Function: SP:-SyzygyIdeal - determine nontrivial relations among polynomials, if any

Calling Sequence:

SyzygyIdeal(F);

Parameters:

- F = [f1,f2,...,fm] is a list of polynomials in the ring *k*[x[1], x[2], x[3],..., x[n]].

Output:

- A list with polynomials in k[_y[1], _y[2], ..., _y[m]] which give a Groebner basis for the syzygy ideal I_F of relations.
- If the syzygy ideal I_F does not contain non-trivial identities, then the procedure returns [0].

Description:

- Procedure 'SyzygyIdeal' (or "ideal of relations") computes non-trivial relations, if any, between polynomials f1, f2,..., fm in a polynomial ring k[x[1],x[2],...,x[n]].
- The ideal of relations I_F is defined as a collection of polynomials h in the polynomial ring *k*[y[1], y[2],..., y[m]] such that h(f1, f2,..., fm) = 0. It is a prime ideal of *k*[y[1], y[2],..., y[m]].
- The ideal I_F can be computed directly using the elimination theory (see Section 7.4 in [1]]. According to Proposition 3 on page 339 in [1], the ideal I_F is the n-th elimination ideal of an ideal J_F = <f1 - y[1], f2 - y[2], ..., fm - y[m]> in the polynomial ring *k*[x[1], x[2], ..., x[n], y[1], y[2],..., y[m]]. That is, I_F is the intersection of J_F with *k*[y[1], y[2],..., y[m]].
- A Groebner basis for I_F can be computed then using the standard elimination theory as follows: Fix any monomial order T where any monomial involving one of x[1], x[2], ..., x[n] is greater than all monomials in *k*[y[1], y[2],... y[m]]. For example, we can set T to be the lexicographic order x[1] > x[2] > ...> x[n] > y[1] > y[2] >...> y[m]. Compute a Groebner basis G for J_F for the order T. Then, the intersection of G with the ring *k*[y[1], y[2],... y[m]] provides a Groebner basis for I_F. That is, the Groebner basis for I_F consists of those polynomials in G, if any, which belong to *k*[y[1], y[2],... y[m]].
- The procedure returns the Groebner basis for I_F or a list [0] if the intersection between G with the ring *k*[y[1], y[2],... y[m]] is empty. In that latter case, this means that the generators f1, f2, ..., fm are algebraically independent, or, in another words, that they do not satisfy any non-trivial relation.
- The Groebner basis for I_F may not be minimal: This is because the original list F of polynomials may contain polynomials which are algebraically dependent. Thus, in order to obtain a minimal Groebner basis for I_F, the smallest number of generating syzygy relations, apply the procedure

SP:-reduceGinvariants to the list F.

- Care needs to be excercised because variables _y[1], _y[2], ..., _y[m] returned by the procedure are defined as local to the procedure. This is so that Maple would not make automatic substitutions for these polynomials should they be defined in the worksheet. In order to be able to replace them with the polynomials f1, f2,..., fm, e.g., to verify that indeed these polynomials satisfy the syzygy relations, the local attribute of these variables need to be changed to global. See example below.
- In practice, the polynomials f1, f2,..., fm, will often be some elementary invariants of a finite group G, for example, like the elementary symmetric functions are invariant under the group S[n], and they will generate a ring of invariants k[x[1], x[2],..., x[n]]^G. Then, we will be interested to find all relations between the generators, if any, in order to reduce these generators to elementary ones.
- References:

[1] D. Cox, J. Little, D. O'Shea: "Ideals, varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra", Third Edition, Springer, New York, 2007
[2] B. Sturmfels: "Algorithms in Invariant Theory", Texts and Monographs in Symbolic Computation, Springer-Verlag/Wien, 1993

Examples:

```
> restart:with(SP);
Remember table of SymmetricGroup has been read and assigned
Remember table of AlternatingGroup has been read and assigned
Remember table of Reynolds has been read and assigned
Remember table of FiniteGroups has been read and assigned
Remember table of generateGinvariants has been read and assigned
```

