL R DO
                BEGIN
                    REAL A;
                    A:=0;FOR F:=1 UNTIL R DO
                    A:=A-M1(F)*INVCAR(F,K);
        P(K):=REFL(K)-A-2*D(K);
        IF P(K)<0 THEN GOTO U;
        IF P(K)-TRUNCATE(P(K))<0.01 THEN P(K):=TRUNCATE(P(K));
         IF P(K)-TRUNCATE(P(K))>0.96 THEN P(K):=TRUNCATE(P(K))+1;
         IF P(K)-TRUNCATE(P(K))>0.01 AND P(K)-TRUNCATE(P(K))<0.96
                                                THEN  P(K):=-1;
         IF P(K)<0 THEN GOTO U
      END;

COMMENT 4  DIM OF THE REP D(M1,M2);
COMMENT CALL PROGRAM B1(2);
----------------------------------------
FOR K:=1 UNTIL LAST DO
BEGIN
     S:=0;
     FOR N:=1 UNTIL R DO
     BEGIN
     B:=TENSOR(K,N)-M1(N);
     S:=S+ABS(B)
END;
IF S=0 THEN GOTO U
END;
```

```
LAST:=LAST+1;
FOR F:=1 UNTIL R DO
TENSOR(LAST,F):=M1(F);TENSOR(LAST,R+1):=ENTIER(MM);ML(LAST):=0;
COMMENT 5  FINDING THE MULT OF D(M1,M2);
FOR T:=1 UNTIL M DO
FOR E:=1 UNTIL M DO
BEGIN
     FOR F:=1 UNTIL R DO REFL(F):=0;
     REFLECTION(W,T,1);REFLECTION(W,E,2);
          FOR K:=1 UNTIL R DO
          BEGIN
               REAL A;
               A:=0;FOR F:=1 UNTIL R DO
               A:=A-TENSOR(LAST,F)*INVCAR(F,K);
               P(K):=REFL(K)+A-2*D(K);
          IF P(K)<0 THEN GOTO V;
           IF P(K)-TRUNCATE(P(K))<0.01 THEN P(K):=TRUNCATE(P(K));
           IF P(K)-TRUNCATE(P(K))>0.96 THEN P(K):=TRUNCATE(P(K))+1;
           IF P(K)-TRUNCATE(P(K))>0.01 AND P(K)-TRUNCATE(P(K))<0.96
                                         THEN P(K):=-1;
          IF P(K)<0 THEN GOTO V
   END;
   Y:=P(1)+P(2);
  X:=0;
   PARTITION(P,Y,X);
   IF Y=0 THEN X:=1;
   IF T=1 THEN O:=1 ELSE O:=-1;
   IF E=1 THEN O1:=1 ELSE O1:=-1;
   ML(LAST):=ML(LAST)+O*O1*X;
V:END;
IF ML(LAST)\=1 THEN
BEGIN
    FOR N:=LAST UNTIL LAST+ML(LAST)-1 DO
    BEGIN
         FOR F:=1 UNTIL 2 DO
         TENSOR(N,F):=M1(F);
         TENSOR(N,3):=TRUNCATE(MM)
    END
END;

COMMENT  6 TEST   IF THE DIM IS EQUAL TO N;
Z:=Z+TRUNCATE(MM)*ML(LAST);
IF ABS(N-Z)=0  THEN GOTO EXIT;
IF ML(LAST)\=1 THEN
LAST:=LAST+ML(LAST)-1;
  U:END
END;
EXIT:WRITE("ANALYSIS OF TENSOR PRODUCT");
OUTPUT(TENSOR,6,3,1)
END
END.
```

III Programs C3(3), C4(3), C5(3), C6(3), C7(3)

```
BEGIN COMMENT CALCULATION OF THE TENSOR PRODUCTS OF CLASSICAL AND
      EXCEPTIONAL ALGEBRAS USING HIGHER ORDER INDICES;
      COMMENT THIS PROGRAM INCLUDES THE EXPRESSIONS OF THE I(2)
      AND I(4) INDICES OF THE ALGEBRAS:
      AL(PROGRAM C3(3)),
      BL(PROGRAM C4(3)),
      CL(PROGRAM C5(3)),
      DL(PROGRAM C6(3)),
      EXCEPTIONAL ALGEBRAS PROGRAM C7(3));
INTEGER N,R,N1,N2,N3; READON(R,N1,N2,N3,N);
BEGIN
INTEGER ARRAY ROT(1:: N,1::R);LONG REAL ARRAY L(1::N1,1::R+1);
INTEGER ARRAY CAR(1::R,1::R);LONG REAL ARRAY INVCAR(1::R,1::R);
LONG REAL ARRAY L1,L2,L3(1::R+1);
INTEGER ARRAY NROT,W,K,M1(1::R);
LONG REAL ARRAY GH(1::R);
INTEGER S,LAST,P,B,A1,A,Y,P1,P2,Q,T;
LONG REAL K1,V,P2L2,P2L3,M,I2,I4,P1L2,P1L3,P3L2,P3L3,P4L2,P4L3;
LONG REAL ARRAY D(1::R);


PROCEDURE INPUTI(INTEGER ARRAY A(*,*); INTEGER VALUE M,L);
BEGIN
     FOR I:=1 UNTIL M DO
     BEGIN
          IOCONTROL(2);
          FOR J:=1 UNTIL L DO
          BEGIN
               I_W:=2;S_W:=2;
               READON(A(I,J));
               WRITEON(A(I,J))
          END
          WRITE(" ")
     END;
END;


PROCEDURE INPUTR(LONG REAL ARRAY A(*,*);INTEGER VALUE M,L);
BEGIN
     FOR I:=1 UNTIL M DO
     BEGIN
          IOCONTROL(2);
          FOR J:=1 UNTIL L DO
          BEGIN
               R_W:=8;S_W:=1;R_FORMAT:="A";R_D:=5;
               READON(A(I,J));
               WRITEON(I,J))
          END;
          WRITE(" ")
     END
END;
```

```
PROCEDURE OUTPUT(LONG REAL ARRAY A(*,*);INTEGER VALUE M,L,B);
BEGIN
  FOR I:=B UNTIL M DO
  BEGIN
    IOCONTROL(2);
    FOR J:=1 UNTIL L-1 DO
    BEGIN
     I_W:=2;S_W:=1;
     WRITEON(TRUNCATE(A(I,J)))
    END;
    R_W:=14;S_W:=1;R_FORMAT:="F";WRITEON("N=",A(I,L))
  END
END;


PROCEDURE OUTPUTI(INTEGER ARRAY A(*,*);INTEGER VALUE M,L);
BEGIN
      FOR I:=1 UNTIL M DO
      BEGIN
            IOCONTROL(2);
            FOR J:=1 UNTIL L DO
            WRITEON(A(I,J));
            WRITE(" ")
      END
END;


PROCEDURE PRODUCT(INTEGER ARRAY A(*,*);INTEGER ARRAY B,C(*);
                 INTEGER VALUE   M,I;        INTEGER VALUE RESULT P);
BEGIN
  FOR J:=1 UNTIL M DO
  P:=P+(C(J)+1)*B(J)*A(I,J)
END;


PROCEDURE DELTA(LONG REAL ARRAY A(*);INTEGER ARRAY B(*,*);
                          INTEGER VALUE L,M);
BEGIN
  FOR I:=1 UNTIL L DO
  BEGIN
    A(I):=0;
    FOR J:=1 UNTIL M DO
    A(I):=A(I)+B(J,I);
    A(I):=A(I)/2
  END
END;


PROCEDURE LABEL(LONG REAL ARRAY A,B(*);INTEGER VALUE L);
BEGIN
  INTEGER J1;
  FOR I:=1 UNTIL L   DO
  BEGIN
        J1:=I;B(I):=0;
        FOR J:=J1 UNTIL L DO
        B(I):=B(I)+A(J);
        B(I):=B(I)+L-I+1
  END;B(L+1):=0
END;
```

```
PROCEDURE LABEL2(LONG REAL ARRAY A(*);INTEGER VALUE L);
BEGIN
      FOR I:=1 UNTIL L DO
      A(I):=L-I+1;A(L+1):=0
END;

PROCEDURE INDEX(LONG REAL ARRAY A,B(*);INTEGER VALUE L);
BEGIN
      FOR I:=1 UNTIL L DO
      BEGIN
            B(I):=0;
            FOR J:=1 UNTIL L DO
            B(I):=B(I)+A(J)*CAR(J,I)
      END
END;

PROCEDURE HWEIGHT(LONG REAL ARRAY A(*);LONG REAL ARRAY B(*,*);
 LONG REAL ARRAY C(*,*); INTEGER VALUE L,M);
BEGIN
   FOR I:=1 UNTIL L DO
   BEGIN
     A(I):=0;
     FOR J:=1 UNTIL L DO
     A(I):=A(I)+B(J,I)*C(M,J)
   END
END;

PROCEDURE PA1(LONG REAL ARRAY A(*);INTEGER VALUE L;
              LONG REAL VALUE RESULT M);
BEGIN
   FOR I:=1 UNTIL L+1 DO
   M:=M+A(I)
END;

PROCEDURE PA2(LONG REAL ARRAY A(*);INTEGER VALUE L;
                             LONG REAL VALUE RESULT M);
BEGIN
     FOR I:=1 UNTIL L+1 DO
     M:=M+A(I)**2;
     FOR I:=1 UNTIL L+1 DO
     FOR J:=1   UNTIL L+1 DO
     BEGIN
           IF I>J THEN M:=M+A(I)*A(J)
      END
END;

PROCEDURE PA3(LONG REAL ARRAY A(*);INTEGER VALUE L;
              LONG REAL VALUE RESULT M);
BEGIN
```

```
FOR I:=1 UNTIL L+1 DO
M:=M+A(I)**3;
FOR I:=1 UNTIL L+1 DO
FOR J:=1 UNTIL L+1 DO
BEGIN
     IF I\=J THEN M:=M+(A(I)**2)*A(J)
END;
FOR I:=1 UNTIL L+1 DO
FOR J:=1 UNTIL L+1 DO
FOR K:=1 UNTIL L+1 DO
BEGIN
     IF I>J AND J>K THEN
     M:=M+A(I)*A(J)*A(K)
END
END;


