CKForms - Manual

Notation and convention

In this manual we use small latin leters `g`,`h`,... or small gothic letters $\mathfrak{g}$, $\mathfrak{h}$,... for Lie algebras.

We use the notation of the theory of real Lie algebra from CoReLG Package, (Dietrich, Faccin, and Graaf 2014). Every real simple Lie algebra is identified with triple `(type,rank, id)`. Be careful, `rank` is not always considered as the dimension of the Cartan subalgebra of the simple Lie algebra $\mathfrak{g}$. It is equal to the dimension of the Cartan subalgebra for all real simple Lie algebras excluding realifications (that is, complex simple lie algebras considered as real Lie algebras). For realifications `id` is equal to zero and `rank` is equal to half of the rank of the Lie algebra $\mathfrak{g}$. Consider the following example.

```
gap> RealFormsInformation("E",6);

  There are 5 simple real forms with complexification E6
    1 is the compact form
    2 is EI   = E6(6), with k_0 of type sp(4) (C4)
    3 is EII  = E6(2), with k_0 of type su(6)+su(2) (A5+A1)
    4 is EIII = E6(-14), with k_0 of type so(10)+R (D5+R)
    5 is EIV  = E6(-26), with k_0 of type f_4 (F4)
  Index '0' returns the realification of E6

gap> g:=RealFormById("E",6,2);
<Lie algebra of dimension 78 over SqrtField>
gap> Dimension(CartanSubalgebra(g));
6
gap> h:=RealFormById("E",6,0);
<Lie algebra of dimension 156 over SqrtField>
gap> Dimension(CartanSubalgebra(h));
12
```

Notice: we found some minor misspellings in the code:

- `"D",4,5` is $\mathfrak{so}(1,7)$,
- `"D",4,4` is $\mathfrak{so}(3,5)$,
- `"E",7,3` is $\mathfrak{e}_{7(-25)} = EVII$,
- `"E",7,4` is $\mathfrak{e}_{7(-5)} = EVI$.

Compare the real rank and the dimension of the real form with the data given in Table 4 in (Onishchik and Vinberg 1990).

To make our work easier, we wrote our own function to recognize the triples.

`GetSymbolSimple(type, rank, id)`

This function works up to `rank` 8 and returns string corresponding to the symbol of real simple Lie algebra. Remark: we use notation from (Helgason 2001).

Example:

```
gap> GetSymbolSimple("C",4,4);
"sp(4,R)"
gap> GetSymbolSimple("A",5,4);
"su(3,3)"
```

```
gap> GetSymbolSimple("A",5,5);
"su*(6)"
```

Semisimple real Lie algebra

We propose the following convention to represent a semisimple real Lie algebra of rank up to 8. Due to the limitations of GAP (see this), we represent a semisimple real Lie algebra $\mathfrak{g}$ as tuple:

`[real rank, non-compact dimension,type1,rank1,id1,...,typen,rankn,idn]`

where `type1, rank1,id1` ... `typen,rankn,idn` represent the corresponding triples of simple components of $\mathfrak{g}$ (we use the fact that a semisimple algebra is a direct sum of simple algebras). The first two coordinates in the input vector are given to reduce the complexity of the algorithm.

`GetSymbolSemisimple(tuple)`

The input is a tuple corresponding to the semisimple real Lie algebra $\mathfrak{g}$ (represented in our notation). This function returns the symbol of $\mathfrak{g}$.

Example:

```
gap> T:=[ 4, 10, "A", 1, 2, "A", 1, 2, "B", 2, 2 ];
[ 4, 10, "A", 1, 2, "A", 1, 2, "B", 2, 2 ]
gap> GetSymbolSemisimple(T);
"sl(2,R)+sl(2,R)+so(2,3)"
```

If you are familiar with CoReLG Package (Dietrich, Faccin, and Graaf 2014) and want to work with Lie algebra object, we have created the following function:

`RealFormByTuple(tuple)`

The input is a tuple corresponding to a semisimple real Lie algebra $\mathfrak{g}$ (represented in our notation). This function returns a Lie algebra object corresponding to $\mathfrak{g}$.

Example:

```
gap> T:=[ 4, 10, "A", 1, 2, "A", 1, 2, "B", 2, 2 ];
[ 4, 10, "A", 1, 2, "A", 1, 2, "B", 2, 2 ]
gap> GetSymbolSemisimple(T);
"sl(2,R)+sl(2,R)+so(2,3)"
gap> g:=RealFormByTuple(T);
<Lie algebra of dimension 16 over SqrtField>
gap> RealRank(g);
4
gap> NonCompactDimension(g);
10
```

The reverse operation is ineffective in GAP. The conversion takes a very long time. Often the program closes without warning. The only option is to manually search for the appropriate isomorphisms.

Example:

```
gap> g:=RealFormById("E",6,3);
<Lie algebra of dimension 78 over SqrtField>
gap> GetSymbolSimple("E",6,3);
"E6(2)"
gap> k:=CartanDecomposition(G).K;
<Lie algebra of dimension 38 over SqrtField>
gap> T:=[0,0,"A",1,1,"A",5,1];
[ 0, 0, "A", 1, 1, "A", 5, 1 ]
```

```
gap> GetSymbolSemisimple(T);
"su(2)+su(6)"
gap> l:=RealFormByTuple(T);
<Lie algebra of dimension 38 over SqrtField>
gap> IsomorphismOfRealSemisimpleLieAlgebras(k,l);
<Lie algebra isomorphism between Lie algebras of dimension 38 over SqrtField>
```

`CheckTuple(tuple)`

The inupt is a tuple. This function returns "true" when the tuple corresponds to a real semisimple Lie algebra (it works only up to 8 simple components).

```
gap> T:=[0,0,"A",1,1,"A",5,1];
[ 0, 0, "A", 1, 1, "A", 5, 1 ]
gap> CheckTuple(T);
true
gap> T:=[0,2,"A",1,1,"A",5,1];
[ 0, 2, "A", 1, 1, "A", 5, 1 ]
gap> CheckTuple(T);
false
```

Functions for complex Lie algebra

`NicerSemisimpleSubalgebras(typeRank)`

The function is based on `LieAlgebraAndSubalgebras` function from SLA Package (Graaf 2018). It returns a list of types of semisimple subalgebras of a complex Lie algebra (for a given type and rank). Remark: some types may correspond to several linearly equivalence subalgebras.

