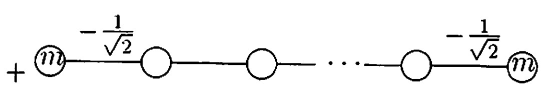
A Computer Program for Obtaining Subsystems

Himmet Can
Department of Mathematics
Faculty of Arts & Sciences
Erciyes University
38039 Kayseri
Turkey
e-mail: can@zirve.erciyes.edu.tr

Lee Hawkins
Department of Mathematics
University of Wales
Aberystwyth SY23 3BZ
United Kingdom

1 Introduction

Dynkin's algorithm [8] gives all the subsystems of a real root system relating to a Weyl group. In this algorithm, the concept of the extended Dynkin diagrams is important. Inspired by these, Hughes [11] introduced what he called extended Cohen diagrams in order to give an algorithm for obtaining subsystems of complex root systems. Unfortunately, this algorithm has its shortcomings, since for type $\pi(m,1,n) = B_n^m$, he gives the following graph



$((n+1)$ points),

as an extended Cohen diagram, where the adjoined root is marked with the sign "+". However, when m is odd, there does not exist a root in $\Phi(m,1,n)$ which can be adjoined in this way. Furthermore, neither Dynkin's nor Hughes' algorithm leads directly to simple systems for subsystems which are subsets of the positive roots.

Subsystems of complex root systems are useful in giving combinatorial constructions of representations of complex reflection groups. For example, they have been used in [3] where the Young tableaux method for generalized symmetric groups [2] have been further generalized.

As the concept of subsystems of root systems for complex reflection groups is not as well developed as in the real case, the first author has

ARS COMBINATORIA 58(2001), pp. 257-269

in [5] presented an alternative algorithm for obtaining all subsystems of a given (real or complex) root system without any reference to extended diagrams. This algorithm has the further advantage that it simultaneously obtains a simple system which is a subset of the positive roots.

Moreover, our method is more useful from the computational point of view. Indeed, in this paper we present a computer program written using the symbolic computation system Maple for the real crystallographic root systems. In a future publication this will be extended to complex root systems. We present the outputs for these root systems for the types E_6, E_7, E_8, F_4 and G_2. The results obtained are a considerable improvement on the ad hoc methods used by Idowu and Morris [12].

2 Subsystems

We now give a brief resume of the main results of [5] in a form suitable for our later purposes. We shall assume the basic notation and terminology as in Can [5], Cohen [7] and Hughes [11]. Let Φ be a root system with a fixed simple system $\pi = (B, \theta)$. The subsystems of Φ fall into two categories. Let Ψ be a subsystem of Φ with simple system $J = (B_\pi, \theta_\pi)$, where $B_\pi \subset B$ and $\theta_\pi = \theta|_{B_\pi}$. Replacing π by another simple system $w\pi$, $w \in W(\pi)$, would just replace Ψ by its conjugate $w\Psi$. All subsystems of Φ obtained in this way are called *parabolic subsystems.* A subsystem of Φ which is not the parabolic is called a *non-parabolic subsystem.* For example, in the type A_n, all subsystems are parabolic but in all the other root systems this is not the case.

The set of all parabolic subsystems of Φ is obtained by removing one or more nodes in all possible ways from the Cohen (Dynkin) diagram (and all equivalent diagrams) of Φ, that is,

2.1

If $\Phi = (R, f)$ is a root system with a fixed simple system $\pi = (B, \theta)$ then the pair $J = (B_\pi, \theta_\pi)$, where $B_\pi \subset B$ and $\theta_\pi = \theta|_{B_\pi}$, is a sub-root graph of π. Furthermore, J yields a parabolic subsystem of Φ [5]. If $\Psi = (S, g)$ is the parabolic subsystem of Φ corresponding to J, recall that its conjugates $w\Psi$, $w \in W(\pi)$, are also parabolic subsystems of Φ.

The set of all non-parabolic subsystems of Φ is obtained by means of the parabolic subsystems of Φ as follows.