PROCEDURE PA4(LONG REAL ARRAY A(*);INTEGER VALUE L;
               LONG REAL VALUE RESULT M);
BEGIN
     FOR I:=1 UNTIL L+1 DO
     M:=M+A(I)**4;
     FOR I:=1 UNTIL L+1 DO
     FOR J:=1 UNTIL L+1 DO
     BEGIN
          IF I\=J THEN M:=M+(A(I)**3)*A(J)
     END;
     FOR I:=1 UNTIL L+1 DO
     FOR J:=1 UNTIL L+1 DO
     BEGIN
          IF I>J THEN M:=M+(A(I)**2)*(A(J)**2)
     END;
     FOR I:=1 UNTIL L+1 DO
     FOR J:=1 UNTIL L+1 DO
     FOR K:=1 UNTIL L+1 DO
     BEGIN
          IF I\=J AND I\=K AND J>K THEN
          M:=M+(A(I)**2)*A(J)*A(K)
     END;
     FOR I:=1 UNTIL L+1 DO
     FOR J:=1 UNTIL L+1 DO
     FOR K:=1 UNTIL L+1 DO
     FOR G:=1 UNTIL L+1 DO
     BEGIN
          IF I>J AND J>K AND K>G THEN
          M:=M+A(I)*A(J)*A(K)*A(G)
     END
END;


COMMENT SECOND AND FOURTH ORDER INDICES OF THE ALGEBRA AL;
---------------------------------------------------------
PROCEDURE FOURTHORDER(LONG REAL ARRAY A,B(*);INTEGER VALUE L;
                      LONG REAL VALUE M,C1,C2,C3,C4,D1,D2,D3,D4;
                      LONG REAL VALUE RESULT F);
```

```
BEGIN
      LONG REAL H,H1,H2,H3,H4;H:=0;
      FOR I:=1 UNTIL L+1 DO
      FOR J:=1 UNTIL L+1 DO
      BEGIN
            IF I<J THEN H:=H+(((A(I)-A(J))**2)-((B(I)-B(J))**2))
      END;
H1:= (L*((L+1)**2+7*(L+1)-6))/(((L+1)**2)*(L+2)*(L+3)*(L+4));
H2:=((L+1)**2+7*(L+1)-6)/(((L+1)**2)*(L+2)*(L+3));
H3:=1/((L+1)*(L+1));
H4:=(L-2)/(((L+1)**2)*(L+2));
F:=M*((C4-D4)*H1
 +(C4-C1*C3-D4+D1*D3)*H2
       +( 3*(C2**2)-3*(C1**2)*C2+(C1**4)-C4-3*(D2**2)+3*(D1**2)*D2
                                   -(D1**4)+D4)*H3
        +((C2**2)+C1*C3-C4-(C1**2)*C2-(D2**2)-D1*D3+D4
                          +(D1**2)*D2)*H4-(1/6)*H)
END;


PROCEDURE SECONDORDER(LONG REAL ARRAY A,B(*);INTEGER VALUE   L;
 LONG REAL VALUE M;  LONG REAL VALUE RESULT F);
BEGIN
      FOR I:=1 UNTIL L+1 DO
      FOR J:=1 UNTIL L+1 DO
      BEGIN
            IF I<J THEN F:=F+(((A(I)-A(J))**2)-((B(I)-B(J))**2))
      END;
      F:=(M/((L+1)*(L+2)))*F
END;

COMMENT SECOND AND FOURTH ORDER INDICES OF THE ALGEBRA CL;
---------------------------------------------------------
PROCEDURE SECONDORDER(LONG REAL ARRAY A,B(*);INTEGER VALUE L;
 LONG REAL VALUE M;   LONG REAL VALUE RESULT F);
BEGIN
      FOR I:=1 UNTIL L DO
      F:=F+((A(I)**2)-(B(I)**2));
      F:=(M/(2*(2*L+1)))*F
END;


PROCEDURE FOURTHORDER(LONG REAL ARRAY A,B(*);INTEGER VALUE L;
LONG REAL VALUE M,C,D;LONG REAL VALUE RESULT F);
BEGIN
LONG REAL G,H,P;
G:=H:=0;
FOR I:=1 UNTIL L DO
BEGIN
      G:=G+(B(I)**2);
      H:=H+((A(I)**2)-(B(I)**2))
END;
P:=0;
FOR I:=1 UNTIL L DO
FOR J:=1   UNTIL L DO
```

```
BEGIN
     IF I>J THEN
P:=P+(((A(I)*A(J))**2)-((B(I)*B(J))**2))
END;
  F:=M*((((L+5)*(C-D))/(4*(L+1)*(2*L+1)*(2*L+3)))
  +(P/(4*(2*L-1)*(2*L+1)))
  -(((L+2)*G*H)/((2*L)*((2*L+1)**2))))
END;


COMMENT IN THE CASE OF BL,THE SECOND AND FOURTH ORDER INDICES
        ARE THE INDICES OF THE ALGEBRA CL MULTIPLIED BY 4 AND 2
        RESPECTIVELY;

COMMENT THE SECOND AND FOURTH ORDER INDICES OF THE ALGEBRA DL;
--------------------------------------------------------------
PROCEDURE SECONDORDER(LONG REAL ARRAY A,B(*);INTEGER VALUE L;
 LONG REAL VALUE M;  LONG REAL VALUE RESULT F);
BEGIN
     FOR I:=1 UNTIL L DO
     F:=F+((A(I)+B(I))*(A(I)-B(I)));
     F:=(M/(2*L-1))*F
END;


PROCEDURE FOURTHORDER(LONG REAL ARRAY A,B(*);INTEGER VALUE L;
LONG REAL VALUE M,C,D;LONG REAL VALUE RESULT F);
BEGIN
LONG REAL G,H,P;
G:=H:=0;
FOR I:=1 UNTIL L DO
BEGIN
      G:=G+(B(I)**2);
      H:=H+((A(I)**2)-(B(I)**2))
END;
P:=0;
FOR I:=1 UNTIL L DO
FOR J:=1   UNTIL L DO
BEGIN
     IF I>J THEN
P:=P+(((A(I)*A(J))**2)-((B(I)*B(J))**2))
END;
  F:=M*((((L+5)*(C-D))/((L+1)*(2*L-1)*(2*L+1)))
  +(P/((2*L-1)*(2*L-3)))
  -((2*(L+2)*G*H)/(L*((2*L-1)**2))))
END;


COMMENT THE INDICES FOR THE EXCEPTIONAL ALGEBRAS;
-------------------------------------------------
PROCEDURE SECONDORDER(LONG REAL ARRAY A(*,*);LONG REAL ARRAY C,D(*);
                      LONG REAL VALUE F,M;INTEGER VALUE T;
                      LONG REAL VALUE RESULT K1,R1,B);
BEGIN
LONG REAL ARRAY E,X,Z(1::T);
FOR I:=1 UNTIL T DO
```

```
E(I):=C(I)+D(I);
FOR I:=1 UNTIL T DO
BEGIN
     X(I):=Z(I):=0;
     FOR J:=1 UNTIL T DO
     BEGIN
          X(I):=X(I)+E(J)*E(I)*A(I,J);
          Z(I):=Z(I)+D(I)*D(J)*A(I,J)
     END;
     K1:=K1+X(I);
   R1:=R1+Z(I)
END;
B:=(T*M*(K1-R1))/F
END;

PROCEDURE FOURTHORDER (LONG REAL VALUE  RESULT A;LONG REAL VALUE K1,
        R1,B;       LONG REAL VALUE M,F;INTEGER VALUE T);
BEGIN
LONG REAL K2,R2;
K2:=K1**2;R2:=R1**2;
A:=((T+2)*(B**2))/(M*T)-(M*(K2-R2)*F)/(120*R2)
END;
WRITE("SROT");INPUTI(ROT,R,R);
WRITE("CAR");INPUTI(CAR,R,R);
WRITE("INVCAR");INPUTR(INVCAR,R,R);

COMMENT   CALCULATION OF THE ROOTS;
COMMENT CALL PROGRAM A1(1);
-----------------------------------------
DELTA(D,ROT,R,N);
FOR I:=1 UNTIL R DO
BEGIN
  READON(W(I));
  WRITEON(W(I))
END;
WRITE("THE HIGHEST ORDER INDICES ARE");
LAST:=0;
FOR I:=1 UNTIL R-1 DO
M1(I):=N2;M1(R):=N2-1;Q:=R;
WHILE 1=1 DO
BEGIN
     IF M1(Q)=N3 THEN
     WHILE M1(Q)=N3 DO
     IF Q>1 THEN Q:=Q-1
     ELSE GOTO EXIT;
     M1(Q):=M1(Q)+1;
     FOR I:=Q+1 UNTIL R DO
     M1(I):=N2;Q:=R;
  M:=1;
  FOR I:=1 UNTIL  N DO
  BEGIN
  P1:=0;P2:=0;
  FOR I:=1 UNTIL R DO K(I):=M1(I);
```

```
   PRODUCT(ROT,W,K, R,I,P1);
   FOR I:=1 UNTIL R DO K(I):=0;
   PRODUCT(ROT,W,K,R,I,P2);
   M:=M*(P1/P2)
   END;
   BEGIN
   LAST:=LAST+1;
   IF LAST=N1+1 THEN GOTO EXIT;
   FOR I:=1 UNTIL R DO
   L(LAST,I):=M1(I);L(LAST,R+1):=M;
  OUTPUT(L,LAST,R+1,LAST);
   END;
 HWEIGHT(GH,INVCAR,L,R,LAST);
 INDEX(GH,L1,R);LABEL(L1,L2,R);LABEL2(L3,R);
 I2:=0;SECONDORDER(L2,L3,  R,L(LAST,R+1),I2);
 P1L2:=P1L3:=P2L2:=P2L3:=P3L2:=P3L3:=P4L2:=P4L3:=0;
 PA1(L2,R,P1L2);PA2(L2,R,P2L2);PA3(L2,R,P3L2);PA4(L2,R,P4L2);
 PA1(L3,R,P1L3);PA2(L3,R,P2L3);PA3(L3,R,P3L3);PA4(L3,R,P4L3);
 I4:=0;
 FOURTHORDER(L2,L3,R,L(LAST,R+1),P1L2,P2L2,P3L2,P4L2,P1L3,
                                              P2L3,P3L3,P4L3,I4);
 R_W:=14;S_W:=1;R_FORMAT:="F";WRITEON("I2=",I2,"I4=",I4)
 END;
 EXIT:END
 END.
```

§C.4 Group D

I Program DI(4)

