Invariant bilinear forms on W-graph representations and linear algebra over integral domains

Meinolf Geck and Jürgen Müller

Abstract Lie-theoretic structures of type E_8 (e.g., Lie groups and algebras, Iwahori–Hecke algebras and Kazhdan–Lusztig cells, ...) are considered to serve as a "gold standard" when it comes to judging the effectiveness of a general algorithm for solving a computational problem in this area. Here, we address a problem that occurred in our previous work on decomposition numbers of Iwahori–Hecke algebras, namely, the computation of invariant bilinear forms on so-called W-graph representations. We present a new algorithmic solution which makes it possible to produce and effectively use the main results in further applications.

1 Introduction

This paper is concerned with the representation theory of Iwahori–Hecke algebras. Such an algebra $\mathscr{H}$ is a certain deformation of the group algebra of a finite Coxeter group W. In [7], the notion of "balanced representations" of $\mathscr{H}$ was introduced, which has turned out to be useful in several applications. We mention here the construction of cellular structures on $\mathscr{H}$ (see, e.g., [10, Chap. 2]), the determination of decomposition numbers of $\mathscr{H}$ (see [11]), and the computation of Lusztig's function $\mathbf{a}\colon W \to \mathbb{Z}$ (see [8, §4]). To check whether a given representation of $\mathscr{H}$ is balanced or not is a computationally hard problem; it involves the construction of a certain invariant bilinear form on the underlying $\mathscr{H}$-module. It has been conjectured in [7] that so-called "W-graph representations" of $\mathscr{H}$ are always balanced. But even if such a theoretical result were known to be true, certain applications (e.g., the de-

Meinolf Geck
IAZ-Lehrstuhl für Algebra, Universität Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany
e-mail: meinolf.geck@mathematik.uni-stuttgart.de

Jürgen Müller
Arbeitsgruppe Algebra und Zahlentheorie, Bergische Universität Wuppertal, Gauß-Straße 20, 42119 Wuppertal, Germany e-mail: juergen.mueller@math.uni-wuppertal.de

termination of decomposition numbers) would still require the explicit knowledge of the Gram matrices of the invariant bilinear forms. In this paper, we discuss algorithms for the construction of these Gram matrices for W of exceptional type. The biggest challenge—by far—is the case where W is of type E_8. (The distinguished role of E_8 when it comes to performing explicit computations is highlighted in various recent survey articles; see, e.g., Garibaldi [6], Lusztig [19], Vogan [26]).

In the situations of interest to us, the algebra $\mathscr{H}$ is defined over the field of rational functions $K = \mathbb{Q}(v)$ (where v is an indeterminate); it has a natural basis $\{T_w \mid w \in W\}$. Explicit models for the irreducible representations of $\mathscr{H}$ are known by the work of Naruse [23], Howlett and Yin [15], [16]. Now let us fix an irreducible matrix representation $\mathfrak{X}\colon \mathscr{H} \to K^{d\times d}$. In order to show that $\mathfrak{X}$ is balanced, one needs to determine a non-zero symmetric matrix $P \in K^{d\times d}$ such that

$$P\,\mathfrak{X}(T_w) = \mathfrak{X}(T_{w^{-1}})^{\mathrm{tr}}\,P \qquad \text{for all } w \in W;$$

this matrix P then has to satisfy certain additional properties. Thus, the computation of P essentially amounts to solving a system of linear equations; for theoretical reasons, we know that this system has a unique solution up to multiplication by a scalar. Rescaling a given solution by a suitable non-zero polynomial in $\mathbb{Q}[v]$, we can assume that all entries of P are in $\mathbb{Z}[v]$ and that their greatest common divisor is ± 1; then P is unique up to sign and is called a “primitive Gram matrix”. The general theory also shows that a particular solution is given by

$$P_0 = \sum_{w\in W} \mathfrak{X}(T_w)^{\mathrm{tr}}\,\mathfrak{X}(T_w) \in K^{d\times d}.$$

Thus, if the matrices $\mathfrak{X}(T_w)$ $(w \in W)$ are known and if $|W|$ is not too large, then we can simply perform the above summation and obtain P_0; rescaling P_0 yields a primitive Gram matrix P. This procedure works for types F_4, E_6, for example.

Already for type E_7, one needs to use a more sophisticated approach as described in [11, §4.3], based on Parker’s “standard basis algorithm” [24], in combination with interpolation and modular techniques. This also works for type E_8, but it is efficient only for irreducible representations of dimension up to about 2500. In our previous work on decomposition numbers, this was sufficient to obtain the desired results for type E_8; see [11, Remark 4.10]. In principle, one could have run the above procedure on all irreducible representations of type E_8, but experiments showed that this would have needed a total of nearly one year of CPU time. On the other hand, from a strictly logical point of view, one does not need to know exactly how the Gram matrices have been obtained, because as an independent verification one can simply check that they form a solution to the above system of linear equations. However, to store the various primitive Gram matrices requires about 28 GB of disk space, and even the verification alone is a major task as it involves the computation of products of (large) matrices with polynomial entries. — In any case, this raises a serious issue of making sure that our results are reliable and reproducible.

In our view, the solution to deal with this issue is to develop better mathematical tools which make it possible to reproduce the results efficiently as needed, and this is what we will do in this paper. Indeed, for example, in order to deal with the irreducible representation of largest dimension for type E_8 (which is 7168), the old approach would have needed roughly seven weeks of CPU time, while the one described here requires only about 20 hours, which amounts to a factor of almost 60. (See Section 9.1 for more details.) In view of the complexity of the task, and the experiences made elsewhere with explicit computations in type E_8 (see the references cited above), it was clear that developing efficient methods would not be a standard, let alone press-button application of existing tools from computer algebra. Maier et al. [20] proposed an approach based on parallel techniques, but type E_8 still seems to be a major challenge there. Hence one of the purposes of this paper is to give a systematic description of the (serial) methods we have used for the computation of Gram matrices of invariant bilinear forms for Iwahori–Hecke algebras.

The basic strategy in our approach is to reduce computational linear algebra over the Laurent polynomial ring $\mathbb{Q}[v, v^{-1}]$ to linear algebra over the integers. Thus, generally speaking, we are faced with the problem of devising efficient tools to do computational linear algebra over integral domains, not just over fields. In order to do so, we build on general ideas from computational representation theory, more precisely on the celebrated so-called MeatAxe philosophy [24], which comprises of specially tailored, highly efficient techniques for computational linear algebra over (small) finite fields. Attempts to generalize these ideas to linear algebra over the (infinite) field of rational numbers, and further to linear algebra over the integers have been coined the IntegralMeatAxe [25]. The last word on this has not been said yet, and in this paper we are trying to contribute here as well. (As future work, we are planning to develop a full IntegralMeatAxe package along the present lines.) But we are additionally going one step further by setting out to extend these ideas to linear algebra over the univariate polynomial rings over the rationals or the integers.

To do so, the basic idea is to reduce to linear algebra over the integers by evaluating polynomials with rational coefficients at integral places, where we are using as few "small" places as possible, and to recover the polynomials in question by a Chinese remainder technique. Hence this strategy, fitting nicely into the IntegralMeatAxe philosophy, differs from those known to the literature, inasmuch we are neither using modular methods (which would mean to go over to polynomial rings over finite fields), nor are we in a position to use interpolation (which would mean to use lots of places to evaluate at). Thus another purpose of this paper is to give a detailed description of the new computational tasks arising in pursuing this strategy, and how we have accomplished them. Although the choice of the material presented is governed by our application to Iwahori–Hecke algebras, it is exhibited with a view towards general applicability.

Here is an outline of the paper: In **Section 2** we recall some basic facts about representations of finite Coxeter groups and Iwahori–Hecke algebras, in particular the notions of W-graphs, balancedness, and invariant bilinear forms. We conclude with Theorem 2.10 saying that for the representations afforded by the W-graphs given by Naruse [23], Howlett and Yin [15], [16] are actually balanced, and in Tables 1 and 2 we list some numerical data associated with their primitive Gram matrices.

In the subsequent sections we describe our general approach towards linear algebra over integral domains, which consists of a cascade of steps: In **Section 3** we first deal with linear algebra over $\mathbb{Z}$. We discuss the key tasks of rational number recovery and of finding integral linear dependencies. Both tasks are known to the literature, but for the former we provide a variant containing a new feature, while for the latter we proceed along another strategy, within the IntegralMeatAxe philosophy. Subsequently, we apply this to computing nullspaces, inverses, and the so-called "exponents" of matrices over $\mathbb{Z}$. In **Section 4** we then describe our general approach to deal with polynomials, in view of our aim to do linear algebra over polynomial rings. The key task is to recover a polynomial with rational coefficients from some of its evaluations at integral places. Here, we are aiming at using as few "small" places as possible, whence we are not in a position to apply interpolation, but we are using a Chinese remainder technique instead. Moreover, we devise a method to recover a polynomial from some of its evaluations where the latter are "rescaled" by unknown scalars; the necessity of being able to solve this task is closely related to our use of the IntegralMeatAxe, hence to our knowledge this method is new as well. In **Section 5** we proceed to show how linear algebra over $\mathbb{Z}$ and polynomial recovery, as discussed in earlier sections, can now be combined to do linear algebra over $\mathbb{Z}[X]$ and $\mathbb{Q}[X]$, by devising methods to computing nullspaces, inverses, exponents and products of matrices using this new approach. In **Section 6** we finally recall the "standard basis algorithm" originally developed in [24] for computations over finite fields. We present a general variant for absolutely irreducible matrix representations over an arbitrary field, show how this can be used to compute homomorphisms between such representations, and discuss how the necessary computations are facilitated over the fields $\mathbb{Q}$ and $\mathbb{Q}(X)$, using the tools we have developed.

Having the general tools in place, in **Section 7** we return to our particular application of computing Gram matrices of invariant bilinear forms for W-graph representations $\mathfrak{X}$ of Iwahori–Hecke algebras. We proceed along the strategy which has already been indicated in [11, Section 4.3], where here we take the opportunity to provide full details. We begin by computing standard bases for the representations $\mathfrak{X}$ and $\mathfrak{X}'$, where the latter is given by $\mathfrak{X}'(T_w) := \mathfrak{X}(T_{w^{-1}})^{\mathrm{tr}}$, for $w \in W$. In order to find suitable seed vectors to start with, we use an observation on restrictions of representations of Iwahori–Hecke algebras to parabolic subalgebras, which naturally leads to certain distinguished elements of $\mathscr{H}$ having actions of co-rank one on $\mathfrak{X}$ and $\mathfrak{X}'$. To actually run the standard basis algorithm subsequently, we again revert to a specialization technique. In **Section 8** we proceed by collecting a few observations on the standard bases B and B' of the representations $\mathfrak{X}$ and $\mathfrak{X}'$ thus obtained. Indeed, the matrix entries occurring seem to be much less arbitrary than expected from general principles, but this has only been verified experimentally for the representations under consideration here, while a priori proofs are largely missing (so far). The final computational step then essentially is to determine the product $B^{-1} \cdot B'$, which up to rescaling is a Gram matrix as desired. To do this efficiently, apart from the general tools developed above, we make heavy use of the special form of the matrix entries of $B^{-1} \cdot B'$ just mentioned. In the concluding **Section 9** we provide running times and workspace requirements for our computations in types E_7 and E_8, and present an explicit (tiny) example for type E_6.

It should be clear from the above description that to pursue our novel approach we had to solve quite a few tasks for which there was no pre-existing implementation, let alone in one and the same computer algebra system. To develop the necessary new code, as our computational platform we have chosen the computer algebra system GAP [4]. This system provides efficient arithmetics for the various basic objects we need: (i) rational integers and rational numbers, which in turn are handled by the GMP library [13]; (ii) row vectors and matrices over the integers, the rationals or (small) finite fields, where in this context the entries of row vectors are actually treated as immediate objects; (iii) floating point numbers, where the limited built-in facilities are sufficient for our purposes. Moreover, the necessary input data on Iwahori–Hecke algebras and their representations is provided by the computer algebra system CHEVIE [21], which conveniently is a branch of GAP.

Contents

2 Iwahori–Hecke algebras and balanced representations

We begin by recalling some basic facts about representations of finite Coxeter groups and Iwahori–Hecke algebras; see [12], [10], [18] for further details.

2.1. We fix a finite Coxeter group W with set of simple reflections S; for $w \in W$, we denote by $l(w)$ the length of w with respect to S. Let $L\colon W \to \mathbb{Z}$ be a weight function as in [18], that is, we have $L(ww') = L(w) + L(w')$ whenever $w, w' \in W$ satisfy $l(ww') = l(w) + l(w')$. Such a weight function is uniquely determined by its values $L(s)$ for $s \in S$. We will assume throughout that

$$L(s) > 0 \qquad \text{for all } s \in S.$$

Let $R \subseteq \mathbb{C}$ be a subring and $A = R[v, v^{-1}]$ be the ring of Laurent polynomials over R in the indeterminate v. Let $\mathscr{H} = \mathscr{H}_A(W, L)$ be the corresponding generic Iwahori–Hecke algebra. Thus, $\mathscr{H}$ is an associative A-algebra which is free over A with a basis $\{T_w \mid w \in W\}$; the multiplication is given by the following rule, where $s \in S$ and $w \in W$:

$$T_sT_w = \begin{cases} T_{sw} & \text{if } l(sw) = l(w)+1, \\ T_{sw} + (v^{L(s)} - v^{-L(s)})T_s & \text{if } l(sw) = l(w)-1. \end{cases}$$

2.2. Let $F \subseteq \mathbb{C}$ be the field of fractions of R and assume that F is a splitting field for W. (For example, we could take $R = F = \mathbb{R}$ since $\mathbb{R}$ is known to be a splitting field for W.) Let $\mathrm{Irr}(W)$ be the set of simple $F[W]$-modules (up to isomorphism); we shall use the following notation:

$$\mathrm{Irr}(W) = \{E^\lambda \mid \lambda \in \Lambda\} \qquad \text{and} \qquad d_\lambda = \dim E^\lambda \;\; (\lambda \in \Lambda),$$

where Λ is a finite index set. Let $K = F(v)$ be the field of fractions of A and $\mathscr{H}_K = K \otimes_A \mathscr{H}$ be the K-algebra obtained by extension of scalars from A to K. Then $\mathscr{H}_K$ is a split semisimple algebra and there is a bijection between $\mathrm{Irr}(W)$ and $\mathrm{Irr}(\mathscr{H}_K)$, the set of simple $\mathscr{H}_K$-modules (up to isomorphism). Given $\lambda \in \Lambda$, we denote by E_v^λ a simple $\mathscr{H}_K$-module corresponding to E^λ. Then E_v^λ is uniquely determined (up to isomorphism) by the following property. For $w \in W$, we have

$$\mathrm{trace}(T_w, E_v^\lambda) \in F[v, v^{-1}] \qquad \text{and} \qquad \mathrm{trace}(w, E^\lambda) = \mathrm{trace}(T_w, E_v^\lambda)|_{v \mapsto 1}.$$

2.3. The algebra $\mathscr{H}_K$ is symmetric, with trace form $\tau\colon \mathscr{H}_K \to K$ given by $\tau(T_1) = 1$ and $\tau(T_w) = 0$ for $1 \neq w \in W$. The basis dual to $\{T_w \mid w \in W\}$ is given by $\{T_{w^{-1}} \mid w \in W\}$. By the general theory of symmetric algebras, there are well-defined elements $0 \neq \mathbf{c}_\lambda \in A$ $(\lambda \in \Lambda)$ such that the following orthogonality relations hold for $\lambda, \mu \in \Lambda$:

$$\sum_{w \in W} \mathrm{trace}(T_w, E_v^\lambda)\mathrm{trace}(T_{w^{-1}}, E_v^\mu) = \begin{cases} d_\lambda \mathbf{c}_\lambda & \text{if } \lambda = \mu, \\ 0 & \text{if } \lambda \neq \mu. \end{cases}$$

As observed by Lusztig, we can write each $\mathbf{c}_\lambda$ uniquely in the form

$$\mathbf{c}_\lambda = f_\lambda v^{-2\mathbf{a}_\lambda} + \text{linear combination of larger powers of } v,$$

where f_λ is a strictly positive real number and $\mathbf{a}_\lambda$ is a non-negative integer. The "a-invariants" $\mathbf{a}_\lambda$ will play a major role in the sequel; these numbers are explicitly known for all types of W and all choices of L (see [10, §1.3], [18, Chap. 22]). Alternatively, $\mathbf{a}_\lambda$ can be characterized as follows:

$$\mathbf{a}_\lambda = \min\{i \geqslant 0 \mid v^i \mathrm{trace}(T_w, E_v^\lambda) \in F[v] \text{ for all } w \in W\}.$$

2.4. Let $\mathscr{O} \subseteq K$ be the localization of $F[v]$ in the prime ideal (v), that is, $\mathscr{O}$ consists of all fractions of the form $f/g \in K$ where $f, g \in F[v]$ and $g(0) \neq 0$. Let $\mathfrak{X}^\lambda : \mathscr{H}_K \to K^{d_\lambda \times d_\lambda}$ be a matrix representation afforded by E_v^λ. Following [7], we say that $\mathfrak{X}^\lambda$ is balanced if

$$v^{\mathbf{a}_\lambda} \mathfrak{X}^\lambda(T_w) \in \mathscr{O}^{d_\lambda \times d_\lambda} \qquad \text{for all } w \in W.$$

This concept plays a crucial role in the study of "cellular structures" on $\mathscr{H}$ (see [7]) and the determination of Kazhdan–Lusztig cells (see [8, §4]). It is known that every E_v^λ affords a balanced representation. Note that, given some matrix representation afforded by E_v^λ, the above condition is hard to verify since it involves representing matrices for *all* $w \in W$. Much better for practical purposes is the following condition.

Proposition 2.5 (See [7, Prop. 4.3, Remark 4.4]). *Assume that $F \subseteq \mathbb{R}$. Let $\lambda \in \Lambda$ and $\mathfrak{X}^\lambda : \mathscr{H}_K \to K^{d_\lambda \times d_\lambda}$ be a matrix representation afforded by E_v^λ. Then $\mathfrak{X}^\lambda$ is balanced if and only if there exists a symmetric matrix $\Omega^\lambda \in \mathrm{GL}_{d_\lambda}(\mathscr{O})$ such that*

$$\Omega^\lambda\, \mathfrak{X}^\lambda(T_s) = \mathfrak{X}^\lambda(T_s)^{\mathrm{tr}}\, \Omega^\lambda \qquad \textit{for all } s \in S. \tag{$*$}$$

Remark 2.6. Note that, if a matrix Ω^λ satisfies $(*)$, then it immediately follows that

$$\Omega^\lambda\, \mathfrak{X}^\lambda(T_{w^{-1}}) = \mathfrak{X}^\lambda(T_w)^{\mathrm{tr}}\, \Omega^\lambda \qquad \text{for all } w \in W.$$

Thus, Ω^λ is the Gram matrix of a symmetric bilinear form $\langle\ ,\ \rangle_\lambda : E_v^\lambda \times E_v^\lambda \to K$ which is $\mathscr{H}_K$-invariant in the sense that

$$\langle T_w.e, e'\rangle_\lambda = \langle e, T_{w^{-1}}.e'\rangle_\lambda \qquad \text{for all } e, e' \in E_v^\lambda \text{ and } w \in W.$$

Remark 2.7. Assume that $F \subseteq \mathbb{R}$. Let $\lambda \in \Lambda$ and $\mathfrak{X}^\lambda : \mathscr{H}_K \to K^{d_\lambda \times d_\lambda}$ be a matrix representation afforded by E_v^λ. Let $\mathscr{E}(\mathfrak{X}^\lambda)$ be the set of all $P \in K^{d_\lambda \times d_\lambda}$ such that $P\,\mathfrak{X}^\lambda(T_s) = \mathfrak{X}^\lambda(T_s)^{\mathrm{tr}} P$ for $s \in S$. Since $\mathfrak{X}^\lambda$ is irreducible, Schur's Lemma implies that all matrices in $\mathscr{E}(\mathfrak{X}^\lambda)$ are scalar multiples of each other. By [10, Remark 1.4.9], there is a specific element $P_0 \in \mathscr{E}(\mathfrak{X}^\lambda)$ given by

$$P_0 := \sum_{w \in W} \mathfrak{X}^\lambda(T_w)^{\mathrm{tr}}\, \mathfrak{X}^\lambda(T_w) \in K^{d_\lambda \times d_\lambda};$$

furthermore, we have $\det(P_0) \neq 0$. By the Schur Relations (see [12, 7.2.1]), we have

$$\sum_{w \in W} \mathfrak{X}^\lambda(T_{w^{-1}}) P_0^{-1} \mathfrak{X}^\lambda(T_w) = \mathrm{trace}(P_0^{-1}) \mathbf{c}_\lambda I_{d_\lambda}.$$

Using the relation $P_0\,\mathfrak{X}^\lambda(T_{w^{-1}}) = \mathfrak{X}^\lambda(T_w)^{\mathrm{tr}} P_0$ for all $w \in W$, we deduce that

$$\mathrm{trace}(P_0^{-1}) \mathbf{c}_\lambda = 1.$$

This provides a direct criterion for checking if a given matrix $P \in \mathscr{E}(\mathfrak{X}^\lambda)$ equals P_0. Furthermore, if $P \neq 0$ is an element of $\mathscr{E}(\mathfrak{X}^\lambda)$, then $P = cP_0$ for some $0 \neq c \in K$ and so $\mathbf{c}_\lambda \mathrm{trace}(P^{-1}) P = \mathbf{c}_\lambda \mathrm{trace}(P_0^{-1}) P_0 = P_0$.