2.2

Let $\Phi = (R, f)$ be a root system with a fixed simple system $\pi = (B, \theta)$ and Φ^+ be the "positive" system determined by π. Let $\Psi = (S, g)$ be a parabolic subsystem of Φ with simple system $J = (B_\pi, \theta_\pi)$, where $B_\pi \subset B$ and $\theta_\pi = \theta|_{B_\pi}$ and let Ψ^+ be the "positive" system determined by J. Define $\Phi_\Psi^+ = \Phi^+ \backslash \Psi^+$, and let B_Ψ be a subset of Φ_Ψ^+ such that

$$B_\pi \subset B_\Psi \text{ is linearly independent over } C. \tag{1}$$

Then the pair $J_0 = (B_0, \theta_0)$, where $B_0 = B_\pi \cup B_\Psi$ and $\theta_0 = f|_{B_0}$, is a root graph which is an extension of J which yields a subsystem $\Psi_0 = (S_0, g_0)$ of Φ. If $B_0 \not\subset wB$ for all $w \in W(\pi)$, then $\Psi_0 = (S_0, g_0)$ is a non-parabolic subsystem of Φ and if $B_0 \subset wB$ for some $w \in W(\pi)$, then $\Psi_0 = (S_0, g_0)$ is a parabolic subsystem of Φ by the above definition [5]. If $\Psi_0 = (S_0, g_0)$ is the nonparabolic subsystem of Φ corresponding to J_0, note that its conjugates $w\Psi_0$, $w \in W(\pi)$, are also non-parabolic subsystems of Φ.

As we run through all the parabolic subsystems, we generate all the non-parabolic subsystems. Therefore, the above construction shows that all subsystems of Φ can be obtained up to conjugacy.

2.3

If Φ is a real root system, then we can replace the hypothesis (1) of (2.2) by $(a, b) \leq 0$ for all pairs $a \neq b$ in B_0 [5].

Let Φ be a root system with a fixed simple system π and Φ^+ be the "positive" system of Φ determined by π. If Ψ is a subsystem of Φ obtained by means of the above construction, then a simple system J of Ψ can always be found such that $J \subset \Phi^+$. We recall that this is also true for its conjugates $w\Psi$, where $w \in W(\pi)$ (for a fuller explanation, see [5]). Therefore, having fixed a simple system $\pi = (B, \theta)$ and the corresponding "positive" system Φ^+ in Φ, the above results enable us to construct all subsystems of Φ whose simple systems $J = (B', \theta')$ are such that $B' \subset \Phi^+$, so we have the following result.

2.4

Let $\Phi = (R, f)$ be a root system with a fixed simple system $\pi = (B, \theta)$ and Φ^+ be the "positive" system determined by π. If Ψ is a subsystem of Φ, then a simple system $J = (B', \theta')$ of Ψ can be chosen such that $B' \subset \Phi^+$. The corresponding result for *real crystallographic* root systems Φ has been proved in Idowu and Morris [12]. Thus, if Φ is a real crystallographic root system, then we recover the result of Idowu and Morris [12].

3 The construction of subsystems

Here we concern ourselves with the real crystallographic root systems. If Φ is a real crystallographic root system with simple system $\pi = \{\alpha_1, \ldots, \alpha_n\}$

and $\alpha \in \Phi$, then $\alpha = \sum_{i=1}^{n} \lambda_i \alpha_i$ where $\lambda_i \in \mathbf{Z}$. From now on, α is denoted by $\lambda_1 \lambda_2 \ldots \lambda_n$. We now give an interpretation of the ideas of (2.1) and (2.3) as a computer program written using the symbolic computation system Maple (see [6]). This will be applied to give a combinatorial construction of representations of Weyl groups in terms of root systems (see, for example, [9], [4] and [10]).

```
enumerate all subsystems of a given root system,
both parabolic and non-parabolic, using H. Can's algorithm
and the Maple 'coxeter' package written by Lee Hawkins,
Department of Mathematics, UWA -- August 1995
```

```
set the printlevel to eliminate spurious output
   printlevel := -1:
read in necessary packages: linalg for linear algebra commands
                            combinat for combinatorial commands
                            share for share library
                            coxeter for Coxeter group commands
   with(linalg):
   with(combinat):
   with(share):
   readshare(coxeter,coxeter):
   with(coxeter):
specify the type of root system, e.g. A3, D4, E6, and display the corres-
ponding Dynkin diagram using the coxeter package command 'diagram'
   R := F4:
   print('The root system is of type',R,'with diagram');
   print(diagram(R)):
compute the number of parabolic subsystems of R
   num_parabolic_subsystems := 2^(rank(R)):
   print('The number of parabolic subsystems is',num_parabolic_subsystems):
set up the simple system of R and compute its powerset
   pi := base(R):
   possible_J := convert(powerset(pi),list):
sort the powerset by subset order, largest first
   possible_J := sort(possible_J, \
         proc(s,t) if nops(s)>nops(t) then RETURN(true); \
         else RETURN(false); fi; end):
set up the positive roots of R i) in terms of the basis vectors
e_i, and ii) in terms of linear combinations of simple roots
   positive_roots := pos_roots(R):
   pos_root_sim := map(root_coords,positive_roots,R):
two procedures for list manipulation: scalar_mult_list takes a scalar x
and a list of numbers l and multiplies each entry of l by x,
```

