Posets of twisted involutions in Coxeter groups

Verbände getwisteter Involutionen in Coxetergruppen

Abschlussarbeit zur Erlangung des akademischen Grades
Master of Science (M. Sc.) im Studiengang Mathematik

eingereicht von

Christian Hoffmeister (B. Sc.)

bei der
Technischen Universität Braunschweig
am
Institut Computational Mathematics

Erstprüfer:	Professor Dr. Harald Löwe
Zweitprüfer:	Professor Dr. Max Horn

Matr.-Nr:	2944344
E-Mail:	choffmeister@choffmeister.de
Adresse:	Laufkötterweg 9c, 22119 Hamburg

Tag der Anmeldung:	01.06.2012
Tag der Einreichung:	

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Masterarbeit „Verbände getwisteter Involutionen in Coxetergruppen“ selbstständig verfasst sowie alle benutzten Quellen und Hilfsmittel vollständig angegeben habe und dass die Arbeit nicht bereits als Prüfungsarbeit vorgelegen hat.

Ort, Datum | Christian Hoffmeister

Deutsche Zusammenfassung

Regelmäßige Flächen und Körper sowie ihre Symmetriegruppen sind in der Mathematik seit jeher von großem Interesse. Mit der Einführung von Coxetergruppen durch Harold Scott MacDonald Coxeter im Jahr 1934 konnten diese Symmetriegruppen abstrakt gefasst und zugleich eine deutlich größere Klasse von Gruppen untersucht werden. Heutzutage spielen Coxetergruppen in vielen Bereichen der Mathematik eine große Rolle.

Für ein sogenanntes Coxetersystem (W, S), bestehend aus einer Menge von involutorischen Erzeugern S und der von ihnen erzeugten Gruppe W, sei θ ein Automorphismus von W der S festhält und maximal Ordnung 2 hat. Dann heißt die Menge $\mathcal{I}_\theta$ der Elemente $w \in W$, die von θ auf ihr Inverses abgebildet werden, die Menge der θ-getwisteten Involutionen. Es existiert dann eine spezielle Abbildung $(w, s) \mapsto w\underline{s}$, welche die Eigenschaft hat, dass der Orbit des neutralen Elements von W bzgl. dieser Abbildung gerade die Menge der θ-getwisteten Involutionen ist und mit dessen Hilfe sich eine bestimmte Halbordnung $\preceq$ auf dieser Menge definieren lässt. Der Verband $(\mathcal{I}_\theta, \preceq)$ heißt dann getwistete schwache Ordnung $Wk(W, \theta)$. Für ein Element $w \in \mathcal{I}_\theta$ und eine Teilmenge von Erzeugern $S' \subseteq S$ heißt die Menge aller getwisteten Involutionen, die von der Form $w\underline{s}_1 \dots \underline{s}_n$ mit $s_1, \dots, s_n \in S'$ sind, das S'-Residuum von w, geschrieben als $wC_{S'}$.

Im Rahmen dieser Arbeit heißt $Wk(W, \theta)$ 3-residuell zusammenhängend, falls Folgendes gilt: Seien $K, S_1, S_2, S_3 \subseteq S$ Mengen von Erzeugern, wobei K sphärisch ist und von θ festgehalten wird und sich die S_1, S_2, S_3 paarweise nicht leer schneiden. Weiter sei w_K das maximale Element im Residuum wC_K. Dann gilt

$$wC_{S_1} \cap wC_{S_2} \cap wC_{S_3} \subseteq wC_{S_1 \cap S_2 \cap S_3}.$$

Die offene Fragestellung, um die es in dieser Arbeit gehen soll, ist, für welche Paare $((W, S), \theta)$ die getwistete schwache Ordnung 3-residuell zusammenhängend ist. Um dies zu überprüfen, wird in dieser Arbeit nach einer Einleitung in die Theorie zuerst ein effizienter Algorithmus entwickelt, um den Verband $Wk(W, \theta)$ berechnen zu können. Dann werden diese Ergebnisse benutzt, um mit einem weiteren Algorithmus nach Gegenbeispielen für den 3-residuellen Zusammenhang zu suchen. Dabei können durch weitere theoretische Vorarbeit auch bestimmte unendliche Coxetersystem untersucht werden. Zum Abschluss der Arbeit wird dann noch ein kurzer Exkurs in die Gebäudetheorie gemacht. Dabei wird gezeigt, dass für alle $Wk(W, \theta)$, für die der 3-residuelle Zusammenhang gezeigt werden konnte, auch gilt, dass für jedes sogenannte Zwillingsgebäude vom Typ W das sogenannte Flipflop-System $\mathcal{C}^\theta$ residuell zusammenhängend ist, sofern der Flip θ noch eine weitere Bedingung erfüllt.

Contents

List of Figures

Introduction

Within this paper we investigate structural properties of the so-called twisted weak ordering $Wk(W, \theta)$. In Chapter 1 we establish some well-known definitions and facts on posets, Coxeter systems and the Bruhat ordering. In Chapter 2 we build up the theory of twisted involutions. For a Coxeter system (W, S) and an involutory Coxeter system automorphism θ the set of θ-twisted involutions is defined as

$$\mathcal{I}_\theta := \{w \in W : \theta(w) = w^{-1}\}.$$

This set represents some kind of generalization of ordinary Coxeter groups (cf. Example 2.4). Many properties of Coxeter groups can be transfered to the θ-twisted involutions. Elements w in ordinary Coxeter groups have representations $w = s_1 \dots s_n$ with involutory generators. There is an analogue construction for θ-twisted involutions, too. Let $\underline{S}$ be set of symbols with same cardinality as S. Then define an action

$$w\underline{s} := \begin{cases} ws & \text{if } \theta(s)ws = w, \\ \theta(s)ws & \text{else.} \end{cases}$$

and extend this action to the whole free monoid over $\underline{S}$ by

$$w\underline{s}_1\underline{s}_2 \dots \underline{s}_k := (\dots((w\underline{s}_1)\underline{s}_2)\dots)\underline{s}_k.$$

It turns out that every expression $e\underline{s}_1 \cdots \underline{s}_n$, called twisted expression, admits a θ-twisted involution and in return that every θ-twisted involution has such a twisted expression representing it. In addition, $w\underline{s}\underline{s} = w$ holds for any θ-twisted involution w and $\underline{s} \in \underline{S}$ just as $wss = w$ holds for any $w \in W$ and $s \in S$. For these twisted expressions there are analogue versions of the Exchange Condition, the Deletion Condition and the Lifting Property. Also an analogue concept to the length l for elements from Coxeter groups exists for θ-twisted involutions. The twisted length ρ is defined as the smallest possible length of a twisted expression representing a twisted involution.

The Bruhat ordering for W can be restricted to the θ-twisted involutions. This restricted Bruhat ordering has another subposet: the twisted weak ordering. It is by the relation

$$v \preceq w \iff w = v\underline{s}_1 \cdots \underline{s}_k \wedge k = \rho(w) - \rho(v).$$

The twisted weak ordering $(\mathcal{I}_\theta, \preceq)$ is denoted by $Wk(W, \theta)$. It is the main object of interest in this paper. In order to develop an efficient algorithm to calculate this poset we further inspect structural properties of the poset. The so-called I-residues, which are subsets of twisted involutions of type

$$wC_I := \{w\underline{s}_1 \cdots \underline{s}_n : w \in \mathcal{I}_\theta, \underline{s}_i \in \overline{I} \subseteq \overline{S}\},$$

are investigated for some type of invariants. In particular the I-residues with $|I| = 2$, the rank-2-residues, have some very useful and interesting constraints for their possible structure. After having investigated their structure in detail we use these results to massively improve a known algorithm for calculating $Wk(W, \theta)$ in Section 2.4. Indeed, we develop an algorithm that has an asymptotical perfect runtime behavior.

In Chapter 3 we address something that is called 3-residually connectedness in this paper. For a set $K \subseteq S$ that is fixed by θ and three sets $S_1, S_2, S_3 \subseteq S$ we ask if

$$wC_{S_1 \cap S_2} \cap wC_{S_2 \cap S_3} \cap wC_{S_3 \cap S_1} \subseteq wC_{S_1 \cap S_2 \cap S_3}$$

holds. If this holds for all K, S_1, S_2, S_3, then we call $Wk(W, \theta)$ 3-residually connect. In Section 3.1 and Section 3.2 we investigate some special configurations. Since it refused to be proven in general or at least for certain types of W or $\theta = \mathrm{id}$ we use the $Wk(W, \theta)$-algorithm to programmatically check if 3-residually connectedness holds in Section 3.3.

We finish the paper with an excursion to building theory in Chapter 4. As it turns out the 3-residually connectedness of $Wk(W, \theta)$ allows to deduce the so-called residually connectedness of so-called flip-flop systems of type (W, S) with flip θ, at least when assuming one additional property for the flip.

1. Preliminaries

We start up with collecting some definitions and facts to ensure an uniform terminology and state of knowledge.

1.1. Posets

Posets are sets M with a partial order $\leq$. In particular, there are pairs $(a, b) \in M \times M$ of distinct elements such that neither $a \leq b$ nor $a \geq b$. The following definitions and examples define this more precisely.

Definition 1.1. Let M be a set. A binary relation $\leq$ is called a **partial order** over M if for all $a, b, c \in M$ it satisfies the conditions

1. $a \leq a$ (**reflexivity**),
2. $a \leq b \wedge b \leq a \Rightarrow a = b$ (**antisymmetry**) and
3. $a \leq b \wedge b \leq c \Rightarrow a \leq c$ (**transitivity**).

In this case $(M, \leq)$ is called a **poset**. If two elements $a \leq b \in M$ are immediate neighbors, i.e. there is no third element $c \in M$ with $a < c < b$ we say that b **covers** a.

Definition 1.2. A poset is called **graded poset** if there is a map $\rho : M \to \mathbb{N}$ such that for all $a, b \in M$ with b covers a we have $\rho(b) = \rho(a) + 1$. In this case ρ is called the **rank function** of the graded poset.

Definition 1.3. A poset is called **directed poset** if for any two elements $a, b \in M$ there is an element $c \in M$ with $a \leq c$ and $b \leq c$. It is called **bounded poset** if it has an unique minimal and maximal element, denoted by $\hat{0}$ and $\hat{1}$.

Definition 1.4. Let $(M, \leq)$ be a poset and $a, b \in M$. Then we call $\{c \in M : a \leq c \leq b\}$ an **interval** and denote it by $[a, b]_{\leq}$. The set $\{c \in M : a < c < b\}$ is called an **open interval** and is denoted by $(a, b)_{\leq}$. In both cases we can omit the $\leq$-index if the relation is clear from context.

Definition 1.5. The **Hasse diagram** of the poset $(M, \leq)$ is the graph obtained in the following way: add a vertex for each element in M. Then add a directed edge from vertex a to b whenever b covers a.

Example 1.6. Suppose we have an arbitrary set M. Then the powerset $\mathcal{P}(M)$ can be partially ordered by the subset relation, so $(\mathcal{P}(M), \subseteq)$ is a poset. Indeed, this poset is always graded with the cardinality function as rank function. In Figure 1.1 we see the Hasse diagram of this poset with $M = \{x, y, z\}$.

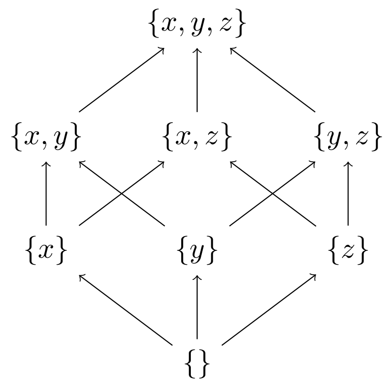

Figure 1.1.: Hasse diagram of the set of all subsets of $\{x, y, z\}$ ordered by the subset relation

Definition 1.7. Let $(M_i, \leq_i), i = 1, \ldots, n$ be a finite set of posets. We call the poset $(M_1 \times \ldots \times M_n, \leq)$ with $(a_1, \ldots, a_n) \leq (b_1, \ldots, b_n) \iff a_i \leq_i b_i$ for $i = 1, \ldots, n$ a **direct product of posets** and denote it by $(M_1, \leq_1) \times \ldots \times (M_n, \leq_n)$.

1.2. Coxeter groups

A Coxeter group, named after Harold Scott MacDonald Coxeter, is an abstract group generated by involutions with specific relations between these generators. A simple class of Coxeter groups are the symmetry groups of regular polyhedras in the Euclidean space.

The symmetry group of the square for example can be generated by two reflections s, t, whose stabilized hyperplanes enclose an angle of $\pi/4$. In this case the map st is a rotation in the plane by $\pi/2$. So we have $s^2 = t^2 = (st)^4 = \mathrm{id}$. In fact, this reflection group is determined up to isomorphy by s, t and these three relations [Hum92, Theorem 1.9]. Furthermore it turns out that the finite reflection groups in the Euclidean space are precisely the finite Coxeter groups [Hum92, Theorem 6.4].

In this chapter we compile some basic well-known facts on Coxeter groups, based on [Hum92].

Definition 1.8. Let $S = \{s_1, \ldots, s_n\}$ be a finite set of symbols and

$$R = \{m_{ij} \in \mathbb{N} \cup \{\infty\} : 1 \leq i, j \leq n\}$$

a set of numbers (or ∞) with $m_{ii} = 1$, $m_{ij} \geq 1$ and $m_{ij} = m_{ji}$. Then the free represented group

$$W = \langle S \mid (s_i s_j)^{m_{ij}} = e, 1 \leq i, j \leq n \rangle$$

is called a **Coxeter group** and (W, S) the corresponding **Coxeter system**. The cardinality of S is called the **rank** of the Coxeter system (and the Coxeter group).

From the definition we see that Coxeter groups only depend on the cardinality of S and the relations between the generators in S. A common way to visualize this information are Coxeter graphs.

Definition 1.9. Let (W, S) be a Coxeter system. Create a graph by adding a vertex for each generator in S, say the vertex v_i corresponds to the generator s_i. Let $(s_i s_j)^m = 1$. In case $m = 2$ then v_i, v_j have no connecting edge. In case $m = 3$ they are connected by an unlabled edge. For $m > 3$ they have a connecting edge with label m. We call this graph the **Coxeter graph** of our Coxeter system (W, S).

Definition 1.10. Let (W, S) be a Coxeter system. For an arbitrary element $w \in W$ we call a product $s_{i_1} \cdots s_{i_n} = w$ of generators $s_{i_1} \dots s_{i_n} \in S$ an **expression** of w. Any expression that can be obtained from $s_{i_1} \cdots s_{i_n}$ by omitting some (or all) generators is called a **subexpression** of w.

The present relations between the generators of a Coxeter group allow us to rewrite expressions. Hence, an element $w \in W$ can have more than one expression. Obviously any element $w \in W$ has infinitly many expressions, since any expression $s_{i_1} \cdots s_{i_n} = w$ can be extended by applying $s_1^2 = 1$ from the right. But there must be a smallest number of generators needed to receive w. For example the neutral element e can be expressed by the empty expression. Or each generator $s_i \in S$ can be expressed by itself, but any expression with less factors (i.e. the empty expression) is unequal to s_i.

Definition 1.11. Let (W, S) be a Coxeter system and $w \in W$ an element. Then there are some (not necessarily distinct) generators $s_i \in S$ with $s_1 \cdots s_r = w$. We call r the **expression length**. The smallest number $r \in \mathbb{N}_0$ for that w has an expression of length r is called the **length** of w and each expression of w that is of minimal length is called **reduced expression**. The map

$$l : W \to \mathbb{N}_0$$

that maps each element in W to its length is called **length function**.

Definition 1.12. Let (W, S) be a Coxeter system. We define

$$D_R(w) := \{s \in S : l(ws) < l(w)\}$$

as the **right descending set** of w. The analogue left version

$$D_L(w) := \{s \in S : l(sw) < l(w)\}$$

is called **left descending set** of w. Since the left descending set is not need in this paper we will often call the right descending just **descending set** of w.

The next lemma yields some useful identities and relations for the length function.

Lemma 1.13. [Hum92, Section 5.2] *Let (W, S) be a Coxeter system, $s \in S$, $u, w \in W$ and $l : W \to \mathbb{N}$ the length function. Then*

1. $l(w) = l(w^{-1})$,
2. $l(w) = 0 \iff w = e$,

3. $l(w) = 1 \iff w \in S$,

4. $l(uw) \leq l(u) + l(w)$,

5. $l(uw) \geq l(u) - l(w)$ *and*

6. $l(ws) = l(w) \pm 1$.

Remark 1.14. Note that $l(ws) = l(w) \pm 1$ has a left analogue by $l(sw) = l(w^{-1}s) = l(w^{-1}) \pm 1 = l(w) \pm 1$.

1.3. Exchange and Deletion Condition

We now obtain a way to get a reduced expression of an arbitrary element $s_1 \cdots s_r = w \in W$.

Definition 1.15. Let (W, S) be a Coxeter system. Any element $w \in W$ that is conjugated to a generator $s \in S$ is called **reflection**. Hence, the set of all reflections in W is

$$T = \bigcup_{w \in W} wSw^{-1}.$$

Theorem 1.16 (Strong Exchange Condition)**.** [Hum92, Theorem 5.8] *Let (W, S) be a Coxeter system, $w \in W$ an arbitrary element and $s_1 \cdots s_r = w$ with $s_i \in S$ a not necessarily reduced expression for w. For each reflection $t \in T$ with $l(wt) < l(w)$ there exists an index i for which $wt = s_1 \cdots \hat{s}_i \cdots s_r$, where $\hat{s}_i$ means omission. In case we start from a reduced expression, then i is unique.*

The Strong Exchange Condition can be weaken, when insisting on $t \in S$ to receive the following corollary.

Corollary 1.17 (Exchange Condition)**.** [Hum92, Theorem 5.8] *Let (W, S) be a Coxeter system, $w \in W$ an arbitrary element and $s_1 \cdots s_r = w$ with $s_i \in S$ a not necessarily reduced expression for w. For each generator $s \in S$ with $l(ws) < l(w)$ there exists an index i for which $ws = s_1 \cdots \hat{s}_i \cdots s_r$, where $\hat{s}_i$ means omission. In case we start from a reduced expression, then i is unique.*

Proof. Directly from Strong Exchange Condition. □

Remark 1.18. Note that both, Strong Exchange Condition and Exchange Condition, have an analogue left-sided version

$$l(tw) < l(w) \Rightarrow tw = ts_1 \cdots s_k = s_1 \cdots \hat{s}_i \cdots s_k$$

for all reflections $t \in T$, hence for all generators $s \in S$ in particular.

Corollary 1.19 (Deletion Condition)**.** [Hum92, Corollary 5.8] *Let (W, S) be a Coxeter system, $w \in W$ and $w = s_1 \cdots s_r$ with $s_i \in S$ an unreduced expression of w. Then there exist two indices $i, j \in \{1, \cdots, r\}$ with $i < j$, such that $w = s_1 \cdots \hat{s}_i \cdots \hat{s}_j \cdots s_r$, where $\hat{s}_i$ and $\hat{s}_j$ mean omission.*

Proof. Since the expression is unreduced there must be an index j for that the twisted length shrinks. That means for $w' = s_1 \cdots s_{j-1}$ is $l(w's_j) < l(w')$. Using the Exchange Condition we get $w's_j = s_1 \cdots \hat{s}_i \cdots s_{j-1}$ yielding $w = s_1 \cdots \hat{s}_i \cdots \hat{s}_j \cdots s_r$. □

This corollary allows us to reduce expressions, i.e. to find a subexpression that is reduced. Due to the Deletion Condition any unreduced expression can be reduced by omitting an even number of generators (we just have to apply the Deletion Condition inductively).

The Strong Exchange Condition, the Exchange Condition and the Deletion Condition, are some of the most powerful tools when investigating properties of Coxeter groups. The second can be used to prove a very handy property of Coxeter groups. The intersection of two parabolic subgroups is again a parabolic subgroup.

Definition 1.20. Let (W, S) be a Coxeter system. For a subset of generators $I \subset S$ we call the subgroup $W_I \leq W$ that is generated by the elements in I with the corresponding relations, a **parabolic subgroup** of W.

Lemma 1.21. [Hum92, Section 5.8] *Let (W, S) be a Coxeter system and $I, J \subset S$ two subsets of generators. Then $W_I \cap W_J = W_{I \cap J}$.*

A related fact, is the following lemma.

Lemma 1.22. [Hum92, Section 5.8] *Let (W, S) be a Coxeter system and $w \in W$. Let $w = s_1 \cdots s_k$ be any reduced expression for w. Then the set of used generators $\{s_1, \ldots, s_k\} \subset S$ is independent of the particular chosen reduced expression. It only depends on w itself.*

This means that two reduced expressions for an element $w \in W$ use exactly the same generators.

1.4. Finite Coxeter groups

Coxeter groups can be finite and infinite. A simple example for the former category is the following. Let $S = \{s\}$. Due to definition it must be $s^2 = e$. So W is isomorph to $\mathbb{Z}_2$ and finite. An example for an infinite Coxeter group can be obtained from $S = \{s, t\}$ with $s^2 = t^2 = e$ and $(st)^\infty = e$ (so we have no relation between s and t). Obviously the element st has infinite order forcing W to be infinite. But there are infinite Coxeter groups without a ∞-relation, as well. An example for this is W obtained from $S = \{s_1, s_2, s_3\}$ with $s_1^2 = s_2^2 = s_3^2 = (s_1s_2)^3 = (s_2s_3)^3 = (s_3s_1)^3 = e$. But how can one decide whether W is finite or not?

To provide a general answer to this question we fallback on a certain class of Coxeter groups, the irreducible ones.

Definition 1.23. A Coxeter system is called **irreducible** if the corresponding Coxeter graph is connected. Else, it is called **reducible**.

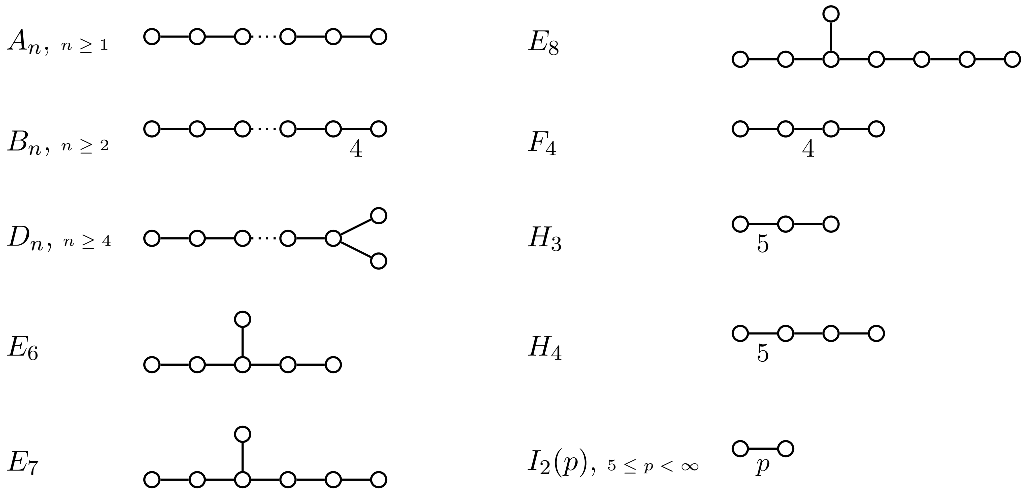

Figure 1.2.: All types of irreducible finite Coxeter systems

If a Coxeter system is reducible, then its graph has more than one component and each component corresponds to a parabolic subgroup of W.

Proposition 1.24. [Hum92, Proposition 6.1] *Let (W, S) be a reducible Coxeter system. Then there exists a partition of S into I, J with $(s_i s_j)^2 = e$ whenever $s_i \in I, s_j \in J$ and W is isomorph to the direct product of the two parabolic subgroups W_I and W_J.*

This proposition tells us that an arbitrary Coxeter system is finite if and only if its irreducible parabolic subgroups are finite. Therefore we can indeed fallback to irreducible Coxeter systems without loss of generality. If we could categorize all irreducible finite Coxeter systems, then we could categorize all finite Coxeter systems. This is done by the following theorem:

Theorem 1.25. [Hum92, Theorem 6.4] *The irreducible finite Coxeter systems are exactly the ones in Figure 1.2.*

This allows us to decide with ease whether a given Coxeter system is finite. Take its irreducible parabolic subgroups and check if each is of type A_n, B_n, D_n, E_6, E_7, E_8, F_4, H_3, H_4 or $I_2(p)$.

1.5. Affine and compact hyperbolic Coxeter groups

Two other important classes of Coxeter systems are the so-called affine and compact hyperbolic ones.

Definition 1.26. An irreducible Coxeter systems (W, S) is called **affine** if it is one of those from Figure 1.3. The affine Coxeter systems arise from the finite ones by appending another generator in a certain manner.

