# HE316: Mathematica Analysis

*Load packages, that's the way life is*

### SU(2) and SU(3) generators

In[∘]:=
```
<< BProbe`
```
```
Loaded BProbe. See the documentation for help.
```

### Vladimorov Package

In[∘]:=
```
<< SimpleGroupsver098b`
```
```
Package SimpleGroups (v.0.98β (date:06.05.10)) is loaded
Use SetGroup[] to set the group to work with.
```

### GroupMath Package

Will be loaded when required.

# SU(2) Representations

## 1D Representation

In[∘]:=
```
R1 = MatrixRepSU2[1]
```
Out[∘]=
```
{{{0}}, {{0}}, {{0}}}
```

## 2D Representation (kind of Pauli Matrices)

In[∘]:=
```
R2 = MatrixRepSU2[2]
```
Out[∘]=

$$\left\{\left\{\left\{0, \frac{1}{2}\right\}, \left\{\frac{1}{2}, 0\right\}\right\}, \left\{\left\{0, -\frac{i}{2}\right\}, \left\{\frac{i}{2}, 0\right\}\right\}, \left\{\left\{\frac{1}{2}, 0\right\}, \left\{0, -\frac{1}{2}\right\}\right\}\right\}$$

In[∘]:=
```
R2[[1]]
```
Out[∘]=

$$\left\{\left\{0, \frac{1}{2}\right\}, \left\{\frac{1}{2}, 0\right\}\right\}$$

In[•]:= R2[[1]] // MatrixForm

Out[•]//MatrixForm=

$$\begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}$$

In[•]:= R2[[2]] // MatrixForm

Out[•]//MatrixForm=

$$\begin{pmatrix} 0 & -\frac{i}{2} \\ \frac{i}{2} & 0 \end{pmatrix}$$

In[•]:= R2[[3]] // MatrixForm

Out[•]//MatrixForm=

$$\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} \end{pmatrix}$$

```
In[•]:= VectorOfR2 = {R2[[1]], R2[[2]], R2[[3]]};

In[•]:= CommutatorMatrixR2 =
  Table[VectorOfR2[[i]].VectorOfR2[[j]] - VectorOfR2[[j]].VectorOfR2[[i]],
    {i, 1, 3}, {j, 1, 3}] // Simplify // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} \frac{i}{2} & 0 \\ 0 & -\frac{i}{2} \end{pmatrix} & \begin{pmatrix} 0 & -\frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix} \\ \begin{pmatrix} -\frac{i}{2} & 0 \\ 0 & \frac{i}{2} \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & \frac{i}{2} \\ \frac{i}{2} & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & \frac{1}{2} \\ -\frac{1}{2} & 0 \end{pmatrix} & \begin{pmatrix} 0 & -\frac{i}{2} \\ -\frac{i}{2} & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \end{pmatrix}$$

# 3D Representation [SU(2) ≅ SO(3)]

*Actually, SU(2) is the double cover of SO(3). They are NOT isomorphic. This example shows how they are similar in one region.*

In[•]:= T = MatrixRepSU2[3]

Out[•]=

$$\left\{\left\{\left\{0, \frac{1}{\sqrt{2}}, 0\right\}, \left\{\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}\right\}, \left\{0, \frac{1}{\sqrt{2}}, 0\right\}\right\}, \left\{\left\{0, -\frac{i}{\sqrt{2}}, 0\right\}, \left\{\frac{i}{\sqrt{2}}, 0, -\frac{i}{\sqrt{2}}\right\}, \left\{0, \frac{i}{\sqrt{2}}, 0\right\}\right\}, \{\{1, 0, 0\}, \{0, 0, 0\}, \{0, 0, -1\}\}\right\}$$

In[•]:= Tx = T[[1]]

Out[•]=

$$\left\{\left\{0, \frac{1}{\sqrt{2}}, 0\right\}, \left\{\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}\right\}, \left\{0, \frac{1}{\sqrt{2}}, 0\right\}\right\}$$

In[∘]:= Ty = T〚2〛

Out[∘]=

$$\left\{\left\{0, -\frac{i}{\sqrt{2}}, 0\right\}, \left\{\frac{i}{\sqrt{2}}, 0, -\frac{i}{\sqrt{2}}\right\}, \left\{0, \frac{i}{\sqrt{2}}, 0\right\}\right\}$$

In[∘]:= Tz = T〚3〛

Out[∘]=

{{1, 0, 0}, {0, 0, 0}, {0, 0, -1}}

In[∘]:= VecT = {Tx, Ty, Tz};

In[∘]:= CommutatorMatrixT =
 Table[VecT〚i〛.VecT〚j〛 - VecT〚j〛.VecT〚i〛, {i, 1, 3}, {j, 1, 3}] // Simplify //
  MatrixForm

Out[∘]//MatrixForm=

$$\begin{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} i & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -i \end{pmatrix} & \begin{pmatrix} 0 & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 \end{pmatrix} \\ \begin{pmatrix} -i & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & i \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & \frac{i}{\sqrt{2}} & 0 \\ \frac{i}{\sqrt{2}} & 0 & \frac{i}{\sqrt{2}} \\ 0 & \frac{i}{\sqrt{2}} & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & -\frac{1}{\sqrt{2}} & 0 \end{pmatrix} & \begin{pmatrix} 0 & -\frac{i}{\sqrt{2}} & 0 \\ -\frac{i}{\sqrt{2}} & 0 & -\frac{i}{\sqrt{2}} \\ 0 & -\frac{i}{\sqrt{2}} & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{pmatrix}$$

## Making a Ladder Operator Type Approach

In[∘]:= Tp = (Tx + I Ty) // Simplify

Out[∘]=

$$\left\{\left\{0, \sqrt{2}, 0\right\}, \left\{0, 0, \sqrt{2}\right\}, \{0, 0, 0\}\right\}$$

In[∘]:= Tm = (Tx - I Ty) // Simplify

Out[∘]=

$$\left\{\{0, 0, 0\}, \left\{\sqrt{2}, 0, 0\right\}, \left\{0, \sqrt{2}, 0\right\}\right\}$$

In[∘]:= $\left\{\{0, 0, 0\}, \left\{\sqrt{2}, 0, 0\right\}, \left\{0, \sqrt{2}, 0\right\}\right\}$

Out[∘]=

$$\left\{\{0, 0, 0\}, \left\{\sqrt{2}, 0, 0\right\}, \left\{0, \sqrt{2}, 0\right\}\right\}$$

In[∘]:= RealVecT = {Tp, Tm, Tz}

Out[∘]=

$$\left\{\left\{\left\{0, \sqrt{2}, 0\right\}, \left\{0, 0, \sqrt{2}\right\}, \{0, 0, 0\}\right\}, \left\{\{0, 0, 0\}, \left\{\sqrt{2}, 0, 0\right\}, \left\{0, \sqrt{2}, 0\right\}\right\}, \{\{1, 0, 0\}, \{0, 0, 0\}, \{0, 0, -1\}\}\right\}$$

In[∘]:=
```
CommutatorMatrixRT = Table[
   RealVecT[[i]].RealVecT[[j]] - RealVecT[[j]].RealVecT[[i]], {i, 1, 3}, {j, 1, 3}] // Simplify
```

Out[∘]=

$$\{\{\{\{0,0,0\},\{0,0,0\},\{0,0,0\}\},\ \{\{2,0,0\},\{0,0,0\},\{0,0,-2\}\},\ \{\{0,-\sqrt{2},0\},\{0,0,-\sqrt{2}\},\{0,0,0\}\}\},\ \{\{\{-2,0,0\},\{0,0,0\},\{0,0,2\}\},\ \{\{0,0,0\},\{0,0,0\},\{0,0,0\}\},\ \{\{0,0,0\},\{\sqrt{2},0,0\},\{0,\sqrt{2},0\}\}\},\ \{\{\{0,\sqrt{2},0\},\{0,0,\sqrt{2}\},\{0,0,0\}\},\ \{\{0,0,0\},\{-\sqrt{2},0,0\},\{0,-\sqrt{2},0\}\},\ \{\{0,0,0\},\{0,0,0\},\{0,0,0\}\}\}\}$$

In[∘]:=
```
StructConstRealT = {{0, 2 Tz, -Tp}, {-2 Tz, 0, Tm }, {Tp, -Tm, 0}}
```

Out[∘]=

$$\{\{0,\ \{\{2,0,0\},\{0,0,0\},\{0,0,-2\}\},\ \{\{0,-\sqrt{2},0\},\{0,0,-\sqrt{2}\},\{0,0,0\}\}\},\ \{\{\{-2,0,0\},\{0,0,0\},\{0,0,2\}\},\ 0,\ \{\{0,0,0\},\{\sqrt{2},0,0\},\{0,\sqrt{2},0\}\}\},\ \{\{\{0,\sqrt{2},0\},\{0,0,\sqrt{2}\},\{0,0,0\}\},\ \{\{0,0,0\},\{-\sqrt{2},0,0\},\{0,-\sqrt{2},0\}\},\ 0\}\}$$

In[∘]:=
```
CommutatorMatrixRT - StructConstRealT // Simplify // MatrixForm
```

Out[∘]//MatrixForm=

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

In[∘]:=
```
Clear[R1, R2, Tx, Ty, Tz, Tm, Tp, T, CommutatorMatrixR2,
  CommutatorMatrixRT, CommutatorMatrixT, StructConstRealT, VecT, RealVecT]
```

# SU(3) Representations

## Gell-Mann Generators

In[∘]:=
```
R = MatrixRepSU3[{1, 0}]
```

Out[∘]=

$$\{\{\{0,\tfrac{1}{2},0\},\{\tfrac{1}{2},0,0\},\{0,0,0\}\},\ \{\{0,-\tfrac{i}{2},0\},\{\tfrac{i}{2},0,0\},\{0,0,0\}\},\ \{\{\tfrac{1}{2},0,0\},\{0,-\tfrac{1}{2},0\},\{0,0,0\}\},\ \{\{0,0,\tfrac{1}{2}\},\{0,0,0\},\{\tfrac{1}{2},0,0\}\},\ \{\{0,0,-\tfrac{i}{2}\},\{0,0,0\},\{\tfrac{i}{2},0,0\}\},\ \{\{0,0,0\},\{0,0,\tfrac{1}{2}\},\{0,\tfrac{1}{2},0\}\},\ \{\{0,0,0\},\{0,0,-\tfrac{i}{2}\},\{0,\tfrac{i}{2},0\}\},\ \{\{\tfrac{1}{2\sqrt{3}},0,0\},\{0,\tfrac{1}{2\sqrt{3}},0\},\{0,0,-\tfrac{1}{\sqrt{3}}\}\}\}$$

```
In[]:= Y = 2 R[[8]] / Sqrt[3]
Out[]=
```

$$\left\{\left\{\frac{1}{3}, 0, 0\right\}, \left\{0, \frac{1}{3}, 0\right\}, \left\{0, 0, -\frac{2}{3}\right\}\right\}$$

```
In[]:= Tz = R[[3]]
Out[]=
```

$$\left\{\left\{\frac{1}{2}, 0, 0\right\}, \left\{0, -\frac{1}{2}, 0\right\}, \{0, 0, 0\}\right\}$$

```
In[]:= Tp = (R[[1]] + I R[[2]])
Out[]=
{{0, 1, 0}, {0, 0, 0}, {0, 0, 0}}

In[]:= Tm = (R[[1]] - I R[[2]])
Out[]=
{{0, 0, 0}, {1, 0, 0}, {0, 0, 0}}

In[]:= Vp = (R[[4]] + I R[[5]])
Out[]=
{{0, 0, 1}, {0, 0, 0}, {0, 0, 0}}

In[]:= Vm = (R[[4]] - I R[[5]])
Out[]=
{{0, 0, 0}, {0, 0, 0}, {1, 0, 0}}

In[]:= Up1 = (R[[6]] + I R[[7]])
Out[]=
{{0, 0, 0}, {0, 0, 1}, {0, 0, 0}}

In[]:= Um = (R[[6]] - I R[[7]])
Out[]=
{{0, 0, 0}, {0, 0, 0}, {0, 1, 0}}

In[]:= VecR = {Tp, Tm, Tz, Up1, Um, Vp, Vm, Y};

In[]:= StructConstR = {{0, 2 Tz, -Tp, Vp, 0, 0, -Um, 0},
     {-2 Tz, 0, Tm, 0, -Vm, Up1, 0, 0}, {Tp, -Tm, 0, -Up1 / 2, Um / 2, Vp / 2, -Vm / 2, 0},
     {-Vp, 0, Up1 / 2, 0, 3 / 2 Y - Tz, 0, Tm, -Up1}, {0, Vm, -Um / 2,
      -3 / 2 Y + Tz, 0, -Tp, 0, Um}, {0, -Up1, -Vp / 2, 0, Tp, 0, 3 / 2 Y + Tz, -Vp},
     {Um, 0, Vm / 2, -Tm, 0, -3 / 2 Y - Tz, 0, Vm}, {0, 0, 0, Up1, -Um, Vp, -Vm, 0}};

In[]:= StructConstR // MatrixForm
Out[]//MatrixForm=
```

| | | |
|---|---|---|
| 0 | {{1, 0, 0}, {0, -1, 0}, {0, 0, 0}} | {{0, -1, 0}, {0, 0, 0 |
| {{-1, 0, 0}, {0, 1, 0}, {0, 0, 0}} | 0 | {{0, 0, 0}, {1, 0, 0} |
| {{0, 1, 0}, {0, 0, 0}, {0, 0, 0}} | {{0, 0, 0}, {-1, 0, 0}, {0, 0, 0}} | 0 |
| {{0, 0, -1}, {0, 0, 0}, {0, 0, 0}} | 0 | $\{\{0, 0, 0\}, \{0, 0, \frac{1}{2}\}$ |
| 0 | {{0, 0, 0}, {0, 0, 0}, {1, 0, 0}} | {{0, 0, 0}, {0, 0, 0}, |
| 0 | {{0, 0, 0}, {0, 0, -1}, {0, 0, 0}} | $\{\{0, 0, -\frac{1}{2}\}, \{0, 0, 0$ |
| {{0, 0, 0}, {0, 0, 0}, {0, 1, 0}} | 0 | {{0, 0, 0}, {0, 0, 0} |
| 0 | 0 | 0 |

In[ ]:=
```
CommutatorMatrixR =
  Table[VecR[[i]].VecR[[j]] - VecR[[j]].VecR[[i]], {i, 1, 8}, {j, 1, 8}] // Simplify;
```

In[ ]:=
```
CommutatorMatrixR // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix}
\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}1&0&0\\0&-1&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&-1&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&1\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&-1&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} \\
\begin{pmatrix}-1&0&0\\0&1&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\1&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\-1&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&1\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} \\
\begin{pmatrix}0&1&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\-1&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&-\frac{1}{2}\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&\frac{1}{2}&0\end{pmatrix} & \begin{pmatrix}0&0&\frac{1}{2}\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\-\frac{1}{2}&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} \\
\begin{pmatrix}0&0&-1\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&\frac{1}{2}\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&1&0\\0&0&-1\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\1&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&-1\\0&0&0\end{pmatrix} \\
\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\1&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&-\frac{1}{2}&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&-1&0\\0&0&1\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&-1&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&1&0\end{pmatrix} \\
\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&-1\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&-\frac{1}{2}\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&1&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}1&0&0\\0&0&0\\0&0&-1\end{pmatrix} & \begin{pmatrix}0&0&-1\\0&0&0\\0&0&0\end{pmatrix} \\
\begin{pmatrix}0&0&0\\0&0&0\\0&1&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\\frac{1}{2}&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\-1&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}-1&0&0\\0&0&0\\0&0&1\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\1&0&0\end{pmatrix} \\
\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&1\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&-1&0\end{pmatrix} & \begin{pmatrix}0&0&1\\0&0&0\\0&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\-1&0&0\end{pmatrix} & \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}
\end{pmatrix}$$