list_add takes two lists l and m and adds them pointwise

```
   scalar_mult_list := proc(x,l) local a,b,c:
        c := []:
        for a to nops(l) do
             b := x * l[a]: c := [op(c),b]:
        od:
        RETURN(c):
   end:
   list_add := proc(l,m) local a,x,c;
        a := nops(l): c := []:
        for x to a do c := [op(c),l[x]+m[x]]: od:
        RETURN(c):
   end:
```

procedure: subsystem(set s)
given a set s of roots expressed in terms of standard co- ordinates, return the type R of the subsystem generated by s (false if no such type), the simple system ordered in accordance with base(R) and the corresponding positive roots of the subsystem
note: uses the type of the underlying root system R as global

```
   subsystem := proc(s) local ss, a, sim_sys, subsys_type, \
                          subsys_canonical_pos_sys, num_entries, \
                          sim_sys_ordered, y, subsys_pos_sys, z, \
                          new_root, t, new_component;
                      global R;
            if s = {} then
                      RETURN(emptyset,[],[]);
            else
            ss := convert(s,list):
            a := traperror(name_of(ss,'p')):
            if a=lasterror then RETURN(false);
            else
                      sim_sys := ss:
                      subsys_type := a:
            subsys_canonical_pos_sys := map(root_coords,pos_roots(a),a):
            num_entries := nops(subsys_canonical_pos_sys[1]):
            sim_sys_ordered := []:
            for y to num_entries do
            sim_sys_ordered := [op(sim_sys_ordered), \
                      sim_sys[p[y]]]:
            od:
            sim_sys_ordered := map(root_coords,sim_sys_ordered,R);
            subsys_pos_sys := []:
            for z to nops(subsys_canonical_pos_sys) do
```

```
                    new_root := [0$rank(R)]:
                    for t to num_entries do
                    new_component := scalar_mult_list(\
                    subsys_canonical_pos_sys[z][t],sim_sys_ordered[t]):
                    new_root := list_add(new_root,new_component):
              od:
              subsys_pos_sys := [op(subsys_pos_sys),new_root]:
              od:
          fi:
          RETURN(subsys_type,sim_sys_ordered,subsys_pos_sys):
   fi:
end:
```

procedure: list_to_e_form(list j)

given a list j representing a linear combination of simple roots return the corresponding root in terms of the basis vectors e_i, using the base of R

```
   list_to_e_form := proc(j) local a, b, q;
                           global R;
          a := 0:
          b := base(R):
          for q to nops(j) do
               a := a + (j[q]*b[q]):
          od:
          RETURN(a);
   end:
```

procedure: inner_prod(list l, list m)

given two lists l and m representing linear combinations of simple roots, compute their inner product using the coxeter package 'iprod' command

note: uses the procedure 'list_to_e_form' to convert such a list into the corresponding basis vector representation

```
   inner_prod := proc(l,m);
          RETURN(iprod(list_to_e_form(l),list_to_e_form(m)));
   end:
```

for each subset J of pi, compute the type of the subsystem whose simple system is the chosen subset J and compute its positive system using the procedure 'subsystem'

```
   print('The parabolic subsystems S are listed below:\ n');
   parabolic_subsystems := []:
   for xx to num_parabolic_subsystems do
          par_subsystem_base := possible_J[xx]:
          r := subsystem(par_subsystem_base):
          if r = false then next;
          else
```