The following concept was introduced by Kazhdan–Lusztig [17] in the equal parameter case (where $L(s) = 1$ for all $s \in S$); for the general case see [10, §1.4].

Definition 2.8. Let V be an $\mathscr{H}_K$-module with $d := \dim V < \infty$. We say that V is *afforded by a W-graph* if there exist

- a basis $\{e_1,\ldots,e_d\}$ of V,
- subsets $I_i \subseteq S$ for $1 \leqslant i \leqslant d$,
- and elements $m_{ij}^s \in A$, where $1 \leqslant i,j \leqslant d$ and $s \in I_i \setminus I_j$,

such that the following hold. First, we require that

$$v^{L(s)} m_{ij}^s \in vR[v] \quad \text{and} \quad m_{ij}^s = m_{ij}^s|_{v\mapsto v^{-1}} \quad \text{for all } 1 \leqslant i,j \leqslant d,\, s \in I_i \setminus I_j.$$

Furthermore, for $s \in S$, the action of T_s on V is given by

$$T_s.e_j = \begin{cases} v^{L(s)}\, e_j + \displaystyle\sum_{1\leqslant i\leqslant d:\, s\in I_i} m_{ij}^s e_i & \text{if } s \notin I_j, \\ -v^{-L(s)}\, e_j & \text{if } s \in I_j. \end{cases}$$

Thus, if V is afforded by a W-graph representation, then the action of T_s on V is given by matrices of a particularly simple form.

It has been conjectured in [7] (see also [10, 1.4.14]) that, if the simple $\mathscr{H}_K$-module E_v^λ is afforded by a W-graph, then the corresponding matrix representation is balanced. We now turn to the problem of explicitly verifying if a given irreducible matrix representation of $\mathscr{H}_K$ is balanced or not.

2.9. We shall assume from now that W is a finite Weyl group and that we are in the equal parameter case where $L(s)=1$ for all $s \in S$; we may take $R=\mathbb{Z}$, $F=\mathbb{Q}$ in the above discussion. (The remaining cases have been dealt with in [7, Examples 4.5, 4.6].) It is known that every simple $\mathscr{H}_K$-module E_v^λ is afforded by a W-graph; see [10, Theorem 2.7.2] and the references there. As far as W of exceptional type is concerned, such W-graphs have been determined explicitly, by Naruse [23], Howlett and Yin [15], [16]. They are available in electronic form through Michel's development version of the CHEVIE system; see [21]. Now let us fix $\lambda \in \Lambda$ and assume that $\mathfrak{X}^\lambda \colon \mathscr{H}_K \to K^{d_\lambda \times d_\lambda}$ is a corresponding representation afforded by a W-graph. Concretely, this will mean that we are given the collection of matrices $\{X_s := \mathfrak{X}^\lambda(T_s) \mid s \in S\}$. Our aim is to find a matrix $P=(p_{ij})_{1\leqslant i,j\leqslant d_\lambda}$ such that

$$PX_s = X_s^{\mathrm{tr}} P \qquad \text{for all } s \in S. \tag{$*$}$$

This is a system of $|S|d_\lambda^2$ homogeneous linear equations for the $d_\lambda(d_\lambda+1)/2$ unknown entries of P. (Recall that P is symmetric.) We know that P is uniquely determined up to scalar multiples. Rescaling a given solution by a suitable non-zero polynomial in $\mathbb{Q}[v]$, we can assume that all entries of P are in $\mathbb{Z}[v]$ and that their greatest common divisor is ± 1; then P is unique up to a sign. Such a solution P will be called a *primitive Gram matrix* for $\mathfrak{X}^\lambda$. As in 2.7, a specific solution P_0 can be singled out by the condition that $\mathrm{trace}(P_0^{-1})\mathbf{c}_\lambda = 1$. We claim that

- the matrix $P_0' := v^{2l(w_0)} P_0$ has entries in $\mathbb{Z}[v]$, and
- the non-zero entries of P_0' have degree at most $2l(w_0)$.

Here, w_0 denotes the longest element of W. Indeed, since all the entries of the matrices X_s ($s \in S$) are in $\mathbb{Z}[v,v^{-1}]$, the same will be true for P_0 as well. The formulae in 2.8 show that each matrix vX_s ($s \in S$) has entries in $\mathbb{Z}[v]$. Hence, all matrices $v^{l(w_0)}\mathfrak{X}^\lambda(T_w)$ have entries in $\mathbb{Z}[v]$ and so P_0' has entries in $\mathbb{Z}[v]$. Furthermore, the non-zero entries of each matrix vX_s have degree 0, 1 or 2. This yields the degree bound for the entries of P_0'.

Since the entries of P_0' are integer polynomials of bounded degree, we can determine P_0' by interpolation and modular techniques (Chinese remainder). Combining this with the techniques described in [11, §4.3], one obtains an algorithm which can be implemented in GAP in a straightforward way. Rescaling these matrices by suitable non-zero polynomials in $\mathbb{Q}[v]$, we obtain primitive Gram matrices as solutions of $(*)$. This approach readily produces primitive Gram matrices for W of type F_4, E_6 and E_7 in a few hours of computing time. As was already advertised in Section 1, we also succeeded in obtaining primitive Gram matrices for type E_8, where it is one of the purposes of this paper to describe the methods involved.

Tables 1 and 2 contain some information about these primitive Gram matrices P:

1st column: usual names of the irreducible representations.
2nd column: maximum degree of a non-zero entry of P.
3rd column: maximum absolute value of a coefficient of an entry of P.
4th column: is the specialized matrix $P|_{v\to 0}$ diagonal?
5th column: prime divisors of the determinant of $P|_{v\to 0}$.
(No entry means that this determinant is ± 1.)

We note that the primes in the 5th column are so-called "bad primes" for W (as in [10, 1.5.11]). In particular, the fact that $P|_{v\to 0}$ always has a non-zero determinant means that $\det(P) \in \mathscr{O}^\times$ (see Proposition 2.5). Thus, we can conclude:

Theorem 2.10. *Let W be of type F_4, E_6, E_7 or E_8 and $L(s) = 1$ for all $s \in S$. Then the W-graph representations of Naruse [23], Howlett and Yin [15], [16] are balanced.*

3 Linear algebra over the integers

As was already mentioned in Section 1, the basic strategy of our approach to determine Gram matrices of invariant bilinear forms for representations of Iwahori–Hecke algebras is to reduce computational linear algebra over the polynomial rings $\mathbb{Z}[X]$ or $\mathbb{Q}[X]$, where from now on X denotes our favorite indeterminate, to computational linear algebra over the integers $\mathbb{Z}$. Thus in this section we begin by describing how we deal with matrices over $\mathbb{Z}$, where we restrict ourselves to the aspects needed for our present application.

Let us fix the following convention: For $x, y \in \mathbb{Z}$, not both zero, let $\gcd(x,y) \in \mathbb{Z}$ denote the positive greatest common divisor of x and y. A vector $0 \neq v \in \mathbb{Q}^m$, where $m \in \mathbb{N}$, is called *primitive*, if actually $v \in \mathbb{Z}^m$, and for the greatest common divisor $\gcd(v)$ of its entries we have $\gcd(v) = 1$. Clearly greatest common divisor computations in $\mathbb{Z}$ yield a $\mathbb{Q}$-multiple of v which is primitive. Similarly, a matrix $0 \neq A \in \mathbb{Z}^{m\times n}$, where $m, n \in \mathbb{N}$, is called *primitive*, if actually $A \in \mathbb{Z}^{m\times n}$, and for the greatest common divisor $\gcd(A)$ of its entries we have $\gcd(A) = 1$.

Table 1 Information on primitive Gram matrices for type F_4, E_6, E_7; cf. 2.9

F_4	deg.	abs. val.	diag.	det
1_1	0	1	y	
1_2	0	1	y	
1_3	0	1	y	
1_4	0	1	y	
2_1	2	1	y	
2_2	2	1	y	
2_3	2	1	y	
2_4	2	1	y	
4	4	2	y	
9_1	4	2	y	
9_2	6	8	y	2
9_3	6	12	y	2
9_4	10	6	y	
6_1	4	2	y	
6_2	4	4	y	2
12	8	54	y	2,3
4_1	2	2	y	2
4_2	2	1	y	
4_3	2	1	y	
4_4	6	4	y	2
8_1	4	2	y	
8_2	6	3	y	
8_3	4	2	y	
8_4	6	3	y	
16	8	16	y	2

E_6	deg.	abs. val.	diag.	det
1_p	0	1	y	
$1_p'$	0	1	y	
10_s	6	3	y	
6_p	2	1	y	
$6_p'$	10	5	y	
20_s	6	3	y	
15_p	4	2	y	
$15_p'$	8	4	y	
15_q	6	3	y	
$15_q'$	8	8	y	
20_p	4	2	y	
$20_p'$	22	61	y	
24_p	6	5	y	
$24_p'$	12	20	y	
30_p	6	6	y	2
$30_p'$	18	304	y	2
60_s	10	26	y	
80_s	14	711	y	2,3
90_s	12	58	n	3
60_p	10	21	y	
$60_p'$	12	44	y	
64_p	8	12	y	
$64_p'$	20	192	y	
81_p	10	24	y	
$81_p'$	12	32	y	

E_7	deg.	abs. val.	diag.	det
1_a	0	1	y	
$1_a'$	0	1	y	
7_a	12	6	y	
$7_a'$	2	1	y	
15_a	8	8	y	
$15_a'$	6	3	y	
21_a	4	2	y	
$21_a'$	10	7	y	
21_b	16	19	y	
$21_b'$	4	2	y	
27_a	4	2	y	
$27_a'$	32	164	y	
35_a	8	6	y	
$35_a'$	6	3	y	
35_b	6	3	y	
$35_b'$	22	144	y	
56_a	26	1082	y	2
$56_a'$	6	6	y	2
70_a	12	56	y	
$70_a'$	10	26	y	
84_a	10	26	y	
$84_a'$	16	148	y	
105_a	22	377	y	
$105_a'$	8	12	y	
105_b	10	21	y	
$105_b'$	20	504	y	
105_c	12	38	y	
$105_c'$	12	44	y	
120_a	8	24	y	2
$120_a'$	30	7516	y	2

E_7	deg.	abs. val.	diag.	det
168_a	10	35	y	
$168_a'$	26	2193	y	
189_a	12	56	y	
$189_a'$	16	112	y	
189_b	28	7498	y	
$189_b'$	10	42	y	
189_c	22	454	y	
$189_c'$	10	38	y	
210_a	10	35	y	
$210_a'$	22	973	y	
210_b	14	253	y	
$210_b'$	16	468	y	
216_a	22	1596	y	
$216_a'$	14	227	y	
280_a	22	1836	n	3
$280_a'$	12	58	n	3
280_b	14	241	y	
$280_b'$	20	2368	y	
315_a	26	47277	y	2,3
$315_a'$	14	4122	y	2,3
336_a	20	892	y	
$336_a'$	14	175	y	
378_a	24	7310	y	
$378_a'$	14	453	y	
405_a	14	637	y	2
$405_a'$	26	46878	y	2
420_a	16	1332	y	2
$420_a'$	20	4148	y	2
512_a	20	6036	y	
$512_a'$	20	6036	y	

Table 2 Information on primitive Gram matrices for type E_8; cf. 2.9

repr.	deg.	abs. val.	diag.	det
1_x	0	1	y	
$1'_x$	0	1	y	
28_x	4	2	y	
$28'_x$	12	10	y	
35_x	4	2	y	
$35'_x$	38	377	y	
70_y	8	6	y	
50_x	8	6	y	
$50'_x$	22	257	y	
84_x	6	3	y	
$84'_x$	38	675	y	
168_y	16	340	y	
175_x	12	52	y	
$175'_x$	20	992	y	
210_x	8	24	y	2
$210'_x$	42	95780	y	2
420_y	16	1432	y	
300_x	10	41	y	
$300'_x$	40	12710	y	
350_x	12	56	y	
$350'_x$	20	290	y	
525_x	12	76	y	
$525'_x$	24	1946	y	
567_x	10	54	y	
$567'_x$	42	57812	y	
1134_y	22	8739	y	
700_xx	18	1399	y	
$700_xx'$	20	5982	y	

repr.	deg.	abs. val.	diag.	det
700_x	12	538	y	2
$700'_x$	54	16489188	y	2
1400_y	22	22286	n	2, 3
840_x	16	6044	y	
$840'_x$	26	37603	y	
1680_y	22	3447	n	2, 5
972_x	16	2098	y	
$972'_x$	36	185342	y	
1050_x	16	3792	y	
$1050'_x$	34	390765	y	
2100_y	22	5561	y	
1344_x	14	1140	y	
$1344'_x$	40	381082	y	
2688_y	24	169180	y	
1400_x	16	41820	y	2, 3
$1400'_x$	48	763453596	y	2, 3
1575_x	14	783	n	3
$1575'_x$	44	850956	n	3
3150_y	26	6166994	y	2
2100_x	20	3514	y	
$2100'_x$	26	12511	y	
4200_y	28	58249760	n	2
2240_x	20	1878156	y	2
$2240'_x$	42	60390945	y	2
4480_y	32	85556320920	y	2, 3, 5
2268_x	16	5948	y	2
$2268'_x$	40	6442224	y	2
4536_y	28	3887856	n	2

repr.	deg.	abs. val.	diag.	det
2835_x	24	1344484	y	
$2835'_x$	32	5391418	y	
5670_y	30	10762741	n	2, 3, 5
3200_x	24	266284	y	
$3200'_x$	30	587345	y	
4096_x	22	531634	y	
$4096'_x$	44	234956568	y	
4200_x	24	5413484	y	2
$4200'_x$	36	129331224	y	2
6075_x	26	894864	y	
$6075'_x$	34	10488013	y	
8_z	2	1	y	
$8'_z$	14	6	y	
56_z	6	3	y	
$56'_z$	10	7	y	
112_z	6	6	y	2
$112'_z$	54	20790	y	2
160_z	8	12	y	
$160'_z$	32	400	y	
448_w	16	128	y	
400_z	12	132	y	
$400'_z$	38	58368	y	
448_z	12	290	y	
$448'_z$	32	17290	y	
560_z	10	73	y	
$560'_z$	46	408409	y	
1344_w	24	177956	y	
840_z	14	643	y	

repr.	deg.	abs. val.	diag.	det
$840'_z$	26	8048	y	
1008_z	12	156	n	3
$1008'_z$	40	66780	n	3
2016_w	28	797422	y	
1296_z	14	345	y	
$1296'_z$	34	23195	y	
1400_{zz}	16	10042	y	
$1400'_{zz}$	34	358379	y	
1400_z	14	8148	y	2, 3
$1400'_z$	50	60122676	y	2, 3
2400_z	22	6380	y	
$2400'_z$	28	55922	y	
2800_z	20	38038	y	2
$2800'_z$	30	882222	y	2
5600_w	26	372230	n	3
3240_z	16	25586	y	
$3240'_z$	48	33653538	y	
3360_z	20	29722	y	
$3360'_z$	32	775084	y	
7168_w	32	1190470476	y	2, 3
4096_z	22	531634	y	
$4096'_z$	44	234956568	y	
4200_z	26	728053	y	
$4200'_z$	28	1298612	y	
4536_z	24	2728756	y	
$4536'_z$	38	50779421	y	
5600_z	26	3115126	y	2
$5600'_z$	30	3848044	y	2

3.1. Continued fractions and the Euclidean algorithm. The first computational task we are going to discuss, in Section 3.4 below, is rational number recovery. This has been discussed in the literature at various places, see for example [3, 22, 25] or [5, Section 5.10]. (We also gratefully acknowledge additional private discussions with R. Parker on this topic.) Although the ideas pursued in these references are closely related to ours, none of them completely coincides with our approach, and proofs (if given at all) are not too elucidating. Hence we present our approach in detail, for which we need a few preparations first:

Continued fraction expansions. We recall a few notions from the theory of continued fraction expansions; as a general reference see for example [14, Chapter 10]: Given $\rho \in \mathbb{R}$ such that $\rho \geqslant 0$, let

$$\mathrm{cf}[q_1, q_2, \ldots] = q_1 + \cfrac{1}{q_2 + \cfrac{1}{\ddots}}$$

be its *(regular) continued fraction* expansion, where $q_1 \in \mathbb{N}_0$ and $q_i \in \mathbb{N}$ for $i \geqslant 2$. This is obtained by letting $q_1 := \lfloor \rho \rfloor$, and, as long as $\rho \neq q_1$, proceeding recursively with $\frac{1}{\rho - q_1}$ instead of ρ. This process terminates, after $l \geqslant 1$ steps say, if and only if $\rho \in \mathbb{Q}$; otherwise we let $l := \infty$. Truncating at $i \leqslant l$ yields the i-th *convergent* $\rho_i := \mathrm{cf}[q_1, \ldots, q_i] \in \mathbb{Q}$ of ρ, hence we may write $\rho_i := \frac{\sigma_i}{\tau_i}$, where $\sigma_i, \tau_i \in \mathbb{N}_0$ such that $\tau_i \geqslant 1$ and $\gcd(\sigma_i, \tau_i) = 1$. Letting additionally $\sigma_{-1} := 0$ and $\tau_{-1} := 1$, as well as $\sigma_0 := 1$ and $\tau_0 := 0$, for $i \geqslant 1$ we get by induction

$$\sigma_i = q_i \sigma_{i-1} + \sigma_{i-2} \quad \text{and} \quad \tau_i = q_i \tau_{i-1} + \tau_{i-2}.$$

Hence the sequences $[\sigma_1, \sigma_2, \ldots, \sigma_l]$ and $[\tau_2, \tau_3, \ldots, \tau_l]$ are strongly increasing.

Now let $\rho = \frac{a}{b} \in \mathbb{Q}$, where $a, b \in \mathbb{N}$. Then the continued fraction expansion of ρ can be computed by the extended Euclidean algorithm, see [1, Algorithm 1.3.6], as follows: Setting $r_0 := a$ and $r_1 := b$, for $1 \leqslant i \leqslant l$ let recursively $q_i \in \mathbb{N}_0$ and

$$r_{i+1} := r_{i-1} - q_i r_i \in \mathbb{N}_0 \quad \text{such that} \quad r_{i+1} < r_i,$$

where $l \geqslant 1$ is defined by $r_l > 0$ but $r_{l+1} = 0$; actually we have $q_i \geqslant 1$ for $i \geqslant 2$, and of course $r_l = \gcd(a, b)$. Hence the sequence $[r_1, \ldots, r_{l+1}]$ has non-negative entries and is strongly decreasing. Moreover, setting $s_0 := 1$ and $t_0 := 0$, as well as $s_1 := 0$ and $t_1 := 1$, and for $1 \leqslant i \leqslant l$ letting recursively

$$s_{i+1} := s_{i-1} - q_i s_i \quad \text{and} \quad t_{i+1} := t_{i-1} - q_i t_i,$$

we get $r_i = s_i a + t_i b$. Then it is immediate by induction that $\sigma_i = (-1)^i \cdot t_{i+1}$ and $\tau_i = (-1)^{i+1} \cdot s_{i+1}$, for $i \geqslant 1$, and hence

$$\rho_i = -\frac{t_{i+1}}{s_{i+1}}, \quad \text{where} \quad \gcd(s_{i+1}, t_{i+1}) = 1, \quad \text{for} \quad 1 \leqslant i \leqslant l.$$

Hence the sequences $[-s_3, s_4, -s_5 \ldots, \pm s_{l+1}]$ and $[-t_2, t_3, -t_4 \ldots, \pm t_{l+1}]$ have positive entries and are strongly increasing. Finally, a direct computation yields

$$\rho - \rho_i = \frac{a}{b} - \frac{\sigma_i}{\tau_i} = \frac{\tau_i a - \sigma_i b}{\tau_i b} = \frac{s_{i+1}a + t_{i+1}b}{s_{i+1}b} = \frac{r_{i+1}}{bs_{i+1}}, \quad \text{for} \quad 1 \leqslant i \leqslant l.$$

Another view on the Euclidean algorithm. For $a, b \in \mathbb{N}$ we consider the $\mathbb{Z}$-lattice

$$L_{a,b} := \langle [1,a], [0,b] \rangle_{\mathbb{Z}} \subseteq \mathbb{Z}^2.$$

Then we have $|\det(L_{a,b})| = b$, and it is immediate that $[x,y] \in \mathbb{Z}^2$ is an element of $L_{a,b}$ if and only if $y \equiv ax \pmod{b}$. Note that if $0 \neq [x,y] \in L_{a,b}$ is primitive, then we necessarily have $\gcd(x,b) = 1$. Moreover, the extended Euclidean algorithm shows that $L_{a,b} = \langle [s_i, r_i], [s_{i+1}, r_{i+1}] \rangle_{\mathbb{Z}}$, for all $0 \leqslant i \leqslant l$. We collect a few properties of $L_{a,b}$:

Lemma 3.2. *(a) For all $0 \leqslant i \leqslant l+1$ we have $\langle [s_i, r_i] \rangle_{\mathbb{Q}} \cap L_{a,b} = \langle [s_i, r_i] \rangle_{\mathbb{Z}}$.*
(b) We have $\langle [s_i, r_i] \rangle_{\mathbb{Q}} = \langle [s_j, r_j] \rangle_{\mathbb{Q}}$, where $1 \leqslant i, j \leqslant l+1$, if and only if $i = j$.

