

PyCox: computing with (finite) Coxeter groups and Iwahori–Hecke algebras

Meinolf Geck

Dedicated to the memory of Professor H. Pahlings

ABSTRACT

We introduce the computer algebra package PyCox, written entirely in the Python language. It implements a set of algorithms, in a spirit similar to the older CHEVIE system, for working with Coxeter groups and Hecke algebras. This includes a new variation of the traditional algorithm for computing Kazhdan–Lusztig cells and W -graphs, which works efficiently for all finite groups of rank ≤ 8 (except E_8). We also discuss the computation of Lusztig’s leading coefficients of character values and distinguished involutions (which works for E_8 as well). Our experiments suggest a re-definition of Lusztig’s ‘special’ representations which, conjecturally, should also apply to the unequal parameter case.

Supplementary materials are available with this article.

1. Introduction

The computer algebra system CHEVIE [13] has been designed to facilitate computations with various combinatorial structures arising in Lie theory, like finite Coxeter groups and Hecke algebras. It was initiated about 20 years ago and has been further developed ever since; see [11] for a discussion of some recent applications of this system. However, there are some limitations to its use due to its dependence on GAP3 [30] which is still available, but no longer supported (the last release of GAP3 was in 1997). Therefore, it seemed desirable to implement a core set of algorithms around Coxeter groups and Hecke algebras in a more modern and widely available environment. The success of Sage [31] suggested the use of the Python language [29]. This led to the development of PyCox, which we present in this note.

Although in some areas of algebraic manipulations (like permutations, algebraic numbers) the performance is inferior to that of GAP3, some of the advantages of the new system are: it works on every computer where Python is installed; we can now run jobs which use main memory well over 4 GB; and we can just import PyCox as a module into Sage, whereby we have immediate access to all the further functionality of Sage (including the Sage notebook and the interfaces to GAP4 and, even, GAP3).

In Section 2, we briefly describe the basic design features of PyCox and give some examples of how to use it; more details are available through the online help within PyCox.

We shall then discuss some concrete applications of our programs to questions related to the theory of Kazhdan–Lusztig cells and the associated W -graphs. (The basic definitions will be recalled in Section 3.) The problem of computing such cells has been addressed by several authors, most notably by Alvis [1] and DuCloux [4–6]. In Section 4, we present a variation of the known algorithm where the new ingredient is the use of ‘relative’ Kazhdan–Lusztig polynomials, as defined in [7, 16]. (Neither CHEVIE nor DuCloux’s Coxeter [5] contains an implementation of these.) As may be expected, the systematic use of these relative polynomials instead of the ordinary Kazhdan–Lusztig polynomials leads to a significant efficiency

Received 3 February 2012.

2000 Mathematics Subject Classification 20C40 (primary), 20C08, 20F55 (secondary).

gain for the determination of left cells. (This idea was essentially already formulated by DuCloux [4, Section 5.3]; Howlett and Yin [18] constructed W -graphs affording irreducible representations in this way.) While Howlett and Yin [16] only considered the equal parameter case, we shall describe recursion formulae in the general case of unequal parameters.

Within PyCox, it is now possible to deal with left cells and the corresponding W -graphs for finite Coxeter groups of rank up to around 8, including types H_4 , E_6 and E_7 but, not surprisingly, with the exception of type E_8 . There are also programs for computing Lusztig's leading coefficients of character values [26] and distinguished involutions [25]; these do work for type E_8 as well. As far as we are aware, this is the first general program which is capable of dealing with this level of information for a group of size like that of type E_7 or E_8 . As an example of an application we just mention that it is now straightforward to verify that Kottwitz's conjecture [20] on the intersections of left cells with conjugacy classes of involutions, holds for type E_7 , following the general methods explained by Casselman [2]. (Abbie Halls, at the University of Aberdeen, is currently working on type E_8 , where more specialised methods and programming are required.) We also stress the fact that our programs only use results concerning cells which are generally known to hold by elementary arguments; in particular, we do not rely on any 'positivity' results or facts concerning Lusztig's \mathbf{a} -function [24].

Finally, in Section 5, we consider the problem of computing the character tables of certain symmetric algebras which have been associated by Lusztig [26] with the various left cells of a finite Coxeter group. In PyCox, these tables can be determined by an automatic procedure for all groups of rank up to around 7. In Lusztig's work [26], the non-crystallographic types $I_2(m)$, H_3 and H_4 have been excluded from the discussion. Here, we complete the picture by treating these cases as well. This will allow us to formulate in full generality an alternative characterisation of 'special' representations, which were originally defined by Lusztig [21]. This characterisation should also make sense in the general case of unequal parameters; see Conjecture 5.11. We discuss some examples to support this conjecture.

2. Design of PyCox

The whole PyCox system is contained in one file accompanying this article; it is freely available for download, under the GPL licence. (The file is called `chv1r61.py`, it has 13441 lines and the size is roughly 525 KB; updates will also be made available at the author's homepage.) So, in order to use PyCox on your computer, all you need to do (once you have downloaded the file) is to launch Python (2.6 or higher) and import the file as a module, for example, by typing the following.

```
>>> from chv1r61 import *
```

(You should then see a welcome banner.) Similarly, if Sage (version 4.7 or higher) is installed on your computer, you can import PyCox as a module into Sage.

```
sage: from chv1r61 import *
```

A good place to start is to type `'help(coxeter)'` or `'allfunctions()'`.

We shall now discuss some of the basic principles of the system and show some concrete examples. As in CHEVIE, the basic object from which everything is built up is that of a Cartan matrix. Let S be a finite non-empty index set and $C = (c_{st})_{s,t \in S}$ be a matrix with entries in \mathbb{R} . Following [15, Section 1.1], we say that C is a *Cartan matrix* if the following conditions are satisfied.

(C1) For $s \neq t$ we have $c_{st} \leq 0$; furthermore, $c_{st} \neq 0$ if and only if $c_{ts} \neq 0$.

(C2) We have $c_{ss} = 2$ and, for $s \neq t$, we have $c_{st}c_{ts} = 4 \cos^2(\pi/m_{st})$, where $m_{st} \geq 2$ is an integer or $m_{st} = \infty$.

Let $C = (c_{st})_{s,t \in S}$ be a Cartan matrix and V be an \mathbb{R} -vector space of dimension $|S|$, with a fixed basis $\{\alpha_s \mid s \in S\}$. For each $s \in S$, we define a linear map on V as follows:

$$\dot{s}: V \rightarrow V, \quad \alpha_t \mapsto \alpha_t - c_{st}\alpha_s \quad (t \in S).$$

We shall act from the right so we use the row convention for matrices of linear maps. Let

$$W := \langle \dot{s} \mid s \in S \rangle \subseteq \text{GL}(V).$$

In what follows, we shall often omit the dot when referring to the maps $\dot{s}: V \rightarrow V$; in particular, S will be regarded directly as a subset of $\text{GL}(V)$. With this convention, the group W has a presentation with generators S and defining relations as follows (see [15, Section 1.2.7]):

$$s^2 = 1 \quad \text{for all } s \in S \quad \text{and} \quad (st)^{m_{st}} = 1 \quad \text{for all } s \neq t \text{ in } S \text{ with } m_{st} < \infty.$$

Thus, W is a Coxeter group and all Coxeter groups arise in this way. The matrix $M = (m_{st})_{s,t \in S}$ is called the *Coxeter matrix* of W .

The set $\Phi := \{\alpha_s.w \mid s \in S, w \in W\}$ is the corresponding root system. There is a well-defined partition $\Phi = \Phi^+ \amalg \Phi^-$ where Φ^+ is the set of all $\alpha \in \Phi$ which can be expressed in terms of the basis $\{\alpha_s \mid s \in S\}$ where all coefficients are non-negative, and $\Phi^- = \{-\alpha \mid \alpha \in \Phi\}$; see [15, Section 1.1.9]. Based on this information alone, we already have an efficient way of testing if an element $w \in W$ (given as a word in the generators in S) equals the identity or not: it suffices to compute the corresponding linear map of V and check if its matrix is the identity or not. More generally, if $w \neq 1$, we can efficiently find an $s \in S$ such that $l(sw) < l(w)$ (see [15, Section 1.1.9]):

$$l(sw) = l(w) - 1 \quad \text{if and only if} \quad \alpha_s.w \in \Phi^-.$$

(Here, $l(w)$ denotes the usual length of $w \in W$.)

Following the general ideas in CHEVIE, the basic function in PyCox is that of creating a Coxeter group from a Cartan matrix (as a Python ‘class’).

```
>>> W=coxeter([[2, -1, -1], [-1, 2, -1], [-1, -1, 2]])
```

When the function `coxeter` is called, it computes some basic information directly from the Cartan matrix, for example, the matrices of the generating reflections and the Coxeter matrix. These pieces of information are saved as components in the resulting Python class.

```
>>> W.coxetermat
[[1, 3, 3], [3, 1, 3], [3, 3, 1]]
>>> W.cartantype
[['U', [0, 1, 2]]]          # 'U' stands for infinite
>>> W.matgens              # matrices of the generators
[((-1, 0, 0), (1, 1, 0), (1, 0, 1)),
 ((1, 1, 0), (0, -1, 0), (0, 1, 1)),
 ((1, 0, 1), (0, 1, 1), (0, 0, -1))]
```

We can now do some basic operations with the elements of W .

```
>>> m=W.wordtomat([1, 0, 1, 2, 1, 0]); m
((-1, 0, -1), (-2, -2, -1), (4, 3, 3))
>>> W.mattoword(m)        # lexicographically smallest
[0, 1, 0, 2, 1, 0]       # reduced expression of m
>>> W.leftdescentsetmat(m)
[0, 1]
>>> W.rightdescentsetmat(m)
[0]
```

When W is finite, the function `coxeter` also computes, for example, the number of positive roots and the order of W .

```
>>> A=[[ 2, 0, -1, -ir5, 0], # ir5=(1+sqrt(5))/2;
        [ 0, 2, 0, 0, -1], # see the function
        [-1, 0, 2, 0, 0], # zeta5
        [-ir5, 0, 0, 2, 0],
        [ 0, -3, 0, 0, 2]]
>>> W=coxeter(A)
>>> W.N; W.order; W.degrees
21 # number of positive roots
1440 # order of the group
[2, 2, 6, 6, 10] # reflection degrees
```

There are further components, like `W.roots` (containing a list of all roots), `W.permgens` (the permutation action of the generators on the roots) etc.; the list of all components is obtained by `dir(W)` (a general Python function). In order to check if W is finite, `coxeter` uses the known list of Coxeter matrices of irreducible finite Coxeter groups, encoded in terms of graphs (with a standard labelling of the vertices) as in Table 1.

Note that, in general, there may be several Cartan matrices which give rise to the same Coxeter matrix. In PyCox (as in CHEVIE) we have adopted the following conventions.

- If m_{st} is odd, then $c_{st} = c_{ts}$. (This has the consequence that the root system is reduced; see [15, Section 1.3.6].)
- If m_{st} is even, then $c_{st} = -1$ or $c_{ts} = -1$.

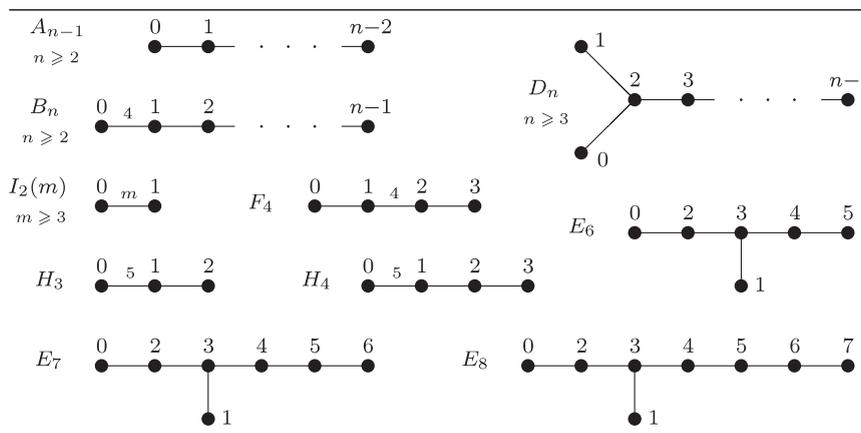
For example, the following two Cartan matrices both correspond to the Coxeter matrix of type B_3 .

```
>>> cartanmat("B",3)
[[2, -2, 0], [-1, 2, -1], [0, -1, 2]]
>>> cartanmat("C",3)
[[2, -1, 0], [-2, 2, -1], [0, -1, 2]]
```

See the help for the function `cartanmat` for a detailed description of the resulting choices of Cartan matrices for the various finite types.

When `coxeter` is called, it decomposes the Cartan matrix into its indecomposable components and checks if the corresponding Coxeter graphs appear in the list in Table 1. If this is so, it matches the Cartan matrices of the indecomposable components to those returned by the function `cartanmat`. This information is kept in the component `W.cartantype`. In the

TABLE 1. Coxeter graphs of irreducible finite Coxeter groups.



above example (where W is defined by a 5×5 Cartan matrix), we have the following.

```
>>> W.cartantype
[['H', [3, 0, 2]], ['G', [1, 4]]]
```

This means that the submatrix of A with rows and columns indexed by 3, 0, 2 (in this order) is the standard Cartan matrix of type H_3 , as returned by calling `cartanmat('H',3)`; similarly, the submatrix of A with rows and columns indexed by 1, 4 (in this order) is the standard Cartan matrix of type G_2 , as returned by calling `cartanmat('G',2)`. In particular, we see that our group W is of type $H_3 \times G_2$.