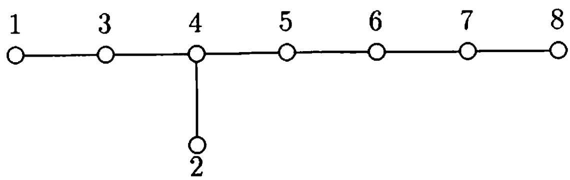
```
            print('S=',r[1],'with simple system J=',r[2]);
            parabolic_subsystems := [op(parabolic_subsystems),[r]];
        fi;
    od;
now look at the non-parabolic subsystems – we construct these
from the parabolic subsystems using Himmet Can's algorithm
exhibit the simple system as a list of linear combinations
    pi := map(root_coords,pi,R):
apply Himmet Can's algorithm
    nonparabolic_info := []:
    for yy to nops(parabolic_subsystems) do
        infor := []:
        par_subs := parabolic_subsystems[yy]:
        if par_subs[1]=emptyset then next; fi:
        infor := [op(infor),par_subs[1]]:
        R_S_plus := convert((convert(pos_root_sim,set) minus \
            convert(par_subs[3],set)),list):
        J := par_subs[2]:
        T_S := []:
        condition := proc(s) local u, f;
                        global J;
            for u to nops(J) do
                        f := inner_prod(s,J[u]):
                        if f > 0 then RETURN(false): fi:
            od:
            RETURN(true):
        end:
        for t to nops(R_S_plus) do
            if condition(R_S_plus[t]) then
                        T_S := [op(T_S),R_S_plus[t]]:
            fi:
        od:
        if nops(T_S)>12 then print('too large'); next; fi;
        if T_S = [] then
            infor := [op(infor),[]]:
            nonparabolic_info := [op(nonparabolic_info),infor]:
            next;
        fi;
        permissible_JPsi := []:
        possible_JPsi := powerset(T_S):
        for f to nops(possible_JPsi) do
            chosen_JPsi := possible_JPsi[f]:
            K := convert(J,set) union chosen_JPsi:
```

```
            if convert(pi,set) union K <> convert(pi,set) then
                    test_condition := proc(K) local t, u, g;
                      for t to nops(K) do
                      for u to nops(K) do
                      if u>t then
                      g := inner_prod(K[t],K[u]):
                      if g>0 then
                      RETURN(false):
                      fi;
                    fi;
            od;
            od;
            RETURN(true):
        end:
        more := test_condition(K):
        if more then
            K_e := []:
            for g to nops(K) do
            K_e := [op(K_e),list_to_e_form(K[g])]:
            od:
            permissible_JPsi :=
                    [op(permissible_JPsi), \
                    [name_of(K_e), chosen_JPsi]]:
                    fi;
            fi;
        od;
    infor := [op(infor), permissible_JPsi]:
    nonparabolic_info := [op(nonparabolic_info),infor]:
    od;
display the results for non-parabolic subsystems
    num_nonparabolic_subsystems := 0:
    for x to nops(nonparabolic_info) do
        num_nonparabolic_subsystems := num_nonparabolic_subsystems + \
            nops(nonparabolic_info[x][2]):
    od:
    print('\ nThere are',num_nonparabolic_subsystems, \
        'non-parabolic subsystems\ n');
    for g to nops(nonparabolic_info) do
        case := nonparabolic_info[g]:
        corresp_parabolic := parabolic_subsystems[g]:
        print('For S=',case[1],'and simple system J=',\
            corresp_parabolic[2],'we have');
        if nops(case[2])=0 then
```

```
            print('no corresponding non-parabolic subsystems');
        else
            for t to nops(case[2]) do
                    Jdash := case[2][t][2] union
                    (convert(corresp_parabolic[2],set)):
                    print('J'=',Jdash,'of type',case[2][t][1]);
            od;
        fi;
    od;
quit;
```

By using the above computer program, all the subsystems of a given real crystallographic root system can be obtained. Since the outputs connected with these are too long to be presented here, we only give all the non-conjugate non-parabolic subsystems for the exceptional types. Comparing these results with the work of Idowu and Morris [12], we have obtained explicit simple systems as subsets of the positive roots of the root system for more non-conjugate non-parabolic subsystems. (The simple roots are as in Bourbaki [1].)

