AUTOMATA TO PERFORM BASIC CALCULATIONS IN COXETER GROUPS

W. A. Casselman

Abstract. Algorithms are exhibited for constructing automata to perform efficient calculations in Coxter groups.

In the paper [Casselman 1994] I explained in a rough fashion how to build automata for recognizing various languages of strings in the Coxeter generators of finite and affine Weyl groups, in effect showing how to implement a result of [Brink-Howlett 1993] in a practical way. I also sketched a procedure to multiply elements of a Weyl group by a generator, which is related closely to the multiplier automata of Brink and Howlett.

In this paper I will give more details of these algorithms, and indeed for arbitrary Coxeter groups, following Brink and Howlett even more closely in using their notion of minimal roots. In fact, much of what I will cover is at least implicit in the work of Brink and Howlett. What is essentially new is the formulation of their material in geometrical terms, along the lines of my earlier paper. On the one hand this may make some of their arguments clearer. On the other, as I pointed out in that paper, the use of saturation in building recognizing automata, which is justified by geometry, renders the construction practical. Also, I will explain fine details of algorithms which are perhaps rather insignificant from a mathematical standpoint, but which are required in order to make them efficient.

All algorithms will depend on a simple geometrical realization of a Coxeter group. The group will stabilize a certain convex cone in the vector space of the realization, called the **Tits cone**. The realizations considered here are those in which the fundamental domain of the action of the group on this cone is simplicial. The **simple roots** are linear functions which are non-negative on the fundamental domain and vanish on the walls of this domain. The **roots** of the realization are the transforms of the simple roots under the group. The **positive roots** are those roots which are non-negative on the fundamental domain. What I call the **minimal roots** of the realization are, roughly speaking, those positive roots which are not screened off in the Tits cone from the fundamental domain by other root hyperplanes. Although the notion of minimal roots is rather natural from a geometrical standpoint, it was apparently first introduced in [Brink-Howlett 1993]. (These roots were given no name there, but in Brink's thesis [Brink 1994] they are called **elementary** roots.) The main result of Brink and Howlett is that there are only a finite number of them.

1991 *Mathematics Subject Classification.* Primary 20F55..
The research described in this paper was supported by a grant from NSERC.

It seems an interesting question as to whether or not minimal roots are generally significant in the structure of the groups attached to Kac-Moody algebras, as those for the affine Weyl groups are for loop groups.

There are two ways in which the minimal roots can be used for calculations with Coxeter groups. (1) The `ShortLex` language of strings in the Coxeter generators is that of reduced expressions which are lexicographically minimal. Every element of the group can be labelled uniquely by such an expression. A finite automaton can be constructed to recognize this language whose nodes are subsets of the minimal roots, and whose transitions are also characterized in terms of these subsets. (2) One crucial property of the minimal roots is this: If s is a defining generator of the Coxeter group and λ is a minimal root, then exactly one of these three cases occurs: (a) $s\lambda = -\lambda$, in which case λ is the simple root corresponding to s; (b) $s\lambda$ is again a minimal root, in which case s and the reflection s_λ through the hyperplane $\lambda = 0$ generate a finite group; (c) $s\lambda$ is not minimal, in which case s and s_λ generate an infinite group. We can therefore make up what I call the **minimal root reflection table**, whose rows are simple reflections, whose columns are minimal roots, and whose entry for (α, λ) is $s_\alpha\lambda$ if it is a minimal root, $-$ if $\lambda = \alpha$, and $+$ otherwise.

These two items—the `ShortLex` automaton and the reflection table—are all that is needed for basic calculations in Coxeter groups. The automaton can be used to generate the elements of the group in a stream, for example, and the reflection table can be used for multiplication in the group. There are several ways in which algorithms using these structures are an improvement on previous ones. $\diamond$ In many circumstances they are perhaps as efficient as one can imagine any algorithm being. $\diamond$ They allow an absolutely uniform approach to all Coxeter groups. $\diamond$ As Fokko du Cloux pointed out to me, no inequality testing and no floating point calculations are necessary, even for non-crystallographic groups. All that is needed is exact arithmetic in the ring generated by the coefficients of the Cartan matrix, which can be chosen to be cyclotomic integers. $\diamond$ For many calculations, complexity does not depend on rank, once the initial structures are built. The initial structures themselves can be stored in a file so that they need by calculated only once for each Coxeter group, although in practice this does not seem to be important.

In this paper I will explain how to

(1) list the minimal roots and build the reflection table;
(2) construct the `ShortLex` recognizing automaton from subsets of minimal roots;
(3) find the `ShortLex` expression for ws, where w is an element of the group whose `ShortLex` word is known and s is one of the Coxeter generators.

At the end I shall look at an example derived from ideas of Lusztig and Bédard.

I wish to thank Joel Feldman as well as the Institute for Advanced Study for making possible a month free of other responsibilities so that I could finish this paper. I also wish to thank Fokko du Cloux for helpful suggestions.

1. Geometric realizations of Coxeter groups

I recall here results which are well known, and can be found in the standard references [Bourbaki 1968], [Humphreys 1990], and [Vinberg 1971].

In this paper, a **Cartan matrix** will be a square matrix with real coefficients $c_{s,t}$, its rows and columns both indexed by elements of some finite set S, satisfying the following conditions.

(1) The diagonal elements $c_{s,s}$ are all equal to 2;
(2) For $s \neq t$, $c_{s,t} \leq 0$;
(3) For $s \neq t$, let $n_{s,t} = c_{s,t}c_{t,s}$. According to (2), it is a non-negative number. Then (a) if $n_{s,t} = 0$, both $c_{s,t}$ and $c_{t,s}$ are also 0; (b) if $0 < n_{s,t} < 4$ we must have

$$n_{s,t} = 4\cos^2(\pi/m_{s,t})$$

for some integer $m_{s,t} \geq 2$.

In case 3(a) define $m_{s,t}$ to be 2; in the case when $n_{s,t} \geq 4$, and there is no further condition on $c_{s,t}$ and $c_{t,s}$, define $m_{s,t}$ to be ∞; define $m_{s,s}$ to be 1 for all s.

The integer matrix $(m_{s,t})$ is then a **Coxeter matrix**. Let (W, S) be the Coxeter group determined by it, that is to say the group with generators in S and relations $(st)^{m_{s,t}} = 1$ whenever $m_{s,t}$ is finite. In particular $s^2 = 1$ for each s in S, and s and t commute when $m_{s,t} = 2$. Different Cartan matrices may very well be associated with the same Coxeter group.

Every Coxeter group arises from at least one Cartan matrix, called the standard one (see §V.3 of [Humphreys 1990] or §V.4 of [Bourbaki 1968]). Here

$$c_{s,t} = -2\cos(\pi/m_{s,t})$$

for all s, t in S.

Let V be a real vector space, and assume given a basis $\Delta = (\alpha_s)$ of its linear dual V^* indexed by elements of S. For each t in S there will exist a unique element $\alpha_t^\vee$ in V such that $\langle \alpha_s, \alpha_t^\vee \rangle = c_{s,t}$ for all s in S. Since $\langle \alpha_s, \alpha_s^\vee \rangle = 2$, the linear transformation

$$\sigma_s(v) = v - \langle \alpha_s, v \rangle \alpha_s^\vee$$

is then a reflection in the hyperplane $\langle \alpha_s, v \rangle = 0$ taking $\alpha_s^\vee$ to $-\alpha_s^\vee$.

Suppose that s and t are distinct in S. If $m_{s,t}$ is equal to 2, which means that $c_{s,t}$ and $c_{t,s}$ both vanish, then the reflections σ_s and σ_t commute. Otherwise, if $m_{s,t}$ is finite then it is simple to see that there exists a unique bilinear form on the space spanned by $\alpha_s^\vee$ and $\alpha_t^\vee$ with respect to which σ_s and σ_t are orthogonal, and that this form is positive definite. From this it is an easy step to see that Figure 2 in §4.5 of [Bourbaki 1968] remains valid, and that $\sigma_s\sigma_t$ amounts to rotation through the angle $2\pi/m_{s,t}$. In particular, it has order $m_{s,t}$. If $m_{s,t}$ is infinite then $\sigma_s\sigma_t$ has infinite order. In other words, the map taking s to σ_s extends to a representation of W on V.

All of the arguments in §4.5 of [Bourbaki 1968] indeed remain valid, so that among other things this representation is faithful. I shall call it, together with the particular choice of basis Δ for V^*, a **geometric realization** of (W, S). There are in general many different representations of a Coxeter group which represent the generators by reflections; the distinguishing characteristic of the ones we are looking at, which are associated directly to a Cartan matrix, is that the elements of Δ form a basis of V^*.

From now on in this paper, fix a Coxeter group (W, S) and a realization of it, and identify W with its image under σ.

