Crystal monoids & crystal bases: rewriting systems and biautomatic structures for plactic monoids of types A_n, B_n, C_n, D_n, and G_2

Alan J. Cain[1,3,4,*]

Centro de Matemática e Aplicações, Faculdade de Ciências e Tecnologia Universidade Nova de Lisboa, 2829–516 Caparica, Portugal

Robert Gray[2,3,4]

School of Mathematics, University of East Anglia, Norwich NR4 7TJ, United Kingdom

António Malheiro[3,4]

Departamento de Matemática, Faculdade de Ciências e Tecnologia Universidade Nova de Lisboa
and
Centro de Matemática e Aplicações, Faculdade de Ciências e Tecnologia Universidade Nova de Lisboa, 2829–516 Caparica, Portugal

Abstract

*Corresponding author

Email addresses: a.cain@fct.unl.pt (Alan J. Cain), Robert.D.Gray@uea.ac.uk (Robert Gray), ajm@fct.unl.pt (António Malheiro)

URL: www.fc.up.pt/pessoas/ajcain/ (Alan J. Cain)

[1]The first author was supported by an Investigador FCT fellowship (IF/01622/2013/CP1161/CT0001).

[2]The second author was partially supported by the EPSRC grant EP/N033353/1 'Special inverse monoids: subgroups, structure, geometry, rewriting systems and the word problem'.

[3]The first and third authors were partially supported by by the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the project UID/MAT/00297/2013 (Centro de Matemática e Aplicações) and the project PTDC/MHC-FIL/2583/2014.

[4]Much of the research leading to this paper was undertaken during visits by the second author to the Centro de Álgebra da Universidade de Lisboa and the Centro de Matemática e Aplicações, Universidade Nova de Lisboa. We thank both centres and universities for their hospitality. These visits were funded by the FCT project PEst-OE/MAT/UI0143/2014 (held by CAUL) and the FCT exploratory project IF/01622/2013/CP1161/CT0001 (attached to the first author's fellowship).

The authors thank Cédric Lecouvey for supplying offprints, Duarte Chambel Ribeiro for pointing out an error, and Vanda Martins for dealing with administrative matters arising from the second author's visits to Lisbon.

The authors also thank two anonymous reviewers for carefully reading the paper and making many valuable suggestions.

The vertices of any (combinatorial) Kashiwara crystal graph carry a natural monoid structure given by identifying words labelling vertices that appear in the same position of isomorphic components of the crystal. Working on a purely combinatorial and monoid-theoretical level, we prove some foundational results for these crystal monoids, including the observation that they have decidable word problem when their weight monoid is a finite rank free abelian group. The problem of constructing finite complete rewriting systems, and biautomatic structures, for crystal monoids is then investigated. In the case of Kashiwara crystals of types A_n, B_n, C_n, D_n, and G_2 (corresponding to the q-analogues of the Lie algebras of these types) these monoids are precisely the generalised plactic monoids investigated in work of Lecouvey. We construct presentations via finite complete rewriting systems for all of these types using a unified proof strategy that depends on Kashiwara's crystal bases and analogies of Young tableaux, and on Lecouvey's presentations for these monoids. As corollaries, we deduce that plactic monoids of these types have finite derivation type and satisfy the homological finiteness properties left and right FP_∞. These rewriting systems are then applied to show that plactic monoids of these types are biautomatic and thus have word problem soluble in quadratic time.

Keywords: crystal basis, plactic monoid, tableaux, rewriting system, automatic monoid
2010 MSC: 17B10, 05E10, 16S15, 16T30, 20M42, 20M05, 20M35, 68Q42, 68Q45, 68R15

---

1. Introduction

The Plactic monoid is a fundamental algebraic object which captures a natural monoid structure carried by the set of semistandard Young tableaux. It arose originally in the work of Schensted [1] on algorithms for finding the maximal length of a nondecreasing subsequence of a given word over the ordered alphabet $\mathcal{A}_n = \{1 < 2 < \ldots < n\}$. The output of Schensted's algorithm is a tableau and, by identifying pairs of words that lead to the same tableau, one obtains the Plactic monoid $\mathrm{Pl}(A_n)$ of rank n. Following this, Knuth [2] found a finite set of defining relations for the Plactic monoid. An in-depth systematic study of the Plactic monoid was then carried out in the work of Schützenberger [3] and Lascoux and Schützenberger [4]. Since then, the Plactic monoid and its corresponding semigroup algebra, the Plactic algebra, have found applications in various aspects of representation theory and algebraic combinatorics. Schützenberger [5] argues that the Plactic monoid ought to be considered as one of the fundamental monoids in algebra. He gives several reasons to support this claim, including the fact that the Plactic monoid was used to give the first correct proofs of the Littlewood–Richardson rule for products of Schur functions by Schützenberger himself [3] and independently by Thomas [6, 7]. (For further details on the Littlewood–Richardson rule and the history of attempts to prove it, see [8, Section 5.4], [9, Appendix], [10, § 4], and [11, Chapter 7, Appendix 1].)

Numerous other applications of the Plactic monoid have since been discovered including a combinatorial description of Kostka–Foulkes polynomials [4, 12], a noncommutative version of the Demazure character formula, and of the Schubert polynomials [13, 14]. The Plactic monoid has motivated a wide range of other interesting work including the discovery of variations on this monoid like the shifted [15] and hypoplactic monoids [16], Littelmann's generalization to Plactic algebras for semisimple Lie algebras [17], the investigation of the Chinese monoid [18], Hilbert series (growth functions) [19], the conjugacy problem [20], homogeneous monoids and algebras which include monoids attached to set-theoretic solutions to Yang–Baxter equations [21, 22, 23, 24], semigroup identities [25], and the theory of quadratic normalization [26]. Some structural results for Plactic algebras were obtained in [27, 28]. An excellent general introduction to the Plactic monoid is given in the article of Lascoux, Leclerc and Thibon [8, Chapter 5].

One of the most exciting connections which has recently emerged are the links between the Plactic monoid and Kashiwara's crystal basis theory. This subject has its origins in the theory of quantum groups [29]. The notion of the quantised enveloping algebra, or quantum group, $U_q(\mathfrak{g})$ associated with a symmetrisable Kac–Moody Lie algebra $\mathfrak{g}$ was discovered independently by Drinfeld [30] and Jimbo [31] in 1985 while studying solutions of the quantum Yang–Baxter equations. Kashiwara [32, 33] introduced crystals in order to give a combinatorial description of modules over $U_q(\mathfrak{g})$ when q tends to zero. Crystals are extremely useful combinatorial tools for studying representations of these algebras. For example, knowing the crystal of a representation allows one to deduce tensor product and branching rules involving that representation. Since its introduction this important theory has been developed and generalised in multiple directions e.g. to quantum affine algebras, superalgebras and quantum queer superalgebras; see [34, 35, 36, 37].

The connection with the Plactic monoid comes via the study of crystal bases of $U_q(\mathfrak{gl}_n)$-modules. These type-A_n crystals have vertex set corresponding to all words over the alphabet $\mathcal{A}_n = \{1 < 2 < \ldots < n\}$, directed edges labelled by colours from the set $I = \{1, 2, \ldots, n-1\}$ which are determined by the Kashiwara operators $\tilde{e}_i$ and $\tilde{f}_i$, and weights coming from the free abelian group $\mathbb{Z}^n$ given by word content (see Section 2.2 for full details of this construction). An isomorphism between two connected components of the crystal is a weight preserving bijection which maps edges to edges preserving colours. If one defines a relation by saying that two words are equivalent if there is an isomorphism between their respective connected components mapping one vertex to the other then it turns out that this relation on $\mathcal{A}_n^*$ is equal to the Plactic relation mentioned above. In this way, the Plactic monoid $\mathrm{Pl}(A_n)$ may be defined in terms of crystals of type A_n. There are a number of explicit constructions known for crystals of representations of other quantum algebras. In addition to type A_n, explicit descriptions of crystals are known for simple Lie algebras of types B_n, C_n, D_n, and the exceptional type G_2; see [29, 38, 39, 40, 41]. For crystals of each of these types, aspects of theory have been developed. As part of their description of crystals of types A_n, B_n, C_n, and D_n, Kashiwara and Nakashima [39] develop the correct generalisation of semistandard tableaux for classical types via the notion of admissible column. For all of these types, Lecouvey obtained finite presentations via Knuth-type relations for the corresponding crystal monoids (as defined in Section 2.4 below), he also gives Schensted–type insertion algorithms and establishes a Robinson–Schensted type correspondence in all of these cases [42, 43, 41]. Bumping and sliding algorithms for C_n-tableaux were also independently obtained by Baker [44]. Analogous results for infinite rank quantum groups were given by Lecouvey in [45].

In addition to shedding new light on the connection between the Plactic monoid and the representation theory of Lie algebras, this viewpoint also gives rise to a natural family of monoids arising from crystals, generalising the classical Plactic monoid. Following Kashiwara [46] a crystal is an edge-coloured directed graph satisfying a certain simple set of axioms. As we shall see in Section 2.4 below, every abstract combinatorial crystal gives rise to a monoid, in the same way that the classical Plactic monoid arises from A_n above. Examples of Crystal monoids (with weights from a free abelian group) include the classical Plactic monoid $Pl(A_n)$, each of the Plactic-type monoids studied by Lecouvey in [42, 43, 41], and also other important well-studied monoids such as the bicyclic monoid.

In more detail, as mentioned above, in the general abstract definition of combinatorial crystal (see Section 2.2 below for a full definition) the vertices correspond to words over a finite alphabet X, and weight-preserving isomorphisms between connected components define a congruence $\sim$ on the free monoid X^*. The corresponding crystal monoid is then the monoid $X^*/\sim$ obtained by factoring the free monoid by this congruence. This connects the theory of Kashiwara crystals directly to combinatorial semigroup theory (the study of semigroups defined by generators and relations), combinatorics on words, and formal language theory. For instance, Lecouvey's results [42, 43, 41] show in particular that for all classical types, these crystal monoids $X^*/\sim$ are all finitely presented. Powerful tools exist for studying monoids defined by presentations in this way, including the theories of (Noetherian and confluent) string rewriting systems [47] and automata theory, specifically the theory of automatic groups and monoids [48, 49].

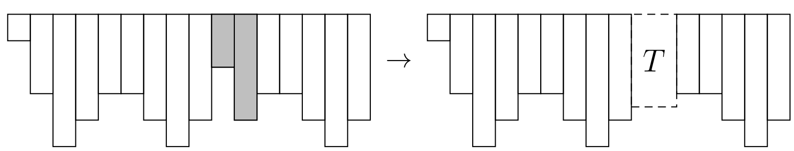
The defining property for automatic groups and monoids is the existence of a rational set of normal forms (with respect to some finite generating set A) such that we have, for each generator in A, a finite automaton that recognizes pairs of normal forms that differ by multiplication by that generator. It is a consequence of the definition that automatic monoids (and in particular automatic groups) have word problem that is soluble in quadratic time [49, Corollary 3.7]. Automatic groups have attracted a lot of attention over the last 25 years, in part because of the large number of natural and important classes of groups that have this property. The class of automatic groups includes: various small cancellation groups [50], Artin groups of finite and large type [51], braid groups, and hyperbolic groups in the sense of Gromov [52]. In parallel, the theory of automatic monoids has been extended and developed over recent years. Classes of monoids that have been shown to be automatic include divisibility monoids

[53], singular Artin monoids of finite type [54], and monoids arising from confluence monadic rewriting systems [55, 56]. Several complexity and decidability results for automatic monoids are obtained in [57]. Other aspects of the theory of automatic monoids that have been investigated include connections with the theory of Dehn functions [58] and complete rewriting systems [59].

In the cases that they are applicable, these tools of string rewriting systems and automatic structures give rise to algorithms for working with the monoids, which can in particular be used to study decidability and complexity questions. These are very natural aspects of theory to develop given the fundamental role that algorithms play in the theory of Plactic monoids, tableaux and Kashiwara crystals outlined above. Of course any results about the complexity of algorithms for working with these monoids (algorithms that operate on words) may be translated to results about algorithms for working with the corresponding tableaux and crystal graphs (see Section 7 for examples of this). It was precisely these kinds of ideas that motivated the current authors' paper [60] on the classical Plactic monoid. It was pointed out by E. Zelmanov [during his plenary lecture at the international conference *Groups and Semigroups: Interactions and Computations* (Lisbon, 25–29 July 2011)] that since Schensted's algorithm can be used to show that the Plactic monoid has word problem that is soluble in quadratic time, it is natural to ask whether Plactic monoids are automatic. This is a natural question since (as mentioned above) all automatic monoids have word problem decidable in quadratic time. In [60] we gave an affirmative answer to this question. We did this by first constructing a finite complete rewriting system for the Plactic monoid, with respect to the set of column generators. Beginning with this finite complete rewriting system, we then showed that for Plactic monoids, finite transducers may be constructed to perform left (respectively right) multiplication by a generator. We then applied this result to show that Plactic monoids of arbitrary finite rank are biautomatic (the strongest form of automaticity for monoids). Other consequences of these results include the fact that Plactic algebras of finite rank admit finite Gröbner–Shirshov bases, Plactic monoids of finite rank satisfy the homological finiteness property FP_∞, and the homological finiteness property FDT, and that Plactic algebras are automaton algebras in the sense of Ufnarovski; see [61] or more recently [21].

From the point of view of crystals, these results say that string rewriting systems and transducers can be used to compute efficiently with crystals of type A_n. Our interest in this paper is to investigate the extent to which these tools can be applied to other Kashiwara crystals and crystal monoids. The results in this article will show that such tools can be successfully developed for all of the classical types A_n, B_n, C_n, D_n, and for the exceptional type G_2. As in the case of the classical Plactic monoid the existence of finite complete rewriting systems implies that these monoids have finite derivation type and satisfy the homological finiteness properties left and right FP_∞, and that the corresponding semigroup algebras are automaton algebras and all admit finite Gröbner–Shirshov bases. Also, the existence of biautomatic structures implies these monoids all have word problem soluble in quadratic time.

We now give a brief overview of the main ideas, constructions and results that shall be obtained in this paper. We begin by using some results from the theory of crystal bases to construct finite complete rewriting systems presenting the Plactic monoids of types A_n, B_n, C_n, D_n, and G_2. (We refer the reader forward to Subsection 5.1 for definitions and terminology on rewriting systems.) We use column generators and our rewriting system has rules that replace an adjacent pair of columns by the unique tableau that represents their product. The set of Young tableaux serves as a cross-section of the plactic monoid of type A_n: two words in $\mathcal{A}_n^*$ represent the same element of $\mathrm{Pl}(A_n)$ if and only if they give the same tableau when Schensted's insertion algorithm (see [1] and [8, Ch. 5]) is applied to them. The other types of plactic monoids have analogous (but substantially different) types of tableaux. Any of these tableaux, when read column-by-column from right to left, yields a word that represents the corresponding element of the monoid. Thus the columns of a given type are generators for the plactic monoid of that type. Most products of columns are not tableaux. Following [43], we call an arbitrary product of columns a tabloid. The key to constructing our rewriting systems and automatic structures is to use column generators and rewrite tabloids to tableaux. More formally, we consider a pair of columns that form a tabloid that is *not* a tableau. This is the left-hand side of a rewriting rule. The right-hand side of the corresponding rewriting rule is the unique tableau that represents the same element of the monoid as this tabloid. Pictorially, rewriting will look like the following:



where T is the tableau representing the same element as the two shaded columns. Thus we gradually rewrite a tabloid towards a product of columns where every adjacent pair of columns forms a tableau; as we shall see, the whole product then forms a tableau. We prove that this rewriting is terminating by anayzing what shapes of tableaux can result from a product of two columns. For the classical types A_n, B_n, C_n and D_n this is done by applying the generalized Littlewood–Richardson rule for decomposing tensor products of crystals into connected components (see [29, Theorem 8.6.6.]). The case of G_2 is dealt with separately using an analysis of products of columns, working with highest-weight words.

Equipped with our finite complete rewriting systems, we then proceed to prove that the Plactic monoids of types A_n, B_n, C_n, D_n, and G_2 are biautomatic. (We refer the reader forward to Subsection 7 for definitions and terminology on automatic semigroups.) In each case the language of representatives of the biautomatic structure will be the language of irreducible words of the rewriting complete system (Σ, T) described above. To obtain a biautomatic structure, we first investigate what happens when we take a tableau and left multiply by a single generator. We show how the corresponding word over Σ can be rewritten by T to an irreducible word by a single left-to-right pass through the word, and that this only changes the length of the word by at most 1, in all the classical cases A_n, B_n, C_n and D_n, and by at most 2 in the case of G_2. Analogous results are proved for right multiplication by a single generator, although the proofs are more involved than the corresponding results for left multiplication. These results are then used to build biautomatic structures for the plactic monoids of each type. The strategy is to show that the same kind of rewriting occurs when a normal form word, not necessarily of highest weight, is left- or right-multiplied by a generator, and thus that such rewriting can be carried out by a two-tape automaton.

Note that we recover in this paper a new proof of our previous results that classical Plactic monoids (of type A_n) can be presented by finite complete rewriting systems and are biautomatic [60]. While writing this paper, we came across the work of Hage [62], who independently constructed a finite complete rewriting system for $\mathrm{Pl}(C_n)$. Hage's approach differs from ours in making use of Lecouvey's insertion algorithms, whereas we use Lecouvey's presentations. (Hage does not consider biautomaticity or its consequences.) We should also note that an alternative approach to obtaining complete rewriting systems for the Plactic monoids considered in this paper is to apply the results of Littelmann [17, Theorem B, § 8] which he obtained using his path model. In contrast, as far as the authors are aware, the results we obtain here are the first to appear in the literature on biautomatic structures and complexity of the word problem for plactic monoids of types A_n, B_n, C_n, D_n, and G_2. It is important to note that there exist finitely presented monoids which are defined by finite complete rewriting systems but which are not automatic. Indeed, there even exist multihomogeneous finitely presented monoids with this property; see [63]. Thus our results on biautomaticity are in no sense immediate consequences of the existence of complete rewriting systems defining these monoids. Indeed, in order to obtain our results on automatic structures, and the corollaries on the complexity of the word problem, we shall need to prove results which give detailed information about how products of columns are rewritten to normal form using the finite complete rewriting systems.

2. Crystals and plactic monoids

In this section we will formulate the main concepts that are used throughout the paper. We will present Kashiwara's characterization of plactic monoids in terms of crystal graphs. We first outline a pure combinatorial abstract theory of crystal monoids, that avoids delving into the deep theory underlying crystal graphs, thus providing a general framework for all the different types of plactic monoids (A_n B_n, C_n, D_n and G_2). This general theory gives us an abstract version of known results from [39, 41, 46] for the different types of plactic monoids. For the underlying theory of crystal bases we refer the reader to [29].

2.1. Notation

We denote the empty word (over any alphabet) by ε. For an alphabet X, we denote by X^* the set of all words over X including the empty word ε. When X is a generating set for a monoid M, every element of X^* can be interpreted either as a word or as an element of M. For words $u, v \in X^*$, we write $u = v$ to indicate that u and v are equal as words and $u =_X v$ to denote that u and v represent the same element of the monoid M. The length of $u \in X^*$ is denoted $|u|$, and, for any $x \in X$, the number of occurences of the symbol x in u is denoted $|u|_x$.

2.2. Definition of crystal graph

For the purposes of this paper, a directed graph with labels from I is a set V of vertices equipped with a set E of triples drawn from $V \times I \times V$. A triple $(v, i, v') \in E$ is interpreted as an edge from the vertex v to a vertex v' with label i. A path starting at $u \in V$ and ending at $w \in V$ is a (possibly empty) sequence of edges (u, i_0, v_1), (v_1, i_1, v_2), $\ldots$, (v_n, i_n, w); note that all paths are directed. Notice that vertices and edges may appear multiple times on a path.

Definition 2.1. A *crystal basis* is a directed labelled graph with vertex set X and label set I satisfying the conditions:

- For all $x \in X$ and $i \in I$, there is at most one edge starting at x labelled by i and at most one edge ending at x labelled by i.
- For all $i \in I$, there is no infinite path made up of edges labelled by i.

Notice that the second condition implies that a crystal basis cannot contain an i-labelled directed circuit.

(Strictly speaking, such a graph is a graphical description of the representation-theoretic notion of a crystal basis; see [29, § 4.2] for details. More precisely, every (integrable highest weight) representation of a symmetrizable Kac–Moody algebra has a crystal associated to it. However, not every crystal arises from such a representation. Indeed, there has been research on finding a simple set of local axioms that characterize those crystals that arise from such representations; see [64, 65, 66]. In fact, the two conditions above coincide with axioms (P1) and (P2) in the characterization of the crystal graphs of integrable highest-weight modules for simply-laced quantum Kac–Moody algebras in [64].)

For each $i \in I$, define partial maps $\tilde{e}_i$ and $\tilde{f}_i$ called the *Kashiwara operators* on the set X as follows: for each edge (a, i, b), which we will represent graphically as

$$a \xrightarrow{\ i\ } b\ ,$$

define $\tilde{f}_i(a) = b$ and $\tilde{e}_i(b) = a$.

Using the definition of $\tilde{e}_i$ and $\tilde{f}_i$, we can build an extended directed labelled graph:

Definition 2.2. A *crystal graph* arising from a given crystal basis with vertex set X and label set I, is a directed labelled graph, denoted Γ_X, with vertex set X^*, the free monoid on X. The edges are defined by partially extending the operators $\tilde{e}_i$ and $\tilde{f}_i$ to X^*, as follows: for all $u, v \in X^*$ and $i \in I$, define inductively

$$\tilde{e}_i(uv) = \begin{cases} u\,\tilde{e}_i(v) & \text{if } \varphi_i(u) < \epsilon_i(v) \\ \tilde{e}_i(u)\,v & \text{if } \varphi_i(u) \geq \epsilon_i(v) \end{cases}; \tag{2.1}$$

$$\tilde{f}_i(uv) = \begin{cases} \tilde{f}_i(u)\,v & \text{if } \varphi_i(u) > \epsilon_i(v) \\ u\,\tilde{f}_i(v) & \text{if } \varphi_i(u) \leq \epsilon_i(v) \end{cases}, \tag{2.2}$$

where ϵ_i and φ_i are auxiliary maps on X^* defined as follows: for $w \in X^*$, let

$$\epsilon_i(w) = \max\big\{\, k \in \mathbb{N} \cup \{0\} : \underbrace{\tilde{e}_i \cdots \tilde{e}_i}_{k \text{ times}}(w) \text{ is defined} \,\big\};$$

$$\varphi_i(w) = \max\big\{\, k \in \mathbb{N} \cup \{0\} : \underbrace{\tilde{f}_i \cdots \tilde{f}_i}_{k \text{ times}}(w) \text{ is defined} \,\big\}.$$

This extension of the operators $\tilde{e}_i$ and $\tilde{f}_i$ to words on X^* replicates the properties of the action of the Kashiwara operators as in [39, Theorem 1.1.4].

For each $i \in I$, define a map $\rho_i : X^* \to \{\, -^p +^q : p, q \in \mathbb{N} \cup \{0\} \,\}$. (Note that the symbols $+$ and $-$ here, and in the following discussion, are simply letters in the alphabet $\{+, -\}$.) For a word $w \in X^*$, define $\rho_i(w)$ to be the word obtained by replacing each symbol x of w by $-^{\epsilon_i(x)}+^{\varphi_i(x)}$, then iteratively deleting subwords $+-$ until a word of the form $-^p +^q$ remains. Note further that each symbol $+$ or $-$ in the computed word $\rho_i(w)$ is a symbol that 'survives' from the original replacement of symbols x by $-^{\epsilon_i(x)}+^{\varphi_i(x)}$. Furthermore, each symbol $+$ or $-$ in $\rho_i(w)$ is contributed by a uniquely determined symbol of w (since two subwords $+-$ cannot partially overlap with each other).

The following result shows the connection between ρ_i and the action of the operators $\tilde{e}_i$ and $\tilde{f}_i$.

For classical Lie algebras the properties on operators given in the following result may be found in [39]. The generalisation below is proved in a similar way, directly from the definitions above, so we omit the proof.

Proposition 2.3. *Let $w = w_1 \cdots w_k$, where $w_h \in X$, and $i \in I$. Then*

1. (a) *$\tilde{e}_i(w)$ is defined if and only if $\rho_i(w)$ contains at least one symbol $-$.*
   (b) *If $\tilde{e}_i(w)$ is defined, $\tilde{e}_i(w) = w_1 \cdots w_{j-1} \tilde{e}_i(w_j) w_{j+1} \cdots w_k$, where w_j is the symbol that contributed the rightmost symbol $-$ in $\rho_i(w)$.*
   (c) *If $\tilde{e}_i(w)$ is defined, $w = \tilde{f}_i(\tilde{e}_i(w))$.*
2. (a) *$\tilde{f}_i(w)$ is defined if and only if $\rho_i(w)$ contains at least one symbol $+$.*
   (b) *If $\tilde{f}_i(w)$ is defined, $\tilde{f}_i(w) = w_1 \cdots w_{j-1} \tilde{f}_i(w_j) w_{j+1} \cdots w_k$, where w_j is the symbol that contributed the leftmost symbol $+$ in $\rho_i(w)$.*
   (c) *If $\tilde{f}_i(w)$ is defined, $w = \tilde{e}_i(\tilde{f}_i(w))$.*
3. *$\rho_i(w) = -^{\epsilon_i(w)}+^{\varphi_i(w)}$.*

Furthermore, the actions of the operators $\tilde{e}_i$ and $\tilde{f}_i$ are well-defined.

The previous proposition gives the following practical method, first described in [39], for computing the actions of $\tilde{e}_i$ and $\tilde{f}_i$ on a word $w \in X^*$: Compute $\rho_i(w)$ by writing down the word obtained by replacing each symbol x by $-^{\epsilon_i(x)}+^{\varphi_i(x)}$ and then deleting subwords $+-$. The resulting word will have the form $-^{\epsilon_i(w)}+^{\varphi_i(w)}$. If $\epsilon_i(w) = 0$, then $\tilde{e}_i(w)$ is undefined. If $\epsilon_i(w) > 0$ then we obtain $\tilde{e}_i(w)$ by taking the symbol x that contributed the rightmost $-$ of $\rho_i(w)$ and changing it to $\tilde{e}_i(x)$. If $\varphi_i(w) = 0$, then $\tilde{f}_i(w)$ is undefined. If $\varphi_i(w) > 0$ the we obtain $f_i(w)$ by taking the symbol x that contributed the leftmost $+$ of $\rho_i(w)$ and changing it to $\tilde{f}_i(x)$.

Notice in particular that if, during the deletion of subwords $+-$, the word that we obtain begins with $-$, then this symbol $-$ will remain in place throughout all subsequent deletions, and so $\epsilon_i(w) > 0$, and so $\tilde{e}_i(w)$ is defined. This observation is important, and we will use it repeatedly throughout the paper. (There is a dual observation for words ending in $+$ implying that $\tilde{f}_i(w)$ is defined, but we will not need this.)

In the crystal graph, we have an edge from w to w' labelled by i if and only if $w' = \tilde{f}_i(w)$ (or, equivalently, $w = \tilde{e}_i(w')$). Note that $\epsilon_i(u)$ is the length of the longest path consisting of edges labelled by i that ends at u. Dually, $\varphi_i(u)$ is the length of the longest path consisting of edges labelled by i that starts at u.