(1) *Type* E_n $(n = 6,7,8)$. The roots are numbered $1,2,3,\ldots$. Let $V = \mathbf{R}^8$ be the real vector space of dimension 8 with standard basis $\{e_i(i = 1,\ldots,8)\}$. We let the Dynkin diagram of type E_8 be



with simple system given by $\pi_{E_8} = \{1,2,3,4,5,6,7,8\}$, where

$$1 = \frac{1}{2}(e_1 - e_2 - e_3 - e_4 - e_5 - e_6 - e_7 + e_8), 2 = e_1 + e_2, 3 = e_2 - e_1,$$

$$4 = e_3 - e_2, 5 = e_4 - e_3, 6 = e_5 - e_4, 7 = e_6 - e_5, 8 = e_7 - e_6.$$

The corresponding postive system for type E_8 is given by

$$\Phi^+_{E_8} == \begin{cases} \pm e_i + e_j; & 1 \le i < j \le 8 \\ \frac{1}{2}\left(e_8 + \sum_{i=1}^{7}(-1)^{v(i)}e_i\right); & \sum_{i=1}^{7} v(i) \text{ is even} \end{cases}$$

The Dynkin diagram of type E_7 is $E_8\backslash\{8\}$ with simple system $\pi_{E_7} = \pi_{E_8}\backslash\{8\}$. The corresponding positive system for type E_7 is

$$\Phi^+_{E_7} = \begin{cases} \pm e_i + e_j; & 1 \le i < j \le 6 \\ e_8 - e_7, & \\ \frac{1}{2}\left(e_8 - e_7 + \sum_{i=1}^{6}(-1)^{v(i)}e_i\right); & \sum_{i=1}^{6} v(i) \text{ is odd} \end{cases}.$$

The Dynkin diagram of type E_6 is $E_7\backslash\{7\}$ with simple system $\pi_{E_6} = \pi_{E_7}\backslash\{7\}$, and the corresponding positive system for type E_6 is

$$\Phi^+_{E_6} = \begin{cases} \pm e_i + e_j; & 1 \le i < j \le 5 \\ \frac{1}{2}\left(e_8 - e_7 - e_6 + \sum_{i=1}^{5}(-1)^{v(i)}e_i\right); & \sum_{i=1}^{5} v(i) \text{ is even} \end{cases}.$$

We now give the non-conjugate non-parabolic subsystems of E_n ($n = 6, 7, 8$), where

$$\begin{aligned} 9 &= 01122100, & 10 &= 11232100, & 11 &= 12232100, \\ 12 &= 01122210, & 13 &= 11233210, & 14 &= 12243210, \\ 15 &= 22343210, & 16 &= 11221000, & 17 &= 01121000, \\ 18 &= 23465432, & 19 &= 23465431, & 20 &= 23465421, \\ 21 &= 23465321, & 22 &= 23464321, & 23 &= 12343210, \\ 24 &= 23354321. \end{aligned}$$

(a) *Type* E_6

Subsystem	Simple system	Subsystem	Simple system
$A_1 + A_5$	$\{1,2,3,4,6,9\}$	$3A_2$	$\{1,2,3,5,6,10\}$
$2A_1 + A_3$	$\{1,3,4,6,11\}$	$4A_1$	$\{1,4,6,11\}$

