SEQUENTIAL STRUCTURES IN CLUSTER ALGEBRAS AND REPRESENTATION THEORY

Florian Gellert

Bielefeld University

A thesis submitted for the degree of
Doctor of Mathematics (Dr. math.)

Bielefeld in April 2017

Abstract

This thesis deals with a range of questions in cluster algebras and the representation theory of quivers. In particular, we provide solutions to the following problems:

1. Does a cluster algebra admit a quantisation and if it does, how unique is it?
2. What is the smallest simply-laced quiver without loops and 2-cycles whose principal extension does not admit a maximal green sequence?
3. Considering the poset of quiver representations of certain orientations of type A_n diagrams induced by inclusion, what is the width of such a poset?

In particular, for a given cluster algebra we construct a basis of those matrices which provide a quantisation. Leading to the smallest simply-laced quiver as proposed above, we prove several combinatorial lemmas for particular quivers with up to four mutable vertices. Furthermore, we introduce a new kind of periodicity in the oriented exchange graph of principally extended cluster algebras. This periodicity we study in more detail for a particular extended Dynkin quiver of type $\tilde{A}_{n-1}$ and show that it yields an infinite sequence of cluster tilting objects inside the preinjective component of the associated cluster category.

Key words: (Quantum) Cluster Algebra, Cluster Category, (Maximal) Green Sequences, Representation Theory of Quivers

Acknowledgements

I would like to thank my supervisor Prof. Henning Krause for his support and advice during the work on my PhD project. Special thanks go to Dr. Philipp Lampe with whom I was fortunate to do research together, leading to many interesting conversations and results.

Besides creating a pleasant environment in itself, thanks go to all members of the BIREP research group for plenty talks, meetings and social gatherings that have accompanied me throughout my time here.

I am also grateful to Bielefeld University and the Bielefeld Graduate School in Theoretical Sciences for financial support, as well as to the many organising parties of conferences and workshops I was lucky to attend.

Bielefeld, 21 April 2017 Florian Gellert

Man braucht fürs Schreiben jede
Menge Zeit zum Verschwenden.

Ian McEwan[a]

[a] ZEITMAGAZIN NR. 27/2015, available at http://www.zeit.de/zeit-magazin/2015/27/ian-mcewan-rettung

Contents

1

Introduction

In the early years of this millennium, Sergey Fomin and Andrei Zelevinsky introduced and studied cluster algebras in a series of four articles [FZ02; FZ03; BFZ05; FZ07], one of which is co-authored by Arkady Berenstein. Their motivation was to gain new insights into Lusztig's canonical basis of quantum groups and total positivity. Since the introduction of cluster algebras, these algebraic structures have become a wide-ranged and intense research topic with many connections to several branches of mathematics such as representation theory, geometry, topology and even to theoretical physics.
Cluster algebras are commutative algebras which are constructed by generators — called cluster variables — which are grouped into overlapping sets of the same cardinality n — called clusters — and relations inside an ambient field. Whenever two clusters share $n-1$ cluster variables, the relation between the two non-identical elements of these two sets is encoded in a skew-symmetric $n \times n$ integer matrix — the so-called exchange matrix — and the operation exchanging these two cluster variables is called mutation.

Generalising this concept, Arkady Berenstein and Andrei Zelevinsky defined quantum cluster algebras in [BZ05]. These are q-deformations which specialise to ordinary cluster algebras in the classical limit $q = 1$. These generalised, non-commutative algebras play an important rôle in cluster theory: on the one hand, quantisations are essential when trying to link cluster algebras to Lusztig's canonical bases, see for example [Lus93; Lec04; Lam11; Lam14; GLS13; HL10]. On the other hand, Goodearl and Yakimov [GY14] compare quantum cluster algebras and their so-called upper bounds, which are intersections of Laurent polynomial rings generated by an initial and once-mutated clusters. It is shown that for a large class of

cases these two algebras coincide, yielding an approximation of cluster algebras by their respective upper bounds in the classical limit.
Unfortunately, not every cluster algebra admits a quantisation. If it exists, then the associated exchange matrix $\tilde{B}$ is necessarily of full rank as shown in [BZ05]. In this thesis we consider the reverse direction and prove in conjunction with the result of [BZ05] that the existence of quantisations of a cluster algebra depends only on the rank of $\tilde{B}$.

Theorem [Cf. Theorem 3.2.4]**.** *A cluster algebra $\mathcal{A}$ associated to an exchange matrix $\tilde{B}$ admits a quantisation if and only if $\tilde{B}$ has full rank.*

Depending on the dimension of $\tilde{B}$, a quantisation of the associated cluster algebra is not necessarily unique. This ambiguity we make more precise by relating all possible quantisations via matrices we explicitly construct from a given $\tilde{B}$ using particular minors. In this fashion, we reobtain the following result from [GSV03], where the subject is considered in the language of certain Poisson structures.

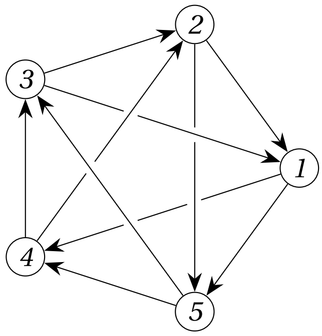
Theorem [Cf. Corollary 3.2.12]**.** *Let $\tilde{B} = \left[\begin{smallmatrix} B \\ C \end{smallmatrix} \right]$ be an $m \times n$ exchange matrix of full rank and r the number of connected components of the mutable part of the associated quiver. Then the solution space of matrices Λ satisfying the definition of quantum cluster algebras to a given skew-symmetriser D of B is a vector space over the rational numbers of dimension $\binom{m-n}{2}$.*

In particular, the set of all quantisations lies in a rational vector space of dimension $r + \binom{m-n}{2}$.

As previously remarked, cluster algebras also have strong connections to theoretical physics. One such intersection is given by quantum dilogarithms, for which Bernhard Keller showed in [Kel11] that certain mutation sequences — called green sequences — in a principally extended cluster algebra lead to identities of such functions. The red or green colouring of mutable vertices in seeds of such cluster algebras is determined by the sign of the columns of the associated C-matrix. Here, the sign-coherence of c-vectors, as proven in full generality in [Gro+14], is an essential ingredient of the well-definedness of this notion.
The definition of green sequences in turn led to a wealth of new questions. In particular, the existence of maximal green sequences — those ending in quivers in which each mutable vertex is coloured red — given a particular quiver is of high interest. In [BDP14] it was proven that all acyclic quivers admit a maximal green sequence and this result has since been extended to all finite-type cluster algebras except those which are of mutation type $\mathbb{X}_7$, cf. the work of Matthew Mills in [Mil16]. The known cases of cyclic quivers which do not admit a maximal green sequence are rather limited. Besides those quivers of type $\mathbb{X}_7$, covered by Ahmet Seven in [Sev14], it was shown by Greg Muller in [Mul16] that cyclic quivers on three vertices for which every side of the 3-cycle is a multi-edge do not admit a maximal green sequence.

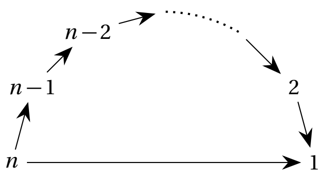
In this work we determine the smallest simply-laced quiver case:

Theorem [Cf. Theorem 4.3.9]**.** *The only simply-laced quiver with up to five vertices which does not admit a maximal green sequence is*



Turning one of the arrows on the outer oriented cycle into a multi-edge, we use the same combinatorial techniques to further provide a new inifinite class of quivers which do not admit maximal green sequences. The combinatorial discussion of these quivers utilises recent results of [Br17] and those branches of the associated oriented exchange graphs which might admit maximal green sequences are explicitly computed.
In order to reduce the a priori infinite oriented exchange graphs to finite cases, we consider recurring subquivers of the mutable parts and relate these observations to existing notions of periodicities in cluster algebras as considered by Allan Fordy and Robert Marsh in [FM11] and Tomoki Nakanishi in [Nak11], among others. We show that these pre-existing notions do not assist in finding maximal green sequences, and we develop the concept of *green permissible periods* instead. Besides proving general results for these particular periods, we further study their appearance in the case of extended Dynkin type $\tilde{A}_{n-1}$ quivers. We show that for each n there exists a particular green permissible period which forms an infinite green sequence in the preinjective component of the associated cluster category, as studied in [BMR08; Bua+06; BMR07; Kel05; Ami09; Ami11] among many others:

Theorem [Cf. Theorem 4.4.24]**.** *The mutation sequence* $(n, n-1, \ldots, 2, 1, n, n-1, \ldots)$ *is a green sequence for the quiver*



and yields an infinite family of cluster tilting objects in the preinjective component of the associated cluster category.

Our result makes use of recent work on the root poset of $\tilde{A}_{n-1}$ quivers in the setting of infinite periodic trees from [ITW14] and we also outline a proof based on the well-developed connection between cluster algebras and triangulated surfaces as first described in [FST08].

