Conjugates of the Standard Representation of S_3 in 3×3 matrices

Peter M. Maurer
Dept. of Computer Science
Baylor University
Waco, TX 76798-7356

These representations were found using the following procedure.

1. Generate the general linear group of $GF(2)^3$
2. Go through each matrix of the GLG and add the order 2 matrices to one list and the order 3 matrices to another.
3. For each order 2 matrix M
   a. add the order 3 matrices, one at a time to M
   b. close the group.
   c. If the size of the group is 6, add it and all its conjugates to the list of found groups
4. Print each generator pair.

This can be done in the GF2Matrices package using the following code.

This code enumerates all possible generators of a faithful representation, then it generates the group using the potential generators. If the group is of size 6 and not previously discovered, then all conjugates of the group are added to the list of discovered groups. (Note that every group is a conjugate of itself by the identity matrix.) When a new group is discovered, the generators are printed. This algorithm is extremely slow.

```
Group3 G3,G3x2,G3x3;
G3.GenerateGLG(  );
for (Matrix3 * Temp=G3.GetFirstMatrix() ; Temp ; Temp=G3.GetNextMatrix( ))
{
   if (Temp->Order( ) == 3)
   {
      G3x3.Add(*Temp);
   }
   else if (Temp->Order( ) == 2)
   {
      G3x2.Add(*Temp);
   }
}
GroupList3 GL3;
for (Matrix3 * Temp = G3x2.GetFirstMatrix( ) ; Temp ;
Temp=G3x2.GetNextMatrix( ))
{
   for (Matrix3 * Temp2 = G3x3.GetFirstMatrix( ) ; Temp2 ;
Temp2=G3x3.GetNextMatrix( ))
```

```
    {
       Group3 Tg3;
       Tg3.Add(*Temp);
       Tg3.Add(*Temp2);
       Tg3.Close( );
       Tg3.Sort( );
       if (Tg3.Order( ) == 6 && !GL3.Contains(Tg3))
       {
          cout<<Temp->SerializeNL();
          cout<<Temp2->SerializeNL();
          cout<<"------------------\n";
          for (Matrix3 * Temp3 = G3.GetFirstMatrix( ) ; Temp3 ;
                      Temp3 = G3.GetNextMatrix( ))
          {
             Group3 Tg3a;
             Tg3a = Tg3.Conjugate(*Temp3);
             Tg3a.Sort( );
             if (!GL3.Contains(Tg3a))
             {
                GL3.Add(Tg3a);
             }
          }
       }
    }
}
```

This algorithm will print a single pair of generators, the following:

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

To generate the data given below, use the following GF2Matrices code.

```
Group3 GLG3;
GLG3.GenerateGLG();
Group3 SR3;
SR3.GenerateS3();
GroupList3 SRConj;
for (Matrix3 *Temp=GLG3.GetFirstMatrix() ; Temp ; Temp=GLG3.GetNextMatrix())
{
      Group3 Tg;
      Tg = SR3.Conjugate(*Temp);
      Tg.Sort();
      if (!SRConj.Contains(Tg))
      {
            SRConj.Add(Tg);
      }
}
cout<<SRConj.SerializeNL();
cout<<endl<<endl<<"Groups="<<SRConj.Order()<<endl;
```

This code has been incorporated, without the printing, into the following GroupList3 function.

```
void GenerateS3Conj(void);
```

The side-notes show which representations are reducible and which are decomposable. A representation is decomposable if every matrix is of the following form.

$$\begin{pmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{or} \begin{pmatrix} 1 & 0 & 0 \\ 0 & a & b \\ 0 & c & d \end{pmatrix}$$

A representation is reducible if it is decomposable, or if every matrix in the representation has one of the following four forms. (The pattern must be uniform.)

$$\begin{pmatrix} a & b & x \\ c & d & y \\ 0 & 0 & 1 \end{pmatrix} \text{or} \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ x & y & 1 \end{pmatrix} \text{or} \begin{pmatrix} 1 & 0 & 0 \\ x & a & b \\ y & c & d \end{pmatrix} \text{or} \begin{pmatrix} 1 & x & y \\ 0 & a & b \\ 0 & c & d \end{pmatrix}$$

In either case, the matrices designated by the variables a, b, c, and d constitute a 2x2 representation of S_3, which may or may not be faithful.

In most cases, the representation will be:

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

This is a super-symmetric representation of S_3.

The reductions and decompositions were done by hand, so I may have missed some.

001	001	010	010	100	100
010	100	001	100	001	010
100	010	100	001	010	001

011	011	100	100	111	111
101	101	010	010	001	001
001	010	001	101	010	101

011	011	100	100	111	111
001	010	010	110	001	110
110	110	001	001	010	010

100	100	100	100	100	100	Reduction
001	001	010	010	111	111	
010	111	001	111	001	010	

001	001	011	011	100	100
111	111	101	101	010	010
011	100	001	100	001	011

010	010	100	100	110	110
100	110	010	110	010	100
001	011	001	101	011	101

100	100	100	100	100	100	Reduction
010	010	101	101	111	111	
001	011	011	110	001	110	

010	010	100	100	110	110
100	110	010	110	010	100
111	111	001	001	011	011

001	001	100	100	110	110
101	111	010	101	010	111
110	100	001	110	001	100

001	001	100	100	111	111
010	010	010	010	010	010
100	111	001	111	001	100

011	011	100	100	111	111
010	010	010	010	010	010
101	110	001	101	001	110

010	010	100	100	110	110	
100	110	010	110	010	100	
111	101	001	101	001	111	

010	010	011	011	100	100	
011	100	010	100	010	011	
111	111	110	110	001	001	

100	100	100	100	100	100	Reduction
010	010	011	011	101	101	
001	111	001	110	110	111	

001	001	100	100	101	101	
010	011	010	110	011	110	
100	101	001	101	001	100	

001	001	100	100	101	101	
111	111	010	010	011	011	
100	101	001	101	001	100	

010	010	100	100	101	101	
100	101	010	101	010	100	
111	110	001	110	001	111	

010	010	100	100	111	111	Reduction
100	111	010	111	010	100	
001	001	001	001	001	001	

011	011	100	100	111	111	Reduction
101	110	010	110	010	101	
001	001	001	001	001	001	

001	001	100	100	101	101	
110	111	010	110	010	111	
101	100	001	101	001	100	

011	011	100	100	101	101	
010	010	010	010	010	010	
110	111	001	111	001	110	

011	011	100	100	110	110	Reduction
101	111	010	111	010	101	
001	001	001	001	001	001	

010	010	100	100	110	110	Decomposition
100	110	010	110	010	100	
001	001	001	001	001	001	
001	001	100	100	101	101	
010	010	010	010	010	010	
100	101	001	101	001	100	
100	100	101	101	110	110	Reduction
001	010	001	011	010	011	
010	001	011	001	011	010	
100	100	101	101	111	111	Reduction
010	010	011	011	001	001	
001	011	001	010	010	011	
100	100	110	110	111	111	Reduction
010	011	001	010	001	011	
001	001	011	011	010	010	
100	100	100	100	100	100	Decomposition
001	001	010	010	011	011	
010	011	001	011	001	010	

Groups=28