# Version 1.0 of the Atlas Software

Jeffrey Adams

March 16, 2007

## 1 Introduction

This is a description of what the Atlas software version 1.0 should be. We begin with a general description of the software, followed by a list of commands. See *Algorithms for Representation Theory of Real Reductive Groups* [**?**] for an extensive discussion of the mathematical background. *Parameters for Representations of Real Groups* [1] is a somewhat more casual introduction.

Thanks to David, Fokko and Marc for advice.

## 2 General description of the Software

Version 1.0 will treat structure theory of real reductive groups and their admissible representations, with arbitary infinitesimal character. It will not include much about unitary representations.

Here is a brief outline.

**Structure Theory:**

(1) Root systems

(2) Complex reductive groups

(3) Inner classes of complex groups

(4) Real forms of complex groups

(5) Maximal compact subgroup $K$ of a real group

(6) Cartan subgroups and Weyl groups

**Representation Theory:**

(1) Parametrization of irreducible and standard representations at

   (a) regular integral infinitesimal character

   (b) singular infinitesimal character: maybe in version 1.0

   (c) arbitary regular infinitesimal character: probably not in 1.0

(2) $K\backslash G/B$ picture

(3) Cross action, Cayley transforms and Kazhdan-Lusztig algorithm

(4) Parametrization of K-types

(5) Lowest K-types of a representation

(6) All K-types of a representation

(7) Unitary representations with regular integral infinitesimal character

# 3 Structure

There are four fundamental types of object, each corresponding to a mathematical construct:

(1) RootDatum: two lattices and two finite subsets $(X^*, \Delta, X_*, \Delta^\vee)$

(2) ComplexGroup: connected complex reductive group

(3) GaloisExtendedGroup: a group $G \rtimes \Gamma$ as in [**?**, Section 6]. This is the same as specifying $\gamma \in \mathrm{Out}(G)$ of order 2, and determines an inner class of real forms.

(4) RealForm: a particular real form

The next three subsections give structure found at each of these levels.

## 3.1 Root System

(1) Specifying a Lie type:

   (a) L:=LieType(A1)

   (b) L:=LieType(E8.A1)

   (c) L:=LieType(A1.E8.A1.B2)

   (d) L:=LieType(A1.T1)

   (e) L:=LieType(A1.T1.T1)

   (f) L:=LieType(A1.T2)

   (g) L:=LieType(T1.A1.T1)

(2) CartanMatrix(L)

(3) R:=RootDatum(L): the default is simply connected in the strong sense: the corresponding complex group is the product of a torus and a semisimple, simply connected group (the derived group of the dual group is adjoint).

(4) R:=RootDatum(L,M): root datum of the LieType L with matrix M. Each row of M is an element of $(P_0^\vee \times A)/R^\vee$, where $R^\vee$ is the coroot lattice, $P_0^\vee$ is the corresponding co-weight lattice (really a lattice), and $A = \mathbb{Q}^{\times n}$ where $n$ is the number of torus factors. This defines a general root system starting from a simply connected one. See 3.2(17) for more details.

(5) R:=RootDatum(L,s): simply connected

(6) R:=RootDatum(L,a): adjoint (error if any T factors)

(7) DualRootDatum(R)

(8) R:=RootDatum(A1.A2): skip the LieType step

(9) R:=RootDatum(A1.A2,M)

(10) L:=LieType(R): recover LieType from RootDatum

(11) CartanMatrix(R)=CartanMatrix(LieType(R))

## 3.2 Complex Group

### 3.2.1 Defining a complex group

(1) Specifying the complex group, including isogeny.

Usually the user will skip the root system step, so the basic commands parallel the ones in the previous section. What is needed to define a complex group is a Lie type and (optionally) a matrix.