```
BEGIN COMMENT CALCULATION OF THE MATRIX REPR OF A1,PROGRAM D1(4);
LONG REAL J,M,A;REAL N;INTEGER Y,I,N1;
READ(Y);J:=J/2;
U:N:=2*J+1;I:=0;N1:=TRUNCATE(N);
BEGIN
LONG REAL ARRAY MATRIX(1::N1,1::N1);
M:=J;
FOR F:=1 UNTIL N1 DO
FOR G:=1 UNTIL N1 DO
MATRIX(F,G):=0;
WHILE ABS(M)<=J DO
BEGIN
     A:=LONGSQRT((J-M)*(J+M+1));
     IF M\=J THEN
     BEGIN
          I:=I+1;MATRIX(I,I+1):=A
     END;
     M:=M-1
END;
WRITE("MATRIX REPR WITH J=",J);WRITE(" ");
FOR F:=1 UNTIL N1 DO
```

```
BEGIN
     IOCONTROL(2);
     FOR G:=1 UNTIL N1 DO
     BEGIN
          R_W:=7;S_W:=1;R_FORMAT:="A";R_D:=4;
          WRITEON(MATRIX(F,G))
     END;
     WRITE(" ");
END
END;
J:=J+1/2;IF J<=Y THEN GOTO U
END.
```

II Program D2(4)

```
BEGIN COMMENT CALCULATION OF DIAGONAL GENERATORS,PROGRAM D2(4);
INTEGER N,R,N1,LON,SOR;READON(N,R,N1,LON,SOR);I_W:=1;S_W:=2;
WRITE("WE ARE CALCULATING THE DIAG. GEN. OF  REPR. DIM=",N );
WRITE("THE RANK OF THE ALGEBRA IS=",R,"THE NUMBER OF POSITIVE ROOTS"
WRITE(" IS=",N1,"THE LEN. OF THE ROOTS IS=",LON,",",SOR);
BEGIN
INTEGER ARRAY ROT(1::N1,1::R);INTEGER ARRAY CAR(1::R,1::R);
LONG REAL ARRAY SCALAR,INVCAR(1::R,1::R);
LONG REAL ARRAY DIAG(1::1,1::N);
LONG REAL ARRAY WEIGHTS(1::N,1::R+2);
LONG REAL ARRAY GH,D,X,NWEIGHT(1::R);
INTEGER ARRAY L(1::1,1::R);INTEGER ARRAY M,NROT(1::R);
INTEGER S,LAST,B,W,A,Y,LEVEL,IN;
LONG REAL S1,S2,I1,I2,F,Z,P1;


PROCEDURE INPUTI(INTEGER ARRAY A(*,*); INTEGER VALUE M,L);
BEGIN
     FOR I:=1 UNTIL M DO
     FOR J:=1 UNTIL L DO
     READON(A(I,J))
END;


PROCEDURE INPUTR(LONG REAL ARRAY A(*,*);INTEGER VALUE M,L);
BEGIN
     FOR I:=1 UNTIL M DO
     FOR J:=1 UNTIL L DO
     READON(A(I,J))
END;


PROCEDURE SCA;
BEGIN
     INTEGER X;
     FOR I:=1 UNTIL R DO
     FOR J:=1 UNTIL R DO
     BEGIN
          IF J\=R THEN X:=LON ELSE X:=SOR;
          SCALAR(I,J):=IF I\=J THEN (CAR(I,J)*X)/2 ELSE X
     END
END;
```

```
PROCEDURE OUTPUTR(LONG REAL ARRAY A(*,*);INTEGER VALUE M,L);
BEGIN
  B:=0;
  FOR I:=1 UNTIL M DO
  BEGIN
       R_W:=8; R_FORMAT:="A";R_D:=5;S_W:=1;
       IOCONTROL(2);
    FOR J:=1 UNTIL L DO
    BEGIN
         B:=B+1;IF B<=10 THEN WRITEON(A(I,J))
                          ELSE WRITE(A(I,J));
         IF B=11 THEN B:=1
    END;
    WRITEON(")");
    WRITE("-----------------------------------------------------------------
  END
END;

PROCEDURE OUTPUTI(INTEGER ARRAY A(*,*); INTEGER VALUE M,L,C);
BEGIN
  FOR I:=C UNTIL M DO
  BEGIN
    IOCONTROL(2);
    FOR J:=1 UNTIL L DO
    WRITEON(A(I,J));
    WRITE(" ")
  END
END;

COMMENT 1 BASIC INPUT;
INPUTI(ROT,R,R);INPUTI(CAR,R,R);
INPUTR(INVCAR,R,R);INPUTI(L,1,R);

COMMENT 2 CALCULATION OF THE ROOTS;
COMMENT CALL PROGRAM A1(1);
-----------------------------

COMMENT 3 CALCULATION OF THE WEIGHTS;
COMMENT CALL PROGRAM B3(2);
-----------------------------

COMMENT 4 KNOWING THE WEIGHT SYSTEM WE CALCULATE THE
        DIAGONAL GENERATORS;
FOR I:=1 UNTIL R DO
BEGIN
     IN:=0;
     FOR F:=1 UNTIL LAST DO
     BEGIN
          S1:=0;
          FOR J:=1 UNTIL R DO
          BEGIN
               P1:=SCALAR(I,J)*WEIGHTS(F,J);
               S1:=S1+P1
          END;
```

```
            FOR K:=1 UNTIL  ROUND(WEIGHTS(F,R+1)) DO
            DIAG(1,IN+K):=S1;IN:=IN + ROUND(WEIGHTS(F,R+1))
       END;
       WRITE("THE DIAGONAL GENERATOR HA",I,"IS");
       WRITE("HA",I,"=DIAG(");
       OUTPUTR(DIAG,1,IN)
  END
  END
  END
```

III Program D3(4)

```
BEGIN COMMENT CALCULATION OF THE DIAG GENERATORS,PROGRAM D3(4);
INTEGER N,R,N1,N2;READON(N,R,N1,N2);
BEGIN
INTEGER ARRAY CAR(1::R,1::R);LONG REAL ARRAY SCALAR(1::R,1::R);
LONG REAL ARRAY WEIGHTS(1::N,1::R);
LONG REAL ARRAY DIAG(1::R,1::N);
LONG REAL P,S;


PROCEDURE INPUTI(INTEGER ARRAY A(*,*); INTEGER VALUE M,L);
BEGIN
FOR I:=1 UNTIL M DO
BEGIN
FOR J:=1 UNTIL L DO
READON(A(I,J));
END
END;


PROCEDURE INPUTR(LONG REAL ARRAY A(*,*);INTEGER VALUE M,L);
BEGIN
FOR I:=1 UNTIL M DO
BEGIN
FOR J:=1 UNTIL L DO
READON(A(I,J));
END
END;


PROCEDURE OUTPUT(LONG REAL ARRAY A(*,*);INTEGER VALUE M,L,B);
BEGIN
  FOR I:=B UNTIL M DO
  BEGIN
    IOCONTROL(2);
    I_W:=1;S_W:=1;
    WRITE("THE GENERETOR CORRESPONDING TO A",I,"IS.");
    WRITE("HA",I,"=DIAG(");
    FOR J:=1 UNTIL L DO
    BEGIN
    R_W:=7;S_W:=1;R_FORMAT:="A";R_D:=4;
      WRITEON(A(I,J))
    END;
    WRITEON(")");
  WRITE(" ")
  END
END;
```

```
PROCEDURE SCA(INTEGER VALUE F,T,L);
BEGIN
      INTEGER X;
      FOR I:=1 UNTIL L DO
      BEGIN
            FOR J:=1 UNTIL L DO
            BEGIN
                  IF J\=L THEN X:=F ELSE X:=T;
                  IF I\=J THEN SCALAR(I,J):=(CAR(I,J)*X)/2
                                                ELSE SCALAR(I,J):=X;
            END
      END
END;

PROCEDURE OUTPUTI(INTEGER ARRAY A(*,*);INTEGER VALUE M,L);
BEGIN
      FOR I:=1 UNTIL M DO
      BEGIN
            IOCONTROL(2);
            FOR J:=1 UNTIL L DO
            BEGIN
                  WRITEON(A(I,J))
            END;
            WRITE(" ")
      END
END;

INPUTI(CAR,R,R);INPUTR(WEIGHTS,N,R);SCA(N1,N2,R);

FOR I:=1 UNTIL R  DO
BEGIN
   FOR F:=1 UNTIL  N DO
   BEGIN
     S:=0;
     FOR J:=1 UNTIL R DO
     BEGIN
P:=SCALAR(I,J)*WEIGHTS(F,J);
       S:=S+P
     END;
     DIAG(I,F):=S
   END
END;
OUTPUT (DIAG,R,N,1);
END
END.
```

IV Program D4(4)

```
BEGIN COMMENT MATRIX REPR OF THE ALGEBRA G2,PROGRAM D4(4);
BEGIN
INTEGER R,N,M;READON(R,N,M);
BEGIN
INTEGER ARRAY T,GA(1::M);
INTEGER ARRAY ROT(1::R,1::R);
LONG REAL ARRAY E(1::R,1::N,1::3);
REAL ARRAY WEIGHTS(1::N,1::R);
INTEGER ARRAY FUNCTION(1::N);
INTEGER COUNT,FIRST,G,LAST,INDEX,P ,MULT;
LONG REAL A,B,D,CC;REAL AA;


PROCEDURE INPUTI(INTEGER ARRAY A(*,*); INTEGER VALUE M,L);
BEGIN
      FOR I:=1 UNTIL M DO
      BEGIN
            FOR J:=1 UNTIL L DO
            READON(A(I,J))
      END
END;