In[ ]:=
```
StructConstR - CommutatorMatrixR /. {ConstantArray[0, {3, 3}] → 0} // Simplify //
  MatrixForm
```

Out[ ]//MatrixForm=

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

In[ ]:=
```
Clear[R, Y, Tp, Tm, Tz, Vp, Vm, Up1, Um]
```

# Adjoint Representation

In[ ]:=
```
R = MatrixRepSU3[{1, 1}];
```

In[ ]:=
```
Y = 2 R[[8]] / Sqrt[3];
```

In[ ]:=
```
Tp = (R[[1]] + I R[[2]]);
```

```
In[•]:= Tm = (R⟦1⟧ - I R⟦2⟧);
In[•]:= Tz = R⟦3⟧;
In[•]:= Vp = (R⟦4⟧ + I R⟦5⟧);
In[•]:= Vm = (R⟦4⟧ - I R⟦5⟧);
In[•]:= Upl = (R⟦6⟧ + I R⟦7⟧);
In[•]:= Um = (R⟦6⟧ - I R⟦7⟧);
```

## Killing Form

Defined as (a,b) = Tr[$ad_a ad_b$]

if you expect it be zero, you can compute [a,[b,x]] for some x ∈ L, and then show it be zero. This kind of thing is done for the Killing forms of root vectors (later).

```
In[•]:= Tr[Tz.Tz]
Out[•]=
3
```

```
In[•]:= Tz // MatrixForm
Out[•]//MatrixForm=
```

$$\begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} \end{pmatrix}$$

```
In[•]:= Eigenvectors[Tz]
Out[•]=
{{0, 0, 0, 0, 1, 0, 0, 0}, {0, 0, 1, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 1}, {0, 1, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 1, 0},
 {1, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 1, 0, 0}, {0, 0, 0, 1, 0, 0, 0, 0}}
```

```
In[•]:= Tr[Tz.Y]
Out[•]=
0
```

```
In[•]:= Tr[Y.Y]
Out[•]=
4
```

```
In[•]:= Tr[Tz.Tp]
Out[•]=
0
```

In[∘]:= **Tr[Tz.Tm]**

Out[∘]=

0

In[∘]:= **Tr[Tz.Vp]**

Out[∘]=

0

In[∘]:= **Tr[Tz.Vm]**

Out[∘]=

0

In[∘]:= **VecAdR = {Tz, Y, Tp, Tm, Vp, Vm, Upl, Um};**

In[∘]:= **VecAdR⟦1⟧ - Tz // Simplify**

Out[∘]=

```
{{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}}
```

In[∘]:= **KillingTable = Table[Tr[VecAdR⟦i⟧.VecAdR⟦j⟧], {i, 1, 8}, {j, 1, 8}] // Simplify**

Out[∘]=

```
{{3, 0, 0, 0, 0, 0, 0, 0}, {0, 4, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 6, 0, 0, 0, 0}, {0, 0, 6, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 6, 0, 0},
 {0, 0, 0, 0, 6, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 6}, {0, 0, 0, 0, 0, 0, 6, 0}}
```

In[∘]:= **KillingTable // MatrixForm**

Out[∘]//MatrixForm=

$$\begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 \end{pmatrix}$$

In[∘]:= **X = a Tz + b Y;**

In[∘]:= **X // MatrixForm**

Out[∘]//MatrixForm=

$$\begin{pmatrix} \frac{a}{2}+b & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{a}{2}+b & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{a}{2}-b & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{a}{2}-b \end{pmatrix}$$

In[∘]:= **Tr[X.X] // Simplify**

Out[∘]=

$3\,a^2 + 4\,b^2$

Killing forms are not inner products. In particular, they are not positive definite. There is an inner product associated with the Lie Algebra but it is defined on the space containing the roots. Roots are explained below.

```
In[∘]:= CommutatorMatrixAdR =
          Table[VecAdR[[i]].VecAdR[[j]] - VecAdR[[j]].VecAdR[[i]], {i, 1, 8}, {j, 1, 8}] // Simplify;
```

```
In[∘]:= CommutatorMatrixAdR /. {Tz → tz, Tp → tp, Tm → tm, Vp → vp, Vm → vm, Upl → up, Um → um,
          Y → y, ConstantArray[0, {8, 8}] → 0, -Tz → -tz, -Tp → -tp, -Tm → -tm, -Vp → -vp,
          -Vm → -vm, -Upl → -up, -Um → -um, -Y → -y, 2 Tz → 2 tz, -2 Tz → -2 tz, Vp / 2 → vp / 2,
          Vm / 2 → vm / 2, Upl / 2 → up / 2, Um / 2 → um / 2, -Vp / 2 → -vp / 2, -Vm / 2 → -vm / 2,
          -Upl / 2 → -up / 2, -Um / 2 → -um / 2, 3 Y / 2 - Tz → 3 y / 2 - tz, 3 Y / 2 + Tz → 3 y / 2 + tz,
          -3 Y / 2 - Tz → - 3 y / 2 - tz, -3 Y / 2 + Tz → - 3 y / 2 + tz } // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix}
0 & 0 & t_p & -t_m & \frac{v_p}{2} & -\frac{v_m}{2} & -\frac{u_p}{2} & \frac{u_m}{2} \\
0 & 0 & 0 & 0 & v_p & -v_m & u_p & -u_m \\
-t_p & 0 & 0 & 2 t_z & 0 & -u_m & v_p & 0 \\
t_m & 0 & -2 t_z & 0 & u_p & 0 & 0 & -v_m \\
-\frac{v_p}{2} & -v_p & 0 & -u_p & 0 & \frac{3 y}{2}+t_z & 0 & t_p \\
\frac{v_m}{2} & v_m & u_m & 0 & -\frac{3 y}{2}-t_z & 0 & -t_m & 0 \\
\frac{u_p}{2} & -u_p & -v_p & 0 & 0 & t_m & 0 & \frac{3 y}{2}-t_z \\
-\frac{u_m}{2} & u_m & 0 & v_m & -t_p & 0 & -\frac{3 y}{2}+t_z & 0
\end{pmatrix}$$

Note that Y can form a Cartan Subalgebra only with Tz (since all the others are basis eigenvectors of $ad_x$ if x is in the subalgebra).

Y can form an abelian subalgebra with either of Tz, Tp, or Tm.

To form a Cartan Subalgebra, we consider the maximally commutating subalgebra that contains a **regular element** (element with the smallest dimensionality of the zero eigenvalue subspace).

## Roots in the Dual Space of Cartan Subalgebra

Cartan Subalgebra is the maximally commutating Lie subalgebra. We consider that the adjoint representation means -> $ad_y x = [y,x]$, and we are trying to represent all $y \in g$ in the $ad_y$ format.

Dual Space is the space of linear functionals on our Lie Algebra.
Roots lie in Dual Space of the Cartan Subalgebra (in its adjoint representation) and Root vectors are the corresponding generators.

For $\alpha \in H^*$ a root, we have a unique element (unique only for semi-simple Lie algebras) $h_\alpha \in H$, which behaves like

$$\alpha(h) = (h_\alpha, h) \quad \forall\, h \in H$$

Example:

$X_q(aT_z + bY)$ = some linear combination of a and b depending on q
Here, X is the root, q is a root vector, a and b are arbitrary constants, and $T_z$ and Y are the Cartan subalgebra elements of SU(3).

In[∘]:=
```
h1 = Tz / 3; h2 = -1 / 6 Tz + Y / 4; h3 = Tz / 6 + Y / 4; RootVecAdR = {h1, h2, h3};
```

In[∘]:=
```
al1 = Tr[h1.(a Tz + b Y)] // Simplify
```

Out[∘]=

$$a$$

In[∘]:=
```
al2 = Tr[h2.(a Tz + b Y)] // Simplify
```

Out[∘]=

$$-\frac{a}{2} + b$$

In[∘]:=
```
al3 = Tr[h3.(a Tz + b Y)] // Simplify
```

Out[∘]=

$$\frac{a}{2} + b$$

$h_1$, $h_2$, $h_3$ belong to the Cartan subalgebra. Their Killing form with any general element of H gives the eigenvalues for roots corresponding to the root vectors $t_p$, $u_p$, $v_p \in$ L-H.

In[∘]:=
```
RootKillingTable =
  Table[Tr[RootVecAdR⟦i⟧.RootVecAdR⟦j⟧], {i, 1, 3}, {j, 1, 3}] // Simplify
```

Out[∘]=

$$\left\{\left\{\frac{1}{3}, -\frac{1}{6}, \frac{1}{6}\right\}, \left\{-\frac{1}{6}, \frac{1}{3}, \frac{1}{6}\right\}, \left\{\frac{1}{6}, \frac{1}{6}, \frac{1}{3}\right\}\right\}$$

Observe that these are positive-semidefinite.

In[∘]:=
```
RootKillingTable // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} \frac{1}{3} & -\frac{1}{6} & \frac{1}{6} \\ -\frac{1}{6} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{3} \end{pmatrix}$$

*Root vectors live in L-H and roots live in H* (dual space of Cartan Subalgebra).*

We can partition L as the Cartan subalgebra and the space spanned by the root vectors.
Let $\Sigma$ = {set of roots}.
Let the set $\{e_\alpha\}$ of root vectors for $\alpha \in \Sigma$ be written as $\{e_\alpha\} = E^o$.
Then

$$L = H \cup E^o \qquad \text{and} \qquad H \cap E^o = \phi.$$

We can now define $<\alpha,\beta>$ for roots $\alpha$, $\beta \in H^*$ to be equal to $(h_\alpha, h_\beta)$ for the corresponding roots $\in$ H*. This is a valid **inner product**.

## *Relation between Root Vectors and SU(3) operations on 3 points (related to a*

*later section)*

```
In[]:= Clear[R, Tz, Y, Upl, Um, Vm, Vp, Tm, Tp]
In[]:= R = MatrixRepSU3[{1, 0}];
In[]:= Y = 2 R[[8]] / Sqrt[3];
In[]:= Tp = (R[[1]] + I R[[2]]);
In[]:= Tm = (R[[1]] - I R[[2]]);
In[]:= Tz = R[[3]];
In[]:= Vp = (R[[4]] + I R[[5]]);
In[]:= Vm = (R[[4]] - I R[[5]]);
In[]:= Upl = (R[[6]] + I R[[7]]);
In[]:= Um = (R[[6]] - I R[[7]]);
In[]:= sol = Solve[{al1 == Al1, al2 == Al2}, {a, b}]
```

Out[]=

$$\left\{\left\{a \to Al1,\ b \to \frac{1}{2}\ (Al1 + 2\,Al2)\right\}\right\}$$

```
In[]:= (a Tz + b Y).{1, 0, 0} /. sol // Expand
```

Out[]=

$$\left\{\left\{\frac{2\,Al1}{3} + \frac{Al2}{3},\ 0,\ 0\right\}\right\}$$

```
In[]:= (a Tz + b Y).{0, 0, 1} /. sol // Expand
```

Out[]=

$$\left\{\left\{0,\ 0,\ -\frac{Al1}{3} - \frac{2\,Al2}{3}\right\}\right\}$$

```
In[]:= (a Tz + b Y).{0, 1, 0} /. sol // Expand
```

Out[]=

$$\left\{\left\{0,\ -\frac{Al1}{3} + \frac{Al2}{3},\ 0\right\}\right\}$$

```
In[]:= Clear[R, Y, Tp, Tm, Tz, Vp, Vm, Upl, Um];
In[]:= R = MatrixRepSU3[{0, 1}];
In[]:= Y = 2 R[[8]] / Sqrt[3];
In[]:= Tp = (R[[1]] + I R[[2]]);
In[]:= Tm = (R[[1]] - I R[[2]]);
In[]:= Tz = R[[3]];
In[]:= Vp = (R[[4]] + I R[[5]]);
In[]:= Vm = (R[[4]] - I R[[5]]);
In[]:= Upl = (R[[6]] + I R[[7]]);
```

In[∘]:= **sol = Solve[{al1 == Al1, al2 == Al2}, {a, b}]**

Out[∘]=

$$\left\{\left\{a \to \text{Al1},\ b \to \frac{1}{2}\ (\text{Al1} + 2\ \text{Al2})\right\}\right\}$$

In[∘]:= **(a Tz + b Y).{1, 0, 0} /. sol // Expand**

Out[∘]=

$$\left\{\left\{\frac{\text{Al1}}{3} + \frac{2\ \text{Al2}}{3},\ 0,\ 0\right\}\right\}$$

In[∘]:= **(a Tz + b Y).{0, 1, 0} /. sol // Expand**

Out[∘]=

$$\left\{\left\{0,\ \frac{\text{Al1}}{3} - \frac{\text{Al2}}{3},\ 0\right\}\right\}$$

In[∘]:= **(a Tz + b Y).{0, 0, 1} /. sol // Expand**

Out[∘]=

$$\left\{\left\{0,\ 0,\ -\frac{2\ \text{Al1}}{3} - \frac{\text{Al2}}{3}\right\}\right\}$$

