Computation in Coxeter groups III. Recognizing automata

Bill Casselman
Mathematics Department
University of British Columbia
Canada
cass@math.ubc.ca

Abstract. *In this, the third of a series of papers about combinatorial algorithms for arbitrary Coxeter groups, I'll discuss practical ways to compute automata recognizing* `ShortLex` *and reduced words in the generators.*

Introduction

A **Coxeter system** is a pair (W, S) where W is a group with a set of generators S and relations

$$(st)^{m_{s,t}} = 1$$

for pairs s, t in S. This is the third in a series of papers concerned with purely combinatorial algorithms for working with such groups. The earlier papers (Casselman, 2002(a)) and (Casselman, 2002(b)) were largely concerned with multiplication. This paper will be concerned with practical algorithms to produce the recognizing automata for the `ShortLex` language of words in the generators, which was first proven to be a regular language in (Brink and Howlett, 1993). Brink and Howlett did include a theoretical construction of such automata, but as I explained in (Casselman, 1994) these were very far from minimal in size, and were impractically large even for affine Weyl groups of relatively small rank. I introduced a technique called **saturation** in that paper which produced much smaller automata, but it worked only with crystallographic Coxeter groups—i.e. those possessing a realization by reflections on a lattice of roots. Only Coxeter groups with all $m_{s,t} = 2$, 3, 4, 6, or ∞ qualify. Most embarrassing is that this restriction excludes even the finite groups H_3 and H_4.

In this paper, I shall explain how to implement saturation in a purely combinatorial manner. Ironically, my current program using this new method seems to be noticeably more efficient than the original one using the integral root lattice. I begin by recalling the geometric construction of the automaton.

1. Geometry and `ShortLex` words

An expression for w in W as a product of generators in S is said to be **reduced** if it is of the least length possible. Assume S to be ordered or, equivalently, index the elements of S from 1 to $n = |S|$. I shall represent them as $\langle 1\rangle$, $\langle 2\rangle$, ..., $\langle n\rangle$. The `ShortLex` word associated to w in W is the reduced expression for w lexicographically least among all reduced expressions. Suppose w has the reduced expression $w = s_1 \dots s_n$, let $w_i = s_i \dots s_n$. This is the `ShortLex` expression for w if s_i is the least generator with $s_i w_i < w_i$. In this paper, the **normal form** $NF(w)$ of an element w will always be its `ShortLex` expression. In the earlier papers of this series I worked with an alternative normal form, `InverseShortLex` expressions, because left multiplication by a generator has a simple geometric interpretation in those terms. Similarly, I work with `ShortLex` here because they have certain useful geometric properties.

The `ShortLex` **tree** of W is a directed graph with labeled edges whose nodes are elements of W and with an edge $x \to_s y$ if $NF(y) = NF(x){\bullet}s$ for some s in S.

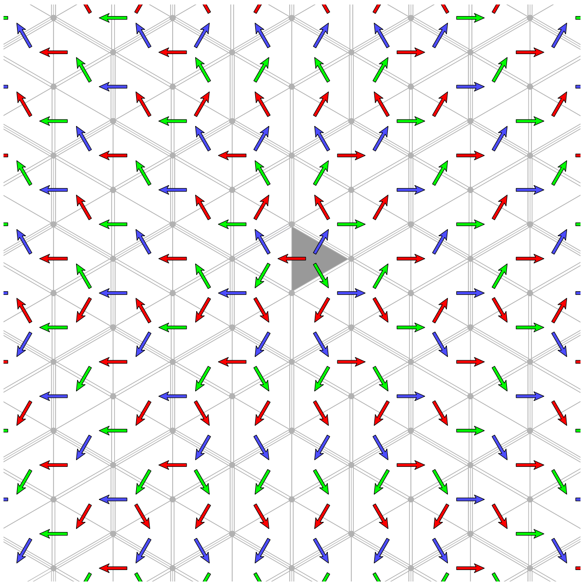

Figure 1. *The* `ShortLex` *tree for affine* A_2.