PROCEDURE OUTPUT(LONG REAL ARRAY A(*,*,*);INTEGER VALUE B,C,D,L);
BEGIN
  FOR I:=L UNTIL B DO
  BEGIN
       I_W:=1;S_W:=1;
       WRITE("THE GENERATOR EA",I,"IS");
       WRITE("********************");
       WRITE(" ");
       FOR J:=1 UNTIL C DO
       BEGIN
        IF ABS(A(I,J,3))<1'-2 THEN A(I,J,3):=0;
        IF A(I,J,3)\=0 THEN
        BEGIN
          I_W:=3;S_W:=2;
          FOR K:=1 UNTIL D-1 DO
          WRITEON(TRUNCATE(A(I,J,K)));
          R_W:=6;S_W:=2;R_FORMAT:="A";R_D:=2;
          WRITEON(A(I,J,3));WRITE(" ")
        END
      END
     END
END;


PROCEDURE INITIALIZE3R(LONG REAL ARRAY A(*,*,*);INTEGER VALUE B,C,I
BEGIN
      FOR I:=1 UNTIL B  DO
      FOR J:=1 UNTIL C  DO
      FOR K:=1 UNTIL D  DO
      A(I,J,K):=0
END;
```

```
PROCEDURE INITIALIZELI(INTEGER ARRAY A(*);INTEGER VALUE B);
BEGIN
      FOR I:=1 UNTIL B DO
      A(I):=0
END;


LONG REAL PROCEDURE LOWER(LONG REAL VALUE A,B);
BEGIN
      LONG REAL O;
      O:=SQRT((A-B)*(A+B+1));
      O
END;


PROCEDURE FILLING(INTEGER VALUE B,C,F;LONG REAL VALUE D;
                                       INTEGER VALUE RESULT A);
BEGIN
       IF D=0 THEN GOTO EXIT;
       A:=A+1;
      E(F,A,1):=B;E(F,A,2):=C;E(F,A,3):=D;
EXIT:END;


PROCEDURE VERTICAL(INTEGER VALUE A,D;INTEGER VALUE RESULT B,C);
BEGIN
      FOR I:=1 UNTIL N  DO
      BEGIN
            AA:=0;
            FOR K:=1 UNTIL R DO
       AA:=AA+ABS(WEIGHTS(A,K)-WEIGHTS(I,K)-ROT(D,K));
            IF AA=0 THEN
            BEGIN
                  FOR W:=1 UNTIL  N DO
                  IF FUNCTION(I)=W THEN GOTO L1;
                  T(C):=A;
                  C:=C+1;B:=1;
                  T(C):=I;FUNCTION(I):=I;
                  G:=1;GOTO L2
            END;
      L1:END;
      L2:IF B=0 THEN G:=0;
      FUNCTION(A) :=A;
      IF G\=0 THEN
      BEGIN
            B:=0;
            VERTICAL(T(C) ,D,B,C)
      END
END;


PROCEDURE HORIZONTAL(INTEGER VALUE A,X;INTEGER VALUE RESULT B,C,D);
BEGIN
     FOR I:=1 UNTIL  N DO
     BEGIN
           AA:=0;
           FOR K:=1 UNTIL R DO
```

```
      AA:=AA +ABS( WEIGHTS(A,K)-WEIGHTS(I,K)-ROT(X,K));
            IF AA=0 THEN
            BEGIN
                  B:=B+1;
                  GA(B):=I;
                  FUNCTION(I):=I;
                  C:=C+1;
                  T(C):=I
            END
      END;
      IF B>1 THEN D:=B-1;
      IF B=0 THEN GA(1):=0;
      FUNCTION(A):=A;
      IF GA(1)\=0 THEN
      BEGIN
            B:=0;
            HORIZONTAL(GA(1),X,B,C,D)
       END
END;


PROCEDURE INPUTR ( REAL  ARRAY A(*,*); INTEGER VALUE M,L);
BEGIN
      FOR I:=1 UNTIL M DO
      BEGIN
            FOR J:=1 UNTIL L DO
            READON(A(I,J))
      END
END;


PROCEDURE LOOP(INTEGER VALUE A,C,D;
                         INTEGER VALUE RESULT B;
                               LONG REAL VALUE RESULT G);
BEGIN
      FOR F:=1 UNTIL N DO
      IF E(A,F,2)=T(D) AND E(A,F,3)\=0 THEN
      BEGIN
            B:=1;
            FOR X:=1 UNTIL N DO
            BEGIN
                  IF E(A,X,2)=LAST THEN
                  BEGIN
                        FOR WW:=1 UNTIL INDEX DO
                        IF E(C,WW,1)=E(A,X,1)
                              AND E(C,WW,2)=E(A,F,1) THEN
                        BEGIN
                            CC:=E(A,X,3)*E(C,WW,3);
                            G:=CC/E(A,F,3)
                         END
                  END
            END
      END
END;
INPUTR(WEIGHTS, N,R);
INPUTI(ROT,R,R);

INITIALIZE1I(FUNCTION, N);
INITIALIZE3R(E,R,N,3);
```

```
LAST:=1;
WHILE LAST< N DO
BEGIN
     FOR W:=1 UNTIL  N DO
     IF FUNCTION(LAST)=W THEN GOTO L;
     COUNT:=0;FIRST:=1;
     VERTICAL(LAST,R,COUNT,FIRST);
     FOR W:=1 UNTIL FIRST-1 DO
     BEGIN
          E(R,T(W),  3):=LOWER((FIRST-1)/2,(FIRST-1)/2-W);
          E(R,T(W),  1):=T(W);
          E(R,T(W),  2):=T(W+1)
      END;
     L: LAST:=LAST+1
END;
FOR K:=R-1 STEP -1 UNTIL 1 DO
BEGIN
     LAST:=1;B:=0;INDEX:=0;INITIALIZE1I(FUNCTION,N);
     WHILE LAST<N DO
     BEGIN
          FOR W:=1 UNTIL N DO
          IF FUNCTION(LAST)=W THEN GOTO UU;
          COUNT:=FIRST:=MULT:=0;
          HORIZONTAL(LAST,K,COUNT,FIRST,MULT);
          IF FIRST=0 THEN GOTO UU;
          IF MULT=0 THEN
          BEGIN
               FOR W:=1 UNTIL FIRST DO
               BEGIN
                    A:=LOWER(FIRST/2,FIRST/2-W);
                    FILLING(LAST,T(FIRST),K,A,INDEX)
               END
          END
                    ELSE
          BEGIN
                P:=0;D:=0;
               FOR W:=1 UNTIL MULT DO
               BEGIN
                    FOR X:=R STEP -1 UNTIL 1 DO
                    BEGIN
                              A:=0;P:=0;
                         IF X\=K THEN LOOP(X,K,W,P,A);
                         IF P\=0 THEN
                         BEGIN
                              FILLING(LAST,T(W),K,A,INDEX);
                              IF FIRST>2 THEN
                                   FILLING(T(W),T(FIRST),K,A,INDEX
                                   D:=D+A**2;GOTO U1
                         END
                    END;
               U1:END;
               IF P=0 THEN
               BEGIN
                    FOR W:=1 UNTIL FIRST DO
                    IF FUNCTION(T(W))\=B THEN  FUNCTION(T(W)):=0;
          COUNT:=0;FIRST:=1;VERTICAL(LAST,K,COUNT,FIRST);
                    A:=LOWER(1/2,-1/2);
```

```
                    FILLING(T(FIRST-1),T(FIRST),K,A,INDEX);
                    B:=T(FIRST)
                END;
                IF P\=0 THEN
                BEGIN
                IF FIRST>2 THEN
   FILLING(LAST,T(FIRST-1),K,SQRT(FIRST-MULT -D),INDEX)
                          ELSE
   FILLING(LAST,T(FIRST),K,SQRT(FIRST-MULT-D),INDEX);
                    IF FIRST>2 THEN
 FILLING(T(FIRST-1),T(FIRST),K,SQRT(FIRST-MULT -D),INDEX)
             END
          END;
          UU:LAST:=LAST+1
      END
END;OUTPUT(E,R,N,3,1)
END.
END
END.
```

V Program D5(4)

```
BEGIN COMMENT MATRIX REPR OF THE 126  DIM REPR OF D5,PROGRAM D5(4);
BEGIN
INTEGER R,N,M;READON(R,N,M);
BEGIN
INTEGER ARRAY T,GA(1::M);
INTEGER ARRAY ROT(1::R,1::R);
LONG REAL ARRAY E(1::R,1::N,1::3);
REAL     ARRAY WEIGHTS(1::N,1::R);
INTEGER ARRAY FUNCTION(1::N);
INTEGER COUNT,FIRST,G,LAST,INDEX,P ,MULT,DIFF,DD;
LONG REAL   A,B,D,CC;
REAL AA;


PROCEDURE INPUTI(INTEGER ARRAY A(*,*);INTEGER VALUE M,L);
BEGIN
     FOR I:=1 UNTIL M DO
     BEGIN
          FOR J:=1 UNTIL L DO
          READON(A(I,J))
     END
END;


PROCEDURE OUTPUT;
BEGIN
     FOR I:=1 UNTIL R DO
     BEGIN
          I_W:=1;S_W:=1;WRITE("THE GENERATOR EA",I,"IS");
          WRITE("**********************");WRITE(" ");
          FOR J:=1 UNTIL N DO
          BEGIN
               IF E(I,J,3)<1'-2 THEN E(I,J,3):=0;
               IF E(I,J,3)\=0 THEN
```

```
                  BEGIN
                        I_W:=3;S_W:=2;
                        FOR K:=1 UNTIL 2 DO
                        WRITEON(TRUNCATE(E(I,J,K)));
                        R_W:=6;S_W:=2;R_FORMAT:="A";R_D:=2;
                        WRITEON(E(I,J,3));WRITE(" ")
                  END
            END
      END
END;

PROCEDURE INITIALIZE3R(LONG REAL ARRAY A(*,*,*);INTEGER VALUE B,C,D);
BEGIN
     FOR I:=1 UNTIL B  DO
     FOR J:=1 UNTIL C  DO
     FOR K:=1 UNTIL D  DO
     A(I,J,K):=0
END;

PROCEDURE INITIALIZE1I(INTEGER ARRAY A(*);INTEGER VALUE B);
BEGIN
     FOR I:=1 UNTIL B DO
     A(I):=0
END;

LONG REAL PROCEDURE LOWER(LONG REAL VALUE A,B);
BEGIN
     LONG REAL O;
     O:=SQRT((A-B)*(A+B+1));
     O
END;