Proposition 1.27. *Let (W, S) be an affine Coxeter system. Then every proper parabolic subgroup is finite.*

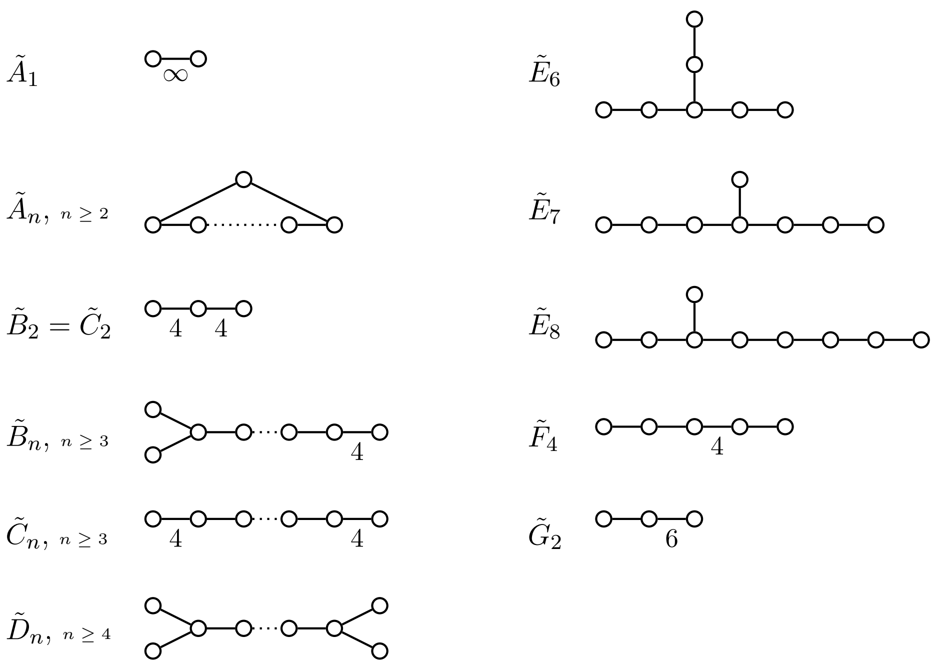

Figure 1.3.: All types of affine Coxeter systems

Proof. Immediate by simple inspection. □

Definition 1.28. An irreducible Coxeter systems (W, S) is called **compact hyperbolic** if it is infinite, not affine and if for each $S' \subsetneq S$ the parabolic subgroup $W_{S'}$ is finite. In this paper the compact hyperbolic Coxeter systems are denoted with X, but note that this is not common in literature.

For a more detailed introduction of affine and compact hyperbolic Coxeter systems refer to [Hum92, Section 2.5, 6.7 – 6.9]. By the way we defined compact hyperbolic Coxeter systems it is clear that the affine and compact hyperbolic Coxeter systems are precisely the ones that are infinite, irreducible and whose proper parabolic subgroups are finite. For more details on the type of their parabolic subgroups see Table 3.2.

Theorem 1.29. [Che69, Section 5] *The compact hyperbolic Coxeter systems are precisely those from Figure 1.4.*

1.6. Bruhat ordering

We now investigate ways to partially order the elements of a Coxeter group. Furthermore, this ordering should be compatible with the length function, i.e. for $w, v \in W$ we have $l(w) < l(v)$ whenever $w < v$.

Definition 1.30. Let (W, S) be a Coxeter system and $T = \cup_{w \in W} wSw^{-1}$ the set of all reflections in W. We write $w' \to w$ if there is a $t \in T$ with $w't = w$ and $l(w') < l(w)$. If there is a sequence $w' = w_0 \to w_1 \to \ldots \to w_m = w$, we say $w' < w$. The resulting relation $w' \leq w$ is called **Bruhat ordering**, denoted by $\mathrm{Br}(W)$.

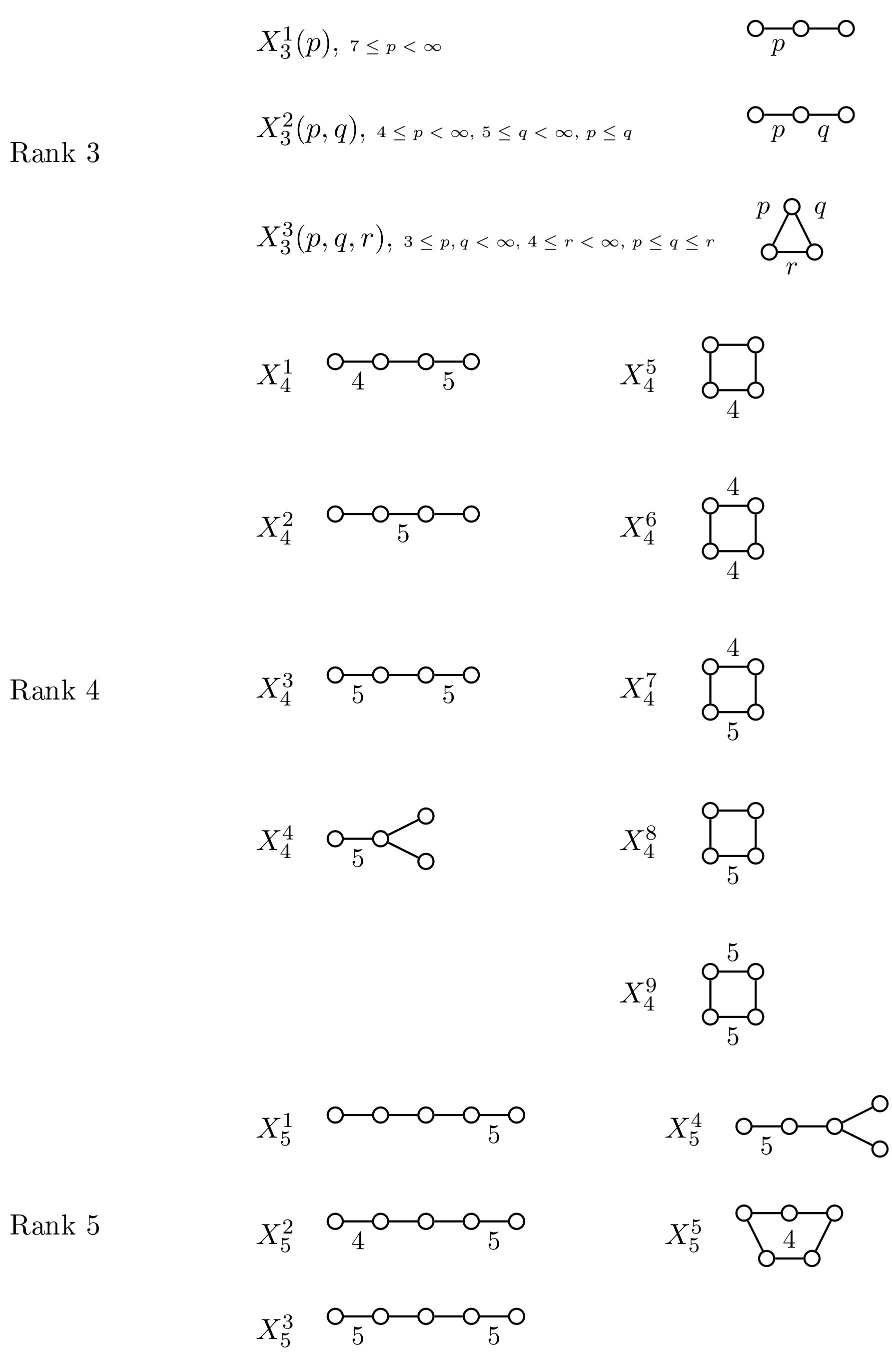

Figure 1.4.: All types of compact hyperbolic Coxeter systems

Lemma 1.31. [Hum92, Section 5.9] *Let (W, S) be a Coxeter system. Then* $\mathrm{Br}(W)$ *is a poset.*

Proof. The Bruhat ordering is reflexive by definition. Since the elements in sequences $e \to w_1 \to w_2 \to \ldots$ are strictly ascending in length, it must be antisymmetric. By concatenation of sequences we get the transitivity. □

We now show that $Br(W)$ is graded. By definition we already have $v < w$ if and only if $l(v) < l(w)$, but its not that obvious that two immediately adjacent elements differ in length by exactly 1. Beforehand let us just mention two other partial orderings that are graded by definition.

Definition 1.32. Let (W, S) be a Coxeter system. The ordering $\leq_R$ defined by $u \leq_R w$ if and only if $uv = w$ for some $u \in W$ with $l(u) + l(v) = l(w)$ is called the **right weak ordering**. The left-sided version $u \leq_L w$ if and only if $vu = w$ is called the **left weak ordering**.

To ensure the Bruhat ordering is graded as well, we need another characterization of the Bruhat ordering in terms of subexpressions.

Proposition 1.33. [Hum92, Proposition 5.9] *Let (W, S) be a Coxeter system, $u, w \in W$ with $u \leq w$ and $s \in S$. Then $us \leq w$ or $us \leq ws$ or both.*

Theorem 1.34 (Subword property)**.** [Hum92, Theorem 5.10] *Let (W, S) be a Coxeter system and $w \in W$ with a fixed, but arbitrary, reduced expression $w = s_1 \cdots s_r$, $s_i \in S$. Then $u \leq w$ (in the Bruhat ordering) if and only if u can be obtained as a subexpression of this reduced expression.*

Corollary 1.35. *Let $u, w \in W$. Then the interval $[u, w]$ in the Bruhat order* $\mathrm{Br}(W)$ *is finite.*

Proof. We have $[u, w] \subseteq [e, w]$. All elements $v \in [e, w]$ can be obtained as subexpressions of one fixed reduced expression for w. Let $s_1 \ldots s_k = w$ be such a reduced expression. Then there are at most 2^k many subexpressions, hence $[u, w]$ is finite. □

This characterization of the Bruhat ordering is very handy. With it and the following short lemma we will be in the position to show that $\mathrm{Br}(W)$ is graded with rank function l.

Lemma 1.36. [Hum92, Lemma 5.11] *Let (W, S) be a Coxeter system, $u, w \in W$ with $u < w$ and $l(w) = l(u) + 1$. In case there is a generator $s \in S$ with $u < us$ but $us \neq w$, then both $w < ws$ and $us < ws$.*

Proposition 1.37. [Hum92, Proposition 5.11] *Let (W, S) be a Coxeter system and $u < w$. Then there are elements $w_0, \ldots, w_m \in W$ such that $u = w_0 < w_1 < \ldots < w_m = w$ with $l(w_i) = l(w_{i-1}) + 1$ for $1 \leq i \leq m$.*

Corollary 1.38. *Let (W, S) be a Coxeter system and* $\mathrm{Br}(W)$ *the Bruhat ordering poset of W. Then* $\mathrm{Br}(W)$ *is graded with $l : W \to \mathbb{N}$ as rank function.*

Proof. Let $u, w \in W$ with w covering u. Then Proposition 1.37 says there is a sequence $u = w_0 < \ldots < w_m = w$ with $l(w_i) = l(w_{i-1}) + 1$ for $1 \leq i \leq m$. Since w covers u it must be $m = 1$ and so $u < w$ with $l(w) = l(u) + 1$. □

Theorem 1.39 (Lifting Property). [Deo77, Theorem 1.1] *Let (W, S) be a Coxeter system and $v, w \in W$ with $v \leq w$. Suppose $s \in S$ with $s \in D_R(w)$. Then*

1. $vs \leq w$,
2. $s \in D_R(v) \Rightarrow vs \leq ws$.

Remark 1.40. Note that the Lifting Property has an analogue left-sided version: Let (W, S) be a Coxeter system and $v, w \in W$ with $v \leq w$. Suppose $s \in S$ with $s \in D_L(w)$. Then

1. $sv \leq w$,
2. $s \in D_L(v) \Rightarrow sv \leq sw$.

The Lifting Property seems quite innocent, but when trying to investigate facts around the Bruhat ordering it proves to be one of the key tools in many cases.

Proposition 1.41. [Den09, Proposition 7] *The poset* $\mathrm{Br}(W)$ *is directed.*

Proposition 1.42. [Den09, Proposition 8]

1. *Let W be finite, then there exists an unique element $w_0 \in W$ with $w \leq w_0$ for all $w \in W$.*
2. *If W contains an element w, with $D_R(w) = S$, then W is finite and w is the unique element w_0.*

Corollary 1.43. *Let (W, S) be a finite Coxeter system. Then* $\mathrm{Br}(W)$ *is graded, directed and bounded.*

Proof. $\mathrm{Br}(W)$ is graded due to Corollary 1.38, directed due to Proposition 1.41 and bounded due to Proposition 1.42. □

Corollary 1.44. *Let (W, S) be a Coxeter system and $w, v \in W$ with $w < v$. Then the interval $[w, v]$ is a finite, graded, directed and bounded poset.*

Proof. The poset structure and the graduation transfers directly from $Br(W)$. By Corollary 1.35 intervals in $\mathrm{Br}(W)$ are finite. Since v is the unique maximal element and w the unique minimal element, it is bounded. By definition of intervals we have $u \leq v$ for every element $u \in [w, v]$, hence it is directed. □

2. Twisted involutions in Coxeter groups

In this section we focus on a certain subset of elements in Coxeter groups, the so called twisted involutions. From now on (and in the next sections) we fix some symbols to have always the same meaning (some definitions used here will follow later):

(W, S)	A Coxeter system with generators S and Coxeter group W.
s	A generator in S.
u, v, w	An element in the Coxeter group W.
θ	A Coxeter system automorphism of (W, S) with $\theta^2 = \mathrm{id}$.
$\mathcal{I}_\theta$	The set of θ-twisted involutions of W.
$\underline{S}$	A set of symbols, $\underline{S} = \{\underline{s} : s \in S\}$.

2.1. Introduction to twisted involutions

Twisted involutions generalize the property of being involutive with respect to an involutory automorphism θ. For $\theta = \mathrm{id}$ the set of θ-twisted involutions, denoted by $\mathcal{I}_\theta$ coincides with the set of ordinary involutions in W (cf. Example 2.3). As we will see the set of this θ-twisted involutions equals the e-orbit of a special action, defined in Definition 2.5. For $\mathcal{I}_\theta$ and the mentioned map many properties of ordinary Coxeter groups hold. In particular there is an analogue to the Exchange Condition and Deletion Condition.

Definition 2.1. An automorphism $\theta : W \to W$ with $\theta(S) = S$ is called a **Coxeter system automorphism** of (W, S). We always assume $\theta^2 = \mathrm{id}$.

Definition 2.2. We define the **set of θ-twisted involutions** of W as

$$\mathcal{I}_\theta(W) := \{w \in W : \theta(w) = w^{-1}\}.$$

If θ is clear from the context we just say **set of twisted involutions**. Every element $w \in \mathcal{I}_\theta(W)$ is called a **θ-twisted involution**, resp. **twisted involution**. Often, when W is clear from the context, we will omit it and just write $\mathcal{I}_\theta$.

Example 2.3. Let $\theta = \mathrm{id}_W$. Then θ is a Coxeter system automorphism and the set of all id-twisted involutions coincides with the set of all ordinary involutions of W.

The next example is more helpful, since it reveals a way to think of $\mathcal{I}_\theta$ as a generalization of ordinary Coxeter groups.

Example 2.4. [Hul07, Example 3.2] Let θ be an automorphism of $W \times W$ with $\theta : (u, w) \mapsto (w, u)$. Then θ is a Coxeter system automorphism of the Coxeter system $(W \times W, S \times S)$ and the set of twisted involutions is

$$\mathcal{I}_\theta = \{(w, w^{-1}) \in W \times W : w \in W\}.$$

This yields a canonical bijection between $\mathcal{I}_\theta$ and W.

Definition 2.5. Let $\underline{S} := \{\underline{s} : s \in S\}$ be a set of symbols. Each element in $\underline{S}$ acts from the right on W by the following definition:

$$w\underline{s} = \begin{cases} ws & \text{if } \theta(s)ws = w, \\ \theta(s)ws & \text{else.} \end{cases}$$

This action can be extended on the whole free monoid over $\underline{S}$ by

$$w\underline{s}_1\underline{s}_2 \ldots \underline{s}_k = (\ldots((w\underline{s}_1)\underline{s}_2)\ldots)\underline{s}_k.$$

If $w\underline{s} = \theta(s)ws$, then we say $\underline{s}$ **acts by twisted conjugation** on w. Else we say $\underline{s}$ **acts by multiplication** on w.

Remark 2.6. Let $v = s_1 \cdots s_k \in W$. As abuse of notation we will sometimes write $w\underline{v}$ and define it as $w\underline{s}_1 \ldots \underline{s}_k$.

Note that this is no group action. For example, let W be a Coxeter group with two generators s, t satisfying $\mathrm{ord}(st) = 3$ and let $\theta = \mathrm{id}$. Then $sts = tst$, but

$$e\underline{sts} = st\underline{s} = tst\underline{s} = sts\underline{s} = t \neq s = tst\underline{t} = sts\underline{t} = t\underline{st} = e\underline{tst}.$$

Definition 2.7. Let $k \in \mathbb{N}$ and $s_i \in S$ for all $1 \leq i \leq k$. Then an expression $e\underline{s}_1 \ldots \underline{s}_k$, or just $\underline{s}_1 \ldots \underline{s}_k$, is called **$\theta$-twisted expression**. If θ is clear from the context, we omit θ and call it **twisted expression**. A twisted expression is called **reduced twisted expression** if there is no $k' < k$ with $\underline{s}'_1 \ldots \underline{s}'_{k'} = \underline{s}_1 \ldots \underline{s}_k$.

Lemma 2.8. [Hul07, Lemma 3.4] *Let $w \in \mathcal{I}_\theta$ and $s \in S$. Then*

$$w\underline{s} = \begin{cases} ws & \text{if } l(\theta(s)ws) = l(w), \\ \theta(s)ws & \text{else.} \end{cases}$$

Proof. [Hul07, Lemma 3.4] We reproduce the proof from loc. cit. here., since it is interesting from a technical point of view. Suppose $\underline{s}$ acts by multiplication on w. Then $\theta(s)ws = w$ and so $l(\theta(s)ws) = l(w)$. Conversely, suppose $l(\theta(s)ws) = l(w)$. If $w\underline{s} = ws$, then we are done. So assume $\theta(s)ws \neq w$. Then w must have a reduced expression beginning with $\theta(s)$ or ending with s (else, we could not have $l(\theta(s)ws) = l(w)$). Without loss of generality suppose that $\theta(s)s_1 \cdots s_k$ is such an expression for w. Since w is a θ-twisted involution, i.e. $\theta(w) = w^{-1}$, we have $l(ws) < l(w)$. Since $l(\theta(s)ws) = l(w)$, no reduced expression for w both begins with $\theta(s)$ and ends with s and hence Exchange Condition yields $ws = s_1 \cdots s_k$, which implies $\theta(s)ws = w$, contradicting to our assumption. $\square$

Lemma 2.9. *We have $l(ws) < l(w)$ if and only if $l(w\underline{s}) < l(w)$.*

Proof. Suppose $\underline{s}$ acts by multiplication on w. Then $w\underline{s} = ws$ and there is nothing to prove. So suppose $\underline{s}$ acts by twisted conjugation on w. If $l(ws) < l(w)$, then Lemma 1.13 yields $l(ws) + 1 = l(w)$. Assuming $l(w\underline{s}) = l(\theta(s)ws) = l(w)$ would imply that $\underline{s}$ acts by multiplication on w due to Lemma 2.8, which is a contradiction. So $l(w\underline{s}) = l(\theta(s)ws) < l(w)$. Conversely, suppose $l(w\underline{s}) < l(w)$. Then Lemma 1.13 says $l(w\underline{s}) = l(\theta(s)ws) = l(w) - 2$ and so $l(ws) = l(w) - 1$. □

Lemma 2.10. *For all $w \in W$ and $s \in S$ we have $w\underline{s}\underline{s} = w$.*

Proof. For $w\underline{s}$ there are two cases. Suppose $\underline{s}$ acts by multiplication on w, i.e. $\theta(s)ws = w$. For $ws\underline{s}$ there are again two possible options:

$$ws\underline{s} = \begin{cases} wss = w & \text{if } \theta(s)wss = ws, \\ \theta(s)wss = ws & \text{else.} \end{cases}$$

The second option contradicts itself.

Now suppose $\underline{s}$ acts by twisted conjugation on w. This means $\theta(s)ws \neq w$ and for $(\theta(s)ws)\underline{s}$ there are again two possible options:

$$(\theta(s)ws)\underline{s} = \begin{cases} \theta(s)wss = \theta(s)w & \text{if } \theta(s)\theta(s)wss = \theta(s)ws, \\ \theta(s)\theta(s)wss = w & \text{else.} \end{cases}$$

The first option is impossible since $\theta(s)\theta(s)wss = w$ and we have assumed $\theta(s)ws \neq w$. Hence, the only possible cases yield $w\underline{s}\underline{s} = w$. □

Remark 2.11. Lemma 2.10 allows us to rewrite equations of twisted expressions. For example

$$u = w\underline{s} \iff u\underline{s} = w\underline{s}\underline{s} = w.$$

This can be iterated to get

$$u = w\underline{s}_1 \dots \underline{s}_k \iff u\underline{s}_k \dots \underline{s}_1 = w.$$

Lemma 2.12. *For all θ, $w \in W$ and $s \in S$ it holds that $w \in \mathcal{I}_\theta$ iff $w\underline{s} \in \mathcal{I}_\theta$.*

Proof. Let $w \in \mathcal{I}_\theta$. For $w\underline{s}$ there are two cases. Suppose $\underline{s}$ acts by multiplication on w. Then we get

$$\theta(ws) = \theta(\theta(s)wss) = \theta^2(s)\theta(w) = sw^{-1} = (ws^{-1})^{-1} = (ws)^{-1}.$$

Suppose $\underline{s}$ acts by twisted conjugation on w. Then we get

$$\theta(\theta(s)ws) = \theta^2(s)\theta(w)\theta(s) = sw^{-1}\theta(s) = (\theta^{-1}(s)ws^{-1})^{-1} = (\theta(s)ws)^{-1}.$$

In both cases $w\underline{s} \in \mathcal{I}_\theta$.

Now let $w\underline{s} \in \mathcal{I}_\theta$. Suppose $\underline{s}$ acts by multiplication on w. Then

$$\theta(w) = \theta(\theta(s)ws) = \theta^2(s)\theta(ws) = s(ws)^{-1} = ss^{-1}w^{-1} = w^{-1}.$$

Suppose $\underline{s}$ acts by twisted conjugation on w. Then

$$\begin{aligned}\theta(w) &= \theta(\theta(s)\theta(s)wss) = \theta^2(s)\theta(\theta(s)ws)\theta(s)\\ &= s(\theta(s)ws)^{-1}\theta(s) = s(s^{-1}w^{-1}\theta(s^{-1})\theta(s) = w^{-1}.\end{aligned}$$

In both cases $w \in \mathcal{I}_\theta$. □

A remarkable property of the action from Definition 2.5 is its e-orbit. As the following lemma shows, it coincides with $\mathcal{I}_\theta$.

Lemma 2.13. [Hul07, Proposition 3.5] *The set of θ-twisted involutions coincides with the set of all θ-twisted expressions.*

Proof. By Lemma 2.12, each twisted expression is in $\mathcal{I}_\theta$, since $e \in \mathcal{I}_\theta$. So let $w \in \mathcal{I}_\theta$. If $l(w) = 0$, then $w = e \in \mathcal{I}_\theta$. So assume $l(w) = r > 0$ and that we have already proven that every twisted involution $w' \in \mathcal{I}_\theta$ with $\rho(w') < r$ has a twisted expression. If w has a reduced twisted expression ending with $\underline{s}$, then w also has a reduced expression (in S) ending with s and so $l(ws) < l(w)$. With Lemma 2.9 we get $l(w\underline{s}) < l(w)$. By induction $w\underline{s}$ has a twisted expression and hence $w = (w\underline{s})\underline{s}$ has one, too. □

In the same way, we can use regular expressions to define the length of an element $w \in W$, we can use the twisted expressions to define the twisted length of an element $w \in \mathcal{I}_\theta$.

Definition 2.14. Let $\mathcal{I}_\theta$ be the set of twisted involutions. Then we define $\rho(w)$ as the smallest $k \in \mathbb{N}$ for that a twisted expression $w = \underline{s}_1 \dots \underline{s}_k$ exists. This is called the **twisted length** of w.

Lemma 2.15. [Hul05, Theorem 4.8] *The Bruhat ordering, restricted to the set of twisted involutions $\mathcal{I}_\theta$, is a graded poset with ρ as rank function. We denote this poset by* $\mathrm{Br}(\mathcal{I}_\theta)$.

We now establish many properties from ordinary Coxeter groups for twisted expressions and $\mathrm{Br}(\mathcal{I}_\theta)$. As seen in Example 2.4 there is a Coxeter system (W', S') and a Coxeter system automorphism θ with $\mathrm{Br}(W) \cong \mathrm{Br}(\mathcal{I}_\theta(W'))$. So the hope that many properties can be transfered is eligible.

Lemma 2.16. [Hul07, Lemma 3.8] *Let $w \in \mathcal{I}_\theta$ and $s \in S$. Then $\rho(w\underline{s}) = \rho(w) \pm 1$. In fact it is $\rho(w\underline{s}) = \rho(w) - 1$ if and only if $s \in D_R(w)$.*

Lemma 2.17 (Lifting Property for twisted expressions)**.** [Hul07, Lemma 3.9] *Let $v, w \in W$ with $v \leq w$. Suppose $s \in S$ with $s \in D_R(w)$. Then*

1. $v\underline{s} \leq w$,
2. $s \in D_R(v) \Rightarrow v\underline{s} \leq w\underline{s}$.

Proof. Whenever a relation comes from the ordinary Lifting Property, we denote it by $<_{LP}$ in this proof.

$v\underline{s} = vs \wedge w\underline{s} = ws$: Same situation as in Lifting Property.

$v\underline{s} = vs \wedge w\underline{s} = \theta(s)ws$: The first part $v\underline{s} = vs \leq_{LP} w$ is immediate. Suppose $s \in D_R(v)$. Then $vs \leq_{LP} ws \Rightarrow v = \theta(s)vs \leq ws \Rightarrow v\underline{s} = vs \leq \theta(s)ws = w\underline{s}$.