(a) GC:=Group(A1,s): this is the simply connected group $\mathrm{SL}(2,\mathbb{C})$

(b) GC:=Group(A1): the default is simply connected

(c) GC:=Group(A1, a): the adjoint group $\mathrm{PSL}(2,\mathbb{C})$

(d) GC:=Group(A1.A2, s.a): $\mathrm{SL}(2,\mathbb{C}) \times PSL(3,\mathbb{C})$

(e) GC:=Group(A1,M): quotient of $\mathrm{SL}(2,\mathbb{C})$ specified by matrix M

(f) GC:=Group(A1.T1,[[1/2,1/2]]): $\mathrm{GL}(2,\mathbb{C})$

(g) GC:=Quotient(G,A): A is a finite subgroup of center, see 3.2(17)

(h) Classical groups.

i. GC:=ComplexSpecialLinearGroup(4)

ii. GC:=ComplexGeneralLinearGroup(4)

iii. GC:=ComplexProjectiveLinearGroup(4) (adjoint group of $\mathrm{SL}(4,\mathbb{C})$)

iv. GC:=ComplexSymplecticGroup(4)

v. GC:=ComplexProjectiveSymplecticGroup(4) (adjoint group of $\mathrm{Sp}(4,\mathbb{C})$)

vi. GC:=ComplexGeneralSymplecticGroup(4) ($GSp(4,\mathbb{C})$, one-dimensional center)

vii. GC:=ComplexSpinGroup(4)

viii. GC:=ComplexSpecialOrthogonalGroup(4)

ix. GC:=ComplexProjectiveOrthogonalGroup(4) (adjoint group of $SO(4,\mathbb{C})$)

x. GC:=ComplexGeneralOrthogonalGroup(4) (derived group is $O(4,\mathbb{C})$ with one-dimensional center)

(2) Some other ways of making a complex group

(a) Dual(GC)

(b) SimplyConnectedCover(GC): semisimple, simply connected cover of the derived group.

(c) GC:=Product(GC1,GC2)

(d) H:=Levi(GC,i,{1,3,6}): Levi subgroup given by list of vertices of the Dynkin diagram to include. See also Section 3.5 (5)

(e) H:=Levi(GC,e,{1,3,6}): Levi subgroup given by list of vertices of the Dynkin diagram to exclude. See also Section 3.5 (5)

(f) GC:=ComplexGroup(GGamma): we may have defined a Galois extended group without defining GC first (see Section 3.3)

(g) GC:=ComplexGroup(GR): we may have defined a real group GR without defining GC first (see Section 3.4)

### 3.2.2 Properties of a complex group

(1) DerivedGroup(GC)

(2) AdjointGroup(GC)

(3) DynkinDiagram(GC): a data structure, including a list of vertices

(4) Rank(GC)

(5) SemisimpleRank(GC)=Rank(DerivedGroup(GC))

(6) IsSemisimple(GC): Boolean

(7) IsSimplyConnected(GC): Boolean, true if and only if GC is semisimple and simply connected.
Note: IsSimplyConnected(SimplyConnectedCover(GC)) is always true

(8) HasSimplyConnectedDerivedGroup(GC)=IsSimplyConnected(DerivedGroup(GC))

(9) IsAdjoint(GC): Boolean

(10) Radical(GC): this is another connected reductive group (in fact a torus)

(11) IdentityComponentCenter(GC)=Radical(GC)

(12) MaximalCentralTorus(GC)=Radical(GC)

(13) ComponentGroupCenter(GC): this is a finite abelian group

(14) Center(GC). This should be both a data structure which can be used by the software and something human readable. For the human readable, one possibility is to describe Center(GC)/Radical(GC) as a finite abelian group. I'm not sure how to describe Center(GC), or give it as a data structure. See 3.2(2b).

(15) z:=CentralElement(GC,[1/2, 0/1, 3/17]): element of center of G of finite order.

I think this means an element of the center of the torus times a simply connected semisimple group of which GC is a quotient. This comes with a set of factors in a particular order. Each semisimple factor has a finite cyclic center, except $D_{2n}$ in which it is a Klein 4-group. [Another possibility is to write the (elements of finite order in the) center of GC itself as a product of finite cylice groups and $\mathbb{Q}/\mathbb{Z}$s. I prefer the former.]

In each simple factor except $D_{2n}$ the center is cyclic with a given isomorphism with $\frac{1}{n}\mathbb{Z}/\mathbb{Z}$. For example if n=6 we allow 0/6,1/6,...,5/6, we also allow 1/2=3/6. In a torus factor we have $\mathbb{Q}/\mathbb{Z}$.

