Groups of Lie Type in MAGMA

Overview on some algorithms

Sergei Haller

Magma group
School of Mathematics and Statistics
University of Sydney

Magma 2006, TU Berlin
August 1, 2006

Outline

Outline

Lie Theory in Magma

- Root systems and root data
- Coxeter groups

  (as permutation groups, as matrix reflection groups, as FP groups)
- Groups of Lie type (split and twisted)

  (Steinberg presentation, linear representations)
- Lie Algebras

  (as structure constants algebras, as matrix algebras)
- Universal Enveloping Algebras, Quantum Groups ...

Lie Theory in Magma

- Root systems and root data
- Coxeter groups

  (as permutation groups, as matrix reflection groups, as FP groups)
- Groups of Lie type (split and twisted)

  (Steinberg presentation, linear representations)
- Lie Algebras

  (as structure constants algebras, as matrix algebras)
- Universal Enveloping Algebras, Quantum Groups ...

Lie Theory in Magma

- Root systems and root data
- Coxeter groups

  (as permutation groups, as matrix reflection groups, as FP groups)
- Groups of Lie type (split and twisted)

  (Steinberg presentation, linear representations)
- Lie Algebras

  (as structure constants algebras, as matrix algebras)
- Universal Enveloping Algebras, Quantum Groups ...

Lie Theory in Magma

- Root systems and root data
- Coxeter groups

  (as permutation groups, as matrix reflection groups, as FP groups)
- Groups of Lie type (split and twisted)

  (Steinberg presentation, linear representations)
- Lie Algebras

  (as structure constants algebras, as matrix algebras)
- Universal Enveloping Algebras, Quantum Groups ...

Lie Theory in Magma

- Root systems and root data
- Coxeter groups

  (as permutation groups, as matrix reflection groups, as FP groups)
- Groups of Lie type (split and twisted)

  (Steinberg presentation, linear representations)
- Lie Algebras

  (as structure constants algebras, as matrix algebras)
- Universal Enveloping Algebras, Quantum Groups . . .

Lie Theory in Magma

- Root systems and root data
- Coxeter groups

  (as permutation groups, as matrix reflection groups, as FP groups)
- Groups of Lie type (split and twisted)

  (Steinberg presentation, linear representations)
- Lie Algebras

  (as structure constants algebras, as matrix algebras)
- Universal Enveloping Algebras, Quantum Groups ...

Lie Theory in Magma

- Root systems and root data
- Coxeter groups

  (as permutation groups, as matrix reflection groups, as FP groups)
- Groups of Lie type (split and twisted)

  (Steinberg presentation, linear representations)
- Lie Algebras

  (as structure constants algebras, as matrix algebras)
- Universal Enveloping Algebras, Quantum Groups ...

Outline

Groups of Lie Type

- Group of rational points of a linear algebraic group.
  (Subgroup of some $\mathrm{GL}_n(k)$ given by polynomials)
- Reductive groups are classified by root data.

Cartan type	Group
A_n	$\mathrm{SL}_{n+1}(k)$
B_n	$\mathrm{SO}_{2n+1}(k)$
C_n	$\mathrm{Sp}_{2n}(k)$
D_n	$\mathrm{SO}_{2n}(k)$
G_2, F_4, E_6, E_7, E_8	exceptional

Groups of Lie Type

- Group of rational points of a linear algebraic group.
  (Subgroup of some $\mathrm{GL}_n(k)$ given by polynomials)
- Reductive groups are classified by root data.

Cartan type	Group
A_n	$\mathrm{SL}_{n+1}(k)$
B_n	$\mathrm{SO}_{2n+1}(k)$
C_n	$\mathrm{Sp}_{2n}(k)$
D_n	$\mathrm{SO}_{2n}(k)$
G_2, F_4, E_6, E_7, E_8	exceptional

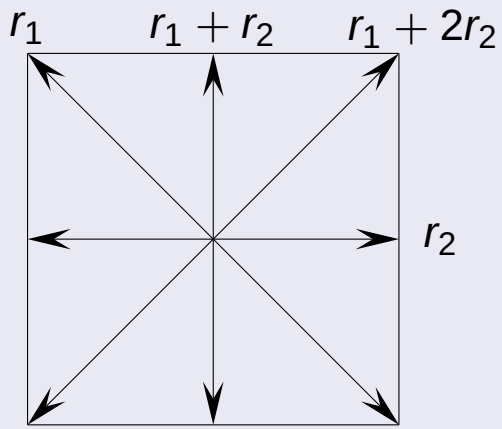
Groups of Lie Type

- Group of rational points of a linear algebraic group.
  (Subgroup of some $\mathrm{GL}_n(k)$ given by polynomials)
- Reductive groups are classified by root data.