Proof. We first show that whenever $[x,y] \in L_{a,b}$ such that $0 < |y| < r_i$, for some $0 \leqslant i \leqslant l$, then $|x| \geqslant |s_{i+1}|$: We may assume that $i \geqslant 2$. Let $c, d \in \mathbb{Z}$ such that

$$[x,y] = [c,d] \cdot \begin{bmatrix} s_i & r_i \\ s_{i+1} & r_{i+1} \end{bmatrix},$$

where we may assume that $c \neq 0$, which entails $d \neq 0$ as well. Since $r_i > r_{i+1} \geqslant 0$, this implies $c \cdot d < 0$. Since the sequence $[s_2, -s_3, s_4, -s_5 \ldots, \pm s_{l+1}]$ has positive entries, we get $|x| = |cs_i + ds_{i+1}| = |c| \cdot |s_i| + |d| \cdot |s_{i+1}| \geqslant |s_{i+1}|$, as asserted.

(a) We may assume that $i \geqslant 2$. Moreover, for $i = l+1$ letting $[x,0] \in L_{a,b}$, it is immediate from $ax \equiv 0 \pmod{b}$ that $|s_{l+1}| = \frac{b}{r_l} = \frac{b}{\gcd(a,b)}$ divides x. Hence we may assume $i \leqslant l$, too. Then let $d \neq 1$ be a divisor of $\gcd(s_i, r_i)$ such that $\frac{1}{d} \cdot [s_i, r_i] \in L_{a,b}$. Then we have $0 < |\frac{r_i}{d}| < r_i$ and $|\frac{s_i}{d}| < |s_i| \leqslant |s_{i+1}|$, contradicting the statement above.

(b) It follows from (a) that there are $c, d \in \mathbb{Z}$ such that $[s_j, r_j] = c \cdot [s_i, r_i]$ and $[s_i, r_i] = d \cdot [s_j, r_j]$. Hence we get $cd = 1$, and since the sequence $[r_1, \ldots, r_{l+1}]$ has non-negative entries and is strongly decreasing, we infer $r_i = r_j$ and $i = j$. $\square$

Note that the statement in (b) is trivial if $[s_i, r_i]$ is primitive, that is $\gcd(s_i, r_i) = 1$. But this is not always fulfilled, as the example in [5, Example 5.27] shows.

Proposition 3.3. *(a) Let $[x,y] \in L_{a,b}$ such that $x \neq 0$ and $|x| \cdot |y| \leqslant \frac{b}{2}$. Then we have $[x,y] \in \langle [s_i, r_i] \rangle_{\mathbb{Z}}$, for a unique $2 \leqslant i \leqslant l+1$. In particular, if $[x,y]$ is primitive then we have $[x,y] = [s_i, r_i]$ or $[x,y] = -[s_i, r_i]$.*

(b) Assume there is $0 \neq [x,y] \in L_{a,b}$ such that $\|[x,y]\| := \sqrt{x^2 + y^2} < \sqrt{b}$. Then there is a unique $2 \leqslant i \leqslant l+1$ such that $\|[s_i, r_i]\| < \sqrt{b}$, and the shortest non-zero elements of $L_{a,b}$ are precisely $[s_i, r_i]$ and $-[s_i, r_i]$.

Proof. (a) Since $[x,y] \in L_{a,b}$ there is $z \in \mathbb{Z}$ such that $y = xa - zb$. Then we have

$$\left|\frac{a}{b}-\frac{z}{x}\right|=\frac{|y|}{b\cdot|x|}=\frac{|x|\cdot|y|}{b\cdot|x|^2}\leqslant\frac{1}{2\cdot|x|^2}.$$

Thus by Legendre's Theorem, see [14, Section 10.15, Theorem 184], we infer that $\frac{z}{x}$ occurs as a convergent in the continued fraction expansion of $\rho=\frac{a}{b}$, that is, there is $2\leqslant i\leqslant l+1$ such that $\frac{z}{x}=\rho_{i-1}$. This yields

$$\frac{y}{x}=\frac{xa-zb}{x}=a-\frac{zb}{x}=a-b\rho_{i-1}=b(\rho-\rho_{i-1})=\frac{r_i}{s_i}.$$

Hence we have $[x,y]\in\langle[s_i,r_i]\rangle_{\mathbb{Q}}$, and thus from Lemma 3.2 we get $[x,y]\in\langle[s_i,r_i]\rangle_{\mathbb{Z}}$, together with the uniqueness statement.

(b) Assume first that $x=0$, then by Lemma 3.2 we infer that b divides y, and hence $\|[x,y]\|\geqslant b\geqslant\sqrt{b}$, a contradiction. Hence we have $x\neq 0$. Moreover, from $(x-y)^2=x^2+y^2-2xy\geqslant 0$ we get $2\cdot|x|\cdot|y|\leqslant x^2+y^2=\|[x,y]\|^2<b$, hence from (a) we see that there is $2\leqslant i\leqslant l+1$ such that $[x,y]=\langle[s_i,r_i]\rangle_{\mathbb{Z}}$. Thus in particular we have $\|[s_i,r_i]\|<\sqrt{b}$.

In order to show uniqueness, and the statement on shortest elements, let $0\neq[x',y']\in L_{a,b}$ such that $\|[x',y']\|<\sqrt{b}$. Then, as above, there is $2\leqslant i\leqslant l+1$ such that $[x',y']=\langle[s_j,r_j]\rangle_{\mathbb{Z}}$, hence in particular we have $\|[s_j,r_j]\|<\sqrt{b}$. Then Hadamard's inequality, see [5, Theorem 16.6], implies that

$$\det\left(\begin{bmatrix}s_i & r_i\\ s_j & r_j\end{bmatrix}\right)\leqslant\|[s_i,r_i]\|\cdot\|[s_j,r_j]\|<b.$$

Since $|\det(L_{a,b})|=b$ divides $\det\left(\begin{bmatrix}s_i & r_i\\ s_j & r_j\end{bmatrix}\right)$ this entails $\langle[s_i,r_i]\rangle_{\mathbb{Q}}=\langle[s_j,r_j]\rangle_{\mathbb{Q}}$, and hence $i=j$ by Lemma 3.2. □

A comparison of the above treatment with the references already mentioned seems to be in order: The statement of Proposition 3.3(a) is roughly equivalent to [3, Theorem] and [22, Theorem 1], respectively. Alone, the proof given in [3] appears to be too concise, and provides a slightly worse bound for b to be large enough. And [22, Theorem 1] is attributed in turn to [2], while for a proof the reader is referred to [5]. Unfortunately, [5, Theorem 5.26] is not immediately conclusive for the statements under consideration here.

The main difference between the above-mentioned approaches and ours is the break condition used to actually determine the index i referred to in Proposition 3.3(a): In [2, 3, 5] a bound on the residues r_i is used, while in [22, Section 3] the quotients q_i are considered instead (yielding a randomized algorithm). In contrast, in our decisive Proposition 3.3(b) we are using the minimum of the lattice $L_{a,b}$, which hence treats both the r_i and s_i (in other words the the unknown numbers y and x) on a "symmetric" footing. To our knowledge, this point of view is new, its algorithmic relevance being explained below.

3.4. Recovering rational numbers. We are now prepared to describe our first computational task, which will appear both in computations over $\mathbb{Z}$ in Section 3.5, and over the polynomial ring $\mathbb{Q}[X]$ in Section 4.2:

Let $x \in \mathbb{N}$ and $0 \neq y \in \mathbb{Z}$ such that $\gcd(x,y) = 1$. Assume we are given $a, b \in \mathbb{N}$ such that $\gcd(x,b) = 1$ and $y \equiv ax \pmod{b}$; note that since x is invertible modulo b we may write $\frac{y}{x} \equiv a \pmod{b}$ instead, which we will feel free to do if convenient. Now, if b is large enough compared to x and $|y|$, the task is to recover $\frac{y}{x} \in \mathbb{Q}$ from its congruence class $a \pmod{b}$.

In view of Proposition 3.3(b), this is straightforward: Assuming that $x^2 + y^2 < b$, the $\mathbb{Z}$-lattice $L_{a,b} = \langle [1,a],[0,b] \rangle_{\mathbb{Z}} \subseteq \mathbb{Z}^2$ has precisely two shortest non-zero elements, namely the primitive elements $\pm[x,y]$. In other words, the rational number $\frac{y}{x} \in \mathbb{Q}$ can be found by computing a shortest non-zero element of $L_{a,b}$. This in turn can be done algorithmically by the Gauß reduction algorithm for $\mathbb{Z}$-lattices of rank 2, see [1, Algorithm 1.3.14]. Moreover, compared to the general case, for the particular lattice $L_{a,b}$ we have a better break condition: We may stop early as soon as we have found an element $[x,y] \in L_{a,b}$ such that $x^2 + y^2 < b$. If then $[x,y]$ is primitive, the rational number $\frac{y}{x}$ fulfills all assumptions made, where of course its correctness has to be verified independently. Otherwise, if $[x,y]$ is not primitive, or the shortest element $[x',y'] \in L_{a,b}$ found fulfills $x'^2 + y'^2 \geqslant b$, then we report failure. Thus, in practice, we choose b small, and rerun the above algorithm with b increasing, until we find a valid candidate passing independent verification.

At this stage, we should point out the algorithmic advantage of our approach, compared to the other ones mentioned: The latter refer to the convergents of continued fraction expansions, and thus to the full sequence of non-negative residues of the extended Euclidean algorithm. In contrast, the Gauß reduction algorithm to find a lattice minimum proceeds by iterated pair reduction, starting with the pair $[0,b]$ and $[1,a]$. Although this is essentially equivalent to running the extended Euclidean algorithm on a and b, here we are allowed to use best approximation. This amounts to using numerically smallest residues, instead of non-negative ones as was necessary in the context of continued fraction expansions. Although we have not carried out a detailed comparison, it is well-known that this saves a non-negligible amount of quotient and remainder steps.

3.5. Finding linear combinations. We are now going to describe *the* basic task we are faced with in order to be able to do computational linear algebra over $\mathbb{Z}$. To do so, we of course avoid the Gauß algorithm over $\mathbb{Q}$, but we also do not refer to pure "lattice algorithms", as they are called in [1, Section 2.1], for example those to compute Hermite normal forms or reduced lattice bases described in [1, Section 2.4–2.7]. Instead, we use a modular technique, which is a keystone to make use of the ideas of the MeatAxe in the framework of the IntegralMeatAxe. To our knowledge, this has only been discussed very briefly in the literature, for example in [3, 25]. Moreover, our approach differs from those cited, at least in detail; in particular, [3] only allows for regular square matrices.

To describe the computational task, we again need some preparations first: Given a (rectangular) matrix $A \in \mathbb{Z}^{m \times n}$, with $\mathbb{Q}$-linearly independent rows $w_1, \ldots, w_m \in \mathbb{Z}^n$, where $m, n \in \mathbb{N}$, let

$$L := \langle w_1, \ldots, w_m \rangle_{\mathbb{Z}} \leqslant \mathbb{Z}^n$$

be the $\mathbb{Z}$-lattice spanned by the rows of A, and let $L \leqslant \widehat{L} \leqslant \mathbb{Z}^n$ be its *pure closure* in $\mathbb{Z}^n$, that is the smallest pure $\mathbb{Z}$-sublattice of $\mathbb{Z}^n$ containing L. Then the index $\det(L) := [\widehat{L} \colon L]$ is finite; of course, if $m = n$ then we have $\det(L) = |\det(A)|$. Thus for any vector $v \in \mathbb{Z}^n$, we have $v \in \widehat{L}$ if and only of there is $a \in \mathbb{N}$ such that $av \in L$; in this case, if a is chosen minimal then it divides $\det(L)$.

Now, given $v \in \mathbb{Z}^n$, the task is to decide whether or not $v \in \widehat{L}$, and if this is the case to compute $a_1, \ldots, a_m \in \mathbb{Z}$ and $a \in \mathbb{N}$ such that $\gcd(a, a_1, \ldots, a_m) = 1$ and

$$v = \frac{1}{a} \cdot \sum_{j=1}^{m} a_j w_j = \frac{1}{a} \cdot [a_1, \ldots, a_m] \cdot A;$$

in this case a and the a_i are uniquely determined.

The p-adic decomposition algorithm. To do so, we choose a (large) prime p. Then reduction modulo p yields the matrix $\overline{A} \in \mathbb{F}_p^{m \times n}$ over the prime field $\mathbb{F}_p$. We assume that the rows $\overline{w}_1, \ldots, \overline{w}_m \in \mathbb{F}_p^n$ of $\overline{A}$ are $\mathbb{F}_p$-linearly independent as well; otherwise we choose another prime p. By the structure theory of finitely generated modules over principal ideal domains, this condition is equivalent to saying $\overline{\widehat{L}} = \overline{L}$, which in turn is equivalent to p not dividing $\det(L)$. In particular, the independence condition on $\overline{w}_1, \ldots, \overline{w}_m \in \mathbb{F}_p^n$ is fulfilled for all but finitely many primes p.

Thus we have $v \in \widehat{L}$ if and only if $\overline{v} \in \overline{L} = \langle \overline{w}_1, \ldots, \overline{w}_m \rangle_{\mathbb{F}_p}$, solving the decision problem. Furthermore, if $v \in \widehat{L}$ then set $v_0 := v$, and for $d \in \mathbb{N}_0$ proceed successively as follows: Since $v_d \in \widehat{L}$, there are $[a_{d,1}, \ldots, a_{d,m}] \in \mathbb{Z}^m$ such that $-\frac{p}{2} < a_{d,j} \leqslant \frac{p}{2}$ for all $1 \leqslant j \leqslant m$, and

$$\overline{v}_d = \sum_{j=1}^{m} \overline{a}_{d,1} \overline{w}_j = [\overline{a}_{d,1}, \ldots, \overline{a}_{d,m}] \cdot \overline{A} \in \mathbb{F}_p^n.$$

Then we let

$$v_{d+1} := \frac{1}{p} \cdot \Big(v_d - [a_{d,1}, \ldots, a_{d,m}] \cdot A \Big) \in \mathbb{Z}^n.$$

Hence we have $v_{d+1} \in \widehat{L}$ as well, and we may recurse. This yields

$$v \equiv \Big(\sum_{i=0}^{d} p^i \cdot [a_{i,1}, \ldots, a_{i,m}] \Big) \cdot A \equiv \Big[\sum_{i=0}^{d} p^i a_{i,1}, \ldots, \sum_{i=0}^{d} p^i a_{i,m} \Big] \cdot A \pmod{p^{d+1}\mathbb{Z}^n},$$

or equivalently

$$\frac{a_j}{a} \equiv \sum_{i=0}^{d} p^i a_{i,j} \pmod{p^{d+1}}, \quad \text{for all} \quad 1 \leqslant j \leqslant m.$$

Thus, if $v \in L$, or equivalently $a = 1$, then since $-\frac{p^{d+1}}{2} < \sum_{i=0}^{d} p^i a_{i,j} \leqslant \frac{p^{d+1}}{2}$ there is some $d \in \mathbb{N}_0$ such that $v_{d+1} = 0$, implying that $a_j = \sum_{i=0}^{d} p^i a_{i,j}$, for all $1 \leqslant j \leqslant$

m, without further independent verification necessary. Otherwise, if $v \in \widehat{L} \setminus L$, then applying rational number recovery for some $d \in \mathbb{N}_0$ large enough, see Section 3.4, reveals the vector $\frac{1}{a} \cdot [a_1, \ldots, a_m] \in \mathbb{Q}^m$; note that under the assumptions made p does not divide a. In the latter case correctness is independently verified by computing $[a_1, \ldots, a_m] \cdot A \in \mathbb{Z}^n$ and checking whether it equals $av \in \mathbb{Z}^n$.

Modular computations. In practice, to check $\overline{w}_1, \ldots, \overline{w}_m \in \mathbb{F}_p^n$ for $\mathbb{F}_p$-linear independence, and to compute the vectors $[\overline{a}_{d,1}, \ldots, \overline{a}_{d,m}] \in \mathbb{F}_p^m$ we use ideas taken from the MeatAxe. In particular, in order to keep the depth d needed smallish, but still to be able to make efficient use of fast arithmetic over small finite prime fields, we choose the prime p amongst the largest primes smaller than $2^8 = 256$. (In our application we for example use $p = 251$ as the default prime.)

3.6. Nullspace. In the framework of the IntegralMeatAxe there is a general method to compute a $\mathbb{Z}$-basis of the row kernel of a matrix with entries in $\mathbb{Z}$, see [25]. But in view of the application to row kernels of matrices over $\mathbb{Q}[X]$ in Section 5.1, here we only deal with the following restricted nullspace problem:

Given a matrix $A \in \mathbb{Q}^{m \times n}$, where $m, n \in \mathbb{N}$, such that $\dim_{\mathbb{Q}}(\ker(A)) = 1$, where $\ker(A)$ denotes the row kernel of A, compute a primitive vector $v \in \mathbb{Z}^m$ such that $\ker(A) = \langle v \rangle_{\mathbb{Q}}$; then v is unique up to sign.

To do so, by going over to a suitable $\mathbb{Q}$-multiple we may assume that $A \in \mathbb{Z}^{m \times n}$. Let $w_1, \ldots, w_m \in \mathbb{Z}^n$ be the rows of A. We may assume that $w_1 \neq 0$, since otherwise we trivially set $v := [1, 0, \ldots, 0] \in \mathbb{Z}^m$. Then for $2 \leqslant i \leqslant m$ we successively check, using the p-adic decomposition algorithm in Section 3.5, whether or not $w_i \in \langle w_1, \ldots, w_{i-1} \rangle_{\mathbb{Q}}$. If this is not the case, that is $\{w_1 \ldots, w_i\}$ is $\mathbb{Q}$-linearly independent, then if $\overline{w}_1, \ldots, \overline{w}_i \in \mathbb{F}_p^n$ turns out to be $\mathbb{F}_p$-linearly independent we increment i, while otherwise we return failure in order to choose another prime p. If $\{w_1 \ldots, w_i\}$ is $\mathbb{Q}$-linearly dependent, then the p-adic decomposition algorithm returns $a_1, \ldots, a_{i-1} \in \mathbb{Z}$ and $a \in \mathbb{N}$ such that $\gcd(a, a_1, \ldots, a_{i-1}) = 1$ and $w_i = \frac{1}{a} \cdot \sum_{j=1}^{i-1} a_j w_j$. Thus $v := [a_1, \ldots, a_{i-1}, -a, 0, \ldots, 0] \in \ker(A) \leqslant \mathbb{Z}^m$ is primitive.

3.7. Inverse. Matrix inversion over $\mathbb{Q}$, from the point of view of reducing to computations over $\mathbb{Z}$ as much as possible, can be formulated as the following task:

Given a matrix $A \in \mathbb{Q}^{n \times n}$, where $n \in \mathbb{N}$, such that $\det(A) \neq 0$, compute $B \in \mathbb{Z}^{n \times n}$ and $c \in \mathbb{N}$, such that $A^{-1} = \frac{1}{c} \cdot B \in \mathbb{Q}^{n \times n}$ and the overall greatest common divisor $\gcd(B, c)$ of the entries of B and c equals $\gcd(B, c) = 1$; then (B, c) is unique.

To do so, by going over to a suitable $\mathbb{Q}$-multiple we may assume that $A \in \mathbb{Z}^{n \times n}$. Then the equation $BA = c \cdot E_n$, where E_n denotes the identity matrix, implies that $\gcd(B)$ divides c, and hence B is necessarily primitive. Solving the equations $\mathscr{X} A = E_n$, for the unknown matrix $\mathscr{X} \in \mathbb{Q}^{n \times n}$, amounts to writing the rows of the identity matrix as $\mathbb{Q}$-linear combinations of the rows of A, which is done using the p-adic decomposition algorithm in Section 3.5; recall that the rows of A indeed are assumed to be $\mathbb{Q}$-linearly independent.

3.8. The exponent of a matrix. Given a square matrix $A \in \mathbb{Z}^{n \times n}$ such that $\det(A) \neq 0$ as above, the number $c \in \mathbb{N}$ found in the expression $A^{-1} = \frac{1}{c} \cdot B$, where $B \in \mathbb{Z}^{n \times n}$ is chosen to be primitive, turns out to have another interpretation:

Let $\operatorname{im}(A) \leqslant \mathbb{Z}^n$ be the $\mathbb{Z}$-span of the rows of A. By the structure theory of finitely generated modules over principal ideal domains, the annihilator of the $\mathbb{Z}$-module $\mathbb{Z}^n/\operatorname{im}(A)$ is a non-zero ideal of $\mathbb{Z}$, the positive generator $\exp(A)$ of which is called the *exponent* of A. Moreover, $\exp(A)$ divides $\det(A)$, which in turn divides some power of $\exp(A)$. Thus the prime divisors of $\exp(A)$ are precisely the primes $p \in \mathbb{Z}$ such that $\overline{A} \in \mathbb{F}_p^{n \times n}$ is not invertible.

Now, actually $\exp(A)$ and c coincide: From $BA = c \cdot E_n$ we conclude that $(c\mathbb{Z})^n \leqslant \operatorname{im}(A)$, hence $\exp(A)$ divides c; conversely, since $(\exp(A) \cdot \mathbb{Z})^n \leqslant \operatorname{im}(A)$ there is $B' \in \mathbb{Z}^{n \times n}$ such that $B'A = \exp(A) \cdot E_n$, implying that $\exp(A) \cdot B = c \cdot B'$, which by the primitivity of B shows that c divides $\exp(A)$. In other words, computing the inverse of A as described in Section 3.7 also yields a method to compute $\exp(A)$.

4 Computing with polynomials

Having the necessary pieces of linear algebra over the integers in place, in this section we describe computational aspects of single polynomials, before we turn to linear algebra over polynomials rings in Section 5.