This tree has a root, the identity element of W. Suppose $s_1 \dots s_n$ to be a `ShortLex` word, and for each i let w_i be the partial product $s_1 \dots s_i$. The path

$$1 \rightarrow_{s_1} w_1 \rightarrow_{s_2} w_2 \dots \rightarrow_{s_n} w_n$$

is then a path from 1 to w in the `ShortLex` tree. Conversely, the unique path from 1 to w in the `ShortLex` tree will correspond to the `ShortLex` expression for W.

More generally, suppose A is a rooted, directed graph with edges labeled by elements of S, with the property that from each node and s there exists at most one edge with source x labeled by s. Any path in the graph starting at the root will correspond to a sequence of elements of S, namely the labels of the edges it follows. The path is said to be **labeled** by the sequence. The sequences arising in this way are said to be **recognized** by the graph. Following standard terminology in the theory of languages, the graph will be called a **recognizing automaton** for `ShortLex` if • the set of paths it recognizes are the `ShortLex` words; and • it is irredundant in the sense that every node in the graph is actually reachable by a path from its root. The `ShortLex` tree is the largest such automaton in the sense that it maps canonically onto any other.

Lemma. *If $\mathcal{A}$ is an automaton recognizing* `ShortLex` *x a node of $\mathcal{A}$, and z a word in S, the following conditions are equivalent:*

(a) There exists a path in $\mathcal{A}$ starting from x labeled by z;
(b) for all words w in the automaton leading from 1 to x, the word $w \bullet z$ is in `ShortLex`.

This should be clear.

If x is a node in an automaton $\mathcal{A}$ recognizing `ShortLex`, its **suffixes** are all the words labeling paths in $\mathcal{A}$ starting at x.

Two elements of W are said to be `ShortLex` equivalent if the corresponding nodes in the `ShortLex` tree have the same suffixes. Define $\mathcal{M}_{\mathrm{SL}}$ to be the directed graph whose nodes are equivalence classes $[w]$ of elements of W, with an edge from $[x]$ to $[y]$ labeled by s if there is an edge from x to y in the `ShortLex` tree labeled by s. It is straightforward to check that this criterion only depends on equivalence classes $[x]$ and $[y]$ and not on the particular elements x and y. The starting node of $\mathcal{M}_{\mathrm{SL}}$ is $[1]$.

Proposition. *The automaton $\mathcal{M}_{\mathrm{SL}}$ recognizes* `ShortLex`*. If $\mathcal{A}$ is any recognizing automaton for* `ShortLex`*, there exists a unique map from $\mathcal{A}$ to $\mathcal{M}$ taking start node to start node and mapping each edge to another one with the same label.*

PROOF. Any `ShortLex` word gives rise to a path in the `ShortLex` tree which maps onto a path in $\mathcal{M}_{\mathrm{SL}}$. It has to be shown that any path in $\mathcal{M}_{\mathrm{SL}}$ is labeled by a `ShortLex` word. This I do by induction on its length. Suppose $y = x{\bullet}s$ is a path in $\mathcal{M}_{\mathrm{SL}}$. The induction assumption allows us to assume that x is labeled by an `ShortLex` word, and therefore corresponds to a path in the `ShortLex` tree. But by definition of the edges in $\mathcal{M}_{\mathrm{SL}}$, $x{\bullet}s$ is in `ShortLex`, too. Therefore $\mathcal{M}_{\mathrm{SL}}$ recognizes `ShortLex`.

Define the **depth** of a node to be the length (i.e. number of edges) of the shortest path to it from the start node. The map σ is to be constructed by recursion on depth. The start in $\mathcal{A}$ is mapped to the start in $\mathcal{M}$. Suppose the map is defined on all nodes of depth n or less, and suppose y is a node of depth $n+1$. Then there exists a path u from 1 to y of length n; suppose x to be the source of the last edge, which is labeled by s, and v the path to x. Then $w = u{\bullet}s$ is in `ShortLex`, and since we know that $\mathcal{M}_{\mathrm{SL}}$ recognizes `ShortLex`, there must be an edge leading from $\sigma(x)$ labeled by s, and $\sigma(y)$ must be its target. This proves uniqueness.