Cartan type	Group
A_n	$\mathrm{SL}_{n+1}(k)$
B_n	$\mathrm{SO}_{2n+1}(k)$
C_n	$\mathrm{Sp}_{2n}(k)$
D_n	$\mathrm{SO}_{2n}(k)$
G_2, F_4, E_6, E_7, E_8	exceptional

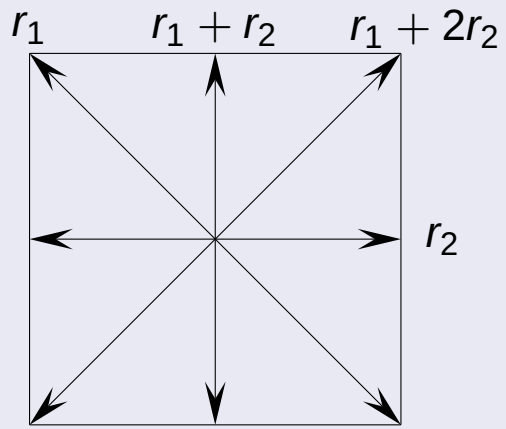
Root Systems

Root System of type B_2



Root Systems

Root System of type B_2

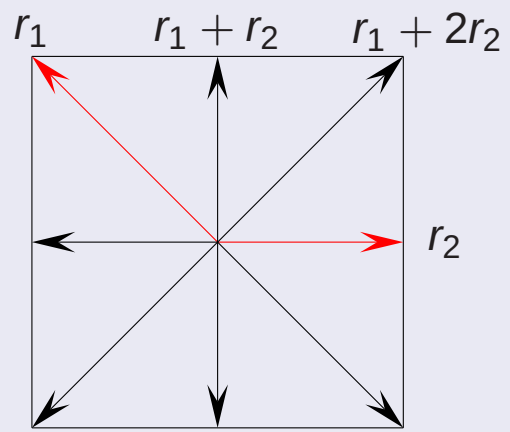


Definition “by example”:

- Π – Set of fundamental roots
- Φ^+ – Set of positive roots
- Φ^- – Set of negative roots

Root Systems

Root System of type B_2

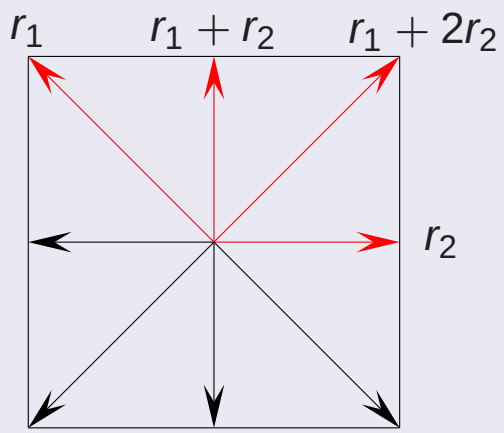


Definition “by example”:

- Π – Set of fundamental roots
- Φ^+ – Set of positive roots
- Φ^- – Set of negative roots

Root Systems

Root System of type B_2

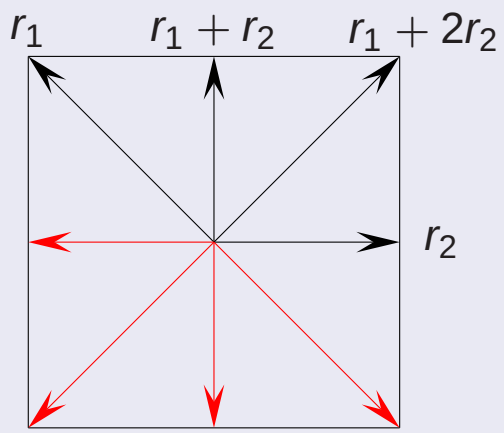


Definition “by example”:

- Π – Set of fundamental roots
- Φ^+ – Set of positive roots
- Φ^- – Set of negative roots

Root Systems

Root System of type B_2

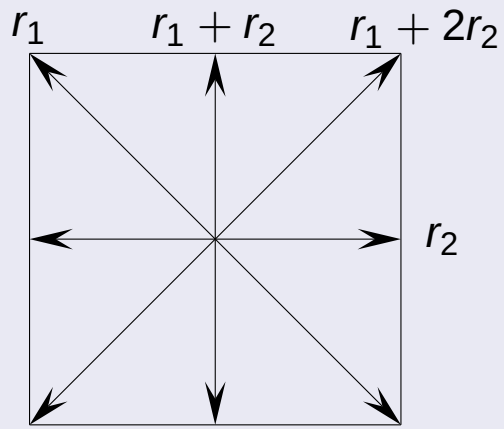


Definition “by example”:

- Π – Set of fundamental roots
- Φ^+ – Set of positive roots
- Φ^- – Set of negative roots

Root Systems

Root System of type B_2



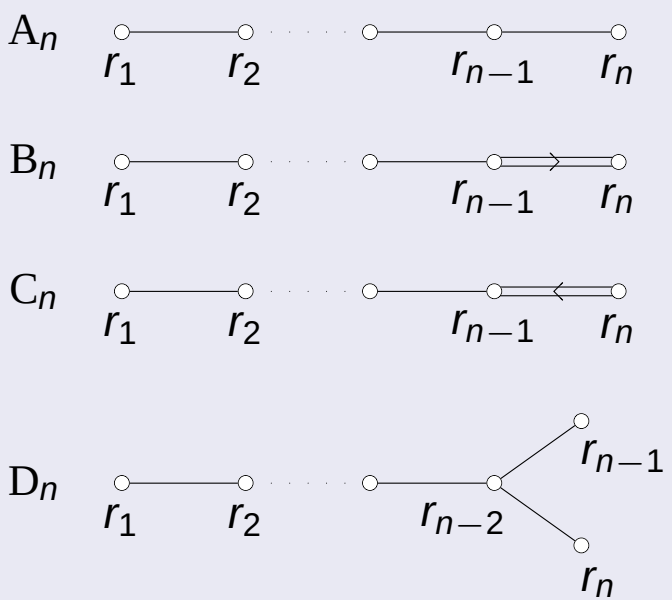
Definition “by example”:

- Π – Set of fundamental roots
- Φ^+ – Set of positive roots
- Φ^- – Set of negative roots

Dynkin diagrams

Dynkin diagrams of reduced irreducible root systems.

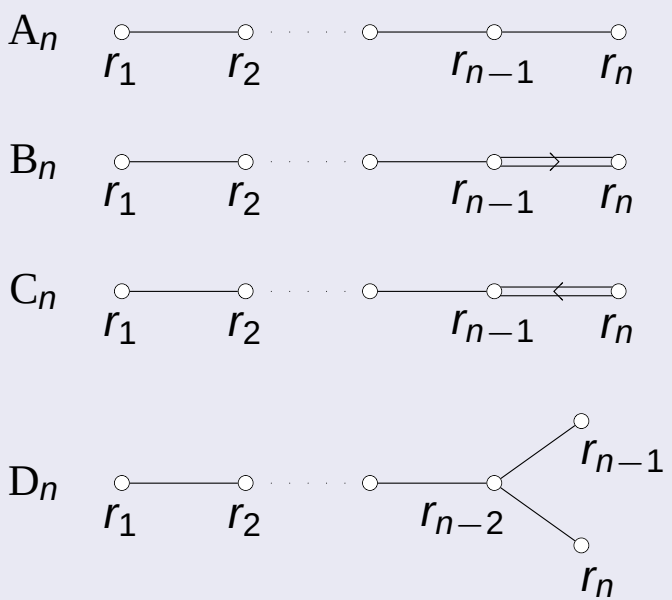
Classical types



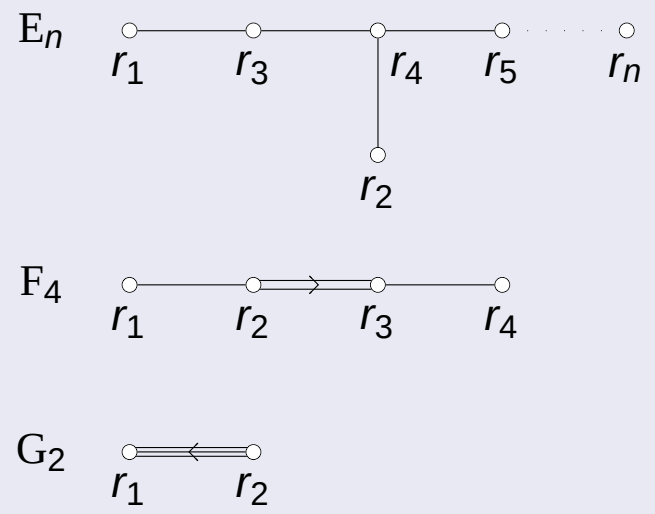
Dynkin diagrams

Dynkin diagrams of reduced irreducible root systems.

Classical types



Exceptional types



Steinberg Presentation

- Generators:
  - $x_r(t)$ with $r \in \Phi^+$ and $t \in k$ (unipotent)
  - $h_r(t)$ with $r \in \Pi$ and $t \in k^*$ (torus)
  - n_r with $r \in \Phi$ (Coxeter group elements)
- Relations:

$$x_r(t)x_r(u) = x_r(t+u)$$

$$[x_r(t), x_s(u)] = \prod_{i,j>0} x_{ir+js}(C_{rsij}t^i u^j)$$

$$\ldots$$

Steinberg Presentation

- Generators:
  - $x_r(t)$ with $r \in \Phi^+$ and $t \in k$ (unipotent)
  - $h_r(t)$ with $r \in \Pi$ and $t \in k^*$ (torus)
  - n_r with $r \in \Phi$ (Coxeter group elements)
- Relations:

$$x_r(t)x_r(u) = x_r(t+u)$$

$$[x_r(t), x_s(u)] = \prod_{i,j>0} x_{ir+js}(C_{rsij}t^i u^j)$$

$$\ldots$$

Unique Bruhat Decomposition

- $x = uhnu'$
  - u – product of $x_r(t)$
  - h – product of $h_r(t)$
  - n – product of n_r
  - u' – product of $x_r(t)$ [for a subset of Φ^+]
- Positive roots ordered with respect to their height. I.e., $r < s < r + s$ whenever $r < s$ and $r + s \in \Phi^+$.
- Unique decomposition of unipotent elements:

$$\prod_{r \in \Phi^+} x_r(t_r),$$

with roots in the above ordering.