[*AlternatingGroup*, *Dpolynom*, *FiniteGroups*, *Hilbert_series*, *MatrixAction*, *ModuleLoad*, *ModuleUnload*, *Molien_series*, *Reynolds*, *SPversion*, *Schur_polynom*, Σ, *SymmetricGroup*, *SyzygyIdeal*, *a_polynom*, *create_partitions*, *generateGinvariants*, *gpolynom*, *hpolynom*, *isContained*, *isGinvariant*, *isSymmetric*, *ispartition*, *load_remember_table*, *maxmindegree*, *permsign*, *powersum*, *reduceGinvariants*, *sigma_to_powersum*]

```
>
```

Example 1: Find a nontrivial syzygy relations among invariants of C4:

```
> C4:=FiniteGroups('C4');
```

$$C4 := \left[\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}, \begin{bmatrix}0 & -1\\1 & 0\end{bmatrix}, \begin{bmatrix}-1 & 0\\0 & -1\end{bmatrix}, \begin{bmatrix}0 & 1\\-1 & 0\end{bmatrix}\right]$$

```
> F:=generateGinvariants[C4]([x,y]);
```

$$F := [x^2 + y^2, y^4 + x^4, x\,y^3 - x^3\,y, x^2\,y^2, x^3\,y - x\,y^3]$$

```
> I_F:=SyzygyIdeal(F);
```

```
nops(I_F);
I_F:=map(convert,I_F,`global`);
```

```
Path set to C:\Maple11\bin.win/libfgbuni.so
    FGb/Maple interface package Version 1.34
    JC Faugere (jcf@calfor.lip6.fr)
    Type ?FGb for documentation
```

$$I\_F := [\_y_1{}^2 - \_y_2 - 2\,\_y_4, \_y_2\,\_y_4 - 2\,\_y_4{}^2 - \_y_5{}^2, \_y_3 + \_y_5]$$

$$3$$

$$I\_F := [\_y_1{}^2 - \_y_2 - 2\,\_y_4, \_y_2\,\_y_4 - 2\,\_y_4{}^2 - \_y_5{}^2, \_y_3 + \_y_5]$$

```
> map(simplify,subs({seq(_y[i]=F[i],i=1..nops(F))},I_F));
```

$$[0, 0, 0]$$

Since the generators in the list F are algebraically related, the ideal I_F is returned with three polynomial relations. However, these relations are not independent because if dependent relations are removed from the list F, the final list of relations reduces to just one polynomial:

```
> 'F'=F;
FF:=reduceGinvariants[C4](F,[x,y]);
```

$$F = [x^2 + y^2, y^4 + x^4, x\,y^3 - x^3\,y, x^2\,y^2, x^3\,y - x\,y^3]$$

$$FF := [x\,y^3 - x^3\,y, x^2\,y^2, x^2 + y^2]$$

```
> I_FF:=SyzygyIdeal(FF);
nops(I_FF);
I_FF:=map(convert,I_FF,`global`);
```

$$I\_FF := [\_y_2\,\_y_3{}^2 - \_y_1{}^2 - 4\,\_y_2{}^2]$$

$$1$$

$$I\_FF := [\_y_2\,\_y_3{}^2 - \_y_1{}^2 - 4\,\_y_2{}^2]$$

```
> map(simplify,subs({seq(_y[i]=FF[i],i=1..nops(FF))},I_FF));
```

$$[0]$$

```
>
```

Example 2: In the above example we found out that the ring *k*[x,y]^C4 = *k*[FF] where the generating C4 polynomials in FF satisfy one syzygy relation displayed in the list I_FF. This means, that although any polynomial f invariant under C4, that is, any polynomial f from *k*[x,y]^C4, can be experssed in terms of polynomials in the list FF, this representation is not unique: It is modulo the syzygy relation. Here is an example:

```
> f:=2*x*y-y^4+x^6;
```

