Computing some KL-polynomials for the poset of $B \times B$-orbits in group compactifications

Wilberd van der Kallen

October 2, 2002

In [3], Springer studies a poset V which depends on a Coxeter group W with generating set S. If W is a finite Weyl group, then V indexes the $B\times B$-orbits in the wonderful compactification of the adjoint semi-simple group G with Weyl group W. In that case Springer shows for instance that one has positivity results for the analogues $c_{x,v}$ of Kazhdan-Lusztig polynomials. But the poset V is much more complicated/larger than W, so that computing the $c_{x,v}$ by hand is only attractive when S has one element. Therefore we wrote Mathematica code to get access to some more examples. Once the code was available we could experiment with the analogues $c^{\mathrm{inv}}_{x,v}$ of inverse Kazhdan-Lusztig polynomials and also with Coxeter groups that are not finite Weyl groups. In our experiments we saw positivity properties for the $c^{\mathrm{inv}}_{x,v}$. And the other Coxeter groups behaved just like finite Weyl groups.

As already mentioned, V is much bigger than W. For instance, when W is of type B_3, the size $|V|$ of V is already 7056, while $|W|$ is only 48. Therefore we further assume W is so small that we do not have to worry about its size.

Our reason to choose Mathematica is that Mathematica provides a powerful high level language with which we are familiar. Our primary task was to code new combinatorics reliably for very small W. Speed was no issue at this prototype stage. Everything could be developed from scratch without any need for libraries of fast specialized tools. Thus the fact that Mathematica is quite ignorant about Coxeter groups was no obstacle at all.

Let us now digress and recall that Mathematica is optimized for replacement rules based on pattern matching. We did find this attractive because it allows to use very simple and transparent code for reducing words in small Coxeter groups. To illustrate, we give the complete code for reducing words to normal form in the Coxeter group (W, S) of type H_3. (We define the normal form of a group element to be the lexicographically first reduced word that represents it.)

```
s/: s::usage := "s[1,2] is the product of reflections 1 and 2"

reduce[s[a___, c_, c_, b___]] := reduce[s[a, b]]

(* The next six rules are generated by the program *)

reduce[s[a___, 3, 1, b___]] := reduce[s[a, 1, 3, b]]

reduce[s[a___, 3, 2, 3, b___]] := reduce[s[a, 2, 3, 2, b]]

reduce[s[a___, 3, 2, 1, 3, b___]] := reduce[s[a, 2, 3, 2, 1, b]]

reduce[s[a___, 2, 1, 2, 1, 2, b___]] := reduce[s[a, 1, 2, 1, 2, 1, b]]

reduce[s[a___, 3, 2, 1, 2, 3, 2, b___]] := reduce[s[a, 2, 3, 2, 1, 2, 3, b]]

reduce[s[a___, 3, 2, 1, 2, 1, 3, 2, 1, 2, 1, b___]] :=
    reduce[s[a, 2, 3, 2, 1, 2, 1, 3, 2, 1, 2, b]]

reduce[other_] := other

reduce/: reduce::usage :=
    "reduce[s[...]] gives normal form of s[...]"
```

Note that most of these rewriting rules were actually generated by the program. So we did not need to know finite automata for H_3. Mathematica even tells us the braid relations. All the other relations are also clearly correct. To get started we had to provide three matrices representing the three elements of S in a faithful representation of the group W. (Here faithful has to be taken in an algorithmic sense. Matrix coefficients have to lie in a ring in which equality is recognized.) To check that a list of rewriting rules is sufficient, one simply generates a full list of non-rewritable words and checks that it has the same size as the group W. Indeed we did this check for the above code. When the list of rewriting rules is still too small, one may discover a new rule by finding two non-rewritable words that represent the same matrix. This is how the rules were found with a systematic search. The search looked for the first non-rewritable word that is not in normal form, sorting words by length and sorting lexicographically for given length. The process stops when a length is reached for which all words are rewritable. Of course we do not recommend such a naive approach for larger Coxeter groups. This ends the digression.

Having reduction of words under control, we can implement the length function ℓ on W. Before telling more about implementing, we have to recall some conventions from [3].

Let (W, S) be an arbitrary Coxeter group (with a finite set of generators S). For $I \subset S$ let W_I be the subgroup generated by I and W^I the set of distinguished coset representatives for W_I, i.e. the set of $x \in W$ with $xs > x$ for all $s \in I$. Observe that we could write $\ell(xs) > \ell(x)$ instead of $xs > x$. In other words, the length function suffices and the partial order on W is not needed to compute W^I. A similar remark applies to the following description of the R-polynomials of [1]. (Our δ is always a Kronecker δ.)

The R-polynomials satisfy the following recursive relations (where $x, y \in W$, $s \in S$). Together with the boundary conditions $R_{y,1} = \delta_{y,1}$ these relations define the R-polynomials uniquely.

$$R_{y,sx} = \begin{cases} R_{sy,x}, & \text{if } sx > x,\ sy < y; \\ (u^2 - 1)R_{y,x} + u^2 R_{sy,x} & \text{if } sx > x,\ sy > y. \end{cases} \tag{1}$$

As we need the R-polynomials anyway, and W is small, we compute and remember all the $R_{y,x}$. Then we use them also for getting the Bruhat order on W for free. (Recall that $R_{y,x} \neq 0$ if and only if $y \leq x$.) By the way, our actual code uses a more elaborate boundary condition:

$$\begin{aligned} &R_{y,x} = 0 \quad \text{if} \quad \ell(y) > \ell(x), \\ &R_{y,y} = 1. \end{aligned} \tag{2}$$

The same strategy can be followed for the poset V. As a set, V is the set of triples $[I, x, w]$ with $I \subseteq S$, $x \in W^I$, $w \in W$. We compute V in the obvious way and remember it. We do not recall the partial order on V now, as it will 'come for free' again. The analogue of the length function ℓ on W is the 'dimension function' d on V defined by

$$d([I, x, w]) = -l(x) + l(w) + \mid I \mid .$$

It may be negative.

The analogue of $R_{x,y}$ is called $b_{w,v}$, but we will work with the Laurent polynomials $\tilde{b}_{w,v} := u^{-d(v)+d(w)} b_{w,v}$ instead. They satisfy the following boundary conditions, of which the middle one was actually discovered using an earlier version of our program.

$$
\begin{array}{rcl}
\tilde{b}_{v,v} & = & 1, \\
\tilde{b}_{[I,a,b],[J,x,1]} & = & \begin{cases} 0, & \text{if } I \not\subseteq J \text{ or } b \notin W_J, \\ (u^{-1}-u)^{|J|-|I|}(-u)^{\ell(x)+\ell(b)-\ell(a)} R_{xb,a}(u^2) & \text{else.} \end{cases} \\
\tilde{b}_{[I,a,b],[J,x,y]} & = & 0 \quad \text{if } I \not\subseteq J \text{ or } \ell(b)-\ell(a) > \ell(y)-\ell(x) \text{ or } x \not\leq a.
\end{array}
\tag{3}
$$

Together with these boundary conditions the following recursive relations define the $\tilde{b}$-polynomials uniquely (where $I, J \subseteq S$, $s \in S$, $a \in W^I$, $x \in W^J$, $b, y \in W$).

$$
\tilde{b}_{[I,a,b],[J,x,sy]} = \begin{cases} \tilde{b}_{[I,a,sb],[J,x,y]}, & \text{if } sb < b,\ sy > y; \\ (u^{-1}-u)\tilde{b}_{[I,a,b],[J,x,y]} + \tilde{b}_{[I,a,sb],[J,x,y]} & \text{if } sb > b,\ sy > y. \end{cases} \tag{4}
$$

This is how we compute the $\tilde{b}_{v,w}$. The partial order on V is then obtained from

$$
v \leq w \quad \Longleftrightarrow \quad \tilde{b}_{v,w} \neq 0. \tag{5}
$$

As V is often too big, we do not compute the $\tilde{b}_{v,w}$ with remembering. But we may choose a part of V and remember $\tilde{b}_{v,w}$ when v, w are in that part.

Springer has conjectured[1] that the following duality relation, which holds when W is a finite Weyl group, holds more generally for Coxeter groups.

$$
\sum_{v \leq z \leq w} (-1)^{d(v)-d(z)} \tilde{b}_{v,z} \tilde{b}_{z,w} = \delta_{v,w}. \tag{6}
$$

Compare with the case of R-polynomials in [1, 2.1(ii)]. One can test (6) by first computing a segment

$$
[v_0, v_1] = \{ z \in V \mid v_0 \leq z \leq v_1 \}
$$

in V and, if it is not too big, then try all v, w in the segment. Recall that by [3, 6.5] (and its proof) each segment is a subset of a known finite set. Therefore one can compute such segments even when W is infinite. (In that case we do not first compute all of V, just the known finite set.) Within the segment we would compute $\tilde{b}_{v,w}$ with remembering. So far the conjecture has held up. As a check on the code, one may also check duality on a segment of V when W is a finite Weyl group.

[1] Added in print: Now proved in preprint by Yu Chen and Matthew Dyer, On the combinatorics of $B \times B$ orbits on group compactifications, submitted to Journal of Algebra.

Now we come to the main task, which was to compute the analogues $c_{x,v}$ of the Kazhdan-Lusztig polynomials. We will work with the polynomials $\tilde{c}_{w,v}(u) := u^{d(v)-d(w)}c_{w,v}(u^{-2})$ instead and compute $c_{w,v}$ in terms of $\tilde{c}_{w,v}$. Let us first assume W is a finite Weyl group, so that we know the $\tilde{c}_{w,v}$ exist. For $v \neq w$ the polynomial $\tilde{c}_{w,v}(u)$ has no constant term, so that it equals minus the polynomial part of the Laurent polynomial $\tilde{c}_{w,v}(u^{-1}) - \tilde{c}_{w,v}(u)$. Thus the $\tilde{c}_{w,v}$ are determined by the recursive relation [3, 4.4 (15)]

$$\tilde{c}_{w,v}(u^{-1}) - \tilde{c}_{w,v}(u) = \sum_{w<y\leq v} \tilde{b}_{w,y}(u)\tilde{c}_{y,v}(u), \tag{7}$$

together with the boundary condition [3, 4.4(a)]

$$\tilde{c}_{v,v} = 1.$$

When computing the right hand side of (7) we first determine the set

$$\{ \, y \in V \mid y \leq v \, \}$$

by means of (5). If W is a finite Weyl group, then we know from [3, 4.2, 4.6] that, if $x \leq v$, then $c_{x,v}$ is a polynomial with constant term one and with positive coefficients. Even if W is no finite Weyl group, we compute $\tilde{c}_{w,v}$ for $w < v$ as the polynomial part of the right hand side of (7). But then we should check that this actually solves (7), as the existence of solutions is not given.

We computed the $c_{x,v}$ for all x, v when W is an irreducible finite Weyl group of rank two, and for $v = B = [D, 1, 1]$ when W is irreducible of rank three. We also explored segments $[v_0, v_1]$ for affine Weyl groups and a few more Coxeter groups that are not finite Weyl groups. We always found that, if $x \leq v$, then $c_{x,v}$ is a polynomial with constant term one and with positive coefficients.

If the $c_{x,v}$ exist and duality (6) holds, as it does when W is a finite Weyl group, then one also has analogues $c^{\mathrm{inv}}_{x,v}$ of the inverse Kazhdan-Lusztig polynomials $Q_{a,b}$. We write $\tilde{c}^{\mathrm{inv}}_{x,v}(u)$ for $u^{d(v)-d(w)}c^{\mathrm{inv}}_{w,v}(u^{-2})$. For $v \neq w$ the polynomial $\tilde{c}^{\mathrm{inv}}_{x,v}$ has no constant term, so that it equals minus the polynomial part of the Laurent polynomial $\tilde{c}^{\mathrm{inv}}_{w,v}(u^{-1}) - \tilde{c}^{\mathrm{inv}}_{w,v}(u)$. Thus the $\tilde{c}^{\mathrm{inv}}_{w,v}$ are determined by the recursive relation

$$\tilde{c}^{\mathrm{inv}}_{w,v}(u^{-1}) - \tilde{c}^{\mathrm{inv}}_{w,v}(u) = \sum_{w\leq y<v} \tilde{c}^{\mathrm{inv}}_{w,y}(u)\tilde{b}_{y,v}(u), \tag{8}$$

compare [2, 3(f)], together with the boundary condition

$$\tilde{c}^{\mathrm{inv}}_{v,v} = 1.$$

We always found that, if $x \leq v$, the $c^{\mathrm{inv}}_{x,v}(u)$ are polynomials with constant term one and with positive coefficients. (Clearly they vanish when $x \not\leq v$.) The coefficients of the $c^{\mathrm{inv}}_{x,v}(u)$ may range over a different interval of integers than the coefficients of the $c_{y,w}(u)$. For instance, for the dihedral group W of order $|W| = 14$ the nonzero terms in the $c_{y,w}(u)$ fill the set $\{1, q, 2q, 3q, 4q, 5q, 6q, 7q, 8q, 9q, 10q, q^2, 2q^2, 3q^2, 4q^2, 5q^2, 7q^2, 9q^2, q^3, 2q^3, 3q^3, q^4, 2q^4, q^5\}$, where q means u^2, but the terms of the nonzero $c^{\mathrm{inv}}_{x,v}(u)$ fill only $\{1, q, 2q, q^2, 2q^2, q^3, 2q^3, q^4, 2q^4, q^5\}$.

The latest Mathematica files are available on our web site. See
`http://www.math.uu.nl/people/vdkallen/kallen.html`
There one also finds some output, most of it in Mathematica InputForm, some of it in PostScript.

References

[1] D. Kazhdan, G. Lusztig, Representations of Coxeter groups and Hecke algebras. Invent. Math. 53 (1979), 165–184.

[2] G. Lusztig, Nonlocal finiteness of a W-graph. Represent. Theory 1 (1997).

[3] T. A. Springer, Intersection cohomology of $B \times B$-orbits in group compactifications, to appear in Journal of Algebra