PROCEDURE FILLING(INTEGER VALUE B,C,F;LONG REAL VALUE D;
                                  INTEGER VALUE RESULT A);
BEGIN
      IF D=0 THEN GOTO EXIT;
      A:=A+1;
     E(F,A,1):=B;E(F,A,2):=C;E(F,A,3):=D;
EXIT:END;

PROCEDURE VERTICAL(INTEGER VALUE A,D;INTEGER VALUE RESULT B,C);
BEGIN
     FOR I:=1 UNTIL N  DO
     BEGIN
          AA:=0;
          FOR K:=1 UNTIL R DO
      AA:=AA+ABS(WEIGHTS(A,K)-WEIGHTS(I,K)-ROT(D,K));
          IF AA=0 THEN
          BEGIN
               FOR W:=1 UNTIL  N DO
               IF FUNCTION(I)=W THEN GOTO L1;
```

```
                T(C):=A;
                C:=C+1;B:=1;
                T(C):=I;FUNCTION(I):=I;
                G:=1;GOTO L2
            END;
        L1:END;
        L2:IF B=0 THEN G:=0;
        FUNCTION(A) :=A;
        IF G\=0 THEN
        BEGIN
            B:=0;
            VERTICAL(T(C) ,D,B,C)
        END
END;

PROCEDURE HORIZONTAL(INTEGER VALUE A,X;INTEGER VALUE RESULT B,C,D);
BEGIN
        FOR I:=1 UNTIL  N DO
        BEGIN
            AA:=0;
            FOR K:=1 UNTIL R DO
      AA:=AA +ABS( WEIGHTS(A,K)-WEIGHTS(I,K)-ROT(X,K));
            IF AA=0 THEN
            BEGIN
                B:=B+1;
                GA(B):=I;
                FUNCTION(I):=I;
                C:=C+1;
                T(C):=I
            END
        END;
        IF B>1 THEN D:=B-1;
        IF B=0 THEN GA(1):=0;
        FUNCTION(A):=A;
        IF GA(1)\=0 THEN
        BEGIN
            B:=0;
            HORIZONTAL(GA(1),X,B,C,D)
        END
END;

PROCEDURE INPUTR(REAL ARRAY A(*,*);INTEGER VALUE M,L);
BEGIN
        FOR I:=1 UNTIL M DO
        BEGIN
            FOR J:=1 UNTIL L DO
            BEGIN
                READON(A(I,J))
            END
        END
END;
```

```
PROCEDURE LOOP(INTEGER VALUE A,C,D;
                        INTEGER VALUE RESULT B;
                              LONG REAL VALUE RESJ_T G);
BEGIN
     FOR F:=1 UNTIL N DO
     IF E(A,F,2)=T(D) AND E(A,F,3)\=0 THEN
     BEGIN
          B:=1;
          FOR X:=1 UNTIL N DO
          BEGIN
               IF E(A,X,2)=LAST THEN
               BEGIN
                    FOR WW:=1 UNTIL INDEX DO
                    IF E(C,WW,1)=E(A,X,1)
                          AND E(C,WW,2)=E(A,F,1) THEN
                    BEGIN
                        CC:=E(A,X,3)*E(C,WW,3);
                        G:=CC/E(A,F,3)
                     END
               END
          END
     END
END;


INPUTR(WEIGHTS,N,R);
INPUTI(ROT,R,R);
INITIALIZE1I(FUNCTION, N);
INITIALIZE3R(E,R,N,3);

LAST:=1;
WHILE LAST< N DO
BEGIN
     FOR W:=1 UNTIL  N DO
     IF FUNCTION(LAST)=W THEN GOTO L;
     COUNT:=0;FIRST:=1;
     VERTICAL(LAST,R,COUNT,FIRST);
     FOR W:=1 UNTIL FIRST-1 DO
     BEGIN
           E(R,T(W),  3):=LOWER((FIRST-1)/2,(FIRST-1)/2-W);
           E(R,T(W),  1):=T(W);
           E(R,T(W),  2):=T(W+1)
      END;
     L: LAST:=LAST+1
END;
FOR K:=R-1 STEP -1 UNTIL 1 DO
BEGIN
     LAST:=1;B:=0;INDEX:=0;INITIALIZE1I(FUNCTION,N);
     WHILE LAST<N DO
     BEGIN
          FOR W:=1 UNTIL N DO
          IF FUNCTION(LAST)=W THEN GOTO UU;
          COUNT:=FIRST:=MULT:=0;
          HORIZONTAL(LAST,K,COUNT,FIRST,MULT);
          DIFF:=FIRST-MULT;
          IF FIRST=0 THEN GOTO UU;
```

```
            IF MULT=0 THEN
            BEGIN
                 FOR W:=1 UNTIL FIRST DO
                 BEGIN
                      A:=LOWER(FIRST/2,FIRST/2-W);
                      IF W=1 THEN FILLING(LAST,T(W),K,A,INDEX)
                             ELSE FILLING(T(W-1),T(W),K,A,INDEX)
                 END
            END
                   ELSE
  BEGIN
  IF DIFF=2 THEN
  BEGIN
       D:=0;P:=0;
       FOR W:=1 UNTIL MULT DO
       BEGIN
            FOR X:=R STEP -1 UNTIL K+1 DO
            BEGIN
                 A:=0;P:=0; DD:=0;
                 LOOP(X,K,W,P,A);
                 FILLING(LAST,T(W),K,A,INDEX);
                 FILLING(T(W),T(FIRST),K,A,INDEX);
                 D:=D+A**2;
                  IF P=1 THEN GOTO U2
            END;
    U2:END;
       IF D=0 THEN
       BEGIN
            FILLING(LAST,T(MULT),K,SQRT(2),INDEX);
            FILLING(T(MULT),T(FIRST),K,SQRT(2),INDEX);
            D:=D+2
        END;
        FILLING(LAST,T(FIRST-1),K,SQRT(DIFF-D),INDEX);
        FILLING(T(FIRST-1),T(FIRST),K,SQRT(DIFF-D),INDEX)
  END;
  IF DIFF=1 THEN
  BEGIN
       D:=0;P:=0;
       FOR W:=1 UNTIL MULT DO
       BEGIN
            FOR X:=R STEP -1 UNTIL K+1 DO
            BEGIN
                 A:=0;P:=0;
                 LOOP(X,K,W,P,A);
                 FILLING(LAST,T(W),K,A,INDEX);
                 D:=D+A**2;
                 IF P=1 THEN GOTO U3
             END;
  U3:  END;
       IF P\=0 THEN
       FILLING(LAST,T(FIRST),K,SQRT(DIFF-D),INDEX);
       IF P=0 THEN
       BEGIN
            INTEGER FF;FF:=FIRST;
            FOR W:=1 UNTIL FIRST DO
```

```
          FUNCTION(T(W)):=0;
          COUNT:=0;FIRST:=1;
          VERTICAL(LAST,K,COUNT,FIRST);
          FILLING(T(1),T(2),K,1,INDEX);
          FOR W:=1 UNTIL FF-1 DO
          BEGIN
               COUNT:=0;FIRST:=1;
               VERTICAL(LAST+W,K,COUNT,FIRST);
               FILLING(T(w),T(2),K,1,INDEX)
          END
      END
 END;
          END;
          UU:LAST:=LAST+1
      END
 END;OUTPUT;
 END
 END
 END.
```

VI Program D6(4)

```
BEGIN COMMENT VERIFICATION OF THE COMMUTATION
      RELATIONS,PROGRAM DZ(4);
INTEGER N,R;READON(N,R);
BEGIN
INTEGER ARRAY T(1::R+1);
FOR I:=1 UNTIL R+1 DO READON(T(I));
BEGIN
LONG REAL ARRAY E(1::R+1,1::T(R+1),1::3);
LONG REAL ARRAY AUX1,AUX2(1::N*2,1::3);
LONG REAL ENTRY;



PROCEDURE INPUTR;
BEGIN
     FOR I:=1 UNTIL R DO
     FOR J:=1 UNTIL T(I) DO
     FOR K:=1 UNTIL 3 DO
     READON(E(I,J,K))
END;


PROCEDURE NEGATIVE(INTEGER VALUE A);
BEGIN
     FOR I:=1 UNTIL T(A) DO
     BEGIN
          FOR J:=1 UNTIL 2 DO
          E(R+1,I,J):=IF J=1 THEN E(A,I,2) ELSE E(A,I,1);
          E(R+1,I,3):=-E(A,I,3)
     END
END;
```

```
PROCEDURE MULTIPLICATION(LONG REAL ARRAY A(*,*);INTEGER VALUE B,C);
BEGIN
     INTEGER F,G,W;
     F:=1;G:=0;
     WHILE F<=T(B) DO
     BEGIN
          W:=0;
          FOR I:=F+1 UNTIL T(B) DO
          IF E(B,I,1)=E(B,F,1) THEN W:=W+1
                                 ELSE GOTO U;
        U:FOR J:=F UNTIL F+W DO
          BEGIN
               FOR K:=1 UNTIL T(C) DO
               BEGIN
                    IF E(C,K,1)=E(B,J,2) THEN
                    BEGIN
                              G:=G+1;A(G,1):=E(B,J,1);A(G,2):=E(C,K,2
                              A(G,3):=E(B,J,3)*E(C,K,3);
                         FOR M:=1 UNTIL G-1 DO
                         IF (A(M,1)=E(B,J,1) AND A(M,2)=E(C,K,2)) THE
                         BEGIN
                         A(M,3):=A(M,3)+A(G,3);
                         G:=G-1
                         END
                    END
               END
          END;
          F:=F+W+1
     END
END;


PROCEDURE ORDER(INTEGER VALUE A,B);
BEGIN
     INTEGER W,K,G;
     LONG REAL ARRAY W1,W2(1::3);
     W:=1;
     FOR I:=2 UNTIL T(B) DO
     IF E(A,I,1)>E(A,W,1) THEN W:=W+1
                               ELSE
     BEGIN
          FOR F:=1 UNTIL 3 DO
          BEGIN
               W1(F):=E(A,W,F);W2(F):=E(A,I,F);
               E(A,W+1,F):=W1(F);E(A,W,F):=W2(F)
          END;
          K:=I-2;G:=W;
          WHILE K>O DO
          BEGIN
               IF E(A,G,1)>E(A,K,1) THEN
               BEGIN
                    W:=W+1;GOTO U
               END
                                          ELSE
```