The connection between cluster algebras and the representation theory of quivers has been studied in a multitude of papers and the above mentioned cluster categories are only one example where these two realms intersect. In particular, Caldero and Chapoton in [CC06] specify a map between representations of quivers and cluster algebras, an approach which has been extended in [Pla11] and many more. In this connection, (maximal) green sequences are also compared with objects on the representation theoretic side, as can be found in [BY13; KY14; KQ15].
Although the representation theory of quivers has been developed since the mid-20th century, building up on works such as [Gab72; Hum72; Aus74; DR76], it is still an area of intense research. Recently, Claus Ringel has studied in [Rin13] antichains within the root poset of quivers of Dynkin type. There, the partial ordering is given by $x \leq y$ if $y-x$ is a non-negative linear combination of elements in the basis of simple roots. For such posets, the maximal cardinality of any antichain clearly equals the number of vertices of the Dynkin diagram in question and antichains of cardinality $n-1$ are investigated.
Rather than following the above approach by studying the dimension vectors of indecomposable representations, in Chapter 5 we investigate the poset which is given by inclusion of indecomposable representations of Dynkin diagrams. For a range of orientations of type A_n diagrams, we determine Dilworth decompositions for and the width of these posets, cf. Proposition 5.2.1 and Theorems 5.2.2, 5.3.1.

Outline

In Chapter 2 we recall basic notions and definitions from set, graph and representation theory, thus fixing notation for the subsequent discussion.
Chapter 3 concerns cluster algebras and their quantisations. In particular, we introduce cluster algebras in Section 3.1 in a very general form, describe the mutation of cluster variables, highlight the importance of principal coefficients and review classical results in cluster theory. We continue studying quantisations of cluster algebras in Section 3.2, their existence and uniqueness.
Returning to cluster algebras with principal coefficients, the notion of (maximal) green sequences is the focus of Chapter 4. After surveying definitions and recent developments in Sections 4.1 and 4.2, we continue with combintorial observations which enable us to provide the smallest — with respect to the number of mutable vertices — simply-laced quiver without a maximal green sequence. Applying the same techniques we also supply a new infinite family of quivers without maximal green sequences.
Lastly, we consider quiver representations of particular orientations of A_n in Chapter5. The poset structure given by inclusion is analysed and the width of the posets determined.

Declaration of contribution

According to §10(2) of the doctoral regulations the author makes the following statement:

The content of Section 3.2 is based on the co-authored paper [GL14] joint with Philipp Lampe which has been accepted for publication in the *Glasgow Mathematical Journal*. Moreover, the results of Chapter 5 are following the preprint [GL16] with the same co-authorship.

The author hereby declares that he has made substantial contributions to the works mentioned above.

2

Preliminaries

In this chapter we recall basic notions and definitions from set, graph and representation theory. We also fix notation that will be used in the subsequent discussion.

2.1 Set Theory

We recall basic notions from set theory as can be found in the standard literature; we follow [Bou68].

Let $(P, \leq)$ be a poset. Two elements $a, b \in P$ are called *incomparable* if neither $a \leq b$ nor $b \leq a$ holds. The elements are called *comparable* otherwise. A subset $\mathcal{F} \subseteq P$ of pairwise incomparable elements is called an *antichain*. An antichain $\mathcal{F} \subseteq P$ is called *maximal* if there does not exist an element $a \in P$ such that $\mathcal{F} \cup \{a\}$ is an antichain. It is called *maximum* if there does not exist an antichain $\mathcal{F}' \subseteq P$ such that $|\mathcal{F}'| > |\mathcal{F}|$.
Note that every maximum antichain is a maximal antichain, but the converese does not hold in general.
The size of a maximum antichain is called the *width* of the poset. Furthermore, a subset $C \subseteq P$ of pairwise comparable elements is called a *chain*. Note that the elements of a chain can be reordered to form a sequence $(a_1 \leq a_2 \leq \ldots \leq a_k)$ and we will often use this notation to describe a chain. *Maximal* and *maximum chains* are defined in a similar way to maximal and maximum antichains.

Let $n \geq 1$ be an integer. We denote by $\mathcal{P}_n$ the set of all subsets of the finite set $\{1,2,\dots,n\}$ and it is partially ordered set by inclusion.

2.2 Graph Theory

We loosely follow [KV02] for basic notions in Graph Theory.

Definition 2.2.1. An *undirected graph* is a triple (V,E,Ψ) where V and E are (possibly infinite) sets and $\Psi\colon E \longrightarrow \{X \subseteq V : |X| = 2\}$. The elements of V are called *vertices*, the elements of E are called *edges*. A *directed graph* or *digraph* is a triple (V,E,Ψ) with V and E as before and the map of edges is given by $\Psi\colon E \longrightarrow \{(v,w) \in V \times V : v \neq w\}$. Edges e and e' with $\Psi(e) = \Psi(e')$ are said to be *parallel* and graphs without parallel edges are called *simple*.

Given a directed graph (V,E,Ψ) the *associated undirected* or *underlying graph* is constructed by turning pairs of vertices in the image of Ψ into sets. Vice versa, an *orientation* of an undirected graph is given by a choice of making the 2-element sets of the image of Ψ into tuples.

A graph can be visualised in the form of a diagram. For $V = \{1,2,3\}$, one undirected and one directed graph with V as their respective vertex sets are drawn in Figure 2.1. The number of parallel edges in a directed graph between two vertices v and w is indicated by an integer at the edge v—w.

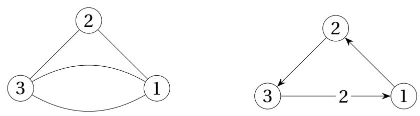


Figure 2.1: Examples of undirected and directed graphs

Note that the undirected graph of Figure 2.1 is indeed the underlying graph of the directed graph, and the latter is an orientation of the former graph.

Assume for the rest of this section that $G = \{V,E,\Psi\}$ is a graph, depending on the context either undirected or directed.

Definition 2.2.2. An *edge progression* W in G from a vertex v_1 to another vertex v_{k+1} is a sequence $(e_1, e_2, \dots, e_{k-1}, e_k)$ such that $k \geq 0$, and $\Psi(e_i) = (v_i, v_{i+1})$ or $\Psi(e_i) = \{v_i, v_{i+1}\}$ for $1 \leq i \leq k$. If also $e_i \neq e_j$ is satisfied for all $1 \leq i < j \leq k$, then W is called a *walk* in G. W is said to be *closed* if $v_1 = v_{k+1}$. A *path* is a walk on pairwise different vertices, i.e. $v_i \neq v_j$ for any two distinct indices $i, j \in \{1,2,\dots,k+1\}$. If a path futher closed, it is called a *cycle* of the graph. If G is directed, an *undirected cycle* is a cycle of the underlying graph. The *length* of an edge progression is the number of edges k and this restricts to walks, paths and cycles accordingly.

Definition 2.2.3. The *adjacency matrix* of an undirected graph G is the matrix $A = (a_{v,w})_{v,w \in V}$ with $a_{v,w}$ denoting the number of edges between v and w. For a directed graph G the *signed adjacency matrix* is defined to be $A = (a_{v,w})_{v,w \in V}$ where $a_{v,w}$ denotes the difference of edges $v \to w$ with $v \leftarrow w$ such that $a_{v,w} = -a_{w,v}$ for all $v, w \in V$.

Definition 2.2.4. For an undirected graph G, two edges e_1 and e_2 are said to be *disjoint* if they have no vertices in common, i.e. if $\Psi(e_i) = \{v_i, w_i\}$ then $\Psi(e_1) \cap \Psi(e_2) = \emptyset$. A *matching* in G is a set of pairwise disjoint edges and a *perfect matching* in G is a matching such that every vertex of V is contained in precisely one edge.

We obtain from Definition 2.2.3 that if G is an undirected graph its adjacency matrix is symmetric. If on the other hand G is directed its adjacency matrix is *skew-symmetric*, i.e. $a_{v,w} = -a_{w,v}$ for all $v, w \in V$. We in fact obtain a bijection between the set of symmetric (resp. skew-symmetric) matrices and undirected graphs (resp. directed graphs without cycles of length 2). The following result applies to the latter kind and will be used in Section 3.2.

Theorem 2.2.5 [Cay49], cf. [Knu96]**.** *Assume A is a skew-symmetric matrix of dimension $n \times n$ with n even. Then there exists a polynomial* $\mathrm{Pf}(A)$ *in the entries of A such that* $\det(A) = \mathrm{Pf}(A)^2$.

In particular, this polynomial is given by

$$\mathrm{Pf}(A) = \sum \mathrm{sgn}(i_1, \ldots, i_{n/2}, j_1, \ldots, j_{n/2}) a_{i_1,j_1} a_{i_2,j_2} \cdots a_{i_{n/2},j_{n/2}} \tag{2.1}$$

where the sum is taken over all perfect matchings $\{\{i_1, j_1\}, \{i_2, j_2\}, \ldots, \{i_{n/2}, j_{n/2}\}\}$ of the underlying graph of the digraph associated to A.

Remark 2.2.6. The polynomial $\mathrm{Pf}(A)$ in Theorem 2.2.5 is called *Pfaffian* after Johann Friedrich Pfaff who first studied such formulas in the context of first-order partial differential equations in the early 19th century.

2.3 Representation Theory

We follow [ASS06] for basic notions and results in representation theory. For simplicity assume that K is an algebraically closed field.

Definition 2.3.1. A quiver $Q = (Q_0, Q_1, s, t)$ is a quadruple where Q_0 and Q_1 are two (possibly infinite) sets together with two maps $s, t : Q_1 \to Q_0$ associating to each *arrow* $\alpha \in Q_1$ its *source* $s(\alpha)$ and *target* $t(\alpha)$.

Remark 2.3.2. We distinguish the notion of directed graphs from Section 2.2 and quivers since loops, i.e. arrows α with $s(\alpha) = t(\alpha)$, are explicitly permitted for the latter objects. Quivers without parallel edges we call *simply-laced* and if there exist arrows α, β in a quiver Q such that $s(\alpha) = s(\beta)$, $t(\alpha) = t(\beta)$, we say that there is a *multiedge* from $s(\alpha)$ to $t(\alpha)$.