The type recognition procedure is particularly helpful when dealing with reflection subgroups.

```
>>> W=coxeter("F",4) # same as coxeter(cartanmat("F",4))
>>> H=reflectionsubgroup(W,[1,2,6,47])
>>> H.cartantype # subgroup generated
[['C', [0, 1, 2]], ['A', [3]]] # by reflections at
>>> H.cartan # roots no. 1,2,6,47
[[2,-1, 0, 0], [-2, 2,-1, 0], [0,-1, 2, 0], [0, 0, 0, 2]]
```

Here, H will be a Coxeter group in its own right. The information about the embedding into W is held in the component `H.fusions`; every Coxeter group in PyCox has such a `fusion` component: it will at least contain the embedding into itself; see the online help of `reflectionsubgroup` for further details. In the above example, we have the following.

```
>>> W.cartanname # unique string identifying W
'F4c0c1c2c3'
>>> H.fusions['F4c0c1c2c3']
{'parabolic': False, 'subJ': [1, 2, 3, 23]}
```

Thus, H is not a parabolic subgroup and the four simple reflections of H correspond to the reflections with roots indexed by 1, 2, 3, 23 in `W.roots`. (This design is different from that in CHEVIE; it appears to be better suited to recursive algorithms involving various reflection subgroups.)

Let us assume from now on that W is finite. Then, in principle, every piece of information about W is ultimately computable from the Cartan matrix of W . However, as in CHEVIE, some very basic and frequently used pieces of information are explicitly stored within the system; this is particularly relevant for data which are accompanied by some more or less natural labellings (like partitions of n for the conjugacy classes and irreducible characters of groups of type A_{n-1}). In PyCox, we store explicitly the following pieces of information (with the appropriate labellings where this applies):

- reflection degrees (see the function `degreesdata`);
- conjugacy classes (see `conjclassdata`);
- character tables (see `irrchardata` and `heckeirrddata`);
- Schur elements (see `schurelmdata`).

For classical types A_n , B_n , D_n , this is done in the form of combinatorial algorithms; for the remaining exceptional types, explicit tables with the relevant information are stored. Then, for example, when the function `chartable(W)` is called, PyCox will build the character table of W from the explicitly stored data for the irreducible components of W . (Note that W is a direct product of its irreducible components, and there is a standard procedure to build the character table of a direct product of finite groups from the character tables of the direct factors.)

Let us now give a concrete example of how to use these programs. We would like to program a function which returns the list of involutions in W , that is, all the elements $w \in W$ such that $w^2 = 1$. To start somewhere, we have a look at the list of all available functions in PyCox; this is printed by calling `allfunctions()`. There is a function `allelmsproperty` which takes as input a group W and a function $f: W \rightarrow \{\text{True}, \text{False}\}$; it returns the list of all $w \in W$ (as reduced words) such that $f(w) = \text{True}$. This certainly fits our problem: we just need to

define f such that $f(w) = \text{True}$ if w has order 1 or 2, and $f(w) = \text{False}$ otherwise. Thus, our first candidate for the desired function is as follows.

```
>>> def involutions1(W):
...     return allelmsproperty(W, lambda x:W.permorder(W.wordtoperm(x))<=2)
>>> len(involutions1(coxeter("E",6)))
892
```

This works fine for groups of moderate size but, eventually, we would also like to apply this to big examples like groups of type E_7 and E_8 ; however, when we do this, we notice that a long time will pass before we see a result. This is because `allelmsproperty` is an ‘all-purpose’ function which runs through all elements of W , transforms every element into a permutation, and checks if this has order 1 or 2. For type E_8 with its 696,729,600 elements this will simply take too long. We need to tailor our program more specifically to the problem that we are dealing with. Now, the set of involutions is invariant under conjugation so it will be a union of the conjugacy classes of W . The function `conjugacyclasses` does return some information about the conjugacy classes of W , including representatives of the classes and the sizes of the classes. (This uses data stored within the system; see `conjclassdata`.) So, alternatively to our first try above, we could just select the class representatives which are involutions and then compute the corresponding conjugacy classes. In Python, this can in fact be done in one line.

```
>>> def involutions(W):
...     return flatlist([conjugacyclass(W, W.wordtoperm(w))
...                     for w in conjugacyclasses(W) ['reps']
...                     if W.wordtocoxelm(2*w)==tuple(W.rank)])
```

This even works in type E_8 where it returns the list of 199952 involutions in about 1 minute. (In GAP3, the analogous function would be roughly twice as fast, thanks to the much more efficient arithmetic for permutations.) It is known that involutions play a special role in the theory of Kazhdan–Lusztig cells; see Kottwitz [20], Lusztig [25] and Lusztig–Vogan [28]. We shall come back to this in Section 5.

3. Cells and W -graphs

Let W be a Coxeter group, with generating set S . In this section, we briefly recall some basic definitions concerning left cells and the corresponding W -graphs, as introduced by Kazhdan and Lusztig [19, 22]. Roughly speaking, these concepts give rise to a partition

$$W = \mathfrak{C}_1 \amalg \mathfrak{C}_2 \amalg \cdots \amalg \mathfrak{C}_r$$

and, for each piece \mathfrak{C}_i in this partition, a W -module $[\mathfrak{C}_i]_1$ with a standard basis $\{b_x \mid x \in \mathfrak{C}_i\}$ where the action of a generator $s \in S$ is described by formulae of a particularly simple form (encoded in a ‘ W -graph’, see Definition 3.3 and Remark 3.5 below). To give more precise definitions, we need to fix some notation. We shall work in the general multi-parameter framework of Lusztig [22, 27], which introduces a weight function into the picture on which all the subsequent constructions depend.

Let Γ be an abelian group (written additively). Let $\{p_s \mid s \in S\} \subseteq \Gamma$ be a collection of elements such that $p_s = p_t$ whenever $s, t \in S$ are conjugate in W . This gives rise to a weight function

$$L: W \rightarrow \Gamma$$

in the sense of Lusztig [27]; we have $L(w) = p_{s_1} + \cdots + p_{s_k}$ where $w = s_1 \dots s_k$ ($s_i \in S$) is a reduced expression for $w \in W$. We shall assume that Γ admits a total ordering \leq which is compatible with the group structure, that is, whenever $g, g' \in \Gamma$ are such that $g \leq g'$, we have $g + h \leq g' + h$ for all $h \in \Gamma$. We assume throughout that

$$L(s) \geq 0 \quad \text{for all } s \in S.$$

(The original ‘equal parameter’ setting of [19] corresponds to the case where $\Gamma = \mathbb{Z}$ with its natural ordering and $p_s = 1$ for all $s \in S$.)

Furthermore, let $R \subseteq \mathbb{C}$ be a subring and $A = R[\Gamma]$ be the free R -module with basis $\{\varepsilon^g \mid g \in \Gamma\}$. (The basic constructions in this section are independent of the choice of R and so we could just take $R = \mathbb{Z}$ here; the flexibility of choosing R will be useful once we consider representations of W .) There is a well-defined ring structure on A such that $\varepsilon^g \varepsilon^{g'} = \varepsilon^{g+g'}$ for all $g, g' \in \Gamma$. We write $1 = \varepsilon^0 \in A$. Let \mathcal{H} be the generic Iwahori–Hecke algebra corresponding to (W, S) , with parameters $\{\varepsilon^{L(s)} \mid s \in S\}$. Thus, \mathcal{H} has an A -basis $\{\tilde{T}_w \mid w \in W\}$ and the multiplication is given by the rule

$$\tilde{T}_s \tilde{T}_w = \begin{cases} \tilde{T}_{sw} & \text{if } l(sw) > l(w), \\ \tilde{T}_{sw} + (\varepsilon^{L(s)} - \varepsilon^{-L(s)})\tilde{T}_w & \text{if } l(sw) < l(w); \end{cases}$$

here, $l: W \rightarrow \mathbb{N}_0$ denotes the usual length function on W with respect to S .

Let $\Gamma_{\geq 0} := \{g \in \Gamma \mid g \geq 0\}$ and denote by $A_{\geq 0}$ (or $R[\Gamma]_{\geq 0}$) the set of all R -linear combinations of terms ε^g where $g \geq 0$. The notations $A_{>0}$, $A_{\leq 0}$, $A_{<0}$ (or $R[\Gamma_{>0}]$, $R[\Gamma_{\leq 0}]$, $R[\Gamma_{<0}]$) have a similar meaning.

Let $a \mapsto \bar{a}$ be the R -linear involution of $A[\Gamma]$ which takes g to g^{-1} for any $g \in \Gamma$. This extends to a ring involution $\mathcal{H} \rightarrow \mathcal{H}$, $h \mapsto \bar{h}$, where

$$\overline{\sum_{w \in W} a_w \tilde{T}_w} = \sum_{w \in W} \bar{a}_w \tilde{T}_{w^{-1}} \quad (a_w \in A \text{ for all } w \in W).$$

We then have a corresponding Kazhdan–Lusztig basis of \mathcal{H} , which we denote by $\{C'_w \mid w \in W\}$ (as in [22]). The basis element C'_w is uniquely determined by the conditions that

$$\bar{C}'_w = C'_w \quad \text{and} \quad C'_w = \sum_{y \in W} P_{y,w}^* \tilde{T}_y$$

where $P_{w,w}^* = 1$ and $P_{y,w}^* \in \mathbb{Z}[\Gamma_{<0}]$ if $y \neq w$; furthermore, we have $P_{y,w}^* = 0$ unless $y \leq w$, where \leq denotes the Bruhat–Chevalley order on W . For $w \in W$ and $s \in S$, we have

$$\tilde{T}_s C'_w = \begin{cases} C'_{sw} & \text{if } L(s) = 0, \\ \varepsilon^{L(s)} C'_w & \text{if } L(s) > 0, sw < w, \\ C'_{sw} - \varepsilon^{-L(s)} C'_w + \sum_{\substack{y \in W \\ sy < y < w}} M_{y,w}^s C'_y & \text{if } L(s) > 0, sw > w, \end{cases}$$

where $M_{y,w}^s$ are certain elements of $\mathbb{Z}[\Gamma]$ such that $\bar{M}_{y,w}^s = M_{y,w}^s$. As explained in [22, Section 3], these elements are determined by the inductive condition

$$M_{y,w}^s - \varepsilon^{L(s)} P_{y,w}^* + \sum_{\substack{z \in W \\ sz < z, y < z < w}} P_{y,z}^* M_{z,w}^s \in \mathbb{Z}[\Gamma_{<0}] \tag{M1}$$

and by the symmetry condition

$$\bar{M}_{y,w}^s = M_{y,w}^s. \tag{M2}$$

By applying the anti-involution $\mathcal{H} \rightarrow \mathcal{H}$, $\tilde{T}_w \mapsto \tilde{T}_{w^{-1}}$, we also obtain ‘right-handed’ versions of the above formulae (see [22, Section 6]).

REMARK 3.1. We set $P_{y,w} = \varepsilon^{L(w)-L(y)} P_{y,w}^*$. Then it is known that $P_{y,w} \in \mathbb{Z}[\Gamma_{\geq 0}]$; see Lusztig [27, Proposition 5.4]. Furthermore, we have

$$\varepsilon^{L(s)} M_{y,w}^s \in \mathbb{Z}[\Gamma_{>0}] \quad \text{where } sy < y < w < sw (s \in S);$$

see [27, Proposition 6.4]. Assume now that $\Gamma = \mathbb{Z}$ and $L(s) = 1$ for all $s \in S$ (equal parameter case, as in [19]). Then A is the ring of Laurent polynomials in the indeterminate ε . Let $y, w \in W$ be such that $y < w$. Let $s \in S$ be such that $sy < y < w < s$. Now $P_{y,w}^*$ is a polynomial in ε^{-1} .

Consequently, $P_{y,w}$ is a polynomial in ε of degree at most $l(w) - l(y) - 1$. In this situation, it is known that $M_{y,w}^s$ has the following simple description:

$$\begin{aligned} M_{y,w}^s &= \text{coefficient of } \varepsilon^{-1} \text{ in } P_{y,w}^* \\ &= \text{coefficient of } \varepsilon^{l(w)-l(y)-1} \text{ in } P_{y,w}; \end{aligned}$$

see Lusztig [27, Corollary 6.5].

EXAMPLE 3.2. The determination of $M_{y,w}^s$ in the case of unequal parameters is considerably more involved than in the case of equal parameters. For example, assume that there exists some $t \in S$ such that $L(t) > 0$, $ty > y$ and $tw < w$. Then $P_{y,w}^* = \varepsilon^{-L(t)} P_{ty,w}^*$. In the equal parameter case, this implies that $M_{y,w}^s = 0$ unless $ty = w$, in which case $M_{y,w}^s = 1$. In the general case of unequal parameters, if $ty = w$, we have

$$M_{y,w}^s = \begin{cases} 0 & \text{if } L(s) < L(t), \\ 1 & \text{if } L(s) = L(t), \\ \varepsilon^{L(s)-L(t)} + \varepsilon^{L(t)-L(s)} & \text{if } L(s) > L(t); \end{cases}$$

see [22, Proposition 5]. Furthermore, if $w \neq ty$, it can happen that $M_{y,w}^s \neq 0$.

DEFINITION 3.3 (Kazhdan–Lusztig [19] (equal parameter case); see [14, 1.4.11] for general L). A W -graph for \mathcal{H} consists of the following data:

- (a) a base set \mathfrak{X} together with a map I which assigns to each $x \in \mathfrak{X}$ a subset $I(x) \subseteq S$;
- (b) for each $s \in S$ with $L(s) > 0$, a collection of elements

$$\{m_{x,y}^s \mid x, y \in \mathfrak{X} \text{ where } s \in I(x), s \notin I(y)\};$$

- (c) for each $s \in S$ with $L(s) = 0$, a bijection $\mathfrak{X} \rightarrow \mathfrak{X}$, $x \mapsto s.x$.

These data are subject to the following requirements. First we require that, for any $x, y \in \mathfrak{X}$ and $s \in S$ where $m_{x,y}^s$ is defined, we have

$$\varepsilon^{L(s)} m_{x,y}^s \in R[\Gamma_{>0}] \quad \text{and} \quad \overline{m}_{x,y}^s = m_{x,y}^s.$$

Furthermore, let $[\mathfrak{X}]$ be a free A -module with a basis $\{b_y \mid y \in \mathfrak{X}\}$. For $s \in S$, define an A -linear map $\rho_s: [\mathfrak{X}] \rightarrow [\mathfrak{X}]$ by

$$\rho_s(b_y) = \begin{cases} b_{s.y} & \text{if } L(s) = 0, \\ -\varepsilon^{-L(s)} b_y & \text{if } L(s) > 0, s \in I(y), \\ \varepsilon^{L(s)} b_y + \sum_{x \in \mathfrak{X}: s \in I(x)} m_{x,y}^s b_x & \text{if } L(s) > 0, s \notin I(y). \end{cases}$$

Then we require that the assignment $\tilde{T}_s \mapsto \rho_s$ defines a representation of \mathcal{H} .