In[∘]:= **Clear[R, Y, Tp, Tm, Tz, Vp, Vm, Upl, Um];**

In[∘]:= **R1 = MatrixRepSU3[{1, 0}];**

In[∘]:= **R2 = MatrixRepSU3[{0, 1}];**

In[∘]:= **R1 – R2 // Simplify**

Out[∘]=

$$\left\{\left\{\left\{0, \frac{1}{2}, 0\right\}, \left\{\frac{1}{2}, 0, -\frac{1}{2}\right\}, \left\{0, -\frac{1}{2}, 0\right\}\right\}, \left\{\left\{0, -\frac{i}{2}, 0\right\}, \left\{\frac{i}{2}, 0, \frac{i}{2}\right\}, \left\{0, -\frac{i}{2}, 0\right\}\right\},\right.$$
$$\left\{\left\{\frac{1}{2}, 0, 0\right\}, \{0, -1, 0\}, \left\{0, 0, \frac{1}{2}\right\}\right\}, \{\{0, 0, 1\}, \{0, 0, 0\}, \{1, 0, 0\}\},$$
$$\{\{0, 0, -i\}, \{0, 0, 0\}, \{i, 0, 0\}\}, \left\{\left\{0, -\frac{1}{2}, 0\right\}, \left\{-\frac{1}{2}, 0, \frac{1}{2}\right\}, \left\{0, \frac{1}{2}, 0\right\}\right\},$$
$$\left.\left\{\left\{0, \frac{i}{2}, 0\right\}, \left\{-\frac{i}{2}, 0, -\frac{i}{2}\right\}, \left\{0, \frac{i}{2}, 0\right\}\right\}, \left\{\left\{-\frac{1}{2\sqrt{3}}, 0, 0\right\}, \left\{0, \frac{1}{\sqrt{3}}, 0\right\}, \left\{0, 0, -\frac{1}{2\sqrt{3}}\right\}\right\}\right\}$$

In[∘]:= **Clear[R1, R2];**

# A brief discussion on definitions:

In[∘]:=

Suppose $L$ is a Lie algebra. A **subalgebra**, M, is simply a $L$ which is closed under the Lie product. For example, $t_z, t_+$, and $t_-$ subalgebra of SU(3) which is indeed isomorphic (equivalent) to SU(2). S if $M$ is a subalgebra and $x, y \in M$, then $[x, y] \in M$. An **ideal** is a spe subalgebra. If $J$ is an ideal, and $x \in J$ and $y$ is any element of $L$, the If $J$ were only a subalgebra instead of an ideal, we would have to restri $J$ rather than just in $L$.

Out[∘]=

Suppose $L$ is a Lie algebra. A **subalgebra**, M, is simply a $L$ which is closed under the Lie product. For example, $t_z, t_+$, and $t_-$ subalgebra of SU(3) which is indeed isomorphic (equivalent) to SU(2). S if $M$ is a subalgebra and $x, y \in M$, then $[x, y] \in M$. An **ideal** is a spe subalgebra. If $J$ is an ideal, and $x \in J$ and $y$ is any element of $L$, the If $J$ were only a subalgebra instead of an ideal, we would have to restri $J$ rather than just in $L$.

In[∘]:=

As an example, consider the group U(3), the set of all three-by-th matrices. We can think of its Lie algebra as being the set of all Hermi by-three matrices. This is the same as for SU(3) except that the ma not be traceless. Thus we might take for a basis, the eight matrices d Eq. (III.1), plus the three-by-three identity matrix.

Out[∘]=

As an example, consider the group U(3), the set of all three-by-th matrices. We can think of its Lie algebra as being the set of all Hermi by-three matrices. This is the same as for SU(3) except that the ma not be traceless. Thus we might take for a basis, the eight matrices d Eq. (III.1), plus the three-by-three identity matrix.

Now consider the one-dimensional space spanned by the identity matrix, that is, the space given by multiples of the identity. This space, $J$, is an ideal because if $x \in J$ and $y$ is any element of the Lie algebra, $[x, y] = 0 \in J$. In fact, if we consider the space of all traceless matrices, $J'$, we see that it too is an ideal. This follows since the trace of a commutator is necessarily traceless. Thus every element in U(3) can be written as a sum of one element from $J$ and one element from $J'$. The full algebra is the sum of the two ideals.

In[∘]:=

A Lie algebra which has no ideals (except the trivial ones comprising the full algebra itself or the ideal consisting solely of 0) is called **simple**. A subalgebra in which all members commute is called **abelian**. An algebra with no abelian ideals is called **semi-simple**. Thus the Lie algebra of SU(3) is simple, while that of U(3) is neither simple nor semi-simple.

Out[∘]=

Now consider the one-dimensional space spanned by the identity matrix, that is, the space given by multiples of the identity. This space, $J$, is an ideal because if $x \in J$ and $y$ is any element of the Lie algebra, $[x, y] = 0 \in J$. In fact, if we consider the space of all traceless matrices, $J'$, we see that it too is an ideal. This follows since the trace of a commutator is necessarily traceless. Thus every element in U(3) can be written as a sum of one element from $J$ and one element from $J'$. The full algebra is the sum of the two ideals.

A Lie algebra which has no ideals (except the trivial ones comprising the full algebra itself or the ideal consisting solely of 0) is called **simple**. A subalgebra in which all members commute is called **abelian**. An algebra with no abelian ideals is called **semi-simple**. Thus the Lie algebra of SU(3) is simple, while that of U(3) is neither simple nor semi-simple.

In[∘]:=

A semi-simple Lie algebra is the sum of simple ideals. Consider, for example, the five-by-five traceless hermitian matrices which are zero except for two diagonal blocks, one three-by-three and one two-by-two. Suppose we consider only matrices where each of these two blocks is separately traceless. The resulting set is a Lie algebra which can be considered the sum of two ideals, one of which is isomorphic to SU(2) and the other of which is isomorphic to SU(3). If we require only that the sum of the traces of the two diagonal blocks vanish, the resulting algebra is larger, including matrices proportional to one whose diagonal elements are $\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, -\frac{1}{2}, -\frac{1}{2}$. This element and its multiples form an abelian ideal so this larger algebra $(SU(3) \times SU(2) \times U(1))$ is not semi-simple.

Out[∘]=

A semi-simple Lie algebra is the sum of simple ideals. Consider, for example, the five-by-five traceless hermitian matrices which are zero except for two diagonal blocks, one three-by-three and one two-by-two. Suppose we consider only matrices where each of these two blocks is separately traceless. The resulting set is a Lie algebra which can be considered the sum of two ideals, one of which is isomorphic to SU(2) and the other of which is isomorphic to SU(3). If we require only that the sum of the traces of the two diagonal blocks vanish, the resulting algebra is larger, including matrices proportional to one whose diagonal elements are $\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, -\frac{1}{2}, -\frac{1}{2}$. This element and its multiples form an abelian ideal so this larger algebra $(SU(3) \times SU(2) \times U(1))$ is not semi-simple.

Because semi-simple Lie algebras are simply sums of simple ones, most of their properties can be obtained by first considering simple Lie algebras.

In[•]:=

There is an intimate relationship between the Killing form and semi-simplicity: the Killing form is non-degenerate if and only if the Lie algebra is semi-simple. It is not hard to prove half of this fundamental theorem [2](which is due to Cartan): if the Killing form is non-degenerate, then L is semi-simple. Suppose $L$ is not semi-simple and let $B$ be an abelian ideal. Let $b_1, b_2, \ldots$ be a basis for $B$. We can extend this to a basis for the full algebra $L$ by adding $y_1, y_2, \ldots$ where $y_i \notin B$. Now let us calculate $(b_1, a)$ where $a \in L$. First consider $[b_1, [a, b_j]]$. The inner commutator lies in $B$ since $B$ is an ideal. But then the second commutator vanishes since $B$ is abelian. Next consider $[b_1, [a, y_j]]$ The final result must lie in $B$ since $b_1 \in B$ so its expansion has no components along the $y_k$'s and along $y_j$ in particular. Thus there is no contribution to the trace. The trace vanishes and the Killing form is degenerate.

Out[•]=

Because semi-simple Lie algebras are simply sums of simple ones, most of their properties can be obtained by first considering simple Lie algebras.

There is an intimate relationship between the Killing form and semi-simplicity: the Killing form is non-degenerate if and only if the Lie algebra is semi-simple. It is not hard to prove half of this fundamental theorem [2](which is due to Cartan): if the Killing form is non-degenerate, then L is semi-simple. Suppose $L$ is not semi-simple and let $B$ be an abelian ideal. Let $b_1, b_2, \ldots$ be a basis for $B$. We can extend this to a basis for the full algebra $L$ by adding $y_1, y_2, \ldots$ where $y_i \notin B$. Now let us calculate $(b_1, a)$ where $a \in L$. First consider $[b_1, [a, b_j]]$. The inner commutator lies in $B$ since $B$ is an ideal. But then the second commutator vanishes since $B$ is abelian. Next consider $[b_1, [a, y_j]]$ The final result must lie in $B$ since $b_1 \in B$ so its expansion has no components along the $y_k$'s and along $y_j$ in particular. Thus there is no contribution to the trace. The trace vanishes and the Killing form is degenerate.

In[•]:=

The roots of SU(3) exemplify a number of characteristics of semi-
algebras in general. First, if $\alpha$ is a root, so is $-\alpha$. This is made
Eq. (II.9), where we see that the root corresponding to $t_-$ is the negat
corresponding to $t_+$, and so on. Second, for each root, there is only o
independent generator with that root. Third, if $\alpha$ is a root, $2\alpha$ is not

How is the Cartan subalgebra determined in general? It turns o
following procedure is required. An element $h \in L$ is said to be **regular**
as few zero eigenvalues as possible, that is, the multiplicity of the zero
In[ ]:= is minimal. In the SU(3) example, from Eq. (II.8) we see that $\mathrm{ad}\, t_z$
dimensional space with eigenvalue zero, while $\mathrm{ad}\, y$ has a four dimension
this sort. The element $t_z$ is regular while $y$ is not. A Cartan subalgebra
by finding a maximal commutative subalgebra containing a regular ele
subalgebra generated by $t_z$ and $y$ is commutative and it is maximal sin
no other element we can add to it which would not destroy the commut

If we take as our basis for the algebra the **root vectors**, $e_{\alpha_1}$, $e$
some basis for the Cartan subalgebra, say $h_1, h_2 \ldots$, then we can writ
representation for $\mathrm{ad}\, h$:

Out[ ]=

The roots of SU(3) exemplify a number of characteristics of semi- algebras in general. First, if $\alpha$ is a root, so is $-\alpha$. This is made Eq. (II.9), where we see that the root corresponding to $t_-$ is the negat corresponding to $t_+$, and so on. Second, for each root, there is only o independent generator with that root. Third, if $\alpha$ is a root, $2\alpha$ is not

How is the Cartan subalgebra determined in general? It turns ou following procedure is required. An element $h \in L$ is said to be **regular** as few zero eigenvalues as possible, that is, the multiplicity of the zero is minimal. In the SU(3) example, from Eq. (II.8) we see that $\mathrm{ad}\, t_z$ dimensional space with eigenvalue zero, while $\mathrm{ad}\, y$ has a four dimension this sort. The element $t_z$ is regular while $y$ is not. A Cartan subalgebra by finding a maximal commutative subalgebra containing a regular ele subalgebra generated by $t_z$ and $y$ is commutative and it is maximal sir no other element we can add to it which would not destroy the commut

If we take as our basis for the algebra the **root vectors**, $e_{\alpha_1}$ , $e$ some basis for the Cartan subalgebra, say $h_1, h_2 \ldots$, then we can writ representation for $\mathrm{ad}\, h$:

In[ ]:=

$$\mathrm{ad}\, h = \begin{bmatrix} 0 & & & & & & & & \\ & 0 & & & & & & & \\ & & \cdot & & & & & & \\ & & & 0 & & & & & \\ & & & & \alpha_1(h) & & & & \\ & & & & & \alpha_2(h) & & & \\ & & & & & & \cdot & & \\ & & & & & & & \cdot & \\ & & & & & & & & \alpha_n(h) \end{bmatrix} . \qquad \text{(IV. 2)}$$

From this we can see that the Killing form, when acting on the Cartan subalgebra can be computed by

$$(h_1, h_2) = \sum_{\alpha \in \Sigma} \alpha(h_1)\alpha(h_2) \ , \qquad \text{(IV.3)}$$

where $\Sigma$ is the set of all the roots.

Out[ ]=

$$\mathrm{ad}\, h = \begin{bmatrix} 0 & & & & & & & & \\ & 0 & & & & & & & \\ & & \cdot & & & & & & \\ & & & 0 & & & & & \\ & & & & \alpha_1(h) & & & & \\ & & & & & \alpha_2(h) & & & \\ & & & & & & \cdot & & \\ & & & & & & & \cdot & \\ & & & & & & & & \alpha_n(h) \end{bmatrix} . \qquad \text{(IV. 2)}$$

From this we can see that the Killing form, when acting on the Cartan subalgebra can be computed by

$$(h_1, h_2) = \sum_{\alpha \in \Sigma} \alpha(h_1)\alpha(h_2) \ , \qquad \text{(IV.3)}$$

where $\Sigma$ is the set of all the roots.

## Commutation Relations of Root Vectors

Let Σ be the set of all roots of the Lie Algebra L. We know by now that $\Sigma \subset H^*$ (Dual space of Cartan subalgebra of L).
For a root $\alpha \in \Sigma$ (which implies $\alpha \in H^*$) and some element $h \in H$ we can write (for a root vector $e_\alpha \in$ L)

$$\mathrm{ad}_h\, e_\alpha = [h, e_\alpha] = \alpha(h)\, e_\alpha$$

which means that $e_\alpha$ is an eigenvector of the adjoint representation of any element in the Cartan subalgebra, with its eigenvalue defined by the dependency of h on the basis of H.

By Jacobi identity, we can move forward and write immediately that:

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

which implies one, and only one, of three things:
1. $[e_\alpha, e_\beta] = 0$
2. $[e_\alpha, e_\beta]$ is a root vector with root $\alpha + \beta$
3. $\alpha + \beta = 0$, in which case $[e_\alpha, e_\beta]$ commutes with h, and thus all elements of H. This by extension makes it an element of the Cartan subalgebra H.