Definition 2.3.3. A *path* of a quiver Q is a sequence $(\alpha_1,\dots,\alpha_r)$ of arrows in Q_1 with $t(\alpha_i) = s(\alpha_{i+1})$ for all $1 \le i < r$. To each vertex $a \in Q_0$ we associate a trivial path ε_a of length zero. The *path algebra* KQ of Q is the K-algebra defined by:

- as a K-vector space, the basis of KQ consists of all paths of finite length,
- the product of two paths is given as concatenation on basis elements, i.e. for two paths $p = (e_1,\dots,e_k)$ and $p' = (e'_1,\dots,e'_{k'})$ the product $p\cdot p'$ is $(e_1,\dots,e_k,e'_1,\dots,e'_{k'})$ if $v_{k+1} = v'_1$ for $\Psi(e_k) = (v_k, v_{k+1}), \Psi(e'_1) = (v'_1, v'_2)$ or 0, and
- K-linear extension of the multiplication to KQ.

Definition 2.3.4. For a quiver Q a K-linear representation $\mathbf{V} = (V_a, \varphi_\alpha)$ is given by:

(R_1**)** K-vector space V_a for all $a \in Q_0$,

(R_2**)** K-linear map $\varphi_\alpha\colon V_a \to V_b$ for all arrows $\alpha \in Q_1$ with $s(\alpha) = a$ and $t(\alpha) = b$.

We denote by $\underline{\dim}(\mathbf{V}) = (\dim_k V_a)_{a\in Q_0}$ the *dimension vector of*$\mathbf{V}$. The *support of*$\mathbf{V}$ is defined as the set $\operatorname{supp}(\mathbf{V}) = \{a \in Q_0\colon V_a \neq 0\}$. Furthermore, the summation $\dim_k(\mathbf{V}) = \sum_{a\in Q_0} \dim_k(V_a)$ is called the *dimension of*$\mathbf{V}$.

For the rest of this section fix a quiver $Q = (Q_0, Q_1, s, t)$.

Definition 2.3.5. A *subrepresentation* $\mathbf{U} = (U_a, \psi_\alpha)$ of $\mathbf{V} = (V_a, \varphi_\alpha)$ is a representation of Q such that $U_a \subseteq V_a$ is a k-vector subspace for every vertex $a \in Q_0$ and $\psi_\alpha(x) = \varphi_\alpha(x)$ for every arrow $\alpha\colon a \to b$ in Q_1 and every element $x \in U_a$. In particular, we have $\psi_\alpha(U_a) \subseteq U_b$ for the arrow α.
Given a subrepresentation $\mathbf{U} \subseteq \mathbf{V}$, we can define the *quotient representation* $\mathbf{V}/\mathbf{U}$ by vector spaces $(\mathbf{V}/\mathbf{U})_a = V_a/U_a$ for all vertices $a \in Q_0$ and induced canonical maps $(\varphi/\psi)_\alpha\colon V_a/U_a \to V_b/U_b$ for all arrows $\alpha\colon a \to b$.
A representation $\mathbf{V}$ is called *simple* if it does not admit a non-zero proper subrepresentation $0 \subsetneqq \mathbf{U} \subsetneqq \mathbf{V}$. Suppose that $\mathbf{V} = (V_a, \varphi_\alpha), \mathbf{W} = (W_a, \psi_\alpha)$ are two representations of the quiver Q. A *morphism* $\phi : \mathbf{V} \to \mathbf{W}$ is a collection of k-linear maps $\phi_a\colon V_a \to W_a$ for all vertices $a \in Q_0$ such that $\psi_\alpha \circ \phi_a = \phi_b \circ \varphi_\alpha$ for all arrows $\alpha\colon a \to b$ in Q_1. The morphism with $\phi_a = 0$ for all $a \in Q_0$ is called the *zero morphism*.
A morphism $\phi = (\phi_a)_{a\in Q_0}$ of two representations is called a *monomorphism* if every linear map ϕ_a is injective. Dually, a morphism ϕ is called an *epimorphism* if every linear map ϕ_a is surjective. A morphism $\phi : \mathbf{V} \to \mathbf{W}$ is called an *isomorphism* if it is both a monomorphism and an epimorphism. In this case we say that $\mathbf{V}$ and $\mathbf{W}$ are *isomorphic* and we write $V \cong W$.

Remark 2.3.6. If $\mathbf{U}$ is a subrepresentation of $\mathbf{V}$, then the family of canonical inclusions $U_a \hookrightarrow V_a$ provides a basic example of a monomorphism $\phi : \mathbf{U} \to \mathbf{V}$.

The representation $\mathbf{V} = (V_a, \varphi_\alpha)$ with $V_a = 0$ for all $a \in Q_0$ is called the *zero representation*, where necessarily $\varphi_\alpha = 0$ for all $\alpha \in Q_1$.

Definition 2.3.7. Suppose that $\mathbf{V} = (V_a, \varphi_\alpha), \mathbf{W} = (W_a, \psi_\alpha)$ are two representations of the quiver Q. The *direct sum* $\mathbf{V} \oplus \mathbf{W}$ is the representation with $(\mathbf{V} \oplus \mathbf{W})_a = V_a \oplus W_a$ for all vertices $a \in Q_0$ and

$$(\varphi \oplus \psi)_\alpha = \begin{pmatrix} V_\alpha & 0 \\ 0 & W_\alpha \end{pmatrix} : V_a \oplus W_a \to V_b \oplus W_b$$

for all arrows $\alpha \colon a \to b$.
A representation is called *decomposable* if it is isomorphic to a direct sum $\mathbf{V} \oplus \mathbf{W}$ with $\mathbf{V}, \mathbf{W} \neq 0$. It is called *indecomposable* otherwise. Note that every simple representation is indecomposable but the reverse statement does not hold in general.
A quiver is called *representation finite* if there are only finitely many indecomposable representations up to isomorphism. It is called *representation infinite* otherwise.

Let a *path quiver of length n* be an undirected graph as in Figure 2.2.

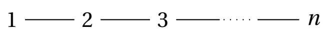


Figure 2.2: A path quiver with n vertices

To unify the description of quivers which are representation finite, let us introduce *star-shaped undirected graphs* as graphs with a central vertex c from which r-many path quivers of varying lengths start. More formally, for integers $r \geq 0$ and $\ell_1, \ldots, \ell_r \geq 1$ let Star$(\ell_1, \ldots, \ell_r)$ be the graph with $n = 1 + \sum_{i=1}^r \ell_i$ many vertices and edges c —— $v_{i,1}$, $v_{i,j}$ —— $v_{i,j+1}$ for $1 \leq i \leq r$ and $1 \leq j \leq \ell_i - 1$. Pictorially such a graph can be seen in Figure 2.3.

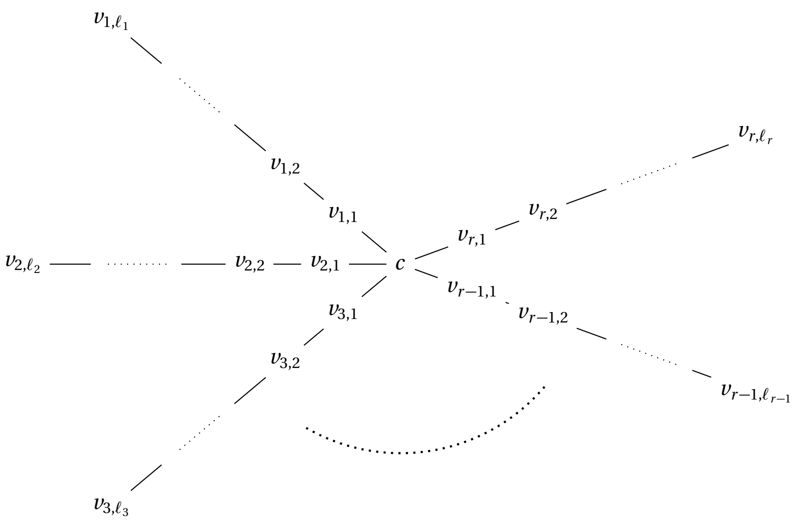


Figure 2.3: Star-shaped undirected graph

A star-shaped undirected graph with n vertices is said to be a *Dynkin diagram of type A_n* if

$r = 1, \ell_1 = n-1$, it is *Dynkin diagram of type* D_n if $r = 3, \ell_1 = \ell_2 = 1, \ell_3 = n-3$ and *Dynkin diagram of type* E_n if $r = 3, \ell_1 = 1, \ell_2 = 2$ and $\ell_3 = n-4$ for $n \in \{6,7,8\}$.

Gabriel [Gab72] classifies representation finite quivers as follows:

Theorem 2.3.8 Gabriel's theorem**.** *A (non-empty) connected quiver with n vertices is representation finite if and only if its underlying undirected graph is a Dynkin diagram of type* A_n*,* D_n *or* E_n*. In this case, the map* $\mathbf{V} \mapsto \underline{\dim}(\mathbf{V})$ *induces a bijection between the isomorphism classes of indecomposable representations and the positive roots in the corresponding root system.*

Especially, representation finiteness does only depend on the underlying diagram but not on the orientation. We say that quivers as in Theorem 2.3.8 are of type A_n, D_n and E_n respectively and call them *Dynkin* if we do not wish to distinguish between these three families.