2.3. Weights

In our abstract combinatorial setting we have the following definition:

Definition 2.4. A *weight function* is a homomorphism $\mathrm{wt} : X^* \to P$, where P is some monoid (called the *weight monoid*) such that there is a partial order $\leq$ on P (not necessarily compatible with multiplication in P) with the following property: for all $u \in X^*$ and $i \in I$,

- if $\tilde{e}_i(u)$ is defined, then $\mathrm{wt}(u) < \mathrm{wt}\big(\tilde{e}_i(u)\big)$; and
- if $\tilde{f}_i(u)$ is defined, then $\mathrm{wt}\big(\tilde{f}_i(u)\big) < u$.

Let $u, v \in X^*$. The word u has *higher weight* than the word v (or, equivalently, the word v has *lower weight* than the word u) if $\mathrm{wt}(v) < \mathrm{wt}(u)$. Thus the operators $\tilde{e}_i$, when defined, always yield a word of higher weight, and the operators $\tilde{f}_i$, when defined, always yield a word of lower weight.

The abstract definitions of weight monoid and weight functions given here are more general than in the literature. In the context of Lie algebras representations the weight is a linear map from the vertices of the crystal components (which is identified with the set of words from X^*) to the weight lattice generated by the fundamental weights $\Lambda_1, \ldots, \Lambda_n$. This weight lattice can be identified (up to isomorphism) with $\mathbb{Z}^n$. For the root system of type A_n, the partial order on $\mathbb{Z}^n$ is the so-called *dominance order* on the set of partitions (see [67, § I.1]).

In the remainder of the paper, we will not need to explicitly compare orders: we simply use the fact that $\tilde{e}_i$, when defined, raise weight, and $\tilde{f}_i$, when defined, lowers weight.

In the crystal graph Γ_X, a vertex that has maximal weight within a particular component is called a *highest-weight vertex*. (In the specific crystal graphs we consider later, it will turn out that each component contains a *unique* highest-weight vertex.)

Lemma 2.5 ([41, Lemma 5.3.1]). *For any words $w_1, w_2 \in X^*$, the word $w_1 w_2$ is a vertex of highest weight of a connected component of the crystal graph Γ_X if and only if:*

1. *w_1 is a vertex of highest weight (that is, $\epsilon_i(w_1) = 0$);*
2. *for all $i = 1, \ldots, n$ we have $\epsilon_i(w_2) \leq \varphi_i(w_1)$.*

2.4. Relations from crystal graphs

For any word $w \in X^*$, let $B(w)$ be the connected component of the crystal graph containing the vertex w. A *crystal isomorphism* is a bijection φ between two connected components $B(w)$ and $B(w')$ that maps directed edges labelled by i to directed edges labelled by i (in the sense that if (x, i, y) is an edge in $B(w)$, then $(\varphi(x), i, \varphi(y))$ is an edge in $B(w')$), sends non-edges to non-edges, and preserves weights (in the sense that $\mathrm{wt}(u) = \mathrm{wt}(\varphi(u))$ for any $u \in B(w)$). If there is a crystal isomorphism between $B(w)$ and $B(w')$, we say that $B(w)$ and $B(w')$ are isomorphic.

We say $u \in B(w)$ and $v \in B(w')$ lie in the *same position* of isomorphic components $B(w)$ and $B(w')$ if there is an isomorphism between $B(w)$ and $B(w')$ that maps u to v; this is denoted by $u \sim v$. This general abstract setting is sufficient to obtain a congruence. For classical crystals this result is well-known (see [41] for a survey).

Proposition 2.6. *The relation $\sim$ is a congruence on the free monoid X^*.*

Definition 2.7. Let X be an alphabet forming the vertex set of a crystal basis, $\mathrm{wt} : X^* \to P$ a weight function, and $\sim$ the congruence on X^* that relates two words if they lie in the same position of isomorphic components of the crystal graph Γ_X. Then we call $X^*/{\sim}$ the *crystal monoid* determined by the crystal Γ_X with weight function wt and weight monoid P.

Note that if multiplication in P is algorithmically computable, then the weights of words in X^* are computable. If the crystal basis is finite (and so the crystal monoid is finitely generated), then it is possible to compute the connected component of any word in X^*. If both these conditions hold, then we can decide whether two components are isomorphic, and thus check whether two words are $\sim$-related. In short, we have the following:

Proposition 2.8. *If a crystal monoid arises from a finite crystal basis, and has a weight monoid in which multiplication is computable, then it has soluble word problem.*

In particular, when the weight monoid P is (isomorphic to) a free abelian group of finite rank (which it will be in all the specific examples we consider below) then Proposition 2.8 applies, and the crystal monoid will have soluble word problem. Notice, however, that this result says nothing about the *complexity* of the word problem. We will see that $\mathrm{Pl}(A_n)$ and the plactic monoids of other types, which we will define shortly, are all biautomatic and thus have word problem soluble in quadratic time [49, Corollary 3.7].

2.5. Crystal graphs of types A_n, B_n, C_n, D_n and G_2

The plactic monoid (of type A_n) parameterizes representations of the q-analogue of the universal enveloping algebra of the semisimple Lie algebras of type A_n. There are analogous plactic monoids of types B_n, C_n, D_n, and G_2, parameterizing representations of the q-analogues of the universal enveloping algebras of the semisimple Lie algebras of the corresponding types.

In our combinatorial abstract framework, a crystal graph is constructed from a crystal basis following the rules given by the action of classical Kashiwara operators. In turn, for all the classical types, this action is given by the simple tensor rule, and thus the crystal graph within this combinatorial abstract setting corresponds to the classical crystal graph arising as tensor powers of the corresponding basis.

If we fix the crystal basis of type A_n to be the irreducible root system of type A_n of the representation of the q-analogue of the classical Lie Algebra of that type, and weight function as in [41, § 3.3], we will obtain the plactic monoid of type A_n. All other plactic monoid types arise as crystal monoids in the same way, but starting from different crystal bases and definitions of the weight function.

For all the classical types, the weight functions arise from the root systems of the corresponding Lie algebras as detailed in [41, § 3.3].

2.5.1. Type A_n

For type A_n we consider the ordered alphabet

$$\mathcal{A}_n = \{1 < 2 < \ldots < n\}.$$

The crystal basis for type A_n is:

$$1 \xrightarrow{1} 2 \xrightarrow{2} \ldots \xrightarrow{n-2} n{-}1 \xrightarrow{n-1} n \tag{2.3}$$

This graph has vertex set $\mathcal{A}_n$ and labels from the set $\{1, \ldots, n{-}1\}$. The resulting graph is the *crystal graph of type* A_n, denoted Γ_{A_n}, and the monoid that arises is the *plactic monoid of type* A_n, denoted $\mathrm{Pl}(A_n)$.

2.5.2. Type B_n

For type B_n we consider the ordered alphabet

$$\mathcal{B}_n = \big\{1 < 2 < \ldots < n < 0 < \overline{n} < \ldots < \overline{2} < \overline{1}\big\}.$$

Note that 0 is greater than n. The crystal basis for type B_n is:

$$1 \xrightarrow{1} 2 \xrightarrow{2} \ldots \xrightarrow{n-1} n \xrightarrow{n} 0 \xrightarrow{n} \overline{n} \xrightarrow{n-1} \ldots \xrightarrow{2} \overline{2} \xrightarrow{1} \overline{1}$$

The resulting graph is the *crystal graph of type* B_n, denoted Γ_{B_n}, and the monoid that arises is the *plactic monoid of type* B_n, denoted $\mathrm{Pl}(B_n)$.

2.5.3. Type C_n

For type C_n we consider the ordered alphabet

$$\mathcal{C}_n = \{1 < 2 < \ldots < n < \overline{n} < \overline{n-1} < \ldots < \overline{1}\}.$$

The crystal basis for type C_n is:

$$1 \xrightarrow{1} 2 \xrightarrow{2} \ldots \xrightarrow{n-1} n \xrightarrow{n} \overline{n} \xrightarrow{n-1} \ldots \xrightarrow{2} \overline{2} \xrightarrow{1} \overline{1}$$

The resulting graph is the *crystal graph of type* C_n, denoted Γ_{C_n}, and the monoid that arises is the *plactic monoid of type* C_n, denoted $\mathrm{Pl}(C_n)$.

2.5.4. Type D_n

For type D_n we consider the ordered alphabet

$$\mathcal{D}_n = \left\{1 < 2 < \ldots < n-1 < \begin{matrix} \overline{n} \\ n \end{matrix} < \overline{n-1} < \ldots < \overline{2} < \overline{1}\right\};$$

note that n and $\overline{n}$ are incomparable and that $n-1 < n < \overline{n-1}$ and $n-1 < \overline{n} < \overline{n-1}$. The crystal basis for type D_n is:

$$\begin{tikzcd}
 & & & & n \arrow[dr, "n"] & & & & \\
1 \arrow[r, "1"] & 2 \arrow[r, "2"] & \ldots \arrow[r, "n-2"] & n-1 \arrow[ur, "n-1"] \arrow[dr, "n"'] & & \overline{n-1} \arrow[r, "n-2"] & \ldots \arrow[r, "2"] & \overline{2} \arrow[r, "1"] & \overline{1} \\
 & & & & \overline{n} \arrow[ur, "n-1"'] & & & &
\end{tikzcd}$$

The resulting graph is the *crystal graph of type* D_n, denoted Γ_{D_n}, and the monoid that arises is the *plactic monoid of type* D_n, denoted $\mathrm{Pl}(D_n)$.

2.5.5. Type G_2

For type G_2 we consider the ordered alphabet

$$\mathcal{G}_2 = \{1 < 2 < 3 < 0 < \overline{3} < \overline{2} < \overline{1}\}.$$

The crystal basis for type G_2 is:

$$1 \xrightarrow{1} 2 \xrightarrow{2} 3 \xrightarrow{1} 0 \xrightarrow{1} \overline{3} \xrightarrow{2} \overline{2} \xrightarrow{1} \overline{1} \tag{2.4}$$

The resulting graph is the *crystal graph of type* G_2, denoted Γ_{G_2}, and the monoid that arises is the *plactic monoid of type* G_2, denoted $\mathrm{Pl}(G_2)$.

2.6. Properties of crystal graphs of types A_n, B_n, C_n, D_n and G_2

Let X be one of the types A_n, B_n, C_n, D_n or G_2, and let $\mathcal{X}$ be the corresponding alphabet $\mathcal{A}_n$, $\mathcal{B}_n$, $\mathcal{C}_n$, $\mathcal{D}_n$ or $\mathcal{G}_2$. As described above, we have a crystal graph Γ_X and a plactic monoid $\mathrm{Pl}(X)$ of each of the given types. For clarity and brevity in explanations, define, for all $x, y \in \mathcal{X}$ with $x \leq y$,

$$\mathcal{X}[x, y] = \{\, z \in \mathcal{X} : x \leq z \leq y \,\}.$$

Recall that the Kashiwara operators $\tilde{e}_i$ and $\tilde{f}_i$ respectively raise and lower weights whenever they are defined.

An important and non-obvious fact for us will be that each connected component of a crystal graph Γ_X contains a *unique* highest-weight vertex [41, § 3.1]. (It is not true for crystal monoids in general that the connected components of the crystal have unique highest-weight vertices.) For any word $w \in \mathcal{X}^*$, denote by w^0 the unique highest-weight vertex in $B(w)$. Thus there exist $i_1, \ldots, i_r \in \{1, \ldots, n\}$ such that $w^0 = \tilde{e}_{i_1} \ldots \tilde{e}_{i_r}(w)$, or, equivalently $w = \tilde{f}_{i_r} \ldots \tilde{f}_{i_1}(w^0)$.

Notice that for Γ_X, we have $u \sim v$ if and only if $u^0 \sim v^0$ and there exist $i_1, \ldots, i_r \in \{1, \ldots, n\}$ such that

$$u = \tilde{f}_{i_r} \cdots \tilde{f}_{i_1}(u^0) \text{ and } v = \tilde{f}_{i_r} \cdots \tilde{f}_{i_1}(v^0).$$

3. Tableaux and tabloids

In this section we give the necessary background on tableaux theory for plactic monoids of types A_n, B_n, C_n, D_n, and G_2, that will be frequently used in the sequel; see [39] and [41] for further details.

3.1. Young tableaux and columns

A Young diagram Y (of shape λ) associated to a partition $\lambda = (\lambda_1, \ldots, \lambda_k)$ is a finite array of left-justified boxes whose i-th row has length λ_i. A *Young tableau* T of shape λ is a filling of a Young diagram by symbols from the fixed alphabet such that (i) the entries of any column strictly increase from top to bottom, and (ii) the entries along each row weakly increase from left to right.

A *column* (of type A_n) is a tableau of column shape $\lambda = (1, \ldots, 1)$:

$$\begin{array}{|c|}\hline x_1 \\ \hline x_2 \\ \hline \vdots \\ \hline x_k \\ \hline \end{array}$$

A *column* of type B_n, C_n and D_n is, respectively, a Young diagram of column shape of the form

$$\begin{array}{|c|}\hline \beta_+ \\ \hline \beta_0 \\ \hline \beta_- \\ \hline\end{array}\,, \qquad \begin{array}{|c|}\hline \gamma_+ \\ \hline \gamma_- \\ \hline\end{array}\,, \qquad \text{and} \qquad \begin{array}{|c|}\hline \delta_+ \\ \hline \delta \\ \hline \delta_- \\ \hline\end{array}\,,$$

where

- β_+ is filled with symbols from $\mathcal{B}_n[1, n]$, and is strictly increasing from top to bottom;
- β_0 is filled with symbols 0;
- β_- is filled with symbols from $\mathcal{B}_n[\overline{n}, \overline{1}]$, and is strictly increasing from top to bottom;
- γ_+ is filled with symbols from $\mathcal{C}_n[1, n]$, and is strictly increasing from top to bottom;
- γ_- is filled with symbols from $\mathcal{C}_n[\overline{n}, \overline{1}]$, and is strictly increasing from top to bottom;
- δ_+ is filled with symbols from $\mathcal{D}_n[1, n-1]$, and is strictly increasing from top to bottom;
- δ is filled with symbols n and $\overline{n}$, with different symbols in vertically adjacent cells.
- δ_- is filled with symbols from $\mathcal{D}_n[\overline{n-1}, \overline{1}]$, and is strictly increasing from top to bottom.

A *column* of type G_2 is a Young tableau with entries from $\mathcal{G}_2$, of column shape, of one of the following three forms:

$$\begin{array}{|c|}\hline a \\ \hline\end{array}\,, \quad \begin{array}{|c|}\hline a \\ \hline b \\ \hline\end{array} \text{ with } a < b, \quad \text{or} \begin{array}{|c|}\hline 0 \\ \hline 0 \\ \hline\end{array}\,.$$

The *height* $h(\beta)$ of a column β (of any type) is the number of boxes in the column. The *reading* $w(\beta)$ of a column is the word obtained by reading the sequence of symbols in the boxes from top to bottom. We identify a column with its reading. A word is a *column word* if it is the reading of a (necessarily unique) column.

3.1.1. Admissible columns

Let β be a column (of any type) and let $z \leq n$. We denote by $N_\beta(z)$ the number of symbols x in β such that $x \leq z$ or $\overline{z} \leq x$.

A column β is *admissible* if each of the following conditions is satisfied:

1. $N_\beta(z) \leq z$, for any $z \leq n$;
2. if β is of type B_n and 0 is in β, then $h(\beta) \leq n$;
3. if $\beta = \begin{array}{|c|}\hline a \\ \hline b \\ \hline \end{array}$ is of type G_2 and height 2, then

$$\begin{cases} \operatorname{dist}(a,b) \leq 2 & \text{for } a \in \{1, 0\}, \\ \operatorname{dist}(a,b) \leq 3 & \text{otherwise,} \end{cases}$$

   where $\operatorname{dist}(a,b)$ is the number of arrows between a and b in the crystal basis (2.4) for G_2.

Note that all columns of type A_n are admissible.

The following is a complete list of all twenty-one admissible columns of type G_2:

$$\left\{ \boxed{1}, \boxed{2}, \boxed{3}, \boxed{0}, \boxed{\overline{3}}, \boxed{\overline{2}}, \boxed{\overline{1}}, \begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline \end{array}, \begin{array}{|c|}\hline 1 \\ \hline 3 \\ \hline \end{array}, \begin{array}{|c|}\hline 2 \\ \hline 3 \\ \hline \end{array}, \begin{array}{|c|}\hline 2 \\ \hline 0 \\ \hline \end{array}, \right.$$
$$\left. \begin{array}{|c|}\hline 2 \\ \hline \overline{3} \\ \hline \end{array}, \begin{array}{|c|}\hline 0 \\ \hline \overline{3} \\ \hline \end{array}, \begin{array}{|c|}\hline 3 \\ \hline \overline{3} \\ \hline \end{array}, \begin{array}{|c|}\hline 3 \\ \hline 0 \\ \hline \end{array}, \begin{array}{|c|}\hline 3 \\ \hline \overline{2} \\ \hline \end{array}, \begin{array}{|c|}\hline 0 \\ \hline \overline{2} \\ \hline \end{array}, \begin{array}{|c|}\hline \overline{3} \\ \hline \overline{2} \\ \hline \end{array}, \begin{array}{|c|}\hline \overline{3} \\ \hline \overline{1} \\ \hline \end{array}, \begin{array}{|c|}\hline \overline{2} \\ \hline \overline{1} \\ \hline \end{array}, \begin{array}{|c|}\hline 0 \\ \hline 0 \\ \hline \end{array} \right\}. \tag{3.1}$$

An *admissible column word* is a word that is the reading of a (necessarily unique) admissible column.

3.1.2. The functions ℓ and r

We say that a column β contains a pair $(z, \overline{z})$ if both symbols z and $\overline{z}$ appear in β, or if β is of type B_n and 0 appears in β. In the following paragraphs we define partial functions ℓ and r on the set of columns of some type. The resulting columns $\ell(\beta)$ and $r(\beta)$, when defined, do not contain pairs $(z, \overline{z})$. For simplicity and uniformity, for columns of type A_n we define $r(\beta) = \ell(\beta) = \beta$.

Let β be a column of type B_n or C_n and let $I_\beta = \{z_s < \ldots < z_{r+1} < z_r = 0, \ldots, z_1 = 0\}$ be the set of symbols z for which β contains the pair $(z, \overline{z})$. We say that a column β of type B_n or C_n can be *split* if there exists a set J_β of symbols $t_s < \cdots < t_1$ such that

- t_1 is maximal such that $t_1 < z_1$ and the symbols t_1 and $\overline{t_1}$ do not appear in β;
- for $i = 2, \ldots, s$, the symbol t_i is maximal such that $t_i < \min\{t_{i-1}, z_i\}$, $t_i \notin \beta$, and $\overline{t_i} \notin \beta$.

If β can be split, $r(\beta)$ is obtained from β by replacing $\overline{z_i}$ with $\overline{t_i}$ for each i, and $\ell(\beta)$ is obtained from β by replacing z_i with t_i for each i, always reordering to obtain a column if necessary (c.f. [43, Example 3.1.7]).

The operators r and ℓ defined for columns of type B_n can be extended to columns of type D_n as follows: for any D_n column β, let β_0 be the column obtained by replacing all subwords $\overline{n}n$ by 00 in β. Note that β_0 is always a B_n column. Let $r(\beta)$ and $\ell(\beta)$ be $r(\beta_0)$ and $\ell(\beta_0)$ (as defined for type B_n columns). Observe that if β is a type D_n column that does not contain a subword $\overline{n}n$, it is also a B_n column and $\beta_0 = \beta$ and so the definitions of $r(\beta)$ and $\ell(\beta)$ coincide regardless of whether β is viewed as a column of type B_n or D_n.

A column β of type B_n, C_n or D_n is admissible if and only if both $r(\beta)$ and $\ell(\beta)$ are defined [41, Proposition 4.3.3] (see also [68] for type C_n). This fact will be important in the definition of tableaux in the following subsection.

3.2. Tabloids and tableaux

Let X be one of the types A_n, B_n, C_n, D_n or G_2. A *tabloid* of type X is a sequence of admissible columns $\beta_r, \ldots, \beta_1$ of type X, which we write in a planar form by writing each column vertically beside each other in the order $\beta_r, \ldots, \beta_1$ from left to right.

For brevity, we also use the inline form $\boxed{\beta_r}\cdots\boxed{\beta_1}$ to denote the tableau with columns $\beta_r, \ldots, \beta_1$. The *reading* $w(T)$ of a tabloid $T = \boxed{\beta_r}\cdots\boxed{\beta_1}$ is the word $w(\beta_1)\cdots w(\beta_r)$. Note that the columns of the tabloid are read from rightmost to leftmost, and each column is read from top to bottom.

Note that different tabloids may have the same reading.

For any word $u \in \mathcal{X}^*$ there is at least one tabloid whose reading is u: if $u = u_1 \cdots u_k$, where $u_i \in \mathcal{X}$, then the tabloid $\boxed{u_k}\cdots\boxed{u_1}$ has reading u. (Notice that each column $\boxed{u_i}$ (of height 1) is admissible.)

We now define a relation $\preceq$ on the sets of admissible columns of each type. For types A_n, B_n, C_n, and D_n, the definition proceeds as follows: for two admissible columns β_1 and β_2, define

- $\beta_2 \leq \beta_1$ if $h(\beta_2) \geq h(\beta_1)$ and the rows of the tabloid $\boxed{\beta_2}\boxed{\beta_1}$ are weakly increasing from left to right;
- $\beta_2 \preceq \beta_1$ if $r(\beta_2) \leq \ell(\beta_1)$.

Note that for any admissible column β, we have $\ell(\beta) \leq \beta \leq r(\beta)$; hence $\beta_2 \preceq \beta_1$ implies $\beta_2 \leq \beta_1$.

For type G_2, the definition is more complicated: for columns β_1 and β_2, define

$$\boxed{a} \preceq \boxed{b} \iff (a \leq b) \wedge \big((a,b) \neq (0,0)\big)$$

$$\begin{array}{|c|}\hline a \\ \hline b \\ \hline\end{array} \preceq \boxed{c} \iff (a \leq c) \wedge \big((a,c) \neq (0,0)\big)$$

$$\begin{array}{|c|}\hline a \\ \hline b \\ \hline\end{array} \preceq \begin{array}{|c|}\hline c \\ \hline d \\ \hline\end{array} \iff \begin{aligned}&(a \leq c) \wedge \big((a,c) \neq (0,0)\big) \\ &\wedge (b \leq d) \wedge \big((b,d) \neq (0,0)\big) \\ &\wedge \big(a \in \{2,3,0\} \implies \operatorname{dist}(a,d) \geq 3\big) \\ &\wedge \big(a = \overline{3} \implies \operatorname{dist}(a,d) \geq 2\big)\end{aligned}$$

Note that the relation $\preceq$ is transitive and antisymmetric, but is not reflexive in general.

Let β_1, β_2 be columns of type D_n such that $h(\beta_2) \geq h(\beta_1)$. We say that the tabloid $\boxed{\beta_2}\boxed{\beta_1}$ contains an *a-configuration*, with $a \notin \{\overline{n}, n\}$, if:

- $a = x_p$, $\overline{n} = x_r$ are symbols of β_2 and $\overline{a} = y_s$, $n = y_q$ symbols of β_1; or
- $a = x_p$, $n = x_r$ are symbols of β_2 and $\overline{a} = y_s$, $\overline{n} = y_q$ symbols of β_1

where the integers p, q, r, s are such that $p \leq q < r \leq s$. Denote by $\mu(a)$ the integer defined by $\mu(a) = s - p$.

A *tableau* of type A_n, B_n, C_n or G_2 is a tabloid $\boxed{\beta_r}\cdots\boxed{\beta_1}$ such that $\beta_{i+1} \preceq \beta_i$ for all $i = 1, \ldots, r-1$. A tableau of type D_n is a tabloid $\boxed{\beta_r}\cdots\boxed{\beta_1}$ such that $\beta_{i+1} \preceq \beta_i$ and the tabloid $\boxed{r(\beta_{i+1})}\boxed{\ell(\beta_i)}$ does not contain an a-configuration with $\mu(a) = n - a$, for all $i = 1, \ldots, r-1$.

Lemma 3.1. *Let $T = \boxed{\beta_m}\cdots\boxed{\beta_1}$ be a tabloid of type A_n, B_n, C_n, D_n, and G_2. Let $i \in \{1, \ldots, n\}$. Let $u_j = w(\beta_j)$ for $j \in 1, \ldots, m$, so that $w(T) = u_1 \cdots u_m$. Suppose $v = \tilde{f}_i(w(T))$ (respectively, $v = \tilde{e}_i(w(T))$) is defined. Factor v as $v = v_1 \cdots v_m$, where $|v_j| = |u_j|$. Then:*

1. *There exists some $k \in \{1, \ldots, m\}$ such that $v_j = u_j$ for $j \neq k$ and $v_k = \tilde{f}_i(u_k)$ (respectively, $v_k = \tilde{e}_i(u_k)$).*
2. *Each word v_j is an admissible column word, and so v is the reading of the tabloid $\boxed{\gamma_m}\cdots\boxed{\gamma_1}$.*
3. *For all $j \in \{1, \ldots, m-1\}$, we have $\beta_{j+1} \preceq \beta_j$ if and only if $\gamma_{j+1} \preceq \gamma_j$. In particular, T is a tableau if and only if $\boxed{\gamma_m}\cdots\boxed{\gamma_1}$ is a tableau.*

Proof. See [39] for types A_n, B_n, C_n, and D_n; see [41] for type G_2. ☐

In light of the preceding lemma, we can think of applying the operators $\tilde{e}_i$ and $\tilde{f}_i$ to a tabloid T: using the notation of the lemma, $\tilde{f}_i(T)$ (respectively, $\tilde{e}_i(T)$), when defined, is the tabloid $\boxed{\gamma_m}\cdots\boxed{\gamma_1}$. Note that $\tilde{f}_i$ and $\tilde{e}_i$ preserve shapes of tabloids and preserve the $\preceq$ relation between adjacent columns, and in particular preserve tableaux. Thus the words in a given connected component are readings of tabloids with the same shape. Furthermore, iterated application of this lemma shows that in a given connected component of one of the crystal graphs, either every word is the reading of a tableau or no word is the reading of a tableau. In a connected component where every word is the reading of a tableau, all the corresponding tableaux have the same shape. (However, it is not true in general that two same-shape tabloids belong to the same component.)

We can now say that a tabloid T has highest weight if $\tilde{e}_i(T)$ is undefined for all i. Note that this is equivalent to the word $w(T)$ being of highest weight. Furthermore, we have the following characterization of highest weight tableaux:

Lemma 3.2. *Let X be one of the types A_n, B_n, C_n, D_n, and G_2. An X tableau has highest weight if and only if it has i-th row filled with i, for $i = 1, \ldots, n$, except that in the D_n case the n-th row can instead be filled with $\overline{n}$.*

Proof. See [39] for types A_n, B_n, C_n, and D_n; see [41] for type G_2. □

Note also that Lemma 3.2 can be recovered easily using the definition of the operators $\tilde{e}_i$ and the relation $\preceq$.

Theorem 3.3 ([41])**.** *Let X be one of the types A_n, B_n, C_n, D_n, and G_2, and let $\mathcal{X}$ be the corresponding alphabet $\mathcal{A}_n$, $\mathcal{B}_n$, $\mathcal{C}_n$, $\mathcal{D}_n$ or $\mathcal{G}_2$. Then for any $u \in \mathcal{X}^*$, there is a unique tableau $P(u)$ such that $u \sim_X w(P(u))$. Thus the set of tableaux form a cross-section of the monoid* $\mathrm{Pl}(X) = \mathcal{X}^*/{\sim_X}$.

3.3. Presentations for plactic monoids

The classical plactic monoid $\mathrm{Pl}(A_n) = \mathcal{A}_n^*/{\sim_{A_n}}$ is presented by the so-called *Knuth relations* [2]. Similarly, all other types of plactic monoids are presented by certain defining relations as described in [41, § 5.1]. In particular, for the cases B_n, C_n and D_n the Knuth relations are also part of the given defining relations.