```
                FOR F:=1 UNTIL 3 DO
                BEGIN
                     W1(F):=E(A,G,F);W2(F):=E(A,K,F);
                     E(A,G,F):=W2(F);E(A,K,F):=W1(F)
                END;
                K:=K-1;G:=G-1
           END;
      U:END;
END;


INPUTR;I_W:=1;S_W:=1;R_W:=8;R_FORMAT:="A";R_D:=3;
FOR I:=1 UNTIL R DO ORDER(I,I);
FOR I:=1 UNTIL R DO
BEGIN
     FOR J:=1 UNTIL R DO
     BEGIN
          IF I<J  THEN
          BEGIN
               FOR Q:=1 UNTIL T(R+1) DO
               FOR G:=1 UNTIL 3 DO
               E(R+1,Q,G):=0;
               NEGATIVE(J);ORDER(R+1,J);
               FOR X:=1 UNTIL N*2 DO
               FOR Y:=1 UNTIL 3 DO
               AUX1(X,Y):=AUX2(X,Y):=0;
               MULTIPLICATION(AUX1,I,R+1);
               MULTIPLICATION(AUX2,R+1,I);
               FOR L:=1 UNTIL N DO
               FOR M:=1 UNTIL N DO
               IF ( AUX1(L,1)=AUX2(M,1) AND AUX1(L,2)=AUX2(M,2)) THEN
               BEGIN
                    ENTRY:=AUX1(L,3)-AUX2(M,3);
                    IF ABS(ENTRY)<1'-2 THEN ENTRY:=0;
                    IF ENTRY\=0 THEN
                              BEGIN
                    WRITE("WARNING' THE ENTRY WITH COORDINATES ");
                    WRITEON(TRUNCATE(AUX1(L,1)),TRUNCATE(AUX2(M,2)));
                    WRITE("OF THE COMMUTATOR (EA",I,",EA_",J,") IS");
                    WRITE("NON_ZERO.ITS VALUE IS=",ENTRY);WRITE(" ")
                              END;
                    FOR X:=1 UNTIL 3 DO AUX1(L,X):=AUX2(M,X):=0;
               END;
               FOR Y:=1 UNTIL N DO
               IF ( AUX1(Y,3)\=0 OR AUX2(Y,3)\=0 ) THEN
               WRITE("WARNING'");
               WRITE("THE COMMUTATOR (EA",I,",EA_",J,")");
               WRITE("HAS  BEEN VERIFIED");
          END
     END
END;
FOR I:=1 UNTIL R DO
BEGIN
```

```
WRITE("THE COMMUTATOR (EA",I,",EA_",I,") IS");WRITE("=DIAG(");
FOR Q:=1 UNTIL T(R+1) DO
FOR G:=1 UNTIL 3 DO
E(R+1,Q,G):=0;
NEGATIVE(I);ORDER(R+1,I);
FOR J:=1 UNTIL N*2 DO
FOR K:=1 UNTIL 3 DO
AUX1(J,K):=AUX2(J,K):=0;
MULTIPLICATION(AUX1,I,R+1);MULTIPLICATION(AUX2,R+1,I);
FOR L:=1 UNTIL N*2 DO
FOR M:=1 UNTIL N*2 DO
IF ( AUX1(L,1)=AUX2(M,1) AND AUX1(L,2)=AUX2(M,2) ) THEN
BEGIN
     ENTRY:=AUX1(L,3)-AUX2(M,3);
   IF ( AUX1(L,1)=AUX1(L,2) AND AUX1(L,1)\=0 ) THEN
     WRITEON(TRUNCATE(AUX1(L,1)),ENTRY,",");
  IF  ( AUX1(L,1)\=AUX1(L,2) AND ENTRY\=0 ) THEN
     BEGIN
                WRITE("WARNING' THE ENTRY WITH CORD. ");
                WRITEON(TRUNCATE(AUX1(L,1)),TRUNCATE(AUX2(M,2)));
                WRITEON("IS NON_ZERO")
     END;
     FOR X:=1 UNTIL 3 DO AUX1(L,X):=AUX2(M,X):=0;
END;
FOR Y:=1 UNTIL N*2 DO
BEGIN
   IF ( AUX1(Y,1)=AUX1(Y,2) AND AUX1(Y,1)\=0 ) THEN
     WRITEON(TRUNCATE(AUX1(Y,1)),AUX1(Y,3),",") ELSE
                IF ( AUX1(Y,1)\=AUX1(Y,2) AND AUX1(Y,3)\=0 ) THEN
                  WRITE("W',1",AUX1(Y,3));
   IF ( AUX2(Y,1)=AUX2(Y,2) AND AUX2(Y,1)\=0 ) THEN
     WRITEON(TRUNCATE(AUX2(Y,1)),-AUX2(Y,3),",") ELSE
                IF ( AUX2(Y,1)\=AUX2(Y,2) AND AUX2(Y,3)\=0 ) THEN
                  WRITE("W',2",AUX2(Y,3));
END;WRITEON(")")
END
END
END
END.
```

§C.5 Group E

I Program E1(5)

```
BEGIN COMMENT C.G.COEF. OF 7*7=27+14+7+1,PROGRAM E1(5);
INTEGER R,M,Q,N1;READON(R,M,Q,N1);
BEGIN
LONG REAL ARRAY E7(1::R,1::N1,1::3);
LONG REAL ARRAY L(1::N1,1::3);
INTEGER ARRAY MULT(1::Q);INTEGER ARRAY N(1::M);
LONG REAL ARRAY AUX,AUX1 (1::N1,1::N1);
INTEGER COUNT;LONG REAL G,H;
INTEGER P;
```

```
PROCEDURE INPUT3R(LONG REAL ARRAY A(*,*,*);INTEGER VALUE B,C,D);
BEGIN
FOR I:=1 UNTIL B DO
FOR J:=1 UNTIL C DO
FOR K:=1 UNTIL D DO
READON(A(I,J,K))
END;

PROCEDURE INITIALIZE1I(INTEGER ARRAY A(*);INTEGER VALUE B);
BEGIN
FOR I:=1 UNTIL B DO
A(I):=0
END;

PROCEDURE INPUT2R(LONG REAL ARRAY A(*,*);INTEGER VALUE B,C);
BEGIN
FOR I:=1 UNTIL B DO
FOR J:=1 UNTIL C DO
READON(A(I,J))
END;

PROCEDURE HIGHEST_WEIGHT(LONG REAL ARRAY A(*,*,*);INTEGER VALUE B,C)
BEGIN
FOR I:=1 UNTIL B DO
FOR J:=1 UNTIL C DO
A(1,I,J):=L(I,J)
END;

PROCEDURE INITIALIZE3R(LONG REAL ARRAY A(*,*,*);INTEGER VALUE B,C,D)
BEGIN
FOR I:=1 UNTIL B DO
FOR J:=1 UNTIL C DO
FOR K:=1 UNTIL D DO
A(I,J,K):=0
END;

PROCEDURE FINDING_MULTIPLET(LONG REAL ARRAY A,B(*,*,*);
                                INTEGER VALUE W,T,Q,C);
BEGIN
     FOR I:=1 UNTIL N1 DO
     IF A(W,I,3)\=0 THEN
     BEGIN
          FOR J:=1 UNTIL N1 DO
          BEGIN
               IF E7(T,J,1)=A(W,I,1) THEN
               BEGIN
                    A(Q,TRUNCATE(E7(T,J,2)),3):=(E7(T,J,3)
                    /B(T,C,3))*A(W,I,3)+A(Q,TRUNCATE(E7(T,J,2)),3)
                     A(Q,TRUNCATE(E7(T,J,2)),1):=E7(T,J,2);
                     A(Q,TRUNCATE(E7(T,J,2)),2):=A(W,I,2)
               END;
```

```
                    IF E7(T,J,1)=A(W,I,2) THEN
                    BEGIN
                         A(Q,TRUNCATE(A(W,I,1)),3):=(E7(T,J,3)
                       /B(T,C,3))*A(W,I,3)+A(Q,TRUNCATE(A(W,I,1)),3);
                         A(Q,TRUNCATE  (A(W,I,1)),1):=A(W,I,1);
                         A( Q,TRUNCATE(A(W,I,1)),2):=E7(T,J,2)
                     END
               END
          END
 END;

 PROCEDURE INITIALIZE2R(LONG REAL ARRAY A(*,*);INTEGER VALUE B,C);
 BEGIN
 FOR I:=1 UNTIL B DO
 FOR J:=1 UNTIL C DO
 A(I,J):=0
 END;

 PROCEDURE LOOP(LONG REAL ARRAY A,B(*,*,*);
                     INTEGER VALUE W,T);
 BEGIN
 INTEGER INDEX;
 INITIALIZE2R(AUX,N(3),N(3));
 FOR I:=1 UNTIL N(3) DO
 IF A(MULT(W),I,3)\=0 THEN
 AUX(TRUNCATE(A(MULT(W),I,1)),TRUNCATE(A(MULT(W),I,2))):=
       A(MULT(W),I,3);
 FOR I:=1 UNTIL W-1 DO
 BEGIN
 INITIALIZE2R(AUX1,N(3),N(3));
 FOR J:=1 UNTIL N(3) DO
 IF A(MULT(I),J,3)\=0 THEN
 AUX1(TRUNCATE(A(MULT(I),J,1)),TRUNCATE(A(MULT(I),J,2)))
      :=A(MULT(I),J,3);
 FOR F:=1 UNTIL N(I) DO
 IF B(1,F,2)=MULT(I) THEN BEGIN G:=B(I,F,3);GOTO E END;
 E:FOR F:=1 UNTIL N(I) DO
        IF B(1,F,2)=MULT(W) THEN BEGIN H:=B(1,F,3);GOTO E1 END;
 E1:
 FOR X:=1 UNTIL N(3) DO
 FOR Y:=1 UNTIL N(3) DO
 AUX(X,Y):=AUX(X,Y)-AUX1(X,Y)*G/H
 END;
 FOR X:=1 UNTIL N(3) DO
 FOR Y:=1 UNTIL 3 DO
 A(MULT(W),X,Y):=0;
 INDEX:=0;
 FOR X:=1 UNTIL N(3) DO
 FOR Y:=1 UNTIL N(3) DO
 IF AUX(X,Y)\=0 THEN
 BEGIN
 INDEX:=INDEX+1;
 A(MULT(W),INDEX,1):=X;A(MULT(W),INDEX,2):=Y;
 A(MULT(W),INDEX,3):=AUX(X,Y)
 END
 END;
```