$v\underline{s} = \theta(s)vs \wedge w\underline{s} = ws$: We have $\theta(s)w = ws$ and therefore $\theta(s) \in D_L(w)$. Suppose $s \in D_R(v)$. Then $\theta(s) \in D_R(vs)$ and hence $v\underline{s} = \theta(s)vs \leq vs \leq_{LP} ws = w\underline{s} \leq w$. In return, suppose $s \notin D_R(v)$. Since $vs \leq_{LP} w$ and $\theta(s) \in D_L(w)$ we can apply the left analogue of Lifting Property on $vs, w, \theta(s)$ to get $v\underline{s} = \theta(s)vs \leq_{LP} w$.

$v\underline{s} = \theta(s)vs \wedge w\underline{s} = \theta(s)ws$: Let $s \in D_R(w)$. Then $vs \leq_{LP} ws$. Since $\theta(s) \in D_L(vs)$ and $\theta(s) \in D_L(ws)$ we can apply the left-sided Lifting Property to get $v\underline{s} = \theta(s)vs \leq_{LP} \theta(s)ws = w\underline{s} \leq w$. In return, let $s \notin D_R(w)$. Since $l(\theta(s)ws) = l(w) - 2$ we have $\theta(s) \in D_L(w)$. So we can use the Lifting Property to get $vs \leq_{LP} w$ and then with the left-sided Lifting Property $v\underline{s} = \theta(s)vs \leq_{LP}$ w. □

Proposition 2.18 (Exchange property for twisted expressions)**.** [Hul07, Proposition 3.10] *Suppose $\underline{s}_1 \ldots \underline{s}_k$ is a reduced twisted expression. If $\rho(\underline{s}_1 \ldots \underline{s}_k\underline{s}) < k$ for some $s \in S$, then $\underline{s}_1 \ldots \underline{s}_k\underline{s} = \underline{s}_1 \ldots \hat{\underline{s}}_i \ldots \underline{s}_k$ for some $i \in \{1, \ldots, k\}$.*

Corollary 2.19 (Deletion property for twisted expressions)**.** [Hul07, Proposition 3.11] *Let $w = s_1 \ldots \underline{s}_k$ be a not reduced twisted expression. Then there are two indices $1 \leq i < j \leq k$ such that $w = \underline{s}_1 \ldots \hat{\underline{s}}_i \ldots \hat{\underline{s}}_j \ldots \underline{s}_k$.*

Proof. Choose j minimal, so $\underline{s}_1 \ldots \underline{s}_j$ is not reduced. By Exchange property for twisted expressions there is an index i with $\underline{s}_1 \ldots \underline{s}_j = s_1 \ldots \hat{\underline{s}}_i \ldots \underline{s}_{j-1}$ yielding our hypothesis $w = \underline{s}_1 \ldots \underline{s}_j \ldots \underline{s}_k = \underline{s}_1 \ldots \hat{\underline{s}}_i \ldots \hat{\underline{s}}_j \ldots \underline{s}_k$. □

When applying the Exchange property for twisted expressions to a twisted expression, there is no hint which $\underline{s}_i$ can be omitted. Consider the following situation: let $w \in \mathcal{I}_\theta$ and $w\underline{s}_1 \ldots \underline{s}_k = w\underline{t}_1 \ldots \underline{t}_k$ two reduced twisted expressions. Then in the twisted expression $w\underline{s}_1 \ldots \ldots \underline{s}_k\underline{t}_k$ we can omit the $\underline{t}_k$ and one other $\underline{s}$ by Exchange property for twisted expressions and get still the same element. It would be nice, when the second omitted $\underline{s}$ is one of the $\underline{s}_i$ in general, but unfortunately this proves to be false.

Example 2.20. Let $W = A_3$, $\theta = \mathrm{id}$ and $w = \underline{s}_3$. Then $w\underline{s}_2\underline{s}_1\underline{s}_2 = w\underline{s}_1\underline{s}_2\underline{s}_3$, but $w\underline{s}_1\underline{s}_2\underline{s}_3\underline{s}_2 \notin \{w\underline{s}_1\underline{s}_2, w\underline{s}_1\underline{s}_3, w\underline{s}_2\underline{s}_3\}$. Hence the omission cannot be chosen after the prefix w, but at least $w\underline{s}_1\underline{s}_2\underline{s}_3\underline{s}_2 = \underline{s}_1\underline{s}_2\underline{s}_3$ works, as guaranteed by Exchange property for twisted expressions.

2.2. Twisted weak ordering

In this section we introduce the twisted weak ordering $Wk(\theta)$ on the set $\mathcal{I}_\theta$ of θ-twisted involutions.

Definition 2.21. For $v, w \in \mathcal{I}_\theta$ we define $v \preceq w$ if and only if there are $\underline{s}_1, \ldots, \underline{s}_k \in \underline{S}$ with $w = v\underline{s}_1 \ldots \underline{s}_k$ and $\rho(v) = \rho(w) - k$. We call the poset $(\mathcal{I}_\theta, \preceq)$ **twisted weak ordering**, denoted by $Wk(W, \theta)$. When the Coxeter group W is clear from the context, we just write $Wk(\theta)$.

Proposition 2.22. *The poset $Wk(\theta)$ is a graded poset with rank function ρ.*

Proof. Follows immediately from the definition of $\preceq$. $\square$

By a diagram of a poset $Wk(\theta)$, we do not just mean the ordinary Hasse diagram. Suppose $w, v \in Wk(\theta)$ with $w\underline{s} = v$. We encode the information if $\underline{s}$ acts as twisted involution or as multiplication on w by drawing either a solid or a dashed edge from w to v. Another possible extension is to label (or color) the edges to encode the involved $\underline{s}$. For simplification of terminology we still just speak of the Hasse diagram of $Wk(\theta)$. The next example shows such a (extended) Hasse diagram.

Example 2.23. In Figure 2.1 we see the (extended) Hasse diagram of $Wk(A_4, \mathrm{id})$.

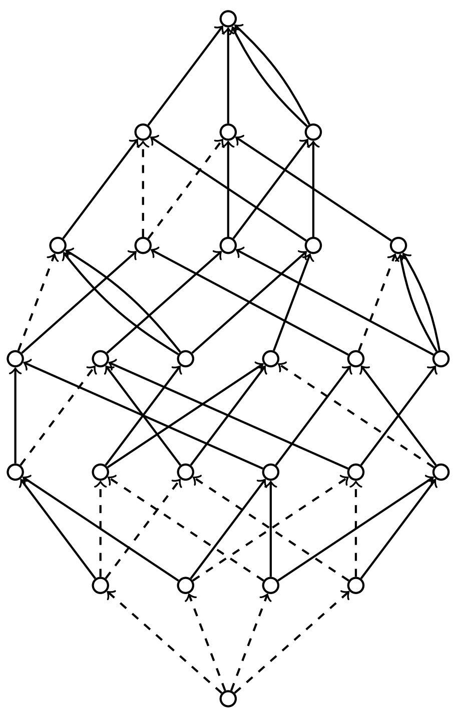

Figure 2.1.: Hasse diagram of $Wk(A_4, \mathrm{id})$

Lemma 2.24. *The poset $Wk(\theta)$ is a subposet of* $\mathrm{Br}(\mathcal{I}_\theta)$.

Proof. Both posets are defined on $\mathcal{I}_\theta$. Let $w, v \in \mathcal{I}_\theta$ be two twisted involutions. Assume $w \preceq v$ with $w\underline{s} = v$ for some $s \in S$. If $\underline{s}$ acts by multiplication on w, then $ws = v$ and since $s \in T$ (T the set of all reflections in W) and $l(w\underline{s}) = l(w) + 1$ we have $w \leq v$. If conversely $\underline{s}$ acts by twisted conjugation on w, then $v = \theta(s)ws = w(w^{-1}\theta(s)w)(e^{-1}se)$ and since $w^{-1}\theta(s)w, s \in T$ and $l(w\underline{s}) = l(\theta(s)w) + 1 = l(w) + 2$ we have again $w \leq v$. $\square$

Proposition 2.25. *For all $w \in \mathcal{I}_\theta$ and $s \in S$ we have $w\underline{s} \prec w$ if and only if $s \in D_R(w)$ and $w\underline{s} \succ w$ if and only if $s \notin D_R(w)$ as well as $w\underline{s} < w$ if and only if $s \in D_R(w)$ and $w\underline{s} > w$ if and only if $s \notin D_R(w)$.*

Proof. We have $w\underline{s}\underline{s} = w$ and $\rho(w\underline{s}) = \rho(w) - 1$ if and only if $s \in D_R(w)$ and $\rho(w\underline{s}) = \rho(w) + 1$ if and only if $s \notin D_R(w)$ by Lemma 2.16. By Lemma 2.24 both statements are true for $\mathrm{Br}(\mathcal{I}_\theta)$, too. ☐

Definition 2.26. Let $v, w \in W$ with $\rho(w) - \rho(v) = n$. A sequence $v = w_0 \prec w_1 \prec \ldots \prec w_n = w$ is called a **geodesic** from v to w.

Proposition 2.27. *Let $v, w \in W$ with $v \prec w$. Then all geodesics from v to w have the same count of twisted conjugated and multiplicative steps.*

Proof. Suppose we have two geodesics from v to w, where the first has n and the second m multiplicative steps. Then $l(w)+n+2(k-n) = l(v) = l(w)+m+2(k-m)$, hence $n = m$. ☐

Lemma 2.28. *Let $w \in W$ and $w\underline{s} \succ w$. Then $|\{t \in S : w\underline{t} = w\underline{s}\}| \in \{1, 2\}$.*

Proof. Suppose $t \in S \setminus D_R(w)$ with $w\underline{t} = w\underline{s}$. Because of the ordinary length either both $\underline{s}$ and $\underline{t}$ act by multiplication on w, or both act by twisted conjugation on w. Suppose they act by multiplication, then $ws = w\underline{s} = w\underline{t} = wt$, hence $s = t$. Conversely, assume they act by twisted conjugation. Then $\theta(s)ws = w\underline{s} = w\underline{t} = \theta(t)wt$. Because of $\theta(t)wtt = \theta(t)w = \theta(s)wst$ we have $l(\theta(s)wst) < l(\theta(s)ws)$ and so by Exchange Condition there are three possible cases

$$\theta(t)w = \theta(s)wst = \begin{cases} \theta(s)w & \Rightarrow s = t, \\ ws & \Rightarrow \theta(t) = wsw^{-1} \text{ or} \\ \theta(s)\overline{w}s & \Rightarrow w = \theta(t)\theta(s)\overline{w}s, \end{cases}$$

where $\overline{w}$ denotes a well chosen subexpression of w. The first case is trivial, the second determines t unambiguously. The third case is impossible, since by Exchange Condition and Remark 1.18 we would have a reduced expression for w beginning with $\theta(s)$ or ending with s (or both), yielding $l(\theta(s)ws) \leq l(w)$, which contradicts to $\rho(w\underline{s}) = \rho(\theta(s)ws) > \rho(w)$. Therefore, there cannot be more than two distinct $s, t \in S \setminus D_R(w)$ with $w\underline{s} = w\underline{t}$. ☐

Lemma 2.29. *Let $w \in \mathcal{I}_\theta$ and $s, t \in S$ be two distinct generators. If $w\underline{s} = w\underline{t}$, then* $\mathrm{ord}(st) = 2$.

Proof. By the proof of Lemma 2.28 we see that $w\underline{s} = w\underline{t}$ for two distinct $s, t \in S$ implies that $\theta(t)w = ws$ holds and that $\underline{s}$ and $\underline{t}$ act by twisted conjugation on w. Since $\theta(w) = w^{-1}$ we also have $\theta(s)w = wt$ by

$$\theta(t)w = ws \iff \theta(\theta(t)w) = \theta(ws) \iff tw^{-1} = w^{-1}\theta(s) \iff wt = \theta(s)w.$$

Hence, we have $wts = \theta(s)ws = \theta(t)wt = wst$, yielding $st = ts$ and $\mathrm{ord}(st) = 2$. ☐

2.3. Residues

Residues in $Wk(\theta)$ are subsets of θ-twisted involutions that can be "reached" from a fixed starting point by using just certain $\underline{s} \in \underline{S}$ as the following definition specifies.

Definition 2.30. Let $w \in \mathcal{I}_\theta$ and $I \subseteq S$ be a subset of generators. Then we define

$$wC_I := \{w\underline{s}_1 \dots \underline{s}_k : k \in \mathbb{N}_0, s_i \in S\}$$

as the I-**residue** of w or just **residue**. To emphasize the size of I, say $|I| = n$, we also speak of a **rank-n-residue**.

Example 2.31. Let $w \in \mathcal{I}_\theta$. Then $wC_\emptyset = \{w\}$ and $wC_S = \mathcal{I}_\theta$.

Proposition 2.32. *Let $w \in \mathcal{I}_\theta$ and $I \subset S$. If $v \in wC_I$, then $vC_I = wC_I$.*

Proof. Suppose $v \in wC_I$. Then $v = w\underline{s}_1 \dots \underline{s}_n$ for some $s_i \in I$. Suppose $u = w\underline{t}_1 \dots \underline{t}_m \in wC_I$ is any other element in wC_I with $t_i \in I$. Then

$$u = w\underline{t}_1 \dots \underline{t}_m = (v\underline{s}_n \dots \underline{s}_1)\underline{t}_1 \dots \underline{t}_m$$

and so $u \in vC_I$. This yields $wC_I \subset vC_I$. Since $w \in vC_I$ we can swap v and w to get the other inclusion. □

Corollary 2.33. *Let $v, w \in \mathcal{I}_\theta$ and $I \subset S$. Then either $vC_I \cap wC_I = \emptyset$ or $vC_I = wC_I$.*

Proof. Immediately follows from Proposition 2.32. □

Lemma 2.34. [Hul07, Lemma 5.6] *Let $w \in \mathcal{I}_\theta$, $I \subseteq S$ be a set of generators. Then there exists an unique element $w_0 \in wC_I$ with $w_0 \preceq w_0\underline{s}$ for all $s \in I$.*

Proof. [Hul07, Lemma 5.6] We reproduce the proof from loc. cit. here. Suppose there is no such element. Then for each $w \in wC_I$ we can find a $s \in I$ with $w' = w\underline{s} \preceq w$ and $e' \in wC_I$. By repetition of Deletion property for twisted expressions we get $e \in wC_I$, but e has the property, which we assumed that no element in wC_I has. Hence, there must be at least one such element. Now suppose there are two distinct elements u, v with the desired property. Note that this means that u and w have no reduced twisted expression ending with some $\underline{s} \in I$. Let v have a reduced twisted expression $v = \underline{s}_1 \dots \underline{s}_k$. Since u and v are both in wC_I there must be a twisted v-expression for u

$$u = v\underline{s}_{k+1} \dots \underline{s}_{k+l} = \underline{s}_1 \dots \underline{s}_{k+l}$$

with $s_n \in I$ for $k+1 \leq n \leq k+l$. This twisted expression cannot be reduced, since it ends with $\underline{s}_{k+l} \in I$. Then Deletion property for twisted expressions yields that this twisted expression contains a reduced twisted subexpression for u. It cannot end with $\underline{s}_n$ for $k+1 \leq n \leq k+l$. Hence, it is a twisted subexpression of $\underline{s}_1 \dots \underline{s}_k = v$, too. So $u \leq v$ by Subword property. Because of symmetry we have $v \leq u$ and so $u = v$, contradicting to our assumption $u \neq v$. □

Corollary 2.35. *Let $w \in \mathcal{I}_\theta$, $I \subseteq S$ be a set of generators and let k be the minimal twisted length within the residue wC_I. Then the element w_0 from Lemma 2.34 is the only element with twisted length k.*

Proof. The minimal twisted length k is well-defined, since the image of ρ is in $\mathbb{N}_0$, which is well-ordered, and $wC_I \neq \emptyset$. Suppose some element $v \neq w_0$ has twisted length k. Then there is a $s \in I$ with $s \in D_R(v)$ and hence $v\underline{s} \in wC_I$ with $\rho(v\underline{s}) < \rho(v) = k$. This contradicts to the minimality of k. Since at least one element has twisted length k and only w_0 is left we are done. $\square$

Definition 2.36. We denote the unique minimal element in wC_I from Corollary 2.35 by $\min(w, I)$.

We proceed with some properties of rank-2-residues. Our interest in these residues stems from the fact that their properties are needed later in Section 2.4 to construct an effective algorithm for calculating the twisted weak ordering, i.e. calculating the Hasse diagram of $Wk(W, \theta)$ for arbitrary Coxeter systems (W, S) and Coxeter system automorphisms θ.

Definition 2.37. Let $s, t \in S$ be two distinct generators. We define:

$$[\underline{st}]^n := \begin{cases} (\underline{st})^{\frac{n}{2}} & n \text{ even}, \\ (\underline{st})^{\frac{n-1}{2}}\underline{s} & n \text{ odd}. \end{cases}$$

This definition allows us to express rank-2-residues differently. Suppose we have an element $w \in \mathcal{I}_\theta$ and two distinct generators $s, t \in S$. Thanks to Proposition 2.32 and Corollary 2.35 we can assume that $w = min(w, \{s, t\})$. Then

$$wC_{\{s,t\}} = \{w\} \ \cup \ \{w[\underline{st}]^n : n \in \mathbb{N}\} \ \cup \ \{w[\underline{ts}]^n : n \in \mathbb{N}\}.$$

This encourages the following definition.

Definition 2.38. Let $w \in \mathcal{I}_\theta$ and let $s, t \in S$ be two distinct generators. Suppose $w = min(w, \{s, t\})$. Then we call $\{w[\underline{st}]^n : n \in \mathbb{N}\}$ the s**-branch** and $\{w[\underline{ts}]^n : n \in \mathbb{N}\}$ the t**-branch** of $wC_{\{s,t\}}$.

One question arises immediately: are the s- and the t-branch disjoint? With the following propositions, corollaries and lemmas we will get a much better idea of the structure of rank-2-residues and answer this question.

Proposition 2.39. *Let $w \in \mathcal{I}_\theta$ and let $s, t \in S$ be two distinct generators. Without loss of generality suppose $w = \min(w, \{s, t\})$. If there is a $v \in wC_{\{s,t\}}$ with $v\underline{s} \prec v$ and $v\underline{t} \prec v$, then it is unique with this property in $wC_{\{s,t\}}$. Hence $wC_{\{s,t\}}$ consists of two geodesics from w to v intersecting only in these two elements. Else, the s- and t-branch are disjoint, strictly ascending in twisted length and of infinite size.*

Proof. Suppose there is a v in the s-branch with $v\underline{s} \prec v$ and $v\underline{t} \prec v$, say $v = w[\underline{st}]^n$ and n is minimal with this property. Because of the uniqueness of a minimal element from Lemma 2.34 we have $w[\underline{st}]^{m+1} \prec w[\underline{st}]^m$ for all $m \in \mathbb{N}$ with $n \leq m \leq 2n - 1$.

With the same argument we have $w[\underline{st}]^{2n} = w$. If no such v exists, then the s- and t-branch must be disjoint, strictly ascending in twisted length and so of infinite size. $\square$

The assertion that Proposition 2.39 makes can be thought of some kind of convexity of rank-2-residues. A rank-2-residue cannot have a concave structure like in Figure 2.2.

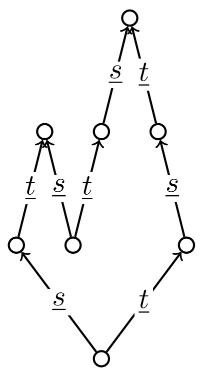

Figure 2.2.: Impossible concave structure of a rank-2-residues

Lemma 2.40. *Let $w \in S$ and $s, t \in S$ be two distinct generators with $w\underline{s} \prec w$. If $\underline{s}$ acts by multiplication on w, then $w\underline{st} \succ w\underline{s}$ or $w\underline{t} \prec w$.*

Proof. Suppose $w\underline{st} \prec w\underline{s} \prec w$, hence $l(w\underline{st}) < l(w\underline{s}) < l(w)$ in particular. If $\underline{t}$ acts by multiplication on $w\underline{s}$, then we have $l(w\underline{st}) = l(\theta(s)(wt)) = l(w) - 2$. If it acts by twisted conjugation, then we have $l(w\underline{st}) = l(\theta(t)\theta(s)(wt)) = l(w) - 3$. In both cases we have $l(wt) < l(w)$, hence $t \in D_R(w)$ and so $w\underline{t} \prec w$. $\square$

Note that this proposition could be strengthend by insisting on an exclusive or, since we cannot have both cases at the same time. By the proof of Lemma 2.28 we see that we cannot have $w\underline{st} = w$, since double edges are always twisted conjugations. Hence having $w\underline{st} \succ w\underline{s} \prec w \succ w\underline{t}$ would contradict to the convexity from Proposition 2.39. The next corollary ensures that multiplicative actions in $Wk(\theta)$ can only occur at the top or bottom end of rank-2-residues.

Corollary 2.41. *Let $w \in S$ and let $s, t \in S$ be two distinct generators and suppose $\underline{s}$ acts by multiplication on w. Then w or $w\underline{s}$ is the unique minimal or maximal element in $wC_{\{s,t\}}$.*

Proof. Suppose w is not maximal, i.e. $w\underline{t} \succ w$. Then by Lemma 2.40 we have $w\underline{st} \succ w\underline{s}$, hence $w\underline{s}$ is minimal. Suppose w is not minimal, i.e. $w\underline{st} \prec w\underline{s}$. Then with the same argument we have $w\underline{t} \prec w$, hence w is maximal. Supposing $w\underline{s}$ not to be maximal or not to be minimal yields analogue results. $\square$

Again, this corollary can be strengthend by insisting on an exclusive or with the same arguments as before.

Definition 2.42. Let $w \in \mathcal{I}_\theta$, $s, t \in S$ be two distinct generators with $\operatorname{ord}(st) < \infty$ and $C := wC_{\{s,t\}}$ the corresponding rank-2-residue. We classify rank-2-residues according to Figure 2.3.

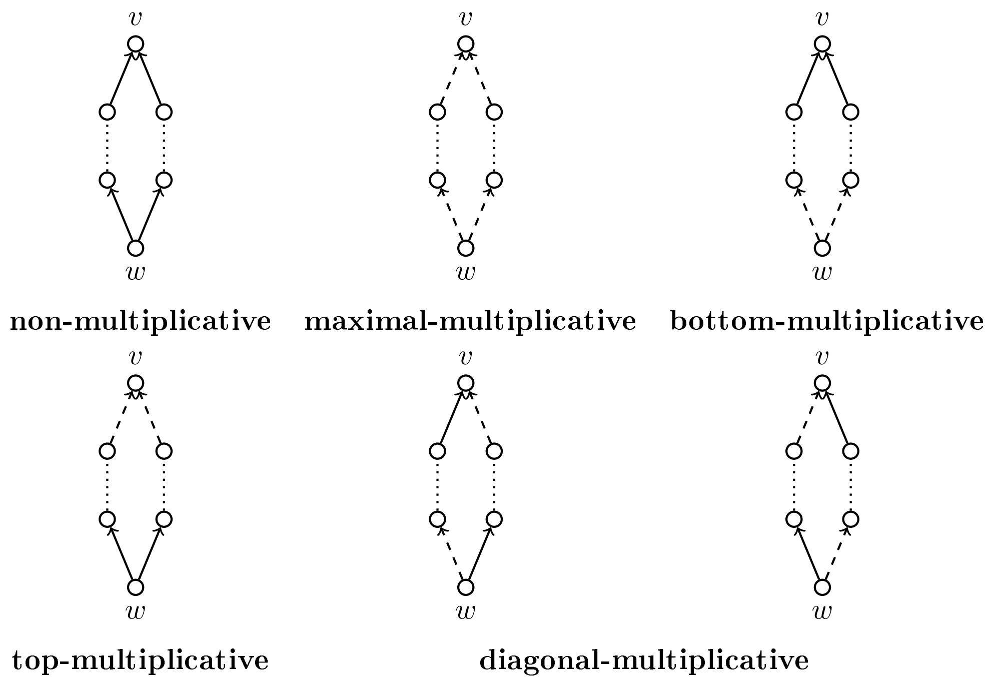

Figure 2.3.: Classification of rank-2-residues

Proposition 2.43. *Let $s, t \in S$ be two distinct generators and $w \in S$ with $w = min(w, \{s, t\})$. Suppose $v \in wC_{\{s,t\}}$ with $v\underline{s} \prec v$ and $v\underline{t} \prec v$. Then $wC_{\{s,t\}}$ is either non-, maximal-, bottom-, top- or diagonal-multiplicative. In particular the twisted conjugations and multiplications are distributed axisymmetrically or pointsymmetrically.*

Proof. If u covers w, then there are only two edges and the assumption holds. So suppose $wC_{\{s,t\}}$ contains at least four edges. Due to Corollary 2.41 the actions by multiplication can only occur next to w and v. Hence, there are $2^4 = 16$ configurations possible. Proposition 2.27 wipes out ten out of the 16 configurations. The remaining are those from Figure 2.3. □

Example 2.44. In Figure 2.4 we see three Hasse diagrams of $Wk(A_4, \mathrm{id})$. The left one only contains edges with labels s_1, s_2, the middle one only edges with labels s_1, s_3 and the right one only edges with labels s_1, s_4.