We have two coordinates for each factor of type $D_{2n}$, i.e. [0,0],[0,1],[1,0], or [1,1]. The quotient by [1,1] is SO(4n); the others are "funny" quotients of Spin(2n).

This is used in defining strong real forms in Section (3.4.2).

(16) A:=CentralSubgroup(GC,z): (finite) central subgroup generated by z.

(17) A:=CentralSubgroup(GC,M): (finite) central subgroup generated by rows of matrix M.

## 3.3 Inner class of real forms

This is the **Galois Extended Group** $G^\Gamma$. This is a step the user will usually skip.
**Dangerous bend**: we have to assume a reductive group G is defined; to specify its inner class we need to remember the order of the factors of G. This also comes up in Section 3.4 (2).

(1) GGamma:=GaloisExtendedGroup(GCC,*inner class*): Here GC is a complex group as in Section 3.2, and *inner class* is a string of elements of {cCesu}:

   (a) c: compact

   (b) C: complex

   (c) e: equal rank=compact

   (d) s: split

   (e) u:unequal rank (only necessary in type $D_{4n}$ for the inner class of $SO(2n+1, 2n-1)$ which contains neither split, compact or complex; is allowed in other situations)

   These have substantial overlaps, and in any given case only certain arguments are allowed.

   For example:

   (a) GC:=ComplexGroup(A1)
   GaloisExtendedGroup(GC,s): the inner class of SL(2,$\mathbb{R}$)

   (b) GC:=ComplexGroup(A1.A2)
   GaloisExtendedGroup(GC, s.c): the inner class of SL(2,$\mathbb{R}$)×SU(3)

   (c) GC:=ComplexGroup(A1.A1)
   GaloisExtendedGroup(GC,CC): the inner class of SL(2,$\mathbb{C}$)

   (d) GC:=ComplexGroup(A1.T1,1/2.1/2)
   GaloisExtendedGroup(GC,ss): the inner class of GL$(2, \mathbb{R})$

   (e) GC:=ComplexClassicalGroup(GL,2)
   GaloisExtendedGroup(GC,ss): the inner class of GL$(2, \mathbb{R})$ (the torus implicitly comes last - the mnemonic is alphabetical order, T comes after A-G)

   (f) GC:=ComplexGroup(D4)
   GaloisExtendedGroup(GC,u): the inner class of Spin(7,1)

(2) GGamma=GaloisExtendedGroup(GR): we may have defined a real group GR without defining GGamma first (see Section 3.4)

## 3.4 Real Groups

### 3.4.1 Defining a Real Group

We can define $G(\mathbb{R})$ directly, or starting from a complex group, or starting from a Galois extended group.

(1) Defining the real form directly. Often the user will skip defining the reductive group and define the real group directly (also skipping the inner class step). The most common version will be to define a simple group directly (or a group like GL(n)).

(a) SL$(n, \mathbb{R})$ and relatives:

- GR:=RealSpecialLinearGroup(4) (SL$(4, \mathbb{R})$)
- GR:=RealGeneralLinearGroup(4) (GL$(4, \mathbb{R})$)
- GR:=RealProjectiveLinearGroup(4) (=PGL$(4, \mathbb{R})$= real points of PGL$(4, \mathbb{C})$ = PSL$(4, \mathbb{C})$, not SL$(4, \mathbb{R})/\pm 1$)

(b) Unitary groups:

- GR:=SpecialUnitaryGroup(3,5) (SU$(3, 5)$)
- GR:=UnitaryGroup(3,5) (U$(3, 5)$)
- GR:=ProjectiveUnitaryGroup(3,5) (real form of PSL$(5, \mathbb{C})$)

(c) Orthogonal groups

- GR:=RealSpinGroup(3,5) (Spin$(3, 5)$)
- GR:=RealSpecialOrthogonalGroup(3,5) (SO$(3, 5)$)
- GR:=RealProjectiveOrthogonalGroup(3,5) (adjoint group PO$(3, 5)$)
- GR:=RealGeneralOrthogonalGroup(3,5) (GO$(3, 5)$, one-dimensional split center)

(d) Symplectic groups

- GR:=RealSymplecticGroup(4) (split real group Sp$(4, \mathbb{R})$)
- GR:=RealProjectiveSymplectGroup(4) (real form of adjoint group PSp$(4, \mathbb{C})$)
- RealGeneralSymplecticGroup(4) (one dimensional split center)

- RealSymplecticGroup(4,3) (Sp(4, 3): I don't know if this is OK, or if there is better terminology. Note RealSymplecticGroup(4,0) is compact.)

