Penvelope

Version 1.0

Contents

Chapter 1

Penvelope

1.1 The functions

1.1.1 PEnvelopingAlgebra

◊ `PEnvelopingAlgebra( L )` (operation)

Here `L` is a restricted Lie algebra. This function returns the `p`-enveloping algebra of `L`.

It is possible to add the option `cut`, as in `PEnvelopingAlgebra( L : cut );`. In this case the quotient of the `p`-enveloping algebra by the $c+1$-st power of the Jacobson radical will be returned. In this case the Lie algebra has to be graded.

Example

```
gap> L:= JenningsLieAlgebra(H1);
<Lie algebra of dimension 6 over GF(3)>
gap> U:= PEnvelopingAlgebra( L : cut );
<algebra of dimension 75 over GF(3)>
```

1.1.2 QuotientByMaximalIdeal

◊ `QuotientByMaximalIdeal( U )` (operation)

Here `U` must be a `p`-enveloping algebra. This function returns a list of two elements. The first is an associative algebra isomorphic to a quotient of `U` (the ideal is constructed by taking consecutive complements to the Lie algebra inside subspaces consisting of `u` such that `Uu=uU=0`). The second element of the list is a set of basis elements that form a basis of the Lie algebra inside the algebra.

Example

```
gap> U:= PEnvelopingAlgebra( L : cut );
<algebra of dimension 26 over GF(3)>
gap> QuotientByMaximalIdeal( U );
[ <algebra of dimension 9 over GF(3)>, [ v.2, v.3, v.4, v.5, v.6, v.7 ] ]
```

1.1.3 UElement

◊ `UElement( a )` (operation)

Here `a` is a nilpotent element of a ring, for example a `p`-enveloping algebra. This function returns the invertible element `1+a`.

Example
```
gap> L:= JenningsLieAlgebra( H2 );
<Lie algebra of dimension 6 over GF(3)>
gap> U:= PEnvelopingAlgebra( L : cut );
<algebra of dimension 23 over GF(3)>
gap> r:= QuotientByMaximalIdeal( U );
[ <algebra of dimension 10 over GF(3)>, [ v.2, v.3, v.4, v.5, v.6, v.7 ] ]
gap> UElement( r[2][1] );
1 + v.2
```

1.1.4 PCGroupByUElements

◊ `PCGroupByUElements( list )` (operation)

Here `list` is a list of elements made by `UElement`. This function returns a pc group isomorphic to the group generated by the elements in `list`.

Example
```
gap> h:= List( r[2], UElement );
[ 1 + v.2, 1 + v.3, 1 + v.4, 1 + v.5, 1 + v.6, 1 + v.7 ]
gap> PCGroupByUElements( h{[1,2]} );
#I  default `IsGeneratorsOfMagmaWithInverses' method returns `true' for
[ 1 + v.2, 1 + v.3 ]
The size of G is 9
Group([ f1, f2 ])
```

Index