$$f := 2\,x\,y - y^4 + x^6$$

```
> isGinvariant[C4](f,[x,y]);
```

false

Thus, the polynomial f defined above is not C4 invariant. We can easily generate out of f a new polynomial that will be C4-invariant by using the Reynolds operator (see SP:-Reynolds) for C4:

```
> f:=Reynolds[C4](f,[x,y]);
```

$$f := -\frac{1}{2}y^4 + \frac{1}{2}x^6 - \frac{1}{2}x^4 + \frac{1}{2}y^6$$

```
> isGinvariant[C4](f,[x,y]);
```

$$true$$

Thus, this redefined polynomial f is now C4 invariant. Hence, it is contained in the polynomial ideal <FF>. We can verify this with the procedure SP:-isContained.

```
> isContained(f,FF);
  g:=isContained(f,FF,'r');
  g:=convert(g,`global`);
```

$$true$$

$$g := \_f_2 - \frac{1}{2}\_f_3^{\,2} - \frac{3}{2}\_f_2\,\_f_3 + \frac{1}{2}\_f_3^{\,3}$$

$$g := \_f_2 - \frac{1}{2}\_f_3^{\,2} - \frac{3}{2}\_f_2\,\_f_3 + \frac{1}{2}\_f_3^{\,3}$$

Thus, observe, that upon substituting FF[1], FF[2], FF[3] for _f[1], _f[2], _f[3] in the polynomial g, we get back the polynomial f:

```
> f = simplify(subs({_f[1]=FF[1],_f[2]=FF[2],_f[3]=FF[3]},g));
```

$$-\frac{1}{2}y^4 + \frac{1}{2}x^6 - \frac{1}{2}x^4 + \frac{1}{2}y^6 = -\frac{1}{2}y^4 + \frac{1}{2}x^6 - \frac{1}{2}x^4 + \frac{1}{2}y^6$$

Of course, the syzygy relation in I_FF is one nontrivial relation satisfied by the three polynomials FF[1], FF[2], and FF[3]. Let's call it h:

```
> h:=op(I_FF);
```

$$h := \_y_2\,\_y_3^{\,2} - \_y_1^{\,2} - 4\,\_y_2^{\,2}$$

Then, obviously polynomial g + h also gives f:

```
> f =
  simplify(subs({_f[1]=FF[1],_f[2]=FF[2],_f[3]=FF[3],_y[1]=FF[1],
  _y[2]=FF[2],_y[3]=FF[3]},g+h));
```

$$-\frac{1}{2}y^4 + \frac{1}{2}x^6 - \frac{1}{2}x^4 + \frac{1}{2}y^6 = -\frac{1}{2}y^4 + \frac{1}{2}x^6 - \frac{1}{2}x^4 + \frac{1}{2}y^6$$

```
>
```

Example 3: Since the elementary symmetric polynomials sigma[1], sigma[2],..., are algebraically independent, their syzygy ideal does not contain any nontrivial relation:

```
> s1:=Sigma[1](x,y);
  s2:=Sigma[2](x,y);
  SyzygyIdeal([s1,s2]);
```

$$s1 := y + x$$

$$s2 := x\,y$$

$$[\,0\,]$$

```
> s1:=Sigma[1](x,y,z);
  s2:=Sigma[2](x,y,z);
  s3:=Sigma[3](x,y,z);
  SyzygyIdeal([s1,s2,s3]);
```

$$s1 := x+y+z$$

$$s2 := x\,y+x\,z+y\,z$$

$$s3 := x\,y\,z$$

$$[0]$$

```
> s1:=Sigma[1](x,y,z,t);
  s2:=Sigma[2](x,y,z,t);
  s3:=Sigma[3](x,y,z,t);
  s4:=Sigma[4](x,y,z,t);
  SyzygyIdeal([s1,s2,s3,s4]);
```

$$s1 := t+x+y+z$$

$$s2 := t\,x+t\,y+t\,z+x\,y+x\,z+y\,z$$

$$s3 := t\,x\,y+t\,x\,z+t\,y\,z+x\,y\,z$$

$$s4 := t\,x\,y\,z$$

$$[0]$$

Thus, this means that any symmetric polynomial f can be uniquely expressed in terms of the elementary symmetric functions because these functions do not satisfy any syzygy relation.

```
>
```