Example 2.3.9. Let Q be the quiver $1 \xrightarrow{\alpha} 2$ of type A_2. One representation is given by $V_1 = k, V_2 = 0$ and the zero map; denote this representation by $S_1 = (k \to 0)$. Since its only proper subrepresentation is the zero representation, we clearly see that it is simple. Similarly, $S_2 = (0 \to k)$ is simple and thus also an indecomposable representation. Gabriel's theorem asserts the existence of a third indecomposable representation with dimension vector $(1,1)$, namely the representation $P_1 = (k \xrightarrow{\text{id}} k)$. It is an easy observation that the zero morphism is the only morphism from S_1 to P_1, i. e. the left diagram in Figure 2.4 commutes if and only if $\phi_1 = 0 = \phi_2$.

$$
\begin{array}{rcccc}
S_1 = & (k & \xrightarrow{\;0\;} & 0) \\
 & \phi_1 \downarrow & & \downarrow \phi_2 \\
P_1 = & (k & \xrightarrow[\text{id}]{} & k)
\end{array}
\qquad\qquad
\begin{array}{rcccc}
S_2 = & (0 & \xrightarrow{\;0\;} & k) \\
 & \psi_1 \downarrow & & \downarrow \psi_2 \\
P_1 = & (k & \xrightarrow[\text{id}]{} & k)
\end{array}
$$

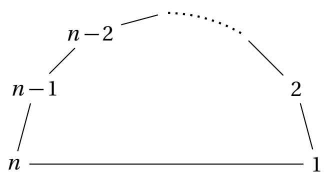
Figure 2.4: Morphisms between indecomposable representations of a quiver of type A_2

On the other hand, the choice $\psi_1 = 0$ and $\psi_2 = \text{id}$ makes the right diagram of Figure 2.4 commutative, hence we obtain a nonzero morphism from S_2 to P_1. Since the identity map is injective, the morphism (ψ_1, ψ_2) is a monomorphism of representations. By Gabriel's theorem, a general representation has the form $V = S_1^a \oplus P_1^b \oplus S_2^c$ for some integers $a, b, c \geq 0$, so that $V_1 = k^a \oplus k^b$, $V_2 = k^b \oplus k^c$, together with the map $\varphi_\alpha = \left(\begin{smallmatrix} 0 & \text{id} \\ 0 & 0 \end{smallmatrix}\right)$ in block form.

For quivers without loops and 2-cycles we may also consider the associated adjacency matrix as introduced in Definition 2.2.3. In this context Gabriel's theorem classifies representation finite skew-symmetric matrices.

Remark 2.3.10. (1) A generalisation of skew-symmetric matrices is given by skew-symmetrisable matrices as introduced in the subsequent chapter. These correspond to so-called *valued quivers* who serve as the underlying structures in the representation-theoretic context of *species*. In [DR76] Gabriel's classification is extended to species, a concise overview can be found in [Lem12].

(2) Generalising Gabriel's theorem, Dlab and Ringel in [DR76] also introduce the so-called *extended Dynkin diagrams* or *Euclidean graphs*. Together with the cases of Theorem 2.3.8 they constitute the most fundamental examples in the representation theory and quiver, whose path algebras are so-called *tame algebras*. An introduction to this elaboration can be found in [ASS06].
One of these extended Dynkin cases will be considered later on, without using the represention theoretic properties of hereditary algebras: the extended Dynkin diagram of type $\tilde{A}_{n-1}$ for $n > 1$ is given by



3

Cluster algebras

This chapter concerns cluster algebras and their quantisations, the purpose being twofold: introducing general cluster algebras together with essential properties of these particular structures, as well as studying their q-deformations called *quantum cluster algebras*.

Section 3.1 is devoted to the review of cluster algebras, classical results for these algebras and the importance of those cluster algebras of geometric type and principal coefficients. We continue studying quantisations of cluster algebras, their existence and uniqueness. In particular, we show that the existence of quantisations of a cluster algebra depends only on the rank of the defining extended exchange matrix $\tilde{B}$, see Theorem 3.2.4.
Depending on the dimension of $\tilde{B}$, a quantisation of the associated cluster algebra is not necessarily unique. This ambiguity we make more precise by relating all possible quantisations via matrices we explicitly construct from a given $\tilde{B}$ using particular minors. In this fashion, we reobtain in Corollary 3.2.12 a result from [GSV03], where the subject is considered in the language of certain Poisson structures.

3.1 Definitions and classical results

For the review of basic definitions in cluster theory we follow the variants in [YZ08, Section 4] and [Nak11, Section 2] of the original definitions in [FZ07].

Let $\mathbb{P}$ be an abelian multiplicative group with an additional binary operation $\oplus$ which is commutative, associative and distributive with respect to the multiplication. A group with these properties is called a *semifield*. It can be readily varified, see [FZ02, Section 5], that $\mathbb{P}$ is torsion free and thus its group ring $\mathbb{Z}\mathbb{P}$ is a domain. Thus, we may form the quotient field of $\mathbb{Z}\mathbb{P}$ which we denote by $\mathbb{Q}(\mathbb{P})$.

Definition 3.1.1. Let J be a finite index set and $\{u_j : j \in J\}$ a family of formal variables. The *tropical semifield* $\mathrm{Trop}(u_j : j \in J)$ associated to this family is given as a multiplicative group by the abelian group freely generated by the u_j and the addition $\oplus$ is determined by

$$\prod_{j\in J} u_j^{a_j} \oplus \prod_{j\in J} u_j^{b_j} = \prod_{j\in J} u_j^{\min(a_j,b_j)} \tag{3.1}$$

with $a_j, b_j \in \mathbb{Z}$. In particular, the group ring of a tropical semifield is the ring of Laurent polynomials in the formal variables u_j.

For the rest of this section let us fix a finite index set I of cardinality $n > 0$ and an *ambient field* $\mathcal{F}$ which we take to be isomorphic to the field of rational functions in n independent variables with coefficients in $\mathbb{Q}(\mathbb{P})$.

Definition 3.1.2. An integer matrix B indexed by I is *skew-symmetrisable* if there exists a diagonal matrix $D = \mathrm{diag}(d_i)_{i\in I}$ with non-zero positive integer entries such that DB is skew-symmetric. In this case, D is called *skew-symmetriser* of B.

Definition 3.1.3. A *seed* is a triplet $(B, \mathbf{x}, \mathbf{y})$ such that $B = (b_{i,j})_{i,j\in I}$ is a skew-symmetrisable integer matrix, $\mathbf{x} = (x_i)_{i\in I}$ is a tuple of elements in $\mathcal{F}$ forming a free generating set for $\mathcal{F}$ and $\mathbf{y} = (y_i)_{i\in I}$ is a tuple of elements in $\mathbb{P}$.
Then $\mathbf{x}$ is called a *cluster* whose members are *cluster variables*, $\mathbf{y}$ is referred to as the *coefficient tuple* and B is called the *exchange matrix*.

At its very core, cluster theory comprises the construction of new seeds out of old ones in a particular operation which we discuss now.

Definition 3.1.4. For any $k \in I$ the *mutation* of $(B, \mathbf{x}, \mathbf{y})$ in direction k is given by the triplet $(B', \mathbf{x}', \mathbf{y}') = \mu_k((B, \mathbf{x}, \mathbf{y}))$, where

$$(M_1) \qquad b'_{i,j} = \begin{cases} -b_{i,j} & \text{if } i = k \text{ or } j = k, \\ b_{i,j} + \operatorname{sgn}\big(b_{i,k}\big)\max\big(0, b_{i,k} b_{k,j}\big) & \text{otherwise,} \end{cases}$$

$$(M_2) \qquad x'_i = \begin{cases} \frac{y_k \prod_{j \in I:\, b_{j,k}>0} x_j^{b_{j,k}} + \prod_{j \in I:\, b_{j,k}<0} x_j^{-b_{j,k}}}{(1 \oplus y_k) x_k} & \text{if } i = k, \\ x_i \text{ if} & i \neq k, \end{cases}$$

$$(M_3) \qquad y'_i = \begin{cases} y_k^{-1} & \text{if } i = k, \\ y_i \big(1 \oplus y_k^{-1}\big)^{-b_{k,i}} & \text{if } i \neq k \text{ and } b_{k,i} \geq 0, \\ y_i \big(1 \oplus y_k\big)^{-b_{k,i}} & \text{if } i \neq k \text{ and } b_{k,i} \leq 0. \end{cases}$$

We can directly compute that B' is indeed a skew-symmetrisable matrix with the same skew-symmetriser $D = \operatorname{diag}(d_i)_{i \in I}$ as B: assume that $b_{i,k} b_{k,j} > 0$ and $b_{i,k} > 0$, then

$$\begin{aligned} d_i b'_{i,j} = -d_j b'_{j,i} \Leftrightarrow & d_i b_{i,j} + d_i b_{i,k} b_{k,j} = -d_j b_{j,i} + d_j b_{j,k} b_{k,i} \\ \Leftrightarrow \qquad & -d_k b_{k,i} b_{k,j} = -d_k b_{k,j} b_{k,i}. \end{aligned}$$

The case $b_{i,k} < 0$ follows analogously, hence B' is again an exchange matrix. By (M_2) and (M_3) it is clear that $\mathbf{x}'$ is a free generating set of $\mathcal{F}$ and $\mathbf{y}' \subseteq \mathbb{P}$. This shows that $(B', \mathbf{x}', \mathbf{y}')$ is again a seed and we call this operation *mutation of seeds* and (M_1)–(M_3) *mutation* of the respective parts of a seed.