In order to facilitate the reading of this article, we shall give more details on some of the defining relations that appear in cases B_n, C_n and D_n. These relations are labelled in [41, § 5.1] as CR^X, for $X \in \{B, C, D\}$. We shall refer to the set of these relation as $\mathcal{R}_5^X$, where X is one of the types B_n, C_n and D_n. For our purposes we use the convention that $\overline{0} = 0$ and that $\overline{\overline{z}} = z$.

A relation from $\mathcal{R}_5^{B_n}$ is defined as follows: let $w = w(C)$ be a non-admissible column word for which each strict factor (that is, a factor of w not equal to w) is admissible; let z be the smallest (with respect to $<$) unbarred symbol of w such that the pair $(z, \overline{z})$ occurs in w and $N_C(z) > z$, otherwise set $z = 0$. Let $\widetilde{w}$ be the column word obtained by erasing the pair $(z, \overline{z})$ in w if $z \leq n$ and erasing 0 otherwise. The relation $\mathcal{R}_5^{B_n}$ consists of all such pairs $(w, \tilde{w})$. (See [43, Definition 3.2.2].)

Both sets $\mathcal{R}_5^{C_n}$ and $\mathcal{R}_5^{D_n}$ are equal to $\mathcal{R}_5^{B_n}$ except that we naturally exclude defining relations that involve 0. (See [41, Definitions 5.1.2 and 5.1.3].)

We now state the following auxiliary results that we will use in the sequel:

Lemma 3.4 ([Commuting columns lemma (CCL)]). *Let X be one of the types A_n, B_n, C_n, D_n and let $\mathcal{X}$ be the corresponding alphabet $\mathcal{A}_n$, $\mathcal{B}_n$, $\mathcal{C}_n$ or $\mathcal{D}_n$. Let $\alpha, \beta \in \mathcal{X}[1,n]^*$ be words that are readings of columns (that is, strictly increasing words) such that every symbol of α appears in β. Then $\alpha\beta =_{\mathrm{Pl}(X)} \beta\alpha$.*

Proof. This follows directly from the Knuth relations; one can also use Schensted's insertion algorithm for $\mathrm{Pl}(A_n)$ (see [8, Chapter 5]) and note that the required defining relations also appear in the presentations for the other types of plactic monoid. $\square$

Lemma 3.5. *Let $\mathcal{X}$ be one of the alphabets $\mathcal{B}_n$, $\mathcal{C}_n$ and $\mathcal{D}_n$. Consider a word $w = 12\cdots q\overline{x_1}\,\overline{x_2}\cdots\overline{x_k}$ for some $q \in \mathcal{X}[1,n]$, $\overline{x_1},\ldots,\overline{x_k} \in \mathcal{X}[\overline{q},\overline{1}]$ and $1 \leq x_k < x_{k-1} < \ldots < x_1 \leq q$. Then $w =_{\mathrm{Pl}(X)} u$, where u is the word obtained from $12\cdots q$ by deleting the symbols $x_1, x_2, \ldots, x_k$. In particular, u is either empty or is an admissible column containing fewer than q symbols.*

Proof. Let $u^{(i)}$ be the word obtained by deleting $x_1,\ldots,x_i$ from $12\cdots q$. Then $u^{(i)}\overline{x_{i+1}} =_{\mathrm{Pl}(X)} u^{(i+1)}$ is an $\mathcal{R}_5^X$ relation. Note that $w = u^{(0)}$. By induction, therefore, $u = u^{(k)}$ is a column with $u^{(0)}\overline{x_1}\ldots\overline{x_k} =_{\mathrm{Pl}(X)} u^{(k)}$. Clearly $|u|$ is less than q. Since u contains only symbols from $\mathcal{X}[1,q]$, it follows that $N_u(z) \leq z$ for all z and so u is an admissible column if it is non-empty. $\square$

We also give details of the presentation defining $\mathrm{Pl}(G_2)$ as it will be frequently mentioned in the following sections. Another reason is because we give this presentation in a slightly different way from Lecouvey [41, Definition 5.1.4]. Note, that the sets of defining relations $\mathcal{R}_1^{G_2}$, $\mathcal{R}_2^{G_2}$, $\mathcal{R}_3^{G_2}$, and $\mathcal{R}_4^{G_2}$, defined below, still correspond to the crystal isomorphisms identified by Lecouvey, and hence these relations generate the same congruence as those of Lecouvey.

Giving a presentation for $\mathrm{Pl}(G_2)$ requires the auxiliary partial map Θ on $\mathcal{G}_2^2$ defined as per the following table:

w	21	31	01	$\overline{3}1$	$\overline{3}2$	$\overline{2}1$	$\overline{2}2$	$\overline{1}1$	$\overline{1}2$	$\overline{2}3$	$\overline{1}3$	$\overline{1}0$	$\overline{1}\overline{3}$	$\overline{1}\overline{2}$
$w\Theta$	12	13	23	20	$2\overline{3}$	30	$3\overline{3}$	00	$0\overline{3}$	$3\overline{2}$	$0\overline{2}$	$\overline{3}\overline{2}$	$\overline{3}\overline{1}$	$\overline{2}\overline{1}$

The monoid $\mathrm{Pl}(G_2)$ is presented by $\langle \mathcal{G}_2 \,|\, \mathcal{R}_1^{G_2} \cup \mathcal{R}_2^{G_2} \cup \mathcal{R}_3^{G_2} \cup \mathcal{R}_4^{G_2} \rangle$ (see [41, Definition 5.1.4]), where

$$\begin{aligned}
\mathcal{R}_1^{G_2} &= \big\{(10,1),(1\overline{3},2),(1\overline{2},3),(2\overline{2},0),(2\overline{1},\overline{3}),(3\overline{1},\overline{2}),(0\overline{1},\overline{1})\big\},\\
\mathcal{R}_2^{G_2} &= \big\{(1\overline{1},\varepsilon)\big\},\\
\mathcal{R}_3^{G_2} &= \big\{\,(abc, a(bc)\Theta) : ab \in \operatorname{im}\Theta, bc \in \operatorname{dom}\Theta\,\big\}\\
&\qquad \cup \big\{\,(abc, (ab)\Theta^{-1}c) : ab \in \operatorname{im}\Theta, b \geq c, bc \neq 00, bc \notin \operatorname{dom}\Theta\,\big\},\\
\mathcal{R}_4^{G_2} &= \big\{(123,110)\big\}\\
&\qquad \cup \big\{\,((abc, (ab)\Theta^{-1}c) : ab \in \operatorname{im}\Theta, bc \in \operatorname{im}\Theta, abc \neq 123\,\big\}.
\end{aligned}$$

4. Basic two-column lemmata

As described in the strategic overview of our proofs in the Introduction, this section examines products of two admissible columns that do not form a tableau. In order to prove that the rewriting system we will construct is terminating, we have to know about the shape of the tableau that result from this product. Informally, we will show that the resulting tableau either:

1. Has fewer entries than the original two columns.
2. Has the same number of entries but only one column.
3. Has the same number of entries, two columns, and has a shorter rightmost column.

The results are given formally in the following two subsections as Lemmata 4.1, 4.2 and 4.4.

To construct finite complete rewriting systems for the Plactic monoids only the basic two-column lemmata, as stated in this section, are needed. In order to establish out biautomaticity results a far more detailed understanding is needed of how products of columns behave. These details will be given in the (non-basic) two-column lemmata in Section 6.

We consider first the classical type A_n, and in a combined way the types B_n, C_n, and D_n, reflecting the increasing order of complexity of the arguments. Type G_2 is considered last, because it uses a rather different approach from the other types.

4.1. Proving the basic two-column lemmata

The following result was originally proved in [60, Lemma 5.7]. We present an alternative proof which uses the Littlewood–Richardson rule for decomposing tensor products of crystals into a disjoint union of connected components; see [29, Theorem 7.4.6.].

Here we use the reference [29]. Full details of the proofs of these results are not given in [29] but may be found in the original paper of Nakashima on this topic [69].

In the following proofs n will be fixed, and by a partition we shall mean a sequence $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$.

Lemma 4.1 (Two-column lemma for type A_n)**.** *If $\alpha, \beta \in \mathcal{A}_n^*$ are columns such that $\beta \not\preceq \alpha$ then the tableau $P(\alpha\beta)$ contains $|\alpha\beta|$ symbols, and consists of either one column or two columns, the rightmost of which contains fewer than $|\alpha|$ symbols.*

Proof. We follow the notation and terminology of [29, Chapter 7]. Let $\alpha, \beta \in \mathcal{A}_n^*$ be columns where $|\alpha| = k$ and $|\beta| = l$, and such that $\beta \not\preceq \alpha$. Let Y be the Young diagram with a single column of height k and let Y' be the Young diagram of a single column of height l. Let $B(Y)$ be the connected component of the crystal graph of all admissible columns with shape Y. Note that α belongs to $B(Y)$. Similarly we define $B(Y')$ and note that β belongs to $B(Y')$.

Now by [29, Theorem 7.4.6.] the tensor product of these two crystal components decomposes as the disjoint union of connected connected components

$$B(Y) \otimes B(Y') \cong \bigoplus_{x_1 x_2 \ldots x_l \in B(Y')} B(Y[x_1, x_2, \ldots, x_l]).$$

By definition $Y[j]$ denotes the diagram obtained by adding a box to the jth row of Y, and $Y[j_1, \ldots, j_r]$ is defined inductively to be the diagram obtained from $Y[j_1, \ldots, j_{r-1}]$ by adding a box at the j_rth row. Here $B(Y[j_1, \ldots, j_r])$ is defined to be $\varnothing$ if at least one of the $Y[j_1, \ldots, j_q]$ with $q \leq r$ is not a Young diagram; see [29, page 165].

In our case, Y is a column of height k and $x_1 x_2 \ldots x_l$ is a reading of a tableau of shape Y', that is, $x_1 < x_2 < \ldots < x_l$ is a strictly increasing sequence from $\mathcal{A}_n$. It follows that $B(Y[x_1, x_2, \ldots, x_l]) \neq \varnothing$ if and only if $x_1, x_2, \ldots, x_l$ is a sequence of the form $1, 2, \ldots, a, k+1, k+2, \ldots, k+(l-a)$ for some non negative integer a such that $a \leq k$ and $k + l - a \leq n$. Note that a can possibly be 0 meaning that the sequence starts at $k+1$. The Young diagram $Y[1, 2, \ldots, a, k+1, k+2, \ldots, k+(l-a)]$ has shape $\nu^{(a)} = (\nu_1, \nu_2, \ldots, \nu_n)$ where

$$\nu_i = \begin{cases} 2 & \text{for } 1 \leq i \leq a \\ 1 & \text{for } a < i \leq k + l - a \\ 0 & \text{for } i > k + l - a. \end{cases}$$

The product $\alpha\beta$ must belong to one of the connect components of $B(Y) \otimes B(Y')$. Therefore $P(\alpha\beta)$ is a tableau of shape $\nu^{(a)}$ for some value of $a \leq k$.

If $a = k$ then $P(\alpha\beta)$ is a tableau with two columns, with the right column being of height k and the left column being of height l. It is straightforward to show that this is only possible if $\beta \preceq \alpha$ (a general argument for this, which also applies in this case, may be found in the proof of Lemma 4.2 below). Since $\beta \not\preceq \alpha$ by assumption, it follows that $P(\alpha\beta)$ has shape $\nu^{(a)}$ for some $a < k$. But then the shape $\nu^{(a)}$ has one column or two columns the rightmost of which has $a < k = |\alpha|$ symbols. This completes the proof. □

One benefit of reproving Lemma 4.1 using the above method is that it can be generalised to the other classical types by applying the generalized Littlewood–Richardson rule for decomposing tensor products of crystals into a disjoint union of connected components; see [29, Theorem 8.6.6.].

Lemma 4.2 (Two-column lemma for types B_n, C_n and D_n)**.** *Let X be one of the types B_n, C_n or D_n, and let $\mathcal{X}$ be the corresponding alphabet from $\mathcal{B}_n$, $\mathcal{C}_n$ or $\mathcal{D}_n$. If $\alpha, \beta \in \mathcal{X}^*$ are admissible columns such that $\beta \not\preceq \alpha$ then the tableau $P(\alpha\beta)$ contains at most $|\alpha\beta|$ symbols, and is either empty or consists of either one column or two columns, the rightmost of which contains fewer than $|\alpha|$ symbols.*

Proof. We follow the notation and terminology of [29, Chapter 8]. The proof is similar to that of Lemma 4.1 but we apply [29, Theorem 8.6.6] in place of [29, Theorem 7.4.6].

Let α and β be admissible columns of one of the types B_n, C_n or D_n, where $|\alpha| = k$ and $|\beta| = l$. Let Y be the Young diagram with a single column of height k. It follows from [29, Theorem 8.6.6] that the shape of the tableau $P(\alpha\beta)$ must be given by a Young diagram of the form $Y[x_1, x_2, \ldots, x_l]$ where $x_1, x_2, \ldots, x_l$ is a reading of an admissible column of height l.

Recall (see Subsection 3.1 above) that readings of the columns types B_n, C_n and D_n have, respectively, the forms $\beta_+\beta_0\beta_- \in \mathcal{B}_n^*$, $\gamma_+\gamma_- \in \mathcal{C}_n^*$ and $\delta_+\delta\delta_- \in \mathcal{D}_n^*$ where these words satisfy the admissibility conditions given in the Subsubsection 3.1.1. In particular δ is filled with symbols n and $\overline{n}$, with different symbols in vertically adjacent cells, and β_0 is filled with the symbol 0. The other sections are filled with strictly increasing sequences with respect to the orderings of the vertices in the respective crystal bases. Given a Young diagram Y of shape $(\lambda_1, \lambda_2, \ldots, \lambda_n)$, $Y[j]$ is defined (see [29, page 205]) by

$$\begin{aligned}
&Y[\,j\,] = (\lambda_1, \ldots, \lambda_j + 1, \ldots, \lambda_n) && \text{for } j = 1, \ldots, n \\
&Y[\,\bar{j}\,] = (\lambda_1, \ldots, \lambda_j - 1, \ldots, \lambda_n) && \text{for } j = 1, \ldots, n \qquad (\diamond)\\
&Y[0] = \begin{cases} Y & \text{if } \lambda_n > 0 \\ (\lambda_1, \ldots, \lambda_{n-1}, -\infty) & \text{if } \lambda_n = 0. \end{cases}
\end{aligned}$$

In general $Y[j]$ will itself not be a Young diagram. Set $B(Y') = \varnothing$ if Y' is not a Young diagram. (Note that in [29, page 205] the authors work with generalised Young diagrams, but for our purposes Young diagrams suffice since we are not concerned with Plactic monoids associated with spin representations in this paper.) Then $Y[x_1, x_2, \ldots, x_l]$ is defined inductively by $Y[x_1, x_2, \ldots, x_{l-1}][x_l]$, where if any of the intermediate stages $Y[x_1, x_2, \ldots, x_q]$ is itself not a Young diagram then we set $B(Y[x_1, x_2, \ldots, x_l]) = \varnothing$.

Now $Y[x_1, x_2, \ldots, x_l]$ is obtained by starting with a column of height k which is a Young diagram with shape $(\lambda_1, \lambda_2, \ldots, \lambda_n)$ where $\lambda_i = 1$ for $i \leq k$ and is equal to 0 otherwise. Then for each of the symbols $x_1, x_2, \ldots, x_l$ and so on from our column reading we carry out one of the operations in $(\diamond)$. If at any stage the symbol $-\infty$ appears the process halts and we set $B(Y[x_1, x_2, \ldots, x_l]) = \varnothing$, so we can assume that is not the case. This means that whenever the symbol 0 is read in this process, the Young diagram remains unchanged. Also, in the D_n case when the δ portion of the word is read, this is an alternating sequence of n and $\overline{n}$ which will ultimately either add 1 to λ_n or subtract 1 from λ_n.

Considering each of the three cases it is straightforward to see that in the end, if $B(Y[x_1, x_2, \ldots, x_l]) \neq \varnothing$, then $Y[x_1, x_2, \ldots, x_l]$ must be a Young diagram with shape $\nu^{(a,b)} = (\nu_1, \nu_2, \ldots, \nu_n)$ for some $a, b \geq 0$, with $2a + b \leq k + l$ and $a \leq k$, where

$$\nu_i = \begin{cases} 2 & \text{for } 1 \leq i \leq a \\ 1 & \text{for } a < i \leq a + b \\ 0 & \text{for } i > a + b. \end{cases}$$

Note that $a = 0$, and $b = 0$, are both possible here.

Suppose that $a = k$. In this case, the diagram $Y[x_1, x_2, \ldots, x_l]$ has more boxes than Y, and so, from the definitions of $Y[j]$, $Y[\bar{j}]$, $Y[0]$, necessarily one of the x_i belongs to $\{1,\ldots,n\}$. Furthermore, since $a = k \geq 1$, then $x_i = 1$ for some $i \in \{1,\ldots,l\}$. Since $x_1 \ldots x_l$ is an admissible column, if $x_i < x_j$ then $i < j$, for all $i,j \in \{1,\ldots,l\}$ (The converse also holds except in case D_n). Thus $x_1 = 1$. Now, suppose that for some $s \in \{1,\ldots,l\}$, we have $x_s = \bar{t}$ for some $t \in \{1,\ldots,n\}$. Let s be minimal under such conditions. From the definitions of $Y[j]$, $Y[\bar{j}]$, and since $Y[x_1,x_2,\ldots,x_l]$ is a Young diagram of shape $\nu^{(k,b)}$, there exists some $i \in \{1,\ldots,l\}$, for which $x_i = t$. Because $x_1 \ldots x_l$ is an admissible column either t appears to the left of $\bar{t}$ in $x_1 \ldots x_l$, or $t = n$ and $\bar{t}t$ is a factor of $x_1 \ldots x_l$ (this situation can only occur in case D_n). Since $x_1 = 1$ and $Y[x_1,x_2,\ldots,x_l]$ is a Young diagram, in the first case we would have $12\ldots tu\bar{t}$, for some word u, as a prefix of $x_1 \ldots x_l$, and in the second case $1\ldots(n-1)\bar{n}n$ as a prefix of $x_1 \ldots x_l$. In both cases, this contradicts the fact that $x_1 \ldots x_l$ is an admissible column. So none of the x_i's is a barred symbol. Now suppose that $x_r = 0$ (only possible in case B_n) for some $r \in \{1,\ldots,l\}$ and choose r to be minimal in those conditions. Because $x_1 \ldots x_l$ is an admissible column and $Y[x_1,x_2,\ldots,x_l]$ is a Young diagram then $x_1 \ldots x_l$ has the form $12\ldots(r-1)0\ldots0$. Also, since $Y[x_1,x_2,\ldots,x_l]$ is a Young diagram and by the definition of $Y[0]$ we necessarily have $r-1 = n$. We get a contradiction since $12\ldots n0\ldots0$ is not an admissible column of type B_n. It follows that $[x_1,x_2,\ldots,x_l] = [1,2,\ldots,l]$, and that $Y[x_1,x_2,\ldots,x_l]$ has shape $\nu^{(k,l-k)}$.

From the above, it is then immediate that $Y[x_1,x_2,\ldots,x_l]$ has shape $\nu^{(k,l-k)}$ if and only if $[x_1,x_2,\ldots,x_l] = [1,2,\ldots,l]$. It follows that in the decomposition

$$B(Y) \otimes B(Y') \cong \bigoplus_{x_1x_2\ldots x_l \in B(Y')} B(Y[x_1,x_2,\ldots,x_l])$$

into connected components given by [29, Theorem 8.6.6] the component $B(Y[1,2,\ldots,l])$ occurs exactly once and thus no other connected component of $B(Y) \otimes B(Y')$ is isomorphic to $B(Y[1,2,\ldots,l])$. Since $\alpha\beta$ belongs to the connected component $B(Y[1,2,\ldots,l])$ of $B(Y) \otimes B(Y')$ it follows that $P(\alpha\beta) = \alpha'\beta'$ where β' and α' are admissible columns with $|\alpha'| = k = |\alpha|$ and $|\beta'| = l = |\beta|$. But then $\alpha'\beta'$ belongs to $B(Y) \otimes B(Y')$ and must also belong to the same connected component $B(Y[1,2,\ldots,l])$. Then from $\alpha'\beta' = P(\alpha\beta)$ it follows that $\alpha\beta$ and $\alpha'\beta'$ have the same position in this connected component and hence $\alpha'\beta'$ and $\alpha\beta$ are identical as words which in turn implies that $\beta' = \beta$ and $\alpha' = \alpha$. This implies that $\beta \preceq \alpha$.

Since by assumption $\beta \not\preceq \alpha$, it follows from the arguments above that $a < k$. Then $Y[x_1,x_2,\ldots,x_l]$ has shape $\nu^{(a,b)}$, with $a < k$, which by inspection satisfies the conclusions given in the statement of the lemma, completing the proof. □

4.2. Two column lemma for type G_2

The proof for G_2 uses a rather different approach from the other types. As in the previous subsection, our aim is to learn about the shape of the tableau $P(\alpha\beta)$, where α and β are admissible G_2 columns and $\beta \not\preceq \alpha$; for the conclusion, see Lemma 4.4. Recall that there are only finitely many admissible G_2 columns (which are listed in (3.1)). Thus, our approach is simply to characterize the finitely many possibilities for α and β when $\alpha\beta$ is highest weight in Lemma 4.3, and then to compute $P(\alpha\beta)$ in each case and derive the conclusion about products of arbitrary pairs of admissible columns in Lemma 4.4.

Lemma 4.3. *Let α and β be admissible G_2 column words such that $\beta \not\preceq \alpha$ and $\alpha\beta$ is a highest-weight word. Either:*

1. *$\alpha = 1$ and $\beta \in \{2, 0, \overline{1}, 23, 00\}$; or*
2. *$\alpha = 12$ and $\beta \in \{1, 3, \overline{2}, 13, 30, 3\overline{3}, \overline{21}\}$.*

Proof. Since $\alpha\beta$ is of highest weight, by Lemma 2.5, α is a highest weight column (and thus a highest-weight tableau). The highest weight admissible columns of lengths 1 and 2 are 1 and 12, so either $\alpha = 1$ or $\alpha = 12$.

1. Suppose $\alpha = 1$. Let $\beta = x\beta'$, where $x \in \mathcal{G}_2$. If $x = 1$, then $\beta \preceq \alpha$, which is a contradiction. Furthermore,
$$\begin{aligned} x = 3 &\implies \rho_2(\alpha\beta) = \rho_2(13\beta') = -\rho_2(\beta') = -\cdots; \\ x = \overline{3} &\implies \rho_1(\alpha\beta) = \rho_1(1\overline{3}\beta') = +--\rho_1(\beta') = -\cdots; \\ x = \overline{2} &\implies \rho_2(\alpha\beta) = \rho_2(1\overline{2}\beta') = -\rho_2(\beta') = -\cdots. \end{aligned}$$
In each case, the supposition contradicts $\alpha\beta$ being of highest weight. So x must be 2, 0, or $\overline{1}$; if $|\beta| = 1$, these are the possibilities for β.
Suppose now that $|\beta| = 2$. This cannot occur when $x = \overline{1}$, for no admissible column begins with $\overline{1}$. The admissible column words of length 2 beginning with 2 and 0 are 23, 20, $2\overline{3}$, and 00, $0\overline{3}$ and $0\overline{2}$. Furthermore,
$$\begin{aligned} \beta = 20 &\implies \rho_1(\alpha\beta) = \rho_1(120) = +--+ = -+, \\ \beta = 2\overline{3} &\implies \rho_1(\alpha\beta) = \rho_1(12\overline{3}) = +--- = --, \\ \beta = 0\overline{3} &\implies \rho_1(\alpha\beta) = \rho_1(10\overline{3}) = +-+-- = -, \\ \beta = 0\overline{2} &\implies \rho_2(\alpha\beta) = \rho_2(10\overline{2}) = -, \end{aligned}$$
each of which contradicts $\alpha\beta$ being of highest weight. The remaining possibilities are $\beta = 23$ and $\beta = 00$.
2. Suppose $\alpha = 12$. Let $\beta = x\beta'$, where $x \in \mathcal{G}_2$. Then
$$\begin{aligned} x = 2 &\implies \rho_1(\alpha\beta) = \rho_1(122\beta') = +--\rho_1(\beta') = -\cdots, \\ x = 0 &\implies \rho_1(\alpha\beta) = \rho_1(120\beta') = +--+\rho_1(\beta') = -+\cdots, \\ x = \overline{3} &\implies \rho_1(\alpha\beta) = \rho_1(12\overline{3}\beta') = +---\rho_1(\beta') = --\cdots, \\ x = \overline{1} &\implies \rho_1(\alpha\beta) = \rho_1(12\overline{1}\beta') = +--\rho_1(\beta') = -\cdots, \end{aligned}$$
each of which contradicts $\alpha\beta$ being of highest weight. So x must be 1, 3, or $\overline{2}$. If $|\beta| = 1$, these are the possibilities for β.
Suppose now that $|\beta| = 2$. The admissible column words of length 2 beginning with 1, 3, and $\overline{2}$ are 12, 13, 30, $3\overline{3}$, $3\overline{2}$, $\overline{21}$. Note first that $\beta \neq 12$ since $\beta \not\preceq \alpha$. Furthermore
$$\beta = 3\overline{2} \implies \rho_2(\alpha\beta) = \rho_2(123\overline{2}) = +-- = -,$$

Table 1: Case analysis for the proof of Lemma 4.4.