For the exceptional groups we specify $K$ - RealExceptionalGroup(*type*[,*isogeny*],$K$):

(e) RealExceptionalGroup(E6,a,X) where X=E6,C4,A1.A5,T1.D5, or F4

(f) RealExceptionalGroup(E6,s,X) where X=E6,C4,A1.A5,T1.D5, or F4

(g) RealExceptionalGroup(E6,X) where X=E6,C4,A1.A5,T1.D5, or F4 (simply connected)

(h) RealExceptionalGroup(E7,a,X) where X=E7,A7,A1.D6, or T1.E6

(i) RealExceptionalGroup(E7,s,X) where X=E7,A7,A1.D6, or T1.E6

(j) RealExceptionalGroup(E7,X) where X=E7,A7,A1.D6, or T1.E6 (simply connected)

(k) RealForm(E8,X) where X=E7,A7,A1.D6, or T1.E6

(l) RealForm(F4,X) where X=F4,B4 or A1.C3

(m) RealForm(G2,X) where X=G2 or A1.A1

(2) We can also define a complex group GC:=ComplexGroup(...), and define the real form of G. See the Dangerous Bend at the beginning of Section 3.3. This skips the GaloisExtendedGroup step.

GC:=ComplexGroup(...)
GR:=RealForm(GC,*real form*)
where *real form* is a string of elements of {s,c,C}, one for each factor:

(a) s: split

(b) c: compact

(c) C: complex (error if not paired up correctly)

Examples:

(a) GC:=ComplexGroup(A1); GR:=RealForm(GC,s)=SL(2, $\mathbb{R}$)

(b) GC:=ComplexGroup(A1); GR:=RealForm(GC,c)=SU(2)

(c) GC:=ComplexGroup(A1.A1)
GR:=RealForm(GC, sc)=$\mathrm{SL}(2,\mathbb{R}) \times \mathrm{SU}(2)$

(d) GC:=ComplexGroup(A1.A1)
GR:=RealForm(GC, CC)=$\mathrm{SL}(2,\mathbb{C})$

(e) GC:=ComplexGroup(T1.A1)
GR:=RealForm(GC,ss)=$\mathbb{R}^\times \times \mathrm{SL}(2,\mathbb{R})$

(f) GC:=ComplexGroup(T1.A1,[1/2,1/2])
GR:=RealForm(GC,ss)=$\mathrm{GL}(2,\mathbb{R})$

(g) GC:=ComplexGroup(T1.A1.T1.A1);
GR:=RealForm(sCcC)=$\mathbb{R}^\times \times S^1 \times \mathrm{SL}(2,\mathbb{C})$ (complex factors don't have to be next to each other)

GR:=RealForm(GC) with no arguments gives a list from which to choose. [If GC is simple this is no problem. Otherwise it could be a very long list. Maybe we need an interactive mechanism to work one factor (or pair of factors) at a time].

(3) If GGamma=GaloisExtendedGroup(...) is defined, we can define certain real forms of it, more possibilities are allowed in this setting.

GGamma:=GaloisExtendedGroup(...)
GR:=RealForm(GGamma,*real form*) where *real form* is a string of elements of {qfscC},one for each factor of GGamma:

(a) q: quasisplit form in the inner class

(b) f: fundamental form in the inner class

(c) s: split (error if the inner class does not contain the split form)

(d) c: compact (error if the inner class does not contain the compact form)

(e) C: complex (error if not the right inner class or not paired up correctly)

GR:=RealForm(GGamma) with no arguments gives a list from which to choose. [If GC is simple this is no problem. Otherwise it could be a very long list. Maybe we need an interactive mechanism to work one factor (or pair of factors) at a time].