It can readily be verified that the mutation of a seed is involutive, i.e. $\mu_k \circ \mu_k((B, \mathbf{x}, \mathbf{y})) = (B, \mathbf{x}, \mathbf{y})$, which imposes the following equivalence relation.

Definition 3.1.5. Two seeds $(B, \mathbf{x}, \mathbf{y})$ and $(B', \mathbf{x}', \mathbf{y}')$ are said to be *mutation equivalent* if there exists a finite tuple $\mathbf{k} = (k_1, k_2, \ldots, k_r)$ of elements in I such that $(B', \mathbf{x}', \mathbf{y}') = \mu_{\mathbf{k}}((B, \mathbf{x}, \mathbf{y})) := \mu_{k_r} \circ \cdots \circ \mu_{k_1}((B, \mathbf{x}, \mathbf{y}))$. We write $(B, \mathbf{x}, \mathbf{y}) \sim (B', \mathbf{x}', \mathbf{y}')$.
Denote by $\mathcal{S}((B', \mathbf{x}', \mathbf{y}'))$ the mutation equivalence class of a seed $(B', \mathbf{x}', \mathbf{y}')$.
The *exchange graph* of $(B, \mathbf{x}, \mathbf{y})$ is the unoriented graph with vertices the mutation equivalence classes of this seed and there is an edge between any two such classes if they can be related to one another by a single mutation.

Definition 3.1.6. A *cluster algebra* $\mathcal{A} = \mathcal{A}((B, \mathbf{x}, \mathbf{y}))$ with coefficients in $\mathbb{P}$ associated to an initial seed $(B, \mathbf{x}, \mathbf{y})$ is the $\mathbb{ZP}$-subalgebra of the ambient field $\mathcal{F}$ generated by all cluster variables in the mutation equivalence class $\mathcal{S}((B, \mathbf{x}, \mathbf{y}))$.
Thus, if we denote the set of all cluster variables inside $\mathcal{S}((B, \mathbf{x}, \mathbf{y}))$ by $\mathcal{X}$, the cluster algebra is given by $\mathbb{ZP}[\mathcal{X}]$.
We call a cluster algebra *skew-symmetric* if the B-matrix in its defining seed $(B, \mathbf{x}, \mathbf{y})$ is skew-symmetric, and *skew-symmetrisable* otherwise. Furthermore, we refer to the exchange graph of $(B, \mathbf{x}, \mathbf{y})$ as the *exchange graph of the cluster algebra* $\mathcal{A}$.

One of the earliest and most fundamental results in cluster theory is the classification of cluster algebras with finitely many mutation equivalence classes, which in some way translates Gabriel's Theorem to this context.

Theorem 3.1.7 [FZ03, Theorem 1.4]**.** *Let $\mathcal{A} = \mathcal{A}((B, \mathbf{x}, \mathbf{y}))$ be some cluster algebra. The mutation equivalence class $\mathcal{S}((B, \mathbf{x}, \mathbf{y}))$ is finite if and only if B is the adjacency matrix of a quiver whose underlying undirected graph is a Dynkin diagram.*

Cluster algebras as in Theorem 3.1.7 are said to be of *finite type*. There exists yet another notion of finiteness in cluster theory.

Definition 3.1.8. For a seed $(B, \mathbf{x}, \mathbf{y})$ assume the integer matrix B is skew-symmetric. The cluster algebra $\mathcal{A}$ associated to B for any choice of cluster variables and coefficients is said to be of *finite mutation type* if the set of exchange matrices that are mutation equivalent to B is finite.

An important part is played by the following particular choice of coefficients.

Definition 3.1.9. We say that a cluster algebra has *geometric type* if the coefficient semifield $\mathbb{P}$ is a tropical semifield. A seed $(B, \mathbf{x}, \mathbf{y})$ is said to have *principal coefficients* if $\mathcal{A}((B, \mathbf{x}, \mathbf{y}))$ has geometric type and $\mathbb{P}$ is generated by the coefficient tuple $\mathbf{y}$.

In the situation of cluster algebras of geometric type, any element of the coefficient tuple of a seed $(B', \mathbf{x}', \mathbf{y}')$ which is mutation equivalent to $(B, \mathbf{x}, \mathbf{y})$ can be expressed as Laurent monomials in the initial coefficient tuple $\mathbf{y}$:

$$y'_j = \prod_{i \in I} y_i^{c_{i,j}}$$

where $c_{i,j} \in \mathbb{Z}$ for all $i, j \in I$. Let $C = (c_{i,j})_{i,j \in I}$ be the so-called *C-matrix*, whose columns are referred to as *c-vectors*, associated to the seed $(B', \mathbf{x}', \mathbf{y}')$. Denote by $\tilde{B}' = \left[\begin{smallmatrix} B' \\ C \end{smallmatrix}\right]$ the *extended exchange matrix* and call B' the *principal part* of $\tilde{B}'$ (sometimes also referred to as the *B-matrix*). In particular, the extended exchange matrix of a seed $(B, \mathbf{x}, \mathbf{y})$ with principal coefficients is $\tilde{B} = \left[\begin{smallmatrix} B \\ \mathbb{I}_n \end{smallmatrix}\right]$, where $\mathbb{I}_n$ denotes the $n \times n$ identity matrix.

The C-matrix complies with the mutation of the exchange matrix: the case $\mu_k(y'_k)$ is clear, so assume $j \neq k$ and $b_{k,j} > 0$, for which we have

$$\begin{aligned}\mu_k(y'_j) = y'_j(1 \oplus (y'_k)^{-1})^{-b_{k,j}} &= \prod_{i \in I} y_i^{c_{i,j}} \left(1 \oplus \prod_{\ell \in I} y_\ell^{-c_{\ell,k}}\right) \\ &= \prod_{i \in I} y_i^{c_{i,j}} \left(\prod_{\ell \in I} y_\ell^{\min(0,-c_{\ell,k})}\right) \\ &= \prod_{i \in I} y_i^{c_{i,j} - b_{k,j}\min(0,-c_{i,k})} \\ &= \prod_{i \in I} y_i^{c_{i,j} + \operatorname{sgn}(b_{j,k})\max(0, b_{k,j} c_{i,k})}.\end{aligned}$$

The third case in (M_3) follows similarly. Setting

$$\hat{y}_j = y_j \prod_{i \in I} x_i^{b_{i,j}} \in \mathcal{F}$$

the theorem below states the importance of a seed with principal coefficients in a given cluster algebra.

Theorem 3.1.10 [FZ07, Cor. 6.3]**.** *Let $(B, \mathbf{x}, \mathbf{y})$ be a seed with principal coefficients and $\mathcal{A}$ the associated cluster algebra. Then for an arbitrary cluster variable $z \in \mathcal{A}$ there exists an integer vector $\mathbf{g} = (g_i)_{i \in I}$ and some integer polynomial $F_z((u_i)_{i \in I}) \in \mathbb{Z}[(u_i)_{i \in I}]$ in indeterminants $(u_i)_{i \in I}$ such that z can be uniquely expressed as*

$$z = \prod_{i \in I} x_i^{g_i} F_z\big((\hat{y}_j)_{j \in I}\big).$$

The integer vectors in Theorem 3.1.10 are called *g-vectors* and the matrix consisting of these vectors as its rows is named *G-matrix*.

Even more remarkable than the expression of any cluster variable by the ones of an initial seed with principal coefficients as described in Theorem 3.1.10 is the general result of [FZ07, Cor. 6.3]: the cluster combinatorics of a cluster algebra with an arbitrary coefficient semifield $\mathbb{P}$ can be obtained by extension of scalars from the cluster algebra associated to a seed with principal coefficients. For a complete overview of these proceedings we refer the reader to [YZ08, Proposition 4.5].

For the rest of this section assume that $(B, \mathbf{x}, \mathbf{y})$ is a seed with prinicipal coefficients, $\tilde{B}$ the associated extended exchange matrix and $\mathcal{A}$ the associated cluster algebra. If B is skew-symmetric, by the above discussion we may visualise $\tilde{B}$ as a quiver $\tilde{Q}$ on $2n$ vertices with one n-tuple of vertices corresponding to the cluster variables, called *mutable vertices*, and a second n-tuple corresponding to the coefficients, called *frozen vertices*. Then mutation may only be applied to those vertices associated to cluster variables and by the subsequent theorem no arrows between frozen vertices can be created under mutation, hence justifying their naming.

Theorem 3.1.11 [Gro+14, Corollary 5.5]**.** *For any seed in $\mathcal{A}$ the c-vectors are sign-coherent, i.e. the columns of the C-matrix either consist of non-negative or non-positive integers.*

We colour frozen vertices light blue to make them easily recognisable. As $(B, \mathbf{x}, \mathbf{y})$ is a seed with prinicipal coefficients, we call $\tilde{Q}$ the *principally extended quiver* or *principal extension* of the full subquiver Q, referred to as the *mutable part of* $\tilde{Q}$, given by mutable vertices, i.e. the quiver whose adjacency matrix is B. Figure 3.1 provides an example of a principally extended quiver.

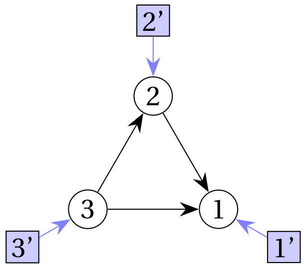


Figure 3.1: Principal extension of a quiver of type $\tilde{A}_2$.

In addition to the fixed initial extended exchange matrix $\tilde{B}$, let $\tilde{B}'$ be mutation equivalent to $\tilde{B}$ and denote by B', C' and G' the associated B-,C- and G-matrix respectively.