We can show that $(e_\alpha, e_\beta) = 0$ unless $\alpha + \beta = 0$. To show this, consider the following construction.

for some arbitrary element $x \in L$, examine $[e_\alpha, [e_\beta, x]] = \mathrm{ad}_{e_\alpha}\mathrm{ad}_{e_\beta}x$. From this we can possibly determine the value of $(e_\alpha, e_\beta)$.
Now, if $x \in H$, then either 1 or 2 above. In either case, there is no contribution to the trace. This is because ?
If $x \notin H$, then x is a root vector. In this case, the double commutator is either zero or of the form $e_{\alpha+\beta+\gamma}$ and does not contribute to the trace unless $\alpha + \beta = 0$

## *Invariance of the Killing Form*

```
In[]:= R = MatrixRepSU3[{1, 1}];
Y = 2 R[[8]] / Sqrt[3];
Tp = (R[[1]] + I R[[2]]);
Tm = (R[[1]] - I R[[2]]);
Tz = R[[3]];
Vp = (R[[4]] + I R[[5]]);
Vm = (R[[4]] - I R[[5]]);
Upl = (R[[6]] + I R[[7]]);
Um = (R[[6]] - I R[[7]]);

In[]:= VecR = {Y, Tp, Tm, Tz, Vp, Vm, Upl, Um};

In[]:= LeftTable = Table[Tr[VecR[[i]].(VecR[[j]] × VecR[[k]] - VecR[[k]] × VecR[[j]])],
    {i, 1, 8 }, {j, 1, 8}, {k, 1, 8}];

In[]:= RightTable = Table[Tr[(VecR[[i]] × VecR[[j]] - VecR[[j]] × VecR[[i]]).VecR[[k]]],
    {i, 1, 8 }, {j, 1, 8}, {k, 1, 8}];
```

In[∘]:=
```
Difference = (LeftTable - RightTable) /. {{0, 0, 0, 0, 0, 0, 0, 0} → 0} // MatrixForm
```
Out[∘]//MatrixForm=

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

Killing Form is the only invariant bilinear form on a Simple Lie Algebra, up to constant scaling.
(a, [b, c]) = ([a,b], c) where a, b, c ∈ L.

In[∘]:=
```
CommutatorOfUPlusAndMinus = Upl.Um - Um.Upl
```
Out[∘]=
```
{{1, 0, 0, 0, 0, 0, 0, 0}, {0, 2, 0, 0, 0, 0, 0, 0},
 {0, 0, -1, 0, 0, 0, 0, 0}, {0, 0, 0, 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, -2, 0}, {0, 0, 0, 0, 0, 0, 0, -1}}
```

In[∘]:=
```
KillingFormOfUPlusAndMinus = Tr[Upl.Um]
```
Out[∘]=
```
6
```

In[∘]:=
```
h2 = -Tz / 6 + Y / 4;
```
In[∘]:=
```
Prod = KillingFormOfUPlusAndMinus * h2 // Simplify;
```
In[∘]:=
```
CommutatorOfUPlusAndMinus - Prod // Simplify // MatrixForm
```
Out[∘]//MatrixForm=

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

We have proved $[e_\alpha, e_{-\alpha}] = (e_\alpha, e_{-\alpha})\, h_\alpha$ which means that commutator of plus and minus ladder operators is equal to their Killing form multiplied by the root vector of the plus operator.

# Representation Theory of Lie Algebras

## Definitions

The elements $e_\alpha$, $e_{-\alpha}$, and $h_\alpha$ have the commutation relations
1. $[h_\alpha, e_\alpha] = \alpha(h_\alpha) e_\alpha = (h_\alpha, h_\alpha)\, e_\alpha = \langle\alpha,\alpha\rangle\, e_\alpha$
2. $[h_\alpha, e_{-\alpha}] = -\langle\alpha,\alpha\rangle\, e_\alpha$

3. $[e_\alpha, e_{-\alpha}] = (e_\alpha, e_{-\alpha})\, h_\alpha$

Consider representation of elements of L as GL(V,V) for some vector space V. Let the set $\{H_i\}$ denote the matrices corresponding to the elements of the Cartan subalgebra, and the set $\{E_\alpha\}$ represent the root vectors in L - H. Since all h ∈ H commute, all $H_i$ also commute. Hence, we can choose them to be diagonal in some basis of V.

In this basis, $H_i\, \phi^a = \lambda_i^a\, \phi^a$ for some eigenvector $\phi^a$. For some general $H = \Sigma_i\, c_i\, H_i$ we can define a functional $M^a(h) = \Sigma_i\, c_i\, \lambda_i^a$. These are functionals on H, and thus lie in the representation of the dual space $H^*$.

The functionals $M^a$ are called weights.

In the following, Tz and Y are the Gell-Mann ({1,0}) representations of the elements of the Cartan Subalgebra of SU(3). $\alpha$, $\beta$, $\gamma$ are the weight vectors corresponding to each weight.
$A_i$ give the weights in the {1,0} rep.
$B_i$ give the weights in the {0,1} rep.
{1,1} is the adjoint representation.

In[ ]:=
```
Tz = DiagonalMatrix[{1 / 2, -1 / 2, 0}]
```
Out[ ]=
$$\left\{\left\{\frac{1}{2}, 0, 0\right\}, \left\{0, -\frac{1}{2}, 0\right\}, \{0, 0, 0\}\right\}$$

In[ ]:=
```
Y = DiagonalMatrix[{1 / 3, 1 / 3, -2 / 3}]
```
Out[ ]=
$$\left\{\left\{\frac{1}{3}, 0, 0\right\}, \left\{0, \frac{1}{3}, 0\right\}, \left\{0, 0, -\frac{2}{3}\right\}\right\}$$

In[ ]:=
```
α = {1, 0, 0};
```
In[ ]:=
```
β = {0, 1, 0};
```
In[ ]:=
```
γ = {0, 0, 1};
```
In[ ]:=
```
H = a * Tz + b * Y
```
Out[ ]=
$$\left\{\left\{\frac{a}{2} + \frac{b}{3}, 0, 0\right\}, \left\{0, -\frac{a}{2} + \frac{b}{3}, 0\right\}, \left\{0, 0, -\frac{2\,b}{3}\right\}\right\}$$

In[ ]:=
```
H.α
```
Out[ ]=
$$\left\{\frac{a}{2} + \frac{b}{3}, 0, 0\right\}$$

In[ ]:=
```
H.β
```
Out[ ]=
$$\left\{0, -\frac{a}{2} + \frac{b}{3}, 0\right\}$$

In[◦]:= H.γ

Out[◦]=

$$\left\{0,\ 0,\ -\frac{2\,b}{3}\right\}$$

```
In[◦]:= A1={ (2*Al1)/3 + Al2/3, 0, 0};
```

$$\text{In[◦]:= } A2 = \left\{0,\ -\frac{Al1}{3} + \frac{Al2}{3},\ 0\right\};$$

$$\text{In[◦]:= } A3 = \left\{0,\ 0,\ -\frac{Al1}{3} - \frac{2\,Al2}{3}\right\};$$

$$\text{In[◦]:= } B1 = \left\{\frac{Al1}{3} + \frac{2\,Al2}{3},\ 0,\ 0\right\};$$

$$\text{In[◦]:= } B2 = \left\{0,\ \frac{Al1}{3} - \frac{Al2}{3},\ 0\right\};$$

$$\text{In[◦]:= } B3 = \left\{0,\ 0,\ -\frac{2\,Al1}{3} - \frac{Al2}{3}\right\};$$

```
In[◦]:= Al1 = {1, 0} / Sqrt[3]; Al2 = {Cos[120 Pi / 180], Sin[120 Pi / 180]} / Sqrt[3];
In[◦]:= p1 = A1[[1]]; p2 = A2[[2]]; p3 = A3[[3]];
In[◦]:= q1 = B1[[1]]; q2 = B2[[2]]; q3 = B3[[3]];
```

Let’s generalise the creation and annihilation operators in representation theory of Lie algebras.
**After this subsection, you will be able to talk about: 1. {a,b} representations of SU(3), 2. Weyl Symmetry and Weyl Group, 3. Weight Space**

Suppose that $\phi^a$ is a weight vector with weight $M^a$. Then,

$$\text{Weight}[E_\alpha \phi^a] = M^a + \alpha \quad \text{unless} \quad E_\alpha \phi^a = 0$$

This can be written more elegantly as

$$\begin{aligned} \mathrm{H} E_\alpha \phi^a &= (E_\alpha \mathrm{H} + \alpha(h)\, E_\alpha)\phi^a \\ &= (M^a(h) + \alpha(h)) E_\alpha \phi^a \end{aligned}$$

by noting that $[H, E_\alpha] = \alpha(h)\, E_\alpha$

If M is a weight, then it lies in some series M*, M* - $\alpha$, M* - 2$\alpha$ ... M* - q$\alpha$, where M* is the highest weight and each $E_{-\alpha}$ acting on the corresponding weight vector will take it to the lower weight.

Choosing a normalization $(e_\alpha, e_{-\alpha}) = 1$, we obtain $[E_\alpha, E_{-\alpha}] = H_\alpha$.

Follow equations V.11 to V.21 in Cahn to get that

$$E_\alpha\, \phi_k = r_k\, \phi_{k-1} \qquad \text{with} \qquad r_k = k\langle M^*,\alpha\rangle - (k(k-1)/2)\langle\alpha,\alpha\rangle$$

which gives

$$q = \frac{2\langle M^*,\alpha\rangle}{\langle\alpha,\alpha\rangle}$$

this gives us the allowed number of steps from one weight vector to another in the direction specified by $\alpha$.

In[ ]:=
```
p1.Al1
```
Out[ ]=
$$\frac{1}{6}$$

In[ ]:=
```
p2.Al1
```
Out[ ]=
$$-\frac{1}{6}$$

In[ ]:=
```
p3.Al1
```
Out[ ]=
0

In[ ]:=
```
2 Transpose[p1].Al1 / (Transpose[Al1].Al1)
```
Out[ ]=
1

In[ ]:=
```
2 Transpose[p2].Al1 / (Transpose[Al1].Al1)
```
Out[ ]=
-1

In[ ]:=
```
2 Transpose[(p3)].Al1 / (Transpose[Al1].Al1)
```
Out[ ]=
0

In[ ]:=
```
VecP = {p1, p2, p3}
```
Out[ ]=
$$\left\{\left\{\frac{1}{2\sqrt{3}}, \frac{1}{6}\right\}, \left\{-\frac{1}{2\sqrt{3}}, \frac{1}{6}\right\}, \left\{0, -\frac{1}{3}\right\}\right\}$$

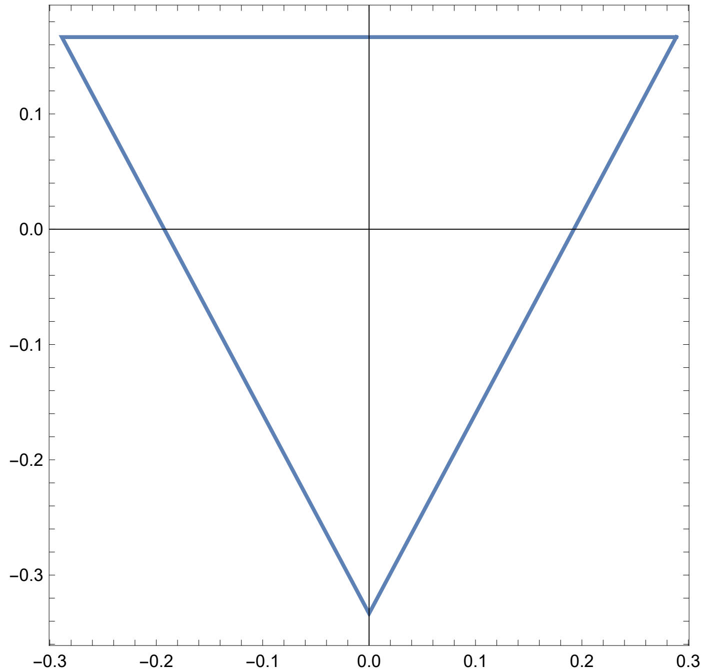
In[ ]:=
```
VecQ = {q1, q2, q3}
```
Out[ ]=
$$\left\{\left\{0, \frac{1}{3}\right\}, \left\{\frac{1}{2\sqrt{3}}, -\frac{1}{6}\right\}, \left\{-\frac{1}{2\sqrt{3}}, -\frac{1}{6}\right\}\right\}$$

In[ ]:=
```
VecAl = {Al1, Al2}
```
Out[ ]=
$$\left\{\left\{\frac{1}{\sqrt{3}}, 0\right\}, \left\{-\frac{1}{2\sqrt{3}}, \frac{1}{2}\right\}\right\}$$

In[ ]:=
```
AllowedStepsP = Table[2 Transpose[VecP[[i]]].VecAl[[j]] / (Transpose[VecAl[[j]]].VecAl[[j]]),
    {i, 1, 3}, {j, 1, 2}] // Simplify;
```

In[ ]:=
```
AllowedStepsP // MatrixForm
```
Out[ ]//MatrixForm=
$$\begin{pmatrix} 1 & 0 \\ -1 & 1 \\ 0 & -1 \end{pmatrix}$$

In[ ]:= 
```
ListPlot[{p1, p2, p3, p1}, Joined → True, Axes → True, Frame → True, AspectRatio → 1]
```
Out[ ]=

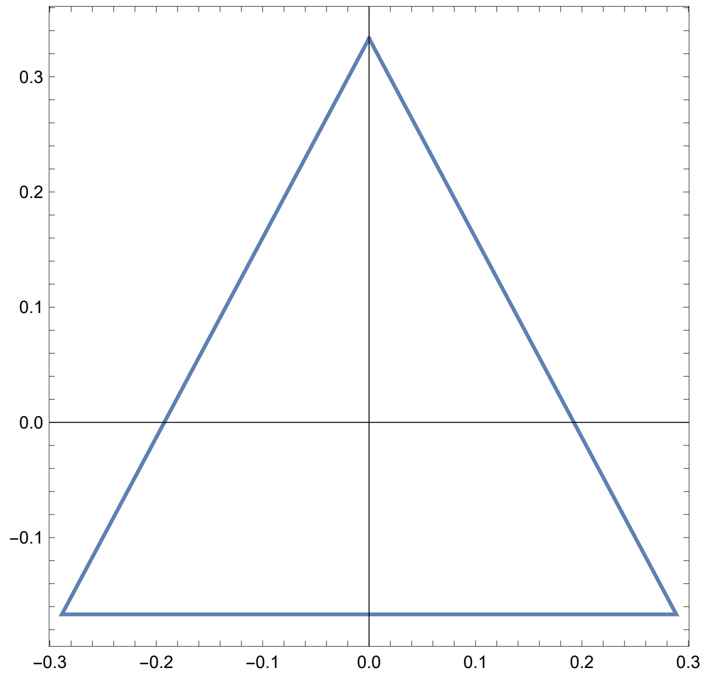


In[ ]:= 
```
AllowedStepsQ = Table[2 Transpose[VecQ[[i]]].VecAl[[j]] / (Transpose[VecAl[[j]]].VecAl[[j]]),
    {i, 1, 3}, {j, 1, 2}] // Simplify;
```