4.1. Polynomial arithmetic. As our general strategy is to use linear algebra over $\mathbb{Z}$ or $\mathbb{Q}$ to do linear algebra over $\mathbb{Z}[X]$ or $\mathbb{Q}[X]$, for all arithmetically heavy computations we recurse to $\mathbb{Z}$ or $\mathbb{Q}$. Consequently, for the remaining pieces of explicit computation in $\mathbb{Z}[X]$ or $\mathbb{Q}[X]$ we may use a simple straightforward approach:

We use our own standard arithmetic for polynomials over $\mathbb{Z}$ or $\mathbb{Q}$, where a polynomial $0 \neq f = \sum_{i=0}^{d} z_i X^i \in \mathbb{Q}[X]$ is just represented by its coefficient list $[z_0, \ldots, z_d] \in \mathbb{Q}^{d+1}$ of length $d+1$, where $d = \deg(f)$. Thus we avoid structural overhead as much as possible, and may use directly the facilities to handle row vectors provided by GAP. But we would like to stress that this is just tailored for our aim of doing linear algebra over polynomial rings, and not intended to become a new general-purpose polynomial arithmetic. For example, we are not providing asymptotically fast multiplication, as is for example described in [5, Section 8.3].

In particular, we only rarely need to compute polynomial greatest common divisors. Hence we avoid sophisticated (modular) techniques, as are for example described and compared in [5, Chapter 6], but we are content with a simple variant of the Euclidean algorithm: Assuming that the operands have integral coefficients, by going over to $\mathbb{Q}$-multiples if necessary, in order to avoid coefficient explosion we just use denominator-free pseudo-division as described in [1, Algorithm 3.1.2], and Collins's sub-resultant algorithm given in [1, Algorithm 3.3.1], albeit the latter without intermediate primitivisation.

On the other hand, we very often have to evaluate polynomials at various places, where our strategy is to use as few of these specializations as possible, so that evaluation at distinct places is done step by step. Thus we are not in a position to use multi-point evaluation techniques, as are for example described in [5, Section 10.1]. Hence we are just using the Horner scheme, which under these circumstances is well-known to need the optimal number of multiplications.

We now describe the special tasks needed to be solved in our approach:

4.2. Recovering polynomials. The aim is to recover a polynomial with rational coefficients, which we are able to evaluate at arbitrary integral places, from as few such evaluations (at "small" places) as possible. More precisely:

Let $0 \neq f := \sum_{i=0}^{d} z_i X^i \in \mathbb{Q}[X]$ be a polynomial of degree $d = \deg(f) \in \mathbb{N}_0$, having coefficients $z_i = \frac{y_i}{x_i} \in \mathbb{Q}$, where $x_i \in \mathbb{N}$ and $y_i \in \mathbb{Z}$ such that $\gcd(x_i, y_i) = 1$. Then the task is to find pairwise coprime places $b_1, \ldots, b_k \in \mathbb{Z} \setminus \{0, \pm 1\}$, for some (small) $k \in \mathbb{N}$, such that the degree d and the coefficients $z_0, \ldots, z_d$ of f can be computed from the values $f(b_1), \ldots, f(b_k) \in \mathbb{Q}$ alone. Note that, in particular, we do not assume that $k > d$, so that polynomial interpolation is not applicable. (Actually, in our application we often enough have $k \ll d$, where for example $k \sim 5$, but $d \lesssim 200$.)

To this end, let $b := \prod_{j=1}^{k} |b_j| \in \mathbb{N}$, and assume that we have $\gcd(x_i, b) = 1$ and $x_i^2 + y_i^2 < b$ for all $0 \leqslant i \leqslant d$. Hence the congruence classes $z_i \equiv \frac{y_i}{x_i} \pmod{b_j}$ and $f(b_j) \pmod{b_j}$ are well-defined, and for the constant coefficient of f we get

$$z_0 \equiv \sum_{i=0}^{d} z_i b_j^i \equiv f(b_j) \pmod{b_j}, \quad \text{for} \quad 1 \leqslant j \leqslant k.$$

Thus by the Chinese Remainder Theorem, see for example [1, Theorem 1.3.9], there is a unique congruence class $a \pmod{b}$, where $a \in \mathbb{Z}$, such that $a \equiv z_0 \pmod{b}$. To compute $a \in \mathbb{Z}$, we let $a_j \in \mathbb{Z}$ such that

$$f(b_j) \equiv a_j \pmod{b_j}, \quad \text{for} \quad 1 \leqslant j \leqslant k.$$

An application of Chinese remainder lifting in $\mathbb{Z}$ to the congruence classes $a_1 \pmod{b_1}, \ldots, a_k \pmod{b_k}$ yields the congruence class $a \pmod{b}$, and by our choice of b applying rational number recovery as described in Section 3.4 reveals $z_0 \in \mathbb{Q}$. Now we recurse to $\widetilde{f} := \frac{f - z_0}{X} \in \mathbb{Q}[X]$, whose value at the place b_j can of course be determined directly from $f(b_j)$ as $\widetilde{f}(b_j) = \frac{f(b_j) - z_0}{b_j} \in \mathbb{Q}$.

Chinese remainder lifting. Hence, apart from rational number recovery, the key computational task to be solved is to perform Chinese remainder lifting in $\mathbb{Z}$:

We are using the straightforward approach based on the extended Euclidean algorithm, as is described in [1, Section 1.3.3]. Since we are computing many lifts with respect to the same places $b_1, \ldots, b_k$, we make use of a precomputation step, as in [1, Algorithm 1.3.11]. But, since again for reasons of time and memory efficiency we are choosing small places b_j, the specially tailored approach in [1, Algorithm 1.3.11] to keep the intermediate numbers occurring small, at the expense of needing more multiplications, does not pay off as experiments show. Moreover, as we are computing the values $f(b_j)$ for $1 \leqslant j \leqslant k$ step by step, where even the number k of places is not determined in advance, we cannot take advantage of fast Chinese remainder lifting techniques, as are described for example in [5, Section 10.3], either.

Our strategy is to rerun the above algorithm with k increasing, choosing small integral $2 \leqslant b_1 < b_2 < \cdots < b_k$, and to discard quickly erroneous guesses by an independent verification, until the correct answer passing the verification is found. By the above discussion, this happens after finitely many iterations. Before that, if $b = |\prod_{j=1}^{k} b_j|$ is too small, or not coprime to all the denominators x_i, the Chinese remainder lifting process does not terminate, or it terminates with a wrong guess. To catch the first case, we impose a degree bound, and stop the lifting process with a failure message if it is exceeded, in order to increment k. (In our application, 200 turned out to be a suitable degree bound in all cases.)

To catch the second case, we only allow for denominators x_i dividing an imposed bound. This is justified, since rational number recovery as described in Section 3.4 is a trade-off between finding the numerator y and the denominator x of the rational number $\frac{y}{x}$ to be reconstructed: In practice, we typically encounter small denominators x and large numerators y, which escape the Gauß reduction algorithm if b is chosen too small, since then the latter tends to return a larger denominator $x' > x$ and a smaller numerator $|y'| < |y|$. (In our application, denominator bounds such as small 2-powers, or 12, or 20 turned out to be sufficient in all cases.)

4.3. Degree detection. We keep the setting of Section 4.2. The technique to be described now has arisen out of an attempt to determine the degree of f without determining its coefficients. Actually, it deals with the following more general situation (whose relevance for our computations will be explained in Section 4.5 below):

Assume that instead of the values $f(b_1),\ldots,f(b_k)$ we are only able to compute "rescaled values" $a_1 f(b_1),\ldots,a_k f(b_k) \in \mathbb{Q}$, with scalar factors $a_j \in \mathbb{Q}$ such that $a_j > 0$, which are only known to come from a finite pool $\mathscr{R}$ of positive rational numbers associated with f. Thus the task now becomes to find $k \in \mathbb{N}$ and coprime places $b_1,\ldots,b_k \in \mathbb{Z}\setminus\{0,\pm1\}$ as above, allowing to determine f up to some positive rational scalar multiple, that is to find $af \in \mathbb{Q}[X]$, for some $a \in \mathbb{Q}$ such that $a > 0$; note that this also determines all the quotients $\frac{a_j}{a}$.

To this end, we let $\alpha_1,\ldots,\alpha_d \in \mathbb{C}$ be the complex roots of f, and set $\mu := \max\{0,|\alpha_1|,\ldots,|\alpha_d|\}$. Moreover, since $\mathscr{R}$ is a finite set, we have

$$\delta := \min\{|\ln(a') - \ln(a)| \in \mathbb{R}; a, a' \in \mathscr{R}, a \neq a'\} > 0.$$

Now, let $k \geqslant 2$, and for the places $b_1,\ldots,b_k$ we additionally assume that

$$(1+2d)\cdot\mu < b_1 < \cdots < b_k \quad \text{and} \quad \ln(b_k) - \ln(b_1) < \delta;$$

hence, in particular, the $f(b_j)$ are non-zero and have the same sign. The necessity of these choices will become clear below. But this forces us to show that for all $k \geqslant 2$ and all $x > 0$ and $\delta > 0$ there actually exist pairwise coprime integers $b_1 < \cdots < b_k$ such that $x < b_1$ and $\ln\left(\frac{b_k}{b_1}\right) < \delta$. Indeed, we are going to show that the latter can always be chosen to be primes (where the mere existence proof to follow is impractical, but in practice considering small primes works well, see Example 4.4):

Let $p_0 < p_1 < \cdots$ be the sequence of all primes exceeding x, and assume to the contrary that for all k-subsets thereof, $q_1 < \cdots < q_k$ say, we have $\ln\left(\frac{q_k}{q_1}\right) \geqslant \delta$. Then we have $p_{k-1} \geqslant e^{\delta}\cdot p_0$, and thus $p_{j(k-1)} \geqslant e^{j\delta}\cdot p_0$, for all $j \in \mathbb{N}$. Using the prime number function $\pi(x) := |\{p \in \mathbb{N}; p \text{ prime}, p \leqslant x\}|$ this implies

$$\pi(e^{j\delta}\cdot p_0) \leqslant \pi(p_0)+j(k-1).$$

From this we get

$$\lim_{j\to\infty}\frac{\pi(e^{j\delta}\cdot p_0)\cdot\ln(e^{j\delta}\cdot p_0)}{e^{j\delta}\cdot p_0} \leqslant \lim_{j\to\infty}\frac{\big(\pi(p_0)+j(k-1)\big)\cdot\big(j\delta+\ln(p_0)\big)}{e^{j\delta}\cdot p_0} = 0,$$

contradicting the Prime Number Theorem, see [14, Section 1.8, Theorem 6], saying that $\lim_{x\to\infty}\frac{\pi(x)\cdot\ln(x)}{x}=1$.

Growth behavior of polynomials. We now consider the growth behavior of the polynomial f. For $x>\mu$ we have

$$\frac{\partial}{\partial x}\big(f(x)\big) = z_d\cdot\frac{\partial}{\partial x}\Big(\prod_{r=1}^{d}(x-\alpha_r)\Big) = f(x)\cdot\sum_{r=1}^{d}\frac{1}{x-\alpha_r},$$

implying

$$\frac{\partial}{\partial x}\big(\ln(f(x))\big) = \frac{\partial}{\partial x}\big(f(x)\big)\cdot\frac{1}{f(x)} = \sum_{r=1}^{d}\frac{1}{x-\alpha_r}.$$

Thus, for $1\leqslant i<j\leqslant k$, by the mean value theorem for derivatives there is $b_i<\beta<b_j$ such that

$$\frac{\ln(f(b_j))-\ln(f(b_i))}{\ln(b_j)-\ln(b_i)} = \sum_{r=1}^{d}\frac{\beta}{\beta-\alpha_r}.$$

Since by assumption $b_i>(1+2d)\cdot\mu\geqslant(1+2d)\cdot|\alpha_r|$, we have

$$\Big|\frac{\beta}{\beta-\alpha_r}-1\Big| = \Big|\frac{\alpha_r}{\beta-\alpha_r}\Big| \leqslant \frac{|\alpha_r|}{\beta-|\alpha_r|} < \frac{|\alpha_r|}{(1+2d)\cdot|\alpha_r|-|\alpha_r|} \leqslant \frac{1}{2d}$$

for all $1\leqslant r\leqslant d$. All differences $\beta-\alpha_r\in\mathbb{C}$ having positive real parts, we get

$$d < \frac{\ln(f(b_j))-\ln(f(b_i))}{\ln(b_j)-\ln(b_i)} < d+\frac{1}{2}.$$

Moreover, by assumption we have $0<\ln(b_j)-\ln(b_i)<\delta\leqslant|\ln(a_j)-\ln(a_i)|$, hence

$$\Big|\frac{\ln(a_j)-\ln(a_i)}{\ln(b_j)-\ln(b_i)}\Big| > 1.$$

Now, letting $\lfloor x\rceil := \lfloor x+\frac{1}{2}\rfloor\in\mathbb{Z}$ denote the integer nearest to $x\in\mathbb{R}$, we set

$$d_{ij} := \Big\lfloor\frac{\ln(a_jf(b_j))-\ln(a_if(b_i))}{\ln(b_j)-\ln(b_i)}\Big\rceil = \Big\lfloor\frac{\ln(f(b_j))-\ln(f(b_i))}{\ln(b_j)-\ln(b_i)}+\frac{\ln(a_j)-\ln(a_i)}{\ln(b_j)-\ln(b_i)}\Big\rceil$$

for all $1 \leqslant i, j \leqslant k$ such that $i \neq j$; note that $d_{ij} = d_{ji}$. Hence from the above estimates we infer that $d_{ij} = d$ if and only if $a_i = a_j$. In particular, all these numbers d_{ij} coincide if and only if $a_1 = \cdots = a_k$, hence in this case immediately determining d.

Combinatorial translation. Thus our task can now be rephrased in combinatorial terms as follows: For $c \in \mathbb{Z}$ let Γ_{d+c} be the undirected graph on the vertex set $\{1,\ldots,k\}$, whose edges are the 2-subsets $\{i,j\} \subseteq \{1,\ldots,k\}$ such that $d_{ij} = d+c$.

Then by the above discussion the connected components of Γ_d are complete graphs, whose vertex sets coincide with the sets of $j \in \{1,\ldots,k\}$ such that the associated scalars a_j assume one and the same value. On the other hand, if Γ_{d+c}, for some $c \neq 0$, has a complete connected component with $r \geqslant 2$ vertices $b_{j_1} < \cdots < b_{j_r}$, then for all $i, j \in \{j_1,\ldots,j_r\}$ such that $i < j$ we have

$$c - 1 < \left| \frac{\ln(a_j) - \ln(a_i)}{\ln(b_j) - \ln(b_i)} \right| < c + \frac{1}{2}.$$

Thus we infer that the sequence $a_{j_1},\ldots,a_{j_r}$ is strictly increasing if $c > 0$, and strictly decreasing if $c < 0$. In particular this implies that $r \leqslant |\mathscr{R}|$. In other words, as soon as we find a complete connected component of a graph Γ_{d+c} having more than $|\mathscr{R}|$ elements, then we may conclude that $c = 0$, and we have determined d. Moreover, if $k > |\mathscr{R}|^2$ than this case actually happens.

Our algorithm to determine the degree d of f, and af for some $a > 0$, is now straightforward: Again our strategy is to increase k step by step, and to choose places $2 \leqslant b_1 < b_2 < \cdots < b_k$ such that b_1 is growing and $\ln(b_k) - \ln(b_1)$ tends to zero. Having made a choice, we compute the numbers $d_{ij} \in \mathbb{Z}$ for all $1 \leqslant i < j \leqslant k$; note that here we do not see a way to avoid using non-exact floating point arithmetic (to evaluate logarithms), while everywhere else we are computing exactly. For all numbers $d' \in \mathbb{Z}$ thus occurring we then determine the graph $\Gamma_{d'}$. Amongst all the graphs found we choose one, again $\Gamma_{d'}$ say, having a complete connected component of maximal cardinality, with vertex set $\mathscr{J} \subseteq \{1,\ldots,k\}$ say. Then we run polynomial recovery, see Section 4.2, using the places $\{b_j; j \in \mathscr{J}\}$ and the values $\{a_j f(b_j); j \in \mathscr{J}\}$, with degree bound d'.

4.4. An example. Here is an example to illustrate the above process. (It is a modified version of an example which actually occurred in our application.) Assume as places b_j, for $1 \leqslant j \leqslant k = 13$, we have chosen the rational primes between 29 and 79, and evaluating the unknown polynomial f has resulted in the list of values $a_j f(b_j)$ given in Table 3; the scalars a_j are of course not known either.

Then it turns out that the numbers $d' \in \mathbb{Z}$, where $1 \leqslant i < j \leqslant 13$, come from an 34-element subset of $\{-27,\ldots,71\}$. For seven of them the associated graph $\Gamma_{d'}$ has a connected component with at least three vertices, but only for two of them we find a complete connected component amongst them: The graph Γ_7 has a complete connected component consisting of the vertices $\mathscr{B}_0 := \{47,61,79\}$, while the graph Γ_{13} consists of three connected components, which all are complete, having the vertices

$$\mathscr{B}_1 := \{37,43,47,53,67,73\}, \quad \mathscr{B}_2 := \{31,41,61,71\}, \quad \mathscr{B}_3 := \{29,59,79\}.$$

Running polynomial recovery, see Section 4.2, using the places $\mathscr{B}_0$ fails by exceeding the degree bound. But running it using $\mathscr{B}_1$ yields $af = \sum_{i=0}^{13} z_i X^i \in \mathbb{Z}[X]$, where

$$[z_0, \ldots, z_{13}] = [1, 4, 8, 11, 12, 12, 12, 12, 12, 12, 11, 8, 4, 1],$$

while running it using $\mathscr{B}_2$ and $\mathscr{B}_3$ yields $\frac{1}{5} \cdot af \in \mathbb{Q}[X]$ and $\frac{1}{25} \cdot af \in \mathbb{Q}[X]$, respectively. Thus we indeed have $d = \deg(f) = 13$, and assuming that $a = 1$ we have determined the scalars a_j, for $1 \leqslant j \leqslant 13$, as well. Note that the bounds assumed in Section 4.2 are fulfilled; and the roots of f turning out to be complex roots of unity, implying $\mu = 1$, the bounds assumed in Section 4.3 are fulfilled as well.

It should be noted that for the preceding discussion we have chosen k large enough to exhibit the occurrence of the erroneous set $\mathscr{B}_0$, for which we indeed observe that the associated scalars a_j are pairwise distinct. But this also reveals another practical observation, at least for polynomials occurring in the applications in Section 5: The scalars a_j, here coming from the three-element set $\mathscr{R} = \{1, \frac{1}{5}, \frac{1}{25}\}$, typically are not uniformly distributed throughout $\mathscr{R}$, but the scalar $a_j = 1$ occurs much more frequently than the other ones.

As was already mentioned, in practice we instead increase k step by step. Then for the smallest $k \geqslant 3$ such that the graph Γ_{13} has a complete connected component with at least three vertices, that is for $k = 6$, we find the set $\mathscr{B} := \{37, 43, 47\}$ of places, indeed being associated to the case $a_j = 1$. Now polynomial recovery using $\mathscr{B}$ readily returns f; note that the bounds assumed in Section 4.2 are still fulfilled.

Table 3 An example for degree detection

j	b_j	$a_j f(b_j)$	a_j
1	29	471132000262895400	$\frac{1}{25}$
2	31	5556161802048405504	$\frac{1}{5}$
3	37	27137887050323114234 4	1
4	41	203982274364082601464	$\frac{1}{5}$
5	43	1885780898401789278912	1
6	47	5946135224244400779264	1
7	53	28077873950889396256392	1
8	59	4493456499569142283200	$\frac{1}{25}$
9	61	34577756822169042208584	$\frac{1}{5}$
10	67	581970465933078043504704	1
11	71	246522309921169431519744	$\frac{1}{5}$
12	73	1766015503219395154436952	1
13	79	196427398952317706342400	$\frac{1}{25}$

4.5. Catching projectivities. We now have to explain where the conditions imposed in Section 4.3 come from: Typically, for example for the tasks described in Sections

5.1 and 5.2, our aim is to determine a matrix over $\mathbb{Z}[X]$ or $\mathbb{Q}[X]$ by computing various specializations first, that is evaluating at certain places $b_1, \dots, b_k$, performing some linear algebra over $\mathbb{Z}$ or $\mathbb{Q}$, as described in Section 3, for each of the specializations, and then lifting back to polynomials as explained in Section 4.2. But the linear algebra step in between might only be unique up to a scalar in $\mathbb{Q}$, which additionally depends on the particular specialization considered. On the other hand, the matrix we are looking for might also only be unique up to a scalar in $\mathbb{Q}(X)$.

Let us now, again, agree on the following convention: Given $f, g \in \mathbb{Z}[X]$, not both zero, let $\gcd(f,g) \in \mathbb{Z}[X]$ denote the polynomial greatest common divisor of f and g with positive leading coefficient. A vector $0 \neq v \in \mathbb{Q}[X]^m$, where $m \in \mathbb{N}$, is called *primitive*, if actually $v \in \mathbb{Z}[X]^m$, and for the greatest common divisor $\gcd(v)$ of its entries we have $\gcd(v) = 1$. Clearly greatest common divisor computations in $\mathbb{Z}$ and in $\mathbb{Z}[X]$ yield a $\mathbb{Q}(X)$-multiple of v which is primitive. Similarly, a matrix $A \in \mathbb{Q}[X]^{m \times n}$, where $m, n \in \mathbb{N}$, is called *primitive*, if actually $A \in \mathbb{Z}[X]^{m \times n}$, and for the greatest common divisor $\gcd(A)$ of its entries we have $\gcd(A) = 1$.