EXAMPLE 3.4 (Kazhdan–Lusztig [19], Lusztig [22]). Let $y, z \in W$. We write $z \leftarrow_{\mathcal{L}} y$ if there exists some $s \in S$ such that C'_z appears with non-zero multiplicity in $C'_s C'_y$ (when expressed in the C' -basis of \mathcal{H}). Thus, we have:

$$z \leftarrow_{\mathcal{L}} y \Leftrightarrow \begin{cases} \text{if } z = sy \text{ for some } s \in S, \text{ where } L(s) = 0 \text{ or } sy > y, \\ \text{or if } M_{z,y}^s \neq 0 \text{ for some } s \in S, \text{ where } L(s) > 0 \text{ and } sz < z < y < sy. \end{cases}$$

Let $\leq_{\mathcal{L}}$ be the pre-order relation on W generated by $\leftarrow_{\mathcal{L}}$, that is, we have $z \leq_{\mathcal{L}} y$ if there exist elements $z = y_0, y_1, \dots, y_m = y$ in W such that $y_{i-1} \leftarrow_{\mathcal{L}} y_i$ for $1 \leq i \leq m$. Let $\sim_{\mathcal{L}}$ denote the associated equivalence relation; the corresponding equivalence classes are called the *left cells* of W .

Let \mathfrak{C} be a left cell of W (or, more generally, a union of left cells). Then we obtain a corresponding W -graph as follows. We set $I(x) := \{s \in S \mid sx < x\}$ for $x \in \mathfrak{C}$. Furthermore,

if $x, y \in \mathfrak{C}$ and $s \in S$ are such that $L(s) > 0$, $s \in I(x)$ and $s \notin I(y)$, we set

$$m_{x,y}^s := \begin{cases} 1 & \text{if } y = sx, \\ -(-1)^{l(x)+l(y)} M_{x,y}^s & \text{if } x < y, \\ 0 & \text{otherwise.} \end{cases}$$

Finally, if $s \in S$ is such that $L(s) = 0$, then $sw \in \mathfrak{C}$ for all $w \in \mathfrak{C}$, so we obtain a natural bijection $\mathfrak{C} \rightarrow \mathfrak{C}$ by left multiplication. It is known that these data give rise to a W -graph structure on the set \mathfrak{C} . (See [22, Section 6].)

REMARK 3.5. Let $\theta: A \rightarrow R$ be the unique R -linear ring homomorphism such that $\theta(\varepsilon^g) = 1$ for all $g \in \Gamma$. Then, regarding R as an A -module via θ , we have $R \otimes_A \mathcal{H} \cong R[W]$, the group algebra of W over R . Let \mathfrak{C} be a left cell of W . Then we obtain a representation of W on $[\mathfrak{C}]_1 := R \otimes_A [\mathfrak{C}]$, called a ‘left cell representation’ of W . If W is a finite Weyl group and $R = \mathbb{Q}$, the study of these left cell representations is of considerable interest in the representation theory of reductive algebraic groups over finite fields; see Lusztig [23].

DEFINITION 3.6. Assume we are given two W -graphs with underlying base sets \mathfrak{X} and \mathfrak{X}' . Then we say that these two W -graphs are *equivalent* if there exists a bijection $\mathfrak{X} \rightarrow \mathfrak{X}'$, $x \mapsto x'$, such that the map

$$[\mathfrak{X}] \rightarrow [\mathfrak{X}'], \quad b_x \mapsto b_{x'}$$

is an \mathcal{H} -module isomorphism. Similarly, if $\mathfrak{C}, \mathfrak{C}'$ are left cells of W , we write $\mathfrak{C} \approx \mathfrak{C}'$ if the W -graphs associated with \mathfrak{C} and \mathfrak{C}' are equivalent.

EXAMPLE 3.7 (Kazhdan–Lusztig [19, Section 4]). Assume that we are in the equal parameter case where $\Gamma = \mathbb{Z}$ and $L(s) = 1$ for all $s \in S$. Let $s, t \in S$ be such that st has order 3. Let

$$D_R(s, t) = \{w \in W \mid \text{either } ws < w, wt > w \text{ or } ws > w, wt < w\}.$$

If $w \in D_R(s, t)$, then exactly one of the two elements ws, wt belongs to $D_R(s, t)$; we denote it w^* . Thus, we obtain an involution

$$D_R(s, t) \rightarrow D_R(s, t), \quad w \mapsto w^*.$$

If \mathfrak{C} is a left cell of W , then it is known that either \mathfrak{C} is contained in $D_R(s, t)$ or does not meet $D_R(s, t)$ at all; see [19, Proposition 2.4]. This also shows that $y^{-1}w \notin \langle s, t \rangle$ for all $y \neq w$ in \mathfrak{C} . Now, if $\mathfrak{C} \subseteq D_R(s, t)$, then

$$\mathfrak{C}^* = \{w^* \mid w \in \mathfrak{C}\} \subseteq D_R(s, t)$$

also is a left cell of W (see [19, Corollary 4.4(ii)]); furthermore, the W -graphs corresponding to \mathfrak{C} and \mathfrak{C}^* yield identical matrix representations of \mathcal{H} (see [19, Theorem 4.2(iii)]). Thus, we have $\mathfrak{C} \approx \mathfrak{C}^*$ in the sense of Definition 3.6, where the bijection is given by $w \mapsto w^*$ ($w \in \mathfrak{C}$).

DEFINITION 3.8 (cf. Lusztig [25], [27, 14.2]). Let $w \in W$ and assume that $P_{1,w}^* \neq 0$. We define an element $\Delta(w) \in \Gamma_{\geq 0}$ and an integer $0 \neq n_w \in \mathbb{Z}$ by the condition $\varepsilon^{\Delta(w)} P_{1,w}^* \equiv n_w \pmod{\mathbb{Z}[\Gamma_{<0}]}$. Then we say that w is *distinguished* (with respect to L) if $\Delta(w) < \Delta(y)$ for any $y \neq w$ such that $P_{1,y}^* \neq 0$ and y, w belong to the same left cell of W . We set

$$\mathcal{D} := \{w \in W \mid w \text{ distinguished}\}.$$

Thus, if $w \in \mathcal{D}$ and \mathfrak{C} is the left cell containing w , then the function

$$\{y \in \mathfrak{C} \mid P_{1,y}^* \neq 0\} \rightarrow \Gamma, \quad y \mapsto \Delta(y),$$

reaches its minimum at w and w is uniquely determined by this property. (It is known that every left cell contains at least one element y such that $P_{1,y}^* \neq 0$; see, for example, [14, 2.4.7].)

In the equal parameter case where $\Gamma = \mathbb{Z}$ and $L(s) = 1$ for all $s \in S$ (and assuming that W is finite) it is known that $w^2 = 1$ and $n_w = 1$ for all $w \in \mathcal{D}$; furthermore, every left cell contains a (unique) distinguished element. (See Lusztig [25]; see [6] for W of non-crystallographic type.) Hence, in particular, \mathcal{D} is a canonical set of representatives for the left cells of W . If W is of type A , then \mathcal{D} consists precisely of all involutions in W ; in general, \mathcal{D} is strictly contained in the set of involutions of W .

We shall now be interested in determining the above data explicitly, especially for groups of exceptional type. Thus, the computational tasks are:

- given W, L , determine the partition of W into left cells;
- for each left cell \mathcal{C} , determine the numbers $\{M_{x,y}^s\}$;
- determine the set \mathcal{D} of distinguished elements (or the related set $\tilde{\mathcal{D}}$ in Conjecture 5.2 below).

The crucial ingredient in these tasks is the computation of the polynomials $P_{y,w}^*$. This is usually done using some known recursion formulae. In the next section, we discuss a variation of this recursion.

4. Relative Kazhdan–Lusztig polynomials

We keep the general setting of the previous section. In addition, we shall now fix a subset $S' \subseteq S$ and consider the corresponding standard parabolic subgroup $W' = \langle S' \rangle$. Let $X \subseteq W$ be the set of distinguished left coset representatives of W' in W . Every element $w \in W$ can be written uniquely in the form $w = xu$ where $x \in X$, $u \in W'$ and $l(w) = l(x) + l(u)$; see [15, Section 2.1]. We shall frequently use the following fact, due to Deodhar (see [15, 2.1.2]). Let $x \in X$ and $s \in S$. Then we are in exactly one of the following three cases:

- (1) $sx < x$ and $sx \in X$;
- (2) $sx > x$ and $sx \in X$;
- (3) $sx > x$ and $sx \notin X$, in which case $sx = xt$ where $t \in S'$.

We have a corresponding parabolic subalgebra $\mathcal{H}' = \langle \tilde{T}_w \mid w \in W' \rangle_A \subseteq \mathcal{H}$. It is known that, for $w \in W'$, the basis element C'_w lies in \mathcal{H}' , and it is the Kazhdan–Lusztig basis element in \mathcal{H}' .

Let $y \in X$ and $v \in W'$. By [7, Proposition 3.3], we have a unique expression

$$C'_{yv} = \sum_{x \in X, u \in W'} p_{xu,yv}^* \tilde{T}_x C'_u$$

where $p_{yv,yv}^* = 1$ and $p_{xu,yv}^* \in A_{<0}$ if $xu \neq yv$; furthermore, $p_{xu,yv}^* = 0$ unless $xu = yv$ or $x < y$. In the proof of [7, Proposition 3.3], we have also seen that

$$P_{xu,yv}^* = \begin{cases} P_{u,v}^* & \text{if } x = y, \\ p_{xu,yv}^* + \sum_{\substack{w \in W' \\ u < w}} P_{u,w}^* p_{xw,yv}^* & \text{if } x < y. \end{cases}$$

Thus, if we have an efficient algorithm for computing the polynomials $p_{xu,yv}^*$, then we can also determine $P_{xu,yv}^*$ and, hence, the elements $\{M_{xu,yv}^s\}$.

PROPOSITION 4.1. *We have the following recursion formulae for $p_{xu,yv}^*$.*

(a) *If $y = 1$, then*

$$p_{xu,v}^* = \begin{cases} 1 & \text{if } x = 1 \text{ and } u = v, \\ 0 & \text{otherwise.} \end{cases}$$

(b) Now assume that $y \neq 1$ and let $s \in S$ be such that $sy < y$. If $L(s) = 0$, then

$$p_{xu,yv}^* = \begin{cases} p_{sxu,syv}^* & \text{if } sx \in X, \\ p_{xtu,syv}^* & \text{if } sx \notin X, \end{cases}$$

where $t = x^{-1}sx \in S'$ (if $sx \notin X$). If $L(s) > 0$, then

$$p_{xu,yv}^* = \begin{cases} p_{sxu,syv}^* + \varepsilon^{L(s)} p_{xu,syv}^* - \tilde{p}_{xu,yv}^s & \text{if } sx < x, \\ \varepsilon^{-L(s)} p_{sxu,yv}^* & \text{if } sx > x, sx \in X, \\ 0 & \text{if } sx \notin X, tu > u, \\ (\varepsilon^{L(s)} + \varepsilon^{-L(s)}) p_{xu,syv}^* - \tilde{p}_{xu,yv}^s \\ + p_{xtu,syv}^* + \sum_{\substack{w \in W' \\ u < w < tw}} M_{u,w}^t p_{xw,syv}^* & \text{if } sx \notin X, tu < u, \end{cases}$$

where $t = x^{-1}sx \in S'$ (if $sx \notin X$) and

$$\tilde{p}_{xu,yv}^s := \sum_{\substack{z \in X, w \in W' \\ x \leq z \leq sz \text{ and } szw < zw < syv}} p_{xu,zw}^* M_{zw,syv}^s.$$

Proof. (a) This is contained in [7, Proposition 3.3].

(b) This is essentially the same as the proofs of [16, Theorem 5.1] and [17, Proposition 4.1]. However, because of the different normalisations and conventions, we shall sketch the main steps. Let $y \neq 1$ and $s \in S$ be such that $sy < y$. First assume that $L(s) = 0$. Then $C'_s = \tilde{T}_s$ and $C'_s C'_{syv} = C'_{yv}$. Furthermore,

$$\begin{aligned} C'_s C'_{syv} &= \sum_{x \in X, u \in W'} p_{xu,syv}^* \tilde{T}_s \tilde{T}_x C'_u \\ &= \sum_{\substack{x \in X, u \in W' \\ sx \in X}} p_{xu,syv}^* \tilde{T}_{sx} C'_u + \sum_{\substack{x \in X, u \in W' \\ sx \notin X, sx = xt}} p_{xu,syv}^* \tilde{T}_x \tilde{T}_t C'_u \\ &= \sum_{\substack{x \in X, u \in W' \\ sx \in X}} p_{xu,syv}^* \tilde{T}_{sx} C'_u + \sum_{\substack{x \in X, u \in W' \\ sx \notin X, sx = xt}} p_{xu,syv}^* \tilde{T}_x C'_{tu}, \end{aligned}$$

where the last equality holds since $L(t) = L(s)$. This yields the desired formulae.