Unique Bruhat Decomposition

- $x = uhnu'$
  - u – product of $x_r(t)$
  - h – product of $h_r(t)$
  - n – product of n_r
  - u' – product of $x_r(t)$ [for a subset of Φ^+]
- Positive roots ordered with respect to their height. I.e., $r < s < r + s$ whenever $r < s$ and $r + s \in \Phi^+$.
- Unique decomposition of unipotent elements:

$$\prod_{r \in \Phi^+} x_r(t_r),$$

with roots in the above ordering.

Unique Bruhat Decomposition

- $x = uhnu'$
  - u – product of $x_r(t)$
  - h – product of $h_r(t)$
  - n – product of n_r
  - u' – product of $x_r(t)$ [for a subset of Φ^+]
- Positive roots ordered with respect to their height. I.e., $r < s < r + s$ whenever $r < s$ and $r + s \in \Phi^+$.
- Unique decomposition of unipotent elements:

$$\prod_{r \in \Phi^+} x_r(t_r),$$

with roots in the above ordering.

Unique Bruhat Decomposition

- $x = uhnu'$
  - u – product of $x_r(t)$
  - h – product of $h_r(t)$
  - n – product of n_r
  - u' – product of $x_r(t)$ [for a subset of Φ^+]
- Positive roots ordered with respect to their height. I.e., $r < s < r + s$ whenever $r < s$ and $r + s \in \Phi^+$.
- Unique decomposition of unipotent elements:

$$\prod_{r \in \Phi^+} x_r(t_r),$$

with roots in the above ordering.

Question

How to compute?

Relations for Collection

$$x_r(0) = 1$$

$$x_r(t)x_r(u) = x_r(t+u)$$

$$x_r(t)x_s(u) = x_s(u)x_r(t) \qquad [\text{if } r+s \notin \Phi]$$

$$\begin{aligned} x_r(t)x_s(u) &= x_s(u)x_r(t) \cdot [x_r(t), x_s(u)] \\ &= x_s(u)x_r(t) \cdot \prod_{i,j>0} x_{ir+js}(C_{rsij}t^i u^j) \end{aligned}$$

Relations for Collection

$$x_r(0) = 1$$

$$x_r(t)x_r(u) = x_r(t+u)$$

$$x_r(t)x_s(u) = x_s(u)x_r(t) \qquad [\text{if } r+s \notin \Phi]$$

$$\begin{aligned} x_r(t)x_s(u) &= x_s(u)x_r(t) \cdot [x_r(t), x_s(u)] \\ &= x_s(u)x_r(t) \cdot \prod_{i,j>0} x_{ir+js}(C_{rsij}t^i u^j) \end{aligned}$$

Relations for Collection

$$x_r(0) = 1$$

$$x_r(t)x_r(u) = x_r(t+u)$$

$$x_r(t)x_s(u) = x_s(u)x_r(t) \qquad [\text{if } r+s \notin \Phi]$$

$$\begin{aligned} x_r(t)x_s(u) &= x_s(u)x_r(t) \cdot [x_r(t), x_s(u)] \\ &= x_s(u)x_r(t) \cdot \prod_{i,j>0} x_{ir+js}(C_{rsij}t^i u^j) \end{aligned}$$

Relations for Collection

$$x_r(0) = 1$$

$$x_r(t)x_r(u) = x_r(t+u)$$

$$x_r(t)x_s(u) = x_s(u)x_r(t) \qquad [\text{if } r+s \notin \Phi]$$

$$x_r(t)x_s(u) = x_s(u)x_r(t) \cdot [x_r(t), x_s(u)]$$
$$= x_s(u)x_r(t) \cdot \prod_{i,j>0} x_{ir+js}(C_{rsij}t^i u^j)$$

Relations for Collection

$$x_r(0) = 1$$

$$x_r(t)x_r(u) = x_r(t+u)$$

$$x_r(t)x_s(u) = x_s(u)x_r(t) \qquad [\text{if } r+s \notin \Phi]$$

$$\begin{aligned} x_r(t)x_s(u) &= x_s(u)x_r(t) \cdot [x_r(t), x_s(u)] \\ &= x_s(u)x_r(t) \cdot \prod_{i,j>0} x_{ir+js}(C_{rsij}t^i u^j) \end{aligned}$$

Collection in General

Algorithm (Collection of u)

1. if u is in normal form, then stop
2. choose any non-normal subword of length 1 or 2 and collect it using above relations
3. go to step 1

- Always terminates! [Sims, 1987]
- Choice in step 2 is crucial for efficiency

Collection in General

Algorithm (Collection of u)

1. if u is in normal form, then stop
2. choose any non-normal subword of length 1 or 2 and collect it using above relations
3. go to step 1

- Always terminates! [Sims, 1987]
- Choice in step 2 is crucial for efficiency

Collection in General

Algorithm (Collection of u)

1. if u is in normal form, then stop
2. choose any non-normal subword of length 1 or 2 and collect it using above relations
3. go to step 1

- Always terminates! [Sims, 1987]
- Choice in step 2 is crucial for efficiency

Collection To Left/From Right

Input: $\prod_{i=1}^{m} x_{r_i}(t_i)$ Output: $\prod_{r=1}^{N} x_r(t_r)$, where $N := |\Phi^+|$