Specializing primitive vectors. Hence, in the above context the task is to recover a primitive vector $[f_1, \dots, f_m] \in \mathbb{Z}[X]^m$ not from specializations $[f_1(b_j), \dots, f_m(b_j)] \in \mathbb{Z}^m$, for $1 \leqslant j \leqslant k$, but from "rescaled" versions $[a_j f_1(b_j), \dots, a_j f_m(b_j)] \in \mathbb{Q}^m$ instead. This places us in the setting of Section 4.3, but it remains to justify the assumption that the scalars $a_j \in \mathbb{Q}$ involved indeed come from a finite pool:

Proposition 4.6. *Let* $f_1, \dots, f_m \in \mathbb{Z}[X]$*, where* $m \in \mathbb{N}$*, such that* $\gcd(f_1, \dots, f_m) = 1 \in \mathbb{Z}[X]$*. Then there is a finite set* $\mathscr{P} \subseteq \mathbb{N}$ *such that for all* $b \in \mathbb{Z}$ *we have*

$$\gcd(f_1(b), \dots, f_m(b)) \in \mathscr{P}.$$

Proof. Note first that by assumption $f_1, \dots, f_m$ do not have any common zeroes, so that $\gcd(f_1(b), \dots, f_m(b)) \in \mathbb{N}$ is well-defined for any $b \in \mathbb{Z}$. We proceed by induction on $m \in \mathbb{N}$. For $m = 1$ we have $f_1 = \pm 1$, and we may let $\mathscr{P} := \{\pm 1\}$. Hence let $m \geqslant 2$, where we may assume that all the f_i, for $1 \leqslant i \leqslant m$, are non-constant. Letting $g := \gcd(f_1, \dots, f_{m-1}) \in \mathbb{Z}[X]$ we have $\gcd(g, f_m) = 1$ Letting $g_i := f_i / g \in \mathbb{Z}[X]$ for $1 \leqslant i \leqslant m-1$, we have $\gcd(g_1, \dots, g_{m-1}) = 1$, thus by induction let $\mathscr{Q} \subseteq \mathbb{N}$ be a set as asserted associated with $g_1, \dots, g_{m-1}$. Now, given $b \in \mathbb{Z}$, we may write

$$x := \gcd(f_1(b), \dots, f_m(b)) = \gcd(g(b)g_1(b), \dots, g(b)g_{m-1}(b), f_m(b))$$

as $x = yz$, where $y = \gcd(g(b), f_m(b))$, and z divides $\gcd(g_1(b), \dots, g_{m-1}(b), f_m(b))$. Hence z divides $\gcd(g_1(b), \dots, g_{m-1}(b))$, and thus divides an element of $\mathscr{Q}$. Moreover, from $\gcd(g, f_m) = 1$ we infer that the resultant $\rho := \mathrm{res}(g, f_m) \in \mathbb{Z}$ is different from zero, see [5, Corollary 6.20], which by [5, Corollary 6.21] implies that $y = \gcd(g(b), f_m(b))$ divides ρ. Thus the set $\mathscr{P}$ of all positive divisors of the elements of $\rho\mathscr{Q} := \{\rho r \in \mathbb{N}; r \in \mathscr{Q}\}$ is as desired. □

5 Linear algebra over polynomial rings

As was already mentioned, our general strategy to determine matrices over $\mathbb{Z}[X]$ or $\mathbb{Q}[X]$ is to specialize first at integral places, to apply linear algebra techniques as described in Section 3 to the matrices over $\mathbb{Z}$ or $\mathbb{Q}$ thus obtained, and subsequently to recover the polynomial entries in question by the Chinese remainder lifting technique described in Section 4.2, applying degree detection as described in Section 4.3 if necessary. In this section we describe how we can do linear algebra over $\mathbb{Z}[X]$ or $\mathbb{Q}[X]$ using this approach.

Since we are faced with both sparse and dense matrices, we keep two corresponding formats for matrices over polynomial rings. (In our application, representing matrices for W-graph representations, see Definition 2.8, are extremely sparse, while Gram matrices for them, see Remark 2.6, typically are dense; see also Example 9.2). We have conversion and multiplication routines between them, but whenever it comes to linear algebra computations we always use the dense matrix format. From the arithmetical side, we are only using standard matrix multiplication, but no asymptotically faster methods, as are for example indicated in [5, Section 12.1].

5.1. Nullspace. We have developed a solution to the following restricted nullspace problem only (which is sufficient for our application):

Given a matrix $A \in \mathbb{Q}[X]^{m\times n}$, where $m,n \in \mathbb{N}$, such that $\mathrm{rk}_{\mathbb{Q}[X]}(\ker(A)) = 1$, the task is to determine a primitive vector $v \in \mathbb{Z}[X]^m$ such that $\ker(A) = \langle v\rangle_{\mathbb{Q}[X]}$; then the vector v is unique up to sign.

To do so, by going over to a suitable $\mathbb{Q}(X)$-multiple we may assume that $A \in \mathbb{Z}[X]^{m\times n}$ is primitive. Then we specialize the matrix A successively at integral places $b_1,\dots,b_k$, yielding matrices $A(b_j) \in \mathbb{Z}^{m\times n}$. Since the rank condition on A is equivalent to saying that $\det(A') = 0$ for all $(m\times m)$-submatrices A' of A, while there is an $((m-1)\times(m-1))$-submatrix A'' of A such that $\det(A'') \neq 0$, we have $\mathrm{rk}_{\mathbb{Z}}(\ker(A(b))) \geqslant 1$ for any $b \in \mathbb{Z}$, and for all but finitely many such b we indeed have $\mathrm{rk}_{\mathbb{Z}}(\ker(A(b))) = 1$. Thus we may assume that all the chosen specializations $A(b_j)$ also fulfill $\mathrm{rk}_{\mathbb{Z}}(\ker(A(b_j))) = 1$. Note that this provides an implicit check whether the rank condition on A indeed holds.

Hence we are in a position to compute the row kernels $\ker(A(b_j)) = \langle v_j\rangle_{\mathbb{Z}} \leqslant \mathbb{Z}^m$ as described in Section 3.6, where the $v_j \in \mathbb{Z}^m$ are primitive, for all $1 \leqslant j \leqslant k$. Thus the latter are of the form $v_j = \frac{1}{a_j}\cdot v(b_j)$, where $a_j = \gcd(v(b_j)) \in \mathbb{N}$, and $v \in \mathbb{Z}[X]^m$ is the desired primitive solution vector from above. By Proposition 4.6 we conclude that the scalars a_j involved indeed come from a finite pool only depending on v.

Now applying degree detection, see Section 4.3, and polynomial recovery, see Section 4.2, yields candidate vectors $0 \neq \widetilde{v} \in \mathbb{Q}[X]^m$, which by going over to a suitable $\mathbb{Q}$-multiple can be assumed to be primitive. Then the correctness of $\widetilde{v}$ can be independently verified by explicitly computing $\widetilde{v}A$ and checking whether this is zero.

5.2. Inverse. Given a matrix $A \in \mathbb{Q}[X]^{n\times n}$, where $n \in \mathbb{N}$, such that $\det(A) \neq 0$, the task is to find $B \in \mathbb{Z}[X]^{n\times n}$ and $c \in \mathbb{Z}[X]$, such that $A^{-1} = \frac{1}{c}\cdot B \in \mathbb{Q}(X)^{n\times n}$ and the overall greatest common divisor $\gcd(B,c) \in \mathbb{Z}[X]$ of the entries of B and c equals $\gcd(B,c) = 1$; then the pair (B,c) is unique up to sign.

To do so, by going over to a suitable $\mathbb{Q}$-multiple we may assume that $A \in \mathbb{Z}[X]^{n\times n}$. Thus the equation $BA = c\cdot E_n$ implies that $\gcd(B)$ divides c, and hence B is primitive. Then we specialize the matrix A successively at integral places $b_1,\ldots,b_k$, yielding matrices $A(b_j)\in\mathbb{Z}^{n\times n}$. Since for all but finitely many $b\in\mathbb{Z}$ we have $\det(A(b))\neq 0$, we may assume that all the chosen specializations $A(b_j)$ indeed also fulfill $\det(A(b_j))\neq 0$. Note that this provides an implicit check whether the invertibility condition on A indeed holds.

Hence we are in a position to compute the inverses $A(b_j)^{-1}\in\mathbb{Q}^{n\times n}$ as described in Section 3.7, yielding $B_j\in\mathbb{Z}^{n\times n}$ and $c_j\in\mathbb{Z}$, such that B_j is primitive and $A(b_j)^{-1}=\frac{1}{c_j}\cdot B_j$, for all $1\leqslant j\leqslant k$. Thus, if $B\in\mathbb{Z}[X]^{n\times n}$ and $c\in\mathbb{Z}[X]$ are the desired solutions from above, we infer

$$B_j=\frac{1}{a_j}\cdot B(b_j)\quad\text{and}\quad c_j=\frac{1}{a_j}\cdot c(b_j),\quad\text{where}\quad a_j:=\gcd(B(b_j),c(b_j))\in\mathbb{N}.$$

By Proposition 4.6 we conclude that the scalars a_j involved indeed come from a finite pool only depending on B and c.

Now applying degree detection, see Section 4.3, and polynomial recovery, see Section 4.2, yields candidate solutions $\widetilde{B}\in\mathbb{Q}[X]^{n\times n}$ and $\widetilde{c}\in\mathbb{Q}[X]^n$, for which by going over to a suitable $\mathbb{Q}$-multiple we may assume that $\widetilde{c}\in\mathbb{Z}[X]^n$ and $\widetilde{B}\in\mathbb{Z}[X]^{n\times n}$ is primitive. Then the correctness of $(\widetilde{B},\widetilde{c})$ can be independently verified by explicitly computing $A\widetilde{B}$ and checking whether it equals $\widetilde{c}\cdot E_n$.

5.3. The exponent of a matrix. In view of the discussion in Section 3.8, and noting that $\mathbb{Q}[X]$ is a principal ideal domain as well, we pursue the analogy between matrix inverses over $\mathbb{Z}$ and over $\mathbb{Q}[X]$ still a little further. Indeed, given a square matrix $A\in\mathbb{Z}[X]^{n\times n}$ such that $\det(A)\neq 0$ as above, the polynomial $c\in\mathbb{Z}[X]$ in the expression $A^{-1}=\frac{1}{c}\cdot B$, where $B\in\mathbb{Z}[X]^{n\times n}$ is chosen primitive, again has another interpretation:

Let the *exponent* $\exp(A)\in\mathbb{Z}[X]$ of A be a primitive generator of the annihilator of the $\mathbb{Q}[X]$-module $\mathbb{Q}[X]^n/\mathrm{im}(A)$, where $\mathrm{im}(A)\leqslant\mathbb{Q}[X]^n$ is the $\mathbb{Q}[X]$-span of the rows of A; then $\exp(A)$ is unique up to sign. Then, similar to Section 3.8, we conclude that $\exp(A)$ and c are associated in $\mathbb{Q}[X]$, and thus the primitivity of $\exp(A)$ yields

$$c=\gcd(c)\cdot\exp(A)\in\mathbb{Z}[X].$$

In other words, computing the inverse of A as described in Section 5.2 also yields a method to compute the exponent of A as $\exp(A)=\frac{1}{\gcd(c)}\cdot c$. Moreover, c governs modular invertibility of A as follows:

Proposition 5.4. *We keep the notation of Section 5.3. Let $\{0\}\neq\mathfrak{p}\lhd\mathbb{Z}[X]$ be a prime ideal, let $Q_{\mathfrak{p}}:=\mathrm{Quot}(\mathbb{Z}[X]/\mathfrak{p})$ be the field of fractions of the integral domain $\mathbb{Z}[X]/\mathfrak{p}$, and let $A_{\mathfrak{p}}\in(\mathbb{Z}[X]/\mathfrak{p})^{n\times n}$ be the matrix obtained from A by reduction modulo $\mathfrak{p}$. Then $A_{\mathfrak{p}}$ is invertible in $Q_{\mathfrak{p}}^{n\times n}$ if and only if $c\notin\mathfrak{p}$.*

Proof. The prime ideals of $\mathbb{Z}[X]$ being well-understood, we are in precisely one of the following cases: (i) We have $\mathfrak{p}=(p)$, where $p\in\mathbb{Z}$ is a prime; then we have $Q_{\mathfrak{p}}\cong\mathrm{Quot}(\mathbb{F}_p[X])=\mathbb{F}_p(X)$, a rational function field; (ii) we have $\mathfrak{p}=(f)$, where

$f \in \mathbb{Z}[X]$ is non-constant and irreducible, hence in particular is primitive; then we have $Q_{\mathfrak{p}} \cong \mathbb{Q}[X]/(f)$, an algebraic number field; (iii) we have $\mathfrak{p} = (p, f)$, where p and f are as above; then we have $Q_{\mathfrak{p}} = \mathbb{Z}[X]/\mathfrak{p} \cong \mathbb{F}_p[X]/(\overline{f})$, a finite field.

Now $A_{\mathfrak{p}}$ is non-invertible in $Q_{\mathfrak{p}}^{n\times n}$ if and only if $\det(A) \in \mathfrak{p}$, which holds if and only if there is an irreducible divisor of $\det(A)$ being contained in $\mathfrak{p}$. Thus is suffices to determine (i) the primes $p \in \mathbb{Z}$, and (ii) the non-constant irreducible polynomials $f \in \mathbb{Z}[X]$ dividing $\det(A)$ in $\mathbb{Z}[X]$.

(i) From $A^{-1} = \frac{1}{\det(A)} \cdot \mathrm{adj}(A) \in \mathbb{Q}(X)^{n\times n}$, where $\mathrm{adj}(A) \in \mathbb{Z}[X]^{n\times n}$ is the adjoint matrix of A, we infer that c divides $\det(A)$ in $\mathbb{Z}[X]$. Hence any prime $p \in \mathbb{Z}$ dividing $\gcd(c)$ also divides $\det(A)$ in $\mathbb{Z}[X]$. Conversely, if p does not divide $\gcd(c)$, then p-modular reduction yields $\overline{AB} = \overline{cE_n} \neq 0 \in \mathbb{F}_p[X]^{n\times n}$, hence $\det(\overline{A}) \neq 0 \in \mathbb{F}_p[X]$. Hence the primes $p \in \mathbb{Z}$ we are looking for are precisely the prime divisors of $\gcd(c)$.

(ii) This is equivalent to finding the irreducible polynomials in $\mathbb{Q}[X]$ dividing $\det(A)$ in $\mathbb{Q}[X]$. Again similar to Section 3.8 we conclude that the latter are precisely the irreducible polynomials dividing $\exp(A)$. Hence the polynomials $f \in \mathbb{Z}[X]$ we are looking for are precisely the non-constant irreducible divisors of $\frac{1}{\gcd(c)} \cdot c$. □

5.5. Product. Given matrices $A \in \mathbb{Q}[X]^{l\times m}$ and $B \in \mathbb{Q}[X]^{m\times n}$, where $l, m, n \in \mathbb{N}$, the task is to compute their product $AB \in \mathbb{Q}[X]^{l\times n}$.

This is straightforwardly done: Again, by going over to suitable $\mathbb{Q}$-multiples we may assume that $A \in \mathbb{Z}[X]^{l\times m}$ and $B \in \mathbb{Z}[X]^{m\times n}$. Then we specialize the matrices A and B successively at integral places $b_1, \ldots, b_k$, yielding matrices $A(b_j) \in \mathbb{Z}^{l\times m}$ and $B(b_j) \in \mathbb{Z}^{m\times n}$, whose products $A(b_j)B(b_j) \in \mathbb{Z}^{l\times n}$ we compute. Now applying polynomial recovery, see Section 4.2, yields candidate solutions $\widetilde{C} \in \mathbb{Q}[X]^{l\times n}$. (Note that since no "rescaling" takes place here it is not necessary to apply degree detection.)

As for correctness, there are a few necessary conditions which can be used as break conditions in polynomial recovery: All entries of $\widetilde{C}$ must be polynomials with integer coefficients, and the degrees of the entries of the input matrices yield bounds on the degrees of those of $\widetilde{C}$. But these conditions are far from being sufficient, so that, in contrast to the tasks in Sections 5.1 and 5.2, here we do not have a general way of independently verifying correctness. (In our application, as a very efficient break condition we have used the fact that the entries of $\widetilde{C}$ have to be of a particular form, see Section 8.4.)

5.6. An alternative approach. The idea of our approach is, essentially, to reduce computations over $\mathbb{Q}[X]$ to computations over $\mathbb{Z}$, where lifting back to polynomials is done in one step by combining specialization and Chinese remainder lifting. In consequence, we almost entirely use arithmetic in characteristic zero (except the use of a large prime field in the p-adic decomposition algorithm in Section 3.5). But it seems to be worth-while to say a few more words on the following "two-step" approach, which was already mentioned briefly in Sections 1 and 2.9:

Assume our aim is to determine a matrix $0 \neq A \in \mathbb{Q}[X]^{m\times n}$, where $m, n \in \mathbb{N}$. To this end, we choose pairwise distinct places $b_1, \ldots, b_k \in \mathbb{Z}$, for some $k \in \mathbb{N}$ such that $k > d$, where $d \in \mathbb{N}_0$ is the maximum of the degrees of the non-zero entries of A. Thus, if we are able to compute the specializations $A(b_j) \in \mathbb{Q}^{n\times n}$, for $1 \leqslant j \leqslant k$, we may recover the entries of A by polynomial interpolation, as for example is described in [5, Section 10.2]. In turn, to find the specializations $A(b_j)$ we choose pairwise distinct primes $p_1,\dots,p_l \in \mathbb{N}$, for some $l \in \mathbb{N}$, such that the denominators of all the entries of $A(b_j)$ are coprime to p_i, for all $1 \leqslant j \leqslant k$ and $1 \leqslant i \leqslant l$. Then reduction modulo the chosen primes yields matrices $A_{p_i}(b_j) \in \mathbb{F}_{p_i}^{m\times n}$. Hence, if $\prod_{i=1}^{l} p_i$ is large enough, and we are able to compute the modular reductions $A_{p_i}(b_j)$, for $1 \leqslant i \leqslant l$, then rational number recovery, see Section 3.4, reveals the entries of $A(b_j)$. Hence this reduces finding the matrix A to finding the matrices $A_{p_i}(b_j)$ over prime fields, for which we in turn may use techniques of the MeatAxe.

Thus here specialization and Chinese remainder lifting are done in two separate steps, aiming at taking advantage of the efficiency of computations in prime characteristic. But the "two-step" approach has severe disadvantages: The number k of places to specialize at is at least as large as the degree of the polynomials in question, hence many more and larger b_j than in our approach are needed, increasing time and memory requirements, presumably drastically. (In our application this means $k \lesssim 200$.) Moreover, in order to use rational number recovery, the number l of primes used for modular reduction must not be too small, at the expense of possibly loosing the very fast arithmetic over small finite fields, which otherwise is a major advantage of the MeatAxe.

Actually, apart from our own experiences, this kind of approach is pursued in [20], and the figures on timings and memory consumption given there seem to support the above comments. But it should be stressed that the emphasis of [20] is on parallelizing this kind of computations, which we here do not consider at all.

6 Computing with representations

As was already mentioned in Section 1, in our application we will make use of a suitable variant of the "standard basis algorithm", which was originally used in [24] for computations over finite fields. In this section we present the necessary ideas from computational representation theory, which can be formulated in terms of the following general setting:

6.1. Standard bases. Let $\mathscr{A}$ be a K-algebra, where K is a field, being generated by the (ordered) set $A_1,\dots,A_r$, where $r \in \mathbb{N}_0$. Moreover, let $\mathfrak{X}\colon \mathscr{A} \to K^{n\times n}$ be an absolutely irreducible matrix representation of $\mathscr{A}$, where $n \in \mathbb{N}$. Then the task is to find a "canonical" K-basis of the row space K^n with respect to the representation $\mathfrak{X}$, where we consider right actions, as is common in the computational world.

To this end, let $A_0 \in \mathscr{A}$ such that $\dim_K(\ker(\mathfrak{X}(A_0))) = 1$; note that whenever $\mathfrak{X}$ is irreducible such an element A_0 exists if and only if $\mathfrak{X}$ is absolutely irreducible. This leads to the following breadth-first search algorithm; see also [24]: Choose a seed vector $0 \neq u \in \ker(\mathfrak{X}(A_0))$, let $\mathfrak{B} := [u]$ and $\mathfrak{T} := [[0,0]]$, and set $i := 1$. As long as i does not exceed the cardinality of $\mathfrak{B}$, let v be the i-th element of $\mathfrak{B}$. Then for $1 \leqslant j \leqslant r$ let successively $w := v \cdot \mathfrak{X}(A_j)$, and check whether or not $w \in \langle \mathfrak{B} \rangle_K$. If so, then discard w; if not, then append w to $\mathfrak{B}$, and append $[i,j]$ to $\mathfrak{T}$. Having done this for all j, increment i and recurse.

Since the growing set $\mathfrak{B}$ is K-linearly independent throughout, this algorithms terminates after at most n loops. After termination, $\langle\mathfrak{B}\rangle_K$ is a non-zero submodule of the irreducible $\mathscr{A}$-module K^n, and thus $\mathfrak{B}$ indeed is a K-basis. (Of course, we may terminate early, without any further checking, as soon as the cardinality of $\mathfrak{B}$ equals n, since from this point on $\mathfrak{B}$ would not change anymore anyway.) The (ordered) set $\mathfrak{B}$ is called a *standard basis* of K^n with respect to the representation $\mathfrak{X}$, the generators $A_1,\ldots,A_r$, and the distinguished element A_0, and the "bookkeeping list" $\mathfrak{T}$ is called the associated *Schreier tree*.