The existence of the map is best shown directly. Let x be a node in $\mathcal{A}$, u a path to x. The suffixes of u are the suffixes of x, so the equivalence class of u is well defined. Map x to $[u]$.□

For x in W, let the `ShortLex` **continuation** SL_x of x be the set of all y in W with the property that $NF(y) = NF(x){\bullet}z$ for some reduced expression z—the normal form of x is a prefix in the normal form of y. The empty string is the `ShortLex` expression for 1, so x itself lies in SL_x. The continuation of x consists of all y one can reach by following such directed edges.

We need also a geometric version of this. Suppose W is realized by a Cartan matrix in the vector space V, that Δ is the set of basic roots, and that C the fundamental chamber $\alpha > 0$ for α in Δ, which is a simplicial cone. The Tits cone $\mathcal{C}$ of the realization is the union of the closures $w\overline{C}$. It is a convex set. For each x in W let $\mathcal{C}_x$ be the union of the $y\overline{C}$ with y in SL_x.

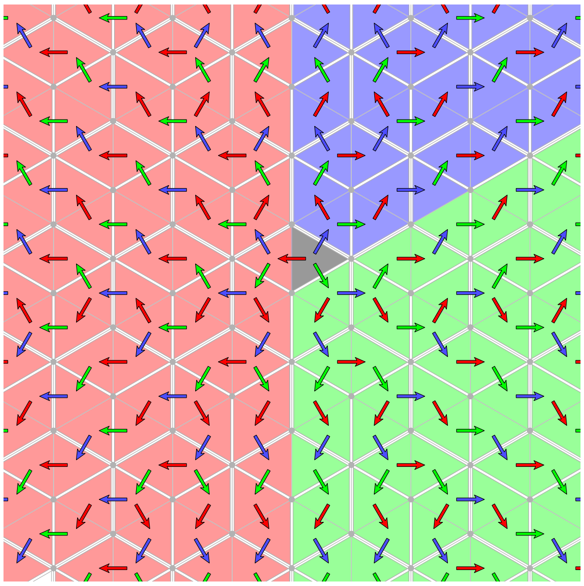

Figure 2. *The domains $\mathcal{C}_{\langle 1\rangle}$, $\mathcal{C}_{\langle 2\rangle}$, $\mathcal{C}_{\langle 3\rangle}$.*

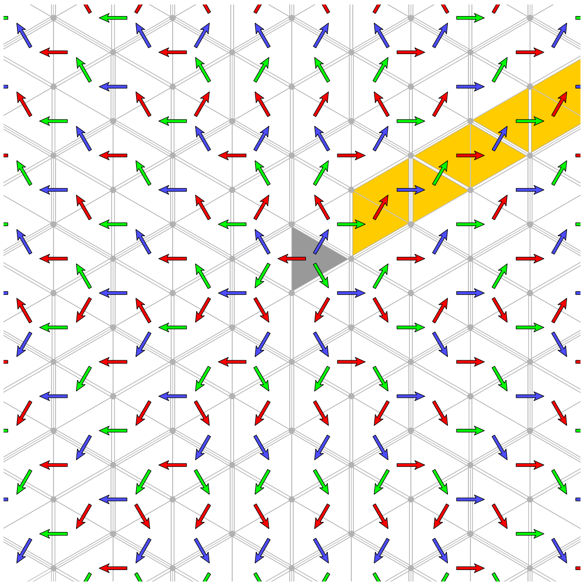

Figure 3. *The domain $\mathcal{C}_{\langle 3\rangle\langle 2\rangle}$.*

The fundamental relationship between `ShortLex` expressions and geometry is that the sets $\mathcal{C}_x$ are also convex. We shall need a more explicit version of this later on:

Proposition.