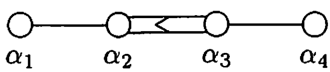
(b) *Type* E_7

Subsystem	Simple system	Subsystem	Simple system
$A_1 + D_6$	$\{1,2,3,4,5,7,12\}$	$A_2 + A_5$	$\{1,2,3,4,6,7,13\}$
A_7	$\{1,2,3,4,6,7,9\}$	$A_1 + 2A_3$	$\{1,2,3,5,6,7,14\}$
$3A_2$	$\{1,2,3,5,6,10\}$	$2A_1 + D_4$	$\{2,3,4,5,7,15\}$
$3A_1 + D_4$	$\{2,3,4,5,7,12,15\}$	$3A_1 + A_3$	$\{1,2,4,5,7,16\}$
$2A_3$	$\{2,3,4,6,7,9\}$	$A_1 + A_5$	$\{1,2,4,5,6,10\}$
A_5	$\{4,5,6,7,11\}$	$2A_1 + A_3$	$\{1,2,4,5,16\}$
$7A_1$	$\{2,3,5,7,15,12,17\}$	$6A_1$	$\{2,3,5,7,15,12\}$
$5A_1$	$\{2,3,5,7,15\}$	$4A_1$	$\{3,5,7,15\}$
$3A_1$	$\{3,6,15\}$		

(c) *Type* E_8

Subsystem	Simple system	Subsystem	Simple system
A_1+E_7	$\{1,2,3,4,5,6,7,18\}$	A_2+E_6	$\{1,2,3,4,5,6,8,19\}$
A_3+D_5	$\{1,2,3,4,5,7,8,20\}$	D_8	$\{1,2,3,4,5,7,8,12\}$
$2A_4$	$\{1,2,3,4,6,7,8,21\}$	A_8	$\{1,2,3,4,6,7,8,13\}$
A_1+A_7	$\{1,2,3,5,6,7,8,14\}$	$A_1+A_2+A_5$	$\{1,2,3,5,6,7,8,22\}$
$2A_1+D_6$	$\{2,3,4,5,6,7,15,18\}$	$8A_1$	$\{2,3,5,7,15,12,17,18\}$
$2A_1+2A_3$	$\{1,2,3,5,6,7,18,14\}$	$2D_4$	$\{1,2,3,4,5,8,12,15\}$
$4A_2$	$\{1,2,3,5,6,8,10,19\}$	$4A_1+D_4$	$\{2,3,4,5,7,18,15,12\}$
A_1+D_6	$\{2,3,4,5,6,7,18\}$	A_2+A_5	$\{1,2,4,5,6,7,23\}$
$2A_1+A_5$	$\{1,2,4,5,6,7,18\}$	A_1+2A_3	$\{1,2,3,5,6,7,14\}$
$2A_1+A_2+A_3$	$\{1,2,3,5,6,7,18\}$	$2A_1+D_5$	$\{1,2,3,4,5,7,18\}$
A_3+D_4	$\{1,2,3,4,5,8,20\}$	$4A_1+A_3$	$\{2,3,5,6,7,18,15\}$
$3A_1+D_4$	$\{2,3,4,5,7,15,12\}$	$7A_1$	$\{2,3,5,7,15,12,17\}$
$6A_1$	$\{2,3,5,7,15,12\}$	$5A_1$	$\{2,3,5,7,12\}$
$4A_1$	$\{7,12,15,18\}$	$2A_1+A_3$	$\{3,5,6,7,18\}$
$2A_3$	$\{3,4,6,7,8,24\}$	A_1+A_5	$\{2,4,5,6,7,18\}$
A_7	$\{3,4,5,6,7,8,15\}$	$4A_1+A_2$	$\{2,3,5,6,15,18\}$
$3A_2$	$\{1,3,5,6,8,19\}$	$3A_1+A_3$	$\{2,3,5,6,7,18\}$
$2A_1+D_4$	$\{2,3,4,5,7,15\}$	A_1+3A_2	$\{1,2,3,5,6,8,19\}$
$3A_2$	$\{1,3,5,6,8,19\}$		

(2) *Type* F_4. Let $V=\mathbf{R}^4$, with standard basis $\{e_1,e_2,e_3,e_4\}$, be the underlying vector space for $W(F_4)$. Let the Dynkin diagram of type F_4 be



with simple system and corresponding positive system given respectively by

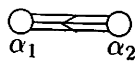
$$\pi_{F_4}=\left\{\alpha_1=\frac{1}{2}(-e_1-e_2-e_3+e_4),\alpha_2=e_1,\alpha_3=e_2-e_1,\alpha_4=e_3-e_2\right\}.$$

$$\Phi^+_{F_4}=\begin{cases}1000,0100,0010,0001,1100,0110,0210,0011,\\1110,0111,2210,0211,1210,1111,2211,0221,\\1211,2221,1221,2421,1321,2431,2321,2432\end{cases}$$