Strictly speaking, $\mathfrak{B}$ also depends on the chosen seed vector, but it is essentially unique in the following sense: If $0 \neq \widetilde{u} \in \ker(\mathfrak{X}(A_0))$ gives rise to the standard basis $\widetilde{\mathfrak{B}}$ with Schreier tree $\widetilde{\mathfrak{T}}$, then we have $\widetilde{u} = c \cdot u$, for some $0 \neq c \in K$, and thus $\widetilde{\mathfrak{B}} = c \cdot \mathfrak{B}$ and $\widetilde{\mathfrak{T}} = \mathfrak{T}$. Moreover, using the Schreier tree $\mathfrak{T} = [[i_1, j_j], \ldots, [i_n, j_n]]$, we may recover $\mathfrak{B} = [u_1,\ldots,u_n]$, up to a scalar, without any searching as follows: Choose $0 \neq u_1 \in \ker(\mathfrak{X}(A_0))$, and for $2 \leqslant k \leqslant n$ let successively $u_k := u_{i_k} \cdot \mathfrak{X}(A_{j_k})$.

In practice. We are able to run the above standard basis algorithm in the following particular cases: If K is a (small) finite field, then this can of course be done using ideas from the MeatAxe, as is already described in [24].

More important from our point of view is the case $K = \mathbb{Q}$. Then we may assume that $u \in \mathbb{Z}^n$, and if additionally $\mathfrak{X}(A_i) \in \mathbb{Z}^{n\times n}$, for all $1 \leqslant i \leqslant r$, then we have $\mathfrak{B} \subseteq \mathbb{Z}^n$, hence the key step in the above algorithm, to decide whether or not $w \in \langle\mathfrak{B}\rangle_{\mathbb{Q}}$, can be done using the p-adic decomposition algorithm in Section 3.5, where whenever $\mathfrak{B}$ is enlarged we also check whether its p-modular reduction $\overline{\mathfrak{B}} \subseteq \mathbb{F}_p^n$ is $\mathbb{F}_p$-linearly independent; if not, then we return failure in order to choose another prime p. (Note that this is reminiscent of the strategy in Section 3.6.)

6.2. Computing homomorphisms. We return to the general setting in Section 6.1, and let $\mathfrak{X}' \colon \mathscr{A} \to K^{n\times n}$ be a matrix representation of $\mathscr{A}$, which is equivalent to $\mathfrak{X}$. Then a standard basis $\mathfrak{B}' = [v'_1,\ldots,v'_n]$ of K^n with respect to the representation $\mathfrak{X}'$ is found by choosing $0 \neq v'_1 \in \ker(\mathfrak{X}'(A_0))$ and just applying the Schreier tree $\mathfrak{T} = [[i_1, j_j], \ldots, [i_n, j_n]]$ already known from the standard basis computation for $\mathfrak{X}$ by letting successively $v'_k := v'_{i_k} \cdot \mathfrak{X}'(A_{j_k})$, for $2 \leqslant k \leqslant n$; note that by assumption we indeed have $\dim_K(\ker(\mathfrak{X}'(A_0))) = 1$.

Now let $0 \neq C \in K^{n\times n}$ be an $\mathscr{A}$-homomorphism from $\mathfrak{X}$ to $\mathfrak{X}'$, that is we have

$$\mathfrak{X}(A) \cdot C = C \cdot \mathfrak{X}'(A) \quad \text{for all} \quad A \in \mathscr{A};$$

of course, it suffices to require this condition for the generators $A_1,\ldots,A_r$ only. Since $\mathfrak{X}$ is absolutely irreducible, it follows that $C \in \mathrm{GL}_n(K)$ and is unique up to a scalar. Moreover, we have $\ker(\mathfrak{X}(A_0)) \cdot C = \ker(\mathfrak{X}'(A_0))$, and thus going over from the standard bases $\mathfrak{B}$ and $\mathfrak{B}'$ with respect to $\mathfrak{X}$ and $\mathfrak{X}'$, respectively, to the associated invertible matrices B and B' with rows $v_1,\ldots,v_n \in K^n$ and $v'_1,\ldots,v'_n \in K^n$, respectively, we get $B \cdot C = B'$, or equivalently

$$C = B^{-1} \cdot B' \in \mathrm{GL}_n(K).$$

Thus to determine C we have to perform the following steps: find $A_0 \in \mathscr{A}$ such that $\dim_K(\ker(\mathfrak{X}(A_0))) = 1$; compute $\ker(\mathfrak{X}(A_0)) \leqslant K^n$ and $\ker(\mathfrak{X}'(A_0)) \leqslant K^n$; compute a Schreier tree $\mathfrak{T}$ with respect to $\mathfrak{X} \cong \mathfrak{X}'$ and A_0; apply the Schreier tree $\mathfrak{T}$ in order to compute standard bases $\mathfrak{B}$ and $\mathfrak{B}'$ of K^n with respect to $\mathfrak{X}$ and $\mathfrak{X}'$, respectively; going over to matrices, compute the inverse $B^{-1} \in \mathrm{GL}_n(K)$; and compute the product $C = B^{-1} \cdot B' \in \mathrm{GL}_n(K)$.

In practice. If $K = \mathbb{Q}(X)$, the nullspaces required can be found as described in Section 5.1, where we may assume that v_1 and v_1' are primitive. Moreover, computing matrix inverses and matrix products can be done as described in Sections 5.2 and 5.5, respectively; by multiplying with a suitable element of K we may assume that C is primitive as well, then C is unique up to sign. Hence for our application it remains to describe how a distinguished element and a Schreier tree can be found, and we have to give an efficient break condition for the algorithm in Section 5.5.

7 Finding standard bases for W-graph representations

We have now described the necessary infrastructure from linear algebra over integral domains, and some relevant general ideas how to compute with representations, to proceed to the explicit determination of Gram matrices of invariant bilinear forms for balanced representations of Iwahori–Hecke algebras. We recall the setting of Section 2.9, which we keep from now on:

Let (W,S) be a finite Coxeter group, and let $\mathscr{H}_A \subseteq \mathscr{H}_K$ be the associated generic Iwahori–Hecke algebras with equal parameters over the ring $A = \mathbb{Z}[v, v^{-1}]$ and the field $K = \mathbb{Q}(v)$, respectively, being generated by $\{T_s; s \in S\}$. Moreover, let $\mathfrak{X}^\lambda \colon \mathscr{H}_K \to K^{n\times n}$, where $n = d_\lambda$, be a W-graph representation associated with $\lambda \in \Lambda$, and let

$$(\mathfrak{X}^\lambda)' \colon \mathscr{H}_K \to K^{n\times n} \colon T_w \mapsto \mathfrak{X}^\lambda(T_{w^{-1}})^{\mathrm{tr}} \quad \text{for all} \quad w \in W.$$

As far as computer implementations are concerned, it is more convenient and more efficient to work with row vectors instead of column vectors. Therefore, we will now work throughout with right actions rather than left actions as in Section 2. Our aim is to find a primitive Gram matrix $P \in \mathbb{Z}[v]^{n\times n}$ for $\mathfrak{X}^\lambda$, that is, using the language of right actions, a primitive matrix such that

$$\mathfrak{X}^\lambda(T_w) \cdot P = P \cdot (\mathfrak{X}^\lambda)'(T_w) \quad \text{for all} \quad w \in W.$$

Thus the task is to find a non-zero $\mathscr{H}_K$-homomorphism from $\mathfrak{X}^\lambda$ to $(\mathfrak{X}^\lambda)'$. In order to use the approach described in Section 6.2, we proceed as follows, where the basic idea of this strategy has already been indicated in [11, Section 4.3]:

7.1. Finding seed vectors. To find a suitable seed vector $u_1 \in K^n$ for the standard basis algorithm with respect to $\mathfrak{X}^\lambda$, we proceed as follows:

Specializing $v \mapsto 1$ we from $\mathscr{H}_A$ recover the group algebra $\mathbb{Q}[W]$, and $\mathfrak{X}^\lambda$ corresponds to an irreducible representation $\mathfrak{Y}^\lambda : \mathbb{Q}[W] \to \mathbb{Q}^{n\times n}$. In particular, the index and sign representations of $\mathscr{H}_K$, given by $\mathrm{ind}_{\mathscr{H}} : T_s \mapsto v$ and $\mathrm{sgn}_{\mathscr{H}} : T_s \mapsto -v^{-1}$, respectively, for all $s \in S$, correspond to the trivial and sign representations of $\mathbb{Q}[W]$, given by $1_W : s \mapsto 1$ and $\mathrm{sgn}_W : s \mapsto -1$, respectively.

As was observed by Benson and Curtis (see [12, Section 6.3] and the references there), there is a subset $J \subseteq S$ (depending on λ, and in general not being unique), such that the restriction of $\mathfrak{Y}^\lambda$ to the parabolic subgroup $\widetilde{W} := W_J \leqslant W$ associated with J fulfills

$$\dim_{\mathbb{Q}}\left(\mathrm{Hom}_{\mathbb{Q}[\widetilde{W}]}(\mathrm{sgn}_{\widetilde{W}}, \mathfrak{Y}^\lambda)\right) = 1.$$

Note that $J = \emptyset$ and $J = S$ if and only if $\mathfrak{Y}^\lambda$ equals 1_W and sgn_W, respectively. Letting $\widetilde{\mathscr{H}_K} \subseteq \mathscr{H}_K$ be the parabolic subalgebra associated with J, this implies

$$\dim_K\left(\mathrm{Hom}_{\widetilde{\mathscr{H}_K}}(\mathrm{sgn}_{\widetilde{\mathscr{H}}}, \mathfrak{X}^\lambda)\right) = 1.$$

In other words, we equivalently have

$$\dim_K\Big(\bigcap_{s\in J} \ker\left(\mathfrak{X}^\lambda(T_s + v^{-1})\right)\Big) = 1.$$

Now we are going to use the fact that $\mathfrak{X}^\lambda$ is a W-graph representation: Using the I-sets associated with $\mathfrak{X}^\lambda$, see Definition 2.8, we conclude that $\ker(\mathfrak{X}^\lambda(T_s + v^{-1})) = \langle e_i; s \in I_i\rangle_K$ for all $s \in S$, where $e_i \in K^n$ denotes the i-th "unit" vector. This implies

$$\bigcap_{s\in J} \ker\left(\mathfrak{X}^\lambda(T_s + v^{-1})\right) = \langle e_i; J \subseteq I_i\rangle_K.$$

Hence we may let $u_1 := e_i$, where $1 \leqslant i \leqslant n$ is the unique index such that $J \subseteq I_i$.

Note that this conversely also yields a way to find all subsets of S fulfilling the Benson–Curtis condition: We run through all subsets $J \subseteq S$, and just check whether there is precisely one index $1 \leqslant i \leqslant n$ such that $J \subseteq I_i$.

7.2. Finding a distinguished element. The above immediate approach strongly uses the fact that $\mathfrak{X}^\lambda$ is a W-graph representation. Thus, in order to find a suitable seed vector $u_1' \in K^n$ for the standard basis algorithm with respect to $(\mathfrak{X}^\lambda)'$ we specify a distinguished element $T^\lambda \in \mathscr{H}_K$ such that $\dim_K(\ker(\mathfrak{X}^\lambda(T^\lambda))) = 1$. Let

$$T^\lambda := \big(\sum_{s\in J} T_s\big) + v^{-1}\cdot|J| \in \mathscr{H}_A \subseteq \mathscr{H}_K.$$

Hence we have $\bigcap_{s\in J}\ker(\mathfrak{X}^\lambda(T_s + v^{-1})) \leqslant \ker(\mathfrak{X}^\lambda(T^\lambda))$, and it remains to be shown that $\dim_K(\ker(\mathfrak{X}^\lambda(T^\lambda))) = 1$:

Assume to the contrary that $\dim_K(\ker\left(\mathfrak{X}^\lambda(T^\lambda)\right)) \geqslant 2$. Then letting

$$\sigma_J := \frac{1}{|J|}\cdot\sum_{s\in J} s \in \mathbb{Q}[\widetilde{W}],$$

specializing $v \mapsto 1$ shows that $\dim_{\mathbb{Q}}(\ker(\mathfrak{Y}^\lambda(1+\sigma_J))) \geqslant 2$ as well. Since for any vector $u \in \ker(\mathfrak{Y}^\lambda(1+\sigma_J))$ we have $u \cdot \mathfrak{Y}^\lambda(\sigma_J^k) = (-1)^k \cdot u$, for all $k \in \mathbb{N}_0$, Lemma 7.3 proven below implies that $\langle u \rangle_{\mathbb{Q}} \leqslant K^n$ is $\mathbb{Q}[\widetilde{W}]$-invariant and carries the sign representation. Thus we have $\dim_{\mathbb{Q}}(\mathrm{Hom}_{\mathbb{Q}[\widetilde{W}]}(\mathrm{sgn}_{\widetilde{W}}, \mathfrak{Y}^\lambda)) \geqslant 2$, a contradiction.

Lemma 7.3. *For $\varepsilon \in \{0,1\}$ let $W_\varepsilon := \{w \in W; \mathrm{sgn}(w) = (-1)^\varepsilon\}$. Moreover, let*

$$\sigma_S := \frac{1}{|S|} \cdot \sum_{s \in S} s \in \mathbb{Q}[W].$$

Then, with respect to the natural topology on $\mathbb{Q}[W] \cong \mathbb{Q}^{|W|}$, we have

$$\lim_{k\to\infty} \sigma_S^{2k+\varepsilon} = \frac{1}{|W_\varepsilon|} \cdot \sum_{w \in W_\varepsilon} w \in \mathbb{Q}[W].$$

Proof. We consider the Markov chain with (finite) state space $W = W_0 \dot\cup W_1$, and transition matrix $M = \mathrm{reg}_W(\sigma_S) \in \mathbb{Q}^{|W|\times|W|}$, where $\mathrm{reg}_W \colon \mathbb{Q}[W] \to \mathbb{Q}^{|W|\times|W|}$ denotes the regular matrix representation of $\mathbb{Q}[W]$. In other words, the matrix entry $M_{w,w'}$, where $w, w' \in W$, is given as

$$M_{w,w'} := \begin{cases} \frac{1}{|S|}, & \text{if} \quad w' = ws \quad \text{for some} \quad s \in S, \\ 0, & \text{otherwise.} \end{cases}$$

Now, since $\mathrm{sgn}(ws) = -\mathrm{sgn}(w)$ for all $w \in W$ and $s \in S$, we conclude that $M^2 = \mathrm{reg}_W(\sigma_S^2)$ induces Markov chains on both W_0 and W_1. Moreover, since any element of W can be written as a word of length at most $l(w_0)$ in the generators S, we infer that $M^{2l(w_0)}$ has positive entries in both the block submatrices belonging to W_0 and W_1, respectively. Hence the induced Markov chains are both irreducible and aperiodic. They thus converge towards stationary distributions, which since M is doubly-stochastic are both equal to the respective uniform distributions. Thus, in particular, the initial state $\sigma_S^\varepsilon \in \langle W_\varepsilon \rangle_{\mathbb{Q}}$ yields

$$\lim_{k\to\infty} \sigma_S^{2k+\varepsilon} = \sigma_S^\varepsilon \cdot \big(\lim_{k\to\infty} (M^2)^k\big) = \frac{1}{|W_\varepsilon|} \cdot \sum_{w \in W_\varepsilon} w.$$

□

7.4. Finding standard bases. The distinguished element T^λ can now be used to find a primitive vector $u_1' \in \ker((\mathfrak{X}^\lambda)'(T^\lambda))$. Next, having both seed vectors u_1 and u_1' in place, we aim at computing the associated standard bases $\mathfrak{B}$ with respect to $\mathfrak{X}^\lambda$, and $\mathfrak{B}'$ with respect to $(\mathfrak{X}^\lambda)'$, for the A-algebra generated by $\{vT_s; s \in S\}$. But since we do not have a standard basis algorithm available for representations over the field K, we again use suitable specializations:

Given a place $0 \neq b \in \mathbb{Z}$, let $\mathfrak{Y}_b^\lambda \colon \mathscr{H}_{\mathbb{Q}} \to \mathbb{Q}^{n\times n}$ be the representation of $\mathscr{H}_{\mathbb{Q}}$ obtained by specializing $v \mapsto b$, that is, considering $\mathscr{H}_{\mathbb{Q}}$ as the $\mathbb{Q}$-algebra generated by $\{bT_s; s \in S\}$ we have

$$\mathfrak{Y}_b^\lambda : bT_s \mapsto \big(\mathfrak{X}^\lambda(vT_s)\big)(b) := \mathfrak{X}^\lambda(vT_s)|_{v\mapsto b} \in \mathbb{Z}^{n\times n};$$

thus in particular for $b=1$, identifying $\mathscr{H}_\mathbb{Q}$ with $\mathbb{Q}[W]$, we recover $\mathfrak{Y}_1^\lambda = \mathfrak{Y}^\lambda$.

Now we compare a putative run of the standard basis algorithm, as described in Section 6.1, with respect to the seed vector $u_1 \in \mathbb{Z}[v]^n$ and the generators $\{\mathfrak{X}^\lambda(vT_s) \in \mathbb{Z}[v]^{n\times n}; s \in S\}$, with a run with respect to the specialized seed vector $u_1(b) \in \mathbb{Z}^n$ and the generators $\{\mathfrak{Y}_b^\lambda(bT_s) \in \mathbb{Z}^{n\times n}; s \in S\}$. These successively produce standard bases $\mathfrak{B} \subseteq \mathbb{Z}[v]^n$ and $\mathfrak{C} \subseteq \mathbb{Z}^n$, respectively. We show by induction on the cardinality $0 \leqslant m \leqslant n$ of the intermediate sets $\mathfrak{B}$, that for all but finitely many b the set $\mathfrak{C}$ is obtained by specializing $\mathfrak{B}$, and that the Schreier trees found in both runs coincide:

Indeed, the key steps are to decide for some $w := u \cdot \mathfrak{X}^\lambda(vT_s) \in \mathbb{Z}[v]^n$ whether or not $w \in \langle\mathfrak{B}\rangle_K$, and similarly for its specialization $w(b) := u(b)\cdot\mathfrak{Y}_b^\lambda(bT_s) \in \mathbb{Z}^n$ whether or not $w(b) \in \langle\mathfrak{C}\rangle_\mathbb{Q}$. Identifying $\mathfrak{B}$ and $\mathfrak{C}$ with matrices $B \in \mathbb{Z}[v]^{m\times n}$ and $C \in \mathbb{Z}^{m\times n}$, respectively, we have $C = B(b)$. Considering the matrix $B_w \in \mathbb{Z}[v]^{(m+1)\times n}$ obtained by concatenating B and w, we have $w \notin \langle\mathfrak{B}\rangle_K$ if and only if there is an $((m+1)\times(m+1))$-submatrix B' of B_w such that $\det(B'_w) \neq 0$. Similarly, we have $w(b) \notin \langle\mathfrak{C}\rangle_\mathbb{Q}$ if and only if there is an $((m+1)\times(m+1))$-submatrix C' of $C_{w(b)} = B_w(b) \in \mathbb{Z}^{(m+1)\times n}$ such that $\det(C') \neq 0$. Hence, whenever $w(b) \notin \langle\mathfrak{C}\rangle_\mathbb{Q}$ we also have $w \notin \langle\mathfrak{B}\rangle_K$, and conversely for all but finitely many b from $w \notin \langle\mathfrak{B}\rangle_K$ we may conclude that $w(b) \notin \langle\mathfrak{C}\rangle_\mathbb{Q}$. (We have used a similar argument in Section 5.1.)

Thus assuming that $0 \neq b \in \mathbb{Z}$ is suitably chosen, we may just run the standard basis algorithm for the seed vector $u_1(b) = u_1 = e_i \in \mathbb{Z}^n$, the i-th "unit" vector, and the generators $\mathfrak{Y}_b^\lambda(bT_s) \in \mathbb{Z}^{n\times n}$, as described in Section 6.1, yielding a Schreier tree $\mathfrak{T}$. Letting $w_1 := 1 \in W$, and $w_i := w_j \cdot s \in W$, if $[j,s]$ is the i-th entry in $\mathfrak{T}$, for $2 \leqslant i \leqslant n$, we thus obtain reduced expressions of the elements $w_i \in W$, and hence the number of steps needed to find the i-th element of $\mathfrak{C}$ equals the length $l(w_i) \in \mathbb{N}_0$. (In practice, it turns out that choosing either $b=1$ or $b=2$ is sufficient, where actually almost always $b=1$ works.)

Applying the Schreier tree $\mathfrak{T}$ to u_1 and $\{\mathfrak{X}^\lambda(vT_s); s \in S\}$ this yields a standard basis $\mathfrak{B} \subseteq \mathbb{Z}[v]^n$ of K^n. Similarly, applying $\mathfrak{T}$ to $u'_1 \in \mathbb{Z}[v]^n$ and $\{(\mathfrak{X}^\lambda)'(vT_s) \in \mathbb{Z}[v]^{n\times n}; s \in S\}$ we get a standard basis $\mathfrak{B}' \subseteq \mathbb{Z}[v]^n$ of K^n. But note that this does *not* ensure that the A-lattices $\langle\mathfrak{B}\rangle_A$ and $\langle\mathfrak{B}'\rangle_A$ are invariant under the A-algebras generated by $\{\mathfrak{X}^\lambda(vT_s); s \in S\}$ and $\{(\mathfrak{X}^\lambda)'(vT_s); s \in S\}$, respectively. (In practice they are not, typically.)