Algorithm (Collection To Left) based on [Hall, 1934]
Algorithm (Collection From Right) [Neubüser, 1961]

```
for r in [1..N] do
    collect all terms x_{r_i}(t_i) with r_i = r to left
    using the above relations
end for
```

before step r: $\prod_{s=1}^{r-1} x_s(t_s) \cdot \prod_{i=1}^{m} x_{r_i}(t_i)$ $[r_i \geq r]$

after step r: $\prod_{s=1}^{r-1} x_s(t_s) \cdot x_r(\sum_{r_i=r} t_i) \cdot \prod_{i=1}^{m'} x_{r_i}(t_i)$ $[r_i > r]$

Collection To Left/From Right

Input: $\prod_{i=1}^{m} x_{r_i}(t_i)$ Output: $\prod_{r=1}^{N} x_r(t_r)$, where $N := |\Phi^+|$

Algorithm (Collection To Left) based on [Hall, 1934]
Algorithm (Collection From Right) [Neubüser, 1961]

```
for r in [1..N] do
    collect all terms x_{r_i}(t_i) with r_i = r to left
    using the above relations
end for
```

before step r: $\prod_{s=1}^{r-1} x_s(t_s) \cdot \prod_{i=1}^{m} x_{r_i}(t_i)$ $[r_i \geq r]$

after step r: $\prod_{s=1}^{r-1} x_s(t_s) \cdot x_r(\sum_{r_i=r} t_i) \cdot \prod_{i=1}^{m'} x_{r_i}(t_i)$ $[r_i > r]$

Collection To Left/From Right

Input: $\prod_{i=1}^{m} x_{r_i}(t_i)$ Output: $\prod_{r=1}^{N} x_r(t_r)$, where $N := |\Phi^+|$

Algorithm (Collection To Left) based on [Hall, 1934]
Algorithm (Collection From Right) [Neubüser, 1961]

```
for r in [1..N] do
    collect all terms x_{r_i}(t_i) with r_i = r to left
    using the above relations
end for
```

before step r: $\prod_{s=1}^{r-1} x_s(t_s) \cdot \prod_{i=1}^{m} x_{r_i}(t_i)$ $[r_i \geq r]$

after step r: $\prod_{s=1}^{r-1} x_s(t_s) \cdot x_r(\sum_{r_i=r} t_i) \cdot \prod_{i=1}^{m'} x_{r_i}(t_i)$ $[r_i > r]$

Collection From Left

Input: $x = \prod_{i=1}^{m} x_{r_i}(t_i)$ Output: $x = \prod_{r=1}^{N} x_r(t_r)$

Algorithm (Collection From Left) [Leedham-Green, Soicher, 1989]

$k := \max\{1 \leq \ell \leq m \mid \prod_{i=1}^{\ell} x_{r_i}(t_i) \text{ is collected } \}$

`while` $k < m$ `do`

collect the term $x_{r_{k+1}}(t_{k+1})$ to left

$m := \text{len}(x)$

$k := \max\{1 \leq \ell \leq m \mid \prod_{i=1}^{\ell} x_{r_i}(t_i) \text{ is collected } \}$

`end while`

Rediscovered in [SH, 2000]

Collection From Left

Input: $x = \prod_{i=1}^{m} x_{r_i}(t_i)$ Output: $x = \prod_{r=1}^{N} x_r(t_r)$

Algorithm (Collection From Left) [Leedham-Green, Soicher, 1989]

$k := \max\{1 \leq \ell \leq m \mid \prod_{i=1}^{\ell} x_{r_i}(t_i) \text{ is collected } \}$
`while` $k < m$ `do`
    collect the term $x_{r_{k+1}}(t_{k+1})$ to left
    $m := \text{len}(x)$
    $k := \max\{1 \leq \ell \leq m \mid \prod_{i=1}^{\ell} x_{r_i}(t_i) \text{ is collected } \}$
`end while`

Rediscovered in [SH, 2000]

Normal form
Using additive ordering on roots

- Positive roots ordered with respect to their height.
  I.e., $r < s < r + s$ whenever $r < s$ and $r + s \in \Phi^+$.
- NEW: Additive ordering on roots.
  I.e., $r < r + s < s$ whenever $r < s$ and $r + s \in \Phi^+$.
- Unique decomposition of unipotent elements:

$$\prod_{r \in \Phi^+} x_r(t_r),$$

with roots in the new ordering.

Normal form
Using additive ordering on roots

- Positive roots ordered with respect to their height.
  I.e., $r < s < r + s$ whenever $r < s$ and $r + s \in \Phi^+$.
- NEW: Additive ordering on roots.
  I.e., $r < r + s < s$ whenever $r < s$ and $r + s \in \Phi^+$.
- Unique decomposition of unipotent elements:

$$\prod_{r \in \Phi^+} x_r(t_r),$$

with roots in the new ordering.

Normal form
Using additive ordering on roots

- Positive roots ordered with respect to their height.
  I.e., $r < s < r + s$ whenever $r < s$ and $r + s \in \Phi^+$.
- NEW: Additive ordering on roots.
  I.e., $r < r + s < s$ whenever $r < s$ and $r + s \in \Phi^+$.
- Unique decomposition of unipotent elements:

$$\prod_{r \in \Phi^+} x_r(t_r),$$

with roots in the new ordering.