In[ ]:= 
```
AllowedStepsQ // MatrixForm
```
Out[ ]//MatrixForm=

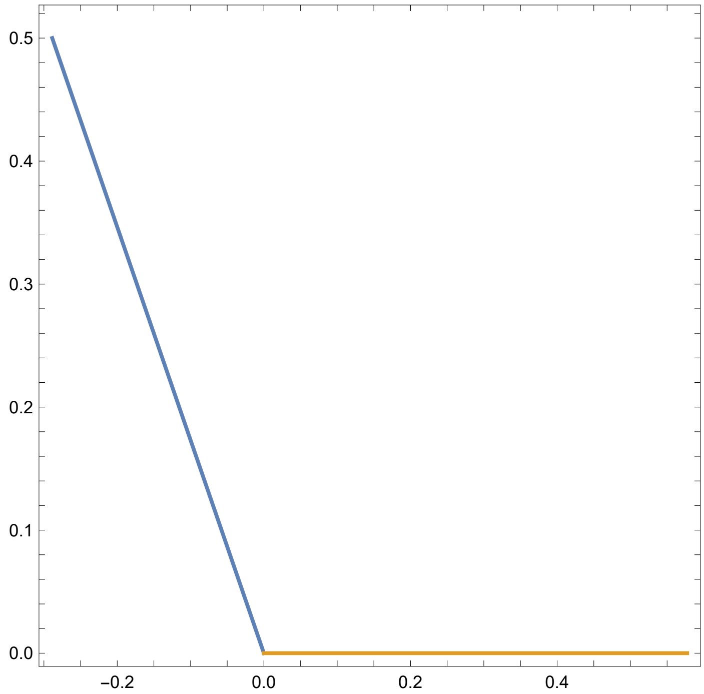
$$\begin{pmatrix} 0 & 1 \\ 1 & -1 \\ -1 & 0 \end{pmatrix}$$

In[ ]:= 
```
ListPlot[{q1, q2, q3, q1}, Joined → True, Axes → True, Frame → True, AspectRatio → 1]
```
Out[ ]=



In[∘]:=
```
ListPlot[{{Al2, {0, 0}}, {Al1, {0, 0}}},
 Joined → True, Axes → False, Frame → True, AspectRatio → 1]
```

Out[∘]=




In[∘]:=
```
SetGroup[SU[3]]
```

Set group is A(2)

Rank of the group = 2

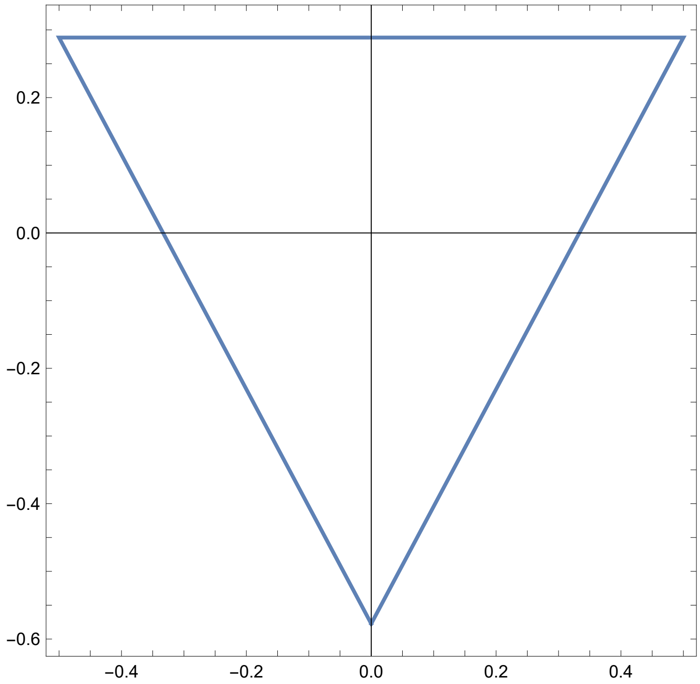
In[∘]:=
```
rep = BuildRep[{1, 0}, OutputMethod → "Weights"] // PhysicalNormalization
```

Out[∘]=

$$\left\{\left\{\left\{0, -\frac{1}{\sqrt{3}}\right\}, 1\right\}, \left\{\left\{-\frac{1}{2}, \frac{1}{2\sqrt{3}}\right\}, 1\right\}, \left\{\left\{\frac{1}{2}, \frac{1}{2\sqrt{3}}\right\}, 1\right\}\right\}$$

In[•]:=
```
ListPlot[{rep〚1〛〚1〛, rep〚2〛〚1〛, rep〚3〛〚1〛, rep〚1〛〚1〛},
 Joined → True, Axes → True, Frame → True, AspectRatio → 1]
```

Out[•]=




In[•]:=
```
{p1, p2, p3} * (Sqrt[3]) // Simplify
```

Out[•]=

$$\left\{\left\{\frac{1}{2}, \frac{1}{2\sqrt{3}}\right\}, \left\{-\frac{1}{2}, \frac{1}{2\sqrt{3}}\right\}, \left\{0, -\frac{1}{\sqrt{3}}\right\}\right\}$$

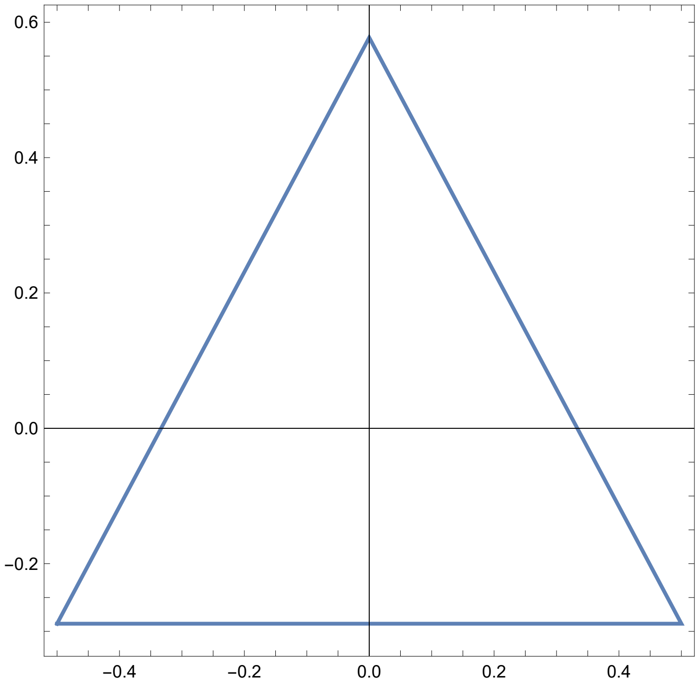
In[•]:=
```
antirep = BuildRep[{0, 1}, OutputMethod → "Weights"] // PhysicalNormalization
```

Out[•]=

$$\left\{\left\{\left\{-\frac{1}{2}, -\frac{1}{2\sqrt{3}}\right\}, 1\right\}, \left\{\left\{\frac{1}{2}, -\frac{1}{2\sqrt{3}}\right\}, 1\right\}, \left\{\left\{0, \frac{1}{\sqrt{3}}\right\}, 1\right\}\right\}$$

In[ ]:=
```
ListPlot[{antirep[[1]][[1]], antirep[[2]][[1]], antirep[[3]][[1]], antirep[[1]][[1]]},
 Joined → True, Axes → True, Frame → True, AspectRatio → 1]
```

Out[ ]=




In[ ]:=
```
{q1, q2, q3} * Sqrt[3] // Simplify
```

Out[ ]=

$$\left\{\left\{0, \frac{1}{\sqrt{3}}\right\}, \left\{\frac{1}{2}, -\frac{1}{2\sqrt{3}}\right\}, \left\{-\frac{1}{2}, -\frac{1}{2\sqrt{3}}\right\}\right\}$$

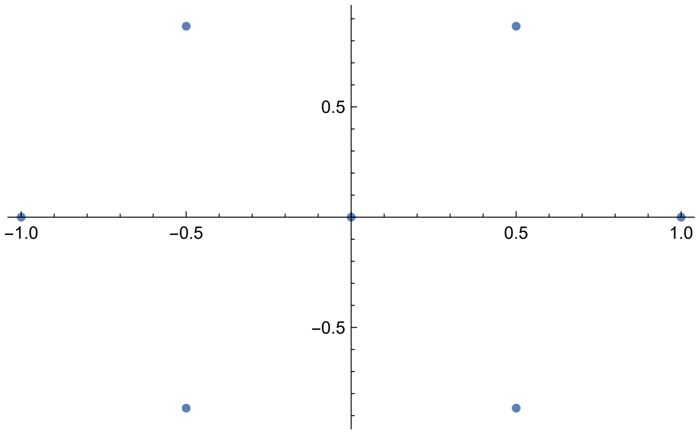
In[ ]:=
```
adj =
 Table[(BuildRep[{1, 1}, OutputMethod → "Weights"] // PhysicalNormalization)[[i]][[1]],
  {i, 1, 7}]
```

Out[ ]=

$$\left\{\left\{-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right\}, \left\{\frac{1}{2}, -\frac{\sqrt{3}}{2}\right\}, \{0, 0\}, \{-1, 0\}, \{1, 0\}, \left\{-\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}, \left\{\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}\right\}$$

In[ ]:=
```
ListPlot[adj]
```

Out[ ]=



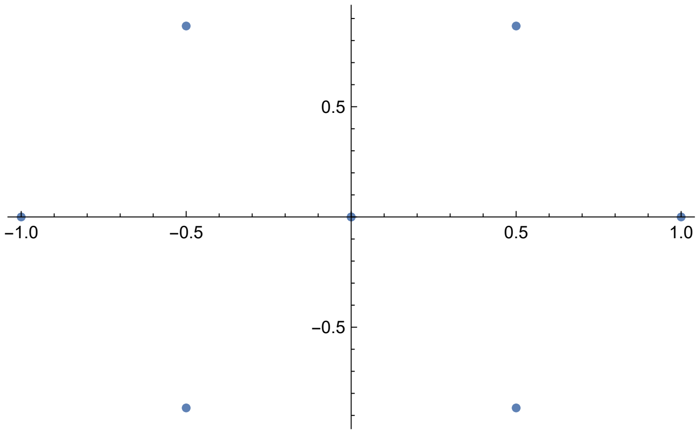
In[∘]:= adj$_{actual}$ = AppendTo[adj, {0, 0}]

Out[∘]=

$$\left\{\left\{-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right\}, \left\{\frac{1}{2}, -\frac{\sqrt{3}}{2}\right\}, \{0, 0\}, \{-1, 0\}, \{1, 0\}, \left\{-\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}, \left\{\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}, \{0, 0\}\right\}$$

In[∘]:= ListPlot[adj$_{actual}$] (*this is shown to remind that {0,0} is present two times and the adjoint representation is eight dimensional*)

Out[∘]=




In[∘]:= {Al1, Al2, Al1 + Al2, -Al1, -Al2, -Al1 - Al2} Sqrt[3] // Simplify

Out[∘]=

$$\left\{\{1, 0\}, \left\{-\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}, \left\{\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}, \{-1, 0\}, \left\{\frac{1}{2}, -\frac{\sqrt{3}}{2}\right\}, \left\{-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right\}\right\}$$

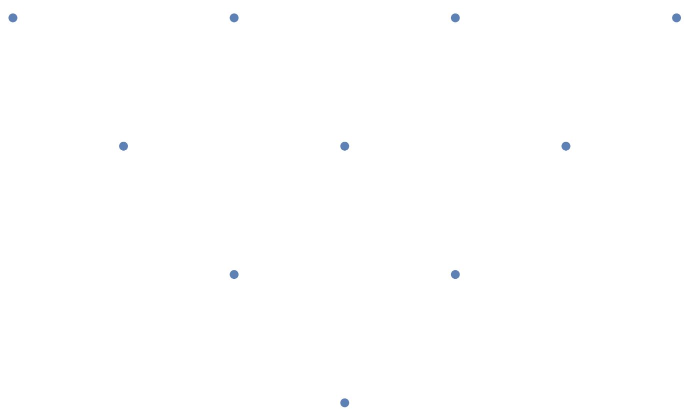
In[∘]:= dec =
 Table[(BuildRep[{3, 0}, OutputMethod → "Weights"] // PhysicalNormalization)⟦i⟧⟦1⟧,
  {i, 1, 10}]

Out[∘]=

$$\left\{\{0, -\sqrt{3}\}, \left\{-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right\}, \left\{\frac{1}{2}, -\frac{\sqrt{3}}{2}\right\}, \{-1, 0\}, \{0, 0\}, \{1, 0\}, \left\{-\frac{3}{2}, \frac{\sqrt{3}}{2}\right\}, \left\{-\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}, \left\{\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}, \left\{\frac{3}{2}, \frac{\sqrt{3}}{2}\right\}\right\}$$

In[∘]:= ListPlot[dec, Axes → False]

Out[∘]=



In[∘]:= Al3 = Al1 + Al2; Clear[VecAl]; VecAl = {Al1, Al2, Al3};

```
In[•]:= AllowedStepsDec =
         Table[2 Transpose[dec[[i]]].VecAl[[j]] / (Transpose[VecAl[[j]]].VecAl[[j]]),
           {i, 1, 10}, {j, 1, 3}] / Sqrt[3] // Simplify
Out[•]=
        {{0, -3, -3}, {-1, -1, -2}, {1, -2, -1}, {-2, 1, -1},
         {0, 0, 0}, {2, -1, 1}, {-3, 3, 0}, {-1, 2, 1}, {1, 1, 2}, {3, 0, 3}}

In[•]:= AllowedStepsDec // MatrixForm
Out[•]//MatrixForm=
```

$$\begin{pmatrix} 0 & -3 & -3 \\ -1 & -1 & -2 \\ 1 & -2 & -1 \\ -2 & 1 & -1 \\ 0 & 0 & 0 \\ 2 & -1 & 1 \\ -3 & 3 & 0 \\ -1 & 2 & 1 \\ 1 & 1 & 2 \\ 3 & 0 & 3 \end{pmatrix}$$

```
In[•]:= dec =
         Table[(BuildRep[{0, 3}, OutputMethod → "Weights"] // PhysicalNormalization)[[i]][[1]],
          {i, 1, 10}]
Out[•]=
```

$$\left\{\left\{-\frac{3}{2}, -\frac{\sqrt{3}}{2}\right\}, \left\{-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right\}, \left\{\frac{1}{2}, -\frac{\sqrt{3}}{2}\right\}, \left\{\frac{3}{2}, -\frac{\sqrt{3}}{2}\right\}, \{-1, 0\}, \{0, 0\}, \{1, 0\}, \left\{-\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}, \left\{\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}, \left\{0, \sqrt{3}\right\}\right\}$$