8 Finding Gram matrices for W-graph representations

We keep the setting of Section 7; in particular $\mathfrak{X}^\lambda$ still is a W-graph representation. Having found standard bases $\mathfrak{B}$ and $\mathfrak{B}'$ for $\mathfrak{X}^\lambda$ and $(\mathfrak{X}^\lambda)'$, respectively, we proceed by writing them as matrices $B \in \mathbb{Z}[v]^{n\times n}$ and $B' \in \mathbb{Z}[v]^{n\times n}$, respectively, where by construction both B and B' are primitive. In order to complete the final task of computing the product $B^{-1}\cdot B' \in \mathbb{Z}[v]^{n\times n}$ efficiently, we need a few preparations.

8.1. Palindromicity. Let $*\colon K \to K$ be the involutory field automorphism given by $*\colon v \mapsto v^{-1}$. Hence A is $*$-invariant, and by entry-wise application we get involutory module automorphisms on K^n and A^n, and algebra automorphisms on $K^{n\times n}$ and $A^{n\times n}$, all of which will also be denoted by $*$.

A polynomial $0 \neq f \in \mathbb{Z}[v]$ is called *(k-)palindromic*, for some $k \in \mathbb{N}_0$, if $v^k \cdot f^* = f \in A$, and f is called *(k-)skew-palindromic* if $v^k \cdot f^* = -f \in A$. In these cases, letting $\delta(f) \in \mathbb{N}_0$ be the maximum power of v dividing f in $\mathbb{Z}[v]$, we have $k = \delta(f) + \deg(f)$. Hence f is palindromic or skew-palindromic if and only if $f \in \mathbb{Z}[v]$ and $f^* \in \mathbb{Z}[v^{-1}]$ are associated in A. Moreover, if f is k-skew-palindromic, then specializing $v \mapsto 1$ we get $f(1) = -f(1)$, implying that $v-1$ divides f in $\mathbb{Z}[v]$; similarly, if f is k-palindromic, then specializing $v \mapsto -1$ we get $(-1)^k \cdot f(-1) = f(-1)$, implying that k is even, or $v+1$ divides f in $\mathbb{Z}[v]$.

Proposition 8.2. *(a) Let $P \in \mathbb{Z}[v]^{n\times n}$ be a primitive Gram matrix for $\mathfrak{X}^\lambda$. Then we have $v^m \cdot P^* = P$, where $m = m_P \in \mathbb{N}$ is even and coincides with the maximum of the degrees of the non-zero entries of P.*

(b) For the primitive seed vector $u_1' \in \mathbb{Z}[v]^n$ we have $v^m \cdot (u_1')^* = u_1'$, where $m = m_{u_1'} \in \mathbb{N}_0$ is even and coincides with the maximum of the degrees of the non-zero entries of u_1'. (Trivially, the analogous statement holds for $u_1 \in \mathbb{Z}[v]^n$ with $m_{u_1} = 0$.)

Proof. Letting $E_n \in A^{n\times n}$ be the identity matrix, by Definition 2.8 for $s \in S$ we have

$$\mathfrak{X}^\lambda(T_s)^* = \mathfrak{X}^\lambda(T_s) - (v - v^{-1}) \cdot E_n = \mathfrak{X}^\lambda\big(T_s - (v - v^{-1})\big).$$

In particular, this yields

$$\mathfrak{X}^\lambda(T_s + v^{-1})^* = \mathfrak{X}^\lambda(T_s)^* + v \cdot E_n = \mathfrak{X}^\lambda\big(T_s - (v - v^{-1})\big) + v \cdot E_n = \mathfrak{X}^\lambda(T_s + v^{-1}).$$

(a) We consider the matrix $P^* \in \mathbb{Z}[v^{-1}]^{n\times n}$: For all $s \in S$ we have

$$\mathfrak{X}^\lambda(T_s) \cdot P^* = \Big(\mathfrak{X}^\lambda\big(T_s - (v - v^{-1})\big) \cdot P\Big)^* = \Big(P \cdot \mathfrak{X}^\lambda\big(T_s - (v - v^{-1})\big)^{\mathrm{tr}}\Big)^*$$

$$= \Big(P \cdot \mathfrak{X}^\lambda(T_s)^{*\mathrm{tr}}\Big)^* = \Big(P \cdot \mathfrak{X}^\lambda(T_s)^{\mathrm{tr}*}\Big)^* = P^* \cdot \mathfrak{X}^\lambda(T_s)^{\mathrm{tr}}.$$

Now $m = m_P \in \mathbb{N}$ as above is minimal such that $v^m P^* \in \mathbb{Z}[v]^{n\times n}$, hence we infer that $v^m P^*$ is a primitive Gram matrix for $\mathfrak{X}^\lambda$ as well, and thus we have $v^m P^* = P$ or $v^m P^* = -P$. Assume the latter case holds, then all non-zero entries of P are m-skew-palindromic, implying that $v-1$ divides $\gcd(P)$, contradicting the primitivity of P. Hence we have $v^m P^* = P$, that is all non-zero entries of P are m-palindromic. Assume that m is odd, then we infer that $v+1$ divides $\gcd(P)$, again contradicting the primitivity of P. Hence m is even.

(b) We consider the vector $(u_1')^* \in \mathbb{Z}[v^{-1}]^n$: We have

$$(u_1')^* \cdot (\mathfrak{X}^\lambda)'(T^\lambda) = \Big(u_1' \cdot (\mathfrak{X}^\lambda)'(T^\lambda)^*\Big)^* = \Big(u_1' \cdot \big(\sum_{s\in J} \mathfrak{X}^\lambda(T_s + v^{-1})\big)^{\mathrm{tr}*}\Big)^*$$

$$= \Big(u'_1 \cdot \big(\sum_{s\in J} \mathfrak{X}^\lambda(T_s + v^{-1})\big)^{\mathrm{tr}}\Big)^* = \Big(u'_1 \cdot (\mathfrak{X}^\lambda)'(T^\lambda)\Big)^* = 0.$$

Now $m = m_{u'_1} \in \mathbb{N}_0$ as above is minimal such that $v^m \cdot (u'_1)^* \in \mathbb{Z}[v]^n$, hence we infer that $v^m \cdot (u'_1)^*$ is primitive. Thus from $\dim_K(\ker((\mathfrak{X}^\lambda)'(T^\lambda))) = 1$ we conclude that $v^m \cdot (u'_1)^* = u'_1$ or $v^m \cdot (u'_1)^* = -u'_1$. Now we argue as above. □

8.3. Properties of the standard bases. We have a closer look at the standard bases $\mathfrak{B}$ and $\mathfrak{B}'$, and the associated matrices B and B', where we assume $\mathfrak{B}$ to be chosen according to Section 7.4. The facts collected are largely due to experimental observation, and will be helpful in the final computational steps in Section 8.4. Still, these properties seem to be stronger than expected from general principles, and it should be worth-while to try and prove the particular observations specified below. (In particular, we have checked the standard bases associated with *all* subsets $J \subseteq S$ fulfilling the Benson–Curtis condition, see Section 7.1, for the types E_6, E_7 and E_8.)

Recall that for all $s \in S$ we have

$$(vT_s)^{-1} = v^{-1} \cdot \big(T_s - (v - v^{-1})\big) = v^{-2} \cdot \big(vT_s - (v^2 - 1)\big),$$

hence by the proof of Proposition 8.2 we get

$$\mathfrak{X}^\lambda(vT_s)^* = v^{-1} \cdot \mathfrak{X}^\lambda\big(T_s - (v - v^{-1})\big) = v^{-2} \cdot \mathfrak{X}^\lambda\big(vT_s - (v^2 - 1)\big) = \mathfrak{X}^\lambda\big((vT_s)^{-1}\big).$$

The elements of $\mathfrak{B}$. For any $u_i \in \mathfrak{B}$, where $2 \leqslant i \leqslant n$, we have $u_i = u_j \cdot \mathfrak{X}^\lambda(vT_s)$, for some $1 \leqslant j < i$ and $s \in S$. This yields

$$v^2 \cdot u_j = v^2 \cdot u_i \cdot \mathfrak{X}^\lambda\big((vT_s)^{-1}\big) = u_i \cdot \mathfrak{X}^\lambda\big(vT_s - (v^2 - 1)\big).$$

We conclude that $\gcd(u_i) \in \mathbb{Z}[v]$ and $\gcd(u_j) \in \mathbb{Z}[v]$ are associated in A. Hence by recursion, since u_1 is primitive, we infer that $\gcd(u_i) = v^{d_i} \in \mathbb{Z}[v]$ for some $d_i \in \mathbb{N}_0$.

Moreover, we have $d_j \leqslant d_i \leqslant d_j + 2$. Since $d_1 = 0 = l(w_1)$, this implies $d_i \leqslant 2l(w_i)$ for all $1 \leqslant i \leqslant n$, where $w_i \in W$ is as in Section 7.4. (Experiments show that all three cases $d_i \in \{d_j, d_j + 1, d_j + 2\}$ actually occur.) But the growth behavior of the d_i seems to be more restricted than given by these bounds: Considering the case $l(w_i) = 1$, we have $w_i = s$ for some $s \in S$ such that the “unit” vector u_1 is not an eigenvector of T_s, hence using the shape of $\mathfrak{X}^\lambda(vT_s)$ we conclude that $d_i = 1 = l(w_i)$.

Now, experimentally, we have made the following

Observation 1. *We have $d_i \leqslant l(w_i) + 1$, for all $1 \leqslant i \leqslant n$.*

(Actually, almost always we have got $d_i \leqslant l(w_i)$, for all $1 \leqslant i \leqslant n$, where often we have even seen equality throughout; the only cases found where actually $d_i = l(w_i) + 1$, for some i, are for type E_8, the representation labeled by 3200_x, and two out of the twelve Benson–Curtis subsets of generators.)

The matrix B. Letting $1 \leqslant j < i \leqslant n$ and $s \in S$ be as above, we get

$$v^2 \cdot u_i^* = v^2 \cdot u_j^* \cdot \mathfrak{X}^\lambda(vT_s)^* = u_j^* \cdot \mathfrak{X}^\lambda\big(vT_s - (v^2 - 1)\big).$$

Since the standard basis algorithm is a breadth-first search, from $u_1^* = u_1$ we conclude that there is lower unitriangular matrix $U \in K^{n\times n}$ and a diagonal matrix $D = \mathrm{diag}[v^{2l(w_1)}, \ldots, v^{2l(w_n)}] \in \mathbb{Z}[v]^{n\times n}$, such that

$$D \cdot B^* = U \cdot B.$$

(Note that if the A-lattice $\langle \mathfrak{B} \rangle_A$ was invariant under the A-algebra generated by $\{\mathfrak{X}^\lambda(vT_s); s \in S\}$, then we even had $U \in A^{n\times n}$.)

In particular, letting $l := \sum_{i=1}^n l(w_i) \in \mathbb{N}_0$, we infer that

$$\det(B) = v^{2l} \cdot \det(B^*),$$

hence $\det(B) \in \mathbb{Z}[v]$ is palindromic. Letting $\exp(B) \in \mathbb{Z}[v]$ denote the exponent of B in the sense of Section 5.3, it follows from Proposition 5.4 that the non-constant irreducible polynomials dividing $\det(B)$ are precisely those dividing $\exp(B)$. Now, experimentally, we have made the following

Observation 2. *Any irreducible divisor of* $\exp(B)$ *in* $\mathbb{Z}[v]$ *is monic and palindromic.*

(Actually, in general the entries of the matrix B are neither palindromic nor skew-palindromic; moreover, quite often $\exp(B)$ is a product of cyclotomic polynomials, but this does not always happen.)

In particular, if $\widehat{u}_k^{\mathrm{tr}} \in \mathbb{Z}[v]^{1\times n}$ denotes the k-th column of B, for $1 \leqslant k \leqslant n$, then $\gcd(\widehat{u}_k) \in \mathbb{Z}[v]$ divides $\det(B)$, hence $\gcd(\widehat{u}_k)$ is palindromic as well. (Actually, contrary to $\gcd(u_k) = v^{d_k}$, in general the $\gcd(\widehat{u}_k)$ are not just powers of v.)

The elements of $\mathfrak{B}'$. The recursion used in the standard basis algorithm only depends on the Schreier tree $\mathfrak{T}$, but is independent of the representation considered. Hence for $u_i' \in \mathfrak{B}'$, where $1 \leqslant i \leqslant n$, and u_1' is primitive, we get $\gcd(u_i') = v^{d_i'} \in \mathbb{Z}[v]$ for some $d_i' \in \mathbb{N}_0$. Moreover, if $1 \leqslant j < i \leqslant n$ and $s \in S$ are as above, we get $d_j' \leqslant d_i' \leqslant d_j' + 2$ and $d_i' \leqslant 2l(w_i)$. Actually, the d_i' seem to be closely related to the d_i from above, inasmuch experimentally we have made the following

Observation 3. *We have* $d_i' = d_i$, *for all* $1 \leqslant i \leqslant n$.

The matrix B'. Again by the fact that the recursion used in the standard basis algorithm only depends on $\mathfrak{T}$, and using $v^m \cdot (u_1')^* = u_1'$, where $m = m_{u_1'} \in \mathbb{N}_0$ is as in Proposition 8.2, we get

$$v^m \cdot D \cdot (B')^* = U \cdot B',$$

for the same matrices U and D. In particular, it follows that $\det(B')$ is palindromic. (In general neither $\det(B')$ and $\det(B)$, nor $\exp(B')$ and $\exp(B)$ are associated in A, so that $\langle \mathfrak{B} \rangle_A$ and $\langle \mathfrak{B}' \rangle_A$ are inequivalent A-sublattices of A^n, which typically are not included in each other.) Again, experimentally we have made the following

Observation 4. *Any irreducible divisor of* $\exp(B')$ *in* $\mathbb{Z}[v]$ *is monic and palindromic.*

In particular, similarly, if $\widehat{u}_k'^{\mathrm{tr}} \in \mathbb{Z}[v]^{1\times n}$ denotes the k-th column of B', for $1 \leqslant k \leqslant n$, then $\gcd(\widehat{u}_k') \in \mathbb{Z}[v]$ is palindromic.

The product $B^{-1}\cdot B'$. In combination the above yields

$$v^m\cdot(B^{-1}\cdot B')^* = v^m\cdot(B^*)^{-1}\cdot(B')^* = (D^{-1}\cdot U\cdot B)^{-1}\cdot(D^{-1}\cdot U\cdot B') = B^{-1}\cdot B'.$$

Hence the non-zero entries of $B^{-1}\cdot B'$ are palindromic.

Letting $0\neq b\in\mathbb{Z}$ and $\widehat{B}\in\mathbb{Z}[v]^{n\times n}$ primitive such that $B^{-1}=\frac{1}{b\cdot\exp(B)}\cdot\widehat{B}$, we get

$$b\cdot\exp(B)\cdot B^{-1}\cdot B' = \widehat{B}\cdot B' = c\cdot P,$$

where $P\in\mathbb{Z}[v]^{n\times n}$ is a primitive Gram matrix, and $0\neq c\in\mathbb{Z}[v]$. In particular, since by Observation 2 the exponent $\exp(B)$ is palindromic, we conclude that the non-zero entries of $\widehat{B}\cdot B'$ are palindromic as well.

Moreover, letting $\widetilde{m}=m_{\exp(B)}\in\mathbb{N}_0$ such that $v^{\widetilde{m}}\cdot\exp(B)^*=\exp(B)$, we get

$$v^{m+\widetilde{m}}\cdot(b\cdot\exp(B)\cdot B^{-1}\cdot B')^* = b\cdot\exp(B)\cdot B^{-1}\cdot B'\in\mathbb{Z}[v]^{n\times n}.$$

Hence from $v^{m_P}\cdot P^*=P$, where $m_P\in\mathbb{N}_0$ is as in Proposition 8.2, we get

$$m_P\leqslant m+\widetilde{m} = m_{u_1'}+m_{\exp(B)},$$

providing an upper bound on the degrees of the non-zero entries of P.

8.4. The final product. We are now prepared to do the last computational steps. To do so, we could quite straightforwardly compute first the inverse B^{-1}, that is essentially $\widehat{B}$, and then the product $\widehat{B}\cdot B'$. But it will substantially add to the efficiency if we keep the degrees of the non-zero entries of the matrices involved as small as possible. Now we have already observed above that the rows of B and B' are far from being primitive, and it turns out in practice that this also holds for their columns. We take advantage of this as follows:

Keeping the notation of Section 8.3, let $R:=\mathrm{diag}[v^{d_1},\ldots,v^{d_n}]\in\mathbb{Z}[v]^{n\times n}$. Then the rows of $R^{-1}\cdot B\in\mathbb{Z}[v]^{n\times n}$ are primitive. As for its columns, letting $\widetilde{u}_k^{\mathrm{tr}}\in\mathbb{Z}[v]^{1\times n}$ denote the k-th column of $R^{-1}\cdot B$, for $1\leqslant k\leqslant n$, let

$$C:=\mathrm{diag}[\gcd(\widetilde{u}_1),\ldots,\gcd(\widetilde{u}_n)]\in\mathbb{Z}[v]^{n\times n}.$$

Since by Observation 2 the polynomial $\gcd(\widehat{u}_k)$ is palindromic, using the particular form of R, we conclude that the $\gcd(\widetilde{u}_k)$ are palindromic as well. We let $0\neq\widehat{c}\in\mathbb{Z}[v]$ and $\widehat{C}\in\mathbb{Z}[v]^{n\times n}$ be primitive such that $C^{-1}=\frac{1}{\widehat{c}}\cdot\widehat{C}$. The latter are of course straightforwardly computed, where both $\widehat{c}$ and the diagonal entries of $\widehat{C}$ are palindromic.

Then we get $\widetilde{B}\in\mathbb{Z}[v]^{n\times n}$ such that $B=R\cdot\widetilde{B}\cdot C$, where now all the rows and all the columns of $\widetilde{B}$ are primitive. We use the algorithm in Section 5.2 to compute $0\neq\widehat{b}\in\mathbb{Z}[v]$ and $\widehat{B}\in\mathbb{Z}[v]^{n\times n}$ primitive such that $\widetilde{B}^{-1}=\frac{1}{\widehat{b}}\cdot\widehat{B}$, Since by Observation 2 the exponent $\exp(B)$ is palindromic, using the particular form of R and C, we conclude that $\widehat{b}$ is palindromic as well. Thus altogether we have

$$B^{-1}=\frac{1}{\widehat{b}\cdot\widehat{c}}\cdot\widehat{C}\cdot\widehat{B}\cdot R^{-1}.$$

Similarly, let $R' := \mathrm{diag}[v^{d'_1}, \ldots, v^{d'_n}] \in \mathbb{Z}[v]^{n \times n}$ and

$$C' := \mathrm{diag}[\gcd(\widetilde{u}'_1), \ldots, \gcd(\widetilde{u}'_n)] \in \mathbb{Z}[v]^{n \times n},$$

where $\widetilde{u}'^{\mathrm{tr}}_k \in \mathbb{Z}[v]^{1 \times n}$ denotes the k-th column of $(R')^{-1} \cdot B'$, for $1 \leqslant k \leqslant n$. As above, using Observation 4 implying the palindromicity of $\gcd(\widetilde{u}'_k)$, we conclude that the diagonal entries of C' are palindromic as well, and thus those of $(C')^{-1}$ are too. Then we get $\widetilde{B}' \in \mathbb{Z}[v]^{n \times n}$ such that $B' = R' \cdot \widetilde{B}' \cdot C'$, where now all the rows and all the columns of $\widetilde{B}$ are primitive.

In combination this yields

$$Q := \widehat{b} \cdot \widehat{c} \cdot B^{-1} \cdot B' = \widehat{C} \cdot \widehat{B} \cdot R^{-1} \cdot R' \cdot \widetilde{B}' \cdot C'.$$

By the above considerations we conclude that the non-zero entries of Q are palindromic, which entails that those of $\widehat{B} \cdot R^{-1} \cdot R' \cdot \widetilde{B}'$ are as well. Now by Observation 3 we have $R' = R$, hence this simplifies to

$$Q = \widehat{C} \cdot (\widehat{B} \cdot \widetilde{B}') \cdot C' \in \mathbb{Z}[v]^{n \times n},$$

where the non-zero entries of $\widehat{B} \cdot \widetilde{B}' \in \mathbb{Z}[v]^{n \times n}$ are palindromic.

In practice. To find Q, finally, we apply the matrix multiplication algorithm in Section 5.5 to compute the product $\widehat{B} \cdot \widetilde{B}'$. As was already mentioned, in order to apply it efficiently we need good break conditions to discard erroneous guesses quickly: Apart from requiring that rational number recovery, see Section 3.4, returns only integral coefficients but not rational ones, it turns out that checking for palindromicity is highly effective in this respect.

Having found a good candidate for $\widehat{B} \cdot \widetilde{B}' \in \mathbb{Z}[v]^{n \times n}$, multiplying with the diagonal matrices $\widehat{C} \in \mathbb{Z}[v]^{n \times n}$ and $C' \in \mathbb{Z}[v]^{n \times n}$ is straightforward. Note that, since the result is expected to be a symmetric matrix, it is sufficient to compute only the lower triangular half of the product. Thus we get a candidate for a primitive Gram matrix P from $Q = \gcd(Q) \cdot P \in \mathbb{Z}[v]^{n \times n}$. (In many cases Q already is primitive, but this does not happen always, in which cases $\gcd(Q)$ typically has a smallish degree.)