### 3.4.2 Strong real forms

Most users will not need to know about strong real forms. However some access to them should be made available.

There are three different notions:

(1) strong real form: $x^2 \in Z^\Gamma$,

(2) *reduced* strong real form: $x^2 \in Z_0$,

(3) real form: involution of $G(\mathbb{C})$.

Here $Z_0$ is a set of representatives of the finite set $Z^\Gamma/\{z\theta(z)\}$. This the basis of the *reduced parameter space* [**?**, Section 13]. In each case equivalence is by conjugation by $G(\mathbb{C})$. The map

The map $x \to \text{int}(x)$ from {strong real forms} to {real forms} is surjective, but far from injective.

The map $x \to \text{int}(x)$ from {reduced strong real forms} to {real forms} is surjective. It is close to being injective.

The simplest version of the algorithm works with a strong real forms (in [**?**] these are called strong involutions). This is a large an unwieldy set (it may be infinite). Computationally it is better to work with reduced strong real forms, which is a finite set, although a choice of $Z_0$ is required.

I think what the software does is this. It computes a set of representatives of the finite set $Z_0 = Z^\Gamma/\{z\theta(z)\}$. Then for each real form it picks a corresponding reduced strong real form $x$ (i.e. $x^2 \in Z_0$). The `blocksizes` command, for example, prints out one row for each real form. I believe the `strongreal` command prints out a list of reduced strong real forms.

Fix GGamma=GaloisExtendedGroup(...).

**Commands:**

(1) SRF:=StrongRealForms(GGamma): a list of all reduced strong real forms in this inner class.

(2) GR:=SRF(j) is an element of this list. I think that if i,j map to the same real form, the SRF(i) is indistinguishable from SRF(j).

(3) StrongRealForms(GGamma,z): list of strong real forms with $x^2 = z$.

### 3.4.3 Properties of a real group

(1) Some other basic structure theory of a real group; these functions take a real group as an argument

   (a) ComponentGroup(GR): this is a two-group

   (b) FundamentalGroup(GR)

   (c) ReducedComponentGroup(GR): this is $G(\mathbb{R})/Z(G(\mathbb{R}))G(\mathbb{R})^0$

   (d) MaximalCompact(GR): this is a real group. I don't know what to ask for here. Certainly the type of the complex Lie algebra. David and Alfred have something to say about this; see Section 6.

   (e) ComplexifiedMaximalCompact(GR): this is the $G(\mathbb{C})^\theta$, and again I don't know what to ask for

   (f) IdentityComponentComplexifiedMaximalCompact(GR): the identity component of $G(\mathbb{C})^\theta$. This is a connected reductive algebraic group, as in Section 3.2, together with a real form as in Section 3.4. Its real points contain the identity component of the MaximalCompact(GR).

## 3.5 Cartan Subgroups

Some functions take GGamma, some GR.

(1) C:=Cartans(GR): returns a list of Cartans

(2) H:=FundamentalCartan(GGamma) or FundamentalCartan(GR): fundamental, i.e. most compact, Cartan

(3) H:=MostSplitCartan(GR): most split Cartan of GR

(4) H:=MostSplitCartan(GGamma)
   =MostSplitCartan(RealForm(GGamma,q))
   =most split Cartan of quasisplit form in the inner class

(5) M:=RealLevi(H,GR): the Levi factor given by H, i.e. $\mathrm{Cent}_G(A)$. This is the Levi factor of a real parabolic, and is a connected reductive group.

(6) MGalois:=RealLevi(H,GGalois). This is the Galois Extended Group of RealLevi(H,GR) for any real form GR in the inner class given by GGamma.

(7) L:=ImaginaryLevi(H,GR): this is the Levi factor $L = \mathrm{Cent}_G(T)$, a theta-stable parabolic.

(8) LGalois:=ImaginaryLevi(H,GGalois)

(9) SplitRank(H) and CompactRank(H): dimension of T and A. We've decided to drop SplitPart(H) and CompactPart(H) as unnecessary.

## 3.6 Weyl Groups

Most arguments require a Cartan H to be defined.