Example 4: This command SyzygyIdeal can be used to find relations among any set of polynomials, that is, not necessarily some G-invariants:

```
> f1:=x^2+y^2;
  f2:=x^3*y-x*y^3;
  f3:=x^2*y^2;
```

$$f1 := x^2+y^2$$

$$f2 := x^3\,y-x\,y^3$$

$$f3 := x^2\,y^2$$

```
> L:=SyzygyIdeal([f1,f2,f3]);
```

$$L := [\_y_1^{\,2}\,\_y_3-\_y_2^{\,2}-4\,\_y_3^{\,2}]$$

```
> L:=map(convert,L,`global`);
```

$$L := [\_y_1^{\,2}\,\_y_3-\_y_2^{\,2}-4\,\_y_3^{\,2}]$$

```
> subs({_y[1]=f1,_y[2]=f2,_y[3]=f3},L);
```

$$[(x^2+y^2)^2\,x^2\,y^2-(x^3\,y-x\,y^3)^2-4\,x^4\,y^4]$$

```
> map(expand,%);
```

$$[0]$$

Here is another example:

```
> f1:=x^2:
  f2:=y^2:
  f3:=x*y:
> L:=SyzygyIdeal([f1,f2,f3]);
```

$$L := [\_y_1\,\_y_2 - \_y_3^{\,2}]$$

```
> L:=map(convert,L,`global`);
```

$$L := [\_y_1\,\_y_2 - \_y_3^{\,2}]$$

```
> subs({_y[1]=f1,_y[2]=f2,_y[3]=f3},L);
```

$$[0]$$

```
>
>
```

Algorithm used:

- Proposition 3 on page 339 in [1]:
- According to Proposition 3 on page 339 in [1], the ideal I_F is the n-th elimination ideal of an ideal J_F = <f1 - y[1], f2 - y[2], ..., fm - y[m]> in the polynomial ring *k*[x[1], x[2], ..., x[n], y[1], y[2],..., y[m]]. That is, I_F is the intersection of J_F with *k*[y[1], y[2],..., y[m]].

  Step 1: Let F = [f1, f2,..., fm] and let T be any monomial order where any monomial involving one of x[1], x[2], ..., x[n] is greater than all monomials in *k*[y[1], y[2],... y[m]]. For example, we can set T to be the lexicographic order x[1] > x[2] > ...> x[n] > y[1] > y[2] >...> y[m].

  Step 2: Define an ideal J_F = <f1 - y[1], f2 - y[2], ..., fm - y[m]> in the polynomial ring *k*[x[1], x[2], ..., x[n], y[1], y[2],..., y[m]] and compute a Groebner basis G for J_F for the monomial order T.

  Step 3: Find the intersection of G with the ring *k*[y[1], y[2],... y[m]]: This interesection provides a Groebner basis for I_F. That is, the Groebner basis for I_F consists of those polynomials in G, if any, which belong to *k*[y[1], y[2],... y[m]].

See Also: SP:-FiniteGroups, SP:-Reynolds, SP:-isSymmetric, SP:-reduceGinvariants, SP:-isContained

Last modified: June 19, 2008

Function: SP:-AlternatingGroup - returns elements of the alternating group A[n]

Calling Sequence:

AlternatingGroup(n);

Parameters:

- n is a positive integer

Output:

- A list of n x n matrices that represent all elements of the alternating group A[n].