Shape of $\boxed{\beta}\boxed{\alpha}$	α	β	*Defining relations applied*	$P(\alpha\beta)$	*Shape of* $P(\alpha\beta)$
$\square\square$	1	2	—	12	$\begin{smallmatrix}\square\\\square\end{smallmatrix}$
$\square\square$	1	0	$10 =_{\mathcal{R}_1^{G_2}} 1$	1	$\square$
$\square\square$	1	$\overline{1}$	$1\overline{1} =_{\mathcal{R}_2^{G_2}} \varepsilon$	ε	
$\begin{smallmatrix}\square&\square\\\square&\end{smallmatrix}$	1	23	$123 =_{\mathcal{R}_4^{G_2}} 110 =_{\mathcal{R}_1^{G_2}} 11$	11	$\square\square$
$\begin{smallmatrix}\square&\square\\\square&\end{smallmatrix}$	1	00	$100 =_{\mathcal{R}_1^{G_2}} 10 =_{\mathcal{R}_1^{G_2}} 1$	1	$\square$
$\begin{smallmatrix}\square&\square\\&\square\end{smallmatrix}$	12	1	$121 =_{\mathcal{R}_3^{G_2}} 112$	112	$\begin{smallmatrix}\square&\square\\\square&\end{smallmatrix}$
$\begin{smallmatrix}\square&\square\\&\square\end{smallmatrix}$	12	3	$123 =_{\mathcal{R}_4^{G_2}} 110 =_{\mathcal{R}_1^{G_2}} 11$	11	$\square\square$
$\begin{smallmatrix}\square&\square\\&\square\end{smallmatrix}$	12	$\overline{2}$	$12\overline{2} =_{\mathcal{R}_1^{G_2}} 10 =_{\mathcal{R}_1^{G_2}} 1$	1	$\square$
$\begin{smallmatrix}\square&\square\\\square&\square\end{smallmatrix}$	12	13	$1213 =_{\mathcal{R}_3^{G_2}} 1123 =_{\mathcal{R}_4^{G_2}} 1110 =_{\mathcal{R}_1^{G_2}} 111$	111	$\square\square\square$
$\begin{smallmatrix}\square&\square\\\square&\square\end{smallmatrix}$	12	30	$1230 =_{\mathcal{R}_4^{G_2}} 1100 =_{\mathcal{R}_1^{G_2}} 110 =_{\mathcal{R}_1^{G_2}} 11$	11	$\square\square$
$\begin{smallmatrix}\square&\square\\\square&\square\end{smallmatrix}$	12	$3\overline{3}$	$123\overline{3} =_{\mathcal{R}_4^{G_2}} 110\overline{3} =_{\mathcal{R}_1^{G_2}} 11\overline{3} =_{\mathcal{R}_1^{G_2}} 12$	12	$\begin{smallmatrix}\square\\\square\end{smallmatrix}$
$\begin{smallmatrix}\square&\square\\\square&\square\end{smallmatrix}$	12	$\overline{21}$	$12\overline{21} =_{\mathcal{R}_1^{G_2}} 10\overline{1} =_{\mathcal{R}_1^{G_2}} 1\overline{1} =_{\mathcal{R}_2^{G_2}} \varepsilon$	ε	

which contradicts $\alpha\beta$ being of highest weight. The remaining possibilities for β are 13, 30, $3\overline{3}$, and $\overline{21}$. □

Lemma 4.4 (Two-column lemma for type G_2)**.** *Let α and β be admissible G_2 columns with $\beta \not\preceq \alpha$. Then either:*

- *$P(\alpha\beta)$ contains fewer that $|\alpha\beta|$ symbols,*
- *$P(\alpha\beta)$ contains exactly $|\alpha\beta|$ symbols and has at most one column,*
- *$P(\alpha\beta)$ contains exactly $|\alpha\beta|$ symbols and has exactly two columns, the rightmost of which contains fewer than $|\alpha|$ symbols.*

Proof. Since the Kashiwara operators preserve shapes of tabloids and also preserves whether the $\preceq$ relation holds between adjacent columns, we can assume that $\alpha\beta$ has highest weight. Using Lemma 4.3, we systematically enumerate the possible words $\alpha\beta$ and calculate their corresponding tableaux. The results are shown in Table 1.

In each case, we get a tableau that contains fewer that $|\alpha\beta|$ symbols (and that is in some case empty), and in the cases when the number of symbols in the tableau is equal to $|\alpha\beta|$, either the tableau contains only one column, or else contains two columns and the number of symbols in the rightmost column is less than $|\alpha|$. □

5. Constructing the rewriting system

We now turn to actually constructing the finite complete rewriting systems presenting $\mathrm{Pl}(A_n)$, $\mathrm{Pl}(B_n)$, $\mathrm{Pl}(C_n)$, $\mathrm{Pl}(D_n)$, and $\mathrm{Pl}(G_2)$. The constructions can be carried out in parallel, because the only differences are the appeals to the different lemmata from Section 4. We first of all recall the necessary definitions about rewriting systems in Subsection 5.1; for further background, see [47] or [70]. For background on semigroup presentations generally, see [71] or [72].

5.1. Preliminaries

Let $\leq$ be a total order on an alphabet A. Define a total order $\leq_{\mathrm{lex}}$ on A^* by $w \leq_{\mathrm{lex}} w'$ if and only if either w is proper prefix of w' or if $w = paq$, $w' = pbr$ and $a \leq b$ for some $p, q, r \in A^*$, and $a, b \in A$. The order $\leq_{\mathrm{lex}}$ is the *lexicographic order induced by* $\leq$. Notice that $\leq_{\mathrm{lex}}$ is not a well-order, but that it is left compatible with concatenation. Define also a total order $\leq_{\mathrm{lenlex}}$ on A^* by

$$w \leq_{\mathrm{lenlex}} w' \iff (|w| < |w'|) \vee \big((|w| = |w'|) \wedge (w \leq_{\mathrm{lex}} w')\big).$$

The order $\leq_{\mathrm{lenlex}}$ is the *length-plus-lexicographic order induced by* $\leq$. The order $\leq_{\mathrm{lenlex}}$ is a well-order and is left compatible with concatenation.

A *string rewriting system*, or simply a *rewriting system*, is a pair (A, R), where A is a finite alphabet and R is a set of pairs (ℓ, r), usually written $\ell \to r$, known as *rewriting rules* or simply *rules*, drawn from $A^* \times A^*$. The single reduction relation $\to_R$ is defined as follows: $u \to_R v$ (where $u, v \in A^*$) if there exists a rewriting rule $(\ell, r) \in R$ and words $x, y \in A^*$ such that $u = x\ell y$ and $v = xry$. That is, $u \to_R v$ if one can obtain v from u by substituting the word r for a subword ℓ of u, where $\ell \to r$ is a rewriting rule. The reduction relation $\to_R^*$ is the reflexive and transitive closure of $\to_R$. The process of replacing a subword ℓ by a word r, where $\ell \to r$ is a rule, is called *reduction* by application of the rule $\ell \to r$; the iteration of this process is also called reduction. A word $w \in A^*$ is *reducible* if it contains a subword ℓ that forms the left-hand side of a rewriting rule in R; it is otherwise called *irreducible*.

The rewriting system (A, R) is *finite* if both A and R are finite. The rewriting system (A, R) is *noetherian* if there is no infinite sequence $u_1, u_2, \ldots \in A^*$ such that $u_i \to_R u_{i+1}$ for all $i \in \mathbb{N}$. That is, (A, R) is noetherian if any process of reduction must eventually terminate with an irreducible word. The rewriting system (A, R) is *confluent* if, for any words $u, u', u'' \in A^*$ with $u \to_R^* u'$ and $u \to_R^* u''$, there exists a word $v \in A^*$ such that $u' \to_R^* v$ and $u'' \to_R^* v$. A rewriting system that is both confluent and noetherian is *complete*. If (A, R) is a complete rewriting system, then for every word u there is a unique irreducible word w such that $u \to_R^* w$; this word is called the *normal form* of u. If (A, R) is complete, then the language of normal form words forms a cross-section of the monoid: that is, each element of the monoid presented by $\langle A \mid R \rangle$ has a unique normal form representative.

5.2. Construction

Let X be one of the types A_n, B_n, C_n, D_n, and G_2, and let $\mathcal{X}$ be the corresponding alphabet from $\mathcal{A}_n$, $\mathcal{B}_n$, $\mathcal{C}_n$, $\mathcal{D}_n$, or $\mathcal{G}_2$. Let

$$\Sigma = \{\, c_\sigma : \sigma \text{ is an admissible } X \text{ column} \,\}.$$

Note that Σ is finite since there are finitely many admissible X columns.

Let T consist of the following rewriting rules:

$$c_\sigma c_\tau \to \varepsilon \qquad \tau \not\preceq \sigma \text{ and } P(\sigma\tau) \text{ is empty,} \tag{5.1}$$

$$c_\sigma c_\tau \to c_\upsilon \qquad \tau \not\preceq \sigma \text{ and } P(\sigma\tau) \text{ is the 1-col. tableau } \boxed{\upsilon}, \tag{5.2}$$

$$c_\sigma c_\tau \to c_\upsilon c_\phi \qquad \tau \not\preceq \sigma \text{ and } P(\sigma\tau) \text{ is the 2-col. tableau } \boxed{\phi}\boxed{\upsilon}, \tag{5.3}$$

$$c_\sigma c_\tau \to c_\upsilon c_\phi c_\chi \qquad \tau \not\preceq \sigma \text{ and } P(\sigma\tau) \text{ is the 3-col. tableau } \boxed{\chi}\boxed{\phi}\boxed{\upsilon}. \tag{5.4}$$

Note that since $P(\sigma\tau)$ is a tableau, the subscripts υ, ϕ, and χ are always admissible columns.

Note that if σ, τ are admissible columns with $\tau \not\preceq \sigma$, then $P(\sigma\tau)$ has at most three columns by Lemmata 4.1, 4.2, and 4.4 (that is, by the two-column lemmata for type A_n, types B_n, C_n, D_n, and type G_2). Thus every such pair of columns gives rise to a rewriting rule in T. (Note that rules of the form $c_\sigma c_\tau \to c_\upsilon c_\phi c_\chi$ only arise when $X = G_2$, because $P(\sigma\tau)$ has at most two columns in the other cases.) Finally, note that T is finite since there are finitely many possibilities for σ and τ, and the right-hand side of each rule is uniquely determined by the left-hand side.

The idea is that a word $c_{\beta^{(1)}} \cdots c_{\beta^{(m)}}$ corresponds to the tabloid $\boxed{\beta^{(m)}} \cdots \boxed{\beta^{(1)}}$, and that if this tabloid is not a tableau, then there are two adjacent columns between which the relation $\preceq$ does not hold. These columns (as represented by some subword $c_\sigma c_\tau$ with $\tau \not\preceq \sigma$) are rewritten to a tableau (as represented by a word in Σ^*). Thus, in terms of words in Σ^*, tabloids are rewritten to become more 'tableau-like', and the irreducible words correspond to tableaux.

Lemma 5.1. *The rewriting system* (Σ, T) *is noetherian.*

Proof. Let $\trianglelefteq$ be any total order on Σ that extends the partial order induced by lengths of columns, in the sense that $|\sigma| \leq |\tau| \implies c_\sigma \trianglelefteq c_\tau$ for any two admissible columns σ and τ.

Let the map $L : \Sigma^* \to \mathbb{N} \cup \{0\}$ send each word to the sum of the lengths of the subscripts of its symbols: that is,

$$L\big(c_{\sigma^{(1)}} c_{\sigma^{(2)}} \cdots c_{\sigma^{(h)}}\big) = \sum_{i=1}^{h} |\sigma^{(i)}|.$$

Define a total order $\sqsubset$ on Σ^* by

$$\begin{aligned} u \sqsubset v \iff & \big(L(u) < L(v)\big) \\ & \vee \Big(\big(L(u) = L(v)\big) \wedge \big(u \trianglelefteq_{\text{lenlex}} v\big)\Big). \end{aligned}$$

That is, $\sqsubset$ first orders by the total number of symbols in the tabloid to which a word corresponds, then by the length of the word, and then lexicographically based on the ordering $\trianglelefteq$ of Σ. Note that $\sqsubset$ is compatible with multiplication in the free monoid Σ^*.

Let $c_\alpha c_\beta$ be the left-hand side of a rewriting rule and let w be its right-hand side. So $\beta \not\preceq \alpha$. Consider two cases:

- For $X = A_n$ (respectively, $X = B_n$, C_n, or D_n), Lemma 4.1 (respectively, 4.2) shows that $P(\alpha\beta)$ contains at most $|\alpha\beta|$ symbols (so that $L(w) \leq L(c_\alpha c_\beta)$), and consists of at most two columns (so that $|w| \leq |\alpha\beta|$) and the rightmost column contains fewer than $|\alpha|$ symbols (so that $w \vartriangleleft_{\text{lex}} \alpha\beta$). Thus $w \sqsubset \alpha\beta$.

- For $X = G_2$, Lemma 4.4 shows that $P(\alpha\beta)$ contains most $|\alpha\beta|$ symbols (so $L(w) \leq L(c_\alpha c_\beta)$), and that, if $P(\alpha\beta)$ contains exactly $|\alpha\beta|$ symbols (so that $L(w) = L(c_\alpha c_\beta)$), then it either consists of one column (so that $|w| < |\alpha\beta|$ and so $w \vartriangleleft_{\text{lenlex}} \alpha\beta$) or it consist of two columns and the rightmost column contains fewer than $|\alpha|$ symbols (so that $|w| = |\alpha\beta|$ and $w \vartriangleleft_{\text{lex}} \alpha\beta$, and hence $w \trianglelefteq_{\text{lenlex}} \alpha\beta$). Thus $w \sqsubset \alpha\beta$.

Since $\sqsubset$ is compatible with multiplication in the free monoid Σ^*, rewriting a word always decreases it with respect to $\sqsubset$. Since there are no infinite $\sqsubset$-descending chains, any process of rewriting must terminate. Hence (Σ, T) is noetherian. $\square$

Lemma 5.2. *The rewriting system* (Σ, T) *is confluent.*

Proof. Let $u \in \Sigma^*$ and let u' and u'' be words with $u \to^* u'$ and $u \to^* u''$. By Lemma 5.1, there are irreducible words $w' = c_{\beta^{(1)}} \cdots c_{\beta^{(k)}}$ and $w'' = c_{\gamma^{(1)}} \cdots c_{\gamma^{(m)}} \in \Sigma^*$ such that $u' \to^* w'$ and $u'' \to^* w''$. Since w' is irreducible, it does not contain the left-hand side of any rule in T. Thus, by the comments after the definition of T, we have $\beta^{(j+1)} \preceq \beta^{(j)}$ for $j = 1, \ldots, k-1$. That is, $\boxed{\beta^{(k)}}\cdots\boxed{\beta^{(1)}}$ is a tableau. Similarly, $\boxed{\gamma^{(m)}}\cdots\boxed{\gamma^{(1)}}$ is a tableau (with m columns). But the readings of these tableau (that is, $\beta^{(1)} \cdots \beta^{(k)}$ and $\gamma^{(1)} \cdots \gamma^{(m)}$) are equal in $\operatorname{Pl}(X)$, and tableaux form a cross-section of $\operatorname{Pl}(X)$ by Theorem 3.3. Hence $k = m$ and $\beta^{(j)} = \gamma^{(j)}$ for $j = 1, \ldots, k$, and so $w' = w''$. Thus $v = w' = w''$ is a word such that $u' \to^* v$ and $u'' \to^* v$. Therefore (Σ, T) is confluent. $\square$

Theorem 5.3. *For any* $X \in \{A_n, B_n, C_n, D_n, G_2\}$*, there is a finite complete rewriting system* (Σ, T) *that presents* $\operatorname{Pl}(X)$*.*

Proof. Construct the finite complete rewriting system (Σ, T) as above. It remains to prove that $\langle \Sigma \mid T \rangle$ presents $\operatorname{Pl}(X)$. To this end, let $\langle \mathcal{X} \mid \mathcal{R}^X \rangle$ be the presentation for $\operatorname{Pl}(X)$ as described in [41, § 5.1] and also in Subsection 3.3 for type G_2. We are going to prove that $\langle \Sigma \mid T \rangle$ and $\langle \mathcal{X} \mid \mathcal{R}^X \rangle$ present the same monoid.

First notice that if $\sigma = \sigma_1 \cdots \sigma_k$ is an admissible column, where $\sigma_i \in \mathcal{X}$, then a sequence of applications of rules from T of type (5.2) lead from $c_{\sigma_1} \cdots c_{\sigma_k}$ to $c_{\sigma_1 \cdots \sigma_k}$:

$$
\begin{aligned}
c_{\sigma_1} c_{\sigma_2} c_{\sigma_3} \cdots c_{\sigma_{k-1}} c_{\sigma_k} &\rightarrow_T c_{\sigma_1\sigma_2} c_{\sigma_3} \cdots c_{\sigma_{k-1}} c_{\sigma_k} \\
&\;\;\vdots \\
&\rightarrow_T c_{\sigma_1\sigma_2\cdots\sigma_{k-1}} c_{\sigma_k} \\
&\rightarrow_T c_{\sigma_1\sigma_2\cdots\sigma_{k-1}\sigma_k}.
\end{aligned}
$$

Thus we can apply Tietze transformations to $\langle \Sigma \,|\, T \rangle$ to replace each symbol $c_{\sigma_1\cdots\sigma_k}$ with $c_{\sigma_1} \cdots c_{\sigma_k}$ and then remove the generators $c_{\sigma_1\cdots\sigma_k}$ with $k > 1$. The result of this is a new presentation $\langle \Sigma' \,|\, T' \rangle$ where the generating symbols in Σ' are c_x for $x \in \mathcal{X}$, so we can replace each c_x by x to obtain a new presentation $\langle \mathcal{X} \,|\, T'' \rangle$. It remains to show that every defining relation in T'' is a consequence of those in $\mathcal{R}^X$ and vice versa.

Note that T'' can be obtained from T by replacing each symbol $c_{\sigma_1\cdots\sigma_k}$ by $\sigma_1 \cdots \sigma_k$. Thus every defining relation in T'' is of the form (u, v), where u is the reading of a two-column tabloid and v is the reading of a tableau, and $u =_{\mathrm{Pl}(X)} v$. Since $\langle \mathcal{X} \,|\, \mathcal{R}^X \rangle$ presents $\mathrm{Pl}(X)$, the defining relation (u, v) is a consequence of $\mathcal{R}^X$.

On the other hand, let (u, v) be a defining relation in $\mathcal{R}^X$. By inspection of the definition of $\mathcal{R}^X$ in [41, § 5.1] and Subsection 3.3, v is the reading of a tableau, and $P(u) = v$. Suppose this tableau is $\boxed{\beta^{(m)}}\;\cdots\;\boxed{\beta^{(1)}}$, where $\beta^{(1)}$, $\ldots$, $\beta^{(m)}$ are admissible columns of type X. Suppose $u = u_1 \cdots u_t$, and note that every symbol u_i is an admissible column of type X. Since $P(u) = v$, the word $c_{u_1} \cdots c_{u_t}$ rewrites to $c_{\beta^{(1)}} \cdots c_{\beta^{(m)}}$ under the rewriting system (Σ, T). Fix a sequence of rewriting $c_{u_1} \cdots c_{u_t} \rightarrow_T^* c_{\beta^{(1)}} \cdots c_{\beta^{(m)}}$. Replacing each symbol $c_{\sigma_1\cdots\sigma_k}$ by $\sigma_1 \cdots \sigma_k$ throughout this sequence of rewriting yields a sequence from $u = u_1 \cdots u_t$ to $\beta^{(1)} \cdots \beta^{(m)} = v$ where every step is an application of a relation from T''. Hence (u, v) is a consequence of T''.

Since every defining relation in T'' is a consequence of those in $\mathcal{R}^X$ and vice versa, $\langle \mathcal{X} \,|\, T'' \rangle$ and $\langle \mathcal{X} \,|\, \mathcal{R}^X \rangle$ present the same monoid, and thus $\langle \Sigma \,|\, T \rangle$ presents $\mathrm{Pl}(X)$. □

The following corollary is immediate [73]:

Corollary 5.4. *The Plactic monoids of types A_n, B_n, C_n, D_n, and G_2 have finite derivation type.*

By a result originally proved by Anick in a different form [74], but also proved by various other authors (see [75, 76]):

Corollary 5.5. *The Plactic monoids of types A_n, B_n, C_n, D_n, and G_2 are of type right and left* FP_∞.

6. Biautomaticity lemmata

In this section, we lay the groundwork for constructing biautomatic structures for plactic monoids in Section 7.

The language of representatives of the biautomatic structure will be the language of irreducible words of the rewriting system (Σ, T) constructed in Section 5. To prove that this gives us a biautomatic structure, we must understand how products of the form $c_x c_{\beta^{(1)}} \cdots c_{\beta^{(\ell)}}$ and $c_{\beta^{(1)}} \cdots c_{\beta^{(\ell)}} c_x$ rewrite, where $c_{\beta^{(1)}} \cdots c_{\beta^{(\ell)}}$ is an irreducible word and $c_x \in \Sigma$ is such that $|x| = 1$. It will suffice to consider the situations where $x\beta^{(1)} \cdots \beta^{(\ell)}$ and $\beta^{(1)} \cdots \beta^{(\ell)} x$ are highest weight words, because, as we shall see, the rewriting of $c_x c_{\beta^{(1)}} \cdots c_{\beta^{(\ell)}}$ and $c_{\beta^{(1)}} \cdots c_{\beta^{(\ell)}} c_x$ proceeds 'in the same way' in the general case.

6.1. Two-column lemma for biautomaticity

Let X be one of the types A_n, B_n, C_n, or D_n, and let $\mathcal{X}$ be the corresponding alphabet from $\mathcal{A}_n$, $\mathcal{B}_n$, $\mathcal{C}_n$, or $\mathcal{D}_n$.

Lemma 6.1. *Let $\alpha, \beta \in \mathcal{X}_n^*$ be admissible X columns such that $\beta \not\preceq \alpha$ and $\alpha\beta$ is a word of highest weight.*

1. *If $X = A_n$, B_n, or C_n, then $\alpha = 1 \cdots p$ for some $p \in \mathcal{X}[1, n]$.*
2. *If $X = D_n$, then either $\alpha = 1 \cdots p$ for some $p \in \mathcal{D}[1, n]$ or $\alpha = 1 \cdots (n - 1)\overline{n}$.*

Proof. By Lemma 2.5, α is a highest weight column (and thus a highest-weight tableau), and thus has the required form by Lemma 3.2. □

Lemma 6.2. *Let $\alpha, \beta \in \mathcal{X}_n^*$ be admissible X columns such that $\beta \not\preceq \alpha$ and $\alpha\beta$ is a word of highest weight. Suppose the first symbol of β is 1. Let $\hat{\beta}$ be the maximal prefix of β whose symbols form an interval of $\mathcal{X}[1, n-1]$ (viewed as an ordered set). Then $P(\alpha\beta)$ consists of two columns and the rightmost column of $P(\alpha\beta)$ is $\hat{\beta}$.*

Proof. By Lemma 6.1, $\alpha = 1 \cdots p$ for some p. Thus both α and β contain 1. Since α and β are admissible columns, both containing 1, neither contains $\overline{1}$. It follows that $P(\alpha\beta)$, as a tableau of the same weight as $\alpha\beta$, also contains two symbols 1 and so has two columns. Suppose $P(\alpha\beta) = \boxed{\delta}\boxed{\gamma}$. By Lemma 4.2, γ contains fewer than $|\alpha|$ symbols. That is, γ contains at most $n-1$ symbols. Thus by Lemma 3.2, since $P(\alpha\beta)$ is highest-weight, $\gamma = 1 \cdots s$ for some $s \in \mathcal{X}[1, n-1]$. Furthermore, again by Lemma 3.2, if $X = A_n$, B_n, or C_n, then $\delta = 1 \cdots t$ for some $t \in \mathcal{X}[1, n]$ with $t \geq s$ (since $\delta \preceq \gamma$), while if $X = D_n$, then either $\delta = 1 \cdots t$ for some $t \in \mathcal{X}[1, n]$ with $t \geq s$ or else $\delta = 1 \cdots (n - 1)\overline{n}$. Thus δ also contains each symbol in $\mathcal{X}[1, s]$ and so $\gamma\delta$ contains two of each symbol in $\mathcal{X}[1, s]$. Since $\alpha\beta$ has the same weight as $\gamma\delta$, it follows that both α and β contain each symbol from $\mathcal{X}[1, s]$. This shows that $\hat{\beta}$ contains $1 \cdots s$ as a prefix; it remains to prove that $\hat{\beta}$ contains no more symbols. If $s = n-1$, this is immediate by the definition of $\hat{\beta}$, so assume henceforth that $s < n - 1$.

Suppose, with the aim of obtaining a contradiction, that $\hat{\beta} \neq 1 \cdots s$. Then, since $1 \cdots s$ is a prefix of $\hat{\beta}$, it follows that $\hat{\beta}$ contains the symbol $s+1$. (Note that $s+1 < n$.)

Consider $\rho_s(\gamma\delta)$. The symbol s in γ contributes $+$ to $\rho_s(\gamma\delta)$. If $\delta = 1 \cdots t$ for $t \in \mathcal{X}[1,n]$, then δ contributes $+$ (if $t = s$) or $+-$ (if $t > s$). If $\delta = 1 \cdots (n-1)\overline{n}$, then δ contributes $++$. In any case, $\rho_s(\gamma\delta)$ contains at least one $+$ and so $\tilde{f}_s(\gamma\delta)$ is defined. Since $\alpha\beta$ and $\gamma\delta$ lie in isomorphic crystal components, $\tilde{f}_s(\alpha\beta)$ is also defined and so $\rho_s(\alpha\beta)$ contains at least one $+$.

Notice that the word α (which, as noted previously, is of the form $1 \cdots p$) contains strictly more than s symbols and so must contain $s+1$. Thus in the calculation of $\rho_s(\alpha\beta)$ the symbols s and $s+1$ in α contribute a $+$ and a $-$, which are deleted. In the word β, the symbols s and $s+1$ also contribute a $+$ and a $-$, which are deleted. The word β cannot contain $\overline{s+1}$, since it is admissible, so no other symbols can contribute a $+$. Hence $\rho_s(\alpha\beta)$ contains no symbols $+$. This is a contradiction, and so $\hat{\beta} = 1 \cdots s$. This completes the proof. $\square$

6.2. Transducers

This subsection briefly recalls the definition of a transducer and the relation it recognizes; for further background, see [77, Chapter IV] or [78].

Informally, a transducer is a (possibly non-deteministic) finite automaton that reads symbols from two tapes (possibly at varying 'speeds') and thus recognizes a binary relation between the sets of words over the two tape alphabets. More formally, a transducer is a tuple (Q, X, Y, I, F, δ), where Q is a finite set of states, X and Y are two finite alphabets, I is a set of distinguished *initial states*, F is a set of distinguished *final states*, and δ is a finite subset of $Q \times X^* \times Y^* \times Q$ called the *transition relation*. When in a state q, it can transition to a state r while reading words $x \in X^*$ and $y \in Y^*$ from its top and bottom input tapes if and only if (q, x, y, r) is in δ. (Note that either or both of x and y can be the empty word.)

The transducer accepts the contents of its input if it can start in some state in I, read the whole content of its input tapes and end in a state in F. More formally, it accepts $(u, v) \in X^* \times Y^*$ if and only if there exist factorizations $u = x_1 \cdots x_k$ and $v = y_1 \cdots y_k$, where $x_i \in X^*$ and $y_i \in Y^*$ and a sequence of states $q_0, \ldots, q_k$ such that $q_0 \in I$, $q_k \in F$, and $(q_{i-1}, x_i, y_i, q_i) \in \delta$ for $i = 1, \ldots, k$.

The transducer is thought of as a finite directed graph with vertex set Q and, for each $(q, x, y, r) \in \delta$, an edge from q to r labelled by (x, y), for some words $x \in X^*$ and $y \in Y^*$. A pair (u, v) is accepted if there is a path from some vertex in I to some vertex in F such that (u, v) is the product in $X^* \times Y^*$ of the labels on that path.

Note that the set of pairs in $X^* \times Y^*$ accepted by the transducer forms a binary relation between X^* and Y^*, called the relation *recognized* by the transducer. A relation between X^* and Y^* recognized by a transducer is said to be *rational* (see [77, Subsection IV.1.2]).

As usual in the theory of automatic groups and semigroups, we will not describe transducers and automata by giving the complete formal definition as tuples of sets and relations; the problem with this is that the technical details become overpowering and obscure the fundamental ideas. Instead, we will give a somewhat higher level description of how the transducers and automata ‘function’. For instance, we will sometimes speak of a transducer or automaton reading a symbol, ‘storing’ it in its state, and later ‘checking’ that symbol. This means that, on reading the symbol, the transition relation must take the transducer or automaton to a state that somehow determines the stored symbol (for instance, states might be tuples and some component of the tuple might be the relevant symbol). ‘Checking’ the stored symbol means that the transducer or automaton enters a failure state if the stored symbol (as determined by the state) is not as required.

6.3. Left-multiplication by transducer

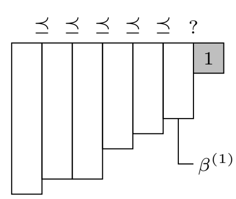
6.3.1. A_n*,* B_n*,* C_n*,* D_n

Let X be one of the types A_n, B_n, C_n, and D_n and let $\mathcal{X}$ be the corresponding alphabet from $\mathcal{A}_n$, $\mathcal{B}_n$, $\mathcal{C}_n$, or $\mathcal{D}_n$. In these cases, the rewriting that occurs on left-multiplication by a generator is very similar, and so we treat these cases in parallel. The goal is to prove Lemma 6.4, which contains all the information we need for the eventual proof of biautomaticity.