(1) WeylGroup(GC) $\quad (= W(G(\mathbb{C}), H(\mathbb{C})))$

(2) Weyl Group(GR)=WeylGroup(ComplexGroup(GR))

(3) RealWeylGroup(H) $\quad (= W(G(\mathbb{R}), H(\mathbb{R})) = W(K, H))$

(4) RealRootWeylGroup(H): the Weyl group of the real roots

(5) ImaginaryRootWeylGroup(H): the Weyl group of the imaginary roots

(6) TwoGroupPartWeylGroup(H): the real Weyl group has the structure

$$W(G(\mathbb{R}), H(\mathbb{R})) \simeq (W^C)^\theta \ltimes ((A \rtimes W_{ic}) \times W^R)$$

where A=TwoGroupPartWeylGroup(H) is a two-group.

Note: A=ReducedComponentGroup(ImaginaryLevi(H))

(7) WeylGroupTheta(H): the group $W^\theta$

## 3.7 Roots, weights

The software provides bases of $X^*(H)$ and $X_*(H)$. Everything is done in these bases. The only information the user needs about these bases is the matrix of roots and coroots in these bases (also fundamental weights and coweights would be nice).

We've dropped all commands involving action.

(1) Roots(G): list of all roots

(2) PositiveRoots(G): list of all positive roots

(3) SimpleRoots(G): list of simple roots

(4) FundamentalWeights(G): list of fundamental weights

(5) CoRoots(G): list of all coroots

(6) PositiveCoRoots(G): list of all positive coroots

(7) SimpleCoRoots(G): list of simple coroots

(8) FundamentalCoWeights(G): list of fundamental coweights

(9) HalfSumPostiveRoots(G)

(10) HalfSumPositiveCoRoots(G)

(11) HighestRoot(G) (here and the next three $G_{der}$ must be simple)

(12) HighestShortRoot(G)

(13) HighestCoRoot(G)

(14) HighestShortCoRoot(G)

# 4 K orbits on G/B

Fix a real group GR, with associated K.

(1) Orbits:=KGB(GR): list of $K\backslash G/B$ orbits, i.e. the x's

(2) O:=Orbit(3): third item in the list

(3) Dimension(O): dimension of orbit O

(4) Cross(i,O): cross action of $i^{th}$ simple root on O. Returns a single orbit.

(5) Cayley(i,O) : Cayley transform of $i^{th}$ simple root (real or non-compact imaginary) on O. Returns a set of one or two orbits. [Or should it return a set of two elements, the second can be undefined?]

(6) Closure(O): list of orbits in the closure of O

(7) CoClosure(O): list of orbits containing O in the closure

(8) MaximalOrbit: (unique) maximal orbit

(9) MinimalOrbits: list of minimal orbits

(10) HasseDiagram(KGB(GR)): matrix of elementary (codimension 1) closure relations - given by Cayley transform, cross actions *and* the parallelogram relation.

# 5 Representation Theory

We work mostly in a given inner class of groups, i.e. with a fixed GaloisExtendedGroup GGamma. Some commands also require a real form GR and/or a choice of real form of the dual group.

See Section 3.4.2 on strong real forms and reduced strong real forms. In particular the software picks a set $Z_0$ of representatives of $Z^\Gamma/\{z\theta(z)\}$.

## 5.1 Commands not requiring an infinitesimal character

Currently the software defines a set of parameters which describe representations at any of a set of regular integral infinitesimal characters in a tranlsate of $X^*(H)$. The theory at any two such infinitesimal characters are related by a translation functor.

(1) L:=TranslationBlocks(GGamma): list of all blocks of all reduced strong real forms of G, i.e. $x^2 \in Z_0$. This is a finite set. This will give a representative of every "type" of block (some more than once). Perhaps the output should include number of representation in the blocks, for help with picking one. The infinitesimal character is not set, for each block it is allowed to lie in a certain lattice.

This should be a matrix, similar to the current `blocksizes` command. (Note: `blocksizes` only prints one row for each real form, not reduced real form).

(2) L:=TranslationBlocks(GGamma, z): subset of the previous, where we assume $x^2 = z$; more precisely $x^2$ and $z$ have the same image in $Z^\Gamma/\{z\theta(z)\}$.