From now on, assume that $L(s) > 0$. We begin by considering the identity $\tilde{T}_s C'_{yv} = \varepsilon^{L(s)} C'_{yv}$. The coefficient of $\tilde{T}_x C'_u$ on the right-hand side is $\varepsilon^{L(s)} p_{xu,yv}^*$. Now we compute

$$\begin{aligned} \tilde{T}_s C'_{yv} &= \sum_{x \in X, u \in W'} p_{xu,yv}^* \tilde{T}_s \tilde{T}_x C'_u \\ &= \sum_{\substack{x \in X, u \in W' \\ sx < x}} p_{xu,yv}^* \tilde{T}_{sx} C'_u + \sum_{\substack{x \in X, u \in W' \\ sx < x}} p_{xu,yv}^* (\varepsilon^{L(s)} - \varepsilon^{-L(s)}) \tilde{T}_x C'_u \\ &\quad + \sum_{\substack{x \in X, u \in W' \\ sx > x, sx \in X}} p_{xu,yv}^* \tilde{T}_{sx} C'_u + \sum_{\substack{x \in X, w \in W' \\ sx = xt \text{ where } t \in S'}} p_{xw,yv}^* \tilde{T}_x (\tilde{T}_t C'_w) \\ &= \sum_{\substack{x \in X, u \in W' \\ sx > x, sx \in X}} p_{sxu,yv}^* \tilde{T}_x C'_u + \sum_{\substack{x \in X, u \in W' \\ sx < x}} p_{xu,yv}^* (\varepsilon^{L(s)} - \varepsilon^{-L(s)}) \tilde{T}_x C'_u \\ &\quad + \sum_{\substack{x \in X, u \in W' \\ sx < x}} p_{sxu,yv}^* \tilde{T}_x C'_u + \sum_{\substack{x \in X, w \in W' \\ sx = xt \text{ where } t \in S'}} p_{xw,yv}^* \tilde{T}_x (\tilde{T}_t C'_w). \end{aligned}$$

Thus, if $sx > x$ and $sx \in X$, then the coefficient of $\tilde{T}_x C'_u$ in this expression is $p_{sxu,yv}^*$. Hence, we obtain $p_{sxu,yv}^* = \varepsilon^{L(s)} p_{xu,yv}^*$ in this case, as required.

Now assume that $sx > x$ and $sx \notin X$. Then, among the various sums in the above expression for $\tilde{T}_s C'_{yv}$, the term $\tilde{T}_x C'_u$ will only appear in the sum

$$\sum_{\substack{x \in X, w \in W' \\ sx = xt \text{ where } t \in S'}} p_{xw,yv}^* \tilde{T}_x(\tilde{T}_t C'_w).$$

If $tw < w$, then $\tilde{T}_t C'_w = \varepsilon^{L(t)} C'_w$. On the other hand, if $tw > w$, then $\tilde{T}_t C'_w$ is equal to $-\varepsilon^{-L(t)} C'_w$ plus an A -linear combination of terms $C'_{w'}$, where $tw' < w'$. Hence, if $tu > u$, then the coefficient of $\tilde{T}_x C'_u$ in $\tilde{T}_s C'_{yv}$ will be $-\varepsilon^{-L(t)} p_{xu,yv}^*$. Thus, we have $-\varepsilon^{-L(t)} p_{xu,yv}^* = \varepsilon^{L(s)} p_{xu,yv}^*$. Since $L(s) = L(t)$ and $\varepsilon^{2L(s)} \neq -1$, we deduce that $p_{xu,yv}^* = 0$, as required.

To obtain the remaining formulae, we now consider the identity

$$\tilde{T}_s C'_{syv} = C'_{yv} - \varepsilon^{-L(s)} C'_{syv} + \sum_{\substack{z \in X, w \in W' \\ swz < wz < syv}} M_{zw,syv}^s C'_{zw}.$$

Writing $C'_{zw} = \sum_{x \in X, u \in W'} p_{xu,zw}^* \tilde{T}_x C'_u$, we obtain that

$$\begin{aligned} \sum_{\substack{z \in X, w \in W' \\ swz < wz < syv}} M_{zw,syv}^s C'_{zw} &= \sum_{x \in X, u \in W'} \left(\sum_{\substack{z \in X, w \in W' \\ swz < wz < syv}} p_{xu,zw}^* M_{zw,syv}^s \right) \tilde{T}_x C'_u \\ &= \sum_{x \in X, u \in W'} \tilde{p}_{xu,yv}^s \tilde{T}_x C'_u. \end{aligned}$$

Thus, we have

$$\begin{aligned} C'_{yv} &= \tilde{T}_s C'_{syv} + \varepsilon^{-L(s)} C'_{syv} - \sum_{x \in X, u \in W'} \tilde{p}_{xu,yv}^s \tilde{T}_x C'_u \\ &= \tilde{T}_s C'_{syv} + \sum_{x \in X, u \in W'} (\varepsilon^{-L(s)} p_{xu,syv}^* - \tilde{p}_{xu,yv}^s) \tilde{T}_x C'_u. \end{aligned}$$

By a similar computation as above, we have

$$\begin{aligned} \tilde{T}_s C'_{syv} &= \sum_{\substack{x \in X, u \in W' \\ sx > x, sx \in X}} p_{sxu,syv}^* \tilde{T}_x C'_u + \sum_{\substack{x \in X, u \in W' \\ sx < x}} p_{xu,syv}^* (\varepsilon^{L(s)} - \varepsilon^{-L(s)}) \tilde{T}_x C'_u \\ &\quad + \sum_{\substack{x \in X, u \in W' \\ sx < x}} p_{sxu,syv}^* \tilde{T}_x C'_u + \sum_{\substack{x \in X, w \in W' \\ sx = xt \text{ where } t \in S'}} p_{xw,syv}^* \tilde{T}_x(\tilde{T}_t C'_w). \end{aligned}$$

Now let $x \in X$ be such that $sx < x$. Then we conclude that

$$\begin{aligned} p_{xu,yv}^* &= p_{xu,syv}^* (\varepsilon^{L(s)} - \varepsilon^{-L(s)}) + p_{sxu,syv}^* + \varepsilon^{-L(s)} p_{xu,syv}^* - \tilde{p}_{xu,yv}^s \\ &= \varepsilon^{L(s)} p_{xu,syv}^* + p_{sxu,syv}^* - \tilde{p}_{xu,yv}^s, \end{aligned}$$

as required. Finally, assume that $sx > x$, $sx \notin X$ and $tu < u$. Then $p_{xu,yv}^*$ will be equal to $\varepsilon^{-L(s)} p_{xu,syv}^* - \tilde{p}_{xu,yv}^s$ plus the coefficient of $\tilde{T}_x C'_u$ in

$$\sum_{\substack{x \in X, w \in W' \\ sx = xt \text{ where } t \in S'}} p_{xw,syv}^* \tilde{T}_x(\tilde{T}_t C'_w).$$

If $tw < w$, then $\tilde{T}_t C'_w = \varepsilon^{L(t)} C'_w$. On the other hand, if $tw > w$, then

$$\tilde{T}_t C'_w = C'_{tw} - \varepsilon^{-L(t)} C'_w + \sum_{\substack{u \in W' \\ tu < u < w}} M_{u,w}^t C'_u.$$

Hence, since $L(s) = L(t)$, we obtain

$$p_{xu,yv}^* = (\varepsilon^{L(s)} + \varepsilon^{-L(s)}) p_{xu,sv}^* - \hat{p}_{xu,yv}^s + p_{xtu,sv}^* + \sum_{\substack{w \in W' \\ u < w < tw}} M_{u,w}^t p_{xw,sv}^*,$$

as desired. □

REMARK 4.2. Assume that $W' = \{1\}$. Then $X = W$ and $P_{x,y}^* = p_{x,y}^*$ for all $x, y \in X$. For any $x \in X$ and $s \in S$, we have $sx \in X$ and either $sx < x$ or $sx > x$. Thus, only the first two out of the four cases in Proposition 4.1(b) will occur. These two cases yield the known recursion formulae for the polynomials $P_{x,y}^*$.

REMARK 4.3. Let us set $p_{xu,yv} = \varepsilon^{L(yv)-L(xu)} p_{xu,yv}^*$. Then $p_{xu,yv} \in A_{\geq 0}$. Indeed, we have the recursion formulae.

(a) If $y = 1$, then

$$p_{xu,v} = \begin{cases} 1 & \text{if } x = 1 \text{ and } u = v, \\ 0 & \text{otherwise.} \end{cases}$$

(b) Now assume that $y \neq 1$ and let $s \in S$ be such that $sy < y$. If $L(s) = 0$, then

$$p_{xu,yv} = \begin{cases} p_{sxu,sv} & \text{if } sx \in X, \\ p_{xtu,sv} & \text{if } sx \notin X, \end{cases}$$

where $t = x^{-1}sx \in S'$ (if $sx \notin X$). If $L(s) > 0$, then

$$p_{xu,yv} = \begin{cases} p_{sxu,sv} + \varepsilon^{2L(s)} p_{xu,sv} - \hat{p}_{xu,yv}^s & \text{if } sx < x, \\ p_{sxu,yv} & \text{if } sx > x, sx \in X, \\ 0 & \text{if } sx \notin X, tu > u, \\ (\varepsilon^{2L(s)} + 1) p_{xu,sv} - \hat{p}_{xu,yv}^s + \varepsilon^{2L(s)} p_{xtu,sv} & \\ + \sum_{\substack{w \in W' \\ u < w < tw}} \varepsilon^{L(tw)-L(u)} M_{u,w}^t p_{xw,sv} & \text{if } sx \notin X, tu < u, \end{cases}$$

where $t = x^{-1}sx \in S'$ (if $sx \notin X$) and

$$\hat{p}_{xu,yv}^s = \sum_{\substack{z \in X, w \in W' \\ x < z \leq sy \text{ and } szw < zw < syv}} \varepsilon^{L(syv)-L(zw)} p_{xu,zw} (\varepsilon^{L(s)} M_{zw,sv}^s).$$

With this renormalisation, it also follows that

$$P_{xu,yv} = \begin{cases} P_{u,v} & \text{if } x = y, \\ p_{xu,yv} + \sum_{\substack{w \in W' \\ u < w}} P_{u,w} p_{xw,yv} & \text{if } x < y. \end{cases}$$

LEMMA 4.4. Let $x, y \in X$, $u, v \in W'$ and $s \in S$ be such that $L(s) > 0$ and $sxu < xu < yv < syv$. Let

$$\pi_{xu,yv}^s := \sum_{\substack{z \in X, w \in W' \\ x < z \leq y \text{ and } szw < zw < yv}} P_{xu,zw}^* M_{zw,yv}^s.$$

Then the following hold. If $sx < x$, then

$$M_{xu,yv}^s - \varepsilon^{L(s)} p_{xu,yv}^* + \pi_{xu,yv}^s \in A_{<0}; \tag{M1'}$$

on the other hand, if $sx \notin X$ and $tu < u$ (where $t = x^{-1}sx \in S'$), then

$$M_{xu,yv}^s - \varepsilon^{L(s)} p_{xu,yv}^* + \pi_{xu,yv}^* - \sum_{\substack{w \in W' \\ u < w < tw}} M_{u,w}^t p_{xw,yv}^* \in A_{<0}. \tag{M1''}$$

Note that these conditions, together with the symmetry condition $\overline{M}_{xu,yv}^s = M_{xu,yv}^s$, determine $M_{xu,yv}^s$ inductively.

Proof. We consider the identity

$$C'_{syv} = \varepsilon^{-L(s)} C'_{yv} + \tilde{T}_s C'_{yv} - \sum_{\substack{z \in X, w \in W' \\ swz < wz < yv}} M_{zw,yv}^s C'_{zw}. \tag{†}$$

The coefficient of $\tilde{T}_x C'_u$ on the left-hand side is $p_{xu,sv}^*$. Assume first that $sx < x$. Then, arguing as in the proof of Proposition 4.1, we find that the coefficient of $\tilde{T}_x C'_u$ on the right-hand side of (†) is

$$p_{sxu,yv}^* + \varepsilon^{L(s)} p_{xu,yv}^* - \tilde{p}_{xu,sv}^s.$$

Now we note that

$$\tilde{p}_{xu,sv}^s = \sum_{\substack{z \in X, w \in W' \\ szw < zw < yv}} p_{xu,zw}^* M_{zw,yv}^s = M_{xu,yv}^s + \pi_{xu,yv}^s.$$

Thus, we conclude that

$$M_{xu,yv}^s - \varepsilon^{L(s)} p_{xu,yv}^* + \pi_{xu,yv}^s = \tilde{p}_{xu,sv}^s - \varepsilon^{L(s)} p_{xu,yv}^* = p_{sxu,yv}^* - p_{xu,sv}^*.$$

This expression lies in $A_{<0}$; thus, we have shown that (M1') holds. On the other hand, if $sx \notin X$ and $tu < u$ (where $t = x^{-1}sx \in S'$), then the coefficient of $\tilde{T}_x C'_u$ on the right-hand side of (†) is

$$(\varepsilon^{L(s)} + \varepsilon^{-L(s)}) p_{xu,yv}^* - \tilde{p}_{xu,sv}^s + p_{xtu,yv}^* + \sum_{\substack{w \in W' \\ u < w < tw}} M_{u,w}^t p_{xw,yv}^*.$$

Then a similar argument shows that (M1'') holds. □

EXAMPLE 4.5. Let $x, y \in X$, $u, v \in W'$ and $s \in S$ be such that $L(s) > 0$ and $sxu < xu < yv < syv$. Assume now that $x = y$. First of all, this forces that $u < v$ and $\pi_{xu,yv}^s = 0$. Furthermore, we must have $sx = sy \notin X$. Thus, condition (M1'') in Lemma 4.4 yields that

$$M_{xu,yv}^s - \varepsilon^{L(s)} p_{xu,xv}^* - \sum_{\substack{w \in W' \\ u < w < tw}} M_{u,w}^t p_{xw,xv}^* \in A_{<0}.$$

Now, recall that for any $w \in W$, we have $p_{xw,xv}^* = 0$ unless $w = v$. It follows that

$$M_{xu,xv}^s = M_{u,v}^t \quad (\text{where } t = x^{-1}sx \in S').$$

(This shows, in particular, that we can have $M_{xu,yv}^s \neq 0$ even if $p_{xu,yv}^* = 0$.)

REMARK 4.6. Assume that $\Gamma = \mathbb{Z}$ and $L(s) = 1$ for all $s \in S$ (equal parameter case). Then A is the ring of Laurent polynomials in the indeterminate ε . Let $s \in S$, $x, y \in X$ and $u, v \in W'$

be such that $sxu < xu < yv < syv$. Then

$$\begin{aligned}
 M_{xu,yv}^s &= \text{coefficient of } \varepsilon^{-1} \text{ in } \begin{cases} P_{u,v}^* & \text{if } x = y, \\ p_{xu,yv}^* & \text{if } x \neq y. \end{cases} \\
 &= \text{coefficient of } \varepsilon^{L(yv)-L(xu)-1} \text{ in } \begin{cases} P_{u,v} & \text{if } x = y, \\ p_{xu,yv} & \text{if } x \neq y. \end{cases}
 \end{aligned}$$

This is easily seen using the formulae in Lemma 4.4 and Example 4.5; note also that Remark 3.1 already shows that $M_{xu,yv}^s \in \mathbb{Z}$ in this case.