Description:

- Procedure 'AlternatingGroup' returns a list with n!/2 square n x n matrices that represent elements of A[n], a normal subgroup of S[n].
- This procedure has a remember table stored in the library.
- This procedure is used later when computing group invariants of SP:-FiniteGroups with the SP:-Reynolds operator.
- References:
  [1] D. Cox, J. Little, D. O'Shea: "Ideals, varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra", Third Edition, Springer, New York, 2007
  [2] B. Sturmfels: "Algorithms in Invariant Theory", Texts and Monographs in Symbolic Computation, Springer-Verlag/Wien, 1993

Examples:

```
> restart:with(SP):
Remember table of SymmetricGroup has been read and assigned
Remember table of AlternatingGroup has been read and assigned
Remember table of Reynolds has been read and assigned
Remember table of FiniteGroups has been read and assigned
Remember table of generateGinvariants has been read and assigned
```

Example 1: Elements of various alternating groups:

```
> AlternatingGroup(1);
```

$$[[\ 1]]$$

```
> AlternatingGroup(2);
```

$$\left[\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right]$$

```
> AlternatingGroup(3);
```

$$\left[\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}, \begin{bmatrix}0 & 0 & 1\\1 & 0 & 0\\0 & 1 & 0\end{bmatrix}, \begin{bmatrix}0 & 1 & 0\\0 & 0 & 1\\1 & 0 & 0\end{bmatrix}\right]$$

```
> AlternatingGroup(4);
```

$$\left[\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\end{bmatrix}, \begin{bmatrix}1 & 0 & 0 & 0\\0 & 0 & 0 & 1\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\end{bmatrix}, \begin{bmatrix}1 & 0 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\\0 & 1 & 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 1 & 0 & 0\\1 & 0 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}, \begin{bmatrix}0 & 0 & 1 & 0\\1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\end{bmatrix},\right.$$

$$\begin{bmatrix}0 & 0 & 0 & 1\\1 & 0 & 0 & 0\\0 & 0 & 1 & 0\\0 & 1 & 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\1 & 0 & 0 & 0\\0 & 0 & 0 & 1\end{bmatrix}, \begin{bmatrix}0 & 0 & 0 & 1\\0 & 1 & 0 & 0\\1 & 0 & 0 & 0\\0 & 0 & 1 & 0\end{bmatrix}, \begin{bmatrix}0 & 0 & 1 & 0\\0 & 0 & 0 & 1\\1 & 0 & 0 & 0\\0 & 1 & 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\\1 & 0 & 0 & 0\end{bmatrix}, \begin{bmatrix}0 & 0 & 1 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\1 & 0 & 0 & 0\end{bmatrix},$$

$$\left.\begin{bmatrix}0 & 0 & 0 & 1\\0 & 0 & 1 & 0\\0 & 1 & 0 & 0\\1 & 0 & 0 & 0\end{bmatrix}\right]$$

```
>
```

Example 2: Let's see an action of the permutation group A[3] on a polynomial:

```
> A3:=AlternatingGroup(3):
> f:=x^3-y^2+x*y+z;
```

$$f := x^3 - y^2 + x\,y + z$$

```
> for A in A3 do MatrixAction(A,f,[x,y,z]) end do;
```

$$x^3 - y^2 + x\,y + z$$

$$z^3 - x^2 + x\,z + y$$

$$y^3 - z^2 + y\,z + x$$

```
>
```

Algorithm used:

None

See Also: SP:-Sigma, SP:-FiniteGroups, SP:-SymmetricGroup, SP:-MatrixAction, SP:-Reynolds, Groebner

Last modified: June 19, 2008

Function: SP[] - <description>

Calling Sequence:

lst := dummy(N,M)

Parameters:

- N,M : <input type>

Output:

- lst : <output type>

WARNING:

iff applicable

Description:

- Bullet item list of properties of the function

Examples:

```
> restart:with(SP):
Remember table of SymmetricGroup has been read and assigned
Remember table of AlternatingGroup has been read and assigned
Remember table of Reynolds has been read and assigned
Remember table of FiniteGroups has been read and assigned
Remember table of generateGinvariants has been read and assigned
>
>
```

mandatory test cases and most likly cases a user want to type in

Algorithm used:

The presently implemented algorithm, if possible with certificate.