(3) TB:=L(3,2): specify a translation block to work on, given by the row and column in the output of TranslationBlocks(...).

(4) CentralElement(TB): the central element z=$x^2$ of TB

(5) RealForm(TB): the real group of which TB is a translation block

(6) DualRealForm(TB): the real group of which TB is the dual translation block

(7) P:=Parameters(TB): a list of pairs (x,y). This is a list of a pair of numbers, corresponding to the output of KGB(GR, z) (see Section 4).

(8) gamma:=P(3): parameter number 3 (starting at 0)

(9) gamma:=P(3,4): parameter indexed by (x,y)=(3,4)

(10) TranslationBlock(gamma): translation block containing gamma.

(11) Cross(i, gamma): cross action of $i^{th}$ simple root on gamma

(12) Cayley(i, gamma) : Cayley transform of $i^{th}$ simple root (real or nci) on gamma. Returns a list of two elements, the second of which may be "Undefined".

(13) Descent(gamma): returns list $\{a_1, \ldots, a_n\}$ corresponding to simple roots. Each $a_i$ is: i1,i2,ic, r1,r2,rn,C+,C-.

(14) Length(gamma)

(15) TwistedInvolution(gamma): image of gamma in $\mathcal{I}_W$ (twisted involutions in the Weyl group), as product of simple reflections times the fundamental element of $\mathcal{I}_W$

(16) CrossStabilizer(gamma): stabilizer of gamma in the cross action

(17) RealForm(gamma)=RealForm(TranslationBlock(gamma))

(18) DualRealForm(gamma)=DualRealForm(TranslationBlock(gamma))

(19) Cartan(gamma): what Cartan this lives on

(20) Maximal(TB): a maximal element of TB

(21) Minimal(TB): a minimal element of TB

## 5.2 Commands requiring an infinitesimal character

To get an actual representation one has to specify an infinitesimal character. All of the preceding commands should also apply at this level. There are also a few additional commands available.

I'm not sure the best way to specify an infinitesimal character $\lambda \in \mathfrak{h}^*$. We have a given basis of $X^*(H)$, but it is very hard to use. The only thing that matters about $\lambda$ is the set of integers $\langle \lambda, \alpha^\vee \rangle$ for $\alpha$ simple.

We are *not* going to define Cayley transforms and the cross action on the level of representations, only on parameters.

(1) lambda:=($a_1$,...,$a_n$) where n is the dimension of $H$: element of $\mathfrak{h}^*$ in terms of the given basis of $X^*(H)$. The $a_i$ are rational numbers. Not required to be dominant.

(2) lambda:=[$a_1$,...,$a_m$] where m is the semisimple rank: element of $\mathfrak{h}^*$ where in terms of the basis of fundamental weights. The $a_i$ are rational numbers. Not required to be dominant. [Better notation for this?]

(3) IsIntegral(lambda): true if $\langle \lambda, \alpha^\vee \rangle \in \mathbb{Z}$ for all simple roots

(4) IsDominant(lambda): true if lambda is dominant

(5) MakeDominant(lambda): put lambda in the dominant chamber

(6) IsConjugate(lambda,lambda′): true if conjugate by W.

(7) lambda:=AllowedInfinitesimalCharacter(TB): choose a dominant infinitesimal character which is allowed for TB. Should be 0 on the center of $\mathfrak{g}$, and as small as possible.

(8) B:=Block(TB, lambda): block of parameters obtained by specializing the translation block TB to lambda (error if lambda is not allowed for TB).

(9) Central Element, ..., Minimal should apply to a block B, as well as a translation block TB.

(10) I:=StandardRepresentation(gamma, lambda): standard representation associated to parameter gamma of translation block TB, at infinitesimal character lambda. Error if lambda is not allowed.

(11) I1:=StandardRepresentation(gamma), I:=I1(lambda): same as previous

(12) Irreducible(gamma, lambda): irreducible representation associated to parameter gamma of translation block TB, at infinitesimal character lambda. Error if lambda not allowed.