As independent verification we of course just explicitly check whether the candidate P fulfills the condition

$$\mathfrak{X}^{\lambda}(vT_s) \cdot P = P \cdot \mathfrak{X}^{\lambda}(vT_s)^{\mathrm{tr}} \in \mathbb{Z}[v]^{n \times n} \quad \text{for all} \quad s \in S.$$

9 Timings

We conclude by providing running times and workspace requirements for our computations in types E_7 and E_8, and by presenting an explicit example for type E_6.

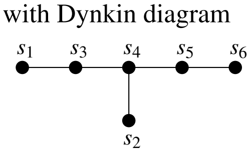
9.1. Timings. In Table 4, we give the running time (on a single processor running at a clock speed of 3.5GHz) and GAP workspace requirements needed to compute primitive Gram matrices for types E_7 and E_8, and the irreducible W-graph representations of $\mathscr{H}_K$ given in [15], [16]. The figures for E_7 should be compared with those given in Section 2.9 for the approach used there. Recalling that in [11, Remark 4.10] degree 2500 was the limit of feasibility, in Table 5 we present the resources now needed for the individual representations of degree at least 2500, where for comparison we repeat the first three columns of the relevant part of Table 2.

Finally, in Table 6 we give some details about the various steps in the computation for the unique representation of largest degree, which is labeled by 7168_w. In the two last columns we indicate the actual size of the object under consideration in the GAP workspace, and the disc space needed to store it (as an uncompressed text file), respectively; the difference is accounted for by the space consumption of the data structure we are using within GAP, where matrices with polynomial entries are kept as lists of lists of (short) lists of (small long) integers. In particular, in the workspace needed to compute the product, next to the matrices $\widehat{B}$ and $\widetilde{B}'$ and (the lower triangular half of) the product $\widehat{B} \cdot \widetilde{B}'$, we also keep various specializations of the right hand factor $\widetilde{B}'$, which have a cumulative size of 7.1GB. Hence to compute a primitive Gram matrix for the representation labeled by 7168_w we need a running time of 1183min $\sim$ 20h and a workspace of size 31.5GB.

Table 4 Time and space consumption

	degree	no. repr.	time	workspace
E_7	all	60	4min	0.2GB
E_8	$\leqslant 1000$	50	30min	0.7GB
	1000—2000	20	137min	2.2GB
	2000—2500	10	329min	4.3GB
	2500—3000	5	350min	5.9GB
	3000—4000	7	874min	11.6GB
	4000—5000	13	3175min	16.3GB
	5000—7000	6	2784min	23.2GB
	$\geqslant 7000$	1	1183min	31.5GB

9.2. An explicit example. We conclude by revisiting the (tiny) example already presented in [11, Example 4.9] (which of course in practice runs in a fraction of a second): Let W be of type E_6 with Dynkin diagram



We consider the irreducible W-graph representation of $\mathscr{H}_K$, see [23], labeled by the representation 10_s of $\mathbb{Q}[W]$, which is the unique one of degree 10, see Table 1. The W-graph in question is depicted in [11, Example 4.9], hence we do not repeat it

Table 5 Time and space consumption for degree $\geqslant 2500$

E_8	m_P	abs.val.	time	workspace
2688_y	24	169180	39min	3.9GB
2800_z	20	38038	61min	3.7GB
$2800'_z$	30	882222	116min	5.9GB
2835_x	24	1344484	52min	3.1GB
$2835'_x$	32	5391418	82min	5.3GB
3150_y	26	6166994	72min	5.8GB
3200_x	24	266284	79min	4.9GB
$3200'_x$	30	587345	104min	6.1GB
3240_z	16	25586	60min	4.0GB
$3240'_z$	48	33653538	326min	11.6GB
3360_z	20	29722	74min	5.1GB
$3360'_z$	32	775084	159min	8.1GB
4096_x	22	531634	156min	8.0GB
$4096'_x$	44	234956568	392min	16.0GB
4096_z	22	531634	143min	8.1GB
$4096'_z$	44	234956568	428min	16.1GB
4200_y	28	58249760	171min	10.1GB
4200_x	24	5413484	171min	9.8GB
$4200'_x$	36	129331224	277min	13.3GB
4200_z	26	728053	183min	10.4GB
$4200'_z$	28	1298612	199min	10.3GB
4480_y	32	85556320920	239min	13.9GB
4536_y	28	3887856	180min	11.7GB
4536_z	24	2728756	217min	11.4GB
$4536'_z$	38	50779421	419min	16.3GB
5600_w	26	372230	331min	16.6GB
5600_z	26	3115126	335min	15.4GB
$5600'_z$	30	3848044	473min	17.5GB
5670_y	30	10762741	351min	21.7GB
6075_x	26	894864	542min	19.5GB
$6075'_x$	34	10488013	752min	23.2GB
7168_w	32	1190470476	1183min	31.5GB

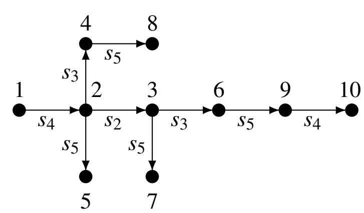
here. But to illustrate the shape, and in particular the sparseness of the representing matrices for the generators $vT_{s_1}, \ldots, vT_{s_6}$ we present a few of them:

Table 6 Time and space consumption for 7168_w

7168_w	time	workspace	space	disc
$\mathfrak{T}$	9min	0.6GB		
u_1'	5min	1.3GB		
$\widehat{B}$	925min	7.6GB	1.7GB	0.3GB
$\widetilde{B}'$	29min	17.5GB	12.6GB	4.7GB
$\widehat{B}\cdot\widetilde{B}'$	207min	31.5GB	5.8GB	2.4GB
P	8min	7.9GB	5.8GB	2.5GB

$$vT_1 \mapsto \begin{bmatrix} v^2 & . & . & v & . & . & . & . & v & . \\ . & v^2 & . & v & . & . & . & . & . & v \\ . & . & -1 & . & . & . & . & . & . & . \\ . & . & . & -1 & . & . & . & . & . & . \\ . & . & . & . & -1 & . & . & . & . & . \\ . & . & v & . & . & v^2 & . & . & . & . \\ . & . & . & . & . & . & v^2 & . & v & v \\ . & . & . & . & v & . & . & v^2 & . & . \\ . & . & . & . & . & . & . & . & -1 & . \\ . & . & . & . & . & . & . & . & . & -1 \end{bmatrix} \qquad vT_6 \mapsto \begin{bmatrix} v^2 & . & . & . & . & . & . & v & v & . \\ . & v^2 & . & . & . & . & . & v & . & v \\ . & . & v^2 & . & . & . & . & . & v & v \\ . & . & . & v^2 & v & . & . & . & . & . \\ . & . & . & . & -1 & . & . & . & . & . \\ . & . & . & . & . & v^2 & v & . & . & . \\ . & . & . & . & . & . & -1 & . & . & . \\ . & . & . & . & . & . & . & -1 & . & . \\ . & . & . & . & . & . & . & . & -1 & . \\ . & . & . & . & . & . & . & . & . & -1 \end{bmatrix}$$

As it turns out, there are 22 possible choices of a distinguished subset $J \subseteq S$. We choose $J := \{s_1, s_2, s_3, s_5, s_6\}$, in accordance with [12, Table C.4]. Then associated primitive seed vectors u_1 and u_1' are as given below, in the first row of the matrices B and $\widetilde{B}'$, respectively. Running the standard basis algorithm on the specialization of the above W-graph representation with respect to $v \mapsto 1$ yields the following Schreier tree $\mathfrak{T}$, which we depict as an oriented graph, whose vertices $1, \ldots, 10$ correspond to the vectors in the (ordered) standard bases, and where an arrow from vertex j to vertex i with label s_k says that $[j, s_k]$ is the i-th entry of $\mathfrak{T}$:



We find the standard basis $\mathfrak{B}$ with associated matrix B as shown below. (It is not always the case that the entries of B are only monomials.) Hence we have $R = \mathrm{diag}[v^{d_1}, \ldots, v^{d_{10}}]$, where $[d_1, \ldots, d_{10}] = [0,1,2,2,2,3,3,3,4,5] = [l(w_1), \ldots, l(w_{10})]$, and C is the identity matrix. Thus we get the matrix $\widetilde{B}$, and from that $\widehat{b} = 1$ and the matrix $\widehat{B}$ as also shown below. Note that the entries of $\widehat{B}$ are not necessarily palindromic or skew-palindromic, and that the maximum degree of the non-zero entries of B, $\widetilde{B}$ and $\widehat{B}$ equals 8, 3 and 5, respectively:

$$B=\begin{bmatrix}
. & . & . & . & . & . & . & . & . & 1\\
. & . & . & . & v & . & . & . & . & v^2\\
. & . & . & . & v^3 & . & . & . & v^2 & .\\
. & . & . & . & v^3 & . & . & v^2 & . & .\\
. & . & . & v^2 & v^3 & . & . & . & . & .\\
. & . & . & . & v^5 & . & v^3 & v^4 & v^4 & v^4\\
. & . & v^3 & v^4 & v^5 & . & . & . & v^4 & v^4\\
. & v^3 & . & v^4 & v^5 & . & . & v^4 & . & v^4\\
. & v^5 & v^5 & v^6 & v^7 & v^4 & v^5 & v^6 & v^6 & v^6\\
v^5 & v^7 & v^7 & v^6 & . & v^6 & v^7 & v^6 & v^6 & v^8
\end{bmatrix}
\qquad
\widetilde{B}=\begin{bmatrix}
. & . & . & . & . & . & . & . & . & 1\\
. & . & . & . & 1 & . & . & . & . & v\\
. & . & . & . & v & . & . & . & 1 & .\\
. & . & . & . & v & . & . & 1 & . & .\\
. & . & . & 1 & v & . & . & . & . & .\\
. & . & . & . & v^2 & . & 1 & v & v & v\\
. & . & 1 & v & v^2 & . & . & . & v & v\\
. & 1 & . & v & v^2 & . & . & v & . & v\\
. & v & v & v^2 & v^3 & 1 & v & v^2 & v^2 & v^2\\
1 & v^2 & v^2 & v & . & v & v^2 & v & v & v^3
\end{bmatrix}$$

$$\widehat{B}=\begin{bmatrix}
2v^5-3v^3 & -2v^4+3v^2 & v^3-v & v^3-v & v^3-v & . & . & . & -v & 1\\
-v^3-v & v^2 & . & -v & -v & . & . & 1 & . & .\\
-v^3-v & v^2 & -v & . & -v & . & 1 & . & . & .\\
v^2 & -v & . & . & 1 & . & . & . & . & .\\
-v & 1 & . & . & . & . & . & . & . & .\\
v^4+2v^2 & -v^3 & v^2 & v^2 & v^2 & -v & -v & -v & 1 & .\\
-v^3-v & v^2 & -v & -v & . & 1 & . & . & . & .\\
v^2 & -v & . & 1 & . & . & . & . & . & .\\
v^2 & -v & 1 & . & . & . & . & . & . & .\\
1 & . & . & . & . & . & . & . & . & .
\end{bmatrix}$$

Similarly, we find the standard basis $\mathfrak{B}'$ with associated matrix B'. As it turns out we indeed have $R'=R$, and C' is the identity matrix. This yields the matrix $\widetilde{B}'$ as shown below. Note that the entries of $\widetilde{B}'$ are not necessarily palindromic or skew-palindromic, and that the maximum degree of the non-zero entries of $\widetilde{B}'$ is 9:

$$\widetilde{B}'=\left[\begin{matrix}
2v^3 & v^5+2v^3+v & v^5+2v^3+v & -v^4-v^2 & v^5+2v^3+v\\
-2v^2 & v^6-v^2 & v^6-v^2 & v^3+v & -v^4-2v^2-1\\
-v^5+v & -2v^5 & -v^5+v & v^6+v^4 & -v^7-2v^5-v^3\\
-v^5+v & -v^5+v & -2v^5 & v^6+v^4 & -v^7-2v^5-v^3\\
-v^5+v & -v^5+v & -v^5+v & -v^2-1 & -v^7-2v^5-v^3\\
2v^4 & 2v^4 & 2v^4 & v^7-v^5 & -v^8+v^4\\
2v^4 & 2v^4 & v^4-1 & -v^5-v^3 & -v^8+v^4\\
2v^4 & v^4-1 & 2v^4 & -v^5-v^3 & -v^8+v^4\\
v^7+v^5-v^3+v & -2v^3 & -2v^3 & -v^6+v^4 & -v^9+v^7-v^5-v^3\\
-v^6-v^4+v^2-1 & -2v^6 & -2v^6 & v^5-v^3 & v^8-v^6+v^4+v^2
\end{matrix}\right.$$

$$\left.\begin{matrix}
-v^4-v^2 & v^5+2v^3+v & -v^4-v^2 & -v^4-v^2 & -v^6-2v^4-2v^2-1\\
-v^5+v^3 & v^6-v^2 & v^3+v & v^3+v & -v^7-v^5\\
v^4-v^2 & -v^5+v & v^6+v^4 & -v^2-1 & v^6+v^4\\
v^4-v^2 & -v^5+v & -v^2-1 & v^6+v^4 & v^6+v^4\\
v^4-v^2 & -2v^5 & v^6+v^4 & v^6+v^4 & v^6+v^4\\
-v^3+v & v^4-1 & -v^5-v^3 & -v^5-v^3 & -v^5-v^3\\
-v^3+v & 2v^4 & v^7-v^5 & -v^5-v^3 & -v^5-v^3\\
-v^3+v & 2v^4 & -v^5-v^3 & v^7-v^5 & -v^5-v^3\\
v^2-1 & -2v^3 & -v^6+v^4 & -v^6+v^4 & v^4+v^2\\
v^7+v^5 & -2v^6 & v^5-v^3 & v^5-v^3 & -v^9+v^7
\end{matrix}\right]$$

From this we get $Q=\widehat{B}\cdot\widetilde{B}'$. As it turns out we already have $\gcd(Q)=1$, thus we may let $P=-Q$ be as shown below. Indeed, independent verification shows that P is a primitive Gram matrix as desired, coinciding with the one already given in [11, Example 4.9]. Note that indeed P is a completely dense matrix, all of whose entries are 6-palindromic, where the maximum degree occurring is 6, and that in accordance with Table 1 the largest coefficient occurring has absolute value 3, and that the specialization $v \mapsto 0$ yields the identity matrix:

$$\left[\begin{array}{ccccc}
v^6+3v^4+3v^2+1 & 2v^4+2v^2 & 2v^4+2v^2 & -v^5-2v^3-v & 2v^4+2v^2 \\
2v^4+2v^2 & v^6+3v^4+3v^2+1 & 2v^4+2v^2 & -v^5-2v^3-v & 2v^4+2v^2 \\
2v^4+2v^2 & 2v^4+2v^2 & v^6+3v^4+3v^2+1 & -v^5-2v^3-v & 2v^4+2v^2 \\
-v^5-2v^3-v & -v^5-2v^3-v & -v^5-2v^3-v & v^6+2v^4+2v^2+1 & -v^5-2v^3-v \\
2v^4+2v^2 & 2v^4+2v^2 & 2v^4+2v^2 & -v^5-2v^3-v & v^6+3v^4+3v^2+1 \\
-v^5-2v^3-v & -v^5-2v^3-v & -v^5-2v^3-v & v^4+v^2 & -2v^3 \\
2v^4+2v^2 & 2v^4+2v^2 & 2v^4+2v^2 & -2v^3 & 2v^4+2v^2 \\
-v^5-2v^3-v & -v^5-2v^3-v & -2v^3 & v^4+v^2 & -v^5-2v^3-v \\
-v^5-2v^3-v & -2v^3 & -v^5-2v^3-v & v^4+v^2 & -v^5-2v^3-v \\
-2v^3 & -v^5-2v^3-v & -v^5-2v^3-v & v^4+v^2 & -v^5-2v^3-v
\end{array}\right.$$

$$\left.\begin{array}{ccccc}
-v^5-2v^3-v & 2v^4+2v^2 & -v^5-2v^3-v & -v^5-2v^3-v & -2v^3 \\
-v^5-2v^3-v & 2v^4+2v^2 & -v^5-2v^3-v & -2v^3 & -v^5-2v^3-v \\
-v^5-2v^3-v & 2v^4+2v^2 & -2v^3 & -v^5-2v^3-v & -v^5-2v^3-v \\
v^4+v^2 & -2v^3 & v^4+v^2 & v^4+v^2 & v^4+v^2 \\
-2v^3 & 2v^4+2v^2 & -v^5-2v^3-v & -v^5-2v^3-v & -v^5-2v^3-v \\
v^6+2v^4+2v^2+1 & -v^5-2v^3-v & v^4+v^2 & v^4+v^2 & v^4+v^2 \\
-v^5-2v^3-v & v^6+3v^4+3v^2+1 & -v^5-2v^3-v & -v^5-2v^3-v & -v^5-2v^3-v \\
v^4+v^2 & -v^5-2v^3-v & v^6+2v^4+2v^2+1 & v^4+v^2 & v^4+v^2 \\
v^4+v^2 & -v^5-2v^3-v & v^4+v^2 & v^6+2v^4+2v^2+1 & v^4+v^2 \\
v^4+v^2 & -v^5-2v^3-v & v^4+v^2 & v^4+v^2 & v^6+2v^4+2v^2+1
\end{array}\right]$$

References

1. Cohen, H.: A course in computational algebraic number theory. Graduate Texts in Mathematics **138**. Springer-Verlag, 1993.
2. Davenport, J., Guy, M., Wang, P.: *P*-adic reconstruction of rational numbers. SIGSAM Bulletin **16** (2) (1982), 2–33.
3. Dixon, J.: Exact solution of linear equations using *p*-adic expansions. Numer. Math. **40** (1982), 137–141.
4. The GAP Group: GAP — Groups, Algorithms, Programming — A System for Computational Discrete Algebra. Version 4.8.5 (2016). `http://www.gap-system.org`.
5. von zur Gathen, J., Gerhard, J.: Modern computer algebra. Third edition. Cambridge University Press, 2013.
6. Garibaldi, S.: E_8, the most exceptional group. Bull. Amer. Math. Soc. **53** (2016), 643–671.
7. Geck, M.: Leading coefficients and cellular bases of Hecke algebras. Proc. Edinburgh Math. Soc. **52** (2009), 653–677.
8. Geck, M., Halls, A,: On the Kazhdan–Lusztig cells in type E_8. Math. Comp. **84** (2015), 3029–3049.
9. Geck, M., Hiß, G., Lübeck, F., Malle, G., Pfeiffer, G.: CHEVIE-A system for computing and processing generic character tables for finite groups of Lie type, Weyl groups and Hecke algebras. Appl. Algebra Engrg. Comm. Comput. **7** (1996), 175–210.
10. Geck, M., Jacon, N.: Representations of Hecke algebras at roots of unity. Algebra and Applications **15**, Springer-Verlag, 2011.
11. Geck, M., Müller, J.: James' conjecture for Hecke algebras of exceptional type I. J. Algebra **321** (2009), 3274–3298.

12. Geck, M., Pfeiffer, G.: Characters of finite Coxeter groups and Iwahori–Hecke algebras. London Math. Soc. Monographs, New Series **21**. Oxford University Press, 2000.
13. The GMP Development Team: GMP — The GNU Multiple Precision Arithmetic Library. Version 6.1.1 (2016). `http://www.gmp-lib.org`.
14. Hardy, G., Wright, E.: An introduction to the theory of numbers. Sixth edition. Oxford University Press, 2008.
15. Howlett, R.B.: W-graphs for the irreducible representations of the Hecke algebra of type E_8. Private communication with J. Michel (December 2003).
16. Howlett, R.B., Yin, Y.: Computational construction of irreducible W-graphs for types E_6 and E_7. J. Algebra **321** (2009), 2055–2067.
17. Kazhdan, D.A., Lusztig, G.: Representations of Coxeter groups and Hecke algebras. Invent. Math. **53** (1979), 165–184.
18. Lusztig, G.: Hecke algebras with unequal parameters. CRM Monographs Ser. **18**. Amer. Math. Soc., 2003. Enlarged and updated version at `arXiv:0208154v2`.
19. Lusztig, G.: Algebraic and geometric methods in representation theory (September 2014). `arxiv:1409.8003`.
20. Maier, P., Livesey, D., Loidl, H.-W., Trinder, P.: High-performance computer algebra: A Hecke algebra case study. *In:* Silva, F., Dutra, I., Santos Costa, V. (eds.): Euro-Par 2014 Parallel Processing: 20th International Conference, Porto (August 2014). Lecture Notes in Computer Science **8632** (2014), 415–426.
21. Michel, J.: The development version of the CHEVIE package of GAP3. J. Algebra **435** (2015), 308–336; see also `http://www.math.rwth-aachen.de/~CHEVIE/`.
22. Monagan, M.: Maximal quotient rational reconstruction: An almost optimal algorithm for rational reconstruction. *In:* Proceedings of ISSAC '04, 243–249. ACM Press, 2004.
23. Naruse, H.: W-graphs for the irreducible representations of the Iwahori–Hecke algebras of type F_4 and E_6. Private communication with M. Geck (January and July, 1998).
24. Parker, R.A.: The computer calculation of modular characters (the Meat-Axe). *In:* Atkinson, M.D. (ed): Computational Group Theory, Durham (1982), 267–274. Academic Press, 1984.
25. Parker, R.A.: An integral meataxe. *In:* The Atlas of Finite Groups: Ten Years On, Birmingham (1995), 215–228. London Mathematical Society Lecture Note Series **249**. Cambridge University Press, 1998.
26. Vogan, D.A.: The character table of E_8. Notices of the Amer. Math. Soc. **9** (2007), 1022–1034; see also `http://atlas.math.umd.edu/AIM_E8/technicaldetails.html`.