The non-conjugate non-parabolic subsystems of F_4 are

Subsystem	Simple system	Subsystem	Simple system
C_4	{1000, 0100, 0010, 0111}	B_4	{0100, 0010, 0001, 2210}
$C_3 + A_1$	{1000, 0100, 0010, 2432}	$B_3 + A_1$	{0100, 0010, 0001, 2321}
D_4	{0010, 0001, 0210, 2210}	$2B_2$	{0100, 0010, 0111, 2210}
$B_2 + 2A_1$	{0100, 0010, 0111, 2321}	$B_2 + A_1$	{0100, 0010, 0111}
$A_3 + \tilde{A}_1$	{1000, 0010, 0001, 2421}	A_3	{0010, 0001, 0210}
$2A_2$	{1000, 0100, 0001, 2431}	$4A_1$	{0100, 0001, 0221, 2321}
$3A_1$	{0100, 0001, 0221}	$\tilde{A}_3$	{1000, 0100, 0111}
A_3	{0010, 0001, 1210}	$3A_1$	{1000, 0010, 1210}
$A_3 + A_1$	{1000, 0100, 0001, 1221}	D_4	{1000, 0100, 0110, 0111}
$4\tilde{A}_1$	{1000, 1210, 1211, 1221}	$4A_1$	{0010, 0210, 2210, 2432}
$B_2 + 2A_1$	{1000, 0010, 0210, 2432}		

(3) *Type* G_2. Let V be the hyperplane in $\mathbf{R}^3$ consisting of vectors whose coordinates add up to 0. The Dynkin diagram of type G_2 is



with simple system $\pi_{G_2} = \{\alpha_1 = e_1 - e_2, \alpha_2 = -2e_1 + e_2 + e_3\}$. The corresponding positive system for type G_2 is $\Phi^+_{G_2} = \{10, 01, 11, 31, 21, 32\}$. The non-conjugate non-parabolic subsystems of G_2 are

Subsystem	Simple system	Subsystem	Simple system
A_2	{10, 11}	$\tilde{A}_2$	{01, 31}
$A_1 + \tilde{A}_1$	{10, 32}		

References

[1] N. Bourbaki, *Groupes et algebres de Lie*, chapitres 4–6, Actualites Sci. Indust. 1337, Hermann, 1968.

[2] H. Can, Representations of the generalized symmetric groups, *Cont. to Algebra and Geometry* **37**(2) (1996), 289–307.

[3] H. Can, Representations of the imprimitive complex reflection groups $G(m, 1, n)$, *Comm. in Algebra* **26**(8) (1998), 2371–2393.

[4] H. Can, On the perfect systems of the Specht modules of the Weyl groups of type C_n, *Indian J. pure appl. Math.* **29**(3) (1998), 253–269.

[5] H. Can, Some combinatorial results for complex reflection groups, *Europ. J. Combinatorics* **19** (1998), 901–909.

[6] B.W. Char et. al., *The Maple V Language Reference Manual*, Springer-Verlag, New York, 1991.

[7] A.M. Cohen, Finite complex reflection groups, *Ann. Scient. Ec. Norm. Sup* 4(9) (1976), 379–436.

[8] E.B. Dynkin, Semisimple subalgebras of semisimple Lie algebras, *Amer. Math. Soc. Transl.* **6**(2) (1957), 111–244.

[9] S. Halicioglu and A.O. Morris, Specht modules for Weyl groups, *Cont. to Algebra and Geometry* **34**(2) (1993), 257–276.

[10] L. Hawkins, Macdonald Modules for Weyl Groups, PhD thesis, University of Wales, 1996.

[11] M.C. Hughes, Complex reflection groups, *Comm. in Algebra* **18**(2) (1990), 3999–4029.

[12] A.J. Idowu and A.O. Morris, Some combinatorial results for Weyl groups, *Math. Proc. Camb. Phil. Soc.* **101** (1987), 405–420.