We emphasize that in the following analysis, Commuting columns lemma 3.4 is used only as an auxiliary result to prove facts about words, and is not in any way treated as a rewriting rule.

Let $x \in \mathcal{X}$ and let $\beta^{(1)}, \ldots, \beta^{(m)}$ be admissible X columns satisfying $\beta^{(i+1)} \preceq \beta^{(i)}$ for $i = 1, \ldots, m-1$ (that is, $\boxed{\beta^{(m)}}\cdots\boxed{\beta^{(1)}}$ is a tableau), such that $x\beta^{(1)} \cdots \beta^{(m)}$ is a highest-weight word. Recall that $x\beta^{(1)} \cdots \beta^{(h)}$ is a highest-weight word for all $h \leq m$ by Lemma 2.5. In particular, x is a highest-weight word and so $x = 1$. The aim is to examine how the corresponding word over Σ (that is, $c_1 c_{\beta^{(1)}} \cdots c_{\beta^{(m)}}$) is rewritten by T to an irreducible word. We are going to prove that this rewriting involves a single left-to-right pass through the word and that it only changes the length of the word by at most 1.

The tabloid corresponding to $c_1 c_{\beta^{(1)}} \cdots c_{\beta^{(m)}}$ has the following form:



(The symbol ? indicates that either $\preceq$ or $\not\preceq$ may hold between these columns.)

First, it is possible that $\preceq$ holds between 1 and $\beta^{(1)}$. In this case, $c_1 c_{\beta^{(1)}} \cdots c_{\beta^{(m)}}$ is irreducible and so no rewriting occurs. So assume that $\preceq$ does not hold between 1 and $\beta^{(1)}$. Then a rewriting rule applies to $c_1 c_{\beta^{(1)}}$. By Lemma 4.2, $P(1\beta^{(1)})$ has at most two columns. Further, again by Lemma 4.2, if it has exactly two columns, its right-hand column would be strictly shorter than the one-symbol column 1, which is impossible. Thus $P(1\beta^{(1)})$ has at most one column.

If $P(1\beta^{(1)})$ has zero columns (that is, is empty), then the rewriting rule that applies is $c_1 c_{\beta^{(1)}} \to \varepsilon$. That is,

$$c_1 c_{\beta^{(1)}} c_{\beta^{(2)}} \cdots c_{\beta^{(m)}} \to c_{\beta^{(2)}} \cdots c_{\beta^{(m)}}.$$

Since $\beta^{(i+1)} \preceq \beta^{(i)}$ for $i = 2, \ldots, m-1$, the word $c_{\beta^{(2)}} \cdots c_{\beta^{(m)}}$ is irreducible and no further rewriting occurs.

So assume $P(1\beta^{(1)})$ has exactly one column $\gamma^{(1)}$. Since $\gamma^{(1)}\beta^{(2)} \cdots \beta^{(m)}$ is highest-weight, Lemma 2.5 and Lemma 3.2 show in particular that either $\gamma^{(1)} = 1 \cdots s$ for some $s \in \mathcal{X}[1,n]$, or $X = D_n$ and $\gamma^{(1)} = 1 \cdots (n-1)\overline{n}$. Furthermore, since $\gamma^{(1)}$ is one of these forms and $\gamma^{(1)}$ and $1\beta^{(1)}$ have the same weight, it follows that $\gamma^{(1)}$ contains at least two symbols. That is, $\gamma^{(1)}$ contains 2 or $\gamma^{(1)} = 1\overline{2}$, with the latter possible only when $X = D_n$ and $n = 2$.

We now need to know about the column $\beta^{(2)}$:

Lemma 6.3. *The column $\beta^{(2)}$ begins with* 1.

Proof. If $\beta^{(2)} \preceq \gamma^{(1)}$, then since $\gamma^{(1)}$ begins with 1, so does $\beta^{(2)}$. So assume $\beta^{(2)} \not\preceq \gamma^{(1)}$. Consider separately the cases $X = A_n, B_n, C_n, D_n$:

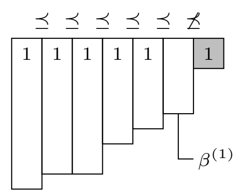
- Suppose $X = A_n$. Since $\gamma^{(1)}$ and $1\beta^{(1)}$ have the same weight and $\gamma^{(1)}$ contains a symbol 2, it follows that the column $\beta^{(1)}$ begins with 2. Since $\beta^{(2)} \preceq \beta^{(1)}$, the column $\beta^{(2)}$ must begin with either 1 or 2. With the aim of obtaining a contradiction, suppose it begins with 2. Then $\rho_1(1\beta^{(1)}\beta^{(2)}) = {+}{-}{-}\cdots = {-}\cdots$, which contradicts $x\beta^{(1)} \cdots \beta^{(m)}$ being of highest weight. Thus $\beta^{(2)}$ begins with 1.

- Suppose $X = B_n$ or $X = C_n$. Since $\gamma^{(1)}$ and $1\beta^{(1)}$ have the same weight and $\gamma^{(1)}$ contains a symbol 2, it follows that $\beta^{(1)}$ begins with 2 and cannot contain $\overline{2}$ or $\overline{1}$. Since $\beta^{(2)} \preceq \beta^{(1)}$, the column $\beta^{(2)}$ must begin with either 1 or 2. With the aim of obtaining a contradiction, suppose it begins with 2. Then $\rho_1(\beta^{(1)})$ is $-$ (the 2 at the start contributes $-$; there cannot be a further $-$ since there is no symbol $\overline{1}$). Hence $\rho_1(1\beta^{(1)}\beta^{(2)}) = {+}{-}{-}\cdots = {-}\cdots$, which contradicts $x\beta^{(1)} \ldots \beta^{(m)}$ being of highest weight. Therefore $\beta^{(2)}$ must begin with 1.

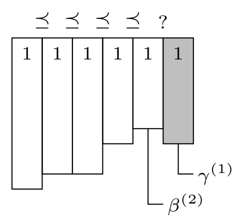
- Suppose $X = D_n$. Consider three sub-cases separately:
  - Suppose that $n = 2$ and $\gamma^{(1)} = 12$. Since $\gamma^{(1)}$ and $1\beta^{(1)}$ have the same weight, it follows that the admissible column $\beta^{(1)}$ is 2. Then $\ell(\beta^{(1)}) = 2$. Hence either $\beta^{(2)} = r(\beta^{(2)}) = 2$ or $\beta^{(2)} = r(\beta^{(2)}) = 1$. With the aim of obtaining a contradiction, suppose the former. Then $\rho_1(1\beta^{(1)}\beta^{(2)}) = {+}{-}{-}\cdots = {-}\cdots$, which contradicts $x\beta^{(1)} \ldots \beta^{(m)}$ being of highest weight. Therefore $\beta^{(2)} = 1$.
  - Suppose that $n = 2$ and $\gamma^{(1)} = 1\overline{2}$. The reasoning showing that $\beta^{(2)} = 1$ proceeds in precisely the same way as the previous sub-case, replacing 2 with $\overline{2}$ and ρ_1 with ρ_2.

– Suppose that $n > 2$. Then $\gamma^{(1)}$ contains a symbol 2. Since $\gamma^{(1)}$ and $1\beta^{(1)}$ have the same weight, it follows that $\beta^{(1)}$ begins with 2 and does not contain $\overline{2}$ or $\overline{1}$. The reasoning reasoning showing that $\beta^{(2)}$ begins with 1 now proceeds in precisely the same way as for $X = B_n$ or $X = C_n$. $\square$

As a consequence of Lemma 6.3 and the fact that the first row of the columns $\beta^{(i)}$ must form a non-decreasing sequence since $\preceq$ holds between adjacent columns, it follows that all columns to the left of $\beta^{(2)}$ also begin with 1. That is, the tabloid corresponding to $c_1 c_{\beta^{(1)}} \cdots c_{\beta^{(m)}}$ has the following form:

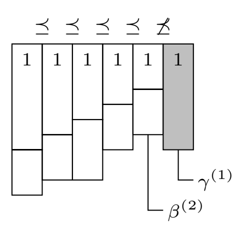


In the situation we are considering, $P(1\beta^{(1)})$ is a single column $\gamma^{(1)}$, so after rewriting $c_1 c_{\beta^{(1)}} \cdots c_{\beta^{(m)}} \to c_{\gamma^{(1)}} c_{\beta^{(2)}} \cdots c_{\beta^{(m)}}$, the corresponding tabloid has the form



If $\beta^{(2)} \preceq \gamma^{(1)}$, then $c_{\gamma^{(1)}} c_{\beta^{(2)}} \cdots c_{\beta^{(m)}}$ is irreducible and no further rewriting occurs. So assume $\beta^{(2)} \not\preceq \gamma^{(1)}$. Note that $\gamma^{(1)}\beta^{(2)} \ldots \beta^{(m)}$ is also a highest weight word.

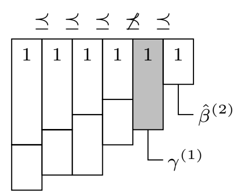
For $j = 2, \ldots, m$, define $\hat{\beta}^{(j)}$ to be the longest contiguous prefix of $\beta^{(j)}$ containing only symbols from $\mathcal{X}[1, n-1]$. Note that because $\beta^{(j+1)} \preceq \beta^{(j)}$, each symbol of $\beta^{(j+1)}$ is less than or equal to the symbol of $\beta^{(j)}$ in the same row. Thus the prefix $\hat{\beta}^{(j+1)}$ must be at least as long as the prefix $\hat{\beta}^{(j)}$, and so $\hat{\beta}^{(j+1)} \preceq \hat{\beta}^{(j)}$. So the situation is as follows, where the horizontal lines in each column indicate the end of $\hat{\beta}^{(j)}$:



Since $\beta^{(2)}$ begins with 1, by Lemma 6.2, $P(\gamma^{(1)}\beta^{(2)})$ has two columns, the rightmost of which is $\hat{\beta}^{(2)}$. Let $\gamma^{(2)}$ be the left column of $P(\gamma^{(1)}\beta^{(2)})$. So we have

$$c_{\gamma^{(1)}}c_{\beta^{(2)}} \ldots c_{\beta^{(m)}} \to c_{\hat{\beta}^{(2)}}c_{\gamma^{(2)}}c_{\beta^{(3)}} \ldots c_{\beta^{(m)}}$$

If $\beta^{(3)} \preceq \gamma^{(2)}$, the word $c_{\hat{\beta}^{(2)}}c_{\gamma^{(2)}}c_{\beta^{(3)}} \ldots c_{\beta^{(m)}}$ is irreducible. So suppose $\beta^{(3)} \not\preceq \gamma^{(2)}$. We claim $\gamma^{(2)}\beta^{(3)}$ is a highest weight word. This follows since $\hat{\beta}^{(2)}$ is a prefix of both $\gamma^{(2)}$ and $\beta^{(3)}$ (since it is a prefix of $\hat{\beta}^{(3)}$) and so commutes with both by the Commuting columns lemma 3.4. Thus $\hat{\beta}^{(2)}\gamma^{(2)}\beta^{(3)} =_{\mathrm{Pl}(X)} \gamma^{(2)}\beta^{(3)}\hat{\beta}^{(2)}$ and so, by Lemma 2.5, $\gamma^{(2)}\beta^{(3)}$ is highest weight. Thus, again by Lemma 6.2, $P(\gamma^{(2)}\beta^{(3)})$ has two columns, the rightmost of which is $\hat{\beta}^{(3)}$.



Continuing in this way, we inductively obtain a sequence of admissible columns $\gamma^{(2)}, \ldots, \gamma^{(k)}$ for some maximal $k \leq m$ such that the following hold for $j = 1, \ldots, k-1$:

- $\beta^{(j+1)} \not\preceq \gamma^{(j)}$.
- $\gamma^{(j)}\beta^{(j+1)}$ is highest weight. This follows since $\hat{\beta}^{(2)}, \ldots, \hat{\beta}^{(j)}$ are all prefixes of $\gamma^{(j)}$ and $\beta^{(j+1)}$, so commute with both by the Commuting columns lemma 3.4, and thus

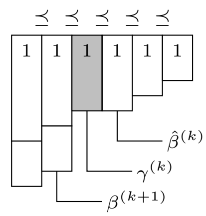
  $$\hat{\beta}^{(2)} \ldots \hat{\beta}^{(j)}\gamma^{(j)}\beta^{(j+1)} =_{\mathrm{Pl}(X)} \gamma^{(j)}\beta^{(j+1)}\hat{\beta}^{(2)} \ldots \hat{\beta}^{(j)}$$

  and so, by Lemma 2.5, $\gamma^{(2)}\beta^{(3)}$ is highest weight.
- $P(\gamma^{(j)}\beta^{(j+1)}) = \boxed{\gamma^{(j+1)}}\boxed{\hat{\beta}^{(j+1)}}$

Therefore rewriting proceeds as follows:

$$\begin{aligned}
&c_{\gamma^{(1)}}c_{\beta^{(2)}} \cdots c_{\beta^{(m)}} \\
\to\ &c_{\hat{\beta}^{(2)}}c_{\gamma^{(2)}}c_{\beta^{(3)}} \cdots c_{\beta^{(m)}} \\
\to\ &c_{\hat{\beta}^{(2)}}c_{\hat{\beta}^{(3)}}c_{\gamma^{(3)}}c_{\beta^{(4)}} \cdots c_{\beta^{(m)}} \\
&\vdots \\
\to\ &c_{\hat{\beta}^{(2)}} \cdots c_{\hat{\beta}^{(k)}}c_{\gamma^{(k)}}c_{\beta^{(k+1)}} \cdots c_{\beta^{(m)}},
\end{aligned} \tag{6.1}$$

The maximality of k means either that $k = m$ or $\beta^{(k+1)} \preceq \gamma^{(k)}$; in either case the word $c_{\hat{\beta}^{(2)}} \cdots c_{\hat{\beta}^{(k)}}c_{\gamma^{(k)}}c_{\beta^{(k+1)}} \cdots c_{\beta^{(m)}}$ is irreducible since $\beta^{(j+1)} \preceq \beta^{(j)}$ for all j. The corresponding tabloid is now a tableau of the form:



Thus far, we have analyzed the rewriting that occurs *at highest weight* when a tableau is left-multiplied by a generator. However, as we shall see, we can now deduce information about the rewriting that occurs in general.

Recall that a word $c_{\alpha^{(1)}} \cdots c_{\alpha^{(k)}} \in \Sigma^*$ represents the tabloid $\boxed{\alpha^{(k)}}\;\;\boxed{\alpha^{(1)}}$. As discussed following Lemma 3.1, we can think of applying the operators $\tilde{e}_i$ and $\tilde{f}_i$ to a tabloid. Thus we can think of applying them to words in Σ: the result of applying $\tilde{e}_i$ or $\tilde{f}_i$ to $c_{\alpha^{(1)}} \cdots c_{\alpha^{(k)}} \in \Sigma^*$ is $c_{\beta^{(1)}} \cdots c_{\beta^{(k)}}$, where $\boxed{\beta^{(k)}}\;\;\boxed{\beta^{(1)}}$ is the result of applying the operator to $\boxed{\alpha^{(k)}}\;\;\boxed{\alpha^{(1)}}$. Recall that the operators $\tilde{e}_i$ and $\tilde{f}_i$ preserve whether the $\preceq$ relation holds between adjacent columns. Thus the operators $\tilde{e}_i$ and $\tilde{f}_i$ preserve whether the $\preceq$ relation holds between adjacent subscripts of a word in Σ^*.

Now let $\beta^{(1)}, \ldots, \beta^{(m)}$ be admissible X columns, such that $\beta^{(i+1)} \preceq \beta^{(i)}$ for $i = 1, \ldots, m-1$ (that is, $\boxed{\beta^{(m)}}\;\;\boxed{\beta^{(1)}}$ is an X tableau), and let $x \in \mathcal{X}$. Let $\tilde{e}_{i_1}, \ldots, \tilde{e}_{i_k}$ be such that $w = \tilde{e}_{i_1} \cdots \tilde{e}_{i_k}(c_x c_{\beta^{(1)}} \cdots c_{\beta^{(m)}})$ is highest weight. The rewriting of the word w to normal form proceeds as described above, via a single left-to-right pass through the word. In particular, until the normal form is reached, there is exactly one pair of adjacent symbols where the relation $\preceq$ does not hold between the adjacent subscripts. Now apply $\tilde{f}_{i_k} \cdots \tilde{f}_{i_1}$ to every word in the sequence of rewriting; this gives a sequence of words starting at $c_x c_{\beta^{(1)}} \cdots c_{\beta^{(m)}}$. Furthemore, since the $\tilde{f}_i$ preserve whether the $\preceq$ relation holds between adjacent subscripts, there is exactly one place in each word where a rewriting rule can be applied. Since rewriting rules correspond to crystal isomorphisms, which are also preserved by the $\tilde{f}_i$, the rewriting rule that can be applied results in the next word in the sequence of rewriting. Thus we have a sequence of rewriting from $c_x c_{\beta^{(1)}} \cdots c_{\beta^{(m)}}$ that also proceeds via a single left-to-right pass.

The aim is now to show that a transducer can recognize the relation consisting of pairs (u, v), where $u, v \in \Sigma^*$ are irreducible and $c_x u \to^* v$, by essentially computing this rewriting. First, note that the transducer can check that the words on both tapes are irreducible: it simply stores the previously-read symbol in its state and checks that the previously-read and next symbols do not form the left-hand side of a rewriting rule. We will assume henceforth that the transducer is doing this and focus on how it computes the rewriting.

The computation is performed as follows. It reads the word $c_{\beta^{(1)}} \cdots c_{\beta^{(m)}}$ on its first tape. It stores one symbol in its state, starting with c_x. (Note that c_x is not read from input; the transducer is recognizing pairs (u, v) such that $c_x u \to^* v$.) At each later step, storing some c_α in its state, it reads the next symbol $c_{\beta^{(j)}}$, applies the rewriting rule $c_\alpha c_{\beta^{(j)}} \to c_\gamma c_\delta$, checks that the next symbol on its second tape is c_γ, and replaces the stored symbol c_α with c_δ. In the case where $c_\alpha c_{\beta^{(j)}}$ is not the left-hand side of a rewritin rule, the transducer simply checks that the next symbols on its second tape are c_α and $c_{\beta^{(j)}}$, then reads the rest of both tapes, checking that symbols on both tapes match. Note that this relies on rewriting proceeding as described above, via a single left-to-right pass.

In summary, we have proven the following lemma:

Lemma 6.4. *Let X be one of the types A_n, B_n, C_n, and D_n and let $\mathcal{X}$ be the corresponding alphabet from $\mathcal{A}_n$, $\mathcal{B}_n$, $\mathcal{C}_n$, or $\mathcal{D}_n$. Let Σ and T be alphabet and set of rewriting rules constucted for type X in Subsection 5.2. Let $x \in \mathcal{X}$. Let $L \subseteq \Sigma^*$ be the languages of irreducible words. Then the relation*

$$_{c_x}L = \left\{ (u, v) \in L \times L : c_x u =_{\mathrm{Pl}(X)} v \right\}$$

is recognized by a transducer. Furthermore, if (u, v) is a pair in this relation, then the lengths of u and v differ by at most 1.

6.3.2. G_2

Let $x \in \mathcal{G}_2$ and let $\beta^{(1)}, \cdots, \beta^{(m)}$ be admissible G_2 columns satisfying $\beta^{(i+1)} \preceq \beta^{(i)}$ for $i = 1, \cdots, m-1$ (that is, $\boxed{\beta^{(m)}}\ \cdots\ \boxed{\beta^{(1)}}$ is a tableau), such that $x\beta^{(1)} \cdots \beta^{(m)}$ is a highest-weight word. Recall that $x\beta^{(1)} \cdots \beta^{(h)}$ is a highest-weight word for all $h \leq m$ by Lemma 2.5. In particular, x is a highest-weight word and so $x = 1$.

We are going to analyze how $c_x c_{\beta^{(1)}} \cdots c_{\beta^{(m)}}$ rewrites to normal form. Like for A_n, B_n, C_n, and D_n, the aim is to prove that this rewriting involves a single left-to-right pass through the word. However, we require a fairly complicated analysis of cases, shown in Table 2. In the table, every possible admissible column is listed as a possibility for $\beta^{(1)}$. In those cases where we also have to consider $\beta^{(2)}$ or $\beta^{(3)}$, there are fewer possibilities because of the restriction $\beta^{(3)} \preceq \beta^{(2)} \preceq \beta^{(1)}$. Most of these cases are ruled out by the requirement that $x\beta^{(1)}\beta^{(2)}\beta^{(3)}$ is of highest weight. For example, the case where $\beta^{(1)} = 2$ and $\beta^{(2)} = 2$ is impossible, because

$$\rho_1(x\beta^{(1)}\beta^{(2)}\cdots) = \overbrace{+}^{x}\ \overbrace{-}^{\beta^{(1)}}\ \overbrace{-}^{\beta^{(2)}}\ \cdots = - \cdots;$$

All other cases listed as ‘not highest weight’ in Table 2 are ruled out in the same way, by considering either ρ_1 or ρ_2.

There are fourteen remaining cases in Table 2, but we reassure the reader that many of these quickly only result in one or two rewriting steps, and in the others the rewriting behaves in a straightforward way. Let us consider each of these cases in turn.

Table 2: Cases for left-multiplication in G_2

$x=1$	$\beta^{(1)}=1$	Case 1		
	$\beta^{(1)}=2$	$\beta^{(2)}=1$	Case 2	
		$\beta^{(2)}=2$	Not highest weight (ρ_1)	
		$\beta^{(2)}=12$	Case 3	
		$\beta^{(2)}=13$	Case 4	
		$\beta^{(2)}=23$	Not highest weight (ρ_1)	
		$\beta^{(2)}=20$	Not highest weight (ρ_1)	
		$\beta^{(2)}=2\overline{3}$	Not highest weight (ρ_1)	
	$\beta^{(1)}=3$	Not highest weight (ρ_2)		
	$\beta^{(1)}=0$	$\beta^{(2)}=1$	Case 5	
		$\beta^{(2)}=2$	$\beta^{(3)}=1$	Case 6
			$\beta^{(3)}=2$	Not highest weight (ρ_1)
			$\beta^{(3)}=12$	Case 7
			$\beta^{(3)}=13$	Case 8
			$\beta^{(3)}=23$	Not highest weight (ρ_1)
			$\beta^{(3)}=20$	Not highest weight (ρ_1)
		$\beta^{(2)}=3$	Not highest weight (ρ_2)	
		$\beta^{(2)}=12$	Case 9	
		$\beta^{(2)}=13$	Not highest weight (ρ_2)	
		$\beta^{(2)}=23$	Case 10	
		$\beta^{(2)}=20$	Not highest weight (ρ_1)	
	$\beta^{(1)}=\overline{3}$	Not highest weight (ρ_1)		
	$\beta^{(1)}=\overline{2}$	Not highest weight (ρ_2)		
	$\beta^{(1)}=\overline{1}$	Case 11		
	$\beta^{(1)}=12$	Case 12		
	$\beta^{(1)}=13$	Not highest weight (ρ_2)		
	$\beta^{(1)}=23$	Case 13		
	$\beta^{(1)}=00$	Case 14		
	$\beta^{(1)}=20$	Not highest weight (ρ_1)		
	$\beta^{(1)}=2\overline{3}$	Not highest weight (ρ_1)		
	$\beta^{(1)}=0\overline{3}$	Not highest weight (ρ_1)		
	$\beta^{(1)}=3\overline{3}$	Not highest weight (ρ_2)		
	$\beta^{(1)}=30$	Not highest weight (ρ_2)		
	$\beta^{(1)}=3\overline{2}$	Not highest weight (ρ_2)		
	$\beta^{(1)}=0\overline{2}$	Not highest weight (ρ_2)		
	$\beta^{(1)}=\overline{32}$	Not highest weight (ρ_1)		
	$\beta^{(1)}=\overline{31}$	Not highest weight (ρ_1)		
	$\beta^{(1)}=\overline{21}$	Not highest weight (ρ_2)		

- *Case 1.* $\beta^{(1)} = 1$. Then $\beta^{(1)} \preceq x$ and so no rewriting occurs: the word $c_x c_{\beta^{(1)}} \cdots c_{\beta^{(m)}}$ is in normal form.

- *Cases 2–4.* $\beta^{(1)} = 2$ and $\beta^{(2)} \in \{1, 12, 13\}$. Now, since $\beta^{(j+1)} \preceq \beta^{(j)}$ for all j, the columns $\beta^{(2)}, \ldots, \beta^{(m)}$ consist of zero or more columns 1, followed by zero or more columns 13, followed by zero or more columns 12. Notice that this subsumes the three possibilities for $\beta^{(2)}$. Note that $P(x\beta^{(1)}) = \boxed{12}$, and $P(121) = \boxed{12}\boxed{1}$, and $P(1213) = \boxed{1}\boxed{1}\boxed{1}$, as can be seen from Table 1 (see page 26).

  When there is at least one column 13, rewriting begins

$$\begin{aligned} c_1 c_2 c_1^p c_{13}^q c_{12}^r &\rightarrow c_{12} c_1^p c_{13}^q c_{12}^r \\ &\rightarrow^* c_1^p c_{12} c_{13}^q c_{12}^r \\ &\rightarrow \begin{cases} c_1^p c_{12}^{r+1} & \text{if } q = 0, \\ c_1^{p+3} c_{13}^{q-1} c_{12}^r & \text{if } q \geq 1. \end{cases} \end{aligned}$$

  In either case, the final word is in normal form since $1 \preceq 13$ and $1 \preceq 12$.

- *Case 5.* $\beta^{(1)} = 0$ and $\beta^{(2)} = 1$. Then $P(x\beta^{(1)}) = \boxed{1}$. Since $1 \preceq 1$, the rewriting to normal form is simply

$$c_1 c_0 c_1 c_{\beta^{(3)}} \cdots c_{\beta^{(m)}} \rightarrow c_1 c_1 c_{\beta^{(3)}} \cdots c_{\beta^{(m)}}.$$

- *Cases 6–8.* $\beta^{(1)} = 0$, $\beta^{(2)} = 2$, and $\beta^{(3)} \in \{1, 12, 13\}$. Since $\beta^{(j+1)} \preceq \beta^{(j)}$ for all j, the columns $\beta^{(3)}, \ldots, \beta^{(m)}$ consist of zero or more columns 1, followed by zero or more columns 13, followed by zero or more columns 12. Since $P(1\beta^{(1)}) = \boxed{1}$ and $P(1\beta^{(2)}) = \boxed{12}$, rewriting proceeds in one of two ways, similarly to case 2. If there is a column 13 present, rewriting proceeds

$$\begin{aligned} c_1 c_0 c_2 c_1 \cdots c_1 c_{13} c_{\beta^{(k)}} \cdots c_{\beta^{(m)}} &\rightarrow c_1 c_2 c_1 \cdots c_1 c_{13} c_{\beta^{(k)}} \cdots c_{\beta^{(m)}} \\ &\rightarrow c_{12} c_1 \cdots c_1 c_{13} c_{\beta^{(k)}} \cdots c_{\beta^{(m)}} \\ &\rightarrow^* c_1 \cdots c_1 c_{12} c_{13} c_{\beta^{(k)}} \cdots c_{\beta^{(m)}} \\ &\rightarrow c_1 \cdots c_1 c_1 c_1 c_1 c_{\beta^{(k)}} \cdots c_{\beta^{(m)}}. \end{aligned}$$

  This word is in normal form since, regardless of whether $\beta^{(k)}$ is 12 or 13, we have $1 \preceq \beta^{(k)}$. When there is no column 13, the columns 1 are followed immediately by columns 12, and so rewriting begins

$$\begin{aligned} c_1 c_0 c_2 c_1 \cdots c_1 c_{12} c_{\beta^{(k)}} \cdots c_{\beta^{(m)}} &\rightarrow c_1 c_2 c_1 \cdots c_1 c_{12} c_{\beta^{(k)}} \cdots c_{\beta^{(m)}} \\ &\rightarrow c_{12} c_1 \cdots c_1 c_{12} c_{\beta^{(k)}} \cdots c_{\beta^{(m)}} \\ &\rightarrow^* c_1 \cdots c_1 c_{12} c_{12} c_{\beta^{(k)}} \cdots c_{\beta^{(m)}}, \end{aligned}$$

  whichs is in normal form.