Collection From Outside

$$\begin{aligned} x_r(t)x_s(u) &= x_s(u)x_r(t) \cdot [x_r(t), x_s(u)] \\ &= x_s(u)x_r(t) \cdot \prod_{i,j>0} x_{ir+js}(C_{rsij}t^i u^j) \end{aligned}$$

Algorithm (Collection From Outside) [Cohen,Haller,Murray, 2006]

- Basic Idea: Run at the same time
  - Collection From Left on the left and
  - Reversed Collection From Left on the right
- Note: Special care needed when both collections "meet" in the middle

Collection From Outside

$$x_r(t)x_s(u) = x_s(u) \cdot [x_s(u), x_r(-t)] \cdot x_r(t)$$

Algorithm (Collection From Outside) [Cohen,Haller,Murray, 2006]

- Basic Idea: Run at the same time
  - Collection From Left on the left and
  - Reversed Collection From Left on the right
- Note: Special care needed when both collections "meet" in the middle

Collection From Outside

$$\begin{aligned} x_r(t)x_s(u) &= x_s(u) \cdot [x_s(u), x_r(-t)] \cdot x_r(t) \\ &= x_s(u) \cdot \left(\prod_{i,j>0} x_{is+jr}(C_{srij} u^i (-t)^j) \right) \cdot x_r(t) \end{aligned}$$

Algorithm (Collection From Outside) [Cohen,Haller,Murray, 2006]

- Basic Idea: Run at the same time
  - Collection From Left on the left and
  - Reversed Collection From Left on the right
- Note: Special care needed when both collections "meet" in the middle

Collection From Outside

$$\begin{aligned} x_r(t)x_s(u) &= x_s(u) \cdot [x_s(u), x_r(-t)] \cdot x_r(t) \\ &= x_s(u) \cdot \left(\prod_{i,j>0} x_{is+jr}(C_{srij} u^i (-t)^j) \right) \cdot x_r(t) \end{aligned}$$

Algorithm (Collection From Outside) [Cohen,Haller,Murray, 2006]

- Basic Idea: Run at the same time
  - Collection From Left on the left and
  - Reversed Collection From Left on the right
- Note: Special care needed when both collections "meet" in the middle

Collection From Outside

$$\begin{aligned} x_r(t)x_s(u) &= x_s(u) \cdot [x_s(u), x_r(-t)] \cdot x_r(t) \\ &= x_s(u) \cdot \left(\prod_{i,j>0} x_{is+jr}(C_{srij} u^i (-t)^j) \right) \cdot x_r(t) \end{aligned}$$

Algorithm (Collection From Outside) [Cohen,Haller,Murray, 2006]

- Basic Idea: Run at the same time
  - Collection From Left on the left and
  - Reversed Collection From Left on the right
- Note: Special care needed when both collections "meet" in the middle

Comparison of Algorithms

- AMD Opteron 150 (2393MHz)

Comparison of Algorithms

- AMD Opteron 150 (2393MHz)

Product of two random (collected) unipotent elements

	CFR	CFL	CFO	
$E_8(17)$		0.060	0.010	
$A_{30}(17)$		0.797	0.085	
$B_{20}(17)$		1.290	0.146	

Comparison of Algorithms

- AMD Opteron 150 (2393MHz)

Product of two random (collected) unipotent elements

	CFR	CFL	CFO	
$E_8(17)$	6.140	0.060	0.010	
$A_{30}(17)$	128.600	0.797	0.085	
$B_{20}(17)$	290.450	1.290	0.146	

Comparison of Algorithms

- AMD Opteron 150 (2393MHz)

Product of two random (collected) unipotent elements

	2.13			2.12
	CFR	CFL	CFO	CFL
$E_8(17)$	6.140	0.060	0.010	15.900
$A_{30}(17)$	128.600	0.797	0.085	155.170
$B_{20}(17)$	290.450	1.290	0.146	2210.100

Comparison of Algorithms

- AMD Opteron 150 (2393MHz)

Product of two random (collected) unipotent elements

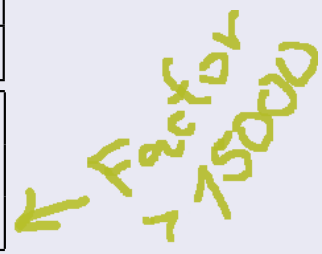
	2.13			2.12
	CFR	CFL	CFO	CFL
$E_8(17)$	6.140	0.060	0.010	15.900
$A_{30}(17)$	128.600	0.797	0.085	155.170
$B_{20}(17)$	290.450	1.290	0.146	2210.100

Comparison of Algorithms

- AMD Opteron 150 (2393MHz)

Product of two random (collected) unipotent elements

	2.13			2.12
	CFR	CFL	CFO	CFL
$E_8(17)$	6.140	0.060	0.010	15.900
$A_{30}(17)$	128.600	0.797	0.085	155.170
$B_{20}(17)$	290.450	1.290	0.146	2210.100