See Also: SP[dummy]

Last modified: June 19, 2008

Function: SP:-FiniteGroups - returns matrix representations of various finite groups

Calling Sequence:

FiniteGroups(name);
FiniteGroups();

Parameters:

- name is a string of the form 'G' where G is one of the names of known groups

Output:

- A list of matrices that represent all elements of the group G

Description:

- Procedure 'FiniteGroups' when used without any argument, returns a names of known groups.
- Procedure 'FiniteGroups' when used with one argument, returns a list of matrices that represent elements of one of the following groups:

  (1) C2 - cyclic group of 2 x 2 matrices of order 2, hence |C2| = 2 (see Example 13 in Section 7.2 in [1]).
  (2) C4 - a cyclic group of 2 x 2 matrices of order 4, hence |C4| = 4 (see Example 3 and Exercise 16 in Section 7.2 in [1]).
  (3) C8 - a cyclic group of 3 x 3 matrices of order 8, hence |C8| = 8 (see Example 3 in Section 7.2 and Example 6 in Section 7.3 in [1]).
  (4) D6 - a dihedral group of 3 x 3 matrices of order 12, hence |D6| =12. D6 is a the symmetry group of an equilateral triangle (see Example 2.2.6 in [2]).
  (5) D8 - a dihedral group of 2 x 2 matrices of order 16, hence |D8| = 16. D8 is the symmetry group of a regular hexagon in the plane (see Section 2.2 in [2]).
  (6) V4 - the Klein four-group of 2 x 2 matrices of order 4, hence |V4| = 4 (see Example 12 and Exercise 15 in Section 7.2 in [1]).
  (7) C3 - a cyclic group of 2 x 2 matrices of order 3, hence |C3| =3 (see Exercise 7 in Section 7.3 in [1]).
  (8) C6 - a cyclic group of 2 x 2 matrices of order 6, hence |C6|=6 (see Exercise 7 in Section 7.3 in [1]) .
  (9) G8 - a group of diagonal 3 x 3 matrice of order 8, hence |G8|=8 (see Exercise 10 in Section 7.3 in [1]).

- This procedure has a remember table stored in the library. This means that new groups could be assigned and remembered to this procedure.

- This procedure is used later when computing group invariants of SP:-FiniteGroups with the SP:-Reynolds operator.

- References:

[1] D. Cox, J. Little, D. O'Shea: "Ideals, varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra", Third Edition, Springer, New York, 2007
[2] B. Sturmfels: "Algorithms in Invariant Theory", Texts and Monographs in Symbolic Computation, Springer-Verlag/Wien, 1993

Examples:

```
> restart:with(SP):
Remember table of SymmetricGroup has been read and assigned
Remember table of AlternatingGroup has been read and assigned
Remember table of Reynolds has been read and assigned
Remember table of FiniteGroups has been read and assigned
Remember table of generateGinvariants has been read and assigned
>
```

Example 1: Display names of groups for which matrix representations are stored in the remember table of 'FiniteGroups':

```
> op(sort(map(op,[indices(op(4,eval(SP:-FiniteGroups)))])));
```

$$C2, C3, C4, C6, C8, \mathrm{D6}, \mathrm{D8}, G8, V4$$

When a user defines a new group and then assigns this new group entry to 'FiniteGroups', the new group is remembered and upon unloading this package, it is stored in the library. Upon loading again this package, 'FiniteGroups' will know this newly defined group. For example, we can define a new group Gamma3. This is a three dimensional representation of the cyclic group of order 4.

```
> A1:=linalg:-diag(1$3);
  A2:=matrix(3,3,[0,1,0,-1,0,0,0,0,-1]);
  A3:=matrix(3,3,[-1,0,0,0,-1,0,0,0,1]);
  A4:=matrix(3,3,[0,-1,0,1,0,0,0,0,-1]);
  Gamma3:=map(evalm,[A1,A2,A3,A4]);
```

