PRESENTATION ON BÉDARD

ELI GARCIA

1. Root systems

We begin with *root systems*, which are special sets of vectors. A *Weyl group*, a type of finite Coxeter group, is a special subgroup of the isometries of a root system. We will focus on the root system whose Weyl group is the symmetric group. First, let E be the subspace of $\mathbb{R}^{n+1}$ containing vectors whose entries sum to zero. (This subspace is n-dimensional.) For example:

$$(2/3, -2/3, 0), (-5, 6, -1), (\sqrt{3}, \sqrt{6}, -\sqrt{3} - \sqrt{6}) \in E \text{ for } n = 2$$
$$(1, 1, -1, -1), (0, 3, 2.1, -5.1), (10\sqrt{5}, 0, 6\sqrt{5}, -16\sqrt{5}) \in E \text{ for } n = 3.$$

Define R to be the subset of E with integer coordinates and length $\sqrt{2}$. We call R the root system A_n. For example:

$$(1, -1, 0), (0, -1, 1), (-1, 0, 1) \in R \text{ for } n = 2$$
$$(1, 0, 0, 0, -1), (-1, 1, 0, 0, 0), (0, 0, -1, 0, 1) \in R \text{ for } n = 4.$$

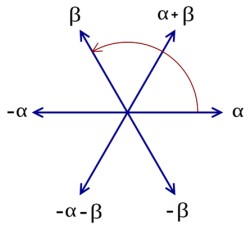

Figure 1. The root system A_2.

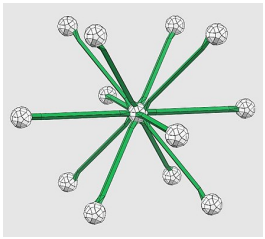

Figure 2. The root system A_3.

We can explicitly describe $\alpha \in R$: since its components sum to zero, are integers, and $|\alpha| = \sqrt{2}$, one component is 1, one component is -1, and the rest are zero. For instance:

$$A_2 = \{(1,-1,0),(-1,1,0),(1,0,-1),(-1,0,1),(0,1,-1),(0,-1,1)\}.$$

Thus we have $|R| = n(n+1)$. We define the *simple roots*

$$\begin{aligned}\alpha_1 &= (1,-1,0,\dots,0)\\ \alpha_2 &= (0,1,-1,0,\dots,0)\\ &\vdots\\ \alpha_n &= (0,\dots,1,-1).\end{aligned}$$

Each simple root corresponds to a *simple reflection*, which is reflection through the hyperplane perpendicular to the corresponding root. The root α_i corresponds to the reflection s_i swapping the ith and $i+1$th coordinates of $x \in \mathbb{R}^{n+1}$. As we know, these reflections generate the full symmetric group S_{n+1}. For this reason, $W = S_{n+1}$ is the Weyl group of the root system A_n. These simple roots can simplify (!) our description of R. We have

$$R = \{\pm(\alpha_i + \cdots + \alpha_j) \mid 1 \le i \le j \le n\}.$$

Naturally, we call the set

$$R^+ = \{(\alpha_i + \cdots + \alpha_j) \mid 1 \le i \le j \le n\}$$

the set of *positive roots*. Some examples and non-examples:

$$\begin{aligned}\alpha_2 &\in R^+\\ \alpha_1+\alpha_2+\alpha_3 &\in R^+\\ \alpha_3+\alpha_4 &\in R^+\\ \alpha_1+\alpha_5 &\notin R^+\\ -\alpha_3 &\notin R^+\\ \alpha_2-\alpha_3+\alpha_4 &\notin R^+.\end{aligned}$$

This description of R^+ can be visually described using "staircase" diagrams, where the dot in the ith column and jth row represents the positive root $\alpha_i+\cdots+\alpha_j$. It has the additional benefit that two distinct roots are in the same row or column if and only if their scalar product is 1.

FIGURE 3. Positive roots of A_5

2. Nature of the problem

We now build toward a recursive method to count the number of elements in a *commutation class* of a *reduced word*. The basic idea of where the recursive formula comes from is helpful for motivating the technical details. We start with a non-identity element $w \in W$ and a reduced word $\mathbf{i} = s_{i_1} s_{i_2} \cdots s_{i_l}$ for w. Denote the commutation class of $\mathbf{i}$ by $[\mathbf{i}]$. Recall that a commutation move is a use of the relation

$$s_{i_1} \cdots s_{i_j} s_{i_{j+1}} \cdots s_{i_l} = s_{i_1} \cdots s_{i_{j+1}} s_{i_j} \cdots s_{i_l},$$

where $|i_j - i_{j+1}| \geq 2$. Now for each $k = 1, \ldots, n$, we denote the elements of $[\mathbf{i}]$ whose final letter is s_k by I_k. Then

$$[\mathbf{i}] = \bigsqcup_{1 \leq k \leq n} I_k.$$

So we can break this down into a recursive problem:

(1) For each k, check whether there exists a word $\mathbf{j}_k \in [\mathbf{i}]$ ending with s_k.
(2) If so, consider the word $\mathbf{j}'_k$ obtained by removing s_k from the end of $\mathbf{j}_k$.
(3) Count the elements of $[\mathbf{j}'_k]$.

We come up with the formula

$$[\mathbf{i}] = \bigsqcup_{1 \leq k \leq n} [\mathbf{j}'_k].$$

This explains the recursive nature of the commutation class counting problem, but in order to obtain a convenient method of computing the size of $[\mathbf{i}]$, we return to thinking about positive roots.

3. Position and level functions

We now describe a way to associate a positive root with each prefix of $\mathbf{i}$. For $1 \leq k \leq l$, define

$$\alpha^{(k)} = s_{i_1} s_{i_2} \cdots s_{i_{k-1}} (\alpha_{i_k}).$$

We call these *relevant roots* and denote them

$$R(w) = \{\alpha^{(1)}, \ldots, \alpha^{(l)}\}.$$

As it turns out, the $\alpha^{(k)}$ are distinct and an equivalent description of $R(w)$ is

$$R(w) = \{\alpha \in R^+ \mid w^{-1}(\alpha) \in R^-\}.$$

Now we define the *position function*

$$\begin{aligned} \pi_{\mathbf{i}} \colon R(w) &\to \mathbb{N} \\ \alpha^{(k)} &\mapsto k. \end{aligned}$$

To represent $\pi_{\mathbf{i}}$ visually, we use a staircase diagram where each root is replaced with its image. For example, with $\mathbf{i} = s_1 s_4 s_3 s_4 s_2 s_3 s_4 s_1 s_2 s_3$, which is a reduced word for $w_0 \in S_5$, this looks like:

$$\pi_{\mathbf{i}} = \begin{array}{cccc} 1 & & & \\ 7 & 10 & & \\ 6 & 9 & 4 & \\ 5 & 8 & 3 & 2 \end{array}$$

Figure 4. An example $\pi_{\mathbf{i}}$ function.

We further define the *level function* $\lambda_{\mathbf{i}}\colon R(w) \to \mathbb{N}$ based on $\pi_{\mathbf{i}}$ using the following rules:

(1) Go through each root $\alpha \in R(w)$ in order of their $\pi_{\mathbf{i}}$ value.
(2) If $\lambda_{\mathbf{i}}$ is not defined for any roots sharing a row or column with α, then set $\lambda_{\mathbf{i}}(\alpha) = 1$.
(3) Otherwise, set $\lambda_{\mathbf{i}}(\alpha) = 1 + M$, where M is the largest defined $\lambda_{\mathbf{i}}$-value sharing a row or column with α.

4. Counting a commutation class

Whew! The hard part is done. Just one more definition: a *top root* is a root whose $\lambda_{\mathbf{i}}$-value is the largest in its row *and* its column. The recursive process works like this:

(1) Identify the top roots and try removing each of them, one at a time.
(2) Keep track of the resulting configurations. (Make note of how many of each kind we get.)
(3) Repeat!

We work through this process with our example word.

5. Remark on bijection

The recursive formula described above is the most relevant and accessible result from Bédard's paper. However, we have built up the necessary background to state another interesting result about commutation classes. The map ϕ that sends the commutation class $[\mathbf{i}]$ to the function $\lambda_{\mathbf{i}}$ is injective. This comes from the fact that two reduced words $\mathbf{i}$ and $\mathbf{j}$ are commutation equivalent if and only if $\lambda_{\mathbf{i}} = \lambda_{\mathbf{j}}$, which has a fairly complicated proof. Injectivity means that if we restrict the codomain of ϕ to its image, we get a bijection

$$\phi\colon \{[\mathbf{i}] \mid \mathbf{i} \text{ is a reduced word for } w\} \to F_w,$$

where F_w is the image of ϕ. In fact, F_w admits a concrete description that is unwieldy, but may be useful.