```
PROCEDURE OUTPUT3R(LONG REAL ARRAY A(*,*,*);
                              INTEGER VALUE B,C,D);
BEGIN
     FOR I:=1 UNTIL B DO
     BEGIN
          I_W:=1;S_W:=1;
          WRITE("THE STATE NUM I=",I);WRITE("______________________");
          WRITE(" ");
          FOR J:=1 UNTIL C DO
          BEGIN
               IF ABS(A(I,J,3))<1'-2 THEN A(I,J,3):=0;
               IF A(I,J,3)\=0 THEN
               BEGIN
                    I_W:=3;S_W:=2;
                    FOR K:=1 UNTIL D-1 DO
                    WRITEON(TRUNCATE(A(I,J,K)));
                    R_W:=5;S_W:=2;R_FORMAT:="A";R_D:=2;
                    WRITEON(A(I,J,3));WRITE(" ")
               END;
          END
     END
END;

FOR I:=1 UNTIL M DO READON(N(I));
INPUT3R(E7,R,N(3),3);

FOR I:=1 UNTIL M DO
BEGIN
LONG REAL ARRAY E(1::R,1::N(I),1::3);
INTEGER ARRAY FUNCTION(1::N(I));
LONG REAL ARRAY MAT(1::N(I),1::N(3),1::3);
INITIALIZE3R(MAT,N(I),N(3),3);
INPUT3R(E,R,N(I),3);
INITIALIZE1I(FUNCTION,N(I));
INPUT2R(L,N(3),3);
HIGHEST_WEIGHT(MAT,N(3),3);

FOR J:=1 UNTIL N(I) DO
BEGIN
FOR K:=1 UNTIL R DO
BEGIN
COUNT:=0;
FOR L:=1 UNTIL N(I) DO
IF E(K,L,1)=J THEN
BEGIN
COUNT:=COUNT+1;P:=L;
MULT(COUNT):=TRUNCATE(E(K,L,2))
END;
IF COUNT=0 THEN GOTO L1;
IF COUNT=1 THEN
BEGIN
FOR W:=1 UNTIL N(I) DO
IF FUNCTION(MULT(COUNT))=W THEN GOTO L1;
FINDING_MULTIPLET(MAT,E,J,K,MULT(COUNT),P);
FUNCTION(MULT(COUNT)):=MULT(COUNT)
END;
```

```
IF COUNT>1 THEN
BEGIN
FOR W:=1 UNTIL N(I) DO
IF FUNCTION(MULT(COUNT))=W THEN GOTO L1;
FINDING_MULTIPLET(MAT,E,J,K,MULT(COUNT),P);
FOR F:=1 UNTIL COUNT-1 DO
BEGIN
FOR X:=1 UNTIL N(I) DO
IF FUNCTION(MULT(F))=X THEN GOTO L2;
FOR W:=1 UNTIL N(I) DO
IF E(2,W,2)=MULT(F) THEN
BEGIN
P:=W;
FINDING_MULTIPLET(MAT,E,TRUNCATE(E(2,W,1)),2,MULT(F),P);
END;
FUNCTION(MULT(F)):=MULT(F);
L2:END;
LOOP(MAT,E,COUNT,J);
FUNCTION(MULT(COUNT)):=MULT(COUNT)
END;
L1:END;I_W:=1;S_W:=1;
END;WRITE("THE C.G.COEF OF THE REPR=",N(I),"ARE");
     WRITE("******************************");
     WRITE(" ");
     OUTPUT3R(MAT,N(I),N(3),3)
END;WRITE(" ")
END
END.
```

II Program E2(5)

```
BEGIN COMMENT C.G.COEF OF 16*16=126+120+10,PROGRAM E2(5);
INTEGER R,M,Q,TT;READON(R,M,Q,TT);
BEGIN
     REAL ARRAY EE(1::R,1::TT,1::3);
     REAL ARRAY L(1::16,1::3);
     INTEGER ARRAY MULT(1::Q);
     REAL ARRAY AUX,AUX1(1::16,1::16);
     INTEGER ARRAY NN(1::R);
     INTEGER COUNT,P;REAL G,H;


PROCEDURE INPUT3R(REAL ARRAY A(*,*,*);INTEGER VALUE B,C,D);
BEGIN
     FOR I:=1 UNTIL B DO
     FOR J:=1 UNTIL C DO
     FOR K:=1 UNTIL D DO
     READON(A(I,J,K))
END;
```

```
PROCEDURE INPUT4R(REAL ARRAY A(*,*,*);INTEGER ARRAY B(*));
BEGIN
     FOR I:=1 UNTIL R DO
     FOR J:=1 UNTIL B(I) DO
     FOR K:=1 UNTIL 3 DO
     READON(A(I,J,K))
END;

PROCEDURE INITIALIZE1I(INTEGER ARRAY A(*);INTEGER VALUE B);
BEGIN
      FOR I:=1 UNTIL B DO
      A(I):=0
END;

PROCEDURE INPUT2R(REAL ARRAY A(*,*);INTEGER VALUE B,C);
BEGIN
     FOR I:=1 UNTIL B DO
     FOR J:=1 UNTIL C DO
     READON(A(I,J))
END;

PROCEDURE HIGHEST_WEIGHT(REAL ARRAY A(*,*,*);INTEGER VALUE B,C);
BEGIN
     FOR I:=1 UNTIL B DO
     FOR J:=1 UNTIL C DO
     A(1,I,J):=L(I,J)
END;

PROCEDURE INITIALIZE3R(REAL ARRAY A(*,*,*);INTEGER VALUE B,C,D);
BEGIN
     FOR I:=1 UNTIL B DO
     FOR J:=1 UNTIL C DO
     FOR K:=1 UNTIL D DO
     A(I,J,K):=0
END;

PROCEDURE FINDING_MULTIPLET(REAL ARRAY A,B(*,*,*);
                            INTEGER VALUE W,T,Q,C);
BEGIN
     FOR I:=1 UNTIL 16 DO
     IF A(W,I,3)\=0 THEN
     BEGIN
          FOR J:=1 UNTIL TT DO
          BEGIN
               IF EE(T,J,1)=A(W,I,1) THEN
               BEGIN
                    A(Q,TRUNCATE(EE(T,J,2)),3):=(EE(T,J,3)
                    /B(T,C,3))*A(W,I,3)+A(Q,TRUNCATE(EE(T,J,2)),3);
                     A(Q,TRUNCATE(EE(T,J,2)),1):=EE(T,J,2);
                     A(Q,TRUNCATE(EE(T,J,2)),2):=A(W,I,2)
               END;
```

```
            IF EE(T,J,1)=A(W,I,2) THEN
            BEGIN
                A(Q,TRUNCATE(A(W,I,1)),3):=(EE(T,J,3)
               /B(T,C,3))*A(W,I,3)+A(Q,TRUNCATE(A(W,I,1)),3);
                A(Q,TRUNCATE  (A(W,I,1)),1):=A(W,I,1);
                A( Q,TRUNCATE(A(W,I,1)),2):=EE(T,J,2)
             END
         END
     END
END;

PROCEDURE INITIALIZE2R(REAL ARRAY A(*,*);INTEGER VALUE B,C);
BEGIN
     FOR I:=1 UNTIL B DO
     FOR J:=1 UNTIL C DO
     A(I,J):=0
END;

PROCEDURE LOOP(REAL ARRAY A,B(*,*,*);INTEGER VALUE W,T,C);
BEGIN
     INTEGER INDEX;INITIALIZE2R(AUX,16,16);
     FOR I:=1 UNTIL 16 DO
     IF A(MULT(W),I,3)\=0 THEN
     AUX(TRUNCATE(A(MULT(W),I,1)),TRUNCATE(A(MULT(W),I,2))):=
     A(MULT(W),I,3);
     FOR I:=1 UNTIL W-1 DO
     BEGIN
         INITIALIZE2R(AUX1,16,16);
         FOR J:=1 UNTIL 16 DO
         IF A(MULT(I),J,3)\=0 THEN
         AUX1((TRUNCATE(A(MULT(I),J,1)),TRUNCATE(A(MULT(I),J,2)))
         :=A(MULT(I),J,3);
         FOR F:=1 UNTIL NN(2) DO
         IF B(C,F,2)=MULT(I) THEN BEGIN G:=B(C,F,3);GOTO E END;
       E:FOR F:=1 UNTIL NN(2) DO
         IF B(C,F,2)=MULT(W) THEN BEGIN H:=B(C,F,3);GOTO E1 END;
      E1:FOR X:=1 UNTIL 16 DO
         FOR Y:=1 UNTIL 16 DO
         AUX(X,Y):=AUX(X,Y)-AUX1(X,Y)*G/H
     END;
     FOR X:=1 UNTIL 16 DO
     FOR Y:=1 UNTIL 16 DO
     A(MULT(W),X,Y):=0;INDEX:=0;
     FOR X:=1 UNTIL 16 DO
     FOR Y:=1 UNTIL 16 DO
     IF AUX(X,Y)\=0 THEN
     BEGIN
         INDEX:=INDEX+1;
         A(MULT(W),INDEX,1):=X;A(MULT(W),INDEX,2):=Y;
         A(MULT(W),INDEX,3):=AUX(X,Y)
     END
END;
```

```
PROCEDURE OUTPUT3R(REAL ARRAY A(*,*,*);INTEGER VALUE B,C,D);
 BEGIN
      FOR I:=1 UNTIL B DO
       BEGIN
             I_W:=1;S_W:=1;
             WRITE("THE STATE NUM I=",I);WRITE("____________________");
             WRITE(" ");
             FOR J:=1 UNTIL C DO
             BEGIN
                   IF ABS(A(I,J,3))<1'-3 THEN A(I,J,3):=0;
                   IF A(I,J,3)\=0 THEN
                   BEGIN
                         I_W:=3;S_W:=2;
                         FOR K:=1 UNTIL D-1 DO
                         WRITEON(TRUNCATE(A(I,J,K)));
                         R_W:=5;S_W:=2;R_FORMAT:="A";R_D:=2;
                         WRITEON(A(I,J,3));WRITE(" ")
                   END;
             END
       END
 END;