COROLLARY 4.7. *Let \mathfrak{C}' be a left cell of W' . Then we have the following recursion formulae for $p_{xu,yv}^*$ where $x, y \in X$ and $u, v \in \mathfrak{C}'$.*

(a) *If $y = 1$, then*

$$p_{xu,v}^* = \begin{cases} 1 & \text{if } x = 1 \text{ and } u = v, \\ 0 & \text{otherwise.} \end{cases}$$

(b) *Now assume that $y \neq 1$ and let $s \in S$ be such that $sy < y$. If $L(s) = 0$, then*

$$p_{xu,yv}^* = \begin{cases} p_{sxu,syv}^* & \text{if } sx \in X, \\ p_{xtu,syv}^* & \text{if } sx \notin X, \end{cases}$$

where $t = x^{-1}sx \in S'$ (if $sx \notin X$). If $L(s) > 0$, then

$$p_{xu,yv}^* = \begin{cases} p_{sxu,syv}^* + \varepsilon^{L(s)} p_{xu,syv}^* - \tilde{p}_{xu,yv}^s & \text{if } sx < x, \\ \varepsilon^{-L(s)} p_{sxu,yv}^* & \text{if } sx > x, sx \in X, \\ 0 & \text{if } sx \notin X, tu > u, \\ (\varepsilon^{L(s)} + \varepsilon^{-L(s)}) p_{xu,syv}^* - \tilde{p}_{xu,yv}^s \\ \quad + \underbrace{p_{xtu,syv}^*}_{\text{only if } tu \in \mathfrak{C}'} + \sum_{\substack{w \in \mathfrak{C}' \\ u < w < tw}} M_{u,w}^t p_{xw,syv}^* & \text{if } sx \notin X, tu < u, \end{cases}$$

where $t = x^{-1}sx \in S'$ (if $sx \notin X$) and

$$\tilde{p}_{xu,yv}^s := \sum_{\substack{z \in X, w \in \mathfrak{C}' \\ x \leq z \leq sy \text{ and } zw < zw < syv}} p_{xu,zw}^* M_{zw,syv}^s.$$

Proof. This immediately follows from Proposition 4.1 and the following facts. Let $\leq_{\mathfrak{L}}$ be the Kazhdan–Lusztig left pre-order relation on W' . By [7, Proposition 3.3], we have $p_{xu,yv}^* = 0$ unless $xu = yv$ or $x < y$ and $u \leq_{\mathfrak{L}} v$; furthermore, by [7, Section 4], we have the implication ' $xu \leq_{\mathfrak{L}} yv \Rightarrow u \leq_{\mathfrak{L}} v$ '. □

ALGORITHM A. The following inductive procedure produces the partition of W into left cells and the associated W -graphs.

If $W = \{1\}$, then $\mathfrak{C} = \{1\}$ is the only left cell and there is a canonical W -graph associated with it. Now assume that $W \neq \{1\}$ and let $W' \subsetneq W$ be a proper parabolic subgroup. By induction, we obtain the partition $W' = \mathfrak{C}'_1 \amalg \dots \amalg \mathfrak{C}'_k$ of W' into left cells and the associated W -graphs. Now fix $i \in \{1, \dots, k\}$. Then, by the main result of [7], the set $X\mathfrak{C}'_i$ is a union of left cells, that is, we have

$$X\mathfrak{C}'_i = \mathfrak{C}_{i,1} \amalg \mathfrak{C}_{i,2} \amalg \dots \amalg \mathfrak{C}_{i,k_i}$$

where $\mathfrak{C}_{i,1}, \mathfrak{C}_{i,2}, \dots, \mathfrak{C}_{i,k_i}$ are left cells of W and X is the set of distinguished left coset representatives of W' in W . These left cells are determined as follows. By Corollary 4.7 and

Lemma 4.4, there is a recursive algorithm for simultaneously computing the polynomials

$$\{p_{xu,yv}^*\} \quad \text{and} \quad \{M_{xu,yv}^s\} \quad (x, y \in X, u, v \in \mathcal{C}'_i).$$

(The computations will only involve the elements in the fixed set $X\mathcal{C}'_i$.) Once this is achieved, the set $X\mathcal{C}'_i$ is decomposed into left cells by the procedure in Example 3.4 (that is, by explicitly working out the transitive closure of the relation $\leftarrow_{\mathcal{L}}$); this also yields automatically the associated W -graphs. By letting i run over all indices in $\{1, \dots, k\}$ we eventually obtain all the left cells of W and the associated W -graphs.

In PyCox, the function `klcells` implements the procedure in Algorithm A. As may be expected this leads to significant efficiency improvements compared with the use of ordinary Kazhdan–Lusztig polynomials (where the recursion involves *all* elements of W). In the equal parameter case, one can apply some further simplifications to reduce the number of left cells that have to be ‘induced’ from W' to W : first of all, it is sufficient to induce only one left cell from each pair of left cells which are related by multiplication with the longest element in W' . (This follows from Yin [32].) Furthermore, assume that $i_1, i_2 \in \{1, \dots, k\}$ are such that $\mathcal{C}'_{i_1} \approx \mathcal{C}'_{i_2}$ in the sense of Definition 3.6. Then, by [9, Corollary 3.10], it is known that, for a suitable labelling, we have $k_{i_1} = k_{i_2}$ and $\mathcal{C}_{i_1,j} \approx \mathcal{C}_{i_2,j}$ for all $j \in \{1, \dots, k_{i_1}\}$. Thus, it is sufficient to induce one left cell from each orbit under the star operations in Example 3.7, and then to apply the star operations to the resulting cells of W . This leads again to an enormous gain in efficiency. For example, in the computation of the left cells for type E_7 , we only need to induce 34 (instead of a total of 578) left cells from a parabolic subgroup of type D_6 ; see Table 2. The efficiency also depends on the choice of W' . For example, in type E_7 it is more efficient to use W' of type D_6 than of type E_6 ; in all other cases, we have chosen W' such that the index $|W : W'|$ is as small as possible. Finally note that, if one is only interested in the partition of the group into left cells, then there are further techniques available; see, for example, Chen–Shi [3].

One of the main advantages of being able to compute left cells and the corresponding W -graphs in a language like Python lies in the fact that it provides immediate functionality for further handling of the data. We shall see a concrete example of this in the next section.

TABLE 2. Examples of computations of left cells (equal parameters).

W	$ W $	W'	$ X $	$\max\{ X\mathcal{C}' \}$	# left cells	# (left cells/ \approx)	Time
$I_2(5)$	10	A_1	5	5	4	4	0.01 s
H_3	120	$I_2(5)$	12	48	22	15	0.2 s
D_4	192	A_3	8	24	36	12	0.1 s
F_4	1 152	B_3	24	120	72	29	1 s
D_5	1 920	D_4	10	140	126	16	0.7 s
H_4	14 400	H_3	120	960	206	90	370 s
D_6	23 040	D_5	12	300	578	34	10 s
E_6	51 840	D_5	27	675	652	21	45 s
D_7	322 560	D_6	14	1 190	2 416	49	190 s
A_8	362 880	A_7	9	810	2 620	30	140 s
E_7	2 903 040	D_6	126	10 710	6 364	56	4 h
D_8	5 160 960	D_7	16	3 696	11 504	90	4 h
B_8	10 321 920	B_7	16	8 848	15 304	346	58 h

(B_8 requires 9 GB main memory; in all other cases, 4 GB are sufficient.)

5. *Leading coefficients of character values*

We keep the general setting of the previous sections; we assume now that W is finite and let $R = \mathbb{R}$. It is known that this is a splitting field for W (see [15, 6.3.8]). Let $\text{Irr}(W)$ denote the set of simple $\mathbb{R}[W]$ -modules (up to isomorphism). Let K be the field of fractions of A and $\mathcal{H}_K = K \otimes_A \mathcal{H}$. Then it is known that \mathcal{H}_K is split semisimple and abstractly isomorphic to $K[W]$ (see [15, 9.3.5]); furthermore, the map $\varepsilon^g \mapsto 1$ ($g \in \Gamma$) induces a bijection between $\text{Irr}(\mathcal{H}_K)$ and $\text{Irr}(W)$ (see [15, 8.1.7]). Given $E \in \text{Irr}(W)$, we denote by E_ε the corresponding irreducible representation of \mathcal{H}_K . It is known that

$$\text{trace}(\tilde{T}_w, E_\varepsilon) \in \mathbb{R}[\Gamma] \quad \text{for all } w \in W$$

(see [15, 9.3.5]). Thus, we can define

$$\mathbf{a}_E := \min\{g \in \Gamma_{\geq 0} \mid \varepsilon^g \text{trace}(\tilde{T}_w, E_\varepsilon) \in \mathbb{R}[\Gamma_{\geq 0}] \text{ for all } w \in W\}.$$

Consequently, there are unique numbers $c_{w,E} \in \mathbb{R}$ ($w \in W$) such that

$$\varepsilon^{\mathbf{a}_E} \text{trace}(\tilde{T}_w, E_\varepsilon) = (-1)^{l(w)} c_{w,E} + \text{‘higher terms’},$$

where ‘higher terms’ means an \mathbb{R} -linear combination of terms ε^g where $g > 0$. These numbers are the ‘leading coefficients of character values’, as defined and studied by Lusztig [23, 26, 27]. Since $\text{trace}(\tilde{T}_w, E_\varepsilon) = \text{trace}(\tilde{T}_{w^{-1}}, E_\varepsilon)$ for all $w \in W$ (see [15, 8.2.6]), we certainly have

$$c_{w,E} = c_{w^{-1},E} \quad \text{for all } w \in W.$$

Given E , there is at least one $w \in W$ such that $c_{w,E} \neq 0$ (by the definition of \mathbf{a}_E). Hence, the sum of all $c_{w,E}^2$ ($w \in W$) will be strictly positive and so we can write that sum as $f_E \dim E$ where $f_E \in \mathbb{R}$ is strictly positive. In fact, we have the following orthogonality relations (see [15, Excercise 9.8]):

$$\sum_{w \in W} c_{w,E} c_{w,E'} = \begin{cases} f_E \dim E & \text{if } E \cong E', \\ 0 & \text{otherwise.} \end{cases}$$

The connection with left cells is given by the following result, first proved by Lusztig [23, 5.7], [26, 3.5] in the equal parameter case (where the proof ultimately relies upon a geometric interpretation of the basis $\{C'_w\}$ of \mathcal{H}); the general case (where no geometric interpretation is available) is proved by an elementary argument in [12, 3.5, 3.8]. Given $E \in \text{Irr}(W)$ and a left cell \mathfrak{C} of W , we denote by $m(\mathfrak{C}, E)$ the multiplicity of E as an irreducible constituent of the left cell module $[\mathfrak{C}]_1$ (as defined in Remark 3.5).

PROPOSITION 5.1. *Let $E \in \text{Irr}(W)$ and \mathfrak{C} be a left cell.*

(a) *Let also $E' \in \text{Irr}(W)$. Then*

$$\sum_{w \in \mathfrak{C}} c_{w,E} c_{w,E'} = \begin{cases} f_E m(\mathfrak{C}, E) & \text{if } E \cong E', \\ 0 & \text{otherwise.} \end{cases}$$

(b) *If $c_{w,E} \neq 0$ for some $w \in \mathfrak{C}$, then we also have $w^{-1} \in \mathfrak{C}$.*

In what follows, it will be important to renormalise the leading coefficients. In the equal parameter case, this renormalisation is suggested by the formula in [26, 3.5(b)] (see Remark 5.5 below). In the unequal parameter case, we cannot just take the analogous formula; instead, we proceed as follows where we partly rely on a conjectural property. Following [10, Section 3] and [14, Section 1.5], we define real numbers

$$\check{n}_w := \sum_{E \in \text{Irr}(W)} f_E^{-1} c_{w,E} \quad \text{for any } w \in W.$$

(Note that, in [14, Section 1.3], we have omitted the factor $(-1)^{l(w)}$ in the definition of $c_{w,E}$; hence, the numbers \check{n}_w in [14, Section 1.5] will be equal to $(-1)^{l(w)}\check{n}_w$.) With this notation, we can now state the following conjecture.

CONJECTURE 5.2. Let $\tilde{\mathcal{D}} := \{w \in W \mid \check{n}_w \neq 0\}$. Then the following hold.

- (a) Every left cell of W contains a unique element of $\tilde{\mathcal{D}}$.
- (b) We have $w^2 = 1$ and $\check{n}_w = \pm 1$ for every $w \in \tilde{\mathcal{D}}$.

It is known that every left cell contains at least one element of $\tilde{\mathcal{D}}$. (This follows from [14, 1.8.5 and 2.1.20].) We expect that $\tilde{\mathcal{D}}$ is precisely the set \mathcal{D} defined in Definition 3.8 and that $\check{n}_w = n_w$ for all $w \in \mathcal{D}$. The advantage of the definition of $\tilde{\mathcal{D}}$ is that this set can actually be computed in an efficient way; see Algorithm B below.

REMARK 5.3. Conjecture 5.2 and the equality $\tilde{\mathcal{D}} = \mathcal{D}$ are known to hold if Lusztig’s properties **P1–P15** in [27, 14.2] are satisfied for W, L (see [14, Section 2.3] for details). By [27, Section 16] (see [6] for W of non-crystallographic type), **P1–P15** do hold in the equal parameter case where $\Gamma = \mathbb{Z}$ and $L(s) = 1$ for all $s \in S$. It is also known that then the coefficients of the polynomials $P_{y,w}^*$ are non-negative; see [1, 24]. Hence, in this case, we have

$$\check{n}_w = n_w = 1 \quad \text{for all } w \in \tilde{\mathcal{D}}.$$

We shall consider some cases of unequal parameters in the examples below.