- *Case 9.* $\beta^{(1)} = 0$ and $\beta^{(2)} = 12$. Then $P(x\beta^{(1)}) = \boxed{1}$. Since $12 \preceq 1$, the rewriting to normal form is simply

$$c_1 c_0 c_{12} c_{\beta^{(3)}} \cdots c_{\beta^{(m)}} \to c_1 c_{12} c_{\beta^{(3)}} \cdots c_{\beta^{(m)}}.$$

- *Case 10.* $\beta^{(1)} = 0$ and $\beta^{(2)} = 23$. Now, since $\beta^{(j+1)} \preceq \beta^{(j)}$ for all j, the remaining columns $\beta^{(3)}, \ldots, \beta^{(m)}$ consist of zero or more columns 23, zero or more columns 13, and zero or more columns 12. Note that $P(x\beta^{(1)}) = \boxed{1}$, and $P(1\beta^{(2)}) = \boxed{1}\boxed{1}$, as can be seen from Table 1. So rewriting proceeds as follows:

$$\begin{aligned} c_1 c_0 c_{23} c_{23} \cdots c_{23} c_{\beta^{(k)}} \cdots c_{\beta^{(m)}} &\to c_1 c_{23} c_{23} \cdots c_{23} c_{\beta^{(k)}} \cdots c_{\beta^{(m)}} \\ &\to c_1 c_1 c_{23} \cdots c_{23} c_{\beta^{(k)}} \cdots c_{\beta^{(m)}} \\ &\to^* c_1 c_1 c_1 \cdots c_1 c_{\beta^{(k)}} \cdots c_{\beta^{(m)}}. \end{aligned}$$

  Regardless of whether $\beta^{(k)}$ is 13 or 12, we have $\beta^{(k)} \preceq 1$, so this word is in normal form. Note that there is exactly one symbol c_1 in the final word for each symbol c_{23} in the initial word.

- *Case 11.* $\beta^{(1)} = \overline{1}$. Since $P(1\overline{1}) = \varepsilon$, as can be seen from Table 1, the rewriting to normal form is simply

$$c_1 c_{\overline{1}} c_{\beta^{(3)}} \cdots c_{\beta^{(m)}} \to c_{\beta^{(3)}} \cdots c_{\beta^{(m)}}.$$

- *Case 12.* $\beta^{(1)} = 12$. Since $12 \preceq 1$, the word $c_1 c_{12} c_{\beta^{(2)}} \cdots c_{\beta^{(m)}}$ is in normal form.

- *Case 13.* $\beta^{(1)} = 23$. Since $\beta^{(j+1)} \preceq \beta^{(j)}$ for all j, the remaining columns $\beta^{(2)}, \ldots, \beta^{(m)}$ consist of zero or more columns 23, zero or more columns 13, and zero or more columns 12. Note that $P(x\beta^{(1)}) = \boxed{1}\boxed{1}$, as can be seen from Table 1. So rewriting proceeds as follows:

$$\begin{aligned} c_1 c_{23} c_{23} \cdots c_{23} c_{\beta^{(k)}} \cdots c_{\beta^{(m)}} &\to c_1 c_1 c_{23} \cdots c_{23} c_{\beta^{(k)}} \cdots c_{\beta^{(m)}} \\ &\to^* c_1 c_1 c_1 \cdots c_1 c_{\beta^{(k)}} \cdots c_{\beta^{(m)}}. \end{aligned}$$

  Regardless of whether $\beta^{(k)}$ is 13 or 12, we have $\beta^{(k)} \preceq 1$, so this word is in normal form.

- *Case 14.* $\beta^{(1)} = 00$. Since $\beta^{(2)} \preceq \beta^{(1)}$, it follows that $\beta^{(2)}$ is either 13 or 12 (note that $00 \not\preceq 00$). Since $P(x\beta^{(1)}) = \boxed{1}$, as can be seen from Table 1, the rewriting to normal form is simply

$$c_1 c_{00} c_{\beta^{(2)}} \cdots c_{\beta^{(m)}} \to c_1 c_{\beta^{(2)}} \cdots c_{\beta^{(m)}}.$$

  Regardless of whether $\beta^{(2)}$ is 13 or 12, we have $\beta^{(2)} \preceq 1$, so this word is in normal form.

This completes the case analysis. Note that in each case, the lengths of $c_x c_{\beta^{(1)}} \cdots c_{\beta^{(m)}}$ and its corresponding normal form differ by at most 2. (The maximum difference 2 occurs in cases 5–7 and 10.)

As in the discussion before Lemma 6.4, the way rewriting proceeds at highest weight is mirrored in how it proceeds in general for type G_2. Thus, with the analysis above, we can prove the following analogue of Lemma 6.4:

Lemma 6.5. *Let Σ and T be alphabet and set of rewriting rules constucted for type G_2 in Subsection 5.2. Let $x \in \mathcal{G}_2$. Let $L \subseteq \Sigma^*$ be the languages of irreducible words. Then the relation*

$$_{c_x}L = \left\{ (u, v) \in L \times L : c_x u =_{\mathrm{Pl}(G_2)} v \right\}$$

is recognized by an transducer. Furthermore, if (u, v) is a pair in this relation, then the lengths of u and v differ by at most 1.

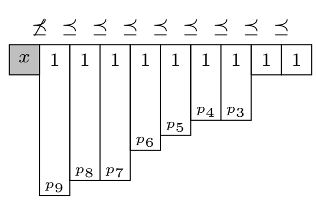
6.4. Right-multiplication by transducer

We now turn our attention to right-multiplication. Unlike left-multiplication, the cases A_n, B_n, C_n, and D_n are sufficiently different that we have to consider them separately.

6.4.1. A_n

Let $\beta^{(1)}, \ldots, \beta^{(m)}$ be admissible A_n columns and let $x \in \mathcal{A}_n$ be such that $\beta^{(i+1)} \preceq \beta^{(i)}$ for $i = 1, \ldots, m-1$ (that is, $\boxed{\beta^{(m)}}\ \cdots\ \boxed{\beta^{(1)}}$ is an A_n tableau), and such that $\beta^{(1)} \cdots \beta^{(m)} x$ is a highest-weight word. We are going to examine how the corresponding word over Σ (that is, $c_{\beta^{(1)}} \cdots c_{\beta^{(m)}} c_x$) rewrites to an irreducible word. The aim is to prove that this rewriting involves a single right-to-left pass through the word.

First, note that since the prefix $\beta^{(1)} \cdots \beta^{(m)}$ is a highest-weight word by Lemma 2.5, each column $\beta^{(i)}$ is of the form $1 \cdots p_i$ for some $p_i \in \mathcal{A}[1, n]$ and $p_{i+1} \geq p_i$ for $i = 1, \ldots, m-1$ by Lemma 3.2. That is, the tabloid corresponding to the word $\beta^{(1)} \cdots \beta^{(m)} x$ is of the form:



The assumption that $\beta^{(1)} \cdots \beta^{(m)} x$ is of highest weight puts a restriction on x, as the following lemma shows:

Lemma 6.6. *Either $x = 1$, or $x = p_k + 1$ for some $k \in \{1, \ldots, m\}$ such that $p_k < n$.*

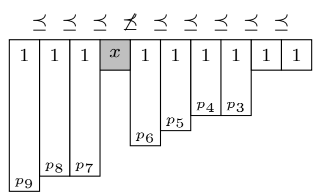
Proof. Suppose that $x \neq 1$ and $x \neq p_k + 1$ for all k. Then for each i, either $\rho_{x-1}(\beta^{(k)}) = \varepsilon$ (when $x-1 > p_k$) or $\rho_{x-1}(\beta^{(k)}) = +- = \varepsilon$ (when $x-1 < p_k$), and so $\rho_{x-1}(\beta^{(1)} \cdots \beta^{(m)} x) = -$, contradicting the assumption of highest weight. $\square$

We consider the cases $x = p_k + 1$ and $x = 1$ separately.

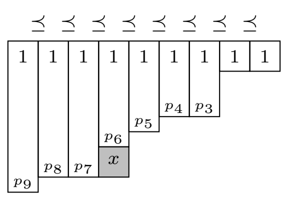
First, suppose $x = p_k + 1$, and assume that k is maximal with this property. Then for $j > k$, we have $x \not\preceq \beta^{(j)}$ and the symbol x appears in $\beta^{(j)}$ and so by the Commuting columns lemma 3.4, $P(\begin{array}{|c|c|}\hline x & \beta^{(j)} \\ \hline\end{array}) = \begin{array}{|c|c|}\hline \beta^{(j)} & x \\ \hline\end{array}$, while $P(\begin{array}{|c|c|}\hline x & \beta^{(k)} \\ \hline\end{array}) = \begin{array}{|c|}\hline \beta^{(k)}x \\ \hline\end{array}$. That is, there are rewriting rules $c_{\beta^{(j)}} c_x \to c_x c_{\beta^{(j)}}$ and $c_{\beta^{(k)}} c_x \to c_{\beta^{(k)}x}$. Further, $\beta^{(k)}x \preceq \beta_{(k+1)}$. Therefore rewriting to normal form proceeds as follows:

$$\begin{aligned}
& c_{\beta^{(1)}} \cdots c_{\beta^{(m)}} c_x \\
\to\ & c_{\beta^{(1)}} \cdots c_{\beta^{(m-1)}} c_x c_{\beta^{(m)}} \\
& \vdots \\
\to\ & c_{\beta^{(1)}} \cdots c_{\beta^{(k)}} c_x c_{\beta^{(k+1)}} \cdots c_{\beta^{(m)}} \\
\to\ & c_{\beta^{(1)}} \cdots c_{\beta^{(k)}x} c_{\beta^{(k+1)}} \cdots c_{\beta^{(m)}}.
\end{aligned}$$

Thus in the case $k = 6$, in terms of the tabloid the column x commutes past the columns $\beta^{(j)}$ with $j > 6$, resulting in a tabloid of the following form:



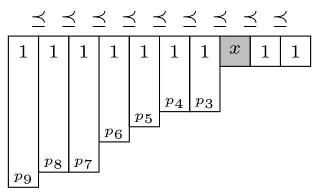
The final rewriting step appends x to the bottom of the column $\beta^{(6)}$, giving a tableau of the following form:



Now consider the other case, when $x = 1$. Define k to be maximal such that $\beta^{(k)} = 1$. Then for $j > k$, we have $x \not\preceq \beta^{(j)}$ and the symbol x appears in $\beta^{(j)}$ and so by the Commuting columns lemma 3.4, $P(\begin{array}{|c|c|}\hline x & \beta^{(j)} \\ \hline\end{array}) = \begin{array}{|c|c|}\hline \beta^{(j)} & x \\ \hline\end{array}$. That is, there are rewriting rules $c_{\beta^{(j)}} c_x \to c_x c_{\beta^{(j)}}$. Further, $x \preceq \beta^{(k)}$. Therefore rewriting to normal form proceeds as follows:

$$
\begin{aligned}
& c_{\beta^{(1)}} \cdots c_{\beta^{(m)}} c_x \\
\to\ & c_{\beta^{(1)}} \cdots c_{\beta^{(m-1)}} c_x c_{\beta^{(m)}} \\
& \vdots \\
\to\ & c_{\beta^{(1)}} \cdots c_{\beta^{(k)}} c_{\beta^{(k+1)}} c_x c_{\beta^{(k+2)}} \cdots c_{\beta^{(m)}} \\
\to\ & c_{\beta^{(1)}} \cdots c_{\beta^{(k)}} c_x c_{\beta^{(k+1)}} \cdots c_{\beta^{(m)}}.
\end{aligned}
$$

Thus in the case $k = 2$, in terms of the tabloid the column x commutes past the columns $\beta^{(j)}$ with $j > 2$, giving a tableau of the following form:



Note that in both cases the length of the normal form word differs from m by at most 1.

As in the discussion before Lemma 6.4, the way rewriting proceeds at highest weight is mirrored in how it proceeds in general. That is, rewriting can be carried out by a single right-to-left pass through the word. The aim is now to show that a transducer can recognize the relation consisting of pairs (u, v), where $u, v \in \Sigma^*$ are irreducible and $uc_x \to^* v$, by essentially computing this rewriting. To see this, consider a transducer that reads its input tapes right-to-left: such a transducer can carry out the rewriting in a way symmetrical to that described in the discussion before Lemma 6.4. Since the class of rational relations is closed under reversal [78, p. 65–66], it follows we have proven the following analogue of Lemma 6.4 for right-multiplication in type A_n:

Lemma 6.7. *Let Σ and T be the alphabet and set of rewriting rules constructed for type A_n in Subsection 5.2. Let $x \in \mathcal{A}_n$. Let $L \subseteq \Sigma^*$ be the languages of irreducible words. Then the relation*

$$L_{c_x} = \left\{ (u, v) \in L \times L : uc_x =_{\mathrm{Pl}(A_n)} v \right\}$$

is recognized by an transducer. Furthermore, if (u, v) is a pair in this relation, then the lengths of u and v differ by at most 1.

6.4.2. C_n

Let $\beta^{(1)}, \ldots, \beta^{(m)}$ be admissible C_n columns and let $x \in \mathcal{C}_n$ be such that $\beta^{(i+1)} \preceq \beta^{(i)}$ for $i = 1, \ldots, m-1$ (that is, $\boxed{\beta^{(m)}}\cdots\boxed{\beta^{(1)}}$ is a C_n tableau), and such that $\beta^{(1)} \ldots \beta^{(m)} x$ is a highest-weight word. As we did for type A_n, we are going to examine how the corresponding word over Σ (that is, $c_{\beta^{(1)}} \cdots c_{\beta^{(m)}} c_x$)

rewrites to an irreducible word. Again, the aim is to prove that this rewriting involves a single right-to-left pass through the word.

Since $\beta^{(1)} \ldots \beta^{(m)}$ is a highest-weight word by Lemma 2.5, each column $\beta^{(i)}$ is of the form $1 \cdots p_i$ for some $p_i \in \mathcal{C}[1,n]$, and $p_{i+1} \geq p_i$ for $i = 1, \ldots, m-1$ by Lemma 3.2.

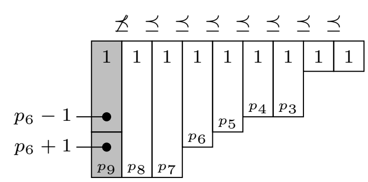
The reasoning will proceed in a similar way to the A_n case, except that there is the additional possibility that x may be $\overline{p_k}$, as shown in the following lemma:

Lemma 6.8. *Either $x = 1$, or $x = p_k + 1$ for some $k \in \{1, \ldots, m\}$ such that $p_k < n$, or $x = \overline{p_k}$ for some $k \in \{1, \ldots, m\}$.*

Proof. Suppose that $x \neq 1$ and $x \neq p_k + 1$ and $x \neq \overline{p_k}$ for all k. If $x \in \mathcal{C}[1,n]$, then the same contradiction arises as in the proof of Lemma 6.6. If $x \in \mathcal{C}[\overline{n}, \overline{1}]$, then for each k, either $\rho_{\overline{x}}(\beta^{(k)}) = \varepsilon$ (when $\overline{x} > p_k$) or $\rho_{\overline{x}}(\beta^{(k)}) = +- = \varepsilon$ (when $\overline{x} < p_k$), and so $\rho_{\overline{x}}(\beta^{(1)} \cdots \beta^{(m)} x) = -$, contradicting the assumption of highest weight. ☐

If $x = 1$ or $x = p_k + 1$, then the rewriting proceeds in the same way as in the A_n case. So suppose $x = \overline{p_k}$. If $p_k > 1$, we will assume that k is minimal with this property; if $p_k = 1$, we will assume that k is maximal with this property.

Now, $P(\boxed{\overline{p_k}}\boxed{\beta^{(m)}}) = \boxed{1 \cdots (p_k - 1)(p_k + 1) \cdots p_m}$, since $\beta^{(m)}\overline{p_k} =_{\mathcal{R}_5^{C_n}} 1 \cdots (p_k - 1)(p_k + 1) \cdots p_m$. Pictorially (using $k = 6$ as an example), we have:



Now, for $m > j > k$, we have

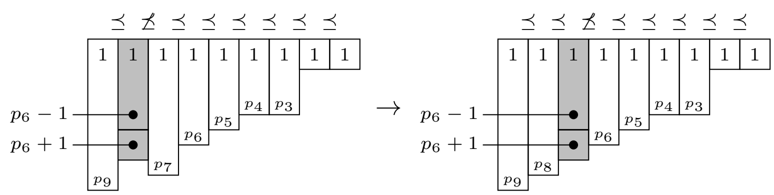
$$\begin{aligned}
&\beta^{(j)} 1 \cdots (p_k - 1)(p_k + 1) \cdots p_{j+1} \\
&= \underbrace{1 \cdots p_j}\,\underbrace{1 \cdots (p_k - 1)(p_k + 1) \cdots p_j}(p_j + 1) \cdots p_{j+1} \\
&\qquad\text{[by the Commuting columns lemma 3.4]} \\
&=_{\mathrm{Pl}(C_n)} \overbrace{1 \cdots (p_k - 1)(p_k + 1) \cdots p_j}\,\overbrace{1 \cdots p_j}(p_j + 1) \cdots p_{j+1} \\
&= 1 \cdots (p_k - 1)(p_k + 1) \cdots p_j \beta^{(j+1)}.
\end{aligned}$$

Write $\beta_*^{(j)}$ for $\beta^{(j)}$ with the symbol p_k deleted. Then we have $P(\boxed{\beta_*^{(j+1)}}\boxed{\beta^{(j)}}) = \boxed{\beta^{(j+1)}}\boxed{\beta_*^{(j)}}$ for all $j = k+1, \ldots, m-1$. (Note that when $p_k = 1$, we know from the maximality of k that $\beta^{(j)} \neq 1$.) Thus we have rewriting rules $c_{\beta^{(j)}} c_{\beta_*^{(j+1)}} \to c_{\beta_*^{(j)}} c_{\beta^{(j+1)}}$.

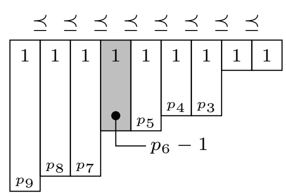
When $p_k \neq 1$, we have $\beta_*^{(k)} = 1 \cdots (p_k - 1)$. Thus $\beta_*^{(k)} \preceq \beta^{(k-1)}$ since by the minimality of k we have $p_{k-1} < p_k$. Thus in this case rewriting to normal form proceeds as follows:

$$\begin{aligned}
&c_{\beta^{(1)}} \cdots c_{\beta^{(m)}} c_{\overline{p_k}} \\
\to\ &c_{\beta^{(1)}} \cdots c_{\beta^{(m-1)}} c_{\beta_*^{(m)}} \\
\to\ &c_{\beta^{(1)}} \cdots c_{\beta_*^{(m-1)}} c_{\beta^{(m)}} \\
&\vdots \\
\to\ &c_{\beta^{(1)}} \cdots c_{\beta^{(k-1)}} c_{\beta_*^{(k)}} c_{\beta^{(k+1)}} \cdots c_{\beta^{(m)}}.
\end{aligned}$$

In the case $k = 6$ with $p_k \neq 1$, in terms of the tabloid the 'gap' in the columns moves from left to right through the tabloid:



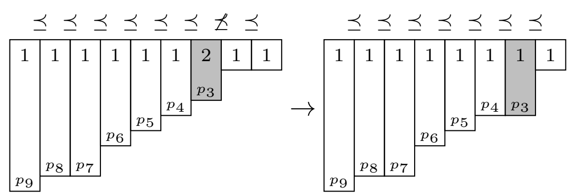
The rewriting continues until the 'gap' reaches the column $\beta^{(k)}$, at which point a tableau is obtained:



When $p_k = 1$, we have $\beta_*^{(k+1)} = 2 \cdots p_{k+1}$ (and we know $p_{k+1} > 1$ by the maximality of k) and so $P(\boxed{\beta_*^{(k+1)}}\boxed{\beta^{(k)}}) = \boxed{\beta^{(k+1)}}$ and so there is a rewriting rule $c_{\beta^{(k)}} c_{\beta_*^{(k+1}} \to c_{\beta^{(k+1)}}$. Thus in this case rewriting to normal form proceeds as follows:

$$\begin{aligned}
&c_{\beta^{(1)}} \cdots c_{\beta^{(m)}} c_{\overline{p_k}} \\
\to\ &c_{\beta^{(1)}} \cdots c_{\beta^{(m-1)}} c_{\beta_*^{(m)}} \\
\to\ &c_{\beta^{(1)}} \cdots c_{\beta_*^{(m-1)}} c_{\beta^{(m)}} \\
&\vdots \\
\to\ &c_{\beta^{(1)}} \cdots c_{\beta^{(k-1)}} c_{\beta^{(k)}} c_{\beta_*^{(k+1)}} \cdots c_{\beta^{(m)}} \\
\to\ &c_{\beta^{(1)}} \cdots c_{\beta^{(k-1)}} c_{\beta^{(k+1)}} \cdots c_{\beta^{(m)}}.
\end{aligned}$$

In the case $k = 2$ with $p_k = 1$, in terms of the tabloid the 'gap' in the columns moves from left to right through the tabloid, just as in the other case, but then there is a final rewriting step:



Note that in each case the length of the normal form word differs from m by at most 1.

As in the discussion before Lemma 6.7, the way rewriting proceeds at highest weight is mirrored in how it proceeds in general and so, using the same argument, we have proven the following analogue of Lemma 6.7 for type C_n:

Lemma 6.9. *Let Σ and T be the alphabet and set of rewriting rules constructed for type C_n in Subsection 5.2. Let $x \in \mathcal{C}_n$. Let $L \subseteq \Sigma^*$ be the languages of irreducible words. Then the relation*

$$L_{c_x} = \left\{ (u, v) \in L \times L : uc_x =_{\mathrm{Pl}(C_n)} v \right\}$$

is recognized by an transducer. Furthermore, if (u, v) is a pair in this relation, then the lengths of u and v differ by at most 1.

6.4.3. B_n

Let $\beta^{(1)}, \ldots, \beta^{(m)}$ be admissible B_n columns and let $x \in \mathcal{B}_n$ be such that $\beta^{(i+1)} \preceq \beta^{(i)}$ for $i = 1, \ldots, m-1$ (that is, $\boxed{\beta^{(m)}}\cdots\boxed{\beta^{(1)}}$ is a B_n tableau), and such that $\beta^{(1)} \ldots \beta^{(m)}x$ is a highest-weight word. As we did for types A_n and C_n, we are going to examine how the corresponding word over Σ (that is, $c_{\beta^{(1)}} \cdots c_{\beta^{(m)}} c_x$) rewrites to an irreducible word. In fact, the analysis reduces almost entirely to the C_n case: there is only one easy extra case. Again, the aim is to prove that this rewriting involves a single right-to-left pass through the word.

Since $\beta^{(1)} \ldots \beta^{(m)}$ is a highest-weight tableau, each column $\beta^{(i)}$ is of the form $1 \cdots p_i$ for some $p_i \in \mathcal{B}[1, n]$, and $p_{i+1} \geq p_i$ for $i = 1, \ldots, m-1$ by Lemma 3.2.

Lemma 6.10. *One of the following holds:*

1. $x = 1$*;*
2. $x = p_k + 1$ *for some* $k \in \{1, \ldots, m\}$ *such that* $p_k < n$*;*
3. $x = 0$ *(only if* $p_m = n$*);*
4. $x = \overline{p_k}$ *for some* $k \in \{1, \ldots, m\}$*.*

Proof. Suppose that $x \neq 1$, $x \neq p_k + 1$, $x \neq 0$, and $x \neq \overline{p_k}$ for all k. If $x \in \mathcal{B}_n[1, n]$, then the same contradiction arises as in the proof of Lemma 6.6. If $x \in \mathcal{B}_n[\overline{n}, \overline{1}]$, then the same contradiction arises as in the proof of Lemma 6.8.

Finally, suppose $x = 0$. If $p_m \neq n$, then $\rho_n(\beta^{(k)}) = \varepsilon$ for each k and so $\rho_n(\beta^{(1)} \cdots \beta^{(m)} 0) = -+$, contradicting the assumption of highest weight. $\square$

If $x = 1$ or $x = p_k + 1$, then the rewriting proceeds in the same way as in the A_n case, and if $x = \overline{p_k}$, then the rewriting proceeds in the same way as the C_n case. So suppose $x = 0$. Then $p_m = n$ and so $\beta^{(m)}0 = 1 \cdots n0 =_{\mathcal{R}_5^{B_n}} 1 \cdots n = \beta^{(m)}$; thus $P(\boxed{0}\boxed{\beta^{(m)}}) = \boxed{\beta^{(m)}}$. So there is a rewriting rule $c_{\beta^{(m)}}c_0 = c_{\beta^{(m)}}$ and so rewriting to normal form is as follows:

$$c_{\beta^{(1)}} \cdots c_{\beta^{(m)}} c_0 \to c_{\beta^{(1)}} \cdots c_{\beta^{(m)}}.$$

Note that in each case the length of the normal form word differs from m by at most 1.

As in the discussion before Lemma 6.7, the way rewriting proceeds at highest weight is mirrored in how it proceeds in general and so, using the same argument, we have proven the following analogue of Lemma 6.7 for type B_n:

Lemma 6.11. *Let Σ and T be the alphabet and set of rewriting rules constructed for type B_n in Subsection 5.2. Let $x \in \mathcal{B}_n$. Let $L \subseteq \Sigma^*$ be the languages of irreducible words. Then the relation*

$$L_{c_x} = \left\{ (u, v) \in L \times L : uc_x =_{\mathrm{Pl}(B_n)} v \right\}$$

is recognized by an transducer. Furthermore, if (u, v) is a pair in this relation, then the lengths of u and v differ by at most 1.

6.4.4. D_n

Let $\beta^{(1)}, \ldots, \beta^{(m)}$ be admissible D_n columns and let $x \in \mathcal{D}_n$ be such that $\beta^{(i+1)} \preceq \beta^{(i)}$ for $i = 1, \ldots, m-1$ (that is, $\boxed{\beta^{(m)}}\cdots\boxed{\beta^{(1)}}$ is a D_n tableau), and such that $\beta^{(1)} \ldots \beta^{(m)} x$ is a highest-weight word. As for the other types, we are going to examine how the corresponding word over Σ (that is, $c_{\beta^{(1)}} \cdots c_{\beta^{(m)}} c_x$) rewrites to an irreducible word. As before, the aim is to prove that this rewriting involves a single right-to-left pass through the word.

Since $\beta^{(1)} \ldots \beta^{(m)}$ is a highest-weight word by Lemma 2.5, each column $\beta^{(i)}$ is of the form $1 \cdots p_i$ for some $p_i \in \mathcal{D}[1, n] \cup \mathcal{D}[1, \overline{n}]$, and $p_{i+1} \geq p_i$ for $i = 1, \ldots, m-1$ by Lemma 3.2.