Define C to be the simplicial cone of v in V where $\langle\alpha, v\rangle > 0$ for all α in Δ. Let $\mathcal{C}^*$ be the union of the transforms of $\overline{C}$ under W, and $\mathcal{C}$ its interior. The set $\mathcal{C}^*$ is called the **Tits cone**, $\mathcal{C}$ the open Tits cone. The group W acts discretely on $\mathcal{C}^*$, and $\overline{C}$ is a fundamental domain for this action. More precisely, every element of $\mathcal{C}^*$ is W-equivalent to exactly one point of $\overline{C}$. If X is any subset of S, let

$$C_X = \{v \in V \mid \langle\alpha, v\rangle = 0\ (\alpha \in X),\ \langle\alpha, v\rangle > 0\ (\alpha \in \Delta - X)\}$$

The C_X are the faces of $\overline{C}$, all elements of the group W_X generated by the s in X fix all elements of C_X, and if w in W is such that $wC_X \cap C_X \neq \emptyset$ then w lies in W_X. The group W is finite if and only if $\mathcal{C}$ and $\mathcal{C}^*$ are all of V, which is to say that the origin lies in $\mathcal{C}$. From this it is simple to see that the group W_X is finite if and only if the face C_X is contained in $\mathcal{C}$.

The **open chambers** of $\mathcal{C}$ are the W-transforms of C, and the **closed** chambers those of $\overline{C}$. Any face of a chamber is the W-transform of a unique chamber C_X of $\overline{C}$, and this subset X is called its **type**. I will call these faces the **Coxeter faces**. The closed chambers make up a triangulation of $\mathcal{C}^*$, and the transforms of the walls C_X with W_X finite triangulate $\mathcal{C}$. I will say that two chambers are **neighbours** if they share a single wall of codimension one, and I define a **Coxeter path** to be a sequence of neighbouring chambers. (This is called an injective gallery in the exercises to Chapter IV.2 of [Bourbaki 1968].) Any Coxeter path $C_0C_1\ldots C_n$ corresponds to a sequence of generators s_i in S, if we take s_i to be the reflection corresponding to the type of the wall between C_{i-1} and C_i. Conversely, given a chamber C_0 and a sequence of elements s_i in S we can construct an associated Coxeter path beginning with C_0. If $w_i = s_1s_2\ldots s_i$ for each i and $C_0 = wC$ then $C_i = ww_iC$, and in particular C_n depends on w_n and not on the particular expression of w_n as a product of elements in S. In other words, in addition to the left action of W on the vector space V, we have a right action of W on the set of chambers. Another formulation is to say that the configuration of chambers and walls in $\mathcal{C}$ is dual to the Cayley graph of (W, S).

I define a **Coxeter geodesic** between two subsets of $\mathcal{C}$ to be a Coxeter path of minimal length between chambers in each. I call this minimal length (that is to say, one less than the number of chambers in the sequence) the **Coxeter distance** between the two sets. The length of an element w, for example, is the length of a Coxeter geodesic between $\overline{C}$ and $w\overline{C}$.

I define the **roots** Σ of the realization to be the Coxeter transforms in V^* of the elements of Δ. If λ is a root then either λ is positive throughout C or it is negative throughout. In the first case it is said to be a positive root, and the expression of λ as a linear combination of elements of Δ has only non-negative coefficients. In the second case it is said to be a negative root.

Let w be an element of W and suppose given a Coxeter path from C to wC. Any root hyperplane separating C from wC must contain one of the walls between successive chambers in the path. In other words, if I define for each w in W

$$\mathcal{L}_w = \{\lambda \in \Sigma \mid \lambda > 0,\ w^{-1}\lambda < 0\}$$

then each root hyperplane $\lambda = 0$ for λ in $\mathcal{L}_w$ separates C from wC, and conversely if the path is geodesic. The set $\mathcal{L}_w$ is finite, and its cardinality is also the length $\ell(w)$, the number of terms in a reduced expression for w.

A simple induction argument on the length of elements of W shows that the cone $\mathcal{C}^*$ may be characterized as the set of vectors v in V with the property that the set of roots

$$\{\lambda \in \Sigma \mid \lambda > 0,\ \langle \lambda, v\rangle < 0\}$$

is finite. In particular both $\mathcal{C}^*$ and $\mathcal{C}$ are convex subsets of V.

If $\lambda = w\alpha$ with α in Δ then let $\lambda^\vee = w\alpha^\vee$. The reflection in the hyperplane $\lambda = 0$ is then $s_\lambda = w s_\alpha w^{-1}$. We still have

$$s_\lambda v = v - \langle \lambda, v\rangle \lambda^\vee\ .$$

If λ is any root then I define

$$\mathcal{C}_{\lambda \geq 0} = \mathcal{C} \cap \{v \mid \langle \lambda, v\rangle \geq 0\}$$

and similarly $\mathcal{C}_{\lambda=0}$, $\mathcal{C}_{\lambda\leq 0}$ etc. All these sets are triangulated by the Coxeter faces contained in them. If λ and μ are two roots, then either the sets $\mathcal{C}_{\lambda=0}$ and $\mathcal{C}_{\mu=0}$ have no intersection, or they intersect in a union of closures of faces of codimension two. If F is one of the open faces contained in this intersection, then there exists w in W with $wF = C_X$ for some $X = \{s, t\}$,and therefore $ws_\lambda w^{-1}$ and $ws_\mu w^{-1}$ are contained in $W_{s,t}$. The converse is also true.

Another way to put this:

Lemma 1.1. *The hyperplanes $\lambda = 0$ and $\mu = 0$ have a non-empty intersection in $\mathcal{C}$ if and only if $s_\lambda s_\mu$ has order dividing $m_{s,t}$ for some distinguished generators s and t, or equivalently if and only if*

$$n_{\lambda,\mu} = \langle \lambda, \mu^\vee \rangle \langle \mu, \lambda^\vee \rangle = 2 + 2\cos(2\pi p/m_{s,t})$$

for some integer p and s and t in S. The two hyperplanes do not intersect in $\mathcal{C}$ if and only if $n_{\lambda,\mu} \geq 4$.

If Λ is a set of m roots where m is at most the cardinality of S, then similar geometrical reasoning will show that if the intersection of hyperplanes $\lambda = 0$ for λ in Λ have a common point in $\mathcal{C}$, then the subgroup generated by the s_λ will be contained in a conjugate of some W_X with $\#X \leq m$. (This is an observation of [Brink 1994].)

That $n_{\lambda,\mu}$ is never negative is a consequence of the discussion of pairs of reflections in [Vinberg 1971].

Many Coxeter groups arise from root systems. In the book [Kac 1985], a **generalized Cartan matrix** is a matrix with integral entries whose rows and columns are indexed by a finite set S, satisfying the conditions (1) $c_{s,s} = 2$ for all s; (2) for $s \neq t$ the coefficient $c_{s,t}$ is non-positive; (3) $c_{s,t} = 0$ if and only if $c_{t,s} = 0$. It is automatically a Cartan matrix in the sense of this paper. Here is a table:

$n_{s,t}$	0	1	2	3	≥ 4
$m_{s,t}$	2	3	4	6	∞

In fact, there exists an integral Cartan matrix whenever all the $m_{s,t}$ are 2, 3, 4, 6, or ∞. Such matrices are associated with certain Lie algebras of possibly infinite dimension. In this case what I call the roots are called by Kac the **real roots** of the Lie algebra. The lattice spanned by the α in Δ is preserved by the transformations in W, which is called in these circumstances a **crystallographic** Coxeter group (more properly one with a crystallographic realization). From a computational point of view, the advantage of crystallographic groups is that one can get along with only integer arithmetic. There exist crystallographic groups where the standard representation does not possess a W-stable lattice, and that is why it is not always convenient to work with it.

The most useful infinite Coxeter groups are probably the affine Weyl groups. If Σ is a finite root system on V, choose a basis Δ for positive roots, and let $\widetilde{\alpha}$ be the dominant positive root. Order the elements of Δ as $\alpha_1, \alpha_2, \ldots, \alpha_n$. Embed V^* in a space U^* of one more dimension, with an added basis element δ. Let U be the dual of U^*. Let α_{n+1} be $-\widetilde{\alpha} + \delta$ in U^*, and let C be the Cartan matrix $c_{i,j}$ with

$$c_{i,j} = \begin{cases} \langle \alpha_i, \alpha_j^\vee \rangle & i,\ j \leq n \\ -\langle \alpha_i, \widetilde{\alpha}^\vee \rangle & i \leq n,\ j = n+1 \\ -\langle \widetilde{\alpha}, \alpha_j^\vee \rangle & j \leq n,\ i = n+1 \\ 2 & i = j = n+1 \end{cases}$$

From this Cartan matrix one obtains a dual basis $\alpha_i^\vee$ of U. The affine hyperplane of U where $\delta = 1$ may be identified with V, and the Tits cone $\mathcal{C}^* = \mathcal{C}$ in U is the half-space $\delta > 0$ containing this copy of V. The Coxeter group corresponding to this Cartan matrix leaves stable this affine space, acting as the affine Weyl group, and the intersections of the chambers with V are the alcoves of the affine system.