Proposition 2.45. *Let $wC_{\{s,t\}}$ be a finite rank-2-residue and suppose $w = \min(w, \{s, t\})$.*

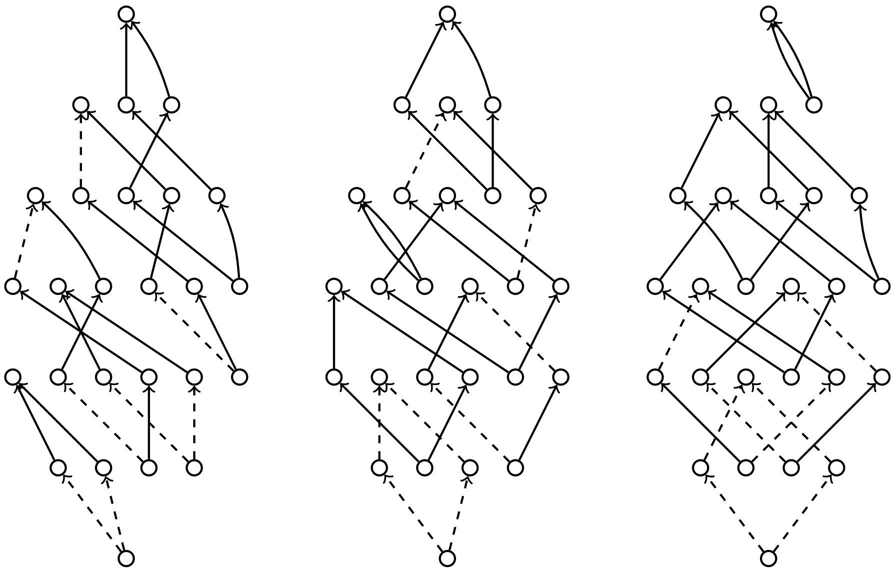

Figure 2.4.: Hasse diagrams of $Wk(A_4, \mathrm{id})$ after removing s_3, s_4 edges in the left, s_2, s_4 edges in the middle and s_2, s_3 edges in the right diagram

Let $n \in \mathbb{N}$ be the smallest number with $w[\underline{st}]^{2n} = w$. Define

$$\begin{aligned} a_1 &= l(w\underline{s}) - l(w), \\ a_2 &= l(w[\underline{st}]^n) - l(w[\underline{st}]^{n-1}), \\ a_3 &= l(w\underline{t}) - l(w), \\ a_4 &= l(w[\underline{ts}]^n) - l(w[\underline{ts}]^{n-1}). \end{aligned}$$

Then $a_1 + a_2 = a_3 + a_4$.

Proof. There are two geodesics from w to the unique maximal element in $wC_{\{s,t\}}$. By Proposition 2.43 in both geodesics at most the first and last step can be multiplicative and by Proposition 2.27 both geodesics must contain the same count of multiplicative steps. The sum $4 - a_1 - a_2$ is the count of multiplicative steps in the first geodesic while $4 - a_3 - a_4$ is the count of multiplicative steps in the second geodesic. Hence, $4 - a_1 - a_2 = 4 - a_3 - a_4 \iff a_1 + a_2 = a_3 + a_4$. □

Lemma 2.46. *Let $w \in S$, $s, t \in S$ be two distinct generators and $m = \mathrm{ord}(st) < \infty$. Then $|wC_{\{s,t\}}| \leq 2m$.*

Proof. Without loss of generality let $w = \min(w, \{s,t\})$. Let n be the smallest number (or ∞ if non such number exists) for that $w[\underline{st}]^n \succ w[\underline{st}]^{n+1}$. Then by the convexity from Proposition 2.39 we have $|wC_{\{s,t\}}| = 2n$. If $n < m$, then we are done. In return assume $n \geq m$, i.e. $w \prec w\underline{s} \prec \ldots \prec w[\underline{st}]^m$. But $w[\underline{st}]^m$ ends with $stst\cdots$ (m-times) and hence by

$$\underbrace{stst\cdots}_{m-\text{times}} = \underbrace{tsts\cdots}_{m-\text{times}}$$

we get $s, t \in D_R(w[\underline{st}]^m)$. So by Lemma 2.16 we have $w[\underline{st}]^m \succ w[\underline{st}]^{m+1}$, hence $n = m$ and so $|wC_{\{s,t\}}| = 2m$. □

Lemma 2.47. *Let $wC_{\{s,t\}}$ be a finite rank-2-residue.*

1. *If the residue is maximal-multiplicative, then* $\operatorname{ord}(st) \equiv 0 \mod 2$.
2. *If the residue is bottom-multiplicative, then* $\operatorname{ord}(st) \equiv 1 \mod 2$.
3. *If the residue is top-multiplicative, then* $\operatorname{ord}(st) \equiv 1 \mod 2$.

Proof. Let $w = \min(w, \{s,t\})$, v be the unique maximal element in the residue and n be the smallest number with $w(\underline{st})^n = w$.

1. We have $\theta(s)w = ws$ and $\theta(t)w = wt$. Hence,

$$\begin{aligned} w = w(\underline{st})^n &= (\theta(s)\theta(t))^{n-2} w(st)^{n-2} stst \\ &= (\theta(s)\theta(t))^{n-3} w(st)^{n-1} stst \\ &= \dots \\ &= w(st)^{2n-4} stst = w(st)^{2n-2}. \end{aligned}$$

   Since $2n-2$ is always even the equation chain cannot be true if $\operatorname{ord}(st)$ is odd. Therefore $\operatorname{ord}(st)$ must be even.

2. Again we have $\theta(s)w = ws$ and $\theta(t)w = wt$. Hence

$$\begin{aligned} w = w(\underline{st})^n &= (\theta(s)\theta(t))^{n-1} w(st)^{n-1} st \\ &= (\theta(s)\theta(t))^{n-2} w(st)^{n} st \\ &= \dots \\ &= w(st)^{2n-2} st = w(st)^{2n-1}. \end{aligned}$$

   Since $2n-1$ is always odd with the same argument the equation chain can only be true if $\operatorname{ord}(st)$ is odd.

3. This is analogue to the previous case when starting from v. □

Lemma 2.48. *Let $wC_{\{s,t\}}$ be a finite rank-2-residue with $m = \operatorname{ord}(st)$.*

1. *If the residue is maximal-multiplicative, then* $|wC_{\{s,t\}}| \leq m + 2$.
2. *If the residue is bottom-multiplicative, then* $|wC_{\{s,t\}}| \leq m + 1$.
3. *If the residue is top-multiplicative, then* $|wC_{\{s,t\}}| \leq m + 1$.

Proof.

1. We have an even m and $\theta(s)w = ws$ and $\theta(t)w = wt$. Hence,

$$w(\underline{st})^{m/2+1} = (\theta(s)\theta(t))^{m/2-1} w(st)^{m/2-1} stst = w(st)^m = w.$$

2. We have an odd m and $\theta(s)w = ws$ and $\theta(t)w = wt$. Hence,

$$w(\underline{st})^{(m+1)/2} = (\theta(s)\theta(t))^{(m-1)/2} w(st)^{(m-1)/2} st = w(st)^m = w.$$

3. Analogue to the previous case when starting from v. □

Proposition 2.49. *Let $w \in S$ and $s, t \in S$ be two distinct generators with* $\operatorname{ord}(st) < \infty$. *Suppose $k \in \mathbb{N}$ to be the smallest number with $w = w(\underline{st})^k$. Then for any $n \in \mathbb{N}$ with $w = w(\underline{st})^n$ we have $k \mid n$.*

Proof. Let $n = qk + r$ for $q \in \mathbb{N}_0$ and $r \in \{0, \ldots, k-1\}$. Then

$$w = w(\underline{st})^n = w(\underline{st})^{qk+r} = w(\underline{st})^{qk}(\underline{st})^r = w(\underline{st})^{q(k-1)}(\underline{st})^r = \ldots = w(\underline{st})^r.$$

For $r > 0$ we would have a contradiction to the minimality of k, hence $r = 0$, $q > 0$ and therefore $k \mid n$. □

Corollary 2.50. *Let $w \in S$ and $s, t \in S$ be two distinct generators with $w\underline{s} \neq w\underline{t}$. Suppose $w = w(\underline{st})^m = w(\underline{st})^n$. Then* $\gcd(m, n) > 1$.

Proof. Let k be the same as in Proposition 2.49. Since $w\underline{s} \neq w\underline{t}$ we have $k > 1$. Both, $k \mid n$ and $k \mid m$, hence $\gcd(m, n) \geq k > 1$. □

This constraints the possible size of rank-2-residues.

2.4. Twisted weak ordering algorithms

Now we address the problem of calculating $Wk(\theta)$ for an arbitrary Coxeter group W, given in form of a set of generating symbols $S = \{s_1, \ldots s_n\}$ and the relations in form of $m_{ij} = \operatorname{ord}(s_i s_j)$. From this input we want to calculate the Hasse diagram, i.e. the vertex set $\mathcal{I}_\theta$ and the edges labeled with $\underline{s}$. Thanks to Lemma 2.13 the vertex set can be obtained by walking the e-orbit of the action from Definition 2.5. The only element of twisted length 0 is e. Suppose we have already calculated the Hasse diagram until the twisted length k, i.e. we know all vertices $w \in \mathcal{I}_\theta$ with $\rho(w) \leq k$ and all edges connecting two vertices u, v with $\rho(u) + 1 = \rho(v) \leq k$. Let $\rho_k := \{w \in \mathcal{I}_\theta : \rho(w) = k\}$. Then all vertices in ρ_{k+1} are of the form $w\underline{s}$ for some $w \in \rho_k, s \in S$. For each $(w, s) \in \rho_k \times S$, we calculate $w\underline{s}$. If $\rho(w\underline{s}) = k + 1$ then $w \prec w\underline{s}$. To avoid having to check the twisted length we use Lemma 2.16. We already know the set $S_w \subseteq S$ of all generators yielding an edge into w. Due to the lemma we have $\rho(w\underline{s}) = k - 1$ for all $s \in S_w$ and $\rho(w\underline{s}) = k + 1$ for all $s \in S \setminus S_w$. Hence, we only calculate $w\underline{s}$ for $s \in S \setminus S_w$ and know $w \prec w\underline{s}$ without checking the twisted length explicitly. The last problem to solve is the possibility of two different $(w, s), (v, t) \in \rho_k \times S$ with $w\underline{s} = v\underline{t}$. To deal with this, we have to compare a potential new twisted involution $w\underline{s}$ with each element of twisted length $k + 1$, already calculated. The concrete problem of comparing two elements in a free presented group, called **word problem for groups**, will not be addressed here. We suppose that whatever computer system is used to implement our algorithm, supplies a suitable way to do that. The only thing to note is that solving the word problem is not a cheap operation. Reducing the count of element comparisons is a major demand to any algorithm, calculating $Wk(\theta)$. For a general approach on effective element multiplication in arbitrary Coxeter groups see [Cas01, Cas08].

The steps discussed have been compiled into an algorithm by [BHH06, Algorithm 2.4] and [HH12, Algorithm 3.1.1]. We take this as our starting point. Since the runtime is far from being optimal, we use the structural properties of rank-2-residues from Section 2.3 to improve the algorithm. As we will show these optimizations yield an algorithm with an asymptotical perfect runtime behavior. TWOA1 and its optimizations have essentially the same structure in common. This is shown in TWOABase.

Algorithm 2.51 (TWOABase)**.** This algorithm takes a Coxeter system (W, S), an involutory Coxeter system automorphism θ and an upper bound k_{max} for the maximal twisted length to consider as input. It returns the Hasse diagram of $Wk(W, \theta)$ with vertex set V and the set of directed and labled edges E.

1: **procedure** TwistedWeakOrderingAlgorithmBase($(W,S), \theta, k_{max}$)
2: $V \leftarrow \{(e, 0)\}$
3: $E \leftarrow \{\}$
4: **for** $k \leftarrow 0$ **to** k_{max} **do**
5: **for all** $(w, k_w) \in V$ **with** $k_w = k$ **do**
6: **for all** $s \in S$ **with** $\nexists(\cdot, w, s) \in E$ **do** ▷ Only for $s \notin D_R(w)$
7: **if** $w\underline{s} \notin V$ **then** ▷ Check if $w\underline{s}$ already known
8: $V \leftarrow V \cup \{(w\underline{s}, k+1)\}$
9: **end if**
10: $E \leftarrow E \cup \{(w, w\underline{s}, l(w\underline{s}) - l(w)), s\}$
11: **end for**
12: **end for**
13: $k \leftarrow k + 1$
14: **end for**
15: **return** (V, E) ▷ The poset graph
16: **end procedure**

Remark 2.52. Note that if W is finite, then we can drop k_{max} and instead use another termination condition: when k reaches the maximal twisted length in $Wk(\theta)$, then the only vertex of twisted length k is the unique element $w_0 \in W$ of maximal ordinary length. Since $s \in D_R(w_0)$ for all $s \in S$, there is no $s' \in S$ remaining to calculate $w_0\underline{s}'$ for. This condition can be checked to terminate the algorithm without knowing k_{max} before. When W is infinite, there is no maximal element and $\mathcal{I}_\theta$ is infinite, too. In this case k_{max} is used to terminate after having calculated a finite part of $Wk(\theta)$.

Lemma 2.53. *TWOABase is a deterministic algorithm if and only if the decision at line 7 is taken by a deterministic algorithm.*

Proof. The outer loop (line 4) is strictly ascending in $k \in \{0, \ldots, k_{max}\}$ and so finite. The innermost loop (line 6) is finite since S is finite and the inner loop (line 5) is finite, since V starts as finite set and in each step there are added at most $|V| \cdot |S|$ many new vertices. Therefore the algorithm terminates. The soundness is due to the arguments at the beginning of Section 2.4. □

When investigating the asymptotic runtime behavior of our algorithms, we will describe them relative to k_{max} and $n := |\{w \in \mathcal{I}_\theta : \rho(w) \leq k_{max}\}|$ while assuming

$\mathcal{O}(|S|) = 1$ and $\mathcal{O}(\mathrm{ord}(st)) = 1$ for all $s, t \in S$ with $\mathrm{ord}(st) < \infty$. We justify this by the fact that the runtime cannot be better than linear in $\mathcal{O}(n)$ (since this is the size of the vertex set V produced by our algorithms) and, moreover, $|S|$ and $\mathrm{ord}(st)$ are tiny in comparison.

Proposition 2.54. *Let A be a concrete algorithm instance of TWOABase. By this we mean an algorithm A that has the form of TWOABase together with an algorithm D to decide $w\underline{s} \in V$ at line 7. Then for $k = k_{max}$ and $n = |\{w \in \mathcal{I}_\theta : \rho(w) \leq k\}|$ we have $A \in \mathcal{O}(n \cdot D)$.*

Proof. The body of the inner loop (line 5) is executed precisely n times and the body contains of D and some instructions with constant runtime. ☐

The poset graph is build up from the unique element of rank 0, the neutral element e. Then all elements of rank 1 are calculated including all edges between elements of rank 0 and rank 1. This is repeated until the rank k_{max} is reached. As we will see, the if-statement at line 7 in TWOABase is the crucial point in the algorithm. The naive way of checking $w\underline{s} \in V$ is to calculate $w\underline{s}$ as group element in W and then do an element comparison of $w\underline{s}$ in W with all elements already in V with twisted length $k + 1$. This is exactly what TWOA1 does.

Algorithm 2.55 (TWOA1)**.** [BHH06, Algorithm 2.4] [HH12, Algorithm 3.1.1] This algorithm is based on TWOABase. It uses the following function to determine if $w\underline{s} \in V$ at line 7 in TWOABase.

procedure CheckIfAlreadyKnown($(W, S), \theta, w, s, V, E$)
    $y \leftarrow ws$
    $z \leftarrow \theta(s)y$
    if $z = w$ **then** ▷ Explicit element comparison in W
        $x \leftarrow y$
    else
        $x \leftarrow z$
    end if
    for all $(v, k_v) \in V$ **with** $k_v = k + 1$ **do** ▷ Check if x already known
        if $x = v$ **then** ▷ Explicit element comparison
            return true
        end if
    end for
    return false
end procedure

Lemma 2.56. *TWOA1 is a deterministic algorithm.*

Proof. Since the algorithm for $w\underline{s} \in V$ just compares $w\underline{s}$ with all elements in V of same twisted length it is sound. For $k \in \mathbb{N}_0$ we have $|\{w \in W : \rho(w) = k\}| < \infty$ and therefore it terminates. ☐

Lemma 2.57. *Let $k \in \mathbb{N}$, $n = |\{w \in \mathcal{I}_\theta : \rho(w) \leq k\}|$. Then TWOA1 $\in \mathcal{O}(n^2/k)$.*

Proof. We omit the detailed worst case analysis of TWOA1. Instead we give an outline of the proof. Let D be the algorithm to check $w\underline{s} \in V$. When D is executed, then $w\underline{s}$ is compared to all $w \in V$ with $\rho(w) = k+1$. If $w\underline{s}$ is the first element of twisted length $k+1$, then there is nothing to compare. If it is the last element with this twisted length that is not already known, then there are almost as many comparisons needed as elements with this twisted length exist in $\mathcal{I}_\theta$. Overall we can assume $D \in \mathcal{O}(n/k)$. By Proposition 2.54 we get TWOA1 $\in \mathcal{O}(n^2/k)$. □

Any algorithm calculating $Wk(\theta)$ must be at least linear in the size n of the resulting vertice set. Our goal is to improve TWOA1 so that we get an algorithm in $\mathcal{O}(n)$, i.e. an asymptotical perfect algorithm for calculating $Wk(\theta)$. As already seen the element comparison of a potential new element with all already known elements of same twisted length (TWOA1 at line 9) is the bottleneck. Here the rank-2-residues become key. Suppose we have a $w \in \mathcal{I}_\theta$ with $\rho(w) = k$ and $s \in S$. In TWOA1 we would now check if $w\underline{s}$ is a new vertex or if we already calculated it by comparing it with all already known vertices of twisted length $k+1$. Assume we have already calculated it. This means there is another twisted involution v with $\rho(v) = k$ and another generator $t \in S$ with $v\underline{t} = w\underline{s}$. With Proposition 2.39 $w\underline{s}$ is the unique element of maximal twisted length in the rank-2-residue $wC_{\{s,t\}}$. This yields a necessary condition for $w\underline{s}$ to be equal to an already known vertex, allowing us to replace the ineffective search all method in TWOA1 at line 9.

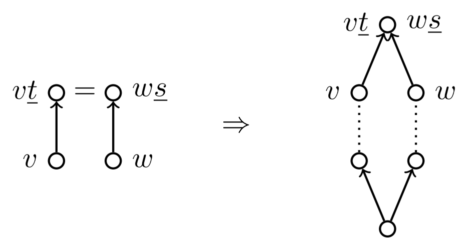

Figure 2.5.: Optimization of TWOA1

Proposition 2.58. *Let $k \in \mathbb{N}$ and suppose we are in the situation described at the beginning of Section 2.4. Let $\rho_i := \{w \in \mathcal{I}_\theta : \rho(w) = i\}$ and ρ'_{k+1} the set of the already calculated vertices with twisted length $k+1$. If $w\underline{s} \in \rho'_{k+1}$ for some $w \in \rho_k, s \in S$, say $w\underline{s} = v\underline{t}$ with $v \in \rho_k$ and $t \in S \setminus \{s\}$, then $w\underline{s} = w[\underline{ts}]^n$ for some $n \in \mathbb{N}$ with $w[\underline{ts}]^j \in \rho_0 \cup \ldots \cup \rho_k \cup \rho'_{k+1}$ for $1 \leq j \leq n$.*

Proof. The equality $w\underline{s} = w[\underline{ts}]^n$ for some $n \in \mathbb{N}$ is due to Proposition 2.39. All vertices in this rank-2-residue except $v\underline{t}$ have a twisted length of k or lower. For $v\underline{t}$ we supposed it is already known, hence $v\underline{t} \in \rho'_{k+1}$. Therefore all vertices $w[\underline{ts}]^j$, $1 \leq j \leq n$ are in $\rho_0 \cup \ldots \cup \rho_k \cup \rho'_{k+1}$. □

This can be checked effectively. Both, w and s are fixed. Start with $M = \emptyset$. For all already known edges from or to w being labeled with $\underline{t} \in \underline{S} \setminus \{\underline{s}\}$ we do the following: walk $w[\underline{ts}]^i$ for $i = 0, 1, \ldots$ until $\rho(w[\underline{ts}]^i) = k+1$. Note that walking in this case really means walking the graph. All involved vertices and edges have already been calculated. So there is no need for more calculations in W to find $w[\underline{ts}]^i$. By Proposition 2.39 such a path must exist (in a completely calculated graph). But we could be in the case, where the last step from $w[\underline{ts}]^{i-1}$ to $w[\underline{ts}]^i$ has not been calculated yet. If it is already calculated, then add this element to M by setting $M = M \cup \{w[\underline{ts}]^i\}$. If not, do not add it to M.

Now M contains all already known elements of twisted length $k+1$, satisfying the necessary condition from Proposition 2.58. Furthermore, $|M| < |S|$. So for each pair (w, s) we have to do at most $|S| - 1$ many element comparisons the determine if $w\underline{s}$ is new or already known, no matter how many elements of twisted length $k+1$ are already known. This can be used to massively improve TWOA1.

Algorithm 2.59 (TWOA2)**.** This algorithm is based on TWOABase. It uses the following function to determine if $w\underline{s} \in V$ at line 7 in TWOABase.

```
procedure CheckIfAlreadyKnown((W, S), θ, w, s, V, E)
    y ← ws
    z ← θ(s)y
    if z = w then                                  ▷ Explicit element comparison in W
        x ← y
    else
        x ← z
    end if
    for all t ∈ S \ {S} do
        if ord(st) < ∞ then
            v ← w
            k ← 1
            (z_0, z_1) ← (s, t)
            while true do                          ▷ Walk wC_{s,t} ∩ V down
                e ← (v_0, v_1, a, l) ∈ E with v_1 = v and a = z_{k mod 2}
                if e = null then
                    break
                end if
                v ← v_0
                k ← k + 1
            end while
            while true do                          ▷ Walk wC_{s,t} ∩ V up the other branch
                e ← (v_0, v_1, a, l) ∈ E with v_0 = v and a = z_{k mod 2}
                if e = null then
                    break
                end if
                v ← v_1
                k ← k − 1
            end while
            if k = 0 then                          ▷ Check if ρ(v) = ρ(w) + 1
                if x = v then                      ▷ Explicit element comparison in W
```

```
32:                 return true
33:             end if
34:         end if
35:     end if
36: end for
37: return false
38: end procedure
```

Lemma 2.60. *TWOA2 is a deterministic algorithm.*

Proof. The outer loop (line 9) is executed $|S| - 1$ times. Its body is only called if $\mathrm{ord}(st)$ is finite. Due to Lemma 2.46 both inner while loops (lines 14,22) are executed at most $2 \cdot \mathrm{ord}(st)$ times. So TWOA2 terminates. The soundness of this improvement is due to Proposition 2.58. ☐

Lemma 2.61. *Let $k \in \mathbb{N}$, $n = |\{w \in \mathcal{I}_\theta : \rho(w) \leq k\}|$. Then $TWOA2 \in \mathcal{O}(n)$.*

Proof. Let D be the algorithm to check $w\underline{s} \in V$. As seen in the proof of Lemma 2.60, the execution count for each while loop in D does not exceed

$$(|S| - 1) \cdot \max\{\mathrm{ord}(st) : t \in S \setminus \{s\}, \mathrm{ord}(st) < \infty\}.$$

Since we considered $|S|$ and $\mathrm{ord}(st)$ constant we have $D \in \mathcal{O}(1)$ and so with Proposition 2.54 we have TWOA2 $\in \mathcal{O}(n)$. ☐

Many more explicit element comparisons can be avoided. In some cases we can deduce the equality $v\underline{t} = w\underline{s}$ as well as $l(w\underline{s}) - l(w)$ just from the already calculated structure of the rank-2-residue $wC_{\{s,t\}}$, while in other cases we can preclude that $v\underline{t}$ equals $w\underline{s}$. The following two corollaries show examples of restrictions that rank-2-residues are subjected to:

Corollary 2.62. *Let $w \in \mathcal{I}_\theta$ with $\rho(w) = k$, s, t be two distinct generators and $s \notin D_R(w)$. Suppose $n \in \mathbb{N}$ to be the smallest number for that $\rho(w[\underline{ts}]^{2n-1}) = k + 1$ holds. Then:*

1. *If $n = \mathrm{ord}(st)$, then $w[\underline{ts}]^{2n-1} = w\underline{s}$.*
2. *If $n \geq 2$ and $l(w[\underline{ts}]^{2n-1}) - l(w[\underline{ts}]^{2n-2}) = 1$, then $w[\underline{ts}]^{2n-1} = w\underline{s}$.*

Proof. 1. Follows immediately from Lemma 2.46.

