CoxIter

Computations of invariants of hyperbolic Coxeter groups

1.0b

17/08/2016

Rafael Guglielmetti

Rafael Guglielmetti
Email: rafael@rgug.ch
Homepage: https://rgug.ch
Address: Chemin du Musee, CH-1700 Fribourg

Contents

Chapter 1

Introduction

This chapter gives general information about the GAP package CoxIter.

1.1 The standalone program

CoxIter was first developped as a standalone C++ program whose goal is to compute invariant of hyperbolic Coxeter groups. We consider a hyperbolic Coxeter group $\Gamma \leq \mathrm{Isom}(\mathbb{H}^n)$ and the associated polyhedron P. The input of the program consists a file containing the description of the Coxeter graph of Γ. Then, the output consists of the following information:

- Euler characteristic (and thus volume of P if n is even)
- f-vector $(f_0, f_1, \ldots, f_n)$: P has f_0 vertices, f_1 edges, f_2 2-faces, ...
- Cofiniteness test: test whether P has finite volume or not
- Cocompactness test: test whether P is compact or not
- Growh series
- Growth rate and some of its algebraic properties (note: this is not available in the GAP package)

A description of the mathematical results behind CoxIter can be found in the article [Gug15].

1.2 The GAP package

1.2.1 Encoding a group

Let $\Gamma = \langle s_1, \ldots, s_d : (s_i \cdot s_j)^{m_{i,j}} \rangle$ be a Coxeter group and let $P = \bigcap_{i=1}^{d} H_i^-$ be its associated polyhedron. There is two ways to encode Γ in for the package:

- Via its Coxeter graph: `CreateCoxIterFromCoxeterGraph` (**??**)
  The graph can be described with a list containing the neighbours of every vertex, together with the weights. CoxIter uses that following convention:
  - If the hyperplanes H_i and H_j are parallel, then the weight is 0
  - If the hyperplanes H_i and H_j are ultra-parallel, then the weight is 1

For example, the graph of the Coxeter group $\tilde{B}_5$ can be described as follows: `[[1,[2,4]],[2,[3,3]],[3,[4,3]],[4,[5,3],[6,3]]]`

- Via its Coxeter matrix: `CreateCoxIterFromCoxeterMatrix` (**??**)

  The Coxeter matrix M is the symmetric $d \times d$ matrix with entries $m_{i,j}$. Again, we use the following convention:

  - If the hyperplanes H_i and H_j are parallel, then $M_{i,j} = 0$
  - If the hyperplanes H_i and H_j are ultra-parallel, then $M_{i,j} = 1$

1.2.2 A complete example

We consider the 4-dimensional simplex $[4,3,3,5]$ whose Coxeter graph is the linear graph with weights 4,3,3,5.
First, we cerate the CoxIter object:

Example
```
gap> ci := CreateCoxIterFromCoxeterGraph([[1,[2,4]],[2,[3,3]],[3,[4,3]],
        [4,[5,5]]],4);
CoxIter : Coxeter group with 5 generators in dimension 4
```

We can now ask for one of the invariants:

Example
```
gap> EulerCharacteristic(ci);
17/28800
gap> Cocompact(ci);
1
gap> Cofinite(ci);
1
gap> FVector(ci);
[ 5, 10, 10, 5, 1 ]
gap> g := GrowthSeries(ci);;
gap> Value(g[2],1)/Value(g[1],1);
17/28800
```

Chapter 2

CoxIter automatic generated documentation

2.1 CoxIter automatic generated documentation of attributes

2.1.1 Cofinite (for IsCoxIter)

▷ `Cofinite(`*`CoxIter, object`*`)` (attribute)

Returns: 1 (cofinite), 0 (not cofinite), -1 (cannot decide)

Test whether the group is cofinite or not.

2.1.2 Cocompact (for IsCoxIter)

▷ `Cocompact(`*`CoxIter, object`*`)` (attribute)

Returns: 1 (cocompact), 0 (not cocompact), -1 (cannot decide)

Test whether the group is cocompact or not.

2.1.3 EulerCharacteristic (for IsCoxIter)

▷ `EulerCharacteristic(`*`CoxIter, object`*`)` (attribute)

Returns: the Euler characteristic

Compute the Euler characteristic

2.1.4 FVector (for IsCoxIter)

▷ `FVector(`*`CoxIter, object`*`)` (attribute)

Returns: the f-vector of the associated polyhedron

Compute the f-vector of the associated polyhedron

2.1.5 GrowthSeries (for IsCoxIter)

▷ `GrowthSeries(`*`CoxIter, object`*`)` (attribute)

Returns: [f,g] where f/g is the rational expansion of the growth series

Compute the rational expansion of the growth series

2.2 CoxIter automatic generated documentation of methods

2.2.1 CreateCoxIterFromCoxeterGraph (CreateCoxIterFromCoxeterGraph)

▷ `CreateCoxIterFromCoxeterGraph(`*`gr, dimension`*`)` (operation)

Returns: a CoxIter object

Creates a CoxIter object from the Coxeter graph *`gr`*. If the dimension *`dim`* is unknown, 0 can be given.

2.2.2 CreateCoxIterFromCoxeterMatrix (for IsMatrix, IsInt)

▷ `CreateCoxIterFromCoxeterMatrix(`*`mat, dimension`*`)` (operation)

Returns: a CoxIter object

Creates a CoxIter object from the Coxeter matrix *`mat`*. If the dimension *`dim`* is unknown, 0 can be given.

2.2.3 CoxIterCompute (for IsCoxIter)

▷ `CoxIterCompute(`*`ci`*`)` (operation)

Returns:

Compute the invariants of the Coxiter object *`ci`*

Chapter 3

Some examples

First, we consider the 8 dimensional cocompact group found by Bugaenko.

Example

```
gap> LoadPackage( "CoxIter" );
gap> buga8 := CreateCoxIterFromCoxeterGraph([[1,[2,5]],[2,[3,3]],[3,[4,3]],[4,
> [5,3],[10,3]],[5,[6,3]],[6,[7,3],[11,3]],[7,[8,3]],[8,[9,5]],[10,[11,1]]],8);
CoxIter : Coxeter group with 11 generators in dimension 8
gap> Cofinite(buga8);
1
gap> Cocompact(buga8);
1
gap> FVector(buga8);
[ 41, 164, 316, 374, 294, 156, 54, 11, 1 ]
gap> EulerCharacteristic(buga8);
24187/8709120000
gap> g := GrowthSeries(cibugaenko8);;
gap> Value(g[2],1)/Value(g[1],1) - EulerCharacteristic(cibugaenko8);
0
```

References

[Gug15] Rafael Guglielmetti. CoxIter - Computing invariants of hyperbolic Coxeter groups. *LMS Journal of Computation and Mathematics*, 18(1):754–773, December 2015. 3

Index