Lemma 6.12. *One of the following holds:*

1. $x = 1$*;*
2. $x = p_k + 1$ *for some* $k \in \{1, \ldots, m\}$ *such that* $p_k < n - 1$*;*
3. $x = n$ *(only if* $\beta^{(k)} = 1 \cdots (n-1)$ *for some* k *or* $\beta^{(m)} = 1 \cdots (n-1)\overline{n}$*);*
4. $x = \overline{n}$ *(only if* $\beta^{(k)} = 1 \cdots (n-1)$ *for some* k *or* $\beta^{(m)} = 1 \cdots n$*);*
5. $x = \overline{p_k}$ *for some* $k \in \{1, \ldots, m\}$ *such that* $p_k \leq n - 1$.

Proof. Suppose that $x \neq 1$, $x \neq p_k + 1$, $x \neq n$, $x \neq \overline{n}$, and $x \neq \overline{p_k}$ for all k. If $x \in \mathcal{D}_n[1, n-1]$ then the same contradiction arises as in the proof of Lemma 6.6. If $x \in \mathcal{D}_n[\overline{n-1}, \overline{1}]$, then the same contradiction arises as in the proof of Lemma 6.8.

Now, suppose $x = n$. If $\beta^{(k)} \neq 1 \cdots (n-1)$ for all k and $\beta^{(m)} \neq 1 \cdots (n-1)\overline{n}$, then $\rho_n(\beta^{(j)}) = \varepsilon$ (when $\beta^{(j)} = 1 \cdots p_j$ for $p_j \leq n-2$) and $\rho_{n-1}(\beta^{(j)}) =$

$+- = \varepsilon$ (when $\beta^{(j)} = 1 \cdots n$) and so $\rho_{n-1}(\beta^{(1)} \cdots \beta^{(m)} n) = -$, contradicting the assumption of highest weight.

Similar reasoning shows that $x = \overline{n}$ only if $\beta^{(k)} = 1 \cdots (n-1)$ for some k or $\beta^{(m)} = 1 \cdots n$, using ρ_n to get the contradictions. $\square$

If cases (1) or (2) of Lemma 6.12 hold, or case (3) holds with $\beta^{(k)} = 1 \cdots (n-1)$ for some k, then the rewriting proceeds in the same way as in the A_n case. If case (5) holds, or case (4) holds with $\beta^{(m)} = 1 \cdots n$, then the rewriting proceeds in the same way as the C_n case.

We thus have two remaining cases: case (3) with $x = n$ and $\beta^{(m)} = 1 \cdots (n-1)\overline{n}$, or case (4) with $x = \overline{n}$ and $\beta^{(k)} = 1 \cdots (n-1)$ for some k.

Suppose $x = \overline{n}$ and $\beta^{(k)} = 1 \cdots (n-1)$ for some k. In the case where $\beta^{(m)} = 1 \cdots n$, rewriting proceeds as in the C_n case. So, by the definition of $\preceq$, either $\beta^{(m)} = 1 \cdots (n-1)$ or $\beta^{(m)} = 1 \cdots (n-1)\overline{n}$. Consider these cases separately:

1. $\beta^{(m)} = 1 \cdots (n-1)$. So $P(\beta^{(m)}\overline{n})$ is the single column $\beta^{(m)}\overline{n}$ and so there is a rewriting rule $c_{\beta^{(m)}} c_{\overline{n}} \to c_{\beta^{(m)}\overline{n}}$ and so rewriting to normal form proceeds as follows:
$$c_{\beta^{(1)}} \cdots c_{\beta^{(m)}} c_{\overline{n}} \to c_{\beta^{(1)}} \cdots c_{\beta^{(m)}\overline{n}}.$$
2. $\beta^{(m)} = 1 \cdots (n-1)\overline{n}$. Then rewriting proceeds in the same way as in the A_n case, but with $\overline{n}$ in place of n.

Finally, suppose $x = n$ and $\beta^{(m)} = 1 \cdots (n-1)\overline{n}$. It is easy to see that rewriting is symmetric to the case C_n where $x = \overline{n}$ and $\beta^{(m)} = 1 \cdots n$.

Note that in each case the length of the normal form word differs from m by at most 1.

As in the discussion before Lemma 6.7, the way rewriting proceeds at highest weight is mirrored in how it proceeds in general and so, using the same argument, we have proven the following analogue of Lemma 6.7 for type D_n:

Lemma 6.13. *Let Σ and T be the alphabet and set of rewriting rules constructed for type D_n in Subsection 5.2. Let $x \in \mathcal{D}_n$. Let $L \subseteq \Sigma^*$ be the languages of irreducible words. Then the relation*
$$L_{c_x} = \left\{ (u, v) \in L \times L : uc_x =_{\mathrm{Pl}(D_2)} v \right\}$$
is recognized by an transducer. Furthermore, if (u, v) is a pair in this relation, then the lengths of u and v differ by at most 1.

6.4.5. G_2

Let $\beta^{(1)}, \ldots, \beta^{(m)}$ be admissible G_2 columns and let $x \in \mathcal{G}_2$ be such that $\beta^{(i+1)} \preceq \beta^{(i)}$ for $i = 1, \ldots, m-1$ (that is, $\boxed{\beta^{(m)}} \cdots \boxed{\beta^{(1)}}$ is a G_2 tableau), and such that $\beta^{(1)} \ldots \beta^{(m)} x$ is a highest-weight word. Since $\beta^{(1)} \ldots \beta^{(m)}$ is a highest-weight word, each column $\beta^{(i)}$ is either 1 or 12 by Lemma 3.2. Notice that, by the definition of $\preceq$ for type G_2, some $\beta^{(j)}$ is 12 if and only if the leftmost column

$\beta^{(m)}$ is 12, and some $\beta^{(j)}$ is 1 if and only if the rightmost column $\beta^{(1)}$ is 1. As for the other types, we are going to examine how the corresponding word over Σ (that is, $c_{\beta^{(1)}} \cdots c_{\beta^{(m)}} c_x$) rewrites to an irreducible word. As before, the aim is to prove that this rewriting involves a single right-to-left pass through the word.

We first prove the following lemma, which tells us about the possible cases for x and the restrictions this puts on the columns $\beta^{(j)}$. We will then consider separately the rewriting that occurs according to whether some $\beta^{(j)}$ is the column 12.

Lemma 6.14. *The generator x can be*

1. 1*;*
2. 2*, only if there is at least one column* 1 *among the* $\beta^{(j)}$*;*
3. 3*, only if there is at least one column* 12 *among the* $\beta^{(j)}$*;*
4. 0*, only if there is at least one column* 1 *among the* $\beta^{(j)}$*;*
5. $\overline{3}$*, only if there are at least two columns* 1 *among the* $\beta^{(j)}$*;*
6. $\overline{2}$*, only if there is at least one column* 12 *among the* $\beta^{(j)}$*;*
7. $\overline{1}$*, only if there is at least one column* 1 *among the* $\beta^{(j)}$*.*

Proof. Note first that $\rho_1(1) = +$, $\rho_1(12) = +- = \varepsilon$, $\rho_2(1) = \varepsilon$, $\rho_2(12) = +$, so $\rho_1(\beta^{(1)} \cdots \beta^{(m)})$ consists of a string of symbols $+$ whose length is the number of columns 1 among the $\beta^{(j)}$, and $\rho_2(\beta^{(1)} \cdots \beta^{(m)})$ consists of a string of symbols $+$ whose length is the number of columns 12 among the $\beta^{(j)}$. The result now follows by considering how many symbols $+$ are required to cancel symbols $-$ in $\rho_i(x)$:

1. Nothing to prove.
2. Since $\rho_1(2) = -$, there must be at least one column 1 among the $\beta^{(j)}$;
3. Since $\rho_2(3) = -$, there must be at least one column 12 among the $\beta^{(j)}$;
4. Since $\rho_1(0) = -+$, there must be at least one column 1 among the $\beta^{(j)}$;
5. Since $\rho_1(\overline{3}) = --$, there must be at least two columns 1 among the $\beta^{(j)}$;
6. Since $\rho_2(\overline{2}) = -$, there must be at least one column 12 among the $\beta^{(j)}$;
7. Since $\rho_1(\overline{1}) = -$, there must be at least one column 1 among the $\beta^{(j)}$. $\square$

Consider first the case where there is no column 12 among the $\beta^{(j)}$. That is, $\beta^{(1)} \cdots \beta^{(m)} x = 1^m x$. In this case, x can be 1, 2, 0, $\overline{3}$ (only if $m \geq 2$), or $\overline{1}$ by Lemma 6.14, and so:

$$\begin{array}{|c|c|c|c|c|c|}\hline x & 1 & 1 & 1 & \cdots & 1 \\ \hline\end{array}$$

$$=_{\mathrm{Pl}(G_2)} \begin{cases} \begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & 1 & 1 & \cdots & 1 \\ \hline\end{array} & \text{if } x = 1; \\[2ex] \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 1 & \cdots & 1 \\ \hline 2 \\ \cline{1-1}\end{array} & \text{if } x = 2; \\[2ex] \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 1 & \cdots & 1 \\ \hline\end{array} & \text{if } x = 0, \text{ since } P(10) = \begin{array}{|c|}\hline 1 \\ \hline\end{array}; \\[2ex] \begin{array}{|c|c|c|c|}\hline 1 & 1 & \cdots & 1 \\ \hline 2 \\ \cline{1-1}\end{array} & \text{if } x = \overline{3}, \text{ since } P(1\overline{3}) = \begin{array}{|c|}\hline 2 \\ \hline\end{array}; \\[2ex] \begin{array}{|c|c|c|c|}\hline 1 & 1 & \cdots & 1 \\ \hline\end{array} & \text{if } x = \overline{1} \text{ since } P(1\overline{1}) \text{ is empty.} \end{cases}$$

In the first case, $c_{\beta^{(1)}} \cdots c_{\beta^{(m)}} c_x$ is in normal form; in the other four cases (respectively) the rewriting to normal form proceeds as follows:

$$
\begin{aligned}
&c_{\beta^{(1)}} \cdots c_{\beta^{(m)}} c_x = c_1 \cdots c_1 c_1 c_1 c_x \\
&\qquad \to \begin{cases}
c_1 \cdots c_1 c_1 c_{12} & \text{using } c_1 c_2 \to c_{12} \text{ since } P(12) = \boxed{12}; \\
c_1 \cdots c_1 c_1 c_1 & \text{using } c_1 c_0 \to c_1 \text{ since } P(10) = \boxed{1}; \\
c_1 \cdots c_1 c_1 c_2 & \text{using } c_1 c_{\bar{3}} \to c_2 \text{ since } P(1\bar{3}) = \boxed{2}; \\
\to c_1 \cdots c_1 c_{12} & \text{using } c_1 c_2 \to c_{12} \text{ since } P(12) = \boxed{12}; \\
c_1 \cdots c_1 c_1 & \text{using } c_1 c_{\bar{1}} \to \varepsilon \text{ since } P(1\bar{1}) \text{ is empty.}
\end{cases}
\end{aligned}
$$

In each case, rewriting $c_{\beta^{(1)}} \cdots c_{\beta^{(m)}} c_x$ to normal form involves at most two rewriting steps at the right-hand end of the word. Note that the length of the normal form differs from m by at most 2.

Next consider the case where there is at least one column 12 among the $\beta^{(j)}$. That is, $\beta^{(1)} \cdots \beta^{(m)} x = 1^h (12)^k x$, with $k \geq 1$ and $h \geq 0$. By Lemma 6.14, x can be 1, 2 (only if $h \geq 1$), 3, 0 (only if $h \geq 1$), $\bar{3}$ (only if $h \geq 2$), $\bar{2}$, or $\bar{1}$ (only if $h \geq 1$). Consider each case in turn:

1. $x = 1$. Then since $121 =_{\mathcal{R}_3^{G_2}} 112$, we have $P\left(\begin{array}{|c|c|}\hline 1 & 1 \\ \hline \multicolumn{1}{c|}{} & 2 \\ \cline{2-2}\end{array}\right) = \begin{array}{|c|c|}\hline 1 & 1 \\ \hline 2 \\ \cline{1-1}\end{array}$ and so $c_{12} c_1 \to c_1 c_{12}$. Thus, using this rule at each step, rewriting to normal form is as follows:

$$
\begin{aligned}
c_1 \cdots c_1 c_{12} \cdots c_{12} c_{12} c_1 &\to c_1 \cdots c_1 c_{12} \cdots c_{12} c_1 c_{12} \\
&\ \ \vdots \\
&\to c_1 \cdots c_1 c_1 c_{12} \cdots c_{12} c_{12}.
\end{aligned}
$$

2. $x = 2$. Then since $122 =_{\mathcal{R}_3^{G_2}} 212$, we have $P\left(\begin{array}{|c|c|}\hline 2 & 1 \\ \hline \multicolumn{1}{c|}{} & 2 \\ \cline{2-2}\end{array}\right) = \begin{array}{|c|c|}\hline 1 & 2 \\ \hline 2 \\ \cline{1-1}\end{array}$, and so $c_{12} c_2 \to c_2 c_{12}$ is a rewriting rule. As noted above, there is at least one column 1 present. So the rewriting to normal form proceeds as follows:

$$
\begin{aligned}
c_1 \cdots c_1 c_1 c_{12} \cdots c_{12} c_{12} c_2 &\to c_1 \cdots c_1 c_1 c_{12} \cdots c_{12} c_2 c_{12} && \text{using } c_{12} c_2 \to c_2 c_{12} \\
&\ \ \vdots \\
&\to c_1 \cdots c_1 c_1 c_2 c_{12} \cdots c_{12} c_{12} && \text{using } c_{12} c_2 \to c_2 c_{12} \\
&\to c_1 \cdots c_1 c_{12} c_{12} \cdots c_{12} c_{12}. && \text{using } c_1 c_2 \to c_{12}
\end{aligned}
$$

3. $x = 3$. Then since $123 =_{\mathcal{R}_4^{G_2}} 110 =_{\mathcal{R}_1^{G_2}} 11$ and $1211 =_{\mathcal{R}_3^{G_2}} 1121 =_{\mathcal{R}_3^{G_2}} 1112$, we have $P\left(\begin{array}{|c|c|}\hline 3 & 1 \\ \hline \multicolumn{1}{c|}{} & 2 \\ \cline{2-2}\end{array}\right) = \begin{array}{|c|c|}\hline 1 & 1 \\ \hline\end{array}$) So $c_{12} c_3 \to c_1 c_1$ is a rewriting rule.

Furthermore, $P\left(\begin{ytableau} 1 & 1 & 1 \\ \none & \none & 2 \end{ytableau}\right) = \begin{ytableau} 1 & 1 & 1 \\ 2 \end{ytableau}$. Thus add the extra rewriting rule $c_{12}c_1c_1 \to c_1c_1c_{12}$. Now rewriting to normal form is

$$\begin{aligned} c_1 \cdots c_1 c_{12} \cdots c_{12}c_{12}c_{12}c_3 &\to c_1 \cdots c_1 c_{12} \cdots c_{12}c_{12}c_1c_1 && \text{using } c_{12}c_3 \to c_1c_1 \\ &\to c_1 \cdots c_1 c_{12} \cdots c_{12}c_1c_1c_{12} && \text{using } c_1c_1c_{12} \to c_{12}c_1c_1 \\ &\ \ \vdots \\ &\to c_1 \cdots c_1c_1c_1c_{12} \cdots c_{12}c_{12}. && \text{using } c_1c_1c_{12} \to c_{12}c_1c_1 \end{aligned}$$

4. $x = 0$. Then since $120 =_{\mathcal{R}_4^{G_2}} 210 =_{\mathcal{R}_1^{G_2}} 21$ and $1221 =_{\mathcal{R}_3^{G_2}} 2121 =_{\mathcal{R}_3^{G_2}} 2112$ and $121 =_{\mathcal{R}_3^{G_2}} 112$, we have $P\left(\begin{ytableau} 0 & 1 \\ \none & 2 \end{ytableau}\right) = \begin{ytableau} 1 & 2 \end{ytableau}$ and so $c_{12}c_0 \to c_2c_1$ is a rewriting rule. Furthermore, $P\left(\begin{ytableau} 1 & 2 & 1 \\ \none & \none & 2 \end{ytableau}\right) = \begin{ytableau} 1 & 1 & 2 \\ 2 \end{ytableau}$ and $P\left(\begin{ytableau} 1 & 2 & 1 \end{ytableau}\right) = \begin{ytableau} 1 & 1 \\ 2 \end{ytableau}$. Thus, we add the extra rewriting rules $c_{12}c_2c_1 \to c_2c_1c_{12}$ and $c_1c_2c_1 \to c_1c_{12}$,
As noted above, there is at least one column 1 present. Rewriting to normal form is therefore as follows:

$$\begin{aligned} c_1 \cdots c_1 c_{12} \cdots c_{12}c_{12}c_{12}c_0 &\to c_1 \cdots c_1 c_{12} \cdots c_{12}c_{12}c_2c_1 && \text{using } c_{12}c_0 \to c_2c_1 \\ &\to c_1 \cdots c_1 c_{12} \cdots c_{12}c_2c_1c_{12} && \text{using } c_{12}c_2c_1 \to c_2c_1c_{12} \\ &\ \ \vdots \\ &\to c_1 \cdots c_1c_2c_1c_{12} \cdots c_{12}c_{12} && \text{using } c_{12}c_2c_1 \to c_2c_1c_{12} \\ &\to c_1 \cdots c_1c_{12}c_{12} \cdots c_{12}c_{12}. && \text{using } c_1c_2c_1 \to c_1c_{12} \end{aligned}$$

5. $x = \overline{3}$. Then since $12\overline{3} =_{\mathcal{R}_4^{G_2}} 21\overline{3} =_{\mathcal{R}_1^{G_2}} 22$, so $c_{12}c_{\overline{3}} \to c_2c_2$ is a rewriting rule. Furthermore, $1222 =_{\mathcal{R}_3^{G_2}} 2122 =_{\mathcal{R}_3^{G_2}} 2212$ and $1122 =_{\mathcal{R}_3^{G_2}} 1212$, we have $P\left(\begin{ytableau} \overline{3} & 1 \\ \none & 2 \end{ytableau}\right) = \begin{ytableau} 2 & 2 \end{ytableau})$ and $P\left(\begin{ytableau} 2 & 2 & 1 \\ \none & \none & 2 \end{ytableau}\right) = \begin{ytableau} 1 & 2 & 2 \\ 2 \end{ytableau}$ and $P\left(\begin{ytableau} 2 & 2 & 1 & 1 \end{ytableau}\right) = \begin{ytableau} 1 & 1 \\ 2 & 2 \end{ytableau}$. Thus, we add the extra rewriting rules $c_{12}c_2c_2 \to c_2c_2c_{12}$ and $c_2c_2c_1c_1 \to c_{12}c_{12}$.
As noted above, there are at least two columns 1 present. Rewriting to

normal form is therefore as follows:

$$
\begin{aligned}
c_1 \cdots c_1 c_1 c_1 c_{12} \cdots c_{12} c_{12} c_{12} c_{\bar{3}} &\to c_1 \cdots c_1 c_1 c_1 c_{12} \cdots c_{12} c_{12} c_2 c_2 && \text{using } c_{12} c_{\bar{3}} \to c_2 c_2 \\
&\to c_1 \cdots c_1 c_1 c_1 c_{12} \cdots c_{12} c_2 c_2 c_{12} && \text{using } c_{12} c_2 c_2 \to c_2 c_2 c_{12} \\
&\ \ \vdots \\
&\to c_1 \cdots c_1 c_1 c_1 c_2 c_2 c_{12} \cdots c_{12} c_{12} && \text{using } c_{12} c_2 c_2 \to c_2 c_2 c_{12} \\
&\to c_1 \cdots c_1 c_{12} c_{12} c_{12} \cdots c_{12} c_{12} && \text{using } c_2 c_2 c_1 c_1 \to c_{12} c_{12}.
\end{aligned}
$$

6. $x = \bar{2}$. Then since $12\bar{2} =_{\mathcal{R}_1^{G_2}} 10 =_{\mathcal{R}_1^{G_2}} 1$ and $121 =_{\mathcal{R}_3^{G_2}} 112$, we have $P\left(\begin{array}{cc} \boxed{\bar{2}} & \boxed{1} \\ & \boxed{2} \end{array}\right) = \boxed{1}$ and $P\left(\begin{array}{cc} \boxed{1} & \boxed{1} \\ & \boxed{2} \end{array}\right) = \begin{array}{cc} \boxed{1} & \boxed{1} \\ \boxed{2} & \end{array}$ and so $c_{12} c_{\bar{2}} \to c_1$ and $c_{12} c_1 \to c_1 c_{12}$ are rewriting rules.
Thus rewriting to normal form proceeds as follows:

$$
\begin{aligned}
c_1 \cdots c_1 c_{12} \cdots c_{12} c_{12} c_{12} c_{\bar{2}} &\to c_1 \cdots c_1 c_{12} \cdots c_{12} c_{12} c_1 && \text{using } c_{12} c_{\bar{2}} \to c_1 \\
&\to c_1 \cdots c_1 c_{12} \cdots c_{12} c_1 c_{12} && \text{using } c_{12} c_1 \to c_1 c_{12} \\
&\ \ \vdots \\
&\to c_1 \cdots c_1 c_1 c_{12} \cdots c_{12} c_{12}. && \text{using } c_{12} c_1 \to c_1 c_{12}
\end{aligned}
$$

7. $x = \bar{1}$. Then since $12\bar{1} =_{\mathcal{R}_1^{G_2}} = 1\bar{3} =_{\mathcal{R}_1^{G_2}} 2$ and $122 =_{\mathcal{R}_3^{G_2}} 212$, we have $P\left(\begin{array}{cc} \boxed{\bar{1}} & \boxed{1} \\ & \boxed{2} \end{array}\right) = \boxed{2}$ and $P\left(\begin{array}{cc} \boxed{2} & \boxed{1} \\ & \boxed{2} \end{array}\right) = \begin{array}{cc} \boxed{1} & \boxed{2} \\ \boxed{2} & \end{array}$, so $c_{12} c_{\bar{1}} \to c_2$ and $c_{12} c_2 \to c_2 c_{12}$.
As noted above, there is at least one column 1 present. Rewriting to normal form therefore proceeds as follows:

$$
\begin{aligned}
c_1 \cdots c_1 c_1 c_{12} \cdots c_{12} c_{12} c_{12} c_{\bar{1}} &\to c_1 \cdots c_1 c_1 c_{12} \cdots c_{12} c_{12} c_2 && \text{using } c_{12} c_{\bar{1}} \to c_2 \\
&\to c_1 \cdots c_1 c_1 c_{12} \cdots c_{12} c_2 c_{12} && \text{using } c_{12} c_2 \to c_2 c_{12} \\
&\ \ \vdots \\
&\to c_1 \cdots c_1 c_1 c_2 c_{12} \cdots c_{12} c_{12} && \text{using } c_{12} c_2 \to c_2 c_{12} \\
&\to c_1 \cdots c_1 c_{12} c_{12} \cdots c_{12} c_{12}. && \text{using } c_1 c_2 \to c_{12}
\end{aligned}
$$

Let Σ and T be the alphabet and set of rewriting rules constucted for type G_2 in Subsection 5.2

Let T' consist of the rules in T and by rules corresponding to:

- tabloids with shape $\begin{smallmatrix} \square & \square & \square \\ & & \square \end{smallmatrix}$ rewriting to tableaux with shape $\begin{smallmatrix} \square & \square & \square \\ \square & & \end{smallmatrix}$ (corresponding to extra rules in cases 3, 4, and 5 above);
- tabloids with shape $\begin{smallmatrix} \square & \square & \square \end{smallmatrix}$ rewriting to tableaux with shape $\begin{smallmatrix} \square & \square \\ \square & \end{smallmatrix}$ (corresponding to an extra rule in case 4 above);

- tabloids with shape $\ydiagram{4}$ rewriting to tableaux with shape $\ydiagram{2,2}$ (corresponding to an extra rule in case 5 above).

Note that the language of irreducible words is the same for the sets of rules T and T', since a left-hand side of some rule in T must appear as a subword of the left-hand side of each rule in T'.

Let $u \in \Sigma^*$ and $c_x \in G_2$. By the analysis above, rewriting uc_x to normal form using T' proceeds via a single right-to-left pass, since rewriting at highest weight using T' mirrors how rewriting proceeds in general. Note that in each case the length of the normal form word differs from m by at most 2.

As in the discussion before Lemma 6.7, the relation consisting of pairs (u, v) such that uc_x rewrites to v can be recognized by a transducer. The only modification to the argument is that the transducer that reads its input tapes right-to-left must stores the previous three symbols read from its first tape, so as to apply the rule in $T' \setminus T$, and will always give these new rules precedence. With that change, the same argument proves the following analogue of Lemma 6.7 for type G_2:

Lemma 6.15. *Let Σ and T' be as above. Let $x \in \mathcal{G}_2$. Let $L \subseteq \Sigma^*$ be the language of irreducible words with respect to T' (which is equal to the language of irreducible words with respect to T). Then the relation*

$$L_{c_x} = \{\, (u, v) \in L \times L : uc_x =_{\mathrm{Pl}(G_2)} v \,\}$$

is recognized by an transducer. Furthermore, if (u, v) is a pair in this relation, then the lengths of u and v differ by at most 2.

7. Building the biautomatic structure

Equipped with the lemmata from Subsections 6.3 and 6.4, we are now ready to prove biautomaticity for the plactic monoids. First, we recall the essential definitions in Subsection 7.1. We also state a result that allows us to discuss rational relations rather than synchronous rational relations, which helps avoids some technical reasoning (Proposition 7.3). In Subsection 7.2, we then proceed to build the biautomatic structures and to examine some consequences and applications of biautomaticity.

7.1. Preliminaries

This subsection contains the definitions and basic results from the theory of automatic and biautomatic monoids needed hereafter. For further information on automatic semigroups, see [49]. We assume familiarity with basic notions of automata and regular languages (see, for example, [79]).

Definition 7.1. Let A be an alphabet and let \$ be a new symbol not in A. Define the mapping $\delta_{\mathrm{R}} : A^* \times A^* \to ((A \cup \{\$\}) \times (A \cup \{\$\}))^*$ by

$$(u_1 \cdots u_m, v_1 \cdots v_n) \mapsto \begin{cases} (u_1, v_1) \cdots (u_m, v_n) & \text{if } m = n, \\ (u_1, v_1) \cdots (u_n, v_n)(u_{n+1}, \$) \cdots (u_m, \$) & \text{if } m > n, \\ (u_1, v_1) \cdots (u_m, v_m)(\$, v_{m+1}) \cdots (\$, v_n) & \text{if } m < n, \end{cases}$$

and the mapping $\delta_{\mathrm{L}} : A^* \times A^* \to ((A \cup \{\$\}) \times (A \cup \{\$\}))^*$ by

$$(u_1 \cdots u_m, v_1 \cdots v_n) \mapsto \begin{cases} (u_1, v_1) \cdots (u_m, v_n) & \text{if } m = n, \\ (u_1, \$) \cdots (u_{m-n}, \$)(u_{m-n+1}, v_1) \cdots (u_m, v_n) & \text{if } m > n, \\ (\$, v_1) \cdots (\$, v_{n-m})(u_1, v_{n-m+1}) \cdots (u_m, v_n) & \text{if } m < n, \end{cases}$$

where $u_i, v_i \in A$.