2. Because of the length difference the step from $w[\underline{ts}]^{2n-2}$ to $w[\underline{ts}]^{2n-1}$ is a multiplication, not a twisted conjugation, and because of $n \geq 1$ this step cannot be next to the smallest element in $wC_{\{s,t\}}$. Hence, $w[\underline{ts}]^{2n-1} = w\underline{s}$ by Corollary 2.41. ☐

Corollary 2.63. *Let $w \in S$ and $s, t \in S$ be two distinct generators. Then the following table shows all possible $n \in \mathbb{N}$ with $w(\underline{st})^n = w$ regarding $\mathrm{ord}(st)$ and the distribution of multiplications and twisted conjugations in $wC_{\{s,t\}}$ (see Figure 2.3).*

	ord(st)						
	2	*3*	*4*	*5*	*6*	*7*	*8*
non-multiplicative	*1,2*	*3*	*2,4*	*5*	*2,3,4,6*	*7*	*2,4,6,8*
diagonal-multiplicative	*2*	*3*	*2,4*	*5*	*2,3,4,6*	*7*	*2,4,6,8*
maximal-multiplicative	*2*	–	*3*	–	*2,4*	–	*5*
bottom- and top-multiplicative	–	*2*	–	*3*	–	*2,4*	–

Proof. In each case we get a m with $w = (\underline{st})^m$ by Lemma 2.46 and Lemma 2.48. By Corollary 2.50 any n with this property has a non trivial divisor in common with m if $w\underline{s} \neq w\underline{t}$. The situation $w\underline{st} = w$ for $s \neq t$ can only occur if $\mathrm{ord}(st) = 2$ and if $\underline{s}$ and $\underline{t}$ act by twisted conjugation on w due to Lemma 2.29 and the proof of Lemma 2.28. □

We use these restrictions to further improve TWOA2:

Proposition 2.64. *Let $w \in \mathcal{I}_\theta$ with $\rho(w) = k$, $s, t \in S$ be two distinct generators with $m := \mathrm{ord}(st) < \infty$ and $n \in \mathbb{N}$ the smallest number with $\rho(w[\underline{ts}]^n) = k+1$. Note that n has to be odd in this case. We define $v := w[\underline{ts}]^{n-1}$, $h := (n+1)/2$ and*

$$\begin{aligned} a_1 &= l(w\underline{s}) - l(w) - 1, \\ a_2 &= l(w[\underline{ts}]^{h-1}) - l(w[\underline{ts}]^{h-2}) - 1, \\ a_3 &= l(w[\underline{ts}]^{h}) - l(w[\underline{ts}]^{h-1}) - 1 \text{ and} \\ a_4 &= l(w[\underline{ts}]^{2h-1}) - l(w[\underline{ts}]^{2h-2}) - 1. \end{aligned}$$

Then the following decision tree allows to decide if $v\underline{t} = w\underline{s}$ or not in many cases.

1. $h = 1$:
   a) $m = 2$:
      i. $a_4 = 1$: *Maybe $v\underline{t} = w\underline{s}$. If it is the case, then $a_1 = 1$.*
      ii. $a_4 = 0$: *Then $v\underline{t} \neq w\underline{s}$.*
   b) $m > 2$: *Then $v\underline{t} \neq w\underline{s}$.*
2. $h > 1$:
   a) $a_4 = 0$: *Then $v\underline{t} = w\underline{s}$ and $a_1 = a_3 + a_4 - a_2$.*
   b) $(a_2, a_3) = (1, 1)$:
      i. $h = m$: *Then $v\underline{t} = w\underline{s}$ and $a_1 = 1$.*
      ii. $\gcd(h, m) > 1$: *Maybe $v\underline{t} = w\underline{s}$. If it is the case, then $a_1 = 1$.*
      iii. *else: Then $v\underline{t} \neq w\underline{s}$.*
   c) $(a_2, a_3) = (1, 0)$:
      i. $h = m$: *Then $v\underline{t} = w\underline{s}$ and $a_1 = 0$.*
      ii. $\gcd(h, m) > 1$: *Maybe $v\underline{t} = w\underline{s}$. If it is the case, then $a_1 = 0$.*
      iii. *else: Then $v\underline{t} \neq w\underline{s}$.*

d) $(a_2, a_3) = (0, 0)$*:*

i. $h = (m+1)/2$*: Then* $v\underline{t} = w\underline{s}$ *and* $a_1 = 1$*.*

ii. $\gcd(h, (m+1)/2) > 1$*: Maybe* $v\underline{t} = w\underline{s}$*. If it is the case, then* $a_1 = 1$*.*

iii. else: Then $v\underline{t} \neq w\underline{s}$*.*

Proof. First of all we convince ourselves that this decision tree is complete. This is immediate, since by $h \geq 0$, $m \geq 2$ and Proposition 2.43. Suppose $h = 1$. This means $v = w$. In case $v\underline{t} = w\underline{s}$, then we have a double edge between w and $w\underline{s}$. By Lemma 2.29 this is possible only if $m = \mathrm{ord}(st) = 2$ and $a_4 = l(w\underline{t}) - l(w) - 1 = 1$. Now suppose $h > 1$ and $a_4 = 0$. By Lemma 2.40 either $v\underline{s} \succ v$ or $v\underline{ts} \prec v\underline{t}$. Since $h > 1$ we cannot have $v\underline{s} \succ v$, hence $v\underline{ts} \prec v\underline{t}$. Then $v\underline{t}$ is the unique maximal element in $wC_{\{s,t\}}$ and so $w\underline{s} = v\underline{t}$. Now suppose $h > 1$ and $a_4 = 1$ and furthermore suppose $(a_2, a_3) = (1, 1)$ (the other cases are analogue). If $h = m$, then by Lemma 2.46 $v\underline{t}$ is again the unique maximal element and $v\underline{t} = w\underline{s}$. If $h < m$ then by Corollary 2.50 $v\underline{t} = w\underline{s}$ is only possible if $\gcd(h, m) > 1$. In all cases the deduction of a_1 is possible with Proposition 2.45. □

Algorithm 2.65 (TWOA3)**.** In general this algorithm proceeds like TWOA2. But instead of comparing $w\underline{s}$ with the list of all possible already known elements $v\underline{t}$, it uses the decision tree from Proposition 2.64 to either directly find $v\underline{t}$ with $v\underline{t} = w\underline{s}$ or at least to sort out elements from the list that cannot be equal to $w\underline{s}$. The information needed for the decision tree, namely $w, s, t, h, a_2, a_3, a_4$ (cf. Proposition 2.64), can easily be extracted, when searching for the already calculated elements $v\underline{t}$ with $\rho(v\underline{t}) = \rho(w) + 1$. This algorithm then applies the decision tree to each of them to decide if $w\underline{s} = v\underline{t}$, or if $w\underline{s} \neq v\underline{t}$ or if explicit element comparison is needed, to get a final answer to this question. We will omit the concrete details and refer to the appendix, where an implementation of this algorithm can be found.

Lemma 2.66. *TWOA3 is a deterministic algorithm.*

Proof. By construction, TWOA3 has the same loops as TWOA2, which is a deterministic algorithm. In addition TWOA3 uses the decision tree from Proposition 2.64. Since the decision tree has no loops, is terminates and we have already proved its correctness. Hence TWOA3 is correct and it terminates. □

Lemma 2.67. *Let* $k \in \mathbb{N}$*,* $n = |\{w \in \mathcal{I}_\theta : \rho(w) \leq k\}|$*. Then* $TWOA3 \in \mathcal{O}(n)$*.*

Proof. Since the decision tree has constant runtime the asymptotical runtime of TWOA3 cannot be worse than the asymptotical runtime of TWOA2. □

2.5. Implementing the twisted weak ordering algorithms

In this section we will look at a concrete implementation of the algorithm TWOA1 from [BHH06] and [HH12] and of the improved versions TWOA2 and TWOA3 that we have just introduced. The source codes of the test implementations can be found in the appendix, Section A. They are written in [GAP], using the [IO] package for reading and writing the results to hard disk. It supplies a powerful programming language and can handle with finitely represented groups (at least the most ones we need in this paper), in particular it allows comparisons of elements in such groups. The following algorithm benchmarks have been executed on a computer running Debian Linux in Verion 6.0.5 with an Intel® Core™ i7-965 CPU (4 cores at 3.2 GHz) and 8 GiB RAM. Note that our implementations do not support multithreading.

At first we compare the count of element comparisons needed for our three algorithms. For this we calculate $Wk(W, \mathrm{id})$ for a selection of finite Coxeter systems and count the comparisons. In Figure 2.6 we see the count of needed element comparisons plotted against the size of the set of id-twisted involutions.

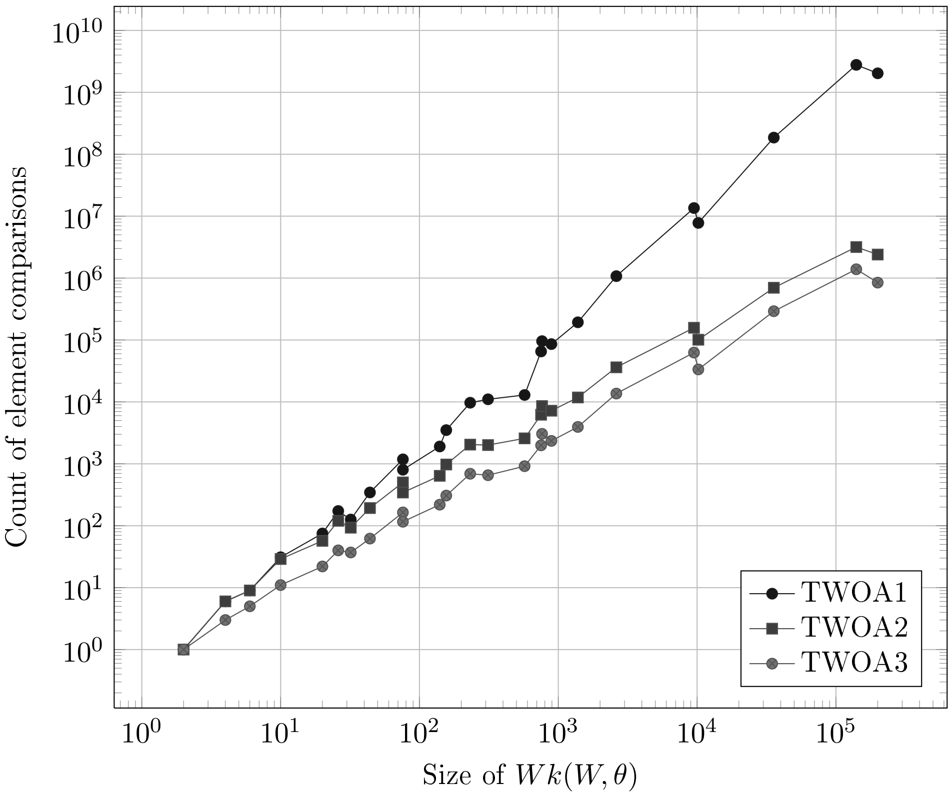

Figure 2.6.: Element comparisons needed for calculating $Wk(\mathrm{id})$ (cf. Tables B.1, B.2 and B.3)

The first observation is the much lower count of needed element comparisons of TWOA2 and TWOA3 in comparison to TWOA1, just as we intended it with our improvements. Figure 2.7 plots the runtimes against the size of $Wk(\theta)$. The complete table of benchmark results can be found in the appendix, Section B.

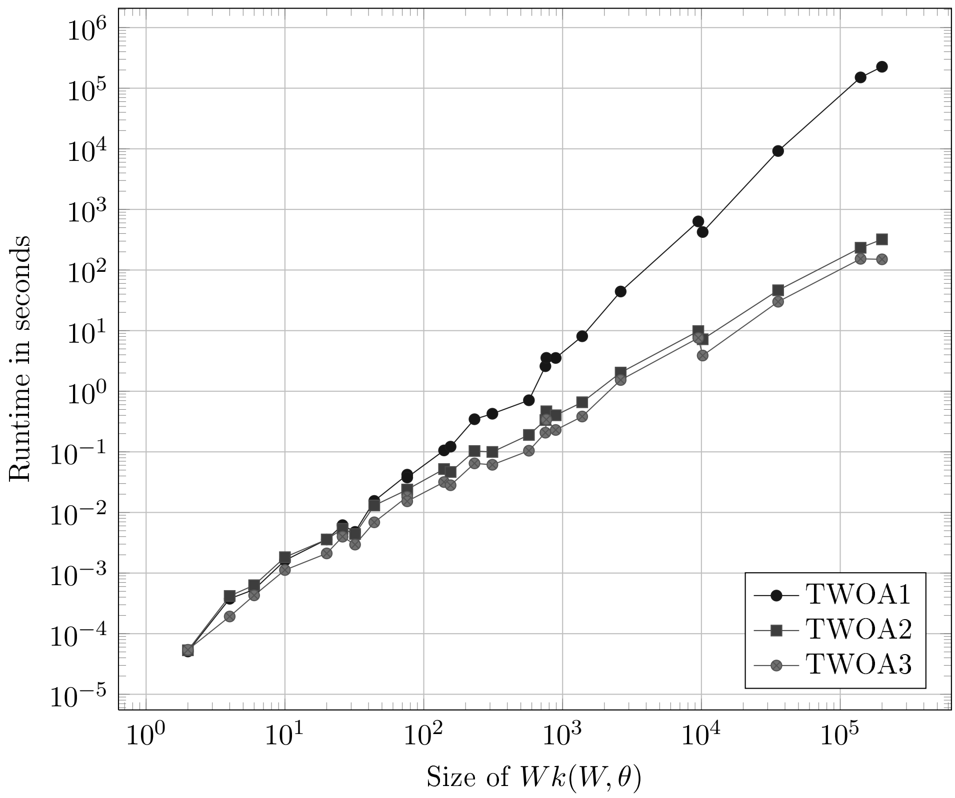

Figure 2.7.: Runtime for calculating $Wk(\mathrm{id})$ (cf. Tables B.1, B.2 and B.3)

3. Twisted weak ordering 3-residually connectedness

Definition 3.1. Let (W, S) be a Coxeter system and $\theta : W \to W$ an automorphism of W with $\theta^2 = \mathrm{id}$ and $\theta(S) = S$. We call $Wk(\theta)$ **3-residually connected** if the following holds: for every possible spherical $K \subseteq S$ with $\theta(K) = K$ and every $S_1, S_2, S_3 \subseteq S$ with pairwise non-empty intersection the statement

(3RC) $w_K C_{S_{12}} \cap w_K C_{S_{23}} \cap w_K C_{S_{31}} \subseteq w_K C_T$

holds, where w_K denotes the longest element in $\langle K \rangle$, $S_{ij} = S_i \cap S_j$ and $T = S_1 \cap S_2 \cap S_3$.

For arbitrary sets S_{12}, S_{23}, S_{31} that do not come from pairwise intersections it is easy to find pairs of Coxeter systems and Coxeter system automorphism that do not satisfy the (modified) 3-residually connectedness, as seen in Example 3.3. The following proposition shows us what distinguishes our special configuration of sets of generators from the arbitrary configuration.

Proposition 3.2. *Let M be a set and $S_{12}, S_{23}, S_{31} \subseteq M$ three subsets. Then there are three sets $S_1, S_2, S_3 \subseteq M$ with $S_{ij} = S_i \cap S_j$ iff no element $x \in M$ is precisely in two of the sets S_{ij}.*

Proof. Let S_{12}, S_{23}, S_{31} be the pairwise intersection of three sets S_1, S_2, S_3. If an element $x \in M$ is in none or in one of the sets S_i, then it is in none of the sets S_{ij}. If it is in two of the sets S_i, say $x \in S_1, S_2$, then $x \in S_{12}$, but x is not in one of the other two S_{ij}. If x is in all three S_i, then it is in all three S_{ij}, too. Hence there is no $x \in M$ that is in precisely two of the sets S_{ij}. Conversely, suppose S_{12}, S_{23}, S_{31} to be arbitrary with the constraint that there is no element $x \in M$ in precisely two of them. Then we can construct three sets S_1, S_2, S_3, whose pairwise intersections coincides with the sets S_{ij} by $x \in S_i \wedge x \in S_j$ if and only if $x \in S_{ij}$. With this construction and the previous considerations, it is clear that these S_i have the S_{ij} as pairwise intersection. Note that this construction is not unique in general, since when there is a $x \in M$ that is in none of the sets S_{ij}, then we could add it to S_1, S_2 or S_3 or just omit it without changing their pairwise intersection. □

3.1. Special cases

In this section, we investigate some results and examples in special situations. We fix some notation, namely let $K \subseteq S$ be fixed by θ and spherical, $S_1, S_2, S_3 \subseteq S$ have a pairwise non-empty intersection, $S_{ij} = S_i \cap S_j$, $T = S_1 \cap S_2 \cap S_3$ and w_K denote the longest element in $\langle K \rangle$.

Example 3.3. Let $W = A_3$ and θ be the Coxeter system autmorphism swapping s_1 and s_3 and let $w = s_1 s_3 = s_3 s_1$. We have $e\underline{s_1} = s_3 s_1 = w = s_1 s_3 = e\underline{s_3}$. Hence, $w \in eC_{\{s_1\}}$ and $w \in eC_{\{s_3\}}$ but $w \notin eC_{\{s_1\}\cap\{s_1\}\cap\{s_3\}} = eC_\emptyset = \{e\}$.

Such a trivial counterexample like in Example 3.3 can not occur in the situation from Definition 3.1.

Proposition 3.4. *Let $w, v \in \mathcal{I}_\theta$ with $\rho(v) - \rho(w) = 1$ and let $v \in wC_{S_{ij}}$ for $1 \le i < j \le 3$. Then we have $v \in wC_T$.*

Proof. By Lemma 2.28 there are at most two (not necessarily distinct) $s, t \in S$ with $w\underline{s} = v$ and $w\underline{t} = v$. Each set S_{12}, S_{23}, S_{31} must at least contain s or t, hence, s or t is at least in two sets, say $s \in S_{12}, S_{23}$. Hence, $s \in S_1, S_2, S_3$ and therefore $v \in wC_T$. ☐

A property that is much stronger than 3-residually connectedness reads $wC_I \cap wC_J = wC_{I\cap J}$. If $Wk(\theta)$ satisfies this, then its 3-residually connectedness could be concluded immediately. Unfortunately, it proves to be false in general. Again, double-edges yield a simple counterexample.

Example 3.5. Let $w \in \mathcal{I}_\theta$ and s, t be two distinct generators with $w\underline{s} = w\underline{t} = v$. Then $wC_{\{s\}} \cap wC_{\{t\}} = \{w, v\} \neq \{w\} = wC_\emptyset = wC_{\{s\}\cap\{t\}}$.

Proposition 3.6. *Suppose one of the following cases is current for some distinct $i, j \in \{1, 2, 3\}$:*

1. *$S_i = \emptyset$,*
2. *$S_i \subseteq S_j$ or*
3. *$S_i = S$.*

*Then **(3RC)** holds.*

Proof.

1. We have $\bigcap_{1\le m<n\le 3} w_K C_{S_{mn}} \subseteq w_K C_{S_{ij}} \subseteq w_K C_{S_i} = w_K C_\emptyset = w_K C_T$.
2. We have $S_{ij} = S_i$, hence $T = S_i \cap S_j \cap S_k = S_{ij} \cap S_k = S_i \cap S_k = S_{ik}$. Therefore $\bigcap_{1\le m<n\le 3} w_K C_{S_{mn}} \subseteq w_K C_{S_{ik}} = w_K C_T$.
3. We have $S_j \subseteq S = S_i$ and so with 3.6.2 we are done. ☐

Corollary 3.7. *Suppose $|S| \le 2$. Then $Wk(\theta)$ is 3-residually connected.*

Proof. If one set of S_1, S_2, S_3 is empty or equal to S, then we are done by 3.6.1 and 3.6.3. Else at least two sets of S_1, S_2, S_3 must be equal. In this case we are done by 3.6.2. ☐

Lemma 3.8. *Suppose $|S| \le 3$. Then $Wk(\theta)$ is 3-residually connected.*

Proof. The pairwise intersections of S_1, S_2, S_3 must be non-empty. By 3.6.2 every configuration where on S_i contains one other S_j cannot yield a counterexample to **(3RC)**. It remains essentially a single configuration to consider: $S_{12} = \{s_1\}, S_{23} = \{s_2\}, S_{31} = \{s_3\}$. But this cannot yield a counterexample either, since by Lemma 2.28 we have $wC_{\{s_1\}} \cap wC_{\{s_2\}} \cap wC_{\{s_3\}} = \{w\} = wC_\emptyset = wC_T$. $\square$

Remark 3.9. It might be possible to handle the case $|S| = 4$ in a similar way. Having Proposition 3.6 in mind there are essentially three configurations left for S_{12}, S_{23}, S_{31} to consider. The first configuration is $S_{12} = \{s_1\}$, $S_{23} = \{s_2\}$, $S_{31} = \{s_3\}$ is analogue to Lemma 3.8 and so cannot yield a counter example to **(3RC)**. Finally, there are two configurations left to investigate:

1. $S_{12} = \{s_1\}$, $S_{23} = \{s_2\}$, $S_{31} = \{s_3, s_4\}$.
2. $S_{12} = \{s_1, s_4\}$, $S_{23} = \{s_2, s_4\}$, $S_{31} = \{s_3, s_4\}$.

Proposition 3.10. *Let $\theta = \mathrm{id}$. Then the poset $Wk(\mathrm{id})$ is 3-residually connected if and only if **(3RC)** holds for all pairs K, S_1, S_2, S_3 that satisfy $S_{ij} \setminus T \subseteq K$ for $1 \le i < j \le 3$.*

Proof. If $Wk(\mathrm{id})$ is 3-residually connected, then **(3RC)** holds for all pairs. In return, suppose $Wk(\mathrm{id})$ not to be 3-residually connected, i.e. there is a $w \in \mathcal{I}_{\mathrm{id}}$ with $w = s_1 \dots s_n \in w_K C_{S_{12}} \cap C_{S_{23}} \cap C_{S_{31}}$ and $w \notin w_K C_T$. Since $w \in w_K C_{S_{12}}$ and Lemma 1.22 every reduced expression for w (and $s_1 \dots s_n$ in particular) uses only generators from $K \cup S_{12}$. This is true with the same argument for $K \cup S_{23}$ and $K \cup S_{31}$. Hence,

$$s_i \in (K \cup S_{12}) \cap (K \cup S_{23}) \cap (K \cup S_{31}) = K \cup (S_{12} \cap S_{23} \cap S_{31}) = K \cup T$$

for $1 \le i \le n$. So our counterexample to **(3RC)** is also a counterexample if we replace S_{ij} by $S'_{ij} := S_{ij} \cap (K \cup T)$ for $1 \le i < j \le 3$. But then

$$S'_{ij} \setminus T = [S_{ij} \cap (K \cup T)] \setminus T \subseteq [K \cup T] \setminus T \subseteq K. \qquad \square$$

Corollary 3.11. *Let $\theta = \mathrm{id}$ and $|K| \le 2$. Then **(3RC)** holds.*

Proof. Assume that **(3RC)** does not hold. Then by Proposition 3.10 we can assume that for our counterexample $S_{ij} \setminus T \subseteq K$ holds. Since $|K| \le 2$ at least two of the sets S_{ij} have to be equal. But then Proposition 3.6 says that **(3RC)** holds, which contradicts to our assumption. $\square$

3.2. Reducible case

Lemma 3.12. *Let $(W, S_1 \dot\cup S_2)$ be a reducible Coxeter system with $\mathrm{ord}(st) = 2$ for $s \in S_1, t \in S_2$. Let $\theta = \mathrm{id}$, $s_1, \dots, s_m, s \in S_1$ and $t_1, \dots, t_n, t \in S_2$. Then*

1. *$\underline{s}$ acts by twisted conjugation on $\underline{s}_1 \dots \underline{s}_m \underline{t}_1 \dots \underline{t}_n$ if and only if it acts by twisted conjugation on $\underline{s}_1 \dots \underline{s}_m$,*

2. *$\underline{t}$ acts by twisted conjugation on $\underline{s}_1 \dots \underline{s}_m \underline{t}_1 \dots \underline{t}_n$ if and only if it acts by twisted conjugation on $\underline{t}_1 \dots \underline{t}_m$, and*

3. $\underline{s}_1 \dots \underline{s}_m \underline{t}_1 \dots \underline{t}_n \underline{s} = \underline{s}_1 \dots \underline{s}_m \underline{s} \underline{t}_1 \dots \underline{t}_n$.

Proof. We have $\underline{s}_1 \dots \underline{s}_m \underline{t}_1 \dots \underline{t}_n = t_{i_q} \cdots t_{i_1} s_{j_r} \cdots s_{j_1} s_1 \cdots s_m t_1 \cdots t_n$ for some well chosen indices $1 \leq i_1 < \dots < i_q \leq m$ and $1 \leq j_1 < \dots < j_r \leq n$.

1. We prove this by a straight forward chain of equivalences.

$$\begin{aligned}
& s(\underline{s}_1 \dots \underline{s}_m \underline{t}_1 \dots \underline{t}_n)s = \underline{s}_1 \dots \underline{s}_m \underline{t}_1 \dots \underline{t}_n \\
\iff & s(t_{i_q} \cdots t_{i_1} s_{j_r} \cdots s_{j_1} s_1 \cdots s_m t_1 \cdots t_n)s = t_{i_q} \cdots t_{i_1} s_{j_r} \cdots s_{j_1} s_1 \cdots s_m t_1 \cdots t_n \\
\iff & (t_{i_q} \cdots t_{i_1} t_1 \cdots t_n) s s_{j_r} \cdots s_{j_1} s_1 \cdots s_m s = (t_{i_q} \cdots t_{i_1} t_1 \cdots t_n) s_{j_r} \cdots s_{j_1} s_1 \cdots s_m \\
\iff & s s_{j_r} \cdots s_{j_1} s_1 \cdots s_m s = s_{j_r} \cdots s_{j_1} s_1 \cdots s_m \\
\iff & s(\underline{s}_1 \dots \underline{s}_m)s = \underline{s}_1 \dots \underline{s}_m
\end{aligned}$$

2. This part is almost the same as before.

$$\begin{aligned}
& t(\underline{s}_1 \dots \underline{s}_m \underline{t}_1 \dots \underline{t}_n)t = \underline{s}_1 \dots \underline{s}_m \underline{t}_1 \dots \underline{t}_n \\
\iff & t(t_{i_q} \cdots t_{i_1} s_{j_r} \cdots s_{j_1} s_1 \cdots s_m t_1 \cdots t_n)t = t_{i_q} \cdots t_{i_1} s_{j_r} \cdots s_{j_1} s_1 \cdots s_m t_1 \cdots t_n \\
\iff & t t_{i_q} \cdots t_{i_1} t_1 \cdots t_n t (s_{j_r} \cdots s_{j_1} s_1 \cdots s_m) = t_{i_q} \cdots t_{i_1} t_1 \cdots t_n (s_{j_r} \cdots s_{j_1} s_1 \cdots s_m) \\
\iff & t t_{i_q} \cdots t_{i_1} t_1 \cdots t_n t = t_{i_q} \cdots t_{i_1} t_1 \cdots t_n \\
\iff & t(\underline{t}_1 \dots \underline{t}_n)t = \underline{t}_1 \dots \underline{t}_n
\end{aligned}$$

   Note that the last equivalence is not true in general. Suppose $v \in \mathcal{I}_\theta$ to be an arbitrary twisted expression. In general we cannot deduce the action of $\underline{s}$ on a subexpression of v from the action of $\underline{s}$ on v itself. But with the first part of this lemma we can first conclude that $\underline{t}_1$ acts by twisted conjugation on e if and only if it acts by twisted conjugation on $\underline{s}_1 \dots \underline{s}_m$. Again with the same argument $\underline{t}_2$ acts by twisted conjugation on $\underline{t}_1$ iff it acts by twisted conjugation on $\underline{s}_1 \dots \underline{s}_m \underline{t}_1$ and so forth.