For $\varepsilon \in \{+,-\}$ and $k \in I$ let X_k^ε be the identity matrix indexed by I except in its k-th row, in which the entries are given by

$$(X_k^\varepsilon)_{k,j} = \begin{cases} -1 & \text{if } k = j, \\ \max(0, \varepsilon b'_{k,j}) & \text{if } k \neq j. \end{cases} \tag{3.2}$$

Then (M_1) applied to both the principal part as well as the C-matrix of an extended exchange matrix can be rephrased to give an expression of the mutation of B- and C-matrices by matrix multiplication.

Proposition 3.1.12 [BFZ05],[NZ12, Proposition 1.3]**.** *The matrix mutation μ_k for the B- and C-matrices associated to the seed $(B', \mathbf{x}', \mathbf{y}')$ in $\mathcal{A}$ can be reformulated as*

$$\mu_k(B') = D\left(X_k^\varepsilon\right)^T D^{-1} B' X_k^\varepsilon, \tag{3.3}$$

$$\mu_k(C') = C' X_k^\varepsilon, \tag{3.4}$$

where ε is given by the sign of the k-th c-vector, i.e. if all entries in the i-th c-vector are non-negative (resp. non-positive) then $\varepsilon = +$ (resp. $\varepsilon = -$).

Remark 3.1.13. The notation 3.2 is first used in this form in [Br17] and in this paper, the statement regarding the C-matrix in Proposition 3.1.12 has been solely attributed to [NZ12].

Yet in the proof of [BFZ05, Lemma 3.2], the very same relation has already been formulated using slightly different notation.

A direct consequence of (3.4) and (3.2), in conjunction with $C = \mathbb{I}_n$ is that C' is always invertible. This fact can be used to express a multiplicative relation between the matrices C' and G' given in the subsequent theorem.

Theorem 3.1.14 [NZ12, Theorem 1.2]**.** *The C- and G-matrices C' and G' are related by the equation*

$$G' = \left(D(C')^{-1}D^{-1}\right)^T. \tag{3.5}$$

Remark 3.1.15. Proposition 3.1.12 together with Theorem 3.1.14 provides a multiplicative mutation formula for the G-matrix by

$$\mu_k(G') = D^{-T}(C')^{-T}\left(X_k^{\varepsilon}\right)^T D^T, \tag{3.6}$$

see also [NZ12, Proposition 1.3].

Considering both Theorems 3.1.11 and 3.1.14, we conclude the sign-coherence for g-vectors. The sign-coherence was first conjectered for such g-vectors in [FZ07] and it remained an open problem for some time. In the influential paper [DWZ10], the case of skew-symmetric cluster algebras was proven. For skew-symmetrisable types the conjecture prevailed until the publication of [Gro+14].

Let us summarise this introduction to cluster algebras.
A cluster algebra is a certain algebra generated by cluster variables which are linked by an operation called mutation. This map between seeds comprises the exchange of one cluster variable by a distinct unique one in accordance to its membership in a seed.
General cluster algebras are governed by cluster algebras given by seeds with principal coefficients. In this case, if one views both the cluster variables and the coefficients as formal variables, all the datum of the cluster algebra stored in the extended exchange matrix $\tilde{B} = \left[\begin{smallmatrix} B \\ C \end{smallmatrix}\right]$ of any seed.

3.2 Quantisation

One way to generalise cluster algebras is by turning these commutative structures into noncommutative algebras, thus considering their q-deformations called *quantum cluster algebras*. The q-commutativity between elements of a given *quantum seed* is stored in an additional matrix Λ. The compatibility condition between Λ and the extended exchange matrix $\tilde{B}$ is the starting point for our results:
Firstly, we reinterpret what it means for exchange matrices to be of full rank via Pfaffians and perfect matchings. Secondly, we show the only-if part of [BZ05, Proposition 3.3]: assuming the extended exchange matrix $\tilde{B}$ of a given quantum seed is of full rank, there always exists

a quantisation. To show this result we make use of concise linear algebra arguments. It should be noted that Gekhtman–Shapiro–Vainshtein in [GSV03, Thm. 4.5] prove a similar statement in the language of Poisson structures. Thirdly, when a quantisation exists, it is not necessarily unique. This ambiguity we make more precise by relating all such quantisations via matrices constructed from a given $\tilde{B}$ using particular minors.

As discussed in Section 3.1, the cluster structure of cluster algebras of geometric type are encoded in the associated extended exchange matrices and we restrict to these cases in the subsequent discussion. We are following [BZ05] for the introduction of quantum cluster algebras.

Assume that $\tilde{B} = \left[\begin{smallmatrix} B \\ C \end{smallmatrix}\right]$ is an extended exchange matrix of dimension $m \times n$ with $m \geq n$ and skew-symmetrisable principal part B. Note that we slightly modify the notation of Section 3.1 and allow coefficient tuples of arbitrary dimension $m - n$. Further let $I = \{1, 2, \ldots, n\}$ and $F = \{n+1, n+2, \ldots, m\}$ be the index set of cluster variables and coefficients respectively. In this notation, the rows of $\tilde{B}$ are indexed by $\tilde{I} := I \cup F = \{1, 2, \ldots, n, n+1, \ldots, m\}$ and the columns by I. Moreover, denote by $\mathbb{I}_r$ the $r \times r$ identity matrix and by $0_{r\times s}$ the zero matrix of dimension $r \times s$ for positive integers r, s.

Definition 3.2.1. A skew-symmetric $m \times m$ integer matrix $\Lambda = (\lambda_{i,j})$ is called *compatible with* $\tilde{B}$ if there exists a diagonal $n \times n$ matrix $D' = \mathrm{diag}(d'_1, d'_2, \ldots, d'_n)$ with positive integers $d'_1, d'_2, \ldots, d'_n$ such that

$$\tilde{B}^T \Lambda = \begin{bmatrix} D' & 0_{n\times(m-n)} \end{bmatrix} \tag{3.7}$$

as a block matrix. Here, the left block of size $n \times n$ is given by D' and the right by the zero matrix of dimension $n \times (m - n)$.
If (3.7) is satisfied, we call $(\tilde{B}, \Lambda)$ a *compatible pair* and Λ *compatible* with $\tilde{B}$.

To any $m \times n$ matrix $\tilde{B}$ there need not exist a compatible Λ. As a necessary condition Berenstein-Zelevinsky [BZ05, Proposition 3.3] note that if a matrix $\tilde{B}$ belongs to a compatible pair $(\tilde{B}, \Lambda)$, then $\tilde{B}$ itself is of full rank. See Proposition 3.2.3 below for the complete statement.

Assume there exists an integer matrix Λ which is compatible with $\tilde{B}$.Denote by $\{\mathbf{e}_i : 1 \leq 1 \leq m\}$ the standard basis of $\mathbb{Q}^m$. With respect to this standard basis the skew-symmetric matrix Λ defines a skew-symmetric bilinear form $\beta : \mathbb{Q}^m \times \mathbb{Q}^m \to \mathbb{Q}$. This bilinear form is used to define a noncommutative structure in the following way.

Definition 3.2.2. The *based quantum torus* $\mathcal{T}_\Lambda$ associated with Λ is the $\mathbb{Z}[q^{\pm 1}]$-algebra with $\mathbb{Z}[q^{\pm 1}]$-basis $\{X^a : a \in \mathbb{Z}^m\}$ where we define the multiplication of basis elements by

$$X^a X^b = q^{\beta(a,b)} X^{a+b} \tag{3.8}$$

for all elements $a, b \in \mathbb{Z}^m$.

Note that we retain the notation basis elements as given in [BZ05], despite our use of the letter X in (3.2). For this section, X always stands for elements of based quantum tori as described above.

The based quantum torus is an associative algebra with unit $1 = X^0$ and every basis element X^a has an inverse $(X^a)^{-1} = X^{-a}$. It is commutative if and only if Λ is the zero matrix, in which case $\mathcal{T}_\Lambda$ is a Laurent polynomial algebra. In general, it is an Ore domain, see [BZ05, Appendix] for further details. We embed $\mathcal{T}_\Lambda \subseteq \mathcal{F}$ into an ambient skew field $\mathcal{F}$.

Although $\mathcal{T}_\Lambda$ is not commutative in general, applying (3.8) both to $X^a X^b$ and $X^b X^a$ yields that

$$X^a X^b = q^{2\beta(a,b)} X^b X^a$$

holds for all elements $a, b \in \mathbb{Z}^m$. Because of this relation we say that the basis elements are *q-commutative*. Put $X_i = X^{\mathbf{e}_i}$ for all $i \in \tilde{I}$. The definition implies $X_i X_j = q^{\lambda_{i,j}} X_j X_i$ for all $i, j \in \tilde{I}$ and we may write $\mathcal{T}_\Lambda = \mathbb{Z}[q^{\pm 1}][X_1^{\pm 1}, X_2^{\pm 1}, \ldots, X_m^{\pm 1}]$. The basis vectors satisfy the relation

$$X^{\mathbf{a}} = q^{\sum_{i>j} \lambda_{i,j} a_i a_j} X_1^{a_1} X_2^{a_2} \cdot \ldots \cdot X_m^{a_m}$$

for all $\mathbf{a} = (a_1, a_2, \ldots, a_m) \in \mathbb{Z}^m$.