DEFINITION 5.4. Assume that Conjecture 5.2 holds for W, L . Let $w \in W$ and $d \in \tilde{\mathcal{D}}$ be the unique element such that w, d belong to the same left cell. Then we set

$$c_{w,E}^* := (-1)^{l(w)+l(d)}\check{n}_d c_{w,E} \quad \text{for all } E \in \text{Irr}(W).$$

REMARK 5.5. Assume that we are in the equal parameter case where $\Gamma = \mathbb{Z}$ and $L(s) = 1$ for all $s \in S$. Let us check that then our renormalisation corresponds to the formula in [26, 3.5(b)]. Thus, we claim that

$$c_{w,E}^* = (-1)^{l(w)+\mathbf{a}_E} c_{w,E} \quad \text{for all } w \in W \text{ and } E \in \text{Irr}(W). \tag{a}$$

This is seen as follows. Let $w \in W$. By Remark 5.3, we have $\check{n}_d = 1$ where $d \in \tilde{\mathcal{D}}$ is the unique element such that w, d belong to the same left cell. Hence, it will be enough to show that

$$l(d) \equiv \mathbf{a}_E \pmod{2} \quad \text{for all } E \in \text{Irr}(W) \text{ such that } c_{w,E} \neq 0. \tag{b}$$

Now let $E \in \text{Irr}(W)$ be such that $c_{w,E} \neq 0$. Then, by [26, 3.3], we have $\mathbf{a}_E = \mathbf{a}(w)$ where $z \mapsto \mathbf{a}(z)$ ($z \in W$) is the function defined by Lusztig [24]. This function is constant on the left cells of W and so $\mathbf{a}_E = \mathbf{a}(d)$. Thus, it remains to show that $l(d) \equiv \mathbf{a}(d) \pmod{2}$. But this immediately follows from [24, 3.2] (see also [14, Remark 2.3.5]) and property **P5** in [27, 14.2]. An explanation for the renormalisation in (a) can be given by using the asymptotic algebra J introduced by Lusztig [25]. This algebra has a basis $\{t_w \mid w \in W\}$ and one can easily check that the map $t_w \mapsto (-1)^{l(w)+l(d)}t_w$ (where $d \in \tilde{\mathcal{D}}$ is such that w, d belong to the same left cell) defines an algebra automorphism of J .

REMARK 5.6. Assume that Conjecture 5.2 holds for W, L . Let \mathfrak{C} be a left cell and consider the unique element $d \in \tilde{\mathcal{D}} \cap \mathfrak{C}$. Then we have:

$$c_{d,E}^* = m(\mathfrak{C}, E) \quad \text{for all } E \in \text{Irr}(W).$$

In the framework of Lusztig’s theory of the asymptotic algebra, the above statement appears in [27, 21.4]; see also [23, Chapter 12]. One can give a more elementary argument, as follows. We consider the algebra \tilde{J} defined in [14, Section 1.5]. Using \tilde{J} , one can define a partition of W into ‘left \tilde{J} -cells’; see [14, Section 1.6]. By [14, Proposition 2.1.20], every Kazhdan–Lusztig left cell is a union of left \tilde{J} -cells. Hence, by [12, Lemma 3.7] and [14, Example 1.8.5], we have

$$m(\mathfrak{C}, E) = \sum_{d \in \tilde{\mathcal{D}} \cap \mathfrak{C}} \check{n}_d c_{d,E} \quad \text{for all } E \in \text{Irr}(W).$$

Thus, the claim immediately follows from the assumption that Conjecture 5.2 holds. In particular, we have the following formula for the decomposition of the left cell module $[\mathfrak{C}]_1$:

$$[\mathfrak{C}]_1 = \sum_{E \in \text{Irr}(W)} m(\mathfrak{C}, E) E = \check{n}_d \sum_{E \in \text{Irr}(W)} c_{d,E} E$$

(in the appropriate Grothendieck group of representations).

DEFINITION 5.7. Assume that Conjecture 5.2 holds for W, L . Let \mathfrak{C} be a left cell of W and denote by $\text{Irr}(W \mid \mathfrak{C})$ the set of all $E \in \text{Irr}(W)$ such that E is an irreducible constituent of $[\mathfrak{C}]_1$. Then we define

$$\mathfrak{X}(W \mid \mathfrak{C}) := (c_{w,E}^*)_{E \in \text{Irr}(W \mid \mathfrak{C}), w \in \mathfrak{C} \cap \mathfrak{C}^{-1}}.$$

(Following Lusztig [26, 27], this table can be interpreted as the character table of the subalgebra of the asymptotic algebra J which is spanned by t_w for $w \in \mathfrak{C} \cap \mathfrak{C}^{-1}$; the unique element $d \in \tilde{\mathcal{D}} \cap \mathfrak{C}$ corresponds to the identity element of this algebra, in accordance with Remark 5.6.) Note that, by Proposition 5.1, we have $E \in \text{Irr}(W \mid \mathfrak{C})$ if and only if $c_{w,E} \neq 0$ for some $w \in \mathfrak{C}$; furthermore, $c_{w,E} = 0$ unless w, w^{-1} belong to the same left cell. Thus, every non-zero leading coefficient will appear in one of the tables $\mathfrak{X}(W \mid \mathfrak{C})$ as \mathfrak{C} runs over the left cells of W .

EXAMPLE 5.8. Assume that W is a finite Weyl group and that we are in the equal parameter case where $\Gamma = \mathbb{Z}$ and $L(s) = 1$ for all $s \in S$. Then the tables $\mathfrak{X}(W \mid \mathfrak{C})$ have been determined explicitly by Lusztig [26, 3.14], based on the results in [23]. In particular, it turns out that, if $E \in \text{Irr}(W)$ is ‘special’ in the sense of Lusztig [21], then $c_{w,E}^* \geq 0$ for all $w \in W$. (Except for some exceptional cases in type E_7 and E_8 , the latter statement already appeared in [23, Proposition 7.1]; one can also check this property directly in the exceptional cases by using the methods in the proof of [23, Proposition 7.1].) Furthermore, still assuming that E is special, we actually have $c_{w,E}^* > 0$ for all $w \in \mathfrak{C} \cap \mathfrak{C}^{-1}$ where \mathfrak{C} is a left cell such that $m(\mathfrak{C}, E) > 0$. Thus, for any given left cell \mathfrak{C} , all the entries in the row of $\mathfrak{X}(W \mid \mathfrak{C})$ corresponding to the unique special representation occurring in $[\mathfrak{C}]_1$ are strictly positive. Note that, by Proposition 5.1, there can be at most one row with this property.

We shall now be interested in computing the tables $\mathfrak{X}(W \mid \mathfrak{C})$ explicitly in the case where W is not of crystallographic type and also in some examples involving unequal parameters.

ALGORITHM B. The following procedure verifies if Conjecture 5.2 holds for W, L and determines the tables $\mathfrak{X}(W \mid \mathfrak{C})$ for all left cells of W .

Step 1. Let $\text{Cl}(W)$ be the set of conjugacy classes of W . Using the inductive description in [15, Proposition 8.2.7], we determine the ‘class polynomials’ $f_{w,C} \in A$ for all $w \in W$ and all $C \in \text{Cl}(W)$. These polynomials have the following property. For $w \in W$, define $T_w := \varepsilon^{L(w)} \tilde{T}_w$;

for any $C \in \text{Cl}(W)$ let $d_{\min}(C) = \min\{l(w) \mid w \in C\}$ and let $w_C \in C$ be a representative such that $l(w_C) = d_{\min}(C)$. Then we have:

$$\text{trace}(T_w, E_\varepsilon) = \sum_{C \in \text{Cl}(W)} f_{w,C} \text{trace}(T_{w_C}, E_\varepsilon) \quad \text{for all } E \in \text{Irr}(W).$$

Step 2. By [15, Chapter 10, 11], the character tables

$$X(\mathcal{H}) = (\text{trace}(T_{w_C}, E_\varepsilon))_{E \in \text{Irr}(W), C \in \text{Cl}(W)}$$

are explicitly known. Furthermore, the functions $E \mapsto \mathbf{a}_E$ and $E \mapsto f_E$ are explicitly known; see, for example, [15, Appendix] (equal parameter case) and the summary in [14, Section 1.3] for unequal parameters. Thus, in combination with the class polynomials in step 1, we can explicitly compute all the leading coefficients $c_{w,E}$ where $w \in W$ and $E \in \text{Irr}(W)$. Consequently, we can then also compute the numbers \check{n}_w for all $w \in W$, and the set \tilde{D} .

Step 3. By Algorithm A, we can determine the partition of W into left cells. (We do not need the additional information on the associated W -graphs here.) Let \mathfrak{C} be a fixed left cell. Using the data in step 2, we can then explicitly verify if Conjecture 5.2 holds. Using the formula in Proposition 5.1(a), we can find the multiplicities $m(\mathfrak{C}, E)$ for all $E \in \text{Irr}(W)$. Thus, the table $\mathfrak{X}(W \mid \mathfrak{C})$ is determined.

In PyCox, the function `leftcellleadingcoeffs` implements the procedure in Algorithm B for a given left cell. This allows the explicit determination of all the tables $\mathfrak{X}(W \mid \mathfrak{C})$ for groups W of rank up to around 7 and any weight function L . All this even works for type E_7 where it takes about 3 hours and requires 4 GB of main memory. With this information, it is then straightforward to verify Kottwitz’s conjecture for type E_7 , as mentioned in the introduction.

Performing only steps 1 and 2 of Algorithm B yields the set \tilde{D} and all the leading coefficients $c_{w,E}$. This even works for type E_8 where it takes nearly 18 days and requires about 22 GB of main memory to compute the 101 796 elements in \tilde{D} . (As far as I am aware, these elements have not been explicitly known before.) All the known sets \tilde{D} for W of exceptional type are explicitly stored in a compact format within PyCox; see the function `libdistinv`.

The explicit data in the examples below have been computed with the help of the PyCox function `leftcellleadingcoeffs`.

EXAMPLE 5.9. Let W be of type H_3 or H_4 . Let \mathfrak{C} be a left cell of W . Using Algorithm A, we obtain the left cells of W ; we have

$$|\mathfrak{C}| \in \begin{cases} \{1, 5, 6, 8\} & \text{in type } H_3, \\ \{1, 8, 18, 25, 32, 36, 326, 392, 436\} & \text{in type } H_4. \end{cases}$$

(See also Alvis [1].) If $|\mathfrak{C}|$ equals 1, 5, 25 or 36, then $[\mathfrak{C}]_1$ is irreducible and the table $\mathfrak{X}(W \mid \mathfrak{C})$ is (1). Now assume that $|\mathfrak{C}|$ equals 6, 8, 18 or 32. Then $[\mathfrak{C}]_1 = E_1 \oplus E_2$ where $E_1 \not\cong E_2$, $\dim E_1 = \dim E_2$ and where we choose the notation such that E_1 is a special representation. Then the table $\mathfrak{X}(W \mid \mathfrak{C})$ is

$$\begin{array}{ccc} E_1 & 1 & 1 \\ E_2 & 1 & -1 \end{array} \quad \text{or} \quad \begin{array}{ccc} E_1 & 1 & \alpha \\ E_2 & 1 & 1 - \alpha, \end{array}$$

according to whether f_{E_1} equals 2 or $2 + \alpha$, respectively, where $\alpha = \frac{1}{2}(1 + \sqrt{5})$. Finally, if $|\mathfrak{C}|$ equals 326, 392 or 436, then $\mathfrak{X}(W \mid \mathfrak{C})$ is given by Table 3. Here, we use the notation for $\text{Irr}(W)$ defined in [15, Appendix]. As in Example 5.8 we note that there is a row in which all entries are strictly positive, and this row corresponds to the unique special representation occurring in $[\mathfrak{C}]_1$ (which is 24_s in Table 3).

TABLE 3. The tables $\mathfrak{X}(W|\mathbf{C})$ for big left cells in type H_4 ; $\alpha = (1 + \sqrt{5})/2$.