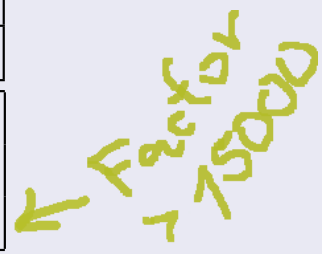
Abstract degrees of Hall Polynomials

Comparison of Algorithms

- AMD Opteron 150 (2393MHz)

Product of two random (collected) unipotent elements

	2.13			2.12
	CFR	CFL	CFO	CFL
$E_8(17)$	6.140	0.060	0.010	15.900
$A_{30}(17)$	128.600	0.797	0.085	155.170
$B_{20}(17)$	290.450	1.290	0.146	2210.100



Abstract degrees of Hall Polynomials

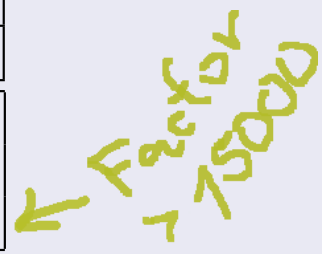
	CFL		CFO	
	max	avg	max	avg
A_n	n	$(n+2)/3$		
B_n	$2n-1$	$\sim (2n)/3$		
C_n	$2n-1$	$\sim (2n)/3$		
D_n	$2n-3$	$(2n-1)/3$		

Comparison of Algorithms

- AMD Opteron 150 (2393MHz)

Product of two random (collected) unipotent elements

	2.13			2.12
	CFR	CFL	CFO	CFL
$E_8(17)$	6.140	0.060	0.010	15.900
$A_{30}(17)$	128.600	0.797	0.085	155.170
$B_{20}(17)$	290.450	1.290	0.146	2210.100



Abstract degrees of Hall Polynomials

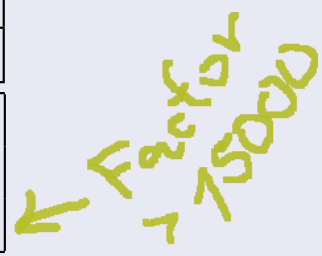
	CFL		CFO	
	max	avg	max	avg
A_n	n	$(n+2)/3$	2	$\to 2$
B_n	$2n-1$	$\sim (2n)/3$	4	$\to 4$
C_n	$2n-1$	$\sim (2n)/3$	3	$\to 3$
D_n	$2n-3$	$(2n-1)/3$	3	$\to 3$

Comparison of Algorithms

- AMD Opteron 150 (2393MHz)

Product of two random (collected) unipotent elements

	2.13			2.12
	CFR	CFL	CFO	CFL
$E_8(17)$	6.140	0.060	0.010	15.900
$A_{30}(17)$	128.600	0.797	0.085	155.170
$B_{20}(17)$	290.450	1.290	0.146	2210.100



Abstract degrees of Hall Polynomials

	CFL		CFO	
	max	avg	max	avg
A_n	n	$(n+2)/3$	2	$\to 2$
B_n	$2n-1$	$\sim (2n)/3$	4	$\to 4$
C_n	$2n-1$	$\sim (2n)/3$	3	$\to 3$
D_n	$2n-3$	$(2n-1)/3$	3	$\to 3$

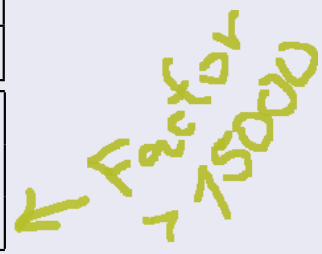
In Type B_{15}, we have $|\Phi^+| = 225$ Hall Polynomials

Comparison of Algorithms

- AMD Opteron 150 (2393MHz)

Product of two random (collected) unipotent elements

	2.13			2.12
	CFR	CFL	CFO	CFL
$E_8(17)$	6.140	0.060	0.010	15.900
$A_{30}(17)$	128.600	0.797	0.085	155.170
$B_{20}(17)$	290.450	1.290	0.146	2210.100



Abstract degrees of Hall Polynomials

	CFL		CFO	
	max	avg	max	avg
A_n	n	$(n+2)/3$	2	$\to 2$
B_n	$2n-1$	$\sim (2n)/3$	4	$\to 4$
C_n	$2n-1$	$\sim (2n)/3$	3	$\to 3$
D_n	$2n-3$	$(2n-1)/3$	3	$\to 3$

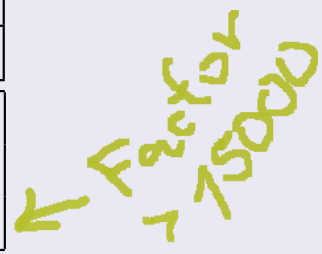
In Type B_{15}, we have $|\Phi^+| = 225$ Hall Polynomials
Largest (printed as string) is 9226 B with CFO

Comparison of Algorithms

- AMD Opteron 150 (2393MHz)

Product of two random (collected) unipotent elements

	2.13			2.12
	CFR	CFL	CFO	CFL
$E_8(17)$	6.140	0.060	0.010	15.900
$A_{30}(17)$	128.600	0.797	0.085	155.170
$B_{20}(17)$	290.450	1.290	0.146	2210.100



Abstract degrees of Hall Polynomials

	CFL		CFO	
	max	avg	max	avg
A_n	n	$(n+2)/3$	2	$\to 2$
B_n	$2n-1$	$\sim (2n)/3$	4	$\to 4$
C_n	$2n-1$	$\sim (2n)/3$	3	$\to 3$
D_n	$2n-3$	$(2n-1)/3$	3	$\to 3$

In Type B_{15}, we have $|\Phi^+| = 225$ Hall Polynomials
Largest (printed as string) is 9226 B with CFO and $\sim$ 297 MB with CFL.