We call a sequence of pairwise q-commutative and algebraically independent elements such as $\mathbf{X} = (X_1, X_2, \ldots, X_m)$ in $\mathcal{F}$ an *extended quantum cluster*, the elements $X_1, X_2, \ldots, X_n$ of an extended quantum cluster *quantum cluster variables*, the elements $X_{n+1}, X_{n+2}, \ldots, X_m$ *frozen variables* (or *coefficients* as in the non-quantised case) and the triple $(\tilde{B}, \mathbf{X}, \Lambda)$ a *quantum seed*.

Let $k \in I$ be a mutable index and denote $\tilde{B} = (\tilde{b}_{i,j})_{i \in \tilde{I}, j \in I}$. Define the *mutation of quantum seeds* $\mu_k : (\tilde{B}, X, \Lambda) \mapsto (\tilde{B}', X', \Lambda')$ as follows:

(QM_1) The matrix $\tilde{B}' = \mu_k(\tilde{B}) = \big(\mu_k(\tilde{B})_{i,j}\big)_{i \in \tilde{I}, j \in I}$ is as in (M_1) extended to indices $1 \le i \le m$, $1 \le j \le n$.

(QM_2) The matrix $\Lambda' = (\lambda'_{i,j})$ is the $m \times m$ matrix with entries $\lambda'_{i,j} = \lambda_{i,j}$ except for

$$\lambda'_{i,k} = -\lambda_{i,k} + \sum_{r \in \tilde{I} \setminus \{k\}} \lambda_{i,r} \max(0, -\tilde{b}_{r,k}) \text{ for all } i \in \{1, \ldots, m\} \setminus \{k\},$$
$$\lambda'_{k,j} = -\lambda_{k,j} - \sum_{r \in \tilde{I} \setminus \{k\}} \lambda_{j,r} \max(0, -\tilde{b}_{r,k}) \text{ for all } j \in \{1, \ldots, m\} \setminus \{k\}.$$

(QM_3) To obtain the quantum cluster $\mathbf{X}'$, we replace the quantum cluster variable X_k with

$$X'_k = X^{-\mathbf{e}_k + \sum_{i=1}^{m} \max(0, \tilde{b}_{i,k}) \mathbf{e}_i} + X^{-\mathbf{e}_k + \sum_{i=1}^{m} \max(0, -\tilde{b}_{i,k}) \mathbf{e}_i} \in \mathcal{F}.$$

Note that (QM_3) agrees with (M_2). The variables $\mathbf{X}' = (X_1', X_2', \dots, X_m')$ are pairwise q-commutative: for all $j \in \tilde{I}$ with $j \neq k$ the integers

$$\beta\left(-\mathbf{e}_k + \sum_{i=1}^{m} \max(0, \tilde{b}_{i,k})\mathbf{e}_i, \mathbf{e}_j\right) = -\lambda_{k,j} + \sum_{i=1}^{m} \max(0, \tilde{b}_{i,k})\lambda_{i,j}$$
$$\beta\left(-\mathbf{e}_k + \sum_{i=1}^{m} \max(0, -\tilde{b}_{i,k})\mathbf{e}_i, \mathbf{e}_j\right) = -\lambda_{k,j} + \sum_{i=1}^{m} \max(0, -\tilde{b}_{i,k})\lambda_{i,j}$$

are equal, because their difference is equal to the sum $\sum_{i=1}^{m} \tilde{b}_{i,k}\lambda_{i,j}$. As we assumed $j \neq k$, the latter sum is the zero entry indexed by (k, j) in the matrix $\tilde{B}^T\Lambda$. So the compatibility condition implies that the variable X_k' q-commutes with all X_j. Hence the variables $\mathbf{X}' = (X_1', X_2', \dots, X_m')$ generate again a based quantum torus whose q-commutativity relations are given by the skew-symmetric matrix Λ'. Moreover, the pair $(\tilde{B}', \Lambda')$ is compatible by [BZ05, Prop. 3.4] and the matrix $\tilde{B}'$ has a skew-symmetrisable principle part.

We conclude that the mutation $\mu_k\left((\tilde{B}', \mathbf{X}', \Lambda')\right) = (\tilde{B}', \mathbf{X}', \Lambda')$ is again an extended quantum seed. It can be readily verified that the mutation of quantum seeds is involutive as in the non-quantum case, i. e. $\mu_k \circ \mu_k\left((\tilde{B}, \mathbf{X}, \Lambda)\right) = (\tilde{B}, \mathbf{X}, \Lambda)$.

A main property of classical cluster algebras are the binomial exchange relations in (M_2). For the quantised version we require pairwise q-commutativity for the quantum cluster variables in a single cluster. This implies that a monomial $X_1^{a_1} X_2^{a_2} \cdot \ldots \cdot X_m^{a_m}$ with $a \in \mathbb{Z}^m$ remains (up to a power of q) a monomial under reordering the quantum cluster variables.

We call two quantum seeds $(\tilde{B}, \mathbf{X}, \Lambda)$ and $(\tilde{B}', \mathbf{X}', \Lambda')$ *mutation equivalent* if one can relate them by a finite sequence of mutations. This defines an equivalence relation on quantum seeds, denoted by $(\tilde{B}, \mathbf{X}, \Lambda) \sim (\tilde{B}', \mathbf{X}', \Lambda')$. The *quantum cluster algebra* $\mathcal{A}_q\left((\tilde{B}, \mathbf{X}, \Lambda)\right)$ associated to a given quantum seed $(\tilde{B}, \mathbf{X}, \Lambda)$ is the $\mathbb{Z}[q^{\pm 1}]$-subalgebra of $\mathcal{F}$ generated by the set

$$\chi\left((\tilde{B}, \mathbf{X}, \Lambda)\right) = \left\{X_i^{\pm 1} \mid i \in F, \right\} \cup \bigcup_{(\tilde{B}', \mathbf{X}', \Lambda') \sim (\tilde{B}, \mathbf{X}, \Lambda)} \left\{X_i' \mid i \in I\right\}.$$

Specialising to $q = 1$ identifies the quantum cluster algebra $\mathcal{A}_q\left((\tilde{B}, \mathbf{X}, \Lambda)\right)$ with the classical cluster algebra $\mathcal{A}\left((\tilde{B}, \mathbf{X})\right) = \mathcal{A}\left((\tilde{B}, (X_i)_{i\in I}, (X_j)_{j\in F})\right)$. Generally, the definitions of classical and quantum cluster algebras admit additional analogies. One such analogy is the quantum Laurent phenomenon, as proven in [BZ05, Cor. 5.2]: we have $\mathcal{A}_q\left((\tilde{B}, \mathbf{X}, \Lambda)\right) \subseteq \mathcal{T}_\Lambda$. Remarkably, $\mathcal{A}_q\left((\tilde{B}, \mathbf{X}, \Lambda)\right)$ and $\mathcal{A}\left((\tilde{B}, \mathbf{X})\right)$ also possess the same exchange graph by [BZ05, Thm. 6.1]. In particular, quantum cluster algebras of finite type are also classified by Dynkin diagrams.

3.2.1 Existence

This subsection is concerned with the existence of a compatible matrix Λ for a given extended exchange matrix $\tilde{B}$. Let $\tilde{B} = \left[\begin{smallmatrix} B \\ C \end{smallmatrix}\right]$ be as in the preceeding section with principal part B and skew-symmetriser D.

Already in the introductory paper to quantum cluster algebras has it been shown that having full rank is a necessary condition for compatible pairs.

Proposition 3.2.3 [BZ05]**.** *If there exists a quantisation of $\mathcal{A}(\tilde{B})$ then $\tilde{B}$ has full rank.*

This in turn raises the question if this condition is also sufficient. This we can answer positively.

Theorem 3.2.4. *If $\tilde{B}$ has full rank then there exists a quantisation of $\mathcal{A}(\tilde{B})$.*

Proof. We show there exists a skew-symmetric $m \times m$-matrix Λ with integer entries and a multiple $D' = \lambda D$ with $\lambda \in \mathbb{Q}^+$ such that $\tilde{B}^T \Lambda = \begin{bmatrix} D' & 0_{n\times(m-n)} \end{bmatrix}$.

By assumption the n column vectors of $\tilde{B}$ are linearly independent elements in $\mathbb{Q}^m$. We can extend this linearly independent set to a basis of $\mathbb{Q}^m$ by adding $m-n$ appropriate column vectors. Hence, there exists an invertible $m \times m$ block matrix $M = \begin{bmatrix} \tilde{B} & \tilde{E} \end{bmatrix}$ whose left block is $\tilde{B}$ and the right block $\tilde{E}$ is of dimension $m \times (m-n)$ We also write $\tilde{E}$ itself in block form as $\tilde{E} = \left[\begin{smallmatrix} E \\ F \end{smallmatrix}\right]$ with an $n \times (m-n)$ matrix E and an $(m-n) \times (m-n)$ matrix F. Of course, the choice for the basis completion is not canonical. In particular, one can choose standard basis vectors for columns of $\tilde{E}$, making it sparse.
Put

$$\Lambda_0 = M^{-T} \begin{bmatrix} DB & DE \\ -E^T D & 0_{(m-n)\times(m-n)} \end{bmatrix} M^{-1} \in \mathrm{Mat}_{m\times m}(\mathbb{Q})$$

and let Λ be an integer multiple of Λ_0 which lies in $\mathrm{Mat}_{m\times m}(\mathbb{Z})$. The matrix Λ is skew-symmetric by construction and the relation $M^T M^{-T} = \mathbb{I}_m$ implies

$$\tilde{B}^T M^{-T} = \begin{bmatrix} \mathbb{I}_n & 0_{n\times(m-n)} \end{bmatrix}.$$

Thus we obtain

$$\tilde{B}^T \Lambda_0 = \begin{bmatrix} DB & DE \end{bmatrix} M^{-1} = D \begin{bmatrix} B & E \end{bmatrix} M^{-1} = D \begin{bmatrix} \mathbb{I}_n & 0_{n\times(m-n)} \end{bmatrix} = \begin{bmatrix} D & 0_{n\times(m-n)} \end{bmatrix}.$$

Scaling the equation yields $\tilde{B}^T \Lambda = \begin{bmatrix} D' & 0_{n\times(m-n)} \end{bmatrix}$ for some multiple D' of D. □

Theorem 3.2.5. *The coefficient-free cluster algebra $\mathcal{A}(B)$ admits a quantisation if and only if one of the following equivalent conditions is satisfied:*

1. *B has full rank,*

2. *the underlying undirected graph Q of B admits a perfect matching.*

Proof. First of all, if n is odd, then B cannot be of full rank since $\det(B) = (-1)^n \det(B)$ implies $\det(B) = 0$. Hence we may assume that n is even. In addition we may make the assumption that B is skew-symmetric as multiplication with a skew-symmetriser D does not change the rank of B.