2. Minimal roots

In this section, I will explain results from [Brink-Howlett 1993] which are necessary to understand the algorithm for producing the minimal roots and using them in further algorithms.

If λ and μ are positive roots, I will say that λ **dominates** μ if $\mathcal{C}_{\lambda\geq 0}$ contains $\mathcal{C}_{\mu\geq 0}$. Taking complements in $\mathcal{C}$, we see that this is equivalent to the condition that $\mathcal{C}_{\mu\leq 0}$ contains $\mathcal{C}_{\lambda\leq 0}$. Roughly speaking, as mentioned in the introduction, this means that $\mu = 0$ screens off $\lambda = 0$ from the fundamental chamber C in $\mathcal{C}$. Since $\mathcal{C}$ is the union of the closures of the open chambers contained in it, this is in turn equivalent to the condition that whenever $\lambda < 0$ on wC then $\mu < 0$ on wC as well. Since $\lambda < 0$ on wC precisely when $w^{-1}\lambda < 0$, the definition here is equivalent to the definition of domination found in [Brink-Howlett 1993], which specifies that λ dominates μ when

$$\{w \mid w^{-1}\lambda < 0\} \subseteq \{w \mid w^{-1}\mu < 0\}.$$

A positive root is said to be **minimal** if the only positive root it dominates is itself. Of course the simple roots are minimal.

Lemma 2.1. *Suppose w to be in W, λ and μ positive roots. If λ dominates μ and $w\mu$ is a positive root, then $w\lambda$ is also a positive root, and dominates $w\mu$.*

Proof. By assumption $\mathcal{C}_{\lambda\geq 0}$ contains $\mathcal{C}_{\mu\geq 0}$. If we apply w to this, we see that $\mathcal{C}_{w\lambda\geq 0}$ contains $\mathcal{C}_{w\mu\geq 0}$. Since $w\mu$ is assumed to be positive, the set $\mathcal{C}_{w\mu\geq 0}$ contains C, and therefore so does $\mathcal{C}_{w\lambda\geq 0}$, so that $\lambda > 0$.

The following result is the single most useful property of minimal roots.

Proposition 2.2. *If α lies in Δ and λ is a minimal root, then exactly one of the following occurs: (a) the root $s_\alpha\lambda$ is negative, which happens just when $\lambda = \alpha$; (b) the root $s_\alpha\lambda$ is positive and dominates α; or (c) the root $s_\alpha\lambda$ is positive and also minimal.*

Proof. If $\lambda \neq \alpha$ then $s_\alpha\lambda > 0$. If $s_\alpha\lambda$ dominates β and $\beta \neq \alpha$ then $s_\alpha\beta > 0$ and by the previous result λ dominates $s_\alpha\beta$. But then by assumption on λ, $s_\alpha\beta = \lambda$ and $\beta = s_\alpha\lambda$. In other words, $s_\alpha\lambda$ can only dominate itself.

If λ is any positive root, then Brink and Howlett define its **depth** to be the length of the shortest element w of W with $w^{-1}\lambda < 0$. It is also the length of a shortest chain $\lambda_1, \ldots, \lambda_n$ where λ_1 is a simple root, $\lambda_n = \lambda$, and $\lambda_{i+1} = s_i\lambda_i$ for some simple reflection s_i. If λ and μ are two positive roots then, still following Brink and Howlett, I write $\lambda \succeq \mu$ if $\lambda = w\mu$ with

$$\mathrm{depth}(\lambda) = \ell(w) + \mathrm{depth}(\mu)\ .$$

If $\lambda \succeq \mu$ and $\mathrm{depth}(\lambda) = \mathrm{depth}(\mu) + n$ then there exists a chain of roots $\mu = \mu_0$, $\mu_1 = s_{\alpha_1}\mu_0, \ldots, \lambda = \mu_n = s_{\alpha_n}\mu_{n-1}$. If α is a simple root, the depth of $s_\alpha\lambda$ is at most one more than that of λ. A geometrical formulation is straightforward:

Lemma 2.3. *The depth of $\lambda > 0$ is the same as the Coxeter distance between $\overline{C}$ and the half-space*

$$\mathcal{C}_{\lambda \le 0} = \{v \in \mathcal{C} \mid \langle \lambda, v \rangle \le 0\} \, .$$

It will be useful to keep in mind:

Lemma 2.4. *If λ is a positive root, then $\mu = w^{-1}\lambda < 0$ if and only if the hyperplane $\lambda = 0$ separates C from wC. Given this, μ lies in $-\Delta$ if and only if the chamber wC has a wall along the hyperplane $\lambda = 0$.*

Proposition 2.5. *Suppose α to be a simple root and λ any positive root other than α. The following are equivalent:*

(a) $s_\alpha\lambda$ dominates α;
(b) $\mathcal{C}_{\lambda=0}$ and $\mathcal{C}_{\alpha=0}$ are disjoint and $s_\alpha\lambda$ has depth greater than that of λ;
(c) The two regions $\mathcal{C}_{\lambda\le 0}$ and $\mathcal{C}_{\alpha\le 0}$ are disjoint.

Proof. If $s_\alpha\lambda$ dominates α, then by definition $\mathcal{C}_{s_\alpha\lambda>0}$ contains $\mathcal{C}_{\alpha\ge 0}$. Applying s_α we see that $\mathcal{C}_{\lambda>0}$ contains $\mathcal{C}_{\alpha\le 0}$, or after taking complements that $\mathcal{C}_{\alpha>0}$ contains $\mathcal{C}_{\lambda\le 0}$. Therefore (a) implies (c). Reversing this argument shows that in fact (a) and (c) are equivalent.

If $\mathcal{C}_{\lambda=0}$ and $\mathcal{C}_{\alpha=0}$ are disjoint, then α preserves sign in the region $\mathcal{C}_{\lambda\le 0}$, which must therefore be contained in one of the two halves $\mathcal{C}_{\alpha>0}$ or $\mathcal{C}_{\alpha<0}$. Suppose the first. Then $\mathcal{C}_{s_\alpha\lambda\le 0}$ is contained in the other half $\mathcal{C}_{\alpha<0}$, and $s_\alpha\lambda$ dominates α. If $C_0C_1\ldots C_n$ is a Coxeter geodesic from $C = C_0$ to $\mathcal{C}_{s_\alpha\lambda\le 0}$ then there must be an m with $\alpha > 0$ on C_m but $\alpha < 0$ on C_{m+1}, and $s_\alpha C_m = C_{m+1}$. Then the sequence

$$C_0 \ldots C_m \; s_\alpha C_{m+2} \ldots s_\alpha C_n$$

will be a chain leading to $\mathcal{C}_{\lambda\le 0}$. Therefore the depth of λ is less than that of $s_\alpha\lambda$. The other case gives the other implication.

Proposition 2.6. *If λ and μ are positive roots such that $\mu \succeq \lambda$ and μ is minimal, then so is λ.*

Proof. It reduces to the case where $\mu = s_\alpha\lambda$ has depth greater than λ and μ is minimal. By Proposition 2.2 either $\lambda = s_\alpha\mu$ is minimal or it dominates α. It cannot be equal to α, so it must properly dominate it. But then by Proposition 2.5 the depth of $s_\alpha\lambda$ is less than that of λ, a contradiction.

The following is one of the main results of [Brink-Howlett 1993]. An alternative proof can be found in [Headley 1994]. Neither of these two proofs of finiteness gives a realistic bound on the number of minimal roots, but a much more thorough discussion can be found in Chapter 6 of [Brink 1994].

Theorem 2.7. *The set of minimal roots is finite.*

If W is finite then all root hyperplanes meet at the origin, so all positive roots are minimal. If (W, S) is an affine Weyl group then the minimal roots are those of the form α or $-\alpha + 1$ for $\alpha > 0$, where α is a linear root, and the only dominance relation is that $\alpha + n$ dominates α if $n \geq 0$.

The Propositions above suggest that the way to find all minimal roots is to start at the bottom of the roots, ordered by $\succeq$, and search upwards through it. Here is the algorithm which follows this strategy. It simultaneously builds the minimal root reflection table.

Algorithm to find all minimal roots. *Maintain a queue and a dictionary of minimal roots. Each minimal root is listed along with entries from the corresponding column in the reflection table, as well as arrays of the couplings $\langle \lambda, \alpha^\vee \rangle$, $\langle \alpha, \lambda^\vee \rangle$.*

Maintain also a dictionary of possible coefficients $n_{\lambda,\mu}$ for the cases when s_λ and s_μ generate a finite group.

A root λ is on the queue if it has been stored in the dictionary and recognized as a minimal root, but the roots $s_\alpha \lambda$ have not yet been examined. The point of using a queue is that we are doing a breadth first search of the bottom of the root graph, and when we pop the root λ off the queue we know that all minimal roots of depth less than or equal to that of λ are in the dictionary.

A dictionary means a listing with two relevant procedures: look-up and installation of items. A queue is a list with two procedures, installation and removal (pushing and popping), which are done on a first-in-first-out basis.