(13) pi1:=IrreducibleRepresentation(gamma), pi:=pi1(lambda): same as previous

(14) StandardCoherent(I): return coherent continuation of standard representation I, i.e. a list $((a_1, I_1), \ldots, (a_n, I_n))$ where each $a_i \in \mathbb{Z}$ and $I_i$ is a standard representation. Corresponds to current `wgraph` command.

(15) IrreducibleCoherent(pi): return coherent continuation of irreducible representation pi, i.e. a list $((a_1, pi_1), \ldots, (a_n, pi_n))$ where each $a_i \in \mathbb{Z}$ and $pi_i$ is an irreducible representation. Computed from output of current `block` command.

(16) UnitaryRepresentations(B): the unitary representations in B. These are the representations $A_{\mathfrak{q}}(\lambda)$. This depends on the infinitesimal character lambda.

(17) StandardRepresentation(pi)

(18) IrreducibleRepresentation(I)

(19) InfinitesimalCharacter(I)

(20) InfinitesimalCharacter(pi)

(21) gamma:=Parameter(I) parameter of standard representation I. This might require constructing the block.

(22) gamma:=Parameter(pi) parameter of irreducible representation pi. This might require constructing the block.

(23) TrivialRepresentation(GR)

## 5.3 Dictionary

Commands to convert to and from human-readable Langlands parameters. Much more work to be done here...

# 6 Representations of K, and K-types of representations of G

See [**?**]. This section is under construction and awaiting further refinements by David and Alfred.

(1) mu:=KType(...) where ... is the data of [**?**, Theorem 14.2].

(2) lambda:=HighestWeight(mu) returns the character of $T_f$ of [**?**, Theorem 3.10] (and the subsequent paragraph), but not the character of $T_{fl}$.

(3) lambda:=LieAlgebraHighestWeight(mu): returns an element of $\mathfrak{t}^*$ where $T$ is a Cartan subgroup of $K$ and $\mathfrak{t} = \mathrm{Lie}_{\mathbb{C}}(T)$.

(4) S:=LowestKTypes(pi): here pi is an irreducible or standard representation of GR. Returns a set of K-types parametrized as in (1).

(5) mult(mu,I): multiplicity of K-type mu in standard representation I.

(6) mult(mu,pi): multiplicity of K-type mu in irreducible representation pi.

# 7 Kazhdan-Lusztig Polynomials

Fix a *block* or a *translation block* B (see Section 5). This comes with a list of parameters $0 \leq i \leq n$, which are also labelled (x,y).

(1) KazhdanLusztigPolynomial(i,j): Kazhdan-Lusztig polynomial for parameters $0 \leq i, j \leq n$.

(2) KazhdanLusztigPolynomial(x,y,x′,y′): Kazhdan-Lusztig polynomial for parameters (x,y), (x′,y′).

(3) KazhdanLusztigBasis(B) (klbasis in the current software)

(4) KazhdanLusztigBasis(B,i):

(5) KazhdanLusztigBasis(B,x,y): all KazhdanLusztigPolynomial(x,y,x′, y′) with given x,y.

(6) KazhdanLusztigDistinct(B) (kllist in current software)

(7) PrimitiveKazhdanLusztigPolynomials(B) (primkl in current software)

(8) StandardMultiplicity(I,pi): multiplicity of standard representation I in character formula of irreducible representation pi

(9) IrreducibleMultiplicity(pi,I): multiplicity of irreducible representation in standard representation I

(10) StandardMultiplicity(i,j): multiplicity of standard representation I(i) in irreducible representation pi(j)

(11) StandardMultiplicity(i,j,k,l): multiplicity of standard representation I(i,j) in irreducible representation pi(k,l)

(12) IrreducibleMultiplicity(i,j): multiplicity of irreducible representation pi(i) in standard representation I(j)

(13) IrreducibleMultiplicity(i,j,k,l): multiplicity of irreducible representation pi(i,j) in standard representation I(k,l)

(14) WGraph(B)

(15) WCells(B)

# 8 Print Commands

I'm not sure what to specify here. Most of the commands discussed above return a data structure. Can we have a single `print` command, which does various things depending on its arguments?

# References

[1] Jeffrey Adams. Parameters for representations of real groups. preprint, atlas web site.