$$A1 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A2 := \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$A3 := \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A4 := \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\Gamma 3 := \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \right]$$

```
> FiniteGroups('Gamma3');
Group Gamma3 is not known to FiniteGroups. You can make it known by assigning i
t to FiniteGroups('Gamma3').
> FiniteGroups('Gamma3'):=Gamma3;
```

$$SP{:-}FiniteGroups(\Gamma 3) := \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \right]$$

```
> FiniteGroups('Gamma3');
```

$$\left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \right]$$

Now, we restart Maple and reload the SP package: Notice that the Gamma3 group is now available and need not be defined once more.

```
> restart:
> with(SP);
Remember table of SymmetricGroup has been read and assigned
Remember table of AlternatingGroup has been read and assigned
Remember table of Reynolds has been read and assigned
Remember table of FiniteGroups has been read and assigned
Remember table of generateGinvariants has been read and assigned
```

[*AlternatingGroup*, *Dpolynom*, *FiniteGroups*, *Hilbert_series*, *MatrixAction*, *ModuleLoad*, *ModuleUnload*, *Molien_series*, *Reynolds*, *SPversion*, *Schur_polynom*, Σ, *SymmetricGroup*, *SyzygyIdeal*, *a_polynom*, *create_partitions*, *generateGinvariants*, *gpolynom*, *hpolynom*, *isContained*, *isGinvariant*, *isSymmetric*, *ispartition*, *load_remember_table*, *maxmindegree*, *permsign*, *powersum*, *reduceGinvariants*, *sigma_to_powersum*]

```
> op(sort(map(op,[indices(op(4,eval(SP:-FiniteGroups)))])));
```

C2, *C3*, *C4*, *C6*, *C8*, D6, D8, *G8*, Γ3, *V4*

```
> FiniteGroups('Gamma3');
```

$$\left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \right]$$

```
> FiniteGroups('C2');
```

$$\left[\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}, \begin{bmatrix}-1 & 0\\0 & -1\end{bmatrix}\right]$$

```
> FiniteGroups('C4');
```

$$\left[\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}, \begin{bmatrix}0 & -1\\1 & 0\end{bmatrix}, \begin{bmatrix}-1 & 0\\0 & -1\end{bmatrix}, \begin{bmatrix}0 & 1\\-1 & 0\end{bmatrix}\right]$$

```
> FiniteGroups('C8');
```

$$\left[\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}, \begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & -1\end{bmatrix}, \begin{bmatrix}1 & 0 & 0\\0 & -1 & 0\\0 & 0 & 1\end{bmatrix}, \begin{bmatrix}1 & 0 & 0\\0 & -1 & 0\\0 & 0 & -1\end{bmatrix}, \begin{bmatrix}-1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}, \begin{bmatrix}-1 & 0 & 0\\0 & 1 & 0\\0 & 0 & -1\end{bmatrix}, \begin{bmatrix}-1 & 0 & 0\\0 & -1 & 0\\0 & 0 & 1\end{bmatrix}, \begin{bmatrix}-1 & 0 & 0\\0 & -1 & 0\\0 & 0 & -1\end{bmatrix}\right]$$

```
> FiniteGroups('D6');
```