 FOR I:=1 UNTIL M DO
 READON(N(I));
 INPUT3R(EE,R,TT,3);

 FOR I:=1 UNTIL M DO
 BEGIN
 FOR W:=1 UNTIL R DO READON(NN(W));
 BEGIN
 REAL ARRAY E(1::R,1::NN(2),1::3);
 INTEGER ARRAY FUNCTION(1::N(I));
 REAL ARRAY MAT(1::N(I),1::16,1::3);
 INITIALIZE3R(E,R,NN(2),3);
 INPUT2R(L,16,3);
 INITIALIZE3R(MAT,N(I),16,3);
 INPUT4R(E,NN);
 INITIALIZE1I(FUNCTION,N(I));
 HIGHEST_WEIGHT(MAT,16,3);
 FOR J:=1 UNTIL N(I) DO
 BEGIN
 FOR K:=R STEP -1 UNTIL 1 DO
 BEGIN
 COUNT:=0;
 FOR L:=1 UNTIL NN(2) DO
 IF E(K,L,1)=J THEN
 BEGIN
 COUNT:=COUNT+1;P:=L;
 MULT(COUNT):=TRUNCATE(E(K,L,2))
 END;
 IF COUNT=0 THEN GOTO L1;
 IF COUNT=1 THEN
 BEGIN
 FOR W:=1 UNTIL N(I) DO
 IF FUNCTION(MULT(COUNT))=W THEN GOTO L1;
 FINDING_MULTIPLET(MAT,E,J,K,MULT(COUNT),P);
 FUNCTION(MULT(COUNT)):=MULT(COUNT)
 END;
```

```
IF COUNT>1 THEN
BEGIN
FOR W:=1 UNTIL N(I) DO
IF FUNCTION(MULT(COUNT))=W THEN GOTO L1;
FINDING_MULTIPLET(MAT,E,J,K,MULT(COUNT),P);
FOR F:=1 UNTIL COUNT-1 DO
BEGIN
FOR X:=1 UNTIL N(I) DO
IF FUNCTION(MULT(F))=X THEN GOTO L2;
FOR FF:=R STEP -1 UNTIL K+1 DO
BEGIN
FOR W:=1 UNTIL N(I) DO
IF FUNCTION(MULT(F))=W THEN GOTO L3;
FOR W:=1 UNTIL NN(2) DO
IF E(FF,W,2)=MULT(F) THEN
BEGIN
P:=W;
FINDING_MULTIPLET(MAT,E,TRUNCATE(E(FF,W,1)),FF,MULT(F),P);
FUNCTION(MULT(F)):=MULT(F)
END;
L3:END;
L2:END;
LOOP(MAT,E,COUNT,J,K);
FUNCTION(MULT(COUNT)):=MULT(COUNT)
END;
L1:END;I_W:=1;S_W:=1;
END;WRITE("THE C.G.COEF OF THE REPR=",N(I),"ARE");
     WRITE("*******************************");
     WRITE(" ");
     OUTPUT3R(MAT,N(I),16   ,3)
END;
END;WRITE(" ")
END
END.
```

BIBLIOGRAPHY

1. M L Perl and P Rapidis, Stanford Linear Accelerator Center, preprint SLAC-PUB-1499.

2. M L Perl, preprint SLAC-PUB-2446 (1980).

3. Y Nambu and G Jona-Lasinio, Phys Rev 122 (1961), 345.

4. Y Nambu and G Jona-Lasinio, Phys Rev 124 (1961), 246.

5. J Goldstone, Nuovo Cimento, 19 (1961), 154.

6. M Baker and S L Glashow, Phys Rev 128 (1962), 2462.

7. S Weinberg, Phys Rev Lett 29 (1972), 1698,
Phys Rev D5 (1972), 1962,
Phys Rev Lett 29 (1972), 388.

8. H Georgi and S L Glashow, Phys Rev D6 (1972), 2977,
Phys Rev D7 (1973), 2457.

9. R N Mohaparta, Phys Rev D9 (1974), 3461.

10. J Frenkel and M E Ebel, preprint COO-363.

11. S M Barr and A Zee, Phys Rev D17 (1978), 1854.

12. P Vinciarelli, Phys Rev D9 (1974), 3456.

13. T Goldman and P Vinciarelli, Phys Rev D10 (1974), 3431.

14. S Coleman and E Weinberg, Phys Rev D7 (1973), 1888.

15. M Gell-Mann, Phys Rev 125 (1962), 1067.

16. C N Yang and R L Mills, Phys Rev 96 (1954), 191.

17. E S Abers and B W Lee, Phys Reports 9C (1973), 1.

18. D E Behrends et al, Rev of Modern Phys 34 (1962), 1.

19. G T Hooff and M Veltman, Diagramar, CERN report 73-9 (1973).

20. S Weinberg, The problem of mass, Harvard preprint HUTP-77/A057.

21. B Lee and S Weinberg, Phys Rev Lett 38 (1977), 1237.

22. D G Sutherland and G Zoupanos, Phys Rev Lett 78B (1978), 455.

23. (a) S L Glashow, Nucl Phys 22 (1961), 579,

(b) S Weinberg, Phys Rev Lett 19 (1967), 1264,

(c) A Salam, Proc 8th Nobel Symposium, 1968 (ed N Svartholm), p 367.

24. H Harari, Phys Rep 426 (1978).

25. Marciano and H Pagels, Phys Rep 36 (1978), 137.

26. (a) H D Politzer, Phys Rev Lett 30 (1973), 1346.

(b) D J Gross and F Wilczek, Phys Rev Lett 30 (1973), 1343.

27. J C Patin and A Salam, Phys Rev D10 (1974), 275.

28. H Georgi and S L Glashow, Phys Rev Lett 32 (1974), 438.

29. M Gell-Mann, P Ramond and R Slansky, Rev of Mod Phys 50 (1978), 721.

30. (a) S L Adler, Phys Rev 177 (1969), 2426,

(b) J S Bell and R Jackiw, Nuovo Cimento 60 (1969), 47.

31. H Georgi and S L Glashow, Phys Rev D6 (1972), 429.

32. M T Vaughn, Desy report 78/78 (1978).

33. H Georgi, Towards a grand unified theory of flouor, Harvard preprint, HUTP-79/A013.

34. (a) M L Metha, J of Math Phys 7 (1966), 1824,

(b) M L Metha and P Srivastava, J of Math Phys 7 (1966), 1833.

35. A J Buras, J Ellis, M K Gaillard and D V Nanopoulos, Nuclear Physics B135 (1978), 66.

36. H Georgi, H R Quinn and S Weinberg, Phys Rev Lett 33 (1974), 451.

37. H Fritzsch and P Minkowski, Annals of Physics 93 (1975), 193.

38. M S Chanowitz, J Ellis and M K Gaillard, Nucl Phys B128 (1977), 506.

39. H Georgi and D V Nanopoulos, Harvard preprints HUTP-78/A039, HUTP-79/A001, HUTP-79/A039.

40. A De Rujula, H Georgi and S L Glashow, Annals of Physics 109 (1977), 242.

41. H Fritzsch, Phys Lett 73B (1978), 317.

42. C E Vayonakis, Phys Lett 32B (1979), 224.

43. F Gürsey, P Ramond and P Sikivie, Phys Lett 60B (1976), 177.

44. Y Achiman and B Stech, Phys Lett 77B (1978), 389.

45. H Ruegg and T Schücker, Nucl Phys B161 (1979), 388.

46. R Barbieri and D V Nanopoulos, preprint CERN TH-2810.

47. M Konuma et al, Sup of Prog Theor Physics 28 (1963), 1.

48. J Balinfante and B Kalman, A survey of Lie groups and Lie algebras with applications and computational methods, Philadelphia (1972).

49. V K Agrawala and B Kolman, BIT 9 (1969), 301.

50. R Beck and B Kolman, J of Comp Phys 7 (1971), 346.

51. J Patera and D Sankoff, Branching rules for representations of simple Lie algebras, Université de Montreal (1972).

52. B G Wybourne and M J Bowick, Aust J of Phys 30 (1977), 259.

53. P H Bulter, R W Haase and B G Wybourne, Aust J of Phys 31 (1978), 131.

54. N Jacobson, Lie algebras, Interscience Publishers (1962).

55. J E Humphreys, Introduction to Lie algebras and representations theory, Springer-Verlag (1972).

56. B G Wybourne, Classical groups for physicists, Wiley-Interscience.

57. D B Lichtenberg, Unitary symmetry and elementary particles, Academic Press (1970).

58. J Patera, R T Sharp and P Winternitz, J of Math Phys 17 (1976), 1972, Errata, J of Math Phys 18 (1977), 1519.

59. N Mckay, J Patera and R T Sharp, preprint Université de Montreal, CRM-601 (1976).

60. E B Dynkin, Am Math Translations 6 (1957), 111.

61. T Maekawa, J of Math Phys 16 (1975), 334.

62. J De Swart, Rev of Mod Phys 35 (1963), 916.

63. P M Van Den Broek and J F Cornwell, Phys Stat Sol (b) 90 (1978), 211.

64. A U Klimyk, Kiev preprint ITP-79-SE (1979).

65. (a) M S Chanowitz, J Ellis and M K Gaillard, Nucl Phys B128 (1977), 506,

(b) M Gourdin, preprint Université Pierve et Marie Curie PAR-LPTHE 80/05, PAR-LPTHE 80/02 (1980).

66. W Miller, Symmetry groups and their applications, p 363, Academic Press (1972).

67. (a) M Gell-Mann, CERN preprint TH.2855,

(b) M Gell-Mann, P Ramond and R Slansky, Supergravity, North-Holland Publishing Company (1979),

(c) A Salam, Trieste preprint IC/79/142 (1979),

(d) P H Frampton, preprint COO-15 45-256.

68. (a) B Stech, preprint HD-THEP -80-4. Presented at the Europhysics Study Conference on Unification of the Fundamental Interactions, Erice, Sicily (1980).

(b) F Gürsey and P Sikivie, Phys Rev Lett 36 (1976), 775..

(c) I Bar and M Günaydin, Yale preprint YTP80-09.