Start by putting all simple roots in the queue and in the dictionary. For each α fill in the column entry for $s_\alpha \alpha$ with a $-$. Put in also the corresponding rows and columns of the Cartan matrix.

While the queue is not empty:

- ◇ *Remove λ from the queue.*
- ◇ *For each simple root α:*
    - ◇ *Look at the reflection column. If there is an entry there for α, do nothing. Otherwise calculate $\mu = s_\alpha \lambda$.*
    - ◇ *Look up μ in the dictionary. If it is there, set $s_\alpha \lambda = \mu$ and $s_\alpha \mu = \lambda$.*
    - ◇ *If it is not there, either the root μ is a new minimal root, or it dominates α. Check whether s_λ and s_α generate a finite group by looking up $n_{\lambda,\alpha}$ in the coefficient dictionary. If they do not, then $s_\alpha \lambda$ will dominate α, and we add an entry $+$ in the reflection column.*
    - ◇ *Otherwise μ is a new minimal root. Install μ in the dictionary; set $s_\alpha \lambda = \mu$ and $s_\alpha \mu = \lambda$; and put μ in the queue.*

At the end, all the minimal roots are in the dictionary, as well as all entries in the reflection table.

3. The recognizing automata

First some remarks on abstract languages.

If A is any finite set of symbols then a **word** in the alphabet A is a sequence $a_1 \bullet a_2 \bullet \ldots \bullet a_n$ of elements of A concatenated together. The **empty** word will be that of length zero. A **language** based on A is any set of words in A. A **complete automaton** based on A will be a kind of oriented labelled graph consisting of (1) nodes; (2) oriented edges from one node to another, labelled by elements of A; (3) a designated **initial** node; and (4) a designated set of **final** nodes. These data must satisfy the condition that each node is the source of exactly one edge labelled by each of the symbols in A. A path in the automaton starting at a given node ν is a sequence of nodes ν_i with $\nu_0 = \nu$ and the additional property that each pair (ν_{i-1}, ν_i) lies on an edge of the automaton. If the edge (ν_{i-1}, ν_i) is labelled by a_i then such a path determines a word $a_1 \bullet a_2 \bullet \ldots \bullet a_n$. The language **recognized** by such an automaton is the set of words ω in A corresponding in this way to paths in the automaton starting at the initial node and ending at a final node.

I call a map from the nodes of one automaton based on A to another **automatic** if it takes edges to edges, preserves labels, takes the initial node to the initial node, and has the property that the final nodes are the images of the final nodes.

There are two canonical automata determining any given language, the maximal one and the minimal one. The maximal one, which I call $\mathcal{M}^L_{\max}$, has for nodes all the words in A, including the empty one. The initial node is the empty word. The final nodes are the words in L. There is an edge from ω to χ labelled by a if and only if $\chi = \omega \bullet a$.

The minimal one, which I call $\mathcal{M}^L_{\min}$, has as its nodes sets of equivalence classes of words in A. Two words ω_1 and ω_2 are defined to be L-equivalent if for all words χ the word $\omega_1 \bullet \chi$ lies in L if and only if the word $\omega_2 \bullet \chi$ lies in L — that is to say, if the words concatenated with ω_1 and ω_2 to obtain words in L are the same. If $[\omega]$ is the equivalence class of a word ω and a is in A, then the edge from $[\omega]$ labelled by a goes to $[\omega \bullet a]$. The initial node is the equivalence class of the empty word. The final nodes are the equivalence classes containing words in L, which are made up only of words in L. The map from a word ω to its class $[\omega]$ is an automatic map from $\mathcal{M}^L_{\max}$ to $\mathcal{M}^L_{\min}$.

A word is called a **prefix** of the language L if it is possesses an extension in L, and otherwise it is called **dead** with respect to L. All dead words comprise one equivalence class in $\mathcal{M}^L_{\max}$, collapsing to a single node in $\mathcal{M}_{\min}$. All edges starting at the dead state node just loop back to it, so that all paths starting at it never leave it. A node x in an automaton is called **accessible** if there exists a path in the automaton from the starting node to x, otherwise **inaccessible**. Every automaton is equivalent to one with only accessible states and at most one dead state, which we may as well assume to be implicit. From now on, all automata I refer to will have these properties, and the transitions to the dead state will be unspecified (and any transition not specified will move to the dead state).

Let $\mathcal{A}^L_{\min}$ be the automaton obtained from $\mathcal{M}^L_{\min}$ by removing the dead node, and similarly $\mathcal{A}^L_{\max}$.

Proposition 3.1. *(Myhill-Nerode) Suppose $\mathcal{X}$ to be an automaton with alphabet A. If $\mathcal{X}$ recognizes L then there are unique automatic maps from $\mathcal{A}^L_{\max}$ to $\mathcal{X}$ and from $\mathcal{X}$ to $\mathcal{A}^L_{\min}$ through which the canonical map from $\mathcal{A}^L_{\max}$ to $\mathcal{A}^L_{\min}$ factors. Conversely, if there is such a map, then $\mathcal{X}$ recognizes L.*

Proof. If ω is a word in L, it gives rise by assumption to a path in $\mathcal{X}$ starting at the initial node of $\mathcal{X}$. An automatic map from $\mathcal{A}^L_{\max}$ to $\mathcal{X}$ is defined by the condition that it must take ω to the last node in this path.

All nodes of $\mathcal{X}$ are assumed accessible. If x is a node of $\mathcal{X}$ and ω_1 and ω_2 are both words corresponding to paths in $\mathcal{X}$ starting at the initial node and ending at x, then ω_1 and ω_2 are equivalent with respect to L. Map x to the equivalence class $[\omega_1] = [\omega_2]$. This determines an automatic map from $\mathcal{X}$ to $\mathcal{A}^L_{\min}$, which is in fact defined in the only way possible.

There is a well known efficient algorithm, described for example in [Aho-Ullman 1974], for constructing from a given finite automaton the minimal one recognizing the same language.

Most often in this paper, and in particular for the rest of this section, the alphabet A will be S. There is thus an obvious correspondence between words and Coxeter paths starting at the fundamental chamber C.

For any word $\omega = s_1{\bullet}s_2{\bullet}\ldots{\bullet}s_n$ let $\overline{\omega}$ be the corresponding element $s_1 s_2 \ldots s_n$ of W. Recall that the length $\ell(w)$ is the length of a word of least length representing w. In this paper, three languages will be of interest. Two of them depend on the choice of an ordering of the elements of S. (1) The first is `Reduced`, the language of all words $\omega = s_1{\bullet}s_2{\bullet}\ldots{\bullet}s_n$ with the property that the expression $w = s_1 s_2 \ldots s_n$ has minimal length among all expressions for w as a product of generators in S. (2) The second is `ShortLex`, which consists of words $s_1{\bullet}s_2{\bullet}\ldots{\bullet}s_n$ in `Reduced` with the extra property that for each i, s_i is the least element of S occurring in a reduced expression for $s_i s_{i+1} \ldots s_n$. (The terminology arises in the book [Epstein et al. 1991], where many of the ideas in this paper also first arose.) (3) The third is `InvShortLex`, which consists of the words of `ShortLex` written in reversed order.

From now on in this paper, fix an ordering of S.

All of these languages have the property that any sub-word in a word of the language is again in the language.

If L is either `ShortLex` or `InvShortLex` then any element in W may be represented by a unique word in L, so that the words in L may be identified with W. Even when L is `Reduced`, we may construct an automaton recognizing L whose nodes are elements of W. In this automaton, there will exist an edge from x to y, labelled by s, if $y = xs$ and $\ell(y) = \ell(x) + 1$. For each of these languages L, let $\mathcal{W}^L$ be the automaton defined in this way. Uniqueness of expression means it is a tree.

There is a useful geometrical description of the minimal automaton for the languages `Reduced` and `ShortLex`. If L is one of these two languages, define $\mathcal{C}^L_w$ to be the union of the geometric successors of the closed chamber $w\overline{C}$ with respect to L, which is to say the union of all chambers $x\overline{C}$ such that there exists a word in L representing x with a prefix representing w or, equivalently, such that there exists a Coxeter path in L from C to xC passing through wC.

Lemma 3.2. *Suppose L to be one of the two languages* `Reduced` *or* `ShortLex`*.*

(a) If the word $\omega = s_1 \bullet s_2 \bullet \ldots \bullet s_n$ lies in L and $w = \overline{\omega}$ then $\omega \bullet s$ will be contained in L precisely when wsC is contained in $\mathcal{C}^L_w$.

(b) In these circumstances

$$\mathcal{C}^L_{ws} = \mathcal{C}^L_w \cap w\mathcal{C}^L_s \ .$$

(c) For words ω and χ in L with $w = \overline{\omega}$ and $x = \overline{\chi}$, the concatenation $\omega \bullet \chi$ lies in L if and only if $wxC \subseteq \mathcal{C}^L_w$.