`NumberOfSubalgebraClasses(typeRankG, typeRankH)`

This function returns the number of linear equivalence classes for a given type of subalgebra (which is the value of the second argument) in complex Lie algebra (which is the value of the first argument).

Example:

```
gap> NicerSemisimpleSubalgebras("E6");
[ "A1", "A2", "B2", "G2", "A1 A1", "A1 A1 A1", "A1 A2", "A1 B2", "A3", "A1 G2", "B3",
"C3", "A1 A1 A1 A1", "A1 A1 A2", "A1 A1 B2", "A2 A2", "A1 A3", "B2 B2", "A2 G2", "A1 C3",
"A1 B3", "A4", "D4", "C4", "B4", "F4", "A1 A2 A2", "A1 A1 A3",  "A1 A4", "A5", "D5",
"A2 A2 A2", "A1 A5" ]
gap> NumberOfSubalgebraClasses("E6","A1 A1");
24
gap> NumberOfSubalgebraClasses("E6","A5");
1
```

Function for real Lie algebras

`RealRank(g)`

The input is a real Lie algebra $\mathfrak{g}$ (as an object, not as a tuple). The output is the real rank of $\mathfrak{g}$ (the dimension of the Cartan subalgebra of $\mathfrak{g}$).

`NonCompactDimension(g)`

The input is a real Lie algebra $\mathfrak{g}$. This function returns non-compact dimension of $\mathfrak{g}$ (dimension of non-compact part from the Cartan decomposition for $\mathfrak{g}$).

`CompactDimension(g)`

The input is a real Lie algebra $\mathfrak{g}$. This function returns the compact dimension of $\mathfrak{g}$ (dimension of the compact part of the Cartan decomposition for $\mathfrak{g}$).

Example:

```
gap> g:=RealFormById("A",5,3);
<Lie algebra of dimension 35 over SqrtField>
gap> RealRank(g);
2
gap> NonCompactDimension(g);
16
gap> CompactDimension(g);
19
gap> T:=[ 5, 19, "A", 1, 2, "A", 1, 2, "A", 1, 1, "D", 4, 4 ];
[ 5, 19, "A", 1, 2, "A", 1, 2, "A", 1, 1, "D", 4, 4 ]
gap> g:=RealFormByTuple(T);
<Lie algebra of dimension 37 over SqrtField>
gap> RealRank(g);
5
gap> NonCompactDimension(g);
19
gap> CompactDimension(g);
18
```

Function for generating potential subalgebras and triples

The following features are described in more detail in our article (Bocheński, Jastrzębski, and Tralle 2019).

`PotentialSubalgebras("type",rank,id)`

The input is a triple corresponding to a simple real Lie algebra $\mathfrak{g}$ of rank up to 8. This function returns a list of semisimple subalgebras in $\mathfrak{g}$ as described in Algorithm 1 (Bocheński, Jastrzębski, and Tralle 2019).

`PotentialTriples("type",rank,id)`

The input is a triple corresponding to a simple real Lie algebra $\mathfrak{g}$ of rank up to 8. This function returns triples in $\mathfrak{g}$ as described in Algorithm 2 (Bocheński, Jastrzębski, and Tralle 2019).

`GetSymbolTriples(output of previous function)`

This function returns symbols of the triples (see example below).

```
gap> L:=PotentialTriples("E",6,2);
Counting: 30%
Counting: 60%
Counting: 90%
Counting completed.
[ [ [ 2, 14, "B", 4, 2 ], [ 4, 28, "F", 4, 2 ] ], [ [ 2, 14, "A", 5, 5 ],
[ 4, 28, "F", 4, 2 ] ], [ [ 3, 21, "D", 5, 6 ], [ 3, 21, "D", 5, 6 ] ],
[ [ 2, 14, "A", 1, 1, "A", 5, 5 ], [ 4, 28, "F", 4, 2 ] ] ]
gap> GetSymbolTriples(L);
h=so(2,7), l=F4(4)
h=su*(6), l=F4(4)
h=so(3,7), l=so(3,7)
h=su(2)+su*(6), l=F4(4)
gap> GetSymbolSimple("E",6,2);
"E6(6)"
```

References

Bocheński, M., P. Jastrzębski, and A. Tralle. 2019. "Non-Existence of Standard Compact Clifford-Klein Forms of Homogeneous Spaces of Exceptional Lie Groups." to do.

Dietrich, H., P. Faccin, and W. A. de Graaf. 2014. "CoReLG, Computation with Real Lie Groups, Version 1.20." http://users.monash.edu/~heikod/corelg/.

Graaf, W. A. de. 2018. "SLA, Simple Lie Algebras, Version 1.5." http://www.science.unitn.it/~degraaf/sla.html.

Helgason, S. 2001. *Differential Geometry and Symmetric Spaces.* American Mathematical Society.

Onishchik, A., and E. Vinberg. 1990. *Lie Groups and Algebraic Groups.* Springer Series in Soviet Mathematics. Springer-Verlag Berlin Heidelberg.