$$\left[\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}, \begin{bmatrix}\frac{1}{2} & -\frac{\sqrt{3}}{2} & 0\\\frac{\sqrt{3}}{2} & \frac{1}{2} & 0\\0 & 0 & 1\end{bmatrix}, \begin{bmatrix}\frac{-1}{2} & -\frac{\sqrt{3}}{2} & 0\\\frac{\sqrt{3}}{2} & \frac{-1}{2} & 0\\0 & 0 & 1\end{bmatrix}, \begin{bmatrix}-1 & 0 & 0\\0 & -1 & 0\\0 & 0 & 1\end{bmatrix}, \begin{bmatrix}\frac{-1}{2} & \frac{\sqrt{3}}{2} & 0\\-\frac{\sqrt{3}}{2} & \frac{-1}{2} & 0\\0 & 0 & 1\end{bmatrix},\right.$$
$$\begin{bmatrix}\frac{1}{2} & \frac{\sqrt{3}}{2} & 0\\-\frac{\sqrt{3}}{2} & \frac{1}{2} & 0\\0 & 0 & 1\end{bmatrix}, \begin{bmatrix}1 & 0 & 0\\0 & -1 & 0\\0 & 0 & -1\end{bmatrix}, \begin{bmatrix}\frac{1}{2} & -\frac{\sqrt{3}}{2} & 0\\-\frac{\sqrt{3}}{2} & \frac{-1}{2} & 0\\0 & 0 & -1\end{bmatrix}, \begin{bmatrix}\frac{-1}{2} & -\frac{\sqrt{3}}{2} & 0\\-\frac{\sqrt{3}}{2} & \frac{1}{2} & 0\\0 & 0 & -1\end{bmatrix}, \begin{bmatrix}-1 & 0 & 0\\0 & 1 & 0\\0 & 0 & -1\end{bmatrix},$$
$$\left.\begin{bmatrix}\frac{-1}{2} & \frac{\sqrt{3}}{2} & 0\\\frac{\sqrt{3}}{2} & \frac{1}{2} & 0\\0 & 0 & -1\end{bmatrix}, \begin{bmatrix}\frac{1}{2} & \frac{\sqrt{3}}{2} & 0\\\frac{\sqrt{3}}{2} & \frac{-1}{2} & 0\\0 & 0 & -1\end{bmatrix}\right]$$

```
> FiniteGroups('D8');
```

$$\left[\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}, \begin{bmatrix}-1 & 0\\0 & 1\end{bmatrix}, \begin{bmatrix}0 & -1\\-1 & 0\end{bmatrix}, \begin{bmatrix}0 & 1\\-1 & 0\end{bmatrix}, \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}, \begin{bmatrix}-1 & 0\\0 & -1\end{bmatrix}, \begin{bmatrix}0 & -1\\1 & 0\end{bmatrix}, \begin{bmatrix}\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2}\\\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2}\end{bmatrix},\right.$$
$$\begin{bmatrix}\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2}\\\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2}\end{bmatrix}, \begin{bmatrix}-\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2}\\\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2}\end{bmatrix}, \begin{bmatrix}-\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2}\\-\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2}\end{bmatrix}, \begin{bmatrix}\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2}\\-\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2}\end{bmatrix}, \begin{bmatrix}\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2}\\-\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2}\end{bmatrix},$$

$$\left[\begin{bmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}, \begin{bmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}\right]$$

```
> nops(%);
```

$$16$$

```
> FiniteGroups('V4');
```

$$\left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}\right]$$

```
> FiniteGroups('C3');
```

$$\left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix}\right]$$

```
> FiniteGroups('C6');
```

$$\left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}\right]$$

```
> FiniteGroups('G8');
```

$$\left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}\right]$$

```
>
```

Example 2: Let's see an action of the group G8 on a polynomial:

```
> G:=FiniteGroups('G8');
```

$$G := \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}\right]$$

```
> f:=x^3-y^2+x*y+z+z^5;
```

$$f := x^3 - y^2 + x\,y + z + z^5$$

```
> for A in G do MatrixAction(A,f,[x,y,z]) end do;
```

$$x^3 - y^2 + x\,y + z + z^5$$
$$x^3 - y^2 + x\,y - z - z^5$$
$$x^3 - y^2 - x\,y + z + z^5$$
$$x^3 - y^2 - x\,y - z - z^5$$

$$-x^3 - y^2 - x\,y + z + z^5$$
$$-x^3 - y^2 - x\,y - z - z^5$$
$$-x^3 - y^2 + x\,y + z + z^5$$
$$-x^3 - y^2 + x\,y - z - z^5$$

>

Algorithm used:

None

See Also: SP:-Sigma, SP:-AlternatingGroup, SP:-SymmetricGroup, SP:-MatrixAction, SP:-Reynolds, Groebner

Last modified: June 19, 2008