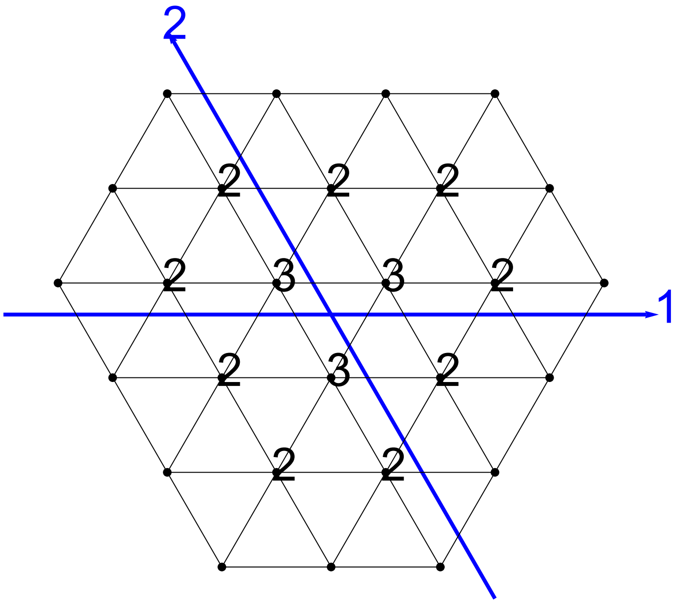
```
In[•]:= AllowedStepsDec =
          Table[2 Transpose[dec[[i]]].VecAl[[j]] / (Transpose[VecAl[[j]]].VecAl[[j]]),
            {i, 1, 10}, {j, 1, 3}] / Sqrt[3] // Simplify;

In[•]:= AllowedStepsDec // MatrixForm
Out[•]//MatrixForm=
```

$$\begin{pmatrix} -3 & 0 & -3 \\ -1 & -1 & -2 \\ 1 & -2 & -1 \\ 3 & -3 & 0 \\ -2 & 1 & -1 \\ 0 & 0 & 0 \\ 2 & -1 & 1 \\ -1 & 2 & 1 \\ 1 & 1 & 2 \\ 0 & 3 & 3 \end{pmatrix}$$

```
In[•]:= Dyn32 = BuildRep[{3, 2}, OutputMethod → "Weights"];
```

In[ ]:= PlotRep2D[Dyn32, SimpleRoots]

Out[ ]=




In[ ]:= Clear[adj]

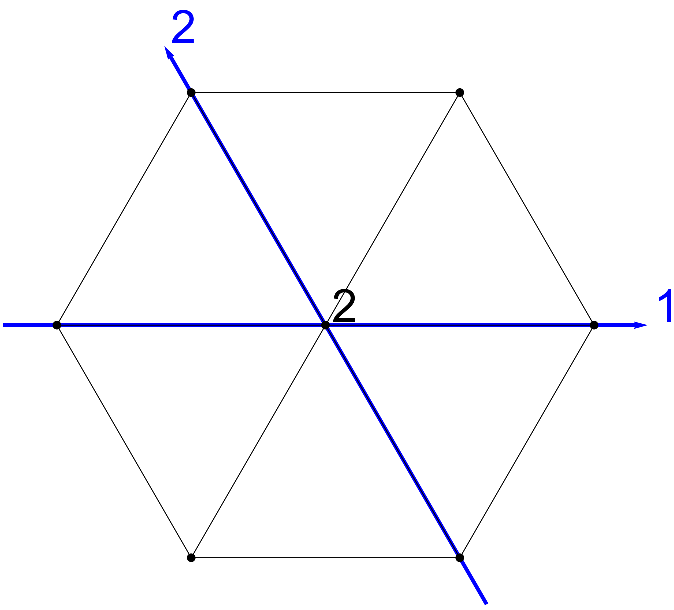
In[ ]:= adj = BuildRep[{1, 1}, OutputMethod → "Weights"]

Out[ ]=

{{{-1, 0, 1}, 1}, {{0, -1, 1}, 1}, {{0, 0, 0}, 2}, {{-1, 1, 0}, 1}, {{1, -1, 0}, 1}, {{0, 1, -1}, 1}, {{1, 0, -1}, 1}}

In[ ]:= PlotRep2D[adj, SimpleRoots]

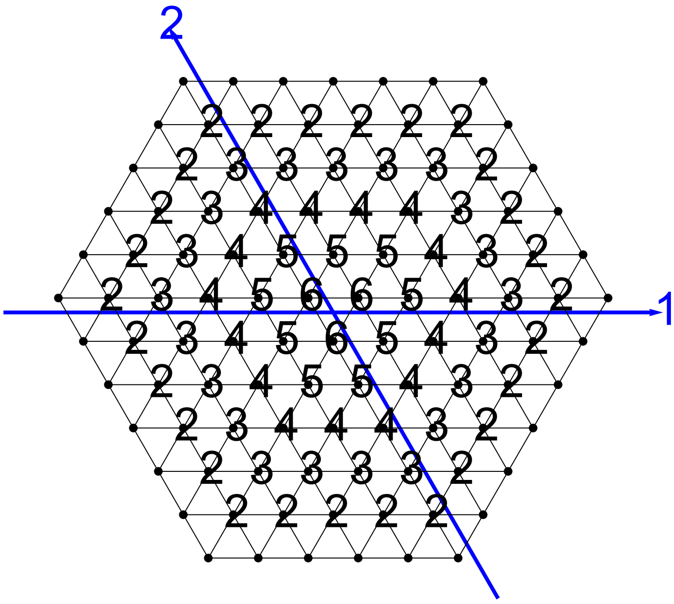
Out[ ]=




In[ ]:= Dyn65 = BuildRep[{6, 5}, OutputMethod → "Weights"];

In[∘]:= `PlotRep2D[Dyn65, SimpleRoots]`

Out[∘]=




## Weyl Group

Symmetries in the representations of Lie Algebras is immediately apparent in the above examples.

Example, the minimal symmetry group in representations of SU(3) is $S_2$ which is deduced from {1,0} and {0,1} reps.

A method to formally develop these group characteristics of reps of Lie Algebras is by using a symmetry which maps some weight vector to another weight vector.
$S_\alpha : M \rightarrow M - \frac{2<M,\alpha>}{<\alpha,\alpha>}\alpha$ is an operation which maps some weight to another. This is called **Weyl Reflection**.

We construct the Weyl Group for irreps of Lie algebras.

In[ ]:=

**Weyl Group meaning**

Input interpretation:

Weyl group

Definition:

More details

Let $\mathcal{L}$ be a finite-dimensional split semisimple Lie algebra over a field of field characteristic 0, $\mathcal{H}$ a splitting Cartan subalgebra, and $\Lambda$ a weight of $\mathcal{H}$ in a representation of $\mathcal{L}$. Then

$$\Lambda' = \Lambda S_\alpha = \lambda - \frac{2(\Lambda,\alpha)}{(\alpha,\alpha)}(\alpha)$$

is also a weight. Furthermore, the reflections $S_\alpha$ with $\alpha$ a root, generate a group of linear transformations in $\mathcal{H}_0^*$ called the Weyl group $W$ of $\mathcal{L}$ relative to $\mathcal{H}$, where $\mathcal{H}^*$ is the algebraic conjugate space of $\mathcal{H}$ and $\mathcal{H}_0^*$ is the $Q$-space spanned by the roots (Jacobson 1979, pp. 112, 117, and 119).

More information »

Related terms:

Cartan matrix | Dynkin diagram | Lie algebra | Lie algebra root | Lie group | Macdonald's constant-term conjecture | root lattice | root system | semisimple Lie algebra

Subject classifications:

Show details

MathWorld:

Lie algebra | finite groups

MSC 2010:

17Bxx | 20Dxx

Associated person:

Hermann Weyl

WolframAlpha

**Some Facts,** *cos who wants to prove stuff when you can just demonstrate it?*

1. If $\alpha \in \Sigma$, then $<\alpha,\alpha> \neq 0$

*we assumed above that our definition of inner product is a valid one, but is it though? you have to prove it using* ***math!***

2. If $\alpha \in \Sigma$, then $\{\alpha, -\alpha, 0\} \subset \Sigma$

*not 2α or some other weird scaling of α, only α, its negative, and zero. this is true for all roots in the root space Σ*

3. $\exists$ only one linearly independent $e_\alpha$ $\forall$ $\alpha \in \Sigma$

*every weight space (except the root zero space which is the Cartan subalgebra H) is one-dimensional fr*

**Allowed steps in general**

Assume that you are sitting at some arbitrary weight M. This weight M lies in a series M+p$\alpha$ ... M ... M-m$\alpha$.

We know from before that m + p = $\frac{2<M+p\alpha,\alpha>}{<\alpha,\alpha>}$. This is the total number of steps in this series. It can go p steps up and m steps down.

$$m - p = \frac{2\langle M,\alpha\rangle}{\langle\alpha,\alpha\rangle}$$

gives us the difference in the number of steps to go on either side.

**Why is this important?**

*Consider the adjoint representation of some arbitrary Lie algebra L.*

We can show that there can exist a maximum of four weights in a string (along any given root) by the above argument.

Assume that there are five weights WLOG labelled as $\beta$-2$\alpha$, $\beta$-$\alpha$, $\beta$, $\beta$+$\alpha$, $\beta$+2$\alpha$. Start at some root, say $\beta$+2$\alpha$, and take steps in the $\beta$ direction. In this direction, it is the only weight *(because 2$\alpha$ and 2($\beta$+$\alpha$) are not allowed)*. So,

$\langle\beta+2\alpha, \beta\rangle = 0$ and $\langle\beta-2\alpha, \beta\rangle = 0$ simultaneously

which implies that $\langle\beta,\beta\rangle = 0$, which is impossible. **Five weights in a single string are not allowed. Higher also not allowed.**

Now, if the $\alpha$-string of four elements.

$\beta + 2\alpha$ is a single element in its $\beta$ string

$\beta + \alpha$ is allowed to be a part of any four element $\beta$-string (which constrains m - p to ±3 or ±1)

$\beta$ behaves like $\beta + 2\alpha$

$\beta - \alpha$ behaves like $\beta + \alpha$

Simply the results are:

1. for 4-string m - p = ±3 or ±1
2. for 3-string m - p = ±2 or 0
3. for 2-string m - p = ±1
4. for 1-string m - p = 0

**Restriction to possible angles!**

Applying math twice: $\frac{\langle\alpha,\beta\rangle\langle\alpha,\beta\rangle}{\langle\alpha,\alpha\rangle\langle\beta,\beta\rangle} = \frac{mn}{4} = \cos^2\theta_{\alpha\beta}$

and we know the allowed values of m and n. So,the only allowed values of $\cos^2\theta_{\alpha\beta}$ are 0, $\frac{1}{4}$, $\frac{1}{2}$, $\frac{3}{4}$.

**Rational combinations of roots!**

Let $\beta = \Sigma_i\, q_i \alpha_i$. Applying math again gives us

$\frac{2\langle\beta,\alpha_j\rangle}{\langle\alpha_j,\alpha_j\rangle} = \Sigma_i\, q_i \frac{2\langle\alpha_i,\alpha_j\rangle}{\langle\alpha_j,\alpha_j\rangle}$ which is a linear equation in $q_i$ with all other terms rational. Moreover, $q_i$ come out to be integers.

**$\langle\alpha,\beta\rangle$ is rational always!**

## The Length and Angle between simple roots are correlated (pp. 129, Ramond)

In[∘]:=
```
{{n, npr, ArcCos[Sqrt[n npr / 4]], 1 / Sqrt[n / npr]} /. {n → 1, npr → 1},
  {n, npr, ArcCos[-Sqrt[n npr / 4]], 1 / Sqrt[n / npr]} /. {n → -1, npr → -1},
  {n, npr, ArcCos[Sqrt[n npr / 4]], 1 / Sqrt[n / npr]} /. {n → 1, npr → 2},
  {n, npr, ArcCos[-Sqrt[n npr / 4]], 1 / Sqrt[n / npr]} /. {n → -1, npr → -2},
  {n, npr, ArcCos[Sqrt[n npr / 4]], 1 / Sqrt[n / npr]} /. {n → 1, npr → 3},
  {n, npr, ArcCos[-Sqrt[n npr / 4]], 1 / Sqrt[n / npr]} /. {n → -1, npr → -3}} // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & \frac{\pi}{3} & 1 \\ -1 & -1 & \frac{2\pi}{3} & 1 \\ 1 & 2 & \frac{\pi}{4} & \sqrt{2} \\ -1 & -2 & \frac{3\pi}{4} & \sqrt{2} \\ 1 & 3 & \frac{\pi}{6} & \sqrt{3} \\ -1 & -3 & \frac{5\pi}{6} & \sqrt{3} \end{pmatrix}$$

# SU(3)

In[∘]:=
```
SetGroup[SU[3]]
```

Set group is A(2)

Rank of the group = 2

In[∘]:=
```
SimpleRoots // PhysicalNormalization
```

Out[∘]=

$$\left\{\{1, 0\}, \left\{-\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}\right\}$$

In[∘]:=
```
PosRoots = PositiveRoots // PhysicalNormalization
```

Out[∘]=

$$\left\{\{1, 0\}, \left\{\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}, \left\{-\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}\right\}$$

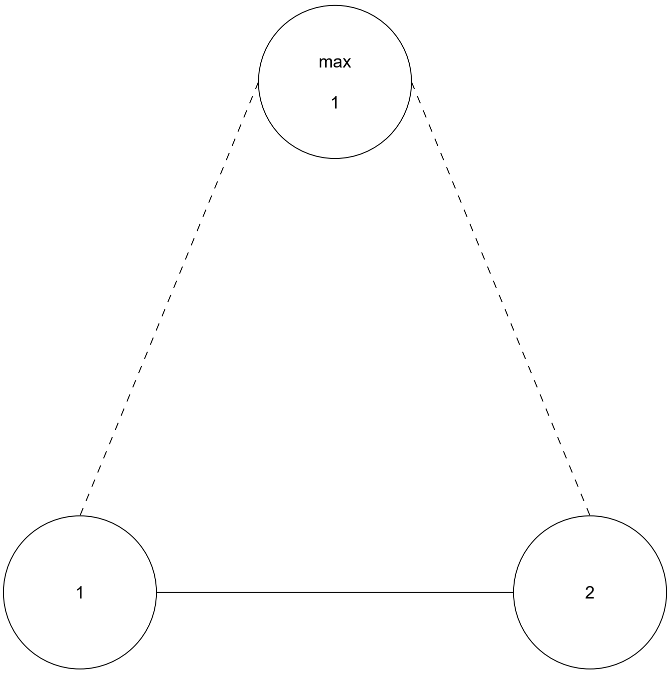
In[∘]:=
```
CartanMatrix[SimpleRoots] // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$$