(a) $\mathcal{C}_1 = \mathcal{C}$;

(b) for $1 \le i \le |S|$, *the set* $\mathcal{C}_{\langle i\rangle}$ *is the intersection of* $\mathcal{C}$ *with*

$$\alpha_{\langle i\rangle} \le 0,\ \langle i\rangle\, \alpha_{\langle j\rangle} \ge 0 \text{ for all } j < i\ .$$

(c) for any x in W and s in S, $NF(x)\bullet s$ is a `ShortLex` *expression if and only if xs lies in SL_x, and in these circumstances*

$$\mathcal{C}_{xs} = \mathcal{C}_x \cap x\mathcal{C}_s\ .$$

PROOF. The first claim is trivial. If $x = \langle 1\rangle$, then since $\langle 1\rangle$ is the least element of S the continuation of x is made up of all the y such that $\langle 1\rangle y < y$ or $y^{-1}\alpha_{\langle 1\rangle} < 0$, which means that $\mathcal{C}_{\langle 1\rangle}$ is the intersection of $\mathcal{C}$ with the half-space $\alpha_{\langle 1\rangle} \le 0$. For any $i \le |S|$, the `ShortLex` expression of x begins with $\langle i\rangle$ if $\langle i\rangle x < x$ but $\langle j\rangle x > x$ for all $j < i$. This proves (b). As for (c), the `ShortLex` expression for y begins with $NF(x)\bullet s$ if and only if it begins with $NF(x)$ and the `ShortLex` expression for $x^{-1}y$ begins with s.□

Call two elements x and y of W **congruent** if the two sets $\mathcal{C}_x$ and $\mathcal{C}_y$ are strongly congruent in the sense that some element of W transforms $\mathcal{C}_x$ to $\mathcal{C}_y$ and xC to yC. Of this element is none other than yx^{-1}. Another way to put this is to say that the two set $x^{-1}\mathcal{C}_x$ and $y^{-1}\mathcal{C}_y$ are the same. The set $\mathcal{S}_w = w^{-1}\mathcal{C}_w$ therefore depends only on the congruence class of w.

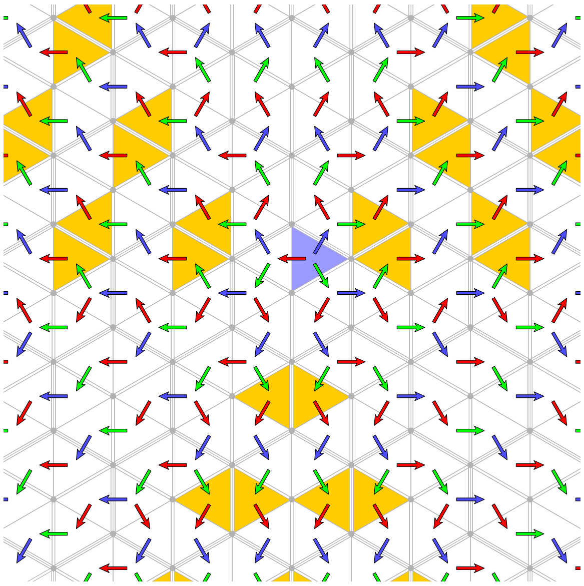

Figure 4. *A congruence class.*

Proposition. *Suppose x and y in W. Then $NF(x)$ and $NF(y)$ are equivalent if and only if x and y are congruent. The word $NF(x){\bullet}s$ is in* `ShortLex` *if and only if sC is contained in $\mathcal{S}_x$, and in this case*

$$\mathcal{S}_{xs} = s\mathcal{S}_x \cap \mathcal{S}_s \ .$$

This follows immediately from the previous Proposition.

A **geometric root** of the realization of a Coxeter group W is the W-transform of a half-space $\alpha \geq 0$, α in Δ. In this paper all roots will be geometric. A (geometric) root is called positive if it contains C, otherwise negative. All roots are either positive or negative. One positive root **dominates** another if it contains it. A minimal root is one which dominates only itself. It is a remarkable result of Brink and Howlett that the number of minimal roots is finite.

If s is in S then s applied to a minimal root λ is either the half-space $\alpha \leq 0$, in which case $\lambda = \alpha_s$ and I set $s\lambda = \ominus$; another minimal root μ; or a positive root which dominates $\alpha \geq 0$, in which case I write $s\lambda = \oplus$. The table recording these data is called the minimal root reflection table. The minimal root reflection table is a fundamental structure for a Coxeter group, and as evidence for this in earlier papers I extended a result of Brink and Howlett by showing that it can be used in an efficient multiplication algorithm. The minimal roots were devised by Brink and Howlett to prove this:

Proposition. *There are only a finite number of equivalence classes in the* `ShortLex` *language of any Coxeter group.*

This is a corollary of:

Proposition. *For each w in W let Λ_w be the set of minimal roots λ containing $\mathcal{S}_w$. Then Λ_w is finite and*

$$\mathcal{S}_w = \bigcap_{\lambda \in \Lambda_w} \lambda \ .$$

PROOF. The sets $\mathcal{S}_s$ are defined by the inequalities

$$\alpha_{\langle i \rangle} \geq 0 \, , \quad \langle i \rangle \alpha_{\langle j \rangle} \geq 0 \text{ for all } j < i$$

and the Proposition is certainly true for them.

The proof now proceeds by induction on the length of x. Suppose true for x. Then $NF(x)\bullet s$ lies in `ShortLex` if and only if sC is contained in $\mathcal{S}_x$, and then $\mathcal{S}_{xs} = s\mathcal{S}_x \cap \mathcal{S}_s$. Since $\alpha_s \geq 0$ on C but ≤ 0 on sC, the condition that sC be contained in $\mathcal{S}_x$ is equivalent to the condition that α_s not be in Λ_x. Therefore if $\lambda \geq 0$ on $\mathcal{S}_x$, $s\lambda$ is again positive. It is either minimal, or it dominates α_s. But α_s is ≥ 0 on $\mathcal{S}_s$, so the condition that $s\lambda \geq 0$ on $\mathcal{S}_{xs}$ is redundant.◻

We can now define a new version of the automaton whose nodes are the sets $\mathcal{S}_x$ to be one whose nodes are the sets Λ_x, with an edge from Λ_x to Λ_y if $y = xs$ and $\alpha_s \notin \Lambda_x$. Theoretically this is fine, and with linear programming one can even find it explicitly, although unless W is crystallographic there will be severe technical difficulties.

Linear programming is a heavy burden to impose on this construction, however. Brink and Howlett defined a much simpler finite automaton made up of the subsets Θ_x of the minimal roots defined inductively by the formulae

$$\begin{aligned} \Theta_1 &= \emptyset \\ \Theta_{\langle i\rangle} &= \{\alpha_{\langle i\rangle}\} \cup \langle i\rangle \bigcup_{j<i} \{\alpha_{\langle j\rangle}\} \\ \Theta_{xs} &= s\Theta_x \cup \Theta_s \quad \text{ whenever } \alpha_s \notin \Theta_x \ . \end{aligned}$$

Then $\lambda \geq 0$ for $\lambda \in \Theta_x$ determines $\mathcal{S}_x$, and this automaton maps to the one defined by the sets $\mathcal{S}_x$. This automaton is, however, very large—several of the sets Θ_x will determine the same set $\mathcal{S}_x$. The automaton is in fact impractically large for even reasonably small Coxeter groups. I recall in the next section how to construct a smaller one along the same lines.

2. Saturation

The problem with the above construction is that the sets Θ_x by no means contain all the minimal roots λ with $\lambda \geq 0$ on $\mathcal{S}_x$. For example, I have defined

$$\Theta_{\langle i\rangle} = \alpha_{\langle i\rangle} \cup \bigcup_{j<i} \langle i\rangle\, \alpha_{\langle j\rangle}$$

but the set of all minimal roots

$$\Lambda_{\langle i\rangle} = \{\lambda \text{ a minimal root } \mid \lambda \geq 0 \text{ on } \mathcal{S}_{\langle i\rangle}\}$$

is much larger. If $\Sigma^{\min}_{j<i}$ is the set of minimal roots for the Coxeter group generated by the $\langle j\rangle$ with $j < i$, then

$$\Lambda_{\langle i\rangle} = \{\alpha_{\langle i\rangle}\} \cup \langle i\rangle\, \Sigma^{\min}_{j<i} \ .$$

In othr words, we want some simple way to enlarge the Brink-Howlett sets Θ_x. In the induction step I set

$$\Theta^*_{xs} = s\Theta^*_x \cup_* \Theta^*_s$$

where $\cup_*$ means an operation which I call the **partial saturation**. The original observation of (Casselman, 1994) was that if λ and μ are linear functions with $\lambda \geq 0$ and $\mu \geq 0$ on $\mathcal{S}_x$, then so is $\lambda + \mu$, and in case λ lies in $s\Theta_x$, μ lies in Θ_s, and $\lambda + \mu$ is a minimal root, then it is also in Λ_{xs}. This observation was made in a context where the group was crystallographic, so that λ and μ were both integral linear combinations of elements of Δ, and the sum of roots was simple to calculate. Arbitrary Coxeter groups do not possess realizations stabilizing a lattice, and implementing this idea would involve some awkward cyclotomic arithmetic. It turns out, however, that there exists a purely combinatorial algorithm to effect the same partial saturation. It was suggested by results of Matthew Dyer.

The sums of roots doesn't really come into it. What we really want to do is find, for all minimal roots λ and μ the set of all minimal roots ν which dominate the pair λ and μ in the sense that $\nu \ge 0$ on the region where both $\lambda \ge 0$, and $\mu \ge 0$. This is, as Dyer emphasizes, a calculation in a Coxeter system of rank two.

For the rest of this section, let λ and μ be arbitrary minimal roots, $s = s_\lambda$ and $t = s_\mu$ the corresponding root reflections.

The first thing to do is decide whether the root reflections s and t generate a finite group or not. We can test this by calculating products $(st)^n$ up to M, the maximum finite value of $m_{s,t}$. If they don't generate a finite group, there is nothing more to be done. If they do, on the other hand, the group they generate will be a Coxeter group of rank two and order n, where n is the order of st. We know everything about such groups.

For efficiency, instead of finding the rotation st and then deciding if $(st)^n = 1$ for $n \le M$, we calculate and store all the elements 1, s, st, sts, $\dots$, $(st)^n$, $(st)^n s$, $\dots$ up to $(st)^M$ if necessary. Each element in this list is the prevous element multiplied by either s or t. We stop if it happens that $(st)^n = 1$. So at the end we have either $(st)^n = 1$ or $n = M$ but $(st)^n \ne 1$. In the second case, or if $n = 2$, our list just contains s and t themselves. At any rate, if $W_{s,t}$ is finite then we now have a list of all its elements. The root reflections are the 'odd' half, of the form $(st)^k s$ for $0 \le k < n$. The reflection $(st)^k s$ corresponds to the root $(st)^{ell}\lambda$ if $k = 2\ell$, and to $(st)^{ell} s\mu$ if $k = 2\ell + 1$. The root hyperplanes will make up an arrangement of hyperplanes all intersection in one linear space of codimension 2. They will partition space into $2N$ regions contained in between pairs of hyperplanes intersecting at an angle π/N. Exactly one of regions, say $C_{s,t}$, will contain the fundamental chamber C of W. It will be a fundamental domain for $W_{s,t}$. Let σ and τ be the reflections in the walls of $C_{s,t}$.

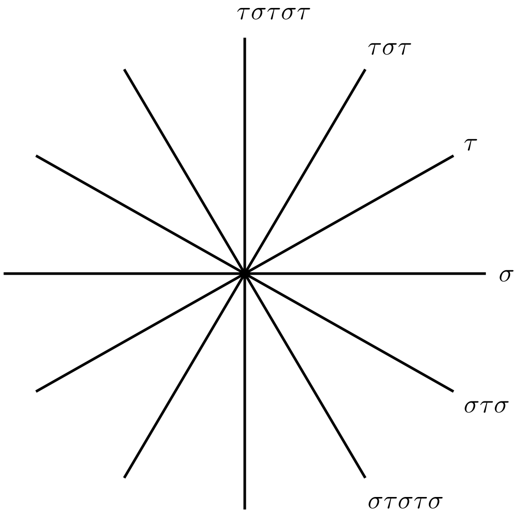

Figure 5. *Root reflection lines.*

Dyer calls these the **canonical generators** of the group $W_{s,t}$. Let λ_σ and λ_τ be the corresponding positive roots. Dyer characterizes the canonical generators in a couple of ways, but for our purposes there is a more convenient combinatorial characterization than his.