This is best left as an exercise. It is to be combined with the simple calculation

Lemma 3.3. *If (a) L is* `Reduced` *then*

$$\mathcal{C}^L_s = \{v \in \mathcal{C} \mid \langle \alpha_s, v \rangle \leq 0\}$$

while if (b) L is `ShortLex` *then*

$$\mathcal{C}^L_s = \{v \in \mathcal{C} \mid \langle \alpha_s, v \rangle \leq 0, \ \langle \alpha_t, v \rangle \geq 0 \text{ for } t < s\} \ .$$

We can now describe the minimal automata recognizing `Reduced` and `ShortLex`. For each w in W, define $\mathcal{S}^L_w$ to be $w^{-1}\mathcal{C}^L_w$. This set will always contain the chamber C. The definitions imply directly that two nodes w_1 and w_2 in $\mathcal{W}^L$ are equivalent if and only if $\mathcal{S}^L_{w_1}$ and $\mathcal{S}^L_{w_1}$ are the same.

The Propositions above translate nicely.

Proposition 3.4. *Suppose L to be one of the two languages* `Reduced` *or* `ShortLex`*.*

(a) If the word $\omega = s_1 \bullet s_2 \bullet \ldots \bullet s_n$ lies in L and $w = \overline{\omega}$ then $\omega \bullet s$ will be contained in L precisely when sC is contained in $s\mathcal{S}^L_w$.

(b) In these circumstances

$$\mathcal{S}^L_{ws} = s\mathcal{S}^L_w \cap \mathcal{S}^L_s \ .$$

Proposition 3.5. *If L is (a)* `Reduced` *then for each s in S*

$$\mathcal{S}^L_s = \{v \in \mathcal{C} \mid \langle \alpha_s, v \rangle \geq 0\}$$

while if L is (b) `ShortLex` *then*

$$\mathcal{S}^L_s = \{v \in \mathcal{C} \mid \langle \alpha_s, v \rangle \geq 0, \ \langle s\alpha_t, v \rangle \geq 0 \text{ for } t < s\} \ .$$

Define the automaton $\mathcal{I}^L$ to have as its nodes the sets $\mathcal{S}^L_w$ as w ranges over W. That is to say, there is one node in $\mathcal{I}^L$ for each of the distinct sets which occur as some $\mathcal{S}^L_w$. The point is that the inductive formula for $\mathcal{S}^L_{ws}$ depends only on $\mathcal{S}^L_w$ and not explicitly on w. There will exist an edge labelled by s from $\mathcal{S}$ to $\mathcal{T}$ if and only if sC is contained in $\mathcal{S}$ and $\mathcal{T} = s\mathcal{S} \cap \mathcal{S}^L_s$. The map taking w to $\mathcal{S}^L_w$ is automatic from $\mathcal{W}^L$, and therefore from Proposition 3.1:

Proposition 3.6. *The automaton $\mathcal{I}^L$ may be identified with $\mathcal{A}^L_{\min}$.*

The description of the sets $\mathcal{S}_s$ shows that they are all convex. The inductive equation for $\mathcal{S}_{ws}$ then shows that all the sets $\mathcal{S}_w$ are also convex. Since the walls of the chambers are root hyperplanes, the walls of the sets $\mathcal{S}_w$ inside $\mathcal{C}$ are also root hyperplanes. The set $\mathcal{S}_w$ can be completely described by listing those positive roots whose hyperplanes form its walls. It follows from the inductive formula for $\mathcal{S}^L_w$ that each one of these sets is bounded by only a finite number of walls. It is therefore possible to construct the $\mathcal{S}_w$ inductively, for example, by linear programming. This is neither efficient nor informative. An explicit automaton recognizing L was first described in [Brink-Howlett 1993]. It shows that the number of nodes necessary is in fact finite.

Theorem 3.7. *For each $\mathcal{S} = \mathcal{S}^L_w$ let $\Theta_{\mathcal{S}}$ be the set of minimal roots which are non-negative on $\mathcal{S}$. Then*

$$\mathcal{S} = \{v \in \mathcal{C} \mid \langle \theta, v\rangle \geq 0 \text{ for all } \theta \text{ in } \Theta_{\mathcal{S}}\}];.$$

Proof. Let $\Sigma^+_{\min}$ be the set of minimal roots.

First of all, the claim is true for each $\mathcal{S}^L_s$. This is immediate for `Reduced`. For `ShortLex` we have the characterization

$$\mathcal{S}^L_s = \{v \in \mathcal{C} \mid \langle \alpha_s, v\rangle \geq 0,\, \langle s\alpha_t, v\rangle \geq 0 \text{ for } t < s\}\ .$$

But Proposition 2.2 implies that either $s\alpha_t$ is minimal or it dominates α_s. If it dominates α_s, then the condition $\langle s\alpha_t, v\rangle \geq 0$ is redundant since it is implied by the condition $\langle \alpha_s, v\rangle \geq 0$.

The induction step is then similar. Let $s = s_\alpha$ and suppose that sC is contained in $\mathcal{S}^L_w$ or equivalently α does not lie in Θ_w. Then by induction

$$\mathcal{S}^L_{ws} = \{v \in \mathcal{C} \mid \langle \theta, v\rangle \geq 0 \text{ for } \theta \text{ in } s\Theta_w \cup \Theta_s\}\ .$$

If λ is a minimal root in Θ_w then either $s\lambda$ is a minimal root or it dominates α. But in the latter case it is a redundant element of $s\Theta_w$, so that in fact

$$\mathcal{S}^L_{ws} = \{v \in \mathcal{C} \mid \langle \theta, v\rangle \geq 0 \text{ for } \theta \text{ in } \left(s\Theta_w \cap \Sigma^+_{\min}\right) \cup \Theta_s\}\ .$$

This concludes the proof.

It is not easy to construct directly the sets $\Theta_{\mathcal{S}}$, which is to say it is not so easy to see how to compute them explicitly. What is not difficult to do is to construct other automata recognizing L whose states are subsets of minimal roots. Brink and Howlett, for example, construct an automaton from subsets of minimal roots according to the following procedure.

First of all define

$$\Theta^L_{s,\mathrm{BH}} = \begin{cases} \{\alpha_s\} & \text{if } L \text{ is } \texttt{Reduced} \\ \{\alpha_s\} \cup_{t<s} \{s\alpha_t\} & \text{if } L \text{ is } \texttt{ShortLex} \end{cases}$$

Then define inductively

$$\Theta^L_{ws,\mathrm{BH}} = \left(s\Theta^L_{w,\mathrm{BH}} \cap \Sigma^+_{\mathrm{min}}\right) \cup \Theta^L_{s,\mathrm{BH}}$$

The distinct sets $\Theta_{w,\mathrm{BH}}$ one gets in this way are the nodes of the automaton $\mathcal{A}^L_{\mathrm{BH}}$. Make an edge from Θ to Φ, labelled by $s = s_\alpha$, if α is not in Θ and

$$\Phi = \left(s\Theta \cap \Sigma^+_{\mathrm{min}}\right) \cup \Theta_s \ .$$

The procedure for constructing the automaton $\mathcal{A}^L_{\mathrm{BH}}$ is quite straightforward. *We need a dictionary of states already constructed, implemented say as bitmaps of minimal roots, denoting the sets Θ_w. We also maintain a stack of states already constructed. We start with the sets $\Theta_{s,\mathrm{BH}}$ in the dictionary and on the stack. While the stack is not empty we build the Θ_w inductively, adding links to old ones and installing new ones in the dictionary and putting them on the stack.* This procedure has the great virtue that the only datum it requires is the minimal root reflection table. It has the drawback that many Θ_w will be equivalent to each other, and the size of the resulting automaton turns out to be impractically large. This is exhibited by statistics in [Casselman 1994].

We can modify the construction slightly according to the following observation: if δ and γ are both non-negative on $\mathcal{S}_w$ then so are all positive linear combinations of the two. Following this principle I modify the definitions of the sets Θ.

For L equal to `Reduced` we let Θ^*_s be Θ_s, but for L equal to `ShortLex` we set

$$\Theta^*_s = \alpha_s \cup s\Sigma^+_{<s}$$

where $\Sigma^+_{<s}$ is the set of positive roots which are linear combinations of the α_t with $t < s$.

Then we define inductively

$$\Theta^*_{ws} = \left(s\Theta^*_w \cap \Sigma^+_{\mathrm{min}}\right) \cup_* \Theta^*_s$$

where the union $\cup_*$ is an operation that adds to the union $s\Theta^*_w \cup \Theta^*_s$ only minimal roots which are positive integral combinations of minimal roots already there. Specify an edge from Θ^*_w to Θ^*_{ws} only if α_s does not lie in Θ_w. We again have an automaton determining L, and the number of distinct sets Θ^*_w turns out to be far smaller, in practice, than the number in $\mathcal{A}^L_{\mathrm{BH}}$.