Left cells with 326 elements													
8_r	1	0	-1	1	0	0	-1	0	0	1	-1	0	1
8_{rr}	1	-1	0	1	-1	-1	0	1	1	-1	0	1	-1
18_r	1	1	0	-1	1	1	0	1	-1	-1	0	1	1
24_t	1	2-\alpha	4-3\alpha	5-3\alpha	2-\alpha	2-\alpha	0	-2+\alpha	-2+\alpha	-5+3\alpha	-4+3\alpha	-2+\alpha	-1
$24_{t'}$	1	1+\alpha	1+3\alpha	2+3\alpha	1+\alpha	1+\alpha	0	-1-\alpha	-1-\alpha	-2-3\alpha	-1-3\alpha	-1-\alpha	-1
24_s	1	2-2\alpha	7-4\alpha	13-8\alpha	6-4\alpha	16-10\alpha	7-4\alpha	7-4\alpha	6-4\alpha	13-8\alpha	7-4\alpha	2-2\alpha	1
24_s	1	2\alpha	3+4\alpha	5+8\alpha	2+4\alpha	6+10\alpha	3+4\alpha	3+4\alpha	2+4\alpha	5+8\alpha	3+4\alpha	2\alpha	1
30_s	1	-1+\alpha	1-\alpha	-2+\alpha	3-\alpha	1-\alpha	-2+2\alpha	-2+2\alpha	1-\alpha	3-\alpha	1-\alpha	-1+\alpha	1
30_s	1	-\alpha	\alpha	-1-\alpha	2+\alpha	\alpha	-2\alpha	-2\alpha	\alpha	2+\alpha	-\alpha	-\alpha	1
40_r	1	2	3	1	0	-2	-1	-1	0	1	3	2	1
48_{rr}	2	0	2	-2	0	0	0	0	0	2	-2	0	-2

Left cells with 392 elements														
10_r	1	0	-1	1	0	0	-1	0	1	0	1	-1	0	1
16_t	1	-1+\alpha	-\alpha	1-\alpha	1	-1	0	-\alpha	0	1	-1	\alpha	-1+\alpha	1-\alpha
16_t	1	-\alpha	-1+\alpha	\alpha	1	-1	0	-1+\alpha	0	1	-1	1-\alpha	-\alpha	-1
18_r	1	0	1	-1	0	0	1	0	1	0	-1	1	0	1
24_t	1	3-2\alpha	1-\alpha	5-3\alpha	2-\alpha	2-\alpha	0	-1+\alpha	-2+\alpha	-5+3\alpha	-3+2\alpha	-1+\alpha	-3+2\alpha	-1
$24_{t'}$	1	1+2\alpha	1+2\alpha	\alpha	2+3\alpha	1+\alpha	0	1+\alpha	-1-\alpha	-2-3\alpha	-1-2\alpha	-\alpha	-1-2\alpha	-1
24_s	1	4-3\alpha	5-3\alpha	2-\alpha	13-8\alpha	7-4\alpha	16-10\alpha	11-7\alpha	7-4\alpha	13-8\alpha	5-3\alpha	2-\alpha	4-3\alpha	1
24_s	1	1+3\alpha	2+3\alpha	1+\alpha	5+8\alpha	3+4\alpha	4+7\alpha	4+7\alpha	2\alpha	3+4\alpha	5+8\alpha	2+3\alpha	1+\alpha	1
30_s	1	1	-1	2-\alpha	-2+\alpha	1-\alpha	2-\alpha	2-2\alpha	1-\alpha	1-\alpha	-2+\alpha	-1	2-\alpha	1
30_s	1	1	-1	1+\alpha	-\alpha	\alpha	1+\alpha	1+\alpha	2\alpha	\alpha	-1-\alpha	-1	1+\alpha	1
40_r	2	1	3	2	-2	-2	-1	-1	2	-2	2	3	1	2
48_{rr}	2	1	1	1	-2	0	0	-1	0	0	0	-1	-1	-2

Left cells with 436 elements															
6_s	1	\alpha	-1-\alpha	-\alpha	1+\alpha	-1	-1	-\alpha	1+\alpha	-\alpha	-1-\alpha	-\alpha	-1	1	\alpha
6_s	1	1-\alpha	-2+\alpha	-1+\alpha	1-\alpha	-1	-1	-1+\alpha	2-\alpha	1-\alpha	1-\alpha	-2+\alpha	-1+\alpha	-1	1-\alpha
16_t	1	\alpha	-\alpha	-1	-1+\alpha	1	1-\alpha	0	0	0	-1	1-\alpha	\alpha	1	-1+\alpha
16_t	1	1-\alpha	-1+\alpha	-1	-\alpha	1	0	0	0	0	-1	1-\alpha	1	1	-\alpha
24_t	1	1-\alpha	3-2\alpha	2-\alpha	3-2\alpha	5-3\alpha	1-\alpha	0	0	0	-5+3\alpha	-3+2\alpha	-2+\alpha	-2+\alpha	-1-\alpha
$24_{t'}$	1	\alpha	1+2\alpha	1+\alpha	1+\alpha	2+3\alpha	\alpha	0	0	0	-2-3\alpha	-1-2\alpha	-1-\alpha	-1-\alpha	-\alpha
24_s	1	1-\alpha	5-3\alpha	3-2\alpha	8-5\alpha	13-8\alpha	2-\alpha	6-4\alpha	10-6\alpha	4-2\alpha	13-8\alpha	8-5\alpha	5-3\alpha	3-2\alpha	2-\alpha
24_s	1	\alpha	2+3\alpha	1+2\alpha	3+5\alpha	5+8\alpha	1+\alpha	4+6\alpha	2+2\alpha	2+2\alpha	5+8\alpha	3+5\alpha	2+3\alpha	1+2\alpha	1+\alpha
30_s	2	1	1-\alpha	0	2-\alpha	-4+2\alpha	1-\alpha	4-3\alpha	1-\alpha	1-\alpha	3-2\alpha	-4+2\alpha	2-\alpha	1-\alpha	2
30_s	2	1	\alpha	0	1+\alpha	-2-2\alpha	\alpha	1+3\alpha	\alpha	1+2\alpha	-\alpha	1+\alpha	0	0	1
40_r	2	1	3	0	0	0	-2	0	-2	2	2	3	0	0	2
48_{rr}	2	1	1	0	0	0	0	0	0	0	0	-1	0	0	2
48_{rr}	2	1	1	0	0	0	0	0	0	0	0	-1	0	0	-2

EXAMPLE 5.10. Let W be of type $I_2(m)$ where $m \geq 3$ and $S = \{s_1, s_2\}$. Assume that we are in the equal parameter case, where $\Gamma = \mathbb{Z}$ and $L(s) = 1$ for all $s \in S$. Let $\zeta \in \mathbb{C}$ be a root of unity of order m , chosen such that $\zeta + \zeta^{-1} = 2 \cos(2\pi/m)$. By [15, Section 5.4], we have

$$\text{Irr}(W) = \begin{cases} \{1_W, \text{sgn}, \sigma_1, \sigma_2, \dots, \sigma_{(m-1)/2}\} & \text{if } m \text{ is odd,} \\ \{1_W, \text{sgn}, \sigma_1, \sigma_2, \dots, \sigma_{(m-2)/2}, \text{sgn}_1, \text{sgn}_2\} & \text{if } m \text{ is even,} \end{cases}$$

where 1_W is the unit and sgn is the sign representation, all σ_j are 2-dimensional, and $\text{sgn}_1, \text{sgn}_2$ are two further 1-dimensional representations when m is even, in which case we fix the notation such that s_1 acts as $+1$ in sgn_1 and as -1 in sgn_2 . The left cells and the corresponding left cell modules are given as follows (see, for example, [14, 2.1.8, 2.2.8]):

$$\begin{aligned} & \{1_0\}, \{1_m\}, \{2_1, 1_2, 2_3, \dots, 1_{m-1}\}, \{1_1, 2_2, 1_3, \dots, 2_{m-1}\} \quad (m \text{ odd}) \\ & \{1_0\}, \{1_m\}, \{2_1, 1_2, 2_3, \dots, 2_{m-1}\}, \{1_1, 2_2, 1_3, \dots, 1_{m-1}\} \quad (m \text{ even}). \end{aligned}$$

Here, for any $k \geq 0$, we write $1_k = s_1 s_2 s_1 \dots$ (k factors) and $2_k = s_2 s_1 s_2 \dots$ (k factors); note that $1_m = 2_m$. We have:

$$\begin{aligned} [1_0]_1 &= 1_W, & [2_1, 1_2, 2_3, \dots, 2_{m-1}]_1 &= (\text{sgn}_1) \oplus (\text{sum of all } \sigma_j), \\ [1_1, 2_2, 1_3, \dots, 1_{m-1}]_1 &= (\text{sgn}_2) \oplus (\text{sum of all } \sigma_j), & [1_m]_1 &= \text{sgn}, \end{aligned}$$

where sgn_1 and sgn_2 have to be omitted if m is odd. (Note that [14, 2.2.8] contains a misprint: the roles of $\text{sgn}_1, \text{sgn}_2$ need to be changed there.) By [14, Example 1.3.7], we have $\mathbf{a}_{1_W} = 0$ and $\mathbf{a}_{\text{sgn}} = m$; all the other irreducible representations have \mathbf{a} -invariant equal to 1. First of all, one easily checks that

$$c_{w,1_W}^* = \begin{cases} 1 & \text{if } w = 1 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad c_{w,\text{sgn}}^* = \begin{cases} 1 & \text{if } w = w_0, \\ 0 & \text{otherwise,} \end{cases}$$

where $w_0 \in W$ is the longest element. Next consider sgn_1 and sgn_2 (in case m is even). Let $w \in W$. For $i = 1, 2$ we denote by $l_i(w)$ the number of occurrences of the generator s_i in a reduced expression for w . Then

$$\text{trace}(\tilde{T}_w, \text{sgn}_1) = (-1)^{l_2(w)} \varepsilon^{l_1(w) - l_2(w)}$$

and so

$$c_{w,\text{sgn}_1}^* = \begin{cases} -(-1)^{l_2(w)} & \text{if } l_1(w) - l_2(w) = -1, \\ 0 & \text{otherwise.} \end{cases}$$

A similar formula holds for c_{w,sgn_2}^* where the roles of $l_1(w)$ and $l_2(w)$ need to be interchanged. Finally, consider σ_j . By [15, Lemma 8.3.3], we have

$$\text{trace}(\tilde{T}_{s_i}, \sigma_j) = \varepsilon - \varepsilon^{-1} \quad \text{and} \quad \text{trace}(\tilde{T}_{w_k}, \sigma_j) = \zeta^{jk} + \zeta^{-jk}$$

where $w_k = (s_1 s_2)^k$ for $0 \leq k \leq m/2$. In particular, we see that

$$c_{s_1, \sigma_j}^* = c_{s_2, \sigma_j}^* = 1 \quad \text{and} \quad c_{w_k, \sigma_j}^* = 0 \quad \text{for all } 0 \leq k \leq m/2.$$

Let $y \in W$ be a conjugate of s_1 or s_2 . Then $l(y)$ is odd and we write $l(y) = 2k + 1$ where $k \geq 0$. Assume that $k \geq 2$ and let $i \in \{1, 2\}$ be such that $y' = s_i y s_i < y$. Then $s_i y$ or $y s_i$ equals w_k . So we have

$$\text{trace}(\tilde{T}_y, \sigma_j) = \text{trace}(\tilde{T}_{y'}, \sigma_j) + (\varepsilon - \varepsilon^{-1}) \text{trace}(\tilde{T}_{w_k}, \sigma_j).$$

Since $\mathbf{a}_{\sigma_j} = 1$, this yields that $c_{y, \sigma_j} = c_{y', \sigma_j} + (\zeta^{jk} + \zeta^{-jk})$. Thus, we have

$$c_{y, \sigma_j}^* = c_{y, \sigma_j} = 1 + \sum_{1 \leq i \leq k} (\zeta^{ji} + \zeta^{-ji}).$$

For example, for $m = 5$, we obtain for the two left cells with $m - 1 = 4$ elements:

$$\mathfrak{X}(W | \mathfrak{C}) : \begin{matrix} \sigma_1 & 1 & \alpha \\ \sigma_2 & 1 & 1 - \alpha \end{matrix} \quad \text{where } \alpha = \frac{1}{2}(1 + \sqrt{5}).$$

Having computed all the leading coefficients for W , we also see that

$$\tilde{\mathcal{D}} = \{1, s_1, s_2, w_0\}.$$

To conclude, let $E \in \text{Irr}(W)$ be special, that is, $E \in \{1_W, \text{sgn}, \sigma_1\}$. By the above computations, we see that $c_{w,E}^* \geq 0$ for all $w \in W$; note also that $c_{y,\sigma_1}^* > 0$ where $l(y) = 2k + 1$ and $1 \leq k \leq m/2 - 1$. Using this property and the explicit description of the left cells, we deduce that $c_{w,E}^* > 0$ for all $w \in \mathfrak{C} \cap \mathfrak{C}^{-1}$ where \mathfrak{C} is a left cell with $m(\mathfrak{C}, E) > 0$.

CONJECTURE 5.11. Assume that Conjecture 5.2 holds for W, L and define

$$\mathcal{S}_L(W) := \{E \in \text{Irr}(W) \mid c_{w,E}^* \geq 0 \text{ for all } w \in W\}.$$

Then, for each left cell \mathfrak{C} of W , there is a unique $E \in \mathcal{S}_L(W)$ such that $m(\mathfrak{C}, E) > 0$; furthermore, for this E , we have $m(\mathfrak{C}, E) = 1$ and $c_{w,E}^* > 0$ for all $w \in \mathfrak{C} \cap \mathfrak{C}^{-1}$.

REMARK 5.12. Let W be a finite Coxeter group and assume that we are in the equal parameter case where $\Gamma = \mathbb{Z}$ and $L(s) = 1$ for all $s \in S$. Then the above conjecture holds where $\mathcal{S}_L(W)$ consists precisely of the ‘special’ representations as originally defined by Lusztig [21].

Indeed, by standard reduction arguments, we can assume that W is irreducible. If W is a finite Weyl group, the assertion holds by the results of Lusztig [23, 26], as already discussed in Example 5.8. If W is of type $I_2(m)$, H_3 or H_4 , then the required assertions are verified by inspection using the data in Examples 5.9 and 5.10.

REMARK 5.13. Assume that Conjectures 5.2 and 5.11 hold for W, L . Then we have

$$\sum_{E \in \mathcal{S}_L(W)} \dim E = \text{number of left cells of } W \text{ (with respect to } L).$$

Proof. We consider the quantity

$$\nu = \sum_{\mathfrak{C}} \sum_{E \in \mathcal{S}_L(E)} m(\mathfrak{C}, E)$$

where the first sum runs over all left cells of W . Since the direct sum of all left cell modules $[\mathfrak{C}]_1$ is isomorphic to the regular representation of W , we have

$$\dim E = \sum_{\mathfrak{C}} m(\mathfrak{C}, E) \quad \text{for every } E \in \text{Irr}(W).$$

This shows that $\nu = \sum_{E \in \mathcal{S}_L(W)} \dim E$. On the other hand, by Conjecture 5.11, we have

$$1 = \sum_{E \in \mathcal{S}_L(W)} m(\mathfrak{C}, E) \quad \text{for each left cell } \mathfrak{C}.$$

So ν equals the number of left cells. This yields the desired equality. □

Let us now consider some examples with unequal parameters.