Definition 7.2. Let M be a monoid. Let A be a finite alphabet representing a set of generators for M and let $L \subseteq A^*$ be a regular language such that every element of M has at least one representative in L. For each $a \in A \cup \{\varepsilon\}$, define the relations

$$\begin{aligned} L_a &= \{(u, v) : u, v \in L, ua =_M v\} \\ {}_aL &= \{(u, v) : u, v \in L, au =_M v\}. \end{aligned}$$

The pair (A, L) is an *automatic structure* for M if $L_a\delta_{\mathrm{R}}$ is a regular language over $(A \cup \{\$\}) \times (A \cup \{\$\})$ for all $a \in A \cup \{\varepsilon\}$. A monoid M is *automatic* if it admits an automatic structure with respect to some generating set.

The pair (A, L) is a *biautomatic structure* for M if $L_a\delta_{\mathrm{R}}$, ${}_aL\delta_{\mathrm{R}}$, $L_a\delta_{\mathrm{L}}$, and ${}_aL\delta_{\mathrm{L}}$ are regular languages over $(A \cup \{\$\}) \times (A \cup \{\$\})$ for all $a \in A \cup \{\varepsilon\}$. A monoid M is *biautomatic* if it admits a biautomatic structure with respect to some generating set. [Note that biautomaticity implies automaticity.]

Unlike the situation for groups, biautomaticity for monoids and semigroups, like automaticity, is dependent on the choice of generating set [49, Example 4.5]. However, for monoids, biautomaticity and automaticity are independent of the choice of *semigroup* generating sets [80, Theorem 1.1].

Hoffmann & Thomas have made a careful study of biautomaticity for semigroups [81]. They distinguish four notions of biautomaticity for semigroups, which are all equivalent for groups and more generally for cancellative semigroups [81, Theorem 1] but distinct for semigroups [81, Remark 1 & § 4]. In the sense used in this paper, 'biautomaticity' implies *all four* of these notions of biautomaticity.

In proving that $R\delta_{\mathrm{R}}$ or $R\delta_{\mathrm{L}}$ is regular, where R is a relation on A^*, a useful strategy is to prove that R is a rational relation (that is, is recognized by a transducer) and then apply the following result, which is a combination of [82, Corollary 2.5] and [81, Proposition 4]:

Proposition 7.3. *If $R \subseteq A^* \times A^*$ is rational relation and there is a constant k such that $\big||u| - |v|\big| \leq k$ for all $(u, v) \in R$, then $R\delta_{\mathrm{R}}$ and $R\delta_{\mathrm{L}}$ are regular.*

7.2. Construction

In Subsections 6.3 and 6.4, we studied the rewriting that occurs when a normal form word is left- or right-multiplied by a generator. We now turn to building biautomatic structures for the plactic monoids of each type. Most of the work has been done; all that remains is to put together the pieces.

Theorem 7.4. *The plactic monoids* $\mathrm{Pl}(A_n)$*,* $\mathrm{Pl}(B_n)$*,* $\mathrm{Pl}(C_n)$*,* $\mathrm{Pl}(D_n)$*, and* $\mathrm{Pl}(G_2)$ *are biautomatic.*

Proof. Let X be one of the types A_n, B_n, C_n, D_n, and G_2 and let $\mathcal{X}$ be the corresponding alphabet from $\mathcal{A}_n$, $\mathcal{B}_n$, $\mathcal{C}_n$, $\mathcal{D}_n$, or $\mathcal{G}_2$. Let (Σ, T) be the rewriting system constructed in Section 5 for $\mathrm{Pl}(X)$. Let $L \subseteq \Sigma^*$ be the language of irreducible words.

Let $x \in \mathcal{X}$. By Lemmata 6.4 and 6.5, the relation

$$ {}_{c_x}L = \left\{ (u, v) \in L \times L : c_x u =_{\mathrm{Pl}(X)} v \right\} $$

is a rational relation. By Lemmata 6.7, 6.9, 6.11, 6.13, and 6.15, the relation

$$ L_{c_x} = \left\{ (u, v) \in L \times L : u c_x =_{\mathrm{Pl}(X)} v \right\} $$

is a rational relation.

Now let $c_\sigma \in \Sigma$. So σ is an admissible X column and $\sigma = \sigma_1 \cdots \sigma_k$ for some $\sigma_i \in \mathcal{X}$, with $k \leq n$ when $X \in \{A_n, B_n, C_n, D_n\}$ and $k \leq 2$ when $X = G_2$. So

$$ \begin{aligned} {}_{c_\sigma}L &= {}_{c_{\sigma_1}}L \circ \cdots \circ {}_{c_{\sigma_k}}L, \\ L_{c_\sigma} &= L_{c_{\sigma_1}} \circ \cdots \circ L_{c_{\sigma_k}}. \end{aligned} \tag{7.1} $$

Since the composition of a rational relation is a rational relation, L_{c_σ} and ${}_{c_\sigma}L$ are rational relations for any $c_\sigma \in \Sigma$.

By Lemmata 6.4 and 6.5, if $(u, v) \in {}_{c_x}L$ then $\big||u| - |v|\big| \leq 1$. Hence if $(u, v) \in {}_{c_\sigma}L$ them $\big||u| - |v|\big| \leq n$. Therefore ${}_{c_\sigma}L\delta_{\mathrm{R}}$ and ${}_{c_\sigma}L\delta_{\mathrm{L}}$ are both regular.

By Lemmata 6.7, 6.9, 6.11, 6.13, and 6.15, if $(u, v) \in L_{c_x}$ then $\big||u| - |v|\big| \leq 1$. Hence if $(u, v) \in L_{c_\sigma}$ them $\big||u| - |v|\big| \leq n$. Therefore $L_{c_\sigma}\delta_{\mathrm{R}}$ and $L_{c_\sigma}\delta_{\mathrm{L}}$ are both regular.

Therefore (Σ, L) is a biautomatic structure for $\mathrm{Pl}(X)$. $\square$

Theorem 7.4 has several important consequences for the plactic monoids $\mathrm{Pl}(A_n)$, $\mathrm{Pl}(B_n)$, $\mathrm{Pl}(C_n)$, $\mathrm{Pl}(D_n)$, and $\mathrm{Pl}(G_2)$. First, an automatic monoid has decidable right-divisibility problem. (This result is well-known but does not seem to be explicitly stated in the literature; it follows from the decidability of the first-order theory of the left Cayley graph of an automatic monoid [57, § 5].) Combining this result and its dual with Theorem 7.4 proves the following result:

Corollary 7.5. *The plactic monoids* $\mathrm{Pl}(A_n)$*,* $\mathrm{Pl}(B_n)$*,* $\mathrm{Pl}(C_n)$*,* $\mathrm{Pl}(D_n)$*, and* $\mathrm{Pl}(G_2)$ *have soluble left- and right-divisibilty problems.*

An immediate consequence of Corollary 7.5 is the following:

Corollary 7.6. *The plactic monoids* $\mathrm{Pl}(A_n)$*,* $\mathrm{Pl}(B_n)$*,* $\mathrm{Pl}(C_n)$*,* $\mathrm{Pl}(D_n)$*, and* $\mathrm{Pl}(G_2)$ *have soluble Green's relation* $\mathcal{L}$ *and* $\mathcal{R}$*.*

There are also several very important crystal-theoretic consequences of the biautomaticity of the plactic monoids:

Corollary 7.7. *For the crystal graphs of types A_n, B_n, C_n, D_n, or G_2, there is a quadratic-time algorithm that takes as input two vertices and decides whether they lie in the same position in isomorphic components.*

Proof. Two vertices lie in the same position in isomorphic connected components if and only if they represent the same element of the plactic monoid of the given type. This monoid is biautomatic by Theorem 7.4, and biautomatic (and automatic) monoids have word problem soluble in quadradic time [49, Corollary 3.7]. □

Note in passing that Corollary 7.7 cannot be deduced directly from tableaux insertion algorithms except in the A_n case. Schensted's insertion algorithm (see [8, Chapter 5]) can solve the word problem in $\mathrm{Pl}(A_n)$ in quadratic time because inserting a single symbol into a tableau takes linear time. However, in types B_n, C_n, and D_n inserting a single symbol into a tableau may take more that linear time (see [42, § 4] and [43, § 3.3]), because in certain cases a recursion arises that requires inserting an entire column symbol by symbol into the remainder of the tableau.

Corollary 7.8. *For the crystal graphs of types A_n, B_n, C_n, D_n, or G_2, there is a quadratic-time algorithm that takes as input two vertices and decides that whether they lie in isomorphic components.*

Proof. Let $B(u_1)$ and $B(u_2)$ be two components of the crystal graph, where u_1 and u_2 are any vertices of these components. Apply operators $\tilde{e}_i$ to transform u_1 and u_2 to highest-weight words v_1 and v_2 respectively. It is easy to see that each application of $\tilde{e}_i$ takes linear time in the length of the word. Each symbol of the word can be altered a bounded number of times by the various $\tilde{e}_i$, so computing v_1 and v_2 takes at most quadratic time in the lengths of u_1 and v_1. Then $B(u_1)$ and $B(u_2)$ are isomorphic if and only if v_1 and v_2 lie in the same position in $B(u_1)$ and $B(u_2)$, which can be decided in quadratic time by Corollary 7.7. □

References

[1] C. Schensted, Longest increasing and decreasing subsequences, Canad. J. Math. 13 (1961) 179–191. doi:10.4153/CJM-1961-015-3.

[2] D. E. Knuth, Permutations, matrices, and generalized Young tableaux, Pacific J. Math. 34 (3) (1970) 709–727.
URL http://projecteuclid.org/euclid.pjm/1102971948

[3] M.-P. Schützenberger, La correspondance de Robinson, in: Combinatoire et représentation du groupe symétrique (Actes Table Ronde CNRS, Univ. Louis-Pasteur Strasbourg, Strasbourg, 1976), Springer, Berlin, 1977, pp. 59–113. Lecture Notes in Math., Vol. 579.

[4] A. Lascoux, M.-P. Schützenberger, Le monoïde plaxique, in: Noncommutative structures in algebra and geometric combinatorics, no. 109 in Quaderni de "La Ricerca Scientifica", CNR, Rome, 1981, pp. 129–156.
URL http://igm.univ-mlv.fr/~berstel/Mps/Travaux/A/1981-1PlaxiqueNaples.pdf

[5] M.-P. Schützenberger, Pour le monoïde plaxique, Mathématiques et sciences humaines 140.
URL http://msh.revues.org/2764

[6] G. P. Thomas, Baxter algebras and schur functions, Ph.d. thesis, University College of Swansea (1974).

[7] G. P. Thomas, On Schensted's construction and the multiplication of Schur functions, Adv. in Math. 30 (1) (1978) 8–32. doi:10.1016/0001-8708(78)90129-9.
URL http://dx.doi.org/10.1016/0001-8708(78)90129-9

[8] M. Lothaire, Algebraic Combinatorics on Words, no. 90 in Encyclopedia of Mathematics and its Applications, Cambridge University Press, 2002.

[9] J. Green, Polynomial Representations of GLn, 2nd Edition, no. 830 in Lecture Notes in Mathematics, Springer, 2006. doi:10.1007/3-540-46944-3.

[10] M. A. van Leeuwen, The Littlewood-Richardson Rule, and Related Combinatorics, in: J. R. Stembridge, J.-Y. Thibon, M. A. van Leeuwen (Eds.), Interaction of Combinatorics and Representation Theory, Mathematical Society of Japan, Tokyo, 2001, pp. 95–145. doi:10.2969/msjmemoirs/01101C030.

[11] R. P. Stanley, Enumerative Combinatorics, Vol. 2 of Cambridge Studies in Advanced Mathematics, Cambridge University Press, 2005.

[12] A. Lascoux, M.-P. Schützenberger, Sur une conjecture de H. O. Foulkes, C. R. Acad. Sci. Paris Sér. A-B 286 (7) (1978) A323–A324.

[13] A. Lascoux, M.-P. Schützenberger, Schubert polynomials and the Littlewood-Richardson rule, Lett. Math. Phys. 10 (2-3) (1985) 111–124. doi:10.1007/BF00398147.
URL http://dx.doi.org/10.1007/BF00398147

[14] A. Lascoux, M.-P. Schützenberger, Noncommutative schubert polynomials, Funct. Anal. Appl. (23) (1990) 223–225.

[15] L. Serrano, The shifted plactic monoid, Mathematische Zeitschrift 266 (2) (2009) 363–392. doi:10.1007/s00209-009-0573-0.

[16] D. Krob, J.-Y. Thibon, Noncommutative symmetric functions. IV. Quantum linear groups and Hecke algebras at $q = 0$, J. Algebraic Combin. 6 (4) (1997) 339–376. doi:10.1023/A:1008673127310.
URL http://dx.doi.org/10.1023/A:1008673127310

[17] P. Littelmann, A Plactic Algebra for Semisimple Lie Algebras, Advances in Mathematics 124 (2) (1996) 312–331. doi:10.1006/aima.1996.0085.

[18] J. Cassaigne, M. Espie, D. Krob, J.-C. Novelli, F. Hivert, The Chinese Monoid, Int. J. Algebra Comput. 11 (3) (2001) 301–334. doi:10.1142/S0218196701000425.

[19] G. Duchamp, D. Krob, Plactic-growth-like monoids, in: Words, languages and combinatorics, II (Kyoto, 1992), World Sci. Publ., River Edge, NJ, 1994, pp. 124–142.

[20] A. J. Cain, A. Malheiro, Deciding conjugacy in sylvester monoids and other homogeneous monoids, Internat. J. Algebra Comput. 25 (5). arXiv:1404.2618, doi:10.1142/S0218196715500241.

[21] J. Okniński, On the semiprimitivity of finitely generated algebras, Proc. Amer. Math. Soc. 142 (12) (2014) 4095–4098. doi:10.1090/S0002-9939-2014-12187-3.
URL http://dx.doi.org/10.1090/S0002-9939-2014-12187-3

[22] E. Jespers, J. Okniński, M. Van Campenhout, Finitely generated algebras defined by homogeneous quadratic monomial relations and their underlying monoids, J. Algebra 440 (2015) 72–99. doi:10.1016/j.jalgebra.2015.05.017.
URL http://dx.doi.org/10.1016/j.jalgebra.2015.05.017

[23] F. Cedó, J. Okniński, Gröbner bases for quadratic algebras of skew type, Proc. Edinb. Math. Soc. (2) 55 (2) (2012) 387–401. doi:10.1017/S0013091511000447.

[24] P. Dehornoy, Set-theoretic solutions of the Yang-Baxter equation, RC-calculus, and Garside germs, Adv. Math. 282 (2015) 93–127. doi:10.1016/j.aim.2015.05.008.
URL http://dx.doi.org/10.1016/j.aim.2015.05.008

[25] Ł. Kubat, J. Okniński, Identities of the plactic monoid, Semigroup Forum 90 (1) (2015) 100–112. doi:10.1007/s00233-014-9609-9.
URL http://dx.doi.org/10.1007/s00233-014-9609-9

[26] P. Dehornoy, Quadratic normalisation in monoids, arXiv:1504.02717.

[27] F. Cedó, J. Okniński, Plactic algebras, J. Algebra 274 (1) (2004) 97–117. doi:10.1016/j.jalgebra.2003.12.004.

[28] Ł. Kubat, J. Okniński, Plactic algebra of rank 3, Semigroup Forum 84 (2) (2012) 241–266. doi:10.1007/s00233-011-9337-3.

[29] J. Hong, S.-J. Kang, Introduction to Quantum Groups and Crystal Bases, no. 42 in Graduate Studies in Mathematics, American Mathematical Society, 2002.

[30] V. G. Drinfel'd, Hopf algebras and the quantum Yang-Baxter equation, Dokl. Akad. Nauk SSSR 283 (5) (1985) 1060–1064.

[31] M. Jimbo, A q-difference analogue of $U(\mathfrak{g})$ and the Yang-Baxter equation, Lett. Math. Phys. 10 (1) (1985) 63–69. doi:10.1007/BF00704588.
URL http://dx.doi.org/10.1007/BF00704588

[32] M. Kashiwara, Crystalizing the q-analogue of universal enveloping algebras, Comm. Math. Phys. 133 (2) (1990) 249–260.
URL http://projecteuclid.org/euclid.cmp/1104201397

[33] M. Kashiwara, On crystal bases of the Q-analogue of universal enveloping algebras, Duke Math. J. 63 (2) (1991) 465–516. doi:10.1215/S0012-7094-91-06321-0.
URL http://dx.doi.org/10.1215/S0012-7094-91-06321-0

[34] S.-J. Kang, Crystal bases for quantum affine algebras and combinatorics of Young walls, Proc. London Math. Soc. (3) 86 (1) (2003) 29–69. doi:10.1112/S0024611502013734.
URL http://dx.doi.org/10.1112/S0024611502013734

[35] G. Benkart, S.-J. Kang, M. Kashiwara, Crystal bases for the quantum superalgebra $U_q(\mathfrak{gl}(m,n))$, J. Amer. Math. Soc. 13 (2) (2000) 295–331. doi:10.1090/S0894-0347-00-00321-0.
URL http://dx.doi.org/10.1090/S0894-0347-00-00321-0

[36] D. Grantcharov, J. H. Jung, S.-J. Kang, M. Kashiwara, M. Kim, Crystal bases for the quantum queer superalgebra and semistandard decomposition tableaux, Trans. Amer. Math. Soc. 366 (1) (2014) 457–489. doi:10.1090/S0002-9947-2013-05866-7.
URL http://dx.doi.org/10.1090/S0002-9947-2013-05866-7

[37] D. Grantcharov, J. H. Jung, S.-J. Kang, M. Kashiwara, M. Kim, Crystal bases for the quantum queer superalgebra, J. Eur. Math. Soc. (JEMS) 17 (7) (2015) 1593–1627. doi:10.4171/JEMS/540.
URL http://dx.doi.org/10.4171/JEMS/540

[38] S.-J. Kang, K. C. Misra, Crystal bases and tensor product decompositions of $U_q(G_2)$-modules, J. Algebra 163 (3) (1994) 675–691. doi:10.1006/jabr.1994.1037.
URL http://dx.doi.org/10.1006/jabr.1994.1037

[39] M. Kashiwara, T. Nakashima, Crystal graphs for representations of the q-analogue of classical Lie algebras, J. Algebra 165 (2) (1994) 295–345. doi:10.1006/jabr.1994.1114.
URL http://dx.doi.org/10.1006/jabr.1994.1114

[40] P. Littelmann, Crystal graphs and Young tableaux, J. Algebra 175 (1) (1995) 65–87. doi:10.1006/jabr.1995.1175.
URL http://dx.doi.org/10.1006/jabr.1995.1175

[41] C. Lecouvey, Combinatorics of crystal graphs for the root systems of types A_n, B_n, C_n, D_n and G_2, in: K. Atsuo, M. Okado (Eds.), Combinatorial aspect of integrable systems, no. 17 in Mathematical Society of Japan Memoirs, Mathematical Society of Japan, Tokyo, 2007.

[42] C. Lecouvey, Schensted-Type Correspondence, Plactic Monoid, and Jeu de Taquin for Type C_n, J. Algebra 247 (2) (2002) 295–331. doi:10.1006/jabr.2001.8905.

[43] C. Lecouvey, Schensted-type correspondences and plactic monoids for types B_n and D_n, Journal of Algebraic Combinatorics 18 (2) (2003) 99–133. doi:10.1023/A:1025154930381.

[44] T. H. Baker, An insertion scheme for C_n crystals, in: Physical combinatorics (Kyoto, 1999), Vol. 191 of Progr. Math., Birkhäuser Boston, Boston, MA, 2000, pp. 1–48.

[45] C. Lecouvey, Crystal bases and combinatorics of infinite rank quantum groups, Trans. Amer. Math. Soc. 361 (1) (2009) 297–329. doi:10.1090/S0002-9947-08-04480-2.
URL http://dx.doi.org/10.1090/S0002-9947-08-04480-2

[46] M. Kashiwara, On crystal bases, in: Representations of groups (Banff, AB, 1994), Vol. 16 of CMS Conf. Proc., Amer. Math. Soc., Providence, RI, 1995, pp. 155–197.

[47] R. V. Book, F. Otto, String Rewriting Systems, Texts and Monographs in Computer Science, Springer, 1993.

[48] D. B. Epstein, J. W. Cannon, D. F. Holt, S. V. Levy, M. S. Paterson, W. P. Thurston, Word Processing in Groups, Jones & Bartlett, Boston, MA, 1992.

[49] C. M. Campbell, E. F. Robertson, N. Ruškuc, R. M. Thomas, Automatic semigroups, Theoret. Comput. Sci. 250 (1-2) (2001) 365–391. doi:10.1016/S0304-3975(99)00151-6.

[50] S. M. Gersten, H. B. Short, Small cancellation theory and automatic groups, Invent. Math. 102 (2) (1990) 305–334. doi:10.1007/BF01233430.
URL http://dx.doi.org/10.1007/BF01233430

[51] D. F. Holt, S. Rees, Artin groups of large type are shortlex automatic with regular geodesics, Proc. Lond. Math. Soc. (3) 104 (3) (2012) 486–512. doi:10.1112/plms/pdr035.
URL http://dx.doi.org/10.1112/plms/pdr035

[52] M. Gromov, Hyperbolic Groups, in: S. Gersten (Ed.), Essays in Group Theory, no. 8 in Math. Sci. Res. Inst. Publ., Springer-Verlag, 1987, pp. 75–263.

[53] M. Picantin, Finite transducers for divisibility monoids, Theoret. Comput. Sci. 362 (1-3) (2006) 207–221. doi:10.1016/j.tcs.2006.06.019.
URL http://dx.doi.org/10.1016/j.tcs.2006.06.019

[54] R. Corran, M. Hoffmann, D. Kuske, R. M. Thomas, Singular Artin Monoids of Finite Coxeter Type Are Automatic, in: Language and Automata Theory and Applications, no. 6638 in Lecture Notes in Comput. Sci., Springer, 2011, pp. 250–261. doi:10.1007/978-3-642-21254-3_19.

[55] F. Otto, N. Ruškuc, Confluent monadic string-rewriting systems and automatic structures, J. Autom. Lang. Comb. 6 (3) (2001) 375–388.

[56] A. J. Cain, Monoids presented by rewriting systems and automatic structures for their submonoids, Internat. J. Algebra Comput. 19 (6) (2009) 771–790. doi:10.1142/s0218196709005317.

[57] M. Lohrey, Decidability and complexity in automatic monoids, Int. J. Found. Comput. Sci. 16 (04) (2005) 707–722. doi:10.1142/S0129054105003248.

[58] F. Otto, On Dehn functions of finitely presented bi-automatic monoids, J. Autom. Lang. Comb. 5 (4) (2000) 405–419.

[59] F. Otto, A. Sattler-Klein, K. Madlener, Automatic monoids versus monoids with finite convergent presentations, in: T. Nipkow (Ed.), Rewriting Techniques and Applications, no. 1379 in Lecture Notes in Comput. Sci., Springer, 1998, pp. 32–46. doi:10.1007/BFb0052359.

[60] A. J. Cain, R. D. Gray, A. Malheiro, Finite Gröbner–Shirshov bases for Plactic algebras and biautomatic structures for Plactic monoids, J. Algebra 423 (2015) 37–53. arXiv:1205.4885, doi:10.1016/j.jalgebra.2014.09.037.

[61] V. Ufnarovskiĭ, Combinatorial and asymptotic methods in algebra, in: A. Kostrikin, I. Shafarevich (Eds.), Algebra VI, Vol. 57 of Encyclopedia of Mathematical Sciences, Springer, 1995, pp. 1–196.

[62] N. Hage, Finite convergent presentation of plactic monoid for type C, Int. J. Algebra Comput. 25 (08) (2015) 1239–1263. doi:10.1142/S0218196715500393.

[63] A. J. Cain, R. D. Gray, A. Malheiro, On finite complete rewriting systems, finite derivation type, and automaticity for homogeneous monoids, Inform. and Comput. 255 (1) (2017) 68–93. arXiv:1407.7428, doi:10.1016/j.ic.2017.05.003.

[64] J. R. Stembridge, A local characterization of simply-laced crystals, Trans. Amer. Math. Soc. 355 (12) (2003) 4807–4823. doi:10.1090/S0002-9947-03-03042-3.

[65] P. Sternberg, On the local structure of doubly laced crystals, J. Combin. Theory Ser. A 114 (5) (2007) 809–824.
URL https://doi.org/10.1016/j.jcta.2006.09.003

[66] V. Danilov, A. Karzanov, G. Koshevoy, B2-crystals: Axioms, structure, models, Journal of Combinatorial Theory, Series A 116 (2) (2009) 265 – 289. doi:https://doi.org/10.1016/j.jcta.2008.06.002.
URL http://www.sciencedirect.com/science/article/pii/S0097316508000903

[67] I. G. Macdonald, Symmetric functions and Hall polynomials, The Clarendon Press, Oxford University Press, New York, 1979, oxford Mathematical Monographs.

[68] J. T. Sheats, A symplectic jeu de taquin bijection between the tableaux of King and of De Concini, Trans. Amer. Math. Soc. 351 (9) (1999) 3569–3607.
URL https://doi.org/10.1090/S0002-9947-99-02166-2

[69] T. Nakashima, Crystal base and a generalization of the littlewood-richardson rule for the classical lie algebras, Comm. Math. Phys. 154 (2) (1993) 215–243.
URL https://projecteuclid.org:443/euclid.cmp/1104252969

[70] F. Baader, T. Nipkow, Term Rewriting and All That, Cambridge University Press, 1999.

[71] N. Ruškuc, Semigroup Presentations, Ph.D. thesis, University of St Andrews (1995).
URL http://hdl.handle.net/10023/2821

[72] P. M. Higgins, Techniques of Semigroup Theory, Oxford University Press, 1991.

[73] C. C. Squier, F. Otto, Y. Kobayashi, A finiteness condition for rewriting systems, Theoret. Comput. Sci. 131 (2) (1994) 271–294. doi:10.1016/0304-3975(94)90175-9.

[74] D. J. Anick, On the homology of associative algebras, Trans. Amer. Math. Soc. 296 (2) (1986) 641–641. doi:10.2307/2000383.

[75] K. S. Brown, The geometry of rewriting systems: a proof of the Anick-Groves-Squier theorem, in: Algorithms and classification in combinatorial group theory (Berkeley, CA, 1989), Vol. 23 of Math. Sci. Res. Inst. Publ., Springer, New York, 1992, pp. 137–163. doi:10.1007/978-1-4613-9730-4_6.
URL http://dx.doi.org/10.1007/978-1-4613-9730-4_6

[76] D. E. Cohen, String rewriting and homology of monoids, Mathematical Structures in Computer Science 7 (3) (1997) 207–240. doi:10.1017/S0960129596002149.

[77] J. Sakarovitch, Elements of Automata Theory, Cambridge University Press, 2009.

[78] J. Berstel, Transductions and Context-Free Languages, no. 38 in Leitfäden der Angewandten Mathematik und Mechanik, B. G. Teubner, Stuttgart, 1979.

[79] J. E. Hopcroft, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, 1st Edition, Addison–Wesley, Reading, MA, 1979.

[80] A. Duncan, E. Robertson, N. Ruškuc, Automatic monoids and change of generators, Mathematical Proceedings of the Cambridge Philosophical Society 127 (3) (1999) 403–409. doi:10.1017/S0305004199003722.

[81] M. Hoffmann, R. M. Thomas, Biautomatic semigroups, in: M. Liskiewicz, R. Reischuk (Eds.), Fundamentals of Computation Theory, no. 3623 in Lecture Notes in Comput. Sci., Springer, 2005, pp. 56–67. doi:10.1007/11537311_6.

[82] C. Frougny, J. Sakarovitch, Synchronized rational relations of finite and infinite words, Theoret. Comput. Sci. 108 (1) (1993) 45–82. doi:10.1016/0304-3975(93)90230-Q.