In[ ]:= **DynkinDiagram**

Out[ ]=




In[ ]:= **rep = BuildRep[{1, 0}, OutputMethod → "Weights"]**

Out[ ]=

$$\left\{\left\{\left\{-\frac{1}{3}, -\frac{1}{3}, \frac{2}{3}\right\}, 1\right\}, \left\{\left\{-\frac{1}{3}, \frac{2}{3}, -\frac{1}{3}\right\}, 1\right\}, \left\{\left\{\frac{2}{3}, -\frac{1}{3}, -\frac{1}{3}\right\}, 1\right\}\right\}$$

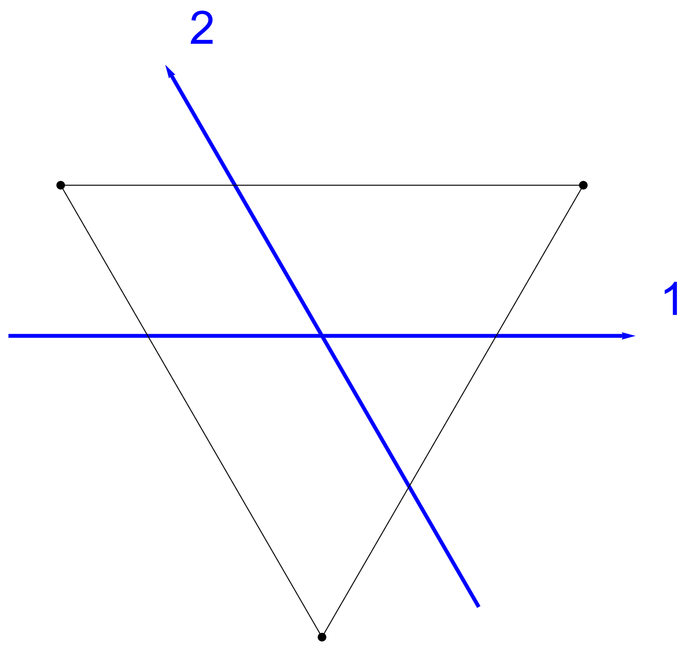
In[ ]:= **antirep = BuildRep[{0, 1}, OutputMethod → "Weights"]**

Out[ ]=

$$\left\{\left\{\left\{-\frac{2}{3}, \frac{1}{3}, \frac{1}{3}\right\}, 1\right\}, \left\{\left\{\frac{1}{3}, -\frac{2}{3}, \frac{1}{3}\right\}, 1\right\}, \left\{\left\{\frac{1}{3}, \frac{1}{3}, -\frac{2}{3}\right\}, 1\right\}\right\}$$

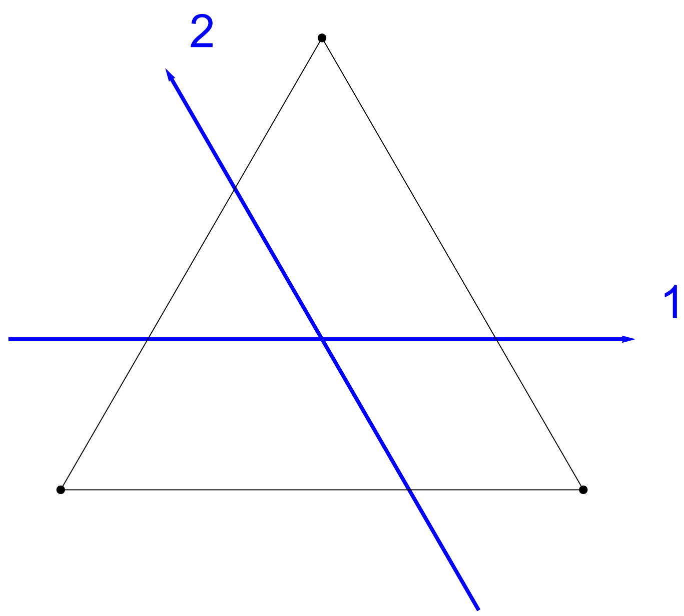
In[ ]:= `PlotRep2D[rep, SimpleRoots]`

Out[ ]=




In[ ]:= `PlotRep2D[antirep, SimpleRoots]`

Out[ ]=




In[ ]:= `rep = BuildRep[{1, 0}]`

Out[ ]=

```
{{{0, -1}, 1}, {{-1, 1}, 1}, {{1, 0}, 1}}
```

In[ ]:= `CGDecomposition[rep, rep]`

$3 \otimes 3 = 6 \oplus 3$

Out[ ]=

```
{{2, 0}, {0, 1}}
```

In[]:= **CGDecomposition[rep, rep, rep]**

$3 \otimes 3 \otimes 3 = 10 \oplus 8 \oplus 8 \oplus 1$

Out[]=

```
{{3, 0}, {1, 1}, {1, 1}, {0, 0}}
```

In[]:= **antirep = BuildRep[{0, 1}]**

Out[]=

```
{{{-1, 0}, 1}, {{1, -1}, 1}, {{0, 1}, 1}}
```

In[]:= **CGDecomposition[rep, antirep]**

$3 \otimes 3 = 8 \oplus 1$

Out[]=

```
{{1, 1}, {0, 0}}
```

In[]:= **CGDecomposition[antirep, rep]**

$3 \otimes 3 = 8 \oplus 1$

Out[]=

```
{{1, 1}, {0, 0}}
```

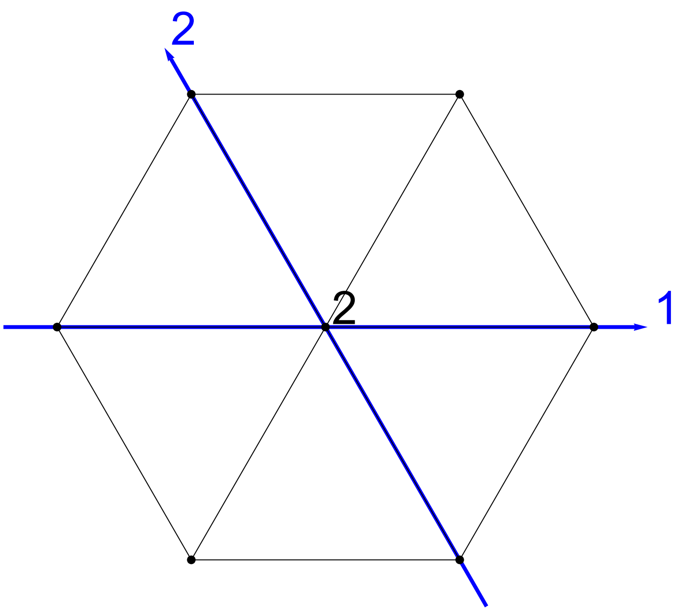
In[]:= **rep2 = BuildRep[{1, 1}, OutputMethod → "Weights"]**

Out[]=

```
{{{-1, 0, 1}, 1}, {{0, -1, 1}, 1}, {{0, 0, 0}, 2},
 {{-1, 1, 0}, 1}, {{1, -1, 0}, 1}, {{0, 1, -1}, 1}, {{1, 0, -1}, 1}}
```

In[]:= **PlotRep2D[rep2, SimpleRoots]**

Out[]=



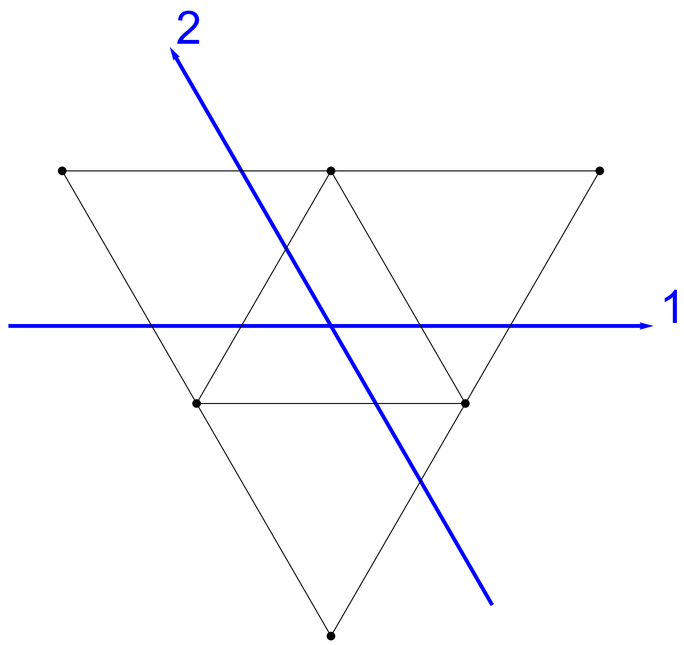

In[]:= **rep3 = BuildRep[{2, 0}, OutputMethod → "Weights"]**

Out[]=

$$\left\{\left\{\left\{-\frac{2}{3}, -\frac{2}{3}, \frac{4}{3}\right\}, 1\right\}, \left\{\left\{-\frac{2}{3}, \frac{1}{3}, \frac{1}{3}\right\}, 1\right\}, \left\{\left\{\frac{1}{3}, -\frac{2}{3}, \frac{1}{3}\right\}, 1\right\},\right.$$
$$\left.\left\{\left\{-\frac{2}{3}, \frac{4}{3}, -\frac{2}{3}\right\}, 1\right\}, \left\{\left\{\frac{1}{3}, \frac{1}{3}, -\frac{2}{3}\right\}, 1\right\}, \left\{\left\{\frac{4}{3}, -\frac{2}{3}, -\frac{2}{3}\right\}, 1\right\}\right\}$$

In[ ]:= 
```
PlotRep2D[rep3, SimpleRoots]
```

Out[ ]=




```
rep4 = BuildRep[{3, 1}, OutputMethod → "Weights"];
```

Out[ ]=

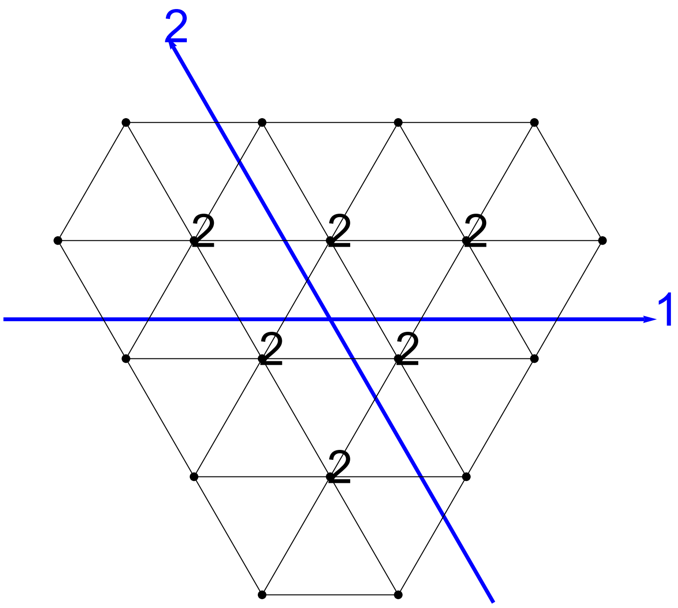
$$\left\{\left\{\left\{-\frac{5}{3}, -\frac{2}{3}, \frac{7}{3}\right\}, 1\right\}, \left\{\left\{-\frac{2}{3}, -\frac{5}{3}, \frac{7}{3}\right\}, 1\right\}, \left\{\left\{-\frac{2}{3}, -\frac{2}{3}, \frac{4}{3}\right\}, 2\right\},\right.$$
$$\left\{\left\{-\frac{5}{3}, \frac{1}{3}, \frac{4}{3}\right\}, 1\right\}, \left\{\left\{\frac{1}{3}, -\frac{5}{3}, \frac{4}{3}\right\}, 1\right\}, \left\{\left\{-\frac{2}{3}, \frac{1}{3}, \frac{1}{3}\right\}, 2\right\}, \left\{\left\{\frac{1}{3}, -\frac{2}{3}, \frac{1}{3}\right\}, 2\right\},$$
$$\left\{\left\{-\frac{5}{3}, \frac{4}{3}, \frac{1}{3}\right\}, 1\right\}, \left\{\left\{\frac{4}{3}, -\frac{5}{3}, \frac{1}{3}\right\}, 1\right\}, \left\{\left\{-\frac{2}{3}, \frac{4}{3}, -\frac{2}{3}\right\}, 2\right\}, \left\{\left\{\frac{1}{3}, \frac{1}{3}, -\frac{2}{3}\right\}, 2\right\},$$
$$\left\{\left\{\frac{4}{3}, -\frac{2}{3}, -\frac{2}{3}\right\}, 2\right\}, \left\{\left\{-\frac{5}{3}, \frac{7}{3}, -\frac{2}{3}\right\}, 1\right\}, \left\{\left\{\frac{7}{3}, -\frac{5}{3}, -\frac{2}{3}\right\}, 1\right\},$$
$$\left.\left\{\left\{-\frac{2}{3}, \frac{7}{3}, -\frac{5}{3}\right\}, 1\right\}, \left\{\left\{\frac{1}{3}, \frac{4}{3}, -\frac{5}{3}\right\}, 1\right\}, \left\{\left\{\frac{4}{3}, \frac{1}{3}, -\frac{5}{3}\right\}, 1\right\}, \left\{\left\{\frac{7}{3}, -\frac{2}{3}, -\frac{5}{3}\right\}, 1\right\}\right\}$$

In[•]:=
```
PlotRep2D[rep4, SimpleRoots]
```

Max: Internal precision limit $MaxExtraPrecision = 50.` reached while evaluating

$$-\frac{2+\sqrt{\frac{2}{3}}\left(\sqrt{\frac{2}{3}}+\frac{1}{\sqrt{6}}\right)}{\frac{1}{2}+\left(\sqrt{\frac{2}{3}}+\frac{1}{\sqrt{6}}\right)^2}+\frac{-\frac{\frac{1}{3\sqrt{2}}-\frac{2\sqrt{2}}{3}}{\sqrt{2}}+\left(\frac{7\sqrt{\frac{2}{3}}}{3}+\frac{1}{3\sqrt{6}}\right)\left(\sqrt{\frac{2}{3}}+\frac{1}{\sqrt{6}}\right)}{\frac{1}{2}+\left(\sqrt{\frac{2}{3}}+\frac{1}{\sqrt{6}}\right)^2}.$$