Outline

Examples I

The following is a captured Magma session presented interactively at the conferecne. It demonstrates some of the new features available in Magma 2.13

```
Got mach: Intel(R) Pentium(R) M processor 2.00GHz
Magma V2.13-2     Tue Aug  1 2006 11:50:08 on 2go      [Seed = 15191]
Type ? for help.  Type <Ctrl>-D to quit.

Loading startup file "/home/sergei/.magmarc"

> iload demo;
Interactive-loading "demo"
>
> R := RootDatum("B2"); R;
R: Adjoint root datum of dimension 2 of type B2
> DynkinDiagram(R);

B2    1 =>= 2
> PositiveRoots(R);
{@
     (1 0),
     (0 1),
     (1 1),
     (1 2)
@}
>
```

Examples II

```
> R := RootDatum("BC2");
> DynkinDiagram(R);

BC2    1 =>= 2
> PositiveRoots(R);
{@
     (1 0),
     (0 1),
     (1 1),
     (0 2),
     (1 2),
     (2 2)
@}
>
> R := RootDatum( "A2" : Twist:=2 ); R;
R: Twisted adjoint root datum of dimension 2 of type 2A2,1
> RelativeRootDatum(R);
Adjoint root datum of dimension 1 of type BC1
> OrbitsPi(R);
[
     GSet{ 1, 2 }
]
> DistinguishedOrbitsPi(R);
[
     GSet{ 1, 2 }
]
>
```

Examples III

```
> R := RootDatum( "E6" : Twist:=<{{2}},2> ); R;
R: Twisted adjoint root datum of dimension 6 of type 2E6,1
> RelativeRootDatum(R);
Adjoint root datum of dimension 1 of type BC1
> OrbitsPi(R);
[
    GSet{ 2 },
    GSet{ 4 },
    GSet{ 1, 6 },
    GSet{ 3, 5 }
]
> DistinguishedOrbitsPi(R);
[
    GSet{ 2 }
]
>
> R := RootDatum( "D4" : Twist:=<{{2}},6> ); R;
R: Twisted adjoint root datum of dimension 4 of type 6D4,1
> RelativeRootDatum(R);
Adjoint root datum of dimension 1 of type BC1
>
```

Examples IV

```
> OrbitsPi(R);
[
    GSet{ 2 },
    GSet{ 1, 3, 4 }
]
> DistinguishedOrbitsPi(R);
[
    GSet{ 2 }
]
>
> R := RootDatum( "A3" : Twist:=2 ); R;
R: Twisted adjoint root datum of dimension 3 of type 2A3,2
> RelativeRootDatum(R);
Adjoint root datum of dimension 2 of type B2
> OrbitsPi(R);
[
    GSet{ 2 },
    GSet{ 1, 3 }
]
> DistinguishedOrbitsPi(R);
[
    GSet{ 2 },
    GSet{ 1, 3 }
]
>
```

Examples V

```
> R;
R: Twisted adjoint root datum of dimension 3 of type 2A3,2
> k := GF(5);
> K := GF(5^2);
> G := TwistedGroupOfLieType(R,k,K); G;
G: Twisted group of Lie type 2A3,2 over GF(5) with entries over GF(5^2)
>
> RelativeRootElement(G,1,[Random(K)]);
x2(2)
> RelativeRootElement(G,2,[Random(K)]);
x1($.1^21) x3($.1^9)
>
> quit;

Total time: 1.750 seconds, Total memory usage: 24.95MB
```

Outline

References I


Bosma, W., Cannon, J. and Playoust, C.
The Magma algebra system. I. The user language
J. Symbolic Computation, **24**(3-4):235–265 (1997)
http://magma.maths.usyd.edu.au/


Cohen, A.M., Murray, S.H. and Taylor, D.E.
Computing in groups of Lie type
Mathematics of Computation, **73**:1477–1498 (2004)


Cohen, A.M., Haller, S. and Murray, S.H.
Computing in unipotent groups
Preprint (2006)

References II

Neubüser, J.
Bestimmung der Untergruppenverbände endlicher *p*-Gruppen auf einer Programmgesteuerten elektronischen Dualmaschine
Numer. Math, **3**:271–278 (1987)

Hall, P
A contribution to the theory of groups of prime-power order.
Proc. London. Math. Soc., **36**:29–95 (1934)

Leedham-Green, C.R. and Soicher, L.H.
Collection from theLeft and Other Strategies
J. Symbolic Computation, **9**:665-675 (1990)

References III

Sims, C.C.
Verifying nilpotence
J. Symbolic Computation, **3**:231–247 (1987)