3. To avoid having to repeat the proof for twisted conjugative and multiplicative action of $\underline{s}$ we set $s' = s$ if $\underline{s}$ acts by twisted conjugation and else $s' = e$.

$$\begin{aligned}
& \underline{s}_1 \cdots \underline{s}_m \underline{t}_1 \cdots \underline{t}_n \underline{s} \\
= & s'(t_{i_q} \cdots t_{i_1} s_{j_r} \cdots s_{j_1} s_1 \cdots s_m t_1 \cdots t_n)s \\
= & t_{i_q} \cdots t_{i_1} (s' s_{j_r} \cdots s_{j_1} s_1 \cdots s_m s) t_1 \cdots t_n \\
= & t_{i_q} \cdots t_{i_1} (s_1 \dots \underline{s}_m \underline{s}) t_1 \cdots t_n \\
= & s_1 \dots \underline{s}_m \underline{s} \underline{t}_1 \dots \underline{t}_n
\end{aligned}$$

   Again note that the last to equalities need the two previous parts of this lemma.

□

Corollary 3.13. *Let $(W, S_1 \dot\cup S_2)$ be Coxeter system with* $\mathrm{ord}(st) = 2$ *whenever* $s \in S_1, t \in S_2$. *In particular W is reducible. Let $W := W_{S_1}$ and $W_2 := W_{S_2}$ be the parabolic subgroups of W corresponding to S_1 and S_2. Then we have $Wk(W, \mathrm{id}) \cong Wk(W_1, \mathrm{id}) \times Wk(W_2, \mathrm{id})$.*

Proof. We denote the relation in W (resp. in W_1, W_2) by $\preceq_W$ (resp. by $\preceq_{W_1}$, $\preceq_{W_2}$). By Lemma 3.12 for every element $w \in \mathcal{I}_{\mathrm{id}}(W)$ we can find a twisted expression like $w = \underline{s}_1 \dots \underline{s}_m \underline{t}_1 \dots \underline{t}_n$ with $s \in S_1, t \in S_2$. Hence, the map

$$\varphi : \mathcal{I}_{\mathrm{id}}(W_1) \times \mathcal{I}_{\mathrm{id}}(W_2) \to \mathcal{I}_{\mathrm{id}}(W) : (\underline{s}_1 \dots \underline{s}_m, \underline{t}_1 \dots \underline{t}_n) \mapsto \underline{s}_1 \dots \underline{s}_m \underline{t}_1 \dots \underline{t}_n$$

is surjective. The injectivity is due to Proposition 1.24. It remains to show that $\preceq_W$ satisfies Definition 1.7. Let $v_1, w_1 \in \mathcal{I}_{\mathrm{id}}(W_1)$, $v_2, w_2 \in \mathcal{I}_{\mathrm{id}}(W_2)$ and $v = v_1 v_2 = \varphi(v_1, v_2), w = w_1 w_2 = \varphi(w_1, w_2) \in \mathcal{I}_{\mathrm{id}}(W)$. Suppose $v_i \preceq_{W_i} w_i$ for $i = 1, 2$. Then we have

$$\begin{aligned} v_1 &= \underline{s}_1 \dots \underline{s}_m, & w_1 &= \underline{s}_1 \dots \underline{s}_m \dots \underline{s}_{m'} = v_1 \underline{s}_{m+1} \dots \underline{s}_{m'}, \\ v_2 &= \underline{t}_1 \dots \underline{t}_n \text{ and} & w_2 &= \underline{t}_1 \dots \underline{t}_n \dots \underline{t}_{n'} = v_2 \underline{t}_{n+1} \dots \underline{t}_{n'} \end{aligned}$$

for some well chosen generators $s_i \in S_1, t_i \in S_2$ and $0 \le m \le m', 0 \le n \le n'$. Hence,

$$\begin{aligned} v = v_1 v_2 &= \underline{s}_1 \dots \underline{s}_m \underline{t}_1 \dots \underline{t}_n \preceq_W \underline{s}_1 \dots \underline{s}_m \underline{t}_1 \dots \underline{t}_n \underline{s}_{m+1} \dots \underline{s}_{m'} \underline{t}_{n+1} \dots \underline{t}_{n'} \\ &= \underline{s}_1 \dots \underline{s}_m \underline{s}_{m+1} \dots \underline{s}_{m'} \underline{t}_1 \dots \underline{t}_n \underline{t}_{n+1} \dots \underline{t}_{n'} = w_1 w_2 = w. \end{aligned}$$

In return, suppose $v \preceq_W w$. Then we have

$$\begin{aligned} v &= \underline{s}_1 \dots \underline{s}_m \underline{t}_1 \dots \underline{t}_n \text{ and} \\ w &= \underline{s}_1 \dots \underline{s}_m \underline{t}_1 \dots \underline{t}_n \underline{s}_{m+1} \dots \underline{s}_{m'} \underline{t}_{n+1} \dots \underline{t}_{n'} \end{aligned}$$

for some well chosen generators $s_i \in S_1, t_i \in S_2$ and $0 \le m \le m', 0 \le n \le n'$. Again with similar arguments we have

$$\begin{aligned} v_1 &= \underline{s}_1 \dots \underline{s}_m \preceq_{W_1} \underline{s}_1 \dots \underline{s}_m \underline{s}_{m+1} \dots \underline{s}_{m'} = w_1 \text{ and} \\ w_1 &= \underline{t}_1 \dots \underline{t}_n \preceq_{W_2} \underline{t}_1 \dots \underline{t}_n \underline{t}_{n+1} \dots \underline{t}_{n'} = w_2. \end{aligned}$$ □

Remark 3.14. Note that Lemma 3.12 and Corollary 3.13 still hold if we drop the premise $\theta = \mathrm{id}$ and instead insist on $\theta(S_i) = S_i$ for $i = 1, 2$. They also remain true if we have a partition of the generator set in more than two subsets. Hence, for $(W, S_1 \dot\cup \dots \dot\cup S_n)$ with $\mathrm{ord}(st) = 2$ whenever $s \in S_i, t \in S_j, i \neq j$ we have

$$Wk(W, \mathrm{id}) = Wk(W_{S_1}, \mathrm{id}) \times \dots \times Wk(W_{S_n}, \mathrm{id}).$$

Theorem 3.15. *Let (W, S) be a reducible Coxeter system with $S = S' \cup S''$ and* $\mathrm{ord}(st) = 2$ *whenever $s \in S', t \in S''$ and let $\theta = \mathrm{id}$. Then $Wk(W, \mathrm{id})$ is 3-residually connected if and only if $Wk(W_{S'}, \mathrm{id})$ and $Wk(W_{S''}, \mathrm{id})$ are 3-residually connected.*

Proof. If $Wk(W, \mathrm{id})$ is 3-residually connected, then $Wk(W_{S'}, \mathrm{id})$ and $Wk(W_{S''}, \mathrm{id})$ are so in particular. In return, suppose $Wk(W_{S'}, \mathrm{id})$ and $Wk(W_{S''}, \mathrm{id})$ to be 3-residually connected. For a set $M \subseteq S$ we define $M' := M \cap S'$ and $M'' := M \cap S''$, hence $M = M' \,\dot\cup\, M''$. This is compatible with our definition of S_{ij} and T:

$$\begin{aligned} S_{ij} &= S_i \cap S_j = (S_i' \,\dot\cup\, S_i'') \cap (S_j' \,\dot\cup\, S_j'') = (S_i' \cap S_j') \,\dot\cup\, (S_i'' \cap S_j'') = S_{ij}' \,\dot\cup\, S_{ij}'' \\ T &= S_1 \cap S_2 \cap S_3 = (S_{12}' \,\dot\cup\, S_{12}'') \cap (S_3' \,\dot\cup\, S_3'') = (S_{12}' \cap S_3') \,\dot\cup\, (S_{12}'' \cap S_3'') = T' \,\dot\cup\, T'' \end{aligned}$$

Let $w_K = \underline{s}_1' \dots \underline{s}_{m'}' \underline{s}_1'' \dots \underline{s}_{m''}''$ with $s_i' \in K', s_i'' \in K''$. Then $w_{K'} = \underline{s}_1' \dots \underline{s}_{m'}'$ (resp. $w_{K''} = \underline{s}_1'' \dots \underline{s}_{m''}''$) is the corresponding longest element in $\langle K' \rangle \leq W_{S'} \leq W$ (resp. $\langle K'' \rangle \leq W_{S''} \leq W$). We have three twisted expressions

$$\begin{aligned} w &= w_K \underline{a}_1' \dots \underline{a}_{n'}' \underline{a}_1'' \dots \underline{a}_{n''}'' \\ &= w_K \underline{b}_1' \dots \underline{b}_{n'}' \underline{b}_1'' \dots \underline{b}_{n''}'' \\ &= w_K \underline{c}_1' \dots \underline{c}_{n'}' \underline{c}_1'' \dots \underline{c}_{n''}'' \end{aligned}$$

with $a_i', a_i'' \in S_1$, $b_i', b_i'' \in S_2$ and $c_i', c_i'' \in S_3$. Thanks to Lemma 3.12 we can assume without loss of generality that $a', b', c' \in S'$ and $a'', b'', c'' \in S''$. Hence, we have also

$$\begin{aligned} w' &= w_{K'} \underline{a}_1' \dots \underline{a}_{n'}' = s_1' \dots \underline{s}_m' \underline{a}_1' \dots \underline{a}_{n'}' \\ &= w_{K'} \underline{b}_1' \dots \underline{b}_{n'}' = s_1' \dots \underline{s}_m' \underline{b}_1' \dots \underline{b}_{n'}' \\ &= w_{K'} \underline{c}_1' \dots \underline{c}_{n'}' = s_1' \dots \underline{s}_m' \underline{c}_1' \dots \underline{c}_{n'}' \end{aligned}$$

and so $w' \in w_{K'} C_{T'}$, since **(3RC)** holds in $Wk(W_{S'}, \mathrm{id})$. Analogue we get $w'' \in w_{K''} C_{T''}$. Hence,

$$w' = \underline{s}_1' \dots \underline{s}_{m'}' \underline{d}_1' \dots \underline{d}_{l'}' \text{ and } w'' = \underline{s}_1'' \dots \underline{s}_{m''}'' \underline{d}_1'' \dots \underline{d}_{l''}''$$

for $d_i' \in T'$ and $d_i'' \in T''$. This yields a twisted expression

$$\begin{aligned} w = w'w'' &= \underline{s}_1' \dots \underline{s}_{m'}' \underline{d}_1' \dots \underline{d}_{l'}' \underline{s}_1'' \dots \underline{s}_{m''}'' \underline{d}_1'' \dots \underline{d}_{l''}'' \\ &= \underline{s}_1' \dots \underline{s}_{m'}' \underline{s}_1'' \dots \underline{s}_{m''}'' \underline{d}_1' \dots \underline{d}_{l'}' \underline{d}_1'' \dots \underline{d}_{l''}'' \\ &= w_K \underline{d}_1' \dots \underline{d}_{l'}' \underline{d}_1'' \dots \underline{d}_{l''}'' \end{aligned}$$

with $d_i', d_i'' \in T' \,\dot\cup\, T'' = T$. Thus $w \in w_K C_T$. □

3.3. Computional testing for 3-residually connectedness

In section 2.4 we introduced an effective algorithm to calculate the poset graph for an arbitrary $Wk(\theta)$ until a given maximal twisted length. So the idea is obvious to use these data and simply test for every combination of K, S_1, S_2, S_3 if they yield a counterexample to **(3RC)**. If no counterexample can be found we have proved that $Wk(\theta)$ is 3-residually connected. If (W, S) is itself finite, then there is no problem with this approach. For infinite Coxeter systems we cannot calculate the whole poset graph. But there are some infinte ones that can also be addressed in this way.

Lemma 3.16. *Suppose the parabolic subgroup $W_{S'}$ to be finite for any $S' \subsetneq S$. Define*

$$E := \{w \in \mathcal{I}_\theta : K, S_1, S_2, S_3 \subseteq S, w \in w_K C_{S_{12}} \cap w_K C_{S_{23}} \cap w_K C_{S_{31}}, w \notin w_K C_T\}$$

and assume $E \neq \emptyset$, i.e. $Wk(\theta)$ is not 3-residually connected. Then there is a $\rho_E \in \mathbb{N}_0$ with $\max_{w \in E} \rho(w) \leq \rho_E$.

Proof. For any counterexample the sets S_1, S_2, S_3 cannot be equal to S by 3.6.3, hence by assumption the sets S_1, S_2, S_3 are spherical. But then S_{12}, S_{23}, S_{31} as well as $\theta(S_{12}), \theta(S_{23}), \theta(S_{31})$ must be spherical, too. Let $w \in E$, then $w \in W_{\theta(S_{12})} w_K W_{S_{12}}$, hence $l(w) \leq l(w_{\theta(S_{12})}) + l(w_K) + l(w_{S_{12}})$. Since there is only a finite count of proper subsets of S, we can choose

$$l_E = \max_{K, S' \subsetneq S} l(w_{\theta(S')}) + l(w_K) + l(w_{S'}) \leq \max_{S' \subsetneq S} 3 \cdot l(w_{S'}) = 3 \cdot \max_{s \in S} l(w_{S \setminus \{s\}}) < \infty$$

as upper bound for $l(w), w \in E$. For all $w' \in \mathcal{I}_\theta$ we have $\rho(w') \leq l(w')$ and so $\rho_E = l_E$ is an upper bound for $\rho(w), w \in E$, too. □

Remark 3.17. Note that the proof of this proposition actually tells us how to compute the upper bound ρ_E, since the length of the longest element in a finite Coxeter group can be easily calculated. There are four families of finite Coxeter groups and six finite Coxeter groups of exceptional type (cf. Theorem 1.25). The length of the longest element in each of them can be seen in Table 3.1. For more details on how to calculate those values (resp. formulas) see [Fra01, Section 1.2] and [Hum92, Section 2.11].

Proposition 3.18. *Let $(W, S) = (W_1 \times W_2, S_1 \cup S_2)$ be a reducible Coxeter group. Then the length of the longest element in W is the sum of the lengths of the longest elements in W_1 and W_2.*

Proof. This is immediate, since $\mathrm{ord}(st) = 2$ for $s \in S_1$, $t \in S_2$. □

W	A_n	B_n	D_n	E_6	E_7	E_8	F_4	H_3	H_4	$I_2(m)$
$l(w_0)$	$n(n+1)/2$	n^2	$n(n-1)$	36	63	120	24	15	60	m

Table 3.1.: Length of longest element in finite Coxeter groups

Example 3.19. Let $W = \tilde{A}_2$ with $S = \{s_1, s_2, s_3\}$. Because of symmetry we can calculate l_E with the set $S' = \{s_1, s_2\}$. Then $\langle S' \rangle \cong A_2$, hence the length of the longest word in $\langle S' \rangle$ is 3. Therefore $l_E = \rho_E = 9$. Hence, to validate if $\tilde{A}_2$ is 3-residually connected, we only need to calculate the poset graph of $Wk(\theta)$ until we have all twisted involutions of twisted length 9.

Now we actually want to calculate the maximum element length over all proper parabolic subgroups for some Coxeter groups, where this value is finite. Table 3.2 shows the results for some interesting one, e.g. the affine Coxeter groups. In order to simplify notation in the table we sometimes do not properly distinguish between certain cases, i.e. we use types like B_{n-3} without insisting on $n \geq 5$. For $n = 3$ we would have B_0. In this case, we will consider this type as invalid possibility. For $n = 4$ we would have B_1, which is not defined either. But in this case we will consider this type to equal A_1 which seems suitable.

W	All possible types for $W_{S\setminus\{s\}}$	$\max_{s\in S} l(w_{S\setminus\{s\}})$
$\tilde{A}_1$	A_1	1
$\tilde{A}_n$	A_n	$n(n+1)/2$
$\tilde{B}_2 = \tilde{C}_2$	$B_2,\ A_1 \times A_1$	4
$\tilde{B}_n$	$B_n,\ D_n,\ D_{n-1} \times A_1,\ D_{n-m} \times B_m,\ A_3 \times B_{n-3},\ B_{n-2} \times A_1 \times A_1$	n^2
$\tilde{C}_n$	$B_n,\ B_{n-1} \times A_1,\ B_{n-m} \times B_m$	n^2
$\tilde{D}_n$	$D_n,\ D_{n-2} \times A_1 \times A_1,\ D_{n-m} \times D_m$	$n(n-1)$
$\tilde{E}_6$	$E_6,\ A_5 \times A_1,\ A_2 \times A_2 \times A_2$	36
$\tilde{E}_7$	$E_7,\ D_6 \times A_1,\ A_5 \times A_2,\ A_7,\ A_3 \times A_3 \times A_1$	63
$\tilde{E}_8$	$E_8,\ E_7 \times A_1,\ E_6 \times A_2,\ D_5 \times A_3,\ A_4 \times A_4,\ A_5 \times A_2 \times A_1,\ A_8,\ A_7 \times A_1,\ D_8$	120
$\tilde{F}_4$	$F_4,\ B_3 \times A_1,\ A_2 \times A_2,\ A_3 \times A_1,\ B_4$	24
$\tilde{G}_2$	$I_2(6),\ A_1 \times A_1,\ A_2$	6
$X_3^1(p)$	$I_2(p),\ A_2,\ A_1 \times A_1$	p
$X_3^2(p,q)$	$I_2(p),\ I_2(q),\ A_1 \times A_1$	q
$X_3^3(p,q,r)$	$I_2(p),\ I_2(q),\ I_2(r)$	r
X_4^1	$H_3,\ I_2(5) \times A_1,\ B_2 \times A_1,\ B_3$	15
X_4^2	$H_3,\ A_2 \times A_1$	15
X_4^3	$H_3,\ I_2(5) \times A_1$	15
X_4^4	$A_3,\ A_1 \times A_1 \times A_1,\ H_3$	15
X_4^5	$A_3,\ B_3$	9
X_4^6	B_3	9
X_4^7	$H_3,\ B_3$	15
X_4^8	$H_3,\ A_3$	15
X_4^9	H_3	15
X_5^1	$H_4,\ H_3 \times A_1,\ A_2 \times I_2(5),\ A_3 \times A_1,\ A_4$	60
X_5^2	$H_4,\ H_3 \times A_1,\ B_2 \times I_2(5),\ B_3 \times A_1,\ B_4$	60
X_5^3	$H_4,\ H_3 \times A_1,\ I_2(5) \times I_2(5)$	60
X_5^4	$D_4,\ A_3 \times A_1,\ I_2(5) \times A_1 \times A_1,\ H_4$	60
X_5^5	$H_4,\ F_4$	60

Table 3.2.: Maximum element lengths in proper parabolic subgroups

Theorem 3.20. *For all pairs (W, θ) of Coxeter groups and Coxeter system automorphisms from Table 3.3 the twisted weak ordering $Wk(W, \theta)$ is 3-residually connected.*

Proof. The cases from the table have been proved by a simple algorithm. In the calculated poset graph we iterate over all combinations of generator sets K, S_1, S_2, S_3 (and ignore some trivial combinations for that we already know that they cannot yield

W	θ
A_n, $4 \leq n \leq 9$	id, $(s_1, \dots, s_n) \mapsto (s_n, \dots, s_1)$
BC_n, $4 \leq n \leq 7$	id
D_n, $4 \leq n \leq 6$	id
E_6	id, $(s_1, s_2, s_3, s_4, s_5, s_6) \mapsto (s_6, s_5, s_3, s_4, s_2, s_1)$
E_7	id
E_8	id
F_4	id
H_4	id
$\tilde{A}_n$, $3 \leq n \leq 6$	id
$\tilde{B}_n$, $3 \leq n \leq 5$	id
$\tilde{C}_n$, $3 \leq n \leq 5$	id

Table 3.3.: Some 3-residually connected weak orderings with rank ≥ 4

a counterexample to **(3RC)**) and then walk the poset graph to extract the residues $w_K C_{S_{ij}}$ and $w_K C_T$. See the appendix for concrete details. $\square$

4. Residually connectedness of flip-flop systems

This section is heavily based on [Hor09], but we will only introduce the bare minimum needed for our purposes. So for more details refer to [Hor09].

4.1. Chamber systems

Definition 4.1. A **chamber system over** I is a pair $\mathcal{C} = (C, (\sim_i, i \in I))$, with a nonempty set C, whose members are called **chambers** and a family of equivalence relations $\sim_i$, indexed by $i \in I$ that satisfies the implication

$$c \sim_i d \wedge c \sim_j d \Rightarrow c = d \vee i = j$$

for all $c, d \in C$ and $i, j \in I$. The cardinality $|I|$ is called the **rank** of $\mathcal{C}$. For all chamber systems we will assume that they have finite rank. If for two chambers c, d we have $c \sim_i d$, then c is called **i-adjacent** to d or just **adjacent**.

So the main assertion for chamber systems is that two distinct chambers $c, d \in C$ are at most adjacent by one $i \in I$. For the rest of this section $\mathcal{C} = (C, (\sim_i, i \in I))$ will denote a chamber system.

Example 4.2. For an arbitrary Coxeter system let W act as set of chambers and for each generator $s \in S$ define an equivalence relation $w \sim_s v$ if and only if either $w = v$ or $ws = v$. That this are really equivalence relations is easy to check. So suppose $w \sim_s v$, $w \sim_t v$ for two distinct generators $s, t \in S$. The assumption $w \neq v$ immediately yields a contradiction by $ws = v = wt \iff s = t$. Hence, this is indeed a chamber system.

The previous example is just a special case of a quite general recipe to create chamber systems from groups, the so-called coset chamber systems.

If two chambers $c, d \in C$ in a chamber system are not adjacent, then there might be a chain of subsequent adjacent chambers with c as first and d as last chamber.

Definition 4.3. Let $G = (c_0, \ldots, c_k)$ be a finite sequence of chambers $c_i \in C$ with c_{i-1} adjacent to c_i for all $1 \leq i \leq k$. Then G is called a **gallery** in $\mathcal{C}$ whereas the integer k is called the **length** of G. The first element c_0 of a gallery G is denoted by $\alpha(G)$ and the last by $\omega(G)$. If for two chambers $c, d \in C$ there is a gallery G with $\alpha(G) = c$ and $\omega(G) = d$, then we say that G **joins** c and d. A gallery with G with $\alpha(G) = \omega(G)$ is called **closed** and a gallery $G = (c_0, \ldots, c_k)$ with $c_{i-1} \neq c_i$ for all $1 \leq i \leq k$ is called **simple**. If a gallery G of length k joins two chambers c, d

and there is no joining gallery of shorter length, then we call G a **minimal gallery joining** c **and** d.

Definition 4.4. The chamber system $\mathcal{C}$ is called **connected** if any two chambers $c, d \in C$ can be joined by a gallery.

Definition 4.5. Let $G = (c_0, \ldots, c_k)$ be a gallery and let $J \subset I$ be a subset. If for $1 \leq i \leq k$ there is a $j \in J$ with $c_{i-1} \sim_j c_i$, then we call G a J**-gallery**. Two chambers $c, d \in C$ that have a J-gallery joining them, are called J**-equivalent**, denoted by $c \sim_J d$.

Definition 4.6. For a chamber $c \in C$ and a subset $J \subseteq I$, we call the set $R_J(c) := \{d \in C : c \sim_J d\}$ a J**-residue**. The set J is also called the **type** of a residue $R_J(c)$. If $|J| = 1$, say $J = \{i\}$, then $R_J(c) = R_{\{i\}}(c)$ is called an i**-panel**.

Note that for any chamber system $(C, (\sim_i, i \in I))$, $c \in C$ and $J \subseteq I$, the chamber system $(R_J(c), (\sim_j, j \in J))$ is connected by construction.

Definition 4.7. Let $\mathcal{C}$ be a chamber system over I. We call it a **residually connected** chamber system if the following holds: for every $J \subseteq I$ and every family of residues $(R_{I \setminus \{j\}}, j \in J)$ with pairwise nonempty intersection we have

$$\bigcap_{j \in J} R_{I \setminus \{j\}} = R_{I \setminus J}(c)$$

for some $c \in C$.