Out[•]=




```
Clear[rep, antirep, rep2, rep3, rep4]
```

In[•]:=
```
rep = BuildRep[{1, 0}]
```

Out[•]=
```
{{{0, -1}, 1}, {{-1, 1}, 1}, {{1, 0}, 1}}
```

In[•]:=
```
rep2 = BuildRep[{1, 1}]
```

Out[•]=
```
{{{-1, -1}, 1}, {{1, -2}, 1}, {{0, 0}, 2},
 {{-2, 1}, 1}, {{2, -1}, 1}, {{-1, 2}, 1}, {{1, 1}, 1}}
```

In[•]:=
```
rep3 = BuildRep[{2, 0}]
```

Out[•]=
```
{{{0, -2}, 1}, {{-1, 0}, 1}, {{1, -1}, 1}, {{-2, 2}, 1}, {{0, 1}, 1}, {{2, 0}, 1}}
```

In[•]:=
```
rep4 = BuildRep[{3, 1}]
```

Out[•]=
```
{{{-1, -3}, 1}, {{1, -4}, 1}, {{0, -2}, 2}, {{-2, -1}, 1}, {{2, -3}, 1}, {{-1, 0}, 2},
 {{1, -1}, 2}, {{-3, 1}, 1}, {{3, -2}, 1}, {{-2, 2}, 2}, {{0, 1}, 2}, {{2, 0}, 2},
 {{-4, 3}, 1}, {{4, -1}, 1}, {{-3, 4}, 1}, {{-1, 3}, 1}, {{1, 2}, 1}, {{3, 1}, 1}}
```

In[•]:=
```
CGDecomposition[rep, rep2, rep3]
```

3 ⊗ 8 ⊗ 6 = 35 ⊕ 27 ⊕ 27 ⊕ 10 ⊕ 10 ⊕ 10 ⊕ 8 ⊕ 8 ⊕ 8 ⊕ 1

Out[•]=
```
{{4, 1}, {2, 2}, {2, 2}, {3, 0}, {3, 0}, {0, 3}, {1, 1}, {1, 1}, {1, 1}, {0, 0}}
```

In[∘]:= `CGDecomposition[rep2, rep]`

8 ⊗ 3 = 15 ⊕ 6 ⊕ 3

Out[∘]=

{{2, 1}, {0, 2}, {1, 0}}

In[∘]:= `CGDecomposition[rep, rep2]`

3 ⊗ 8 = 15 ⊕ 6 ⊕ 3

Out[∘]=

{{2, 1}, {0, 2}, {1, 0}}

In[∘]:= `CGDecomposition[rep4, rep3, rep2]`

24 ⊗ 6 ⊗ 8 = 105 ⊕ 36 ⊕ 90 ⊕ 90 ⊕ 48 ⊕ 48 ⊕ 48 ⊕ 48 ⊕ 60 ⊕ 60 ⊕ 21 ⊕ 42 ⊕ 42 ⊕ 42 ⊕ 42 ⊕ 42 ⊕ 42 ⊕ 15 ⊕ 15 ⊕ 15 ⊕ 15 ⊕ 24 ⊕ 24 ⊕ 24 ⊕ 24 ⊕ 15 ⊕ 15 ⊕ 15 ⊕ 15 ⊕ 15 ⊕ 6 ⊕ 6 ⊕ 3

Out[∘]=

{{6, 2}, {7, 0}, {4, 3}, {4, 3}, {5, 1}, {5, 1}, {5, 1}, {5, 1}, {2, 4}, {2, 4}, {0, 5}, {3, 2}, {3, 2}, {3, 2}, {3, 2}, {3, 2}, {3, 2}, {4, 0}, {4, 0}, {4, 0}, {4, 0}, {1, 3}, {1, 3}, {1, 3}, {1, 3}, {2, 1}, {2, 1}, {2, 1}, {2, 1}, {2, 1}, {0, 2}, {0, 2}, {1, 0}}

In[∘]:= `CGCoefficients[{0, -1}, {1, 0}]`

••• Part: Part 1 of {} does not exist.

••• Part: Part 1 of {} does not exist.

BuildRep: Dynkin labels {0, -1} are not non-negative integers.

Out[∘]=

$Aborted

In[∘]:= `Clear[rep, antirep, rep2, rep3, rep4]`

# SU(4)

In[∘]:= `SetGroup[SU[4]]`

Set group is A(3)

Rank of the group = 3

In[∘]:= `CoxeterNumber`

Out[∘]=

4

In[∘]:= `CoxeterPlane`

Out[∘]=

{{0.5, -0.5, 0.5, -0.5}, {0., 0.707107, -0.707107, 0.}}

In[∘]:= `SimpleRoots`

Out[∘]=

{{1, -1, 0, 0}, {0, 1, -1, 0}, {0, 0, 1, -1}}

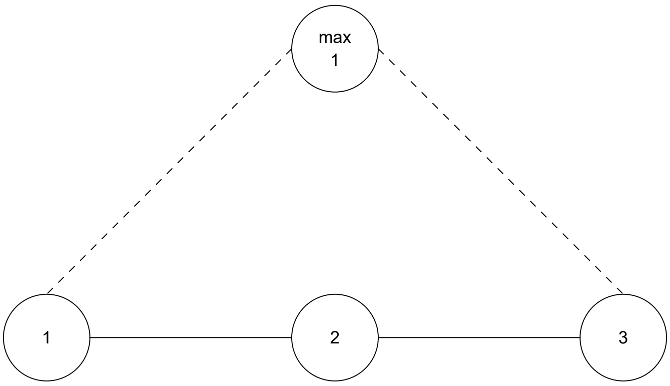
In[∘]:= `SimpleRoots // PhysicalNormalization`

Out[∘]=

$$\left\{\{1, 0, 0\}, \left\{-\frac{1}{2}, \frac{\sqrt{3}}{2}, 0\right\}, \left\{0, -\frac{1}{\sqrt{3}}, \sqrt{\frac{2}{3}}\right\}\right\}$$

In[ ]:= **DynkinDiagram**

Out[ ]=




In[ ]:= **CartanMatrix[SimpleRoots]**

Out[ ]=

{{2, -1, 0}, {-1, 2, -1}, {0, -1, 2}}

In[ ]:= **CartanMatrix[SimpleRoots] // MatrixForm**

Out[ ]//MatrixForm=

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

In[ ]:= **PositiveRoots**

Out[ ]=

{{1, -1, 0, 0}, {1, 0, -1, 0}, {0, 1, -1, 0}, {1, 0, 0, -1}, {0, 1, 0, -1}, {0, 0, 1, -1}}

In[ ]:= **PositiveRoots // PhysicalNormalization**

Out[ ]=

$$\left\{\{1, 0, 0\}, \left\{\frac{1}{2}, \frac{\sqrt{3}}{2}, 0\right\}, \left\{-\frac{1}{2}, \frac{\sqrt{3}}{2}, 0\right\}, \left\{\frac{1}{2}, \frac{1}{2\sqrt{3}}, \sqrt{\frac{2}{3}}\right\}, \left\{-\frac{1}{2}, \frac{1}{2\sqrt{3}}, \sqrt{\frac{2}{3}}\right\}, \left\{0, -\frac{1}{\sqrt{3}}, \sqrt{\frac{2}{3}}\right\}\right\}$$

In[ ]:= **CartanMatrix[PositiveRoots]**

Out[ ]=

{{2, 1, -1, 1, -1, 0}, {1, 2, 1, 1, 0, -1}, {-1, 1, 2, 0, 1, -1}, {1, 1, 0, 2, 1, 1}, {-1, 0, 1, 1, 2, 1}, {0, -1, -1, 1, 1, 2}}

In[ ]:= **CartanMatrix[PositiveRoots // PhysicalNormalization]**

Out[ ]=

{{2, 1, -1, 1, -1, 0}, {1, 2, 1, 1, 0, -1}, {-1, 1, 2, 0, 1, -1}, {1, 1, 0, 2, 1, 1}, {-1, 0, 1, 1, 2, 1}, {0, -1, -1, 1, 1, 2}}

In[ ]:= `CartanMatrix[PositiveRoots] // MatrixForm`

Out[ ]//MatrixForm=

$$\begin{pmatrix} 2 & 1 & -1 & 1 & -1 & 0 \\ 1 & 2 & 1 & 1 & 0 & -1 \\ -1 & 1 & 2 & 0 & 1 & -1 \\ 1 & 1 & 0 & 2 & 1 & 1 \\ -1 & 0 & 1 & 1 & 2 & 1 \\ 0 & -1 & -1 & 1 & 1 & 2 \end{pmatrix}$$

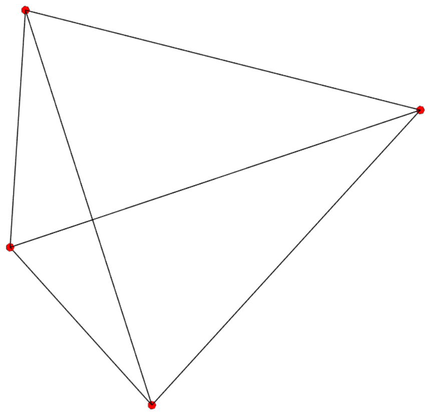
In[ ]:= `rep = BuildRep[{1, 0, 0}, OutputMethod → "Weights"]`

Out[ ]=

$$\left\{\left\{\left\{-\frac{1}{4}, -\frac{1}{4}, -\frac{1}{4}, \frac{3}{4}\right\}, 1\right\}, \left\{\left\{-\frac{1}{4}, -\frac{1}{4}, \frac{3}{4}, -\frac{1}{4}\right\}, 1\right\}, \left\{\left\{-\frac{1}{4}, \frac{3}{4}, -\frac{1}{4}, -\frac{1}{4}\right\}, 1\right\}, \left\{\left\{\frac{3}{4}, -\frac{1}{4}, -\frac{1}{4}, -\frac{1}{4}\right\}, 1\right\}\right\}$$

In[ ]:= `PlotRep3D[rep, SimpleRoots]`

Out[ ]=



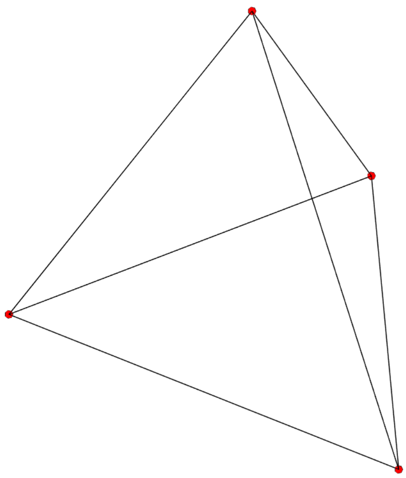
In[ ]:= `antirep = BuildRep[{0, 0, 1}, OutputMethod → "Weights"]`

Out[ ]=

$$\left\{\left\{\left\{-\frac{3}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right\}, 1\right\}, \left\{\left\{\frac{1}{4}, -\frac{3}{4}, \frac{1}{4}, \frac{1}{4}\right\}, 1\right\}, \left\{\left\{\frac{1}{4}, \frac{1}{4}, -\frac{3}{4}, \frac{1}{4}\right\}, 1\right\}, \left\{\left\{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, -\frac{3}{4}\right\}, 1\right\}\right\}$$

In[ ]:= **PlotRep3D[antirep, SimpleRoots]**

Out[ ]=



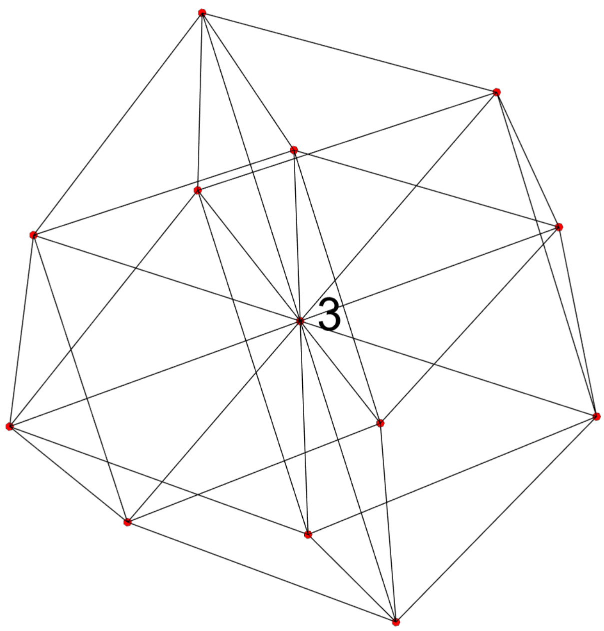
In[ ]:= **rep2 = BuildRep[{1, 0, 1}, OutputMethod → "Weights"]**

Out[ ]=

```
{{{-1, 0, 0, 1}, 1}, {{0, -1, 0, 1}, 1}, {{0, 0, -1, 1}, 1},
 {{0, 0, 0, 0}, 3}, {{-1, 0, 1, 0}, 1}, {{0, -1, 1, 0}, 1}, {{-1, 1, 0, 0}, 1},
 {{1, -1, 0, 0}, 1}, {{0, 1, -1, 0}, 1}, {{1, 0, -1, 0}, 1},
 {{0, 0, 1, -1}, 1}, {{0, 1, 0, -1}, 1}, {{1, 0, 0, -1}, 1}}
```

In[ ]:= **PlotRep3D[rep2, SimpleRoots]**

Out[ ]=



In[•]:=
```
rep = BuildRep[{1, 0, 0}]
```
Out[•]=
```
{{{0, 0, -1}, 1}, {{0, -1, 1}, 1}, {{-1, 1, 0}, 1}, {{1, 0, 0}, 1}}
```
In[•]:=
```
antirep = BuildRep[{0, 0, 1}]
```
Out[•]=
```
{{{-1, 0, 0}, 1}, {{1, -1, 0}, 1}, {{0, 1, -1}, 1}, {{0, 0, 1}, 1}}
```
In[•]:=
```
CGDecomposition[rep, rep]
```
4 ⊗ 4 = 10 ⊕ 6

Out[•]=
```
{{2, 0, 0}, {0, 1, 0}}
```
In[•]:=
```
CGDecomposition[antirep, antirep]
```
4 ⊗ 4 = 10 ⊕ 6

Out[•]=
```
{{0, 0, 2}, {0, 1, 0}}
```
In[•]:=
```
CGDecomposition[rep, antirep]
```
4 ⊗ 4 = 15 ⊕ 1

Out[•]=
```
{{1, 0, 1}, {0, 0, 0}}
```