EXAMPLE 5.14. Let W be of type $I_2(m)$ where $m \geq 3$ is even and $S = \{s_1, s_2\}$. Assume that we have a weight function such that $b = L(s_1) > a = L(s_2) > 0$. The left cells and the

corresponding left cell modules are given as follows (see, for example, [14, 2.1.8, 2.2.8]):

$$\{1_0\}, \{2_1\}, \{1_{m-1}\}, \{1_m\}, \{1_1, 2_2, 1_3, \dots, 2_{m-2}\}, \{1_2, 2_3, 1_4, \dots, 2_{m-1}\}.$$

(Notation as in Example 5.10.) We have:

$$[1_0]_1 = 1_W, \quad [2_1]_1 = \text{sgn}_1, \quad [1_{m-1}]_1 = \text{sgn}_2, \quad [1_m]_1 = \text{sgn},$$

$$[1_1, 2_2, 1_3, \dots, 2_{m-2}]_1 = [1_2, 2_3, 1_4, \dots, 2_{m-1}]_1 = \text{sum of all } \sigma_j.$$

By [14, Example 1.3.7], the \mathbf{a} -invariants are given as follows:

$$\mathbf{a}_{1_W} = 0, \quad \mathbf{a}_{\text{sgn}_1} = a, \quad \mathbf{a}_{\text{sgn}_2} = \frac{m}{2}(b - a) + a, \quad \mathbf{a}_{\text{sgn}} = \frac{m}{2}(a + b), \quad \mathbf{a}_{\sigma_j} = b.$$

Arguing as in Example 5.10, we find the following leading coefficients:

$$c_{w,1_W} = \begin{cases} 1 & \text{if } w = 1_0 \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad c_{w,\text{sgn}} = \begin{cases} 1 & \text{if } w = 1_m \\ 0 & \text{otherwise.} \end{cases}$$

$$c_{w,\text{sgn}_1} = \begin{cases} 1 & \text{if } w = 2_1 \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad c_{w,\text{sgn}_2} = \begin{cases} -(-1)^{m/2} & \text{if } w = 1_{m-1} \\ 0 & \text{otherwise.} \end{cases}$$

For σ_j , we now obtain $c_{1_1,\sigma_j} = 1$, $c_{2_1,\sigma_j} = 0$ and also $c_{w_k,\sigma_j} = 0$ where $w_k = (s_1 s_2)^k$ for $0 \leq k \leq m/2$. Next, assume that $k \geq 3$ is odd; then we find the recursions

$$c_{1_k,\sigma_j} = c_{2_{k-2},\sigma_j} + (\zeta^{jk} + \zeta^{-jk}) \quad \text{and} \quad c_{2_k,\sigma_j} = c_{1_{k-2},\sigma_j}.$$

Finally, the numbers \check{n}_w have been determined in [14, 1.7.4]:

$$\check{n}_w = \begin{cases} 1 & \text{for } w \in \{1_0, 1_1, 2_1, 2_3, 1_m\}, \\ -(-1)^{m/2} & \text{for } w = 1_{m-1}, \\ 0 & \text{otherwise.} \end{cases}$$

This allows us, first of all, to verify that Conjecture 5.2 holds where

$$\tilde{\mathcal{D}} = \{1_0, 1_1, 2_1, 2_3, 1_{m-1}, 1_m\}.$$

Continuing as in Example 5.10, we conclude that Conjecture 5.11 also holds where

$$\mathcal{S}_L(W) = \{1_W, \text{sgn}_1, \text{sgn}_2, \text{sgn}, \sigma_1\}.$$

For example, for $m = 8$ and $b = 2$, $a = 1$, we obtain for the two left cells with $m - 2 = 6$ elements:

$$\mathfrak{X}(W \mid \mathfrak{C}) : \begin{array}{cccc} \sigma_1 & 1 & \sqrt{2} & 1 \\ \sigma_2 & 1 & 0 & -1 \\ \sigma_3 & 1 & -\sqrt{2} & 1. \end{array}$$

EXAMPLE 5.15. Let W be of type F_4 , with generators and diagram as in Table 1. Then a weight function L is specified by two elements $a, b \in \Gamma_{\geq 0}$ where $a = L(s_0) = L(s_1)$ and $b = L(s_2) = L(s_3)$. Let us assume that $a > 0$ and $b > 0$. (By the discussion in [14, Section 2.4], the case where $L(s) = 0$ for some $s \in S$ can always be reduced to the case where all weights are strictly positive, possibly by passing to a proper reflection subgroup of W .) By the symmetry of the diagram, we can also assume that $a \leq b$. Then, by the results in [8, Section 4], there are essentially only four cases to consider:

$$a = b, \quad b = 2a, \quad 2a > b > a, \quad b > 2a.$$

The equal parameter case is already settled by Lusztig [21]. In the remaining cases it turns out that, for every left cell \mathfrak{C} , the representation $[\mathfrak{C}]_1$ is multiplicity-free with at most 3 irreducible constituents. Using Algorithm B we have checked that Conjecture 5.11 holds where

the sets $\mathcal{S}_L(W)$ are given as follows:

$$\begin{aligned} a = b &: 1_1, 1_4, 9_1, 9_4, 12, 4_2, 4_5, 8_1, 8_2, 8_3, 8_4; \\ b = 2a &: 1_1, 1_3, 1_4, 2_2, 2_3, 2_4, 4_1, 9_1, 9_2, 9_3, 12, 4_2, 4_3, 4_4, 4_5, 8_1, 8_2, 8_4; \\ b \notin \{a, 2a\} &: 1_1, 1_2, 1_3, 1_4, 2_1, 2_2, 2_3, 2_4, 4_1, 9_1, 9_2, 9_3, 9_4, 12, 4_2, 4_3, 4_4, 4_5, 8_1, 8_2, 8_3, 8_4. \end{aligned}$$

In all cases where \mathfrak{C} has two irreducible components, the table $\mathfrak{X}(W | \mathfrak{C})$ is given by:

$$\begin{array}{ccc} E_1 & 1 & 1 \\ E_2 & 1 & -1 \end{array} \quad \text{where } E_1 \in \mathcal{S}_L(W).$$

We give one particular example where $\check{n}_d = -1$, for the case $a = 1, b = 2$: There is a left cell \mathfrak{C} such that $\mathfrak{C} \cap \mathfrak{C}^{-1} = \{d, w\}$ where

$$d = s_1 s_0 s_2 s_1 s_0 s_2 s_1 s_2 \quad \text{and} \quad w = s_1 s_2 s_1 s_0 s_2 s_1 s_2 s_3 s_2 s_1 s_0 s_2 s_1 s_2;$$

note that both $l(d)$ and $l(w)$ are even. We have $d \in \tilde{\mathcal{D}}, \check{n}_d = -1$ and

$$\mathfrak{X}(W | \mathfrak{C}) : \begin{array}{ccc} E_1 & 1 & 1 \\ E_2 & 1 & -1 \end{array} \quad \text{where } \begin{array}{l} E_1 = 4_1 \in \mathcal{S}_L(W), \\ E_2 = 16. \end{array}$$

In all cases where \mathfrak{C} has three irreducible components, the table $\mathfrak{X}(W | \mathfrak{C})$ is given by:

$$\begin{array}{ccc} E_1 & 1 & 2 & 1 \\ E_2 & 1 & -1 & 1 \\ E_3 & 1 & 0 & -1 \end{array} \quad \text{where } \begin{array}{l} E_1 = 12 \in \mathcal{S}_L(W), \\ E_2 \in \{6_1, 6_2\}, \\ E_3 = 16. \end{array}$$

We note the following special behaviour in the case where $b = 2a$. By [8, Section 4], there are three left cells $\mathfrak{C}_1, \mathfrak{C}_2, \mathfrak{C}_3$ such that

$$[\mathfrak{C}_1]_1 = 1_3 \oplus 8_3, \quad [\mathfrak{C}_2]_1 = 2_1 \oplus 9_1, \quad [\mathfrak{C}_3]_1 = 9_1 \oplus 8_3.$$

The corresponding representations in $\mathcal{S}_L(W)$ are $1_3, 9_1, 9_1$, respectively. Since we have $\text{Hom}_W([\mathfrak{C}_1]_1, [\mathfrak{C}_3]_1) \neq 0$ and $\text{Hom}_W([\mathfrak{C}_2]_1, [\mathfrak{C}_3]_1) \neq 0$, the three left cells are contained in the same two-sided cell. Thus, there are two representations in $\mathcal{S}_L(W)$ belonging to this two-sided cell. (This is not an isolated event: there are many examples in type B_n with unequal parameters as well.) This phenomenon can not happen in the equal parameter case where every two-sided cell contains a unique special representation (see Lusztig [23, Chapter 5]).

The above examples show that Conjecture 5.11 holds for W of type $I_2(m), F_4$ and any weight function L . Thus, the case that remains to be dealt with is type B_n with unequal parameters. I have checked that Conjecture 5.11 holds for type B_n where $n \in \{2, 3, 4, 5, 6\}$ and any weight function. In general, by the results in [27, Section 22], it is expected that all left cell modules $[\mathfrak{C}]_1$ in type B_n are multiplicity-free; hence, one may hope that the tables $\mathfrak{X}(W | \mathfrak{C})$ might be determined as in [26, Proposition 3.11]. If this were true, then Conjecture 5.11 would follow in this case as well.

References

1. D. ALVIS, ‘The left cells of the Coxeter group of type H_4 ’, *J. Algebra* 107 (1987) 160–168 see also, <http://mypage.iusb.edu/~dalvis/h4data>.
2. B. CASSELMAN, ‘Verifying Kottwitz’ conjecture by computer’, *Represent. Theory* 4 (2000) 32–45.
3. Y. CHEN and J.-Y. SHI, ‘Left cells in the Weyl group of type E_7 ’, *Comm. Algebra* 26 (1998) 3837–3852.
4. F. DUCLOUX, ‘The state of the art in the computation of Kazhdan–Lusztig polynomials’, *Appl. Algebra Engrg. Comm. Comput.* 7 (1996) 211–219.
5. F. DUCLOUX, ‘Coxeter: software for Kazhdan–Lusztig polynomials for Coxeter groups’; available at <http://www.liegroups.org/coxeter/coxeter3/english>.
6. F. DUCLOUX, ‘Positivity results for the Hecke algebras of noncrystallographic finite Coxeter group’, *J. Algebra* 303 (2006) 731–741.

7. M. GECK, 'On the induction of Kazhdan–Lusztig cells', *Bull. Lond. Math. Soc.* 35 (2003) 608–614.
8. M. GECK, 'Computing Kazhdan–Lusztig cells for unequal parameters', *J. Algebra* 281 (2004) 342–365.
9. M. GECK, 'Relative Kazhdan–Lusztig cells', *Represent. Theory* 10 (2006) 481–524.
10. M. GECK, 'Leading coefficients and cellular bases of Hecke algebras', *Proc. Edinb. Math. Soc.* 52 (2009) 653–677.
11. M. GECK, 'Some applications of CHEVIE to the theory of algebraic groups', *Carpathian. Math.* 27 (2011) 64–94.
12. M. GECK, 'Kazhdan–Lusztig cells and the Frobenius–Schur indicator', Preprint, 2011, arXiv:1110.5672.
13. M. GECK, G. HISS, F. LÜBECK, G. MALLE and G. PFEIFFER, 'CHEVIE—A system for computing and processing generic character tables', *Appl. Algebra Engrg. Comm. Comput.* 7 (1996) 175–210, electronically available at <http://www.math.rwth-aachen.de/~CHEVIE>.
14. M. GECK and N. JACON, *Representations of Hecke algebras at roots of unity*, Algebra and Applications 15 (Springer, New York, 2011).
15. M. GECK and G. PFEIFFER, *Characters of finite Coxeter groups and Iwahori–Hecke algebras*, London Mathematical Society Monographs, New Series 21 (Oxford University Press, Oxford, 2000).
16. R. B. HOWLETT and Y. YIN, 'Inducing W -graphs', *Math. Z.* 244 (2003) 415–431.
17. R. B. HOWLETT and Y. YIN, 'Inducing W -graphs II', *Manuscripta Math.* 115 (2004) 495–511.
18. R. B. HOWLETT and Y. YIN, 'Computational construction of irreducible W -graphs for types E_6 and E_7 ', *J. Algebra* 321 (2009) 2055–2067.
19. D. KAZHDAN and G. LUSZTIG, 'Representations of Coxeter groups and Hecke algebras', *Invent. Math.* 53 (1979) 165–184.
20. R. KOTTWITZ, 'Involutions in Weyl groups', *Represent. Theory* 4 (2000) 1–15.
21. G. LUSZTIG, 'A class of irreducible representations of a finite Weyl group', *Indag. Math.* 41 (1979) 323–335.
22. G. LUSZTIG, 'Left cells in Weyl groups', *Lie Group Representations, I*, Lecture Notes in Mathematics 1024 (eds R. L. R. Herb and J. Rosenberg; Springer, New York, 1983) 99–111.
23. G. LUSZTIG, *Characters of reductive groups over a finite field*, Annals of Mathematics Studies 107 (Princeton University Press, 1984).
24. G. LUSZTIG, 'Cells in affine Weyl groups', *Algebraic groups and related topics*, Advanced Studies in Pure Math. 6 (Kinokuniya and North–Holland, 1985) 255–287.
25. G. LUSZTIG, 'Cells in affine Weyl groups II', *J. Algebra* 109 (1987) 536–548.
26. G. LUSZTIG, *Leading coefficients of character values of Hecke algebras*, Proceedings of Symposia in Pure Mathematics, 47 (American Mathematical Society, Providence, RI, 1987) 235–262.
27. G. LUSZTIG, *Hecke algebras with unequal parameters*, CRM Monographs Series 18 (American Mathematical Society, Providence, RI, 2003).
28. G. LUSZTIG and D. A. VOGAN JR., 'Hecke algebras and involutions in Weyl groups', *Bull. Inst. Math. Acad. Sin. (N.S.)*, to appear, arXiv:1109.4606.
29. G. VAN ROSSUM *et al.*, Python Language Website, 1990–2012 (Python Software Foundation, see <http://www.python.org/>).
30. M. SCHÖNERT *et al.*, *GAP – Groups, Algorithms, and Programming*, fourth ed. (Lehrstuhl D für Mathematik, RWTH Aachen, Germany, 1994).
31. W. A. STEIN *et al.*, Sage Mathematics Software (Version 4.7.1) (The Sage Development Team, 2011, see <http://www.sagemath.org/>).
32. Y. YIN, 'An inversion formula for induced Kazhdan–Lusztig polynomials and duality of W -graphs', *Manuscripta Math.* 121 (2006) 81–96.

Meinolf Geck
 Institute of Mathematics
 University of Aberdeen
 Aberdeen AB24 3UE
 United Kingdom

m.geck@abdn.ac.uk