Lemma 4.8. [BC, Lemma 3.4.9] *For a connected chamber system $\mathcal{C}$ over I the following statements are equivalent.*

1. *$\mathcal{C}$ is residually connected.*
2. *If J, K, L are subsets of I and if R_J, R_K, R_L are J-, K-, L-residues which have pairwise non-empty intersections, then $R_J \cap R_K \cap R_L$ is a $(R \cap K \cap L)$-residue.*

4.2. Buildings

Definition 4.9. A **building** of type (W, S) is a pair $(\mathcal{C}, \delta)$ with a nonempty set $\mathcal{C}$ and a map $\delta : \mathcal{C} \times \mathcal{C} \to W$, called **distance function**, so that for $x, y \in \mathcal{C}$ and $w = \delta(x, y)$ we have

(Bu1) $w = e \iff x = y$;

(Bu2) for $z \in \mathcal{C}$ with $\delta(y, z) = s \in S$ we have $\delta(x, z) \in \{w, ws\}$, and if in addition $l(ws) = l(w) + 1$ then we have $\delta(x, z) = ws$;

(Bu3) for $s \in S$ there exists a $z \in \mathcal{C}$ with $\delta(y, z) = s$ and $\delta(x, z) = ws$.

For the rest of the subsection let $(\mathcal{C}, \delta)$ always be a building of type (W, S).

Definition 4.10. The cardinality of S is called the **rank** of the building.

Definition 4.11. For each $s \in S$ we define $c, d \in C$ to be s-adjacent if and only if $\delta(c,d) \in \{e, s\}$. Then $(\mathcal{C}, (\sim_s, s \in S))$ is called the **associated chamber system** to $(\mathcal{C}, \delta)$.

Proposition 4.12. *The associated chamber system is a chamber system.*

Proof. Let $c, d \in \mathcal{C}$ and $s, t \in S$ with $c \sim_s d$ and $c \sim_t d$. If $c \neq d$, then $\delta(c,d) = s$ and $\delta(c,d) = t$, hence $s = t$. □

Definition 4.13. A **gallery**, **residue** or **panel** in a building is a gallery, residue or panel in the associated chamber system.

Definition 4.14. We call the building $(\mathcal{C}, \delta)$ **thick** (resp. **thin**), if for every chamber $c \in \mathcal{C}$ and every $s \in S$ there are at least three (resp. exactly two) chambers s-adjacent to c.

Example 4.15. For a Coxeter system (W, S) define a map

$$\delta_S : W \times W \to W : (x, y) \mapsto x^{-1}y.$$

Then $\delta_S(x,y) = e \iff x = y$. Furthermore, for $z \in W$ with $\delta_S(y,z) = s$, i.e. $z = ys$, we have $\delta_S(x,z) = x^{-1}z = x^{-1}ys = \delta(x,y)s$. For $s \in S$ and $x, y \in W$ choose $z = ys$. Then $\delta_S(y,z) = s$ and as before $\delta_S(x,z) = \delta_S(x,y)s$. Hence, (W, δ_S) is a building of type (W, S). More precisely, it is a thin building, since for every $s \in S$ and $x, y \in W$ we have $\delta_S(x,y) = x^{-1}y \in \{e, s\}$ if and only if $x = y$ or $y = xs$, hence there are exactly two chambers s-adjacent to x.

This example for a thin building of type (W, S) can be indeed called “the” thin building of type (W, S) as the following lemma shows.

Theorem 4.16. [BC, Theorem 4.2.8] *Let $(\mathcal{C}, \delta)$ be thin. Then it is isometric to the building (W, δ_S) (cf. Example 4.15).*

Definition 4.17. We call a subset $\Sigma \subseteq \mathcal{C}$ an **apartment** if $(\Sigma, \delta|_\Sigma)$ is isometric to (W, δ_S) from Example 4.15, or equivalent if $(\Sigma, \delta|_\Sigma)$ is thin.

Theorem 4.18. [BC, Theorem 11.2.5] *For any two chambers $c, d \in \mathcal{C}$ there is an apartment Σ with $c, d \in \Sigma$. In particular, every building contains at least one apartment.*

Proof. The proof for the first statement can be found in [BC, Theorem 11.2.5]. The second is an immediate conclusion of the first, since because of $|S| \geq 1$ and **(Bu3)** every building must at least contain two chambers. And so there is at least one pair of chambers that has to be contained in an apartment by the first statement. □

So thin buildings are precisely those that contain exactly one apparment, i.e. are apartments themself.

Definition 4.19. The building $(\mathcal{C}, \delta)$ is called **spherical** if W is finite. In this case W has a longest element w_0 and two chambers c, d are called **opposite** if $\delta(c,d) = w_0$, denoted by c opp d.

Definition 4.20. A set of chambers $M \subseteq \mathcal{C}$ is called **connected** if any two chambers in M can be joined by a gallery completeley contained in M. If in addition, every minimal gallery joining two chambers in M is completeley contained in M, then M is called **convex**.

Remark 4.21. By definition a building itself is always connected.

4.3. Twin buildings

Definition 4.22. Let $(\mathcal{C}_+, \delta_+)$ and $(\mathcal{C}_-, \delta_-)$ be two buildings of same type (W, S). Then we call the triple $(\mathcal{C}_+, \mathcal{C}_-, \delta^*)$ with

$$\delta^* : (\mathcal{C}_+ \times \mathcal{C}_-) \cup (\mathcal{C}_- \times \mathcal{C}_+) \to W$$

a **twin building of type** (W, S) and δ^* a **codistance function** if for $\varepsilon \in \{+, -\}$, $x \in \mathcal{C}_\varepsilon$, $y \in \mathcal{C}_{-\varepsilon}$ and $w = \delta^*(x, y)$ we have

(Tw1) $\delta^*(y, x) = w^{-1}$;

(Tw2) for $z \in \mathcal{C}_{-\varepsilon}$ with $\delta_{-\varepsilon}(y, z) = s \in S$ and $l(ws) = l(w) - 1$ we have $\delta^*(x, z) = ws$;

(Tw3) for every $s \in S$ there is a $z \in \mathcal{C}_{-\varepsilon}$ with $\delta_{-\varepsilon}(y, z) = s$ and $\delta^*(x, z) = ws$.

For the rest of this subsection let $(\mathcal{C}_+, \mathcal{C}_-, \delta^*)$ be a twin building.

Definition 4.23. A **gallery**, **residue** or **panel** in a twin building $(\mathcal{C}_+, \mathcal{C}_-, \delta^*)$ is a gallery, residue or panel in either $\mathcal{C}_+$ or $\mathcal{C}_-$.

Definition 4.24. Two chambers $c \in \mathcal{C}_+$, $d \in \mathcal{C}_-$ are called **opposite**, denoted by c opp d if $\delta^*(c, d) = e$. Two residues $R_+ \subseteq \mathcal{C}_+$, $R_- \subseteq \mathcal{C}_-$ are called **opposite** if they have the same type and contain opposite chambers.

Definition 4.25. A pair (Σ_+, Σ_-) with $\Sigma_+ \subseteq \mathcal{C}_+$ and $\Sigma_- \subseteq \mathcal{C}_-$ is called a **twin apartment** if Σ_+ is an apartment in $\mathcal{C}_+$, Σ_- is an apartment in $\mathcal{C}_-$ and every chamber in $\mathcal{C}_+ \cup \mathcal{C}_-$ is precisely opposite to one other chamber in $\mathcal{C}_+ \cup \mathcal{C}_-$.

Example 4.26. [Hor09, Example 1.6.8] For an arbitrary spherical building $(\mathcal{C}_+, \delta_+)$ of type (W, S) there is a natural associated twin building $(\mathcal{C}_+, \mathcal{C}_-, \delta^*)$. Here $\mathcal{C}_-$ is just a copy of $\mathcal{C}_+$, i.e. for every chamber $c_+ \in \mathcal{C}_+$ there is a chamber $c_- \in \mathcal{C}_-$, with distance function

$$\delta_- : (c_-, d_-) \mapsto w_0 \delta_+(c_+, d_+) w_0.$$

As codistance function we defined

$$\delta^* : (c_\varepsilon, d_{-\varepsilon}) \mapsto \begin{cases} \delta_+(c_+, d_+) w_0, & \varepsilon = +; \\ w_0 \delta_+(c_+, d_+), & \varepsilon = -. \end{cases}$$

In this case, being opposite as defined for buildings and being opposite as defined for twin buildings coincide, by

$$c_+ \text{ opp } d_+ \iff \delta_+(c_+, d_+) = w_0 \iff \delta_+(c_+, d_+) w_0 = e \iff c_+ \text{ opp } d_-.$$

4.4. Building flips and flip-flop systems

In this section let $\mathcal{C} = (\mathcal{C}_+, \mathcal{C}_-, \delta^*)$ be a twin building of type (W, S).

Definition 4.27. Let θ be a permutation of $\mathcal{C}_+ \cup \mathcal{C}_-$ satisfying

(Fl1) $\theta^2 = \mathrm{id}$,

(Fl2) $\theta(\mathcal{C}_+) = \mathcal{C}_-$ and

(Fl3) for $\varepsilon \in \{+,-\}, x, y \in \mathcal{C}_+$ and $z \in \mathcal{C}_-$ we have $x \sim y \iff \theta(x) \sim \theta(y)$ and x opp $z \iff \theta(x)$ opp $\theta(z)$.

Then we call θ a **building quasi-flip** of $\mathcal{C}$. If in addition

(Fl3a) for $\varepsilon \in \{+,-\}, x, y \in \mathcal{C}_+$ and $z \in \mathcal{C}_-$ we have $\delta_\varepsilon(x, y) = \delta_{-\varepsilon}(\theta(x), \theta(y))$ and $\delta^*(x, z) = \delta^*(\theta(x), \theta(y))$,

then we call θ a **building flip** of $\mathcal{C}$.

So building (quasi-)flips permute the two halfes of a twin building while preserving adjacency and opposition and building flips also flip the distance and preserver the codistance. The next lemma gives a first idea, how building quasi-flips are coherent to the poset $Wk(\theta)$.

Lemma 4.28. [Hor09, Lemma 2.1.4] *Let θ be a building quasi-flip of $\mathcal{C}$. Then θ induces an involutory (i.e. order at most 2) Coxeter system automorphism $\tilde{\theta}$ on (W, S), so that for $\varepsilon \in \{+,-\}, x, y \in \mathcal{C}_+$ and $z \in \mathcal{C}_-$ we have $\tilde{\theta}(\delta_\varepsilon(x, y)) = \delta_{-\varepsilon}(\theta(x), \theta(y))$ and $\tilde{\theta}(\delta^*(x, z)) = \delta^*(\theta(x), \theta(z))$.*

Remark 4.29. We will not distinguish between the automorphism θ in a twin building and its induced automorphism $\tilde{\theta}$ in the Coxeter system. Instead, we will both denote by θ.

Of course the coherence between building quasi-flips and $Wk(\theta)$ is not clear by any means, but at least do building quasi-flips admit a Coxeter system and an involutory Coxeter system automorphism, hence every building quasi-flip has a corresponding twisted weak ordering poset $Wk(W, \theta)$.

Definition 4.30. For a chamber $c \in \mathcal{C}_+ \cup \mathcal{C}_-$ we call $\delta^\theta(c) := \delta^*(c, \theta(c))$ the **θ-codistance** of c and $l^\theta(c) = l(\delta^\theta(c))$ the **numerical θ-codistance** of c.

Definition 4.31. We call a building (quasi-)flip **proper** if there is a chamber $c \in \mathcal{C}_+ \cup \mathcal{C}_-$ with $\delta^\theta(c) = e \iff l^\theta(c) = 0$.

Definition 4.32. Let θ be a building quasi-flip of $\mathcal{C}$ and let $R \subseteq \mathcal{C}_+$ be an arbitrary residue. The **minimal numerical θ-codistance** of R is defined as $\min_{c \in R} l^\theta(c)$.

According to the definition of c_+ opp d_-, i.e. $l(\delta^*(c_+, d_-)) = 0$, we can consider the chambers that actually reach the minimal numerical θ-codistance as those that are mapped away "as far as possible". In particular, if $\min_{c \in R} l^\theta(c) = 0$, these are precisely those chambers, mapped to their opposite.

Definition 4.33. Let θ be a building quasi-flip of $\mathcal{C}$ and let $R \subseteq \mathcal{C}_+$ be an arbitrary residue. The (sub)chamber system of all chambers with minimal numerical θ-codistance

$$R^\theta := \{c \in R : l^\theta(c) = \min_{d \in R} l^\theta(d)\}$$

together with the equivalence relations inherited from $\mathcal{C}_+$ is called the **induced flip-flop system** on R. In case $R = \mathcal{C}_+$, we call $C^\theta := \mathcal{C}_+^\theta = R^\theta$ the **flip-flop system** associated to θ.

Definition 4.34. For a residue R of $\mathcal{C}$ we say that **direct descent into** R^θ is possible if for any chamber $c \in R$ there is a gallery from c to a chamber in R^θ such that the numerical θ-codistance l^θ is strictly decreasing along the gallery.

Definition 4.35. A residue R of $\mathcal{C}$ is called **good** if

1. R admits direct descent into R^θ and
2. R^θ is connected.

Proposition 4.36. [GHM11, Proposition 5.8] *Let θ be a quasi-flip of $\mathcal{C}$ and let all rank-2-residues be good. Then any residue Q is good. In particular, Q^θ is connected.*

Lemma 4.37. *Suppose that all rank-2-residues of $\mathcal{C}$ are good. Then every I-residue of $\mathcal{C}^\theta$ is contained in a unique I-residue of $\mathcal{C}$, and every I-residue of C intersecting C^θ non-trivially contains a unique I-residue of C^θ.*

Proof. By Proposition 4.36 every residue is good. If R be an I-residue of C^θ, then since C^θ is just a chamber subsystem of C two chambers in C^θ are s-adjacent of and only if they are s-adjacent in C. Hence, clearly there is a unique I-residue of C containing R. In return, let R be an I-residue of C intersecting C^θ non-trivially. Then $R^\theta = R \cap C^\theta$ and so R^θ is the unique I-residue of C^θ contained in R. $\square$

Definition 4.38. For an I-residue R of $\mathcal{C}^\theta$ we call the unqiue I-residue of $\mathcal{C}_+$ containing R from Lemma 4.37 its **closure** and denote it by $\overline{R}$.

Remark 4.39. Note that in the case that the correspondence of I-residues in $\mathcal{C}^\theta$ and I-residues in $\mathcal{C}_+$ is unambiguous, hence in particular if all rank-2-residues are good, then $R = \overline{R}^\theta$.

Definition 4.40. A residue R of $\mathcal{C}$ is called a **Phan residue** if R is opposite to $\theta(R)$, i.e. for every chamber $c \in R$ there is a chamber $\theta(c') \in \theta(R)$ with c opp $\theta(c')$. If a Phan residue does not contain any other Phan residue, then we call it a **minimal Phan residue**.

Definition 4.41. A quasi-flip is called **K-homogeneous** or just **homogeneous** if all minimal Phan residues are of identical type K.

Definition 4.42. Let $\mathcal{C}' \subseteq \mathcal{C}$ be two chamber systems such that the relations in $\mathcal{C}'$ are obtained by restricting those from $\mathcal{C}$. If any two chambers $c, d \in \mathcal{C}'$ are connected by a J-gallery in $\mathcal{C}'$ if and only if they are connected by a J-gallery in $\mathcal{C}$, then we say that C' **inherits connectedness from** C.

Definition 4.43. A quasi-flip of C is called **good** if it satisfies all of the following:

1. all rank-2-residues are good,
2. θ is K-homogeneous for some $K \subseteq S$ and
3. for any chamber c with $\delta^\theta(c) = w$ we have: if $s \in S$ satisfies $w_K \preceq w\underline{s} \prec w$, then there is a chamber $c' \in R_{\{s\}}(c)$ such that $\delta^\theta(c') = w\underline{s}$.

Proposition 4.44. [Hor09, Proposition 4.4.4] *Let θ be a quasi-flip of $\mathcal{C}$ and let all rank-2-residues be good. Then*

1. *θ is K-homogeneous for some K and*
2. *C^θ inherits connectedness from C_+.*

Definition 4.45. Let $w \in W$. We say a gallery $G = (c_0, \dots, c_k)$ to be of **type** v, if the following holds: if c_0 and c_1 are in the same s_1-panel, c_1 and c_2 in the same s_2-panel and so on, then $s_1 \cdots s_k = v$.

Lemma 4.46. *Let θ be a good quasi-flip and let c be a chamber with $\delta^\theta(c) = w$. If there exists a word $s_1 \cdots s_n = v \in W$ such that $w_K \underline{s}_1 \dots \underline{s}_k = w$ and $\rho(w) = \rho(w_K) + k$, then there exists a gallery of type v from c to some chamber d in C^θ.*

Proof. We induce on $k = l(v)$. If $k = 0$, i.e. $v = e$, then $w = w_K$ and we are done. So suppose $k > 0$, say v has a reduced expression $v = s_1 \cdots s_k$. Then $w_K \preceq w\underline{s}_k \prec w$. Hence, there is a chamber $c' \in R_{\{s_k\}}(c)$ with $\delta^\theta(c') = w' = w\underline{s}_k$. But since $w'\underline{s}_{k-1} \dots \underline{s}_1 = w\underline{s}_k \dots \underline{s}_1 = w_K$. By induction there is a gallery of type $s_1 \dots s_{k-1}$ from c' to d' in C^θ. Hence, there is a gallery of type v from c over c' to d'. □

The following proposition is a slight variation of [Hor09, Proposition 4.5.4].

Lemma 4.47. *Let θ be a good quasi-flip of a twin building $\mathcal{C}$ of type (W, S). If (W, S) is 3-residually connected, then the flip-flop system $\mathcal{C}^\theta$ is residually connected.*

Proof. Let R_i be residues of type J_i for $1 \le i \le 3$ such that their pairwise intersection is non-empty. By Lemma 4.8 it is sufficient to show that $R_{123} = R_1 \cap R_2 \cap R_3$ is non-empty and connected. By the residually connectedness of buildings we have $\overline{R}_{123} = \overline{R}_1 \cap \overline{R}_2 \cap \overline{R}_3 \neq \emptyset$. We choose a chamber $c \in \overline{R}_{123}$. By Proposition 4.36 we can directly descend from c to $\mathcal{C}^\theta$ by a $(J_1 \cap J_2)$-gallery. Due to symmetry this is also possible by a $(J_2 \cap J_3)$- and a $(J_3 \cap J_1)$-gallery.

Denote by K the homogeneity type of θ and by w the θ-codistance of c. Then there are words $w_{ij} \in W_{J_i \cap J_j}$ such that for each there is a directly descending gallery of type w_{ij} from c with θ-codistance w to a chamber with θ-codistance w_K. Hence, we have $w = w_K \underline{w}_{ij}$ in each case. But we assumed $Wk(W, \theta)$ to be 3-residually connected and therefore we can choose the w_{ij} from $T := J_1 \cap J_2 \cap J_3$. So by Lemma 4.46 we conclude that direct descend from c into $\mathcal{C}^\theta$ is possible via a gallery of type w_{ij}, staying in $\overline{R}_{123}$. Hence, $\overline{R}_{123} \cap \mathcal{C}^\theta \neq \emptyset$ and so $R_{123} \neq \emptyset$. Finally, Proposition 4.36 says that $R_{123} = \overline{R}^\theta_{123}$ is connected. □

Corollary 4.48. *Let $\mathcal{C}$ be a twin building of type (W, S) with a good quasi-flip θ. If (W, S) is of rank ≤ 3 or if (W, θ) is from Table 3.3, then $\mathcal{C}^\theta$ is residually connected.*

Proof. By Lemma 4.47 in combination with Lemma 3.8 and Theorem 3.20. □

5. Loose ends

Unfortunately there are many structural questions on $Wk(W, \theta)$ left unanswered. We will list some conjectures here without any further explanation.

Conjecture 5.1. If $|S| \leq 4$, then 3-residually connectedness holds.

Conjecture 5.2. If W is of type A_n, then 3-residually connectedness holds.

The following conjectures are not directly related to 3-residually connectedness, but might be useful when trying to answer some of the open questions of 3-residually connectedness.

Conjecture 5.3. Let $w \in \mathcal{I}_\theta$ and suppose $w\underline{s} = w\underline{t}$ for two distinct $s, t \in D_R(w)$. Then there are no two distinct $s', t' \in S \setminus D_R(w)$ with $w\underline{s}' = w\underline{t}'$.

Conjecture 5.4. Let $w \prec v \in \mathcal{I}_\theta$ such that w and v can be connected by exactly two distinct geodesics (note, that geodesics only consider the vertices, not the edges between the vertices). Then there are two distinct generators $s, t \in S$ such that the interval $[w, v]_\preceq$ coincides with the rank-2-residue $wC_{\{s,t\}}$.

Last but not least it might be the case that 3-residually connectedness holds for any $Wk(W, \theta)$. We will not put this as a conjecture here, since some unpleasant phenomena need Coxeter groups of huge rank to occur and due to the limitations of our computational approach we could only investigate Coxeter groups of small rank.

Appendix

A. Source codes

This section contains the most important parts of the source code. The full source code can be downloaded from `http://choffmeister.de/coxeter/3rc.zip`.

File 3rc.gap

```
Read("twistedinvolutionweakordering.gap");

tasks := [];

for n in [3..10] do Add(tasks, rec(system := CoxeterGroup_An(n), thetas := [ [1..n], Reversed([1..n]) ], kmax := -1)); od;
for n in [3..8] do Add(tasks, rec(system := CoxeterGroup_BCn(n), thetas := [ [1..n], ], kmax := -1)); od;
for n in [4..8] do Add(tasks, rec(system := CoxeterGroup_Dn(n), thetas := [ [1..n], ], kmax := -1)); od;
Add(tasks, rec(system := CoxeterGroup_E6(), thetas := [ [1..6], [6,5,3,4,2,1] ], kmax := -1));
Add(tasks, rec(system := CoxeterGroup_E7(), thetas := [ [1..7] ], kmax := -1));
Add(tasks, rec(system := CoxeterGroup_E8(), thetas := [ [1..8] ], kmax := -1));
Add(tasks, rec(system := CoxeterGroup_F4(), thetas := [ [1..4] ], kmax := -1));
Add(tasks, rec(system := CoxeterGroup_H4(), thetas := [ [1..4] ], kmax := -1));

for n in [3..7] do Add(tasks, rec(system := CoxeterGroup_TildeAn(n), thetas := [ [1..n+1] ], kmax := 3*n*(n+1)/2)); od;
for n in [3..7] do Add(tasks, rec(system := CoxeterGroup_TildeBn(n), thetas := [ [1..n+1] ], kmax := 3*n*n)); od;
for n in [3..7] do Add(tasks, rec(system := CoxeterGroup_TildeCn(n), thetas := [ [1..n+1] ], kmax := 3*n*n)); od;
for n in [4..4] do Add(tasks, rec(system := CoxeterGroup_TildeDn(n), thetas := [ [1..n+1] ], kmax := 3*n*(n-1))); od;
Add(tasks, rec(system := CoxeterGroup_TildeE6(), thetas := [ [1..7] ], kmax := 3*36));
Add(tasks, rec(system := CoxeterGroup_TildeE7(), thetas := [ [1..8] ], kmax := 3*63));
Add(tasks, rec(system := CoxeterGroup_TildeE8(), thetas := [ [1..9] ], kmax := 3*120));
Add(tasks, rec(system := CoxeterGroup_TildeF4(), thetas := [ [1..5] ], kmax := 3*24));

for p in [7..10] do Add(tasks, rec(system := CoxeterGroup_X31p(p), thetas := [ [1..3] ], kmax := 3*p)); od;
for p in [4..10] do for q in [Maximum(5, p)..10] do Add(tasks, rec(system := CoxeterGroup_X32pq(p, q), thetas := [ [1..3] ], kmax := 3*q)); od; od;
for p in [3..10] do for q in [Maximum(3, p)..10] do for r in [Maximum(4, q)..10] do Add(tasks, rec(system := CoxeterGroup_X33pqr(p, q, r), thetas := [ [1..3] ], kmax := 3*r)); od; od; od;

```

```
Add(tasks, rec(system := CoxeterGroup_X41(), thetas := [ [1..4] ], kmax := 3*15));
Add(tasks, rec(system := CoxeterGroup_X42(), thetas := [ [1..4] ], kmax := 3*15));
Add(tasks, rec(system := CoxeterGroup_X43(), thetas := [ [1..4] ], kmax := 3*15));
Add(tasks, rec(system := CoxeterGroup_X44(), thetas := [ [1..4] ], kmax := 3*15));
Add(tasks, rec(system := CoxeterGroup_X45(), thetas := [ [1..4] ], kmax := 3*9));
Add(tasks, rec(system := CoxeterGroup_X46(), thetas := [ [1..4] ], kmax := 3*9));
Add(tasks, rec(system := CoxeterGroup_X47(), thetas := [ [1..4] ], kmax := 3*15));
Add(tasks, rec(system := CoxeterGroup_X48(), thetas := [ [1..4] ], kmax := 3*15));
Add(tasks, rec(system := CoxeterGroup_X49(), thetas := [ [1..4] ], kmax := 3*15));
Add(tasks, rec(system := CoxeterGroup_X51(), thetas := [ [1..5] ], kmax := 3*60));
Add(tasks, rec(system := CoxeterGroup_X52(), thetas := [ [1..5] ], kmax := 3*60));
Add(tasks, rec(system := CoxeterGroup_X53(), thetas := [ [1..5] ], kmax := 3*60));
Add(tasks, rec(system := CoxeterGroup_X54(), thetas := [ [1..5] ], kmax := 3*60));
Add(tasks, rec(system := CoxeterGroup_X55(), thetas := [ [1..5] ], kmax := 3*60));

for task in tasks do
    W := task.system.group;
    matrix := task.system.matrix;
    kmax := task.kmax;

    for theta in task.thetas do
        # handle Wk(W,\theta)
        Print("Wk(", Name(W), ", ", theta, ")\n");
        filename := StringToFilename(Concatenation(Name(W), "-", String(List(theta, p -> p))));

        # calculate Wk(W,\theta)
        Print("- Calculating poset\n");
        TwistedInvolutionWeakOrdering3(filename, W, matrix, theta, kmax);

        # check if Wk(W,\theta) is 3-residually connected
        Print("- Checking for 3-residually connectedness\n");
        counterexample := TwistedInvolutionWeakOrderingSearchForNon3rcCase(filename);

        if counterexample <> fail then
            counterexamplefile := IO_File(Concatenation("counterexamples/", filename), "w", 1);
            IO_Write(counterexamplefile,
                "W = ", counterexample.W, "\n",
                "theta = ", counterexample.theta, "\n",
                "S = ", counterexample.S, "\n",
                "K = ", counterexample.K, "\n",
                "wK = ", counterexample.wK, "\n",
                "T = ", counterexample.T, "\n",
                "S12 = ", counterexample.S12, "\n",
                "S23 = ", counterexample.S23, "\n",
                "S31 = ", counterexample.S31, "\n",
                "w = ", counterexample.w, "\n\n\n\n");
```

```
            IO_Close(counterexamplefile);

            Print("- IS NOT 3RC *************************************\n\n");
        else
            Print("- IS 3RC                                          \n\n");
        fi;
    od;
od;
```