In the case of even skew-symmetric matrices, Theorem 2.2.5 asserts the existence of the Pfaffian and indeed the equivalence of the two given conditions.
In particular, if there exists no perfect matching of the underlying undirected graph of Q, then the Pfaffian $\mathrm{Pf}(B)$ vanishes, the determinant $\det(B)$ is zero, the matrix B does not have full rank and there exists no quantisation of $\mathcal{A}(B)$. □

Corollary 3.2.6. *A coefficient-free finite type cluster algebra admits a quantisation if and only if it is of Dynkin type A_n for even n or of type E_6 or E_8.*

3.2.2 Uniqueness

We have established in the previous discussion that a cluster algebra $\mathcal{A}(\tilde{B})$ admits a quantisation if and only if $\tilde{B}$ has full rank. Since the rank of an extended exchange matrix is mutation invariant, one can use any seed to check whether a cluster algebra admits a quantisation.

This immediately imposes the question of uniqueness of such quantisations. As a first answer in this direction, let us reformulate the statement in terms of bilinear forms as suggested in a private communication by Zelevinsky.

Lemma 3.2.7. *If there exists up to 1 frozen variable, then the quantisation of $\mathcal{A}(\tilde{B})$ is essentially unique.*

Proof. Assume $(\tilde{B}, \Lambda)$ is a compatible pair yielding a quantisation of $\mathcal{A}(\tilde{B})$. By Proposition 3.2.3 the column vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ of $\tilde{B}$ are linearly independent over $\mathbb{Q}$. Let $V' = \mathrm{span}_{\mathbb{Q}}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$ be the column space of $\tilde{B}$. The column vectors $\tilde{\mathbf{e}}_{n+1}, \tilde{\mathbf{e}}_{n+2}, \ldots, \tilde{\mathbf{e}}_m$ of $\tilde{E}$ as in Theorem 3.2.4 extend $(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$ to a basis of $V = \mathbb{Q}^m$. Let $V'' = \mathrm{span}_{\mathbb{Q}}(\tilde{\mathbf{e}}_{n+1}, \tilde{\mathbf{e}}_{n+2}, \ldots, \tilde{\mathbf{e}}_m)$. The compatibility condition $\tilde{B}^T \Lambda = \begin{bmatrix} D' & 0_{n\times(m-n)} \end{bmatrix}$ gives that for any given D', the skew-symmetric bilinear form $V \times V \to \mathbb{Q}$ is completely determined on $V' \times V$, hence also on $V \times V'$. Such a bilinear form can be chosen freely on $V'' \times V''$ giving a $\frac{1}{2}(m-n-1)(m-n)$-dimensional solution space.
In particular, the quantisation is unique up to a scalar when there are at most 1 frozen vertices present. □

In the construction in the proof of Theorem 3.2.4 we chose some $m \times (m-n)$ integer matrix $\tilde{E}$ which completed a basis for $\mathbb{Q}^m$. This choice we now reformulate by giving a generating set of integer matrices for the equation

$$\tilde{B}^T \Lambda = \begin{bmatrix} 0_{n\times n} & 0_{n\times(m-n)} \end{bmatrix}. \tag{3.9}$$

As previously remarked this ambiguity does not occur for up to one frozen vertex, hence we may start with the case $m = n+2$.

Two frozen variables

For the time being assume there exist precisely two frozen variables.

For distinct $i, j \in \tilde{I}$ define a reduced index set $R(i,j)$ as the n-element subset of $\tilde{I}$ in which i and j do not occur. Let $A = (a_{i,j})_{i\in\tilde{I}, j\in I}$ be an arbitrary $m \times n$ integer matrix. For a subset $S \subseteq \tilde{I}$ denote by A_S the submatrix of A given by the rows indexed by S, i.e. $A_S = (a_{k,\ell})_{k\in S, j\in I}$. To the matrix A we associate the skew-symmetric $m \times m$ integer matrix $M = M(A) = \big(m_{i,j}\big)_{i,j\in\tilde{I}}$ with entries

$$m_{i,j} = \begin{cases} (-1)^{i+j} \cdot \det\big(A_{R(i,j)}\big), & i < j, \\ 0, & i = j, \\ (-1)^{i+j+1} \cdot \det\big(A_{R(i,j)}\big), & j < i. \end{cases} \tag{3.10}$$

Lemma 3.2.8. *For A an $m \times n$ integer matrix, we obtain*

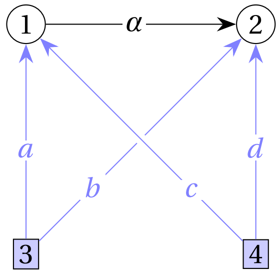
$$A^T \cdot M = \begin{bmatrix} 0_{n\times n} & 0_{n\times(m-n)} \end{bmatrix}.$$

Proof. By definition, we have

$$\big(A^T \cdot M\big)_{i,j} = \sum_{k=1}^{m} a_{k,i} m_{k,j} = \sum_{k\in\tilde{I}\setminus\{j\}} a_{k,i} m_{k,j}.$$

Now let A_j be the matrix we obtain from A by removing the j-th row and A_j^i the matrix that results from attaching the i-th column of A_j to itself on the right. Then $\det(A_j^i) = 0$ and we observe that using the Laplace expansion along the last column, we obtain the right-hand side of the above equation up to sign change. The claim follows. □

Example 3.2.9. Let α, a, b, c and d be positive integers. Then consider the quiver $\tilde{Q}$ given by:



The matrices $\tilde{B}$ and M are

$$\tilde{B} = \begin{bmatrix} 0 & \alpha \\ -\alpha & 0 \\ a & b \\ c & d \end{bmatrix}, \qquad M = \begin{bmatrix} 0 & -ad+bc & -\alpha d & \alpha b \\ ad-bc & 0 & \alpha c & -\alpha a \\ \alpha d & -\alpha c & 0 & -\alpha^2 \\ -\alpha b & \alpha a & \alpha^2 & 0 \end{bmatrix},$$

and we immediately see the result of the previous lemma, namely $\tilde{B}^T \cdot M = \begin{bmatrix} 0_{2\times 2} & 0_{2\times 2} \end{bmatrix}$.

From two to finitely many frozen variables

Now let $n+2 < m$ and as before let $A \in \mathrm{Mat}_{m\times n}(\mathbb{Z})$ be some rectangular integer matrix. Choose a subset $N \subset \tilde{I}$ of cardinality n and obtain a partition of the index set $\tilde{I}$ of the rows of A as $\tilde{I} = N \sqcup R$. Note that $|R| = m - n$. For distinct $i, j \in R$ set the *extended index set associated to* i, j to be

$$E(i,j) := N \cup \{i,j\}.$$

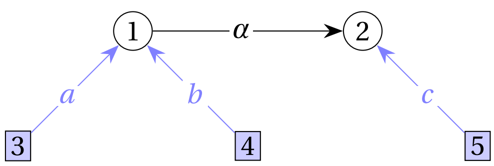
By Lemma 3.2.8 (after a reordering of rows) and slightly abusing the notation, there exists an $(n+2)\times(n+2)$ integer matrix $M_{E(i,j)} = \left(m_{r,s}\right)$ such that

$$A^T_{E(i,j)} \cdot M_{E(i,j)} = \begin{bmatrix} 0_{n\times n} & 0_{n\times 2} \end{bmatrix}. \tag{3.11}$$

Now let $\mathfrak{M}_{E(i,j)} = \mathfrak{M}_{E(i,j)}(A) = \left(\mathfrak{m}_{r,s}\right)_{r,s\in\tilde{I}}$ be the *enhanced solution matrix associated to* i, j, the $m \times m$ integer matrix we obtain from $M_{E(i,j)}$ by filling the entries labeled by $E(i,j)\times E(i,j)$ with $M_{E(i,j)}$ consecutively and setting all other entries to zero.

Example 3.2.10. Consider the quiver $\tilde{Q}$ with associated exchange matrix $\tilde{B}$ as below:

$\tilde{Q}$:



and

$$\tilde{B} = \begin{bmatrix} 0 & \alpha \\ -\alpha & 0 \\ a & 0 \\ b & 0 \\ 0 & c \end{bmatrix}.$$

We choose $N = \{1,2\}$, assuming $\alpha \neq 0$ and get the following matrices $M_{E(i,j)}$ and their enhanced solution matrices for distinct $i, