Coxeter groups homework 2
Alex Fink

4 We prove this in very much the same spirit as the implication that the exchange property is sufficient to be a Coxeter system. We’ll say two reduced expressions are *interconvertible* if one can be converted into the other by a sequence of replacements of substrings $ss's's'\cdots$ by $s's's's'\cdots$, both of length $m(s, s')$. Note that this is an equivalence relation, in particular transitive.

Let $s_1\cdots s_k$ and $s'_1\cdots s'_k$ be two reduced words for some $w \in W$. We proceed by induction on $k$. Taking $k = 0$ as our base case, say, there’s nothing to do. Otherwise, by exchange, since $s'_1s'_1\cdots s'_k = s'_2\cdots s'_k$, we have $s'_1s_1\cdots s_k = s_1\cdots \hat{s}_i\cdots s_k$ for some $i$, i.e.

$$s_1s_2\cdots s_k = s'_1s_1\cdots \hat{s}_i\cdots s_ks'_1\cdots s'_k.$$  

(1)

Now, $s'_1s_1\cdots \hat{s}_i\cdots s_k$ and $s'_1\cdots s'_k$ are both reduced expressions, and accordingly so are $s_1\cdots \hat{s}_i\cdots s_k = s'_2\cdots s'_k$ which by induction are interconvertible. Adding an initial $s'_1$ doesn’t affect any of the necessary replacements, so $s'_1s_1\cdots \hat{s}_i\cdots s_k$ and $s'_1\cdots s'_k$ are interconvertible as well.

As for the left equality in (1), we can fall back on induction to show that $s_1s_2\cdots s_k$ and $s'_1s_1\cdots \hat{s}_i\cdots s_k$ are interconvertible so long as they have any common suffix, i.e. $i < k$. Then, by transitivity, we’d be done. So it remains to handle the case $i = k$, that is $s_1s_2\cdots s_k = s'_1s_1\cdots s_{k-1}$.

In this situation, we exchange the roles of $s'_1s_1\cdots s_{k-1}$ and $s_1s_2\cdots s_k$ and start again. Either we finish the proof by a breakdown like (1), or else we come to this same point in the proof again and get

$$s'_1s_1\cdots s_{k-1} = s_1s'_1s_1\cdots \hat{s}_{k-1} = s_1s'_1s_1\cdots s_{k-2}.$$  

Iterating further, for altogether $k - 1$ steps, either we finish or we come to the conclusion that

$$\cdots s'_1s_1s'_1s_1 = \cdots s_1s'_1s_1s'_1$$  

(2)

both of length $k$.

Now, we know that the order of $(s_1s'_1)$ is $m(s_1, s'_1)$. Our last inequality can be rewritten $(s_1s'_1)^k = 1$, so we find $m(s_1, s'_1) \mid k$. Accordingly the two sides of (2) are interconvertible, by $k/m(s_1, s'_1)$ replacements of the acceptable form. This at last finishes the proof.
We proceed inductively, converting each suffix $s_i \cdots s_k$ of this word to an equivalent reduced word by means of our two permissible kinds of replacement. Proceeding in this way we’ll eventually convert the whole word $s_1 \cdots s_k$ to a reduced word; but as $s_1 \cdots s_k = e$ this must be the empty word.

We can start with the empty suffix, for which we’re vacuously finished. Otherwise, suppose $s'_1 \cdots s'_j$ is a reduced word for $s_1 \cdots s_k$. Now, $l(s_{i-1}s'_1 \cdots s'_j) = l(s'_1 \cdots s'_j) \pm 1$. If the sign here is $+$, then $s_{i-1}s'_1 \cdots s'_j$ is already reduced, and we’re done the inductive step without any further replacements. Otherwise $s_{i-1}s'_1 \cdots s'_j$ has some reduced word $w$ of length $l(s'_1 \cdots s'_j) - 1$, so that $l(s_{i-1}w) = l(s'_1 \cdots s'_j)$ and thus $s_{i-1}w$ is a reduced word for $s'_1 \cdots s'_j$. By the result of problem 4, $s'_1 \cdots s'_j$ can be converted to $s_{i-1}w$ by making only replacements of the form $ss'ss' \cdots \to s'ss'$ while. Performing these replacements on the tail of $s_{i-1}s'_1 \cdots s'_j$ yields $s_{i-1}s_{i-1}w$, and then a single deletion of $s_{i-1}s_{i-1}$ yields the reduced word $w$, as our inductive hypothesis demanded.

Accordingly, we have the following (naïve and atrocious, but at least terminating) algorithm for the word problem in a Coxeter group. Given a word $w$ of length $k$, make all possible replacements of the two permissible kinds, repeatedly, until there are no more replacements that yield a word we haven’t already seen; then conclude $w = e$ if and only if we have seen the empty word.

Since no permissible replacement lengthens the word, we will see at most all words over $\{s_1, \ldots, s_n\}$ of length $\leq k$, and there are finitely many of these. So we see all possible words obtainable from $w$ by permissible replacements in finite time, and we know when this happens. By what we’ve just done, the empty word will appear among these if $w = e$ in the group, and it certainly appears only if $w = e$ since all permissible replacements come from relations in the group. Therefore our algorithm is correct.

Let $E_i$ and $F_i$ be complete flags in $\mathbb{R}^n$. It will make the argument slightly more natural to consider our tables to include zero-indexed entries as well. Since $E_0$ and $F_0$ are each just the origin, we have $\dim(E_0 \cap F_i) = \dim(E_i \cap F_0) = 0$, and the corresponding entries in a partition rank table are evidently also zero.

For any $1 \leq i, j \leq n$ we have
\begin{equation}
    d_{i,j} := \dim(E_i \cap F_j) + \dim(E_{i-1} \cap F_{j-1}) - \dim(E_i \cap F_{j-1}) - \dim(E_{i-1} \cap F_j) \geq 0.
\end{equation}

We could make improvements like rejecting immediately if $k$ is not even, but they’re hardly necessary merely to get an algorithm.