There are efficient ways to carry out this partial saturation process. The trick is that *in saturating the set*

$$\left(s\Theta^*_w \cap \Sigma^+_{\mathrm{min}}\right) \cup \Theta^*_s$$

we may assume that each of the two separate sets is already saturated. But the right hand term is fixed in advance, so we may calculate ahead of time for each s and for each minimal root λ the set of minimal roots of the form $\lambda + \mu$ for μ in Θ^*_s.

For the language `Reduced` we can reformulate the main result in the following way. Call two elements x and y of W equivalent if the set of z in W with $\ell(xz) = \ell(x) + \ell(z)$ is the same as the set of z with $\ell(yz) = \ell(y) + \ell(z)$. Then the number of equivalence classes is always finite. If W is finite then because W possesses a unique longest element w_ℓ an element of W is equivalent only to itself. For affine groups, it is shown in [Headley 1994] that one can make an automaton recognizing `Reduced` from the connected components in $\mathcal{C}$ of the complement of the minimal root hyperplanes. His construction appears to remain valid for general Coxeter groups. It is extremely curious, as Headley points out, that the automaton for the affine groups of type A_n was described implicitly in [Shi 1984] in quite another context. It is not clear to what extent automatic structures underly Shi's computations, nor more generally to what extent they play a role in describing Kazhdan-Lusztig cells for general groups. At any rate, there ought to be some interesting consequences of the finiteness of the automaton recognizing `Reduced`.

4. How to multiply `InvShortLex` words by a generator

If x is a `ShortLex` word and s a Coxeter generator, then the problem of finding the `ShortLex` word for $\overline{x}s$ is the same as that of finding the `InvShortLex` word for $s\overline{x}^{-1}$. Geometrically, this is a simpler problem.

So now we pose the transformed question: *given the* `InvShortLex` *word x, how do we find the* `InvShortLex` *word for $s\overline{x}$?*

The routine explained here has the remarkable property that it depends on the minimal root reflection table, and on no other structure associated to the Coxter group. It was alluded to at the end of [Casselman 1994] and is implicit in the description of the multiplier automata in [Brink-Howlett 1993].

We begin with the question: suppose $x \to y$ is an `InvShortLex` transition, labelled by γ. Given a Coxeter generator $s = s_\alpha$, is $sx \to sy$ also an `InvShortLex` transition? The assumption means that (a) $y = xs_\gamma$; (b) $x\gamma > 0$ or equivalently $y\gamma < 0$; (c) there is no $\beta < \gamma$ with $y\beta < 0$. Since $sxs_\gamma = sy$, we must check properties (b) and (c) for sy and γ. If (b) fails then we will have what I call for a moment failure of the first kind: $sx\gamma < 0$. But then since $x\gamma > 0$ we must have

$$x\gamma = \alpha, \quad \gamma = x^{-1}\alpha, \quad xs_\gamma x^{-1} = s_{x\gamma} = s_\alpha, \quad xs_\gamma = s_\alpha x \ .$$

In other words, the `InvShortLex` edge from x to y crosses the hyperplane $\alpha = 0$, and of course the transformed edge is the opposite of the original edge. In failure of the second kind, $sy\gamma < 0$ but there exists $\beta < \gamma$ with $sy\beta < 0$ also. Since γ satisfies condition (c) for $x \to y$, $y\beta > 0$. But then

$$y\beta = \alpha, \quad \beta = y^{-1}\alpha, \quad s_\alpha y = ys_\beta \ .$$

What happens here is that C_y and C_{sy} share a wall labeled by β on $\alpha = 0$, and that the `InvShortLex` edge coming into C_{sy} is no longer the edge from C_{sx} labeled by γ, but is labeled by β instead and crosses from C_y on the positive side of $\alpha = 0$ to C_{sy} the negative side. In this case we even know that the `InvShortLex` path to C_{sy} passes through C_x and C_y just before C_{sy}. Equivalently, the `InvShortLex` word for sy looks like $\ldots s_\gamma s_\beta$.

It might be worthwhile to discuss here, to some extent, what happens when the elementary reflection s_α is replaced by an arbitrary element w. So we ask: given an `InvShortLex` edge $x \to y$ labelled by γ, is $wx \to wy$ one, too? If not, then one possibility for failure is that $wx\gamma < 0$. In this case, if $\lambda = x\gamma$ then $\lambda > 0$ but

$w\lambda < 0$, and the wall between C_x and C_y is one that separates C from $C_{w^{-1}}$. The other possibility is that $wy\gamma < 0$ but there exists $\beta < \gamma$ with $wy\beta < 0$. In the second case $y\beta > 0$, and if $\lambda = y\beta$ then C_y has a wall $\lambda = 0$ separates C from $C_{w^{-1}}$.

At any rate, we have proven this result, which is what we shall need in a moment.

Lemma 4.1. *If $x \to y$ is an edge in the* `InvShortLex` *tree labelled by γ, and α is a simple root, then $s_\alpha x \to s_\alpha y$ is also an edge in the* `InvShortLex` *tree unless one of these two possibilities occurs:*

(1) $\gamma = x^{-1}\alpha$ and $y = s_\alpha x$, in which case the edge is reversed;
(2) the cell C_y has a wall along the hyperplane $\alpha = 0$, labelled by the simple root β with $\beta < \gamma$. In this case the `InvShortLex` *edge into C_{sy} is through the wall labelled by β from C_y.*

If C_y is a chamber in the region $\alpha > 0$ with a wall on the hyperplane $\alpha = 0$ labeled by β, and there exists an `InvShortLex` edge to $C_{s_\alpha y} = C_{ys_\beta}$, I shall say that C_y is a **potential** **`InvShortLex`** **crossing point** with respect to α. There is a simple criterion for this to happen: if γ be the label of the `InvShortLex` path leading into y, then we must have $y^{-1}\alpha$ equal to a simple root β with $\beta < \gamma$.

Theorem 4.2. *Suppose that w is an* `InvShortLex` *word, s_α a Coxeter generator with α a simple root.*

(1) If $C_{\overline{w}}$ lies in the region $\alpha < 0$, then suppose $w = x{\bullet}s{\bullet}y$ where the cells $C_{\overline{x}}$ and $C_{\overline{x}s}$ lie on opposite sides of $\alpha = 0$. Then the `InvShortLex` *word for $s_\alpha\overline{w}$ is $x{\bullet}y$.*
(2) If $C_{\overline{w}}$ lies in the region $\alpha < 0$, let $w = x{\bullet}y$ where C_x is the last potential `InvShortLex` *crossing point in the string w with respect to α, with crossing labelled by β. Then the* `InvShortLex` *word for $s_\alpha\overline{w}$ is $x{\bullet}s_\beta{\bullet}y$.*

In the first case, this is just the strong exchange condition. The rest is an immediate consequence of the arguments above.

Therefore, in order to calculate the `InvShortLex` word for $s\overline{w}$ we must distinguish these two cases, and we must be able to tell potential crossing points in the string w. The first is simple: $\alpha > 0$ on $C_{\overline{w}} = \overline{w}C$ if and only if $\overline{w}^{-1}\alpha > 0$ on C, or $\overline{w}^{-1}\alpha > 0$. So we distinguish the two possibilities according to whether $\overline{w}^{-1}\alpha$ is a positive or negative root. We can calculate $w^{-1}\alpha$ in a simple fashion as we read the string w. Let $w_i = s_1 \dots s_i$, $s_i = s_{\alpha_i}$, $\lambda_i = w_i^{-1}\alpha$. Then

$$\lambda_{i+1} = w_{i+1}^{-1}\alpha = s_{i+1}w_i^{-1}\alpha = s_{i+1}\lambda_i,$$

and we can calculate the roots λ_i inductively as we read w from left to right. The final root $w^{-1}\alpha$ will be negative if and only if one of the $\lambda_i = \beta$ for some simple root β and $s_{i+1} = s_\beta$.

In other words, tracking the λ_i tells us when we cross over the hyperplane $\alpha = 0$. In fact, the same data will determine potential crossings. The cell C_{w_i} will be a potential crossing if and only if $\lambda_i = \beta$ for some simple root β with $\beta < \alpha_i$.

There is one extra simplification. If we cross a wall $\lambda = 0$ where λ is a positive root with the property that $\mathcal{C}_{\lambda \le 0}$ is disjoint from $\mathcal{C}_{\alpha \le 0}$, then we can never cross back over it, and we can never again cross $\alpha = 0$. Now in going from wC to $ws_\gamma C$ we cross the hyperplane $w\gamma = 0$, and in our circumstances we will have

$$w\gamma > 0, \quad w^{-1}\alpha < 0, \quad \mathcal{C}_{w\gamma \le 0} \cap \mathcal{C}_{\alpha \le 0} = \emptyset\ .$$

But then applying w^{-1} we also have

$$\mathcal{C}_{\gamma \le 0} \cap \mathcal{C}_{w^{-1}\alpha \le 0} = \emptyset$$

and by Proposition 2.5 this is equivalent to the condition that $s_\gamma w^{-1}\alpha$ dominates γ. In other words, we can detect when we cross such a hyperplane in going from w to ws_γ by checking whether $s_\gamma w^{-1}\alpha$ remains a minimal root or not.

Furthermore, once we have crossed $\alpha = 0$ we will stay there. In other words, we have to keep exact track of λ_i only as long as it is a minimal root. *We may use the minimal root reflection table to do this.*

We wind up with this algorithm:

Algorithm for left multiplication by a generator. *Maintain a state described by three items: ⋄ an elementary reflection s, which can be 1; ⋄ an extended minimal root λ (either a minimal root or $\pm$); ⋄ a string σ, which can be empty; ⋄ a string x.*

We are going to read the `InvShortLex` word w from left to right, building up the `InvShortLex` expression x for $s_\alpha \overline{w}$ as we proceed.

At each point, the generator s represents the last potential crossing in an `InvShortLex` path, σ a part of the accumulated string for $s_\alpha \overline{w}$, and λ the relative position of the wall $\alpha = 0$.

At the start $s = s_\alpha$, $\lambda = \alpha$, $\sigma = \emptyset$, $x = \emptyset$.

So for $i = 1$ to n we do this:

- ⋄ *Read the generator s_i.*
- ⋄ *If λ is a minimal root:*
  - ⋄ *Calculate the new $\lambda := s_i\lambda$.*
  - ⋄ *If λ is now equal to a simple root β with $\beta < \alpha_i$, we have a potental crossing. We append σ to x, change s to s_β.*
  - ⋄ *Else if λ is an ordinary minimal root but not simple, we tack s_i onto the end of σ.*
  - ⋄ *Else if $\lambda = -$ (we have followed the path s_β from a potential crossing), we output σ and then s_i, setting $s = 1$, $\sigma = 1$.*
  - ⋄ *Else if $\lambda = +$, we append s, σ, and s_i, setting s and σ to 1.*
- ⋄ *If $\lambda = \pm$: we just append s_i.*

At the end we append the current s and σ (possibly both trivial) to x.

For affine groups and Weyl groups, the possible length of the storage string σ is bounded, but an example due to [Le Chenadec 1986] shows that this is not always true. This is equivalent to the question of whether a certain finite state machine constructed from the `InvShortLex` automaton together with the minimal root reflection table has a loop, and is simple to check. Fokko du Cloux has proven that this never happens for affine Weyl groups.

For contrast, let me recall the simple alternative algorithm for calculating `InvShortLex` products, mentioned in Humphreys' book. Let ρ be the unique vector such that $\langle \alpha, \rho \rangle = 1$ for all simple roots. It lies in the interior of the fundamental chamber. For each w in W, let $v_w = w^{-1}\rho$, recorded by its coordinates $v_\alpha = \langle \alpha, v \rangle$. Then $ws_\alpha > w$ if and only if $w\alpha > 0$, or $\langle w\alpha, \rho \rangle = \langle \alpha, w^{-1}\rho \rangle = v_\alpha$ so that to find the `InvShortLex` word for w we let α be the least α with $v_\alpha < 0$, so that $ws_\alpha < w$. Set $w := ws_\alpha$ and continue until there are no α with $v_\alpha < 0$, in which case $v = \rho$ and $w = 1$. Because of the searching among the coordinates, this algorithm is proportional to the product of the rank and the length of w.

The new algorithm looks better. On the one hand it has no dependence on rank. On the other, for statistical reasons it will probably stop sooner, as soon as λ becomes $\pm$. Also, the algorithm using $w^{-1}\rho$ really requires crystallographic systems, because it needs to test inequality, and hence involves arithmetic of real numbers, in general requiring an unknown degree of precision.

On the other hand, when available, the algorithm using ρ has the advantage that it can find an `InvShortLex` expression for any word in time proportional to the rank times word length. I do not know of anything similar using the machinery explained in this paper.

5. More automatic structures

Most words in W will have several distinct reduced expressions, but Lusztig has pointed out in [Lusztig 1983] that the ones with a unique reduced expression, excepting the identity, make up a single two-sided cell $\mathcal{U}(W)$ in the group (in the sense of [Kazhdan-Lusztig 1979], and that if we identify this cell with the chambers $w\overline{C}$ in $\mathcal{C}^*$ as w ranges over the elements in the cell, its connected components are the right cells contained in it. It is intriguing that this cell has an automatic structure.

Proposition 5.1. *There exists a finite automaton recognizing the* `ShortLex` *words representing elements of* $\mathcal{U}(W)$.

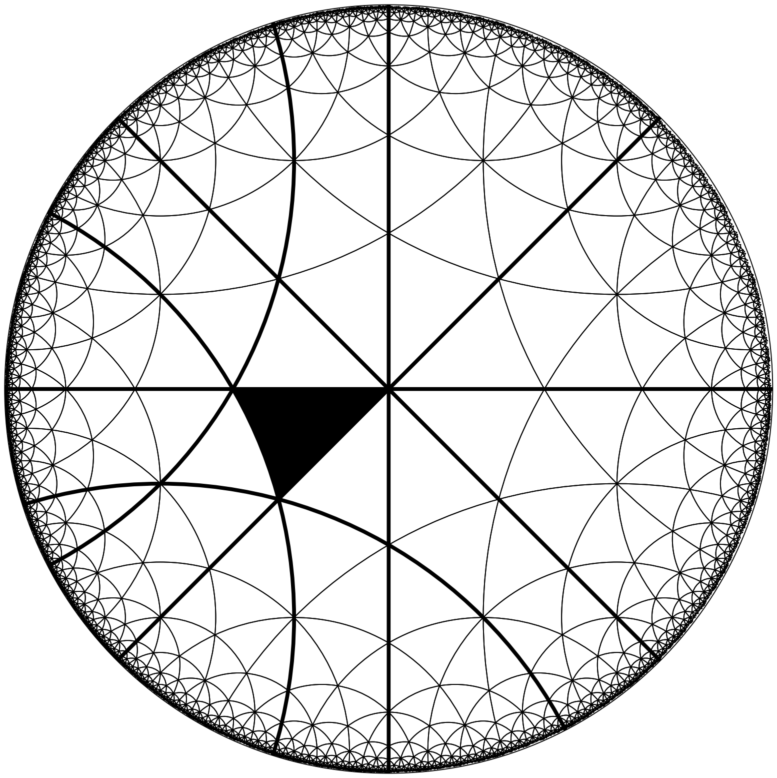
Proof. The simplest way in which w can fail to have a unique reduced expression is a reduced expression for w contains a product $stst\ldots$ ($m_{s,t}$ terms) since this product can also be written as $tsts\ldots$ It is a simple observation that in fact w will have a unique reduced expression if and only if its `ShortLex` expression has no subexpressions of this form. It is a well known result in the theory of automata, however, that the words in any language recognized by a finite automaton which do not contain any of a finite set of substrings is also recognized by a finite automaton. This is part of the algebra of regular expressions and automata (see the chapter on regular expressions in [Aho-Ullman 1974]). It is even a simple matter to find this automaton explicitly, given the automaton recognizing the original language.

The sets one gets in this way have an apparently random appearance. For affine Weyl groups, this is explained by the result of Lusztig's which establishes a bijection between the two-sided cells of W and the nilpotent conjugacy classes in the Lie algebra of an associated semi-simple complex group. The cells described above correspond to the sub-regular nilpotent class. The structure and significance of the right cells does not seem to be completely elucidated, but conjecturally they are related to the cohomology of certain sub-varieties of the flag manifold of this group. Lusztig has included some intriguing pictures of the cells of affine groups of rank three at the end of [Lusztig 1985], but even knowing all of his Theorems in the series on affine cells, these pictures suggest many unsolved puzzles. For finite Weyl groups, the cells occur in several contexts, most notably perhaps in connection with Springer's construction of irreducible representations of the group. All other cases, as far as I can tell, remain a mystery. Bédard has discussed the cells for a small set of hyperbolic Coxeter groups in the two papers [Bédard 1986] and [Bédard 1989]. His results are striking, not least because they probably raise more questions than they answer. In the rest of this paper I will look a little more closely at one of his examples.

We choose three generators s_0, s_1, s_2 with the relations

$$(s_0 s_1)^3 = 1, \quad (s_0 s_2)^4 = 1, \quad (s_1 s_2)^3 = 1 \; .$$

In the standard realization (which I have already mentioned in §1), this group preserves a certain quadratic form of signature $(2,1)$, and by a suitable change of coordinates we may as well assume it to be $x^2 + y^2 - z^2$. The Tits cone is the interior of the null cone, the region $x^2 + y^2 < z^2$. The group also preserves the hyperbolic plane $x^2 + y^2 - z^2 = 1$, which possesses a Riemannian metric of constant curvature equal to -1, and is congruent to the Poincaré upper half-plane and *via* the Cayley transform to the interior of the unit disc $|z| < 1$. In this realization of non-euclidean geometry the geodesics are the arcs of circles cutting the circumference perpendicularly. Things can be arranged so that the fundamental domain is a triangle with one corner at the origin and two sides therefore equal to diameters. See the first figure.



There are in this case exactly seven minimal roots; the lines $\lambda = 0$ with λ minimal are also shown in the figure.

I should say something about how the figures here are drawn. First of all I run a program which reads in the Coxeter numbers $(3, 4, 3)$ and then applies the algorithms described in this paper to produce as output the minimal root reflection table and an automaton which recognizes `ShortLex`.

Here, for example, is the table:

```
name 343
rank 3
minrootno 7

 0:      -   5   3   2   +   1   6
 1:      5   -   4   +   2   0   +
 2:      6   4   -   3   1   +   0
```

Then another program is run on the automaton to find its minimal equivalent. Here it is:

```
name    343
inputno 3
stateno 15
edgeno  23
default 14

  0:  2 1,1 2,0 3,A,S
  1:  1 5,0 6,A
  2:  2 1,0 4,A
  3:  2 1,1 2,A
  4:  2 1,A
  5:  0 8,A
  6:  2 7,1 2,A
  7:  1 5,A
  8:  2 9,A
  9:  1 5,0 10,A
 10:  1 11,A
 11:  2 12,0 4,A
 12:  0 13,A
 13:  2 7,1 4,A
 14:
```

In this case, partial saturation produces an automaton which has only one more state than the minimal one. I do not know what the simpler procedure of Brink and Howlett comes up with.

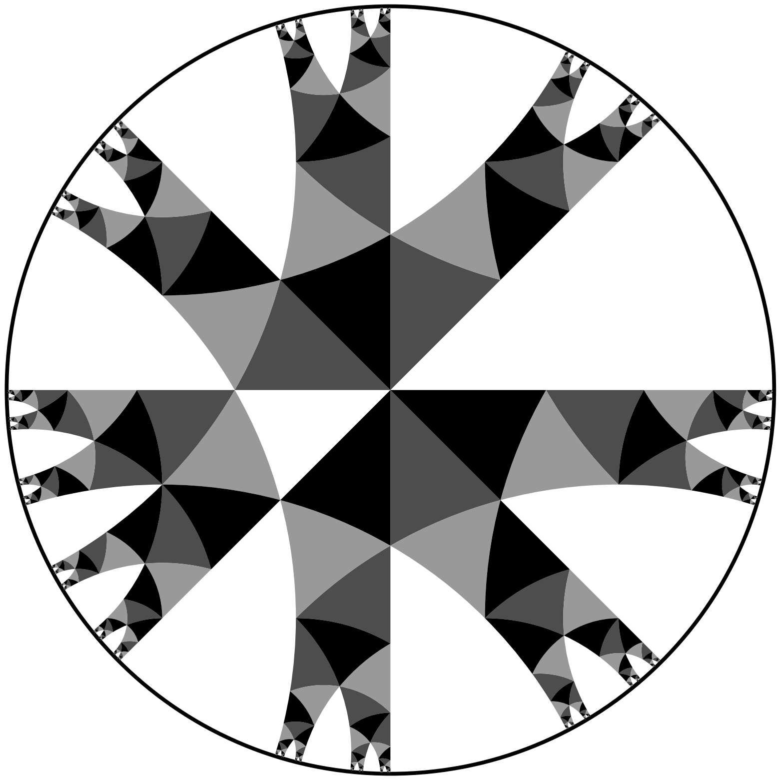
Finally, a program is run which reads the automaton and produces all words in the language recognized by it up to a specified depth, here 18. For each one of these words w the non-Euclidean triangle representing $\overline{w}C$ is drawn. In the previous figure, to be precise, there are exactly 6318 triangles if my count is correct. (It takes far longer for PostScript to render them on the printed page than it does for the program to read words and produce the PostScript.) This is not quite the obvious technique to use, but it has the property of being flexible, which is, as every programming text explains tediously, a great virtue.

One interesting thing you can do with this scheme is select in various ways which triangles you are going to draw. The simplest way to do this, except of course simply spilling out all the words into a file and picking them out by hand, so to speak, is to apply some systematic rule to have the machine do the selection. On UNIX machines, the natural tool to apply is one from the `grep` family, which can select words fitting what is called a regular pattern. But I have mentioned already that the words representing elements with unique reduced expressions match such a pattern, so that we don't have to screen the output from the `ShortLex` generator, but can just use a certain automaton to produce the words, and hence the triangles, that we want. The automaton in this case turns out to be this:

```
inputno 3
stateno 13
edgeno  22
default 6

  0:  2 1,1 2,0 3,A,S
  1:  1 10,0 8,A
  2:  2 7,0 11,A
  3:  2 4,1 5,A
  4:  1 10,0 12,A
  5:  2 7,A
  6:
  7:  0 8,A
  8:  2 9,1 5,A
  9:  1 10,A
 10:  0 11,A
 11:  2 4,A
 12:  2 6,1 5,A
```

and the chambers it produces are the ones shown in the second figure.



This is a programmed version of a picture drawn by Bédard by hand several years ago, and shown to me last spring by him. A much simpler approximation to it is included in the paper [Bédard 1989].

The white triangle in the centre is just $\overline{C}$. This makes up a two-sided cell all by itself. The coloured triangles drawn are the chambers $w\overline{C}$ where w has a unique regular expression. There are three components to the set of white chambers, and each one is, according to Lusztig, a right cell of the group. These triangles are coded in shades of grey according to the generator leading into it, in order to bring out the regular structure. Together they form a two-sided cell. The holes—the white areas without triangles—are proven, in Bédard's second paper, to be what are called the right **Vogan cells** of the group. Each one is known to be a union of right cells in the sense of Kazhdan and Lusztig, and the consequence easily drawn by Bédard is that there are in fact an infinite number of right cells, which is not the case for the affine Weyl groups. There is apparently no stronger assertion known.

A natural guess is that in fact each of these 'holes' is itself a single right cell, but there is as far as I know no real evidence to support this. It is a curious fact that although the original definition of cells involves the coefficients of Kazhdan-Lusztig polynomials, and there are many groups for which the cell structure is known, it is rarely by a detailed calculation of Kazhdan-Lusztig polynomials that the cells are analyzed. There are a number of tricks which one can use instead, and they often suffice. None of them seem to work for the $(3, 4, 3)$ group, and for this reason as well as the intriguing fractal aspect of Bédard's pictures, it and similar groups possess a great deal of charm.

It is perhaps worthy of notice that regular expressions occur, without being named, in both of Bédard's papers. This should not be too surprising, since it is difficult to imagine any description of the sets pictured which does not deal with the recursive characteristics of the shapes one sees, and regular expressions are the natural way to in which to describe this kind of recursion.

Incidentally, some readers may notice that what I refer to as right cells are referred to more often in the literature as left cells. This is because my conventions on left and right actions are opposite to those adopted there.

References

A. Aho and J. Ullman, *Design and analysis of computer algorithms*, Addison-Wesley, New York, 1974.

R. Bédard, *Cells for two Coxeter groups*, Commun. Alg. **14** (1986), 1253–1286.

R. Bédard, *Left V-cells for hyperbolic Coxeter groups*, Commun. Alg. **17** (1989), 2971–2997.

N. Bourbaki, *Groupes et algèbres de Lie IV, V, VI*, Hermann, Paris, 1968.

B. Brink, *On root systems and automaticity of Coxeter groups*, Ph. D. thesis, University of Sydney, 1994.

B. Brink and R. Howlett, *A finiteness property and an automatic structure for Coxeter groups*, Math. Ann. **296** (1993), 179–190.

W. Casselman, *Machine calculations in Weyl groups*, Inv. Math. **116** (1994), 95–108.

D. Epstein, J. W. Cannon, D. F. Holt, S. Levy, M. Paterson, and W. Thurston, *Word processing in groups*, Bartlett and Jones, Cambridge, 1991.

P. Headley, *Reduced expressions in infinite Coxeter groups*, Ph. D. thesis, University of Michigan, 1994.

J. Humphreys, *Reflection groups and Coxeter groups*, Cambridge University Press, 1990.

V. Kac, *Infinite dimensional Lie algebras*, Cambridge University Press, 1985.

D. Kazhdan and G. Lusztig, *Representations of Coxeter groups and Hecke algebras*, Inv. Math. **53** (1979), 165–184.

P. Le Chenadec, *A catalogue of complete group presentations*, Jour. Symb. Comp. **2** (1986), 363–381.

G. Lusztig, *Some examples of square integrable representations of semi-simple p-adic groups*, Trans, Amer. Math. Soc. **277** (1983), 623–653.

G. Lusztig, *Cells in affine Weyl groups*, Adv. Stud. Pure Math. **6** (1985).

J.-Y. Shi, *Kazhdan-Lusztig cells in certain affine Weyl groups*, Lect. Notes. Math. **1179** (1984).

E. Vinberg, *Discrete linear groups generated by reflections*, Math. of the USSR-Izvestia **5** (1971), 1083–1119.

Mathematics Department, University of British Columbia, Vancouver, Canada
E-mail address: cass@math.ubc.ca