Affine Kac-Moody Lie Algebras in MAGMA

Dan Roozemond Computational Algebra Group University of Sydney

Computation in Coxeter groups and Kac-Moody algebras Sydney, 16 November 2011









Serre relations

For A a generalized Cartan matrix, define $\mathfrak{g}'(A)$ to be the Lie algebra generated by 3n generators e_i, f_i, h_i and relations:

$$\begin{cases} [h_i, h_j] = 0, & [e_i, f_i] = h_i, & [e_i, f_j] = 0 & (i \neq j) \\ [h_i, e_j] = a_{ij}e_j, & [h_i, f_j] = -a_{ij}f_j, \\ (ade_i)^{1-a_{ij}}e_j = 0, & (adf_i)^{1-a_{ij}}f_j = 0. & (i \neq j) \end{cases}$$





Three classes

- (a) Finite-dimensional (if there is a positive integral vector v such that Av > 0);
- (b) Affine (if there is a positive integral vector v such that Av = 0);
- (c) Complicated (if there is a positive integral vector v such that Av < 0).





Three classes

(a) Finite-dimensional (if there is a positive integral vector v such that Av > 0);

Affine (if there is a positive integral vector v such that Av = 0);

(c) Complicated (if there is a positive integral vector v such that Av < 0).



For a generalized Cartan matrix A of type $X_l^{(1)}$, take \mathfrak{g}_0 to be the finite-dimensional Lie algebra of type X_l and take

 $\mathfrak{g} = \mathfrak{g}_0 \otimes \mathbb{C}[t, t^{-1}] \oplus \mathbb{C}c \oplus \mathbb{C}d$

with multiplication on ${\mathfrak g}$ given by

$$\begin{bmatrix} t^k \otimes x \oplus \lambda c \oplus \mu d, t^{k_1} \otimes y \oplus \lambda_1 c \oplus \mu_1 d \end{bmatrix} = \\ (t^{k+k_1} \otimes [x, y] + \mu k_1 t^{k_1} \otimes y - \mu_1 k t^k \otimes x) \oplus k \delta_{k, -k_1}(x|y)c$$

Theorem [Kac90, Theorem 7.4]

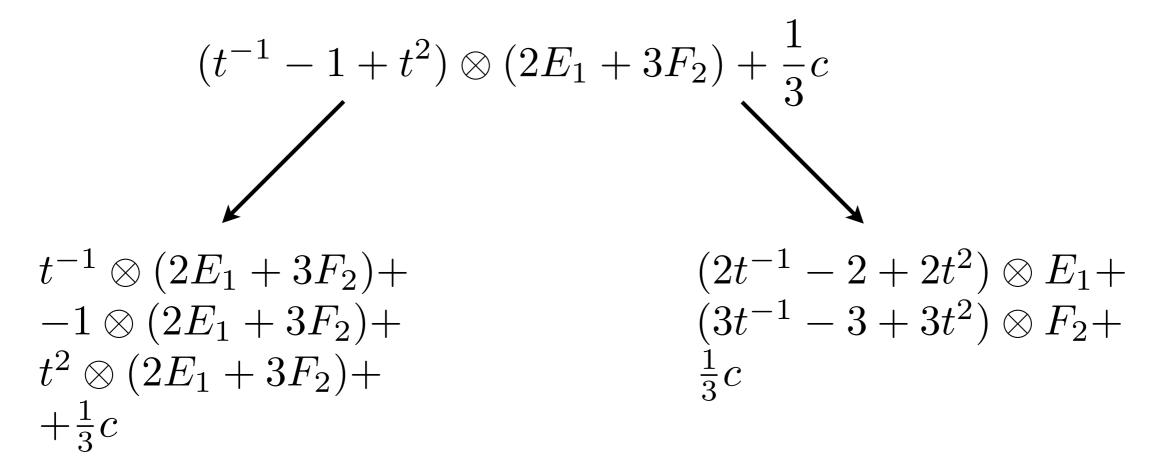
 \mathfrak{g} is the affine Kac-Moody algebra associated to A.



Generators

$$\mathfrak{g} = \mathfrak{g}_0 \otimes \mathbb{C}[t, t^{-1}] \oplus \mathbb{C}c \oplus \mathbb{C}d$$

- Arithmetic is very straightforward; except equality testing
- 2 normal forms:





Demo



- We can currently do a rather small number of things,
- but I'd be interested in what you would want it to do!