File twoa-3rc.gap

```
TwistedInvolutionWeakOrderingSearchForNon3rcCase := function(filename)
    local graph, S, i, j, K, wK, T, UNUSED, part, S12, S23, S31,
        resS12, resS23, resS31, resT, resDiff, theta, isId, pool;

    graph := TwistedInvolutionWeakOrderingPersistReadResults(filename);
    theta := graph.data.automorphism;
    S := [1..graph.data.rank];
    isId := theta = S;
    i := 0;

    for K in IteratorOfCombinations(S) do
        j := 0;
        i := i + 1;

        if K <> Set(theta{K}) then
            continue;
        fi;

        if isId then
            pool := K;
        else
            pool := S;
        fi;

        wK := TwistedInvolutionWeakOrderingLongestWord(graph.vertices[1], K);

        for T in Combinations(S) do
            for UNUSED in Combinations(Difference(pool, T)) do
                for part in PartitionsSet(Difference(pool, Union(T, UNUSED)),
                    3) do
                    j := j + 1;
                    Print(i, " ", j, "                          \r");

                    S12 := Union(part[1], T);
                    S23 := Union(part[2], T);
                    S31 := Union(part[3], T);

                    resS12 := TwistedInvolutionWeakOrderingResiduum(wK, Union(
                        S12, T));
                    resS23 := TwistedInvolutionWeakOrderingResiduum(wK, Union(
                        S23, T));
                    resS31 := TwistedInvolutionWeakOrderingResiduum(wK, Union(
                        S31, T));
                    resT := TwistedInvolutionWeakOrderingResiduum(wK, T);

                    resDiff := Difference(Intersection(resS12, resS23, resS31)
                        , resT);

                    if Length(resDiff) > 0 then
                        return rec(
                            W := graph.data.name,
```

```
                                theta := theta,
                                S := List(S, s -> Concatenation("s", String(s))),
                                K := List(K, s -> Concatenation("s", String(s))),
                                wK := wK.name,
                                T := List(T, s -> Concatenation("s", String(s))),
                                S12 := List(S12, s -> Concatenation("s", String(s))),
                                S23 := List(S23, s -> Concatenation("s", String(s))),
                                S31 := List(S31, s -> Concatenation("s", String(s))),
                                w := List(resDiff, n -> n.name)
                            );
                        fi;
                    od;
                od;
            od;
        od;

        return fail;
    end;
```

File twoa1.gap

```
# Calculates the poset Wk(theta).
TwistedInvolutionWeakOrdering1 := function (filename, W, matrix, theta, kmax)
    local persistInfo, maxOrder, vertices, edges, absVertexIndex, absEdgeIndex, prevVertex, currVertex, newEdge,
        label, type, deduction, startTime, endTime, S, k, i, s, x, y, n, thetamap;

    persistInfo := TwistedInvolutionWeakOrderingPersistResultsInit(filename);

    S := GeneratorsOfGroup(W);
    thetamap := GroupHomomorphismByImages(W, W, S, S{theta});
    maxOrder := Minimum([Maximum(Concatenation(matrix, [1])), 5]);
    vertices := [ [], [ rec(element := One(W), twistedLength := 0, inEdges := [], outEdges := [], absIndex := 1) ] ];
    edges := [ [], [] ];
    absVertexIndex := 2;
    absEdgeIndex := 1;
    k := 0;

    while Length(vertices[2]) > 0 do
        if kmax > -1 and k > kmax then
            break;
        fi;

        for i in [1..Length(vertices[2])] do
            Print(k, " ", i, "         \r");

            prevVertex := vertices[2][i];
            for label in Filtered([1..Length(S)], n -> Position(List(prevVertex.inEdges, e -> e.label), n) = fail) do
                x := prevVertex.element;
                s := S[label];

                type := 1;
                y := s^thetamap*x*s;
                if (CoxeterElementsCompare(x, y)) then
                    y := x * s;
                    type := 0;
```

```
                fi;

                currVertex := fail;
                for n in vertices[1] do
                    if CoxeterElementsCompare(n.element, y) then
                        currVertex := n;
                        break;
                    fi;
                od;

                if currVertex = fail then
                    currVertex := rec(element := y, twistedLength := k + 1,
                        inEdges := [], outEdges := [], absIndex :=
                        absVertexIndex);
                    Add(vertices[1], currVertex);

                    absVertexIndex := absVertexIndex + 1;
                fi;

                newEdge := rec(source := prevVertex, target := currVertex,
                    label := label, type := type, absIndex := absEdgeIndex);

                Add(edges[1], newEdge);
                Add(currVertex.inEdges, newEdge);
                Add(prevVertex.outEdges, newEdge);

                absEdgeIndex := absEdgeIndex + 1;
            od;
        od;

        TwistedInvolutionWeakOrderingPersistResults(persistInfo, vertices[2],
            edges[2]);

        Add(vertices, [], 1);
        Add(edges, [], 1);
        if (Length(vertices) > maxOrder + 1) then
            for n in vertices[maxOrder + 2] do
                n.inEdges := [];
                n.outEdges := [];
            od;
            Remove(vertices, maxOrder + 2);
            Remove(edges, maxOrder + 2);
        fi;
        k := k + 1;
    od;

    TwistedInvolutionWeakOrderingPersistResultsInfo(persistInfo, W, matrix,
        theta, absVertexIndex - 1, k - 1);
    TwistedInvolutionWeakOrderingPersistResultsClose(persistInfo);

    return rec(numVertices := absVertexIndex - 1, numEdges := absEdgeIndex -
        1, maxTwistedLength := k - 1);
end;
```

File twoa2.gap

```
# Calculates the poset Wk(theta).
TwistedInvolutionWeakOrdering2 := function (filename, W, matrix, theta, kmax)
    local persistInfo, maxOrder, vertices, edges, absVertexIndex, absEdgeIndex
        , prevVertex, currVertex, newEdge, possibleResiduums,
        label, type, deduction, startTime, endTime, S, k, i, s, x, y, n, h,
            res, thetamap;
```

```
    persistInfo := TwistedInvolutionWeakOrderingPersistResultsInit(filename);

    S := GeneratorsOfGroup(W);
    thetamap := GroupHomomorphismByImages(W, W, S, S{theta});
    maxOrder := Minimum([Maximum(Concatenation(matrix, [1])), 5]);
    vertices := [ [], [ rec(element := One(W), twistedLength := 0, inEdges := [], outEdges := [], absIndex := 1) ] ];
    edges := [ [], [] ];
    absVertexIndex := 2;
    absEdgeIndex := 1;
    k := 0;

    while Length(vertices[2]) > 0 do
        if kmax > -1 and k > kmax then
            break;
        fi;

        for i in [1..Length(vertices[2])] do
            Print(k, " ", i, "          \r");

            prevVertex := vertices[2][i];
            for label in Filtered([1..Length(S)], n -> Position(List(prevVertex.inEdges, e -> e.label), n) = fail) do
                x := prevVertex.element;
                s := S[label];

                type := 1;
                y := s^thetamap*x*s;
                if (CoxeterElementsCompare(x, y)) then
                    y := x * s;
                    type := 0;
                fi;

                possibleResiduums := DetectPossibleRank2Residuums(prevVertex, label, [1..Length(S)]);
                currVertex := fail;
                for res in possibleResiduums do
                    h := Length(res) / 2;

                    if CoxeterElementsCompare(res[h*2].vertex.element, y) then
                        currVertex := res[h*2].vertex;
                        break;
                    fi;
                od;

                if currVertex = fail then
                    currVertex := rec(element := y, twistedLength := k + 1, inEdges := [], outEdges := [], absIndex := absVertexIndex);
                    Add(vertices[1], currVertex);

                    absVertexIndex := absVertexIndex + 1;
                fi;

                newEdge := rec(source := prevVertex, target := currVertex, label := label, type := type, absIndex := absEdgeIndex);

                Add(edges[1], newEdge);
                Add(currVertex.inEdges, newEdge);
                Add(prevVertex.outEdges, newEdge);

                absEdgeIndex := absEdgeIndex + 1;
```

```
            od;
        od;

        TwistedInvolutionWeakOrderingPersistResults(persistInfo, vertices[2], edges[2]);

        Add(vertices, [], 1);
        Add(edges, [], 1);
        if (Length(vertices) > maxOrder + 1) then
            for n in vertices[maxOrder + 2] do
                n.inEdges := [];
                n.outEdges := [];
            od;
            Remove(vertices, maxOrder + 2);
            Remove(edges, maxOrder + 2);
        fi;
        k := k + 1;
    od;

    TwistedInvolutionWeakOrderingPersistResultsInfo(persistInfo, W, matrix, theta, absVertexIndex - 1, k - 1);
    TwistedInvolutionWeakOrderingPersistResultsClose(persistInfo);

    return rec(numVertices := absVertexIndex - 1, numEdges := absEdgeIndex - 1, maxTwistedLength := k - 1);
end;
```

File twoa3.gap

```
# Calculates the poset Wk(theta).
TwistedInvolutionWeakOrdering3 := function (filename, W, matrix, theta, kmax)
    local persistInfo, maxOrder, vertices, edges, absVertexIndex, absEdgeIndex, prevVertex, currVertex, newEdge, possibleResiduums,
        label, type, deduction, startTime, endTime, endTypes, S, k, i, s, x, _y, y, n, m, h, res, thetamap;

    persistInfo := TwistedInvolutionWeakOrderingPersistResultsInit(filename);

    S := GeneratorsOfGroup(W);
    thetamap := GroupHomomorphismByImages(W, W, S, S{theta});
    maxOrder := Minimum([Maximum(Concatenation(matrix, [1])), 5]);
    vertices := [ [], [ rec(element := One(W), twistedLength := 0, inEdges := [], outEdges := [], absIndex := 1) ] ];
    edges := [ [], [] ];
    absVertexIndex := 2;
    absEdgeIndex := 1;
    k := 0;

    while Length(vertices[2]) > 0 do
        if kmax > -1 and k > kmax then
            break;
        fi;

        for i in [1..Length(vertices[2])] do
            Print(k, " ", i, "         \r");

            prevVertex := vertices[2][i];
            for label in Filtered([1..Length(S)], n -> Position(List(prevVertex.inEdges, e -> e.label), n) = fail) do
                x := prevVertex.element;
                s := S[label];
                y := x*s;
```

```
            _y := s^thetamap*y;
            type := -1;

            possibleResiduums := DetectPossibleRank2Residuums(prevVertex, label, [1..Length(S)]);
            currVertex := fail;
            for res in possibleResiduums do
                m := CoxeterMatrixEntry(matrix, res[1].edge.label, res[2].edge.label);
                h := Length(res) / 2;

                if h = 1 then
                    if m = 2 and res[h*2].edge.type = 1 and CoxeterElementsCompare(res[h*2].vertex.element, _y) then
                        currVertex := res[h*2].vertex;
                        type := 1;
                        break;
                    fi;
                else
                    endTypes := [-1, res[h].edge.type, res[h+1].edge.type, res[h*2].edge.type];
                    endTypes[1] := endTypes[3] + endTypes[4] - endTypes[2];

                    if endTypes[4] = 0 then
                        currVertex := res[h*2].vertex;
                        type := endTypes[1];
                        break;
                    elif endTypes = [1,1,1,1] then
                        if m = h or (Gcd(m,h) > 1 and CoxeterElementsCompare(res[h*2].vertex.element, _y)) then
                            currVertex := res[h*2].vertex;
                            type := 1;
                            break;
                        fi;
                    elif endTypes = [0,1,0,1] then
                        if m = h or (Gcd(m,h) > 1 and CoxeterElementsCompare(res[h*2].vertex.element, y)) then
                            currVertex := res[h*2].vertex;
                            type := 0;
                            break;
                        fi;
                    elif endTypes = [1,0,0,1] and m mod 2 = 1 then
                        if (m+1)/2 = h or (Gcd((m+1)/2,h) > 1 and CoxeterElementsCompare(res[h*2].vertex.element, _y)) then
                            currVertex := res[h*2].vertex;
                            type := 1;
                            break;
                        fi;
                    fi;
                fi;
            od;

            if currVertex = fail then
                if CoxeterElementsCompare(x, _y) then
                    type := 0;
                    _y := y;
                else
                    type := 1;
```

```
                    fi;

                    currVertex := rec(element := _y, twistedLength := k + 1,
                        inEdges := [], outEdges := [], absIndex :=
                        absVertexIndex);
                    Add(vertices[1], currVertex);

                    absVertexIndex := absVertexIndex + 1;
                fi;

                newEdge := rec(source := prevVertex, target := currVertex,
                    label := label, type := type, absIndex := absEdgeIndex);

                Add(edges[1], newEdge);
                Add(currVertex.inEdges, newEdge);
                Add(prevVertex.outEdges, newEdge);

                absEdgeIndex := absEdgeIndex + 1;
            od;
        od;

        TwistedInvolutionWeakOrderingPersistResults(persistInfo, vertices[2],
            edges[2]);

        Add(vertices, [], 1);
        Add(edges, [], 1);
        if (Length(vertices) > maxOrder + 1) then
            for n in vertices[maxOrder + 2] do
                n.inEdges := [];
                n.outEdges := [];
            od;
            Remove(vertices, maxOrder + 2);
            Remove(edges, maxOrder + 2);
        fi;
        k := k + 1;
    od;

    TwistedInvolutionWeakOrderingPersistResultsInfo(persistInfo, W, matrix,
        theta, absVertexIndex - 1, k - 1);
    TwistedInvolutionWeakOrderingPersistResultsClose(persistInfo);

    return rec(numVertices := absVertexIndex - 1, numEdges := absEdgeIndex -
        1, maxTwistedLength := k - 1);
end;
```

File twoa-misc.gap

```
TwistedInvolutionWeakOrderingResiduum := function (vertex, labels)
    local visited, queue, residuum, current, edge;

    visited := [ vertex.absIndex ];
    queue := [ vertex ];
    residuum := [];

    while Length(queue) > 0 do
        current := queue[1];
        Remove(queue, 1);
        Add(residuum, current);

        for edge in current.outEdges do
            if edge.label in labels and not edge.target.absIndex in visited
                then
```

```
                Add(visited, edge.target.absIndex);
                Add(queue, edge.target);
            fi;
        od;

        for edge in current.inEdges do
            if edge.label in labels and not edge.source.absIndex in visited
                then
                Add(visited, edge.source.absIndex);
                Add(queue, edge.source);
            fi;
        od;
    od;

    return residuum;
end;

TwistedInvolutionWeakOrderingLongestWord := function (vertex, labels)
    local current;

    current := vertex;

    while Length(Filtered(current.outEdges, e -> e.label in labels)) > 0 do
        current := Filtered(current.outEdges, e -> e.label in labels)[1].
            target;
    od;

    return current;
end;

DetectPossibleRank2Residuums := function(startVertex, startLabel, labels)
    local comb, trace, v, e, k, possibleTraces;
    possibleTraces := [];

    for comb in List(Filtered(labels, label -> label <> startLabel), label ->
        rec(startVertex := startVertex, st := [startLabel, label])) do
        trace := [ rec(vertex := startVertex, edge := rec(label := comb.st[1],
             type := -1)) ];

        v := startVertex;
        e := fail;
        k := 1;

        while true do
            e := FindElement(v.inEdges, e -> e.label = comb.st[k mod 2 + 1]);
            if e = fail then
                break;
            fi;

            v := e.source;
            k := k + 1;
            Add(trace, rec(vertex := v, edge := e));
        od;

        while true do
            e := FindElement(v.outEdges, e -> e.label = comb.st[k mod 2 + 1]);
            if e = fail then
                break;
            fi;

            v := e.target;
            k := k - 1;
            Add(trace, rec(vertex := v, edge := e));
```

```
        od;

        if k = 0 then
            Add(possibleTraces, trace);
        fi;
    od;

    return possibleTraces;
end;
```

B. Benchmarks

W	$\|Wk(W, \mathrm{id})\|$	Time in seconds	Element comparisons
A_1	2	$5.087 \cdot 10^{-5}$	1
A_2	4	$3.769 \cdot 10^{-4}$	6
BC_2	6	$5.359 \cdot 10^{-4}$	9
A_3	10	$1.618 \cdot 10^{-3}$	31
BC_3	20	$3.586 \cdot 10^{-3}$	75
A_4	26	$6.173 \cdot 10^{-3}$	173
H_3	32	$4.781 \cdot 10^{-3}$	126
D_4	44	$1.551 \cdot 10^{-2}$	345
A_5	76	$4.200 \cdot 10^{-2}$	1,181
BC_4	76	$3.778 \cdot 10^{-2}$	802
F_4	140	$1.056 \cdot 10^{-1}$	1,906
D_5	156	$1.218 \cdot 10^{-1}$	3,502
A_6	232	$3.453 \cdot 10^{-1}$	9,700
BC_5	312	$4.253 \cdot 10^{-1}$	11,024
H_4	572	$7.100 \cdot 10^{-1}$	12,938
D_6	752	$2.589 \cdot 10^{0}$	65,308
A_7	764	$3.552 \cdot 10^{0}$	95,797
E_6	892	$3.540 \cdot 10^{0}$	85,857
BC_6	1,384	$8.073 \cdot 10^{0}$	193,218
A_8	2,620	$4.420 \cdot 10^{1}$	1,074,392
A_9	9,496	$6.342 \cdot 10^{2}$	13,531,414
E_7	10,208	$4.236 \cdot 10^{2}$	7,785,186
A_{10}	35,696	$9.201 \cdot 10^{3}$	185,791,174
A_{11}	140,152	$1.507 \cdot 10^{5}$	2,778,111,763
E_8	199,952	$2.258 \cdot 10^{5}$	2,029,454,701

Table B.1.: Benchmark results for TWOA1

W	$\lvert Wk(W, \mathrm{id})\rvert$	Time in seconds	Element comparisons
A_1	2	$5.322 \cdot 10^{-5}$	1
A_2	4	$4.189 \cdot 10^{-4}$	6
BC_2	6	$6.300 \cdot 10^{-4}$	9
A_3	10	$1.828 \cdot 10^{-3}$	29
BC_3	20	$3.586 \cdot 10^{-3}$	57
A_4	26	$5.369 \cdot 10^{-3}$	120
H_3	32	$4.405 \cdot 10^{-3}$	93
D_4	44	$1.304 \cdot 10^{-2}$	193
A_5	76	$2.372 \cdot 10^{-2}$	501
BC_4	76	$2.390 \cdot 10^{-2}$	344
F_4	140	$5.200 \cdot 10^{-2}$	640
D_5	156	$4.655 \cdot 10^{-2}$	975
A_6	232	$1.032 \cdot 10^{-1}$	2,043
BC_5	312	$9.964 \cdot 10^{-2}$	2,009
H_4	572	$1.900 \cdot 10^{-1}$	2,578
D_6	752	$3.347 \cdot 10^{-1}$	6,206
A_7	764	$4.667 \cdot 10^{-1}$	8,569
E_6	892	$4.013 \cdot 10^{-1}$	7,210
BC_6	1,384	$6.580 \cdot 10^{-1}$	11,794
A_8	2,620	$2.032 \cdot 10^{0}$	36,218
A_9	9,496	$9.837 \cdot 10^{0}$	157,611
E_7	10,208	$7.208 \cdot 10^{0}$	100,996
A_{10}	35,696	$4.633 \cdot 10^{1}$	697,613
A_{11}	140,152	$2.329 \cdot 10^{2}$	3,172,316
E_8	199,952	$3.206 \cdot 10^{2}$	2,399,476

Table B.2.: Benchmark results for TWOA2

W	$\lvert Wk(W, \mathrm{id})\rvert$	Time in seconds	Element comparisons
A_1	2	$5.419 \cdot 10^{-5}$	1
A_2	4	$1.921 \cdot 10^{-4}$	3
BC_2	6	$4.286 \cdot 10^{-4}$	5
A_3	10	$1.122 \cdot 10^{-3}$	11
BC_3	20	$2.110 \cdot 10^{-3}$	22
A_4	26	$3.984 \cdot 10^{-3}$	40
H_3	32	$2.950 \cdot 10^{-3}$	37
D_4	44	$6.877 \cdot 10^{-3}$	62
A_5	76	$1.818 \cdot 10^{-2}$	164
BC_4	76	$1.527 \cdot 10^{-2}$	116
F_4	140	$3.175 \cdot 10^{-2}$	219
D_5	156	$2.811 \cdot 10^{-2}$	307
A_6	232	$6.456 \cdot 10^{-2}$	691
BC_5	312	$6.118 \cdot 10^{-2}$	655
H_4	572	$1.044 \cdot 10^{-1}$	916
D_6	752	$2.072 \cdot 10^{-1}$	1,989
A_7	764	$3.413 \cdot 10^{-1}$	3,048
E_6	892	$2.296 \cdot 10^{-1}$	2,347
BC_6	1,384	$3.827 \cdot 10^{-1}$	3,942
A_8	2,620	$1.532 \cdot 10^{0}$	13,635
A_9	9,496	$7.580 \cdot 10^{0}$	62,630
E_7	10,208	$3.881 \cdot 10^{0}$	33,468
A_{10}	35,696	$2.999 \cdot 10^{1}$	291,699
A_{11}	140,152	$1.530 \cdot 10^{2}$	1,388,533
E_8	199,952	$1.501 \cdot 10^{2}$	844,805

Table B.3.: Benchmark results for TWOA3

Bibliography

[BC] Francis Buekenhout and Arjeh M. Cohen, *Diagram geometry - related to classical groups and buildings*, Draft from `http://www.win.tue.nl/~amc/buek/book1n2.pdf`. 46, 47

[BHH06] Kathryn Brenneman, Ruth Haas, and Aloysius G. Helminck, *Implementing an algorithm for the twisted involution poset for Weyl groups*, `http://www4.ncsu.edu/~loek/research/BHH06.pdf`, 2006. 24, 26, 31

[Cas01] Bill Casselman, *Computation in Coxeter Groups I: Multiplication*, `http://www.math.ubc.ca/~cass/research/pdf/cm.pdf`, 2001. 24

[Cas08] ———, *Computation in Coxeter Groups II: Constructing minimal roots*, An Electronic Journal of the American Mathematical Society Volume 12 (2008), 260–293. 24

[Che69] M. Chein, *Recherche des graphes des matrices de Coxeter hyperboliques d'ordre* ≤ 10, Rev. Francaise Informat. Recherche Opérationnelle 3, Sér. R-3 (1969), 3–16. 7

[Den09] Tom Denton, *Lifting property and poset structure of finite Coxeter groups*, `http://www.math.ucdavis.edu/~anne/WQ2009/MAT280-Lecture9.pdf`, 2009. 10

[Deo77] Vinay V. Deodhar, *Some characterizations of Bruhat ordering on a Coxeter group and determination of the relative Möbius function*, Invent. Math. 39 (1977), 187–198, MR0435249. 10

[Fra01] William N. Franzsen, *Automorphisms of Coxeter groups*, Ph.D. thesis, School of Mathematics and Statistics, University of Sydney, 2001. 41

[GAP] The GAP Group, *GAP – Groups, Algorithms, and Programming, Version 4.5.5*, 2012. 32

[GHM11] Ralf Gramlich, Max Horn, and Bernhard Mühlherr, *Abstract involutions of algebraic groups and of Kac-Moody groups*, Journal of Group Theory, vol. 14 (2) (2011), 213–249. 50

[HH12] Ruth Haas and Aloysius G. Helminck, *Algorithms for twisted involutions in Weyl groups*, Algebra Colloquium 19 (2012), 263–272. 24, 26, 31

[Hor09] Max Horn, *Involutions of Kac-Moody Groups*, Ph.D. thesis, Technische Universität Darmstadt, 2009. 45, 48, 49, 51

[Hul05] Axel Hultman, *Fixed points of involutive automorphisms of the Bruhat order*, Adv. Math. 195 (2005), 283–296, MR2145798. 14

[Hul07] ______, *The combinatorics of twisted involutions in Coxeter groups*, Transactions of the American Mathematical Society, Volume 359 (2007), 2787–2798, MR2286056. 12, 14, 15, 18

[Hum92] James E. Humphreys, *Reflection groups and Coxeter groups*, Cambridge University Press, 1992. 2, 3, 4, 5, 6, 7, 9, 41

[IO] Max Neunhoeffer, *IO – a GAP package, Version 4.2*, 2012. 32

Index