Proposition. *One of the canonical generators σ and τ will be of minimal length among the reflections $\rho = (st)^k s$. If it is σ then τ will be of least length among those with $\sigma\rho > \rho$.*

PROOF. A root reflection has length $2n + 1$ if the root has depth n. Any gallery to one of the regions $\lambda < 0$ for λ in the dihedral root system spanned by λ_σ and λ_τ will have to cross one of the walls $\lambda_\sigma = 0$ or $\lambda_\tau = 0$. This proves the first claim. The second follows similarly from the diagram.□

At this point we know the k and ℓ such that

$$(st)^k s = \sigma, \quad (st)^\ell s = \tau \ .$$

A bit of number theory will now give us k' and ℓ' such that

$$(\sigma\tau)^{k'} \sigma = s, \quad (\sigma\tau)^{\ell'} \sigma = t \ .$$

And a bit of geometry will give us the roots dominating s and t.

If s is a reflection in 2D, say its line of reflection has angular coordinate (with respect to the x-axis) of α. Then any line at angle $\alpha+\beta$ is reflected by s to one at angle $\alpha-\beta$; and one at angle $\beta = \alpha + (\beta-\alpha)$ to one at angle $\alpha - (\beta-\alpha) = 2\alpha - \beta$.

Suppose s and t to be reflections, with an angle of θ between their reflection planes—i.e. rotate by θ to get from the reflection line for s to that for t. Then st is rotation by -2θ. Say st has order N, so that $N\theta$ is an integral multiple of π. All the reflections in the group generated by s and t are of the form $(st)^k s$ where $0 \le k < N$. If the angle of the line of reflection of s is α then that of $w = (st)^k s$ is $\alpha - k\theta$, since w takes the line at angle $\alpha+\beta$ first to one at angle $\alpha-\beta$ and then to one at angle

$$\alpha - \beta - 2k\theta$$

and if $\beta = -k\theta$ this is again $\alpha+\beta$. For $k = N-1$ or equivalently $k=-1$ we get t, which makes sense because the line for t is forward θ from that for s.

A pair of reflections in this group, say with angle β between their lines of reflection, will generate the whole group if and only if

$$\beta = \frac{\pi k}{N} = k\gamma \quad (\gamma = \pi/N)$$

with $\gcd(k,N) = 1$.

We want to recover from this how to express s and t in terms of σ and τ. Let $s = (\sigma\tau)^{k'}\sigma$, $t = (\sigma\tau)^{\ell'}\sigma$. We want to find k', ℓ' from k and ℓ.

We can choose our coordinate system so that the angle for $\sigma = 0$, that for τ is γ. The angle of s is $\gamma - k'\gamma$, that of t is $\gamma - \ell'\gamma$, that between s and t is $(k'-\ell')\gamma$. The angle of $(st)^k s$, which is σ, is $\gamma + \ell'\gamma + k(k'-\ell')\gamma$, that of $(st)^\ell s$, which is τ, is $\gamma + \ell'\gamma + \ell(\ell'-k')\gamma$. So to find k', ℓ' we have modulo $2\pi Z$

$$\begin{aligned} \gamma + \ell'\gamma + k(\ell'-k')\gamma &= 0 \\ \gamma + \ell'\gamma + \ell(\ell'-k')\gamma &= \gamma \end{aligned}$$

and modulo N

$$\begin{aligned} \ell' + k(\ell'-k') &= -1 \\ \ell' + \ell(\ell'-k') &= 0 \end{aligned}$$

$$\begin{aligned} -kk' + (1+k)\ell' &= -1 \\ -\ell k' + (1+\ell)\ell' &= 0 \end{aligned}$$

$$\begin{bmatrix} -k & 1+k \\ -\ell & 1+\ell \end{bmatrix} \begin{bmatrix} k' \\ \ell' \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix} .$$

The determinant of the coefficient matrix is $\ell - k$, which is prime to N, so we can solve this system of equations. Let $D = \ell - k$. Then we have

$$\begin{bmatrix} -k & 1+k \\ -\ell & 1+\ell \end{bmatrix}^{-1} = \frac{1}{D} \begin{bmatrix} 1+\ell & -1-k \\ \ell & -k \end{bmatrix}$$

which makes

$$\begin{bmatrix} k' \\ \ell' \end{bmatrix} = \begin{bmatrix} -(1+\ell)/D \\ -\ell/D \end{bmatrix}$$

again all modulo N.

We want to find the roots that dominate the region where $\alpha_s > 0$, $\alpha_t > 0$. They are the ones corresponding to $(\sigma\tau)^{m'}\sigma$ with m' between k' and ℓ' (inclusive). But now we don't want to calculate all those reflections all over again; we want a translation $(\sigma\tau)^{m'}\sigma = (st)^m t$—i.e. to find m given m'. Equating angles we get

$$\gamma + m'\gamma = \gamma + \ell'\gamma + m(\ell' - k')\gamma, \quad m' - \ell' = m(\ell' - k'), \quad m = \frac{m' - \ell'}{\ell' - k'} = D(m' - \ell') \ .$$

Here we have

$$\ell' - k' = \frac{-\ell + 1 + \ell}{D} = \frac{1}{D}, \quad \frac{1}{\ell' - k'} = D = \ell - k \ .$$

This checks: for t itself, when $m = 0$ and $m' = \ell'$, and for s, when $m = -1$ and $m' = k'$.

3. Constructing the automata

Given the saturation process, I'll briefly recall how to build an automaton recognizing `ShortLex` of a Coxeter group.

When the algorithm begins, we have at hand the minimal root reflection table. States of the `ShortLex` automaton are subsets of minimal roots, namely those bounding the region $\mathcal{S}_w$. We start with the identity element, which is the empty set, as well as the Θ_s^*. We put these on a stack, and while that stack is empty we remove elements from it. When an element Θ is removed, for each s with α_s not in Θ we build the partially saturated set

$$\Xi = (\text{minimal roots in } s\Theta) \cup_* \Theta_s^*$$

where $\cup_*$ means that for each pair λ, μ with λ in Θ and μ in Θ_s^* we add all the ν dominating them. To make this somewhat more efficient, we can calculate and store in advance the sets of elements dominating λ, μ with μ in Θ_s^*—i.e. one for each minimal root λ and s.

We maintain a look-up table of all nodes, and we look up Ξ in it. If it has already been defined, we add a link from Θ to it, because there will be an edge from Θ to Ξ. Otherwise we register Ξ in the look-up table, and put it on the stack. Repeat as often as there is something on the stack to be examined.

Finally, after we have calculated a recognizing automaton in this way we can find the minimal equivalent automaton by the classical and extremely efficient procedure of (Hopcroft and Ullman, 1969).

4. Reduced words

There is a finite automaton for recognizing reduced words as well as one for recognizing `ShortLex`. It is a lot bigger—for a finite group it is just the `ShortLex` tree itself—but theoretically of much more importance. The construction is almost excatly the same as that of the `ShortLex` automaton, except for the definition of the initial states

$$\Theta^{\mathrm{red}}_{\langle i \rangle} = \{\alpha_{\langle i \rangle}\} \ .$$

References

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

2. Bill Casselman, 'Machine calculations in Weyl groups', *Inventiones Mathematicae* **116** (1994), 95–108.

3. Bill Casselman, 'Computation in Coxeter groups I. Multiplication', preprint, 2001. To appear in the *Electronic Journal of Combinatorics*, 2002. Referred to here as (Casselman, 2002(a)).

4. Bill Casselman, 'Computation in Coxeter groups II. Minimal roots', preprint, 2002. Submitted to the *Electronic Journal of Combinatorics*, 2002. Referred to here as (Casselman, 2002(b)).

5. M. Dyer, 'Reflection subgroups of Coxeter systems', *Journal of Algebra* **135** (1990), 57–73.

6. John Hopcroft and J. D. Ullman, *Formal languages and their relation to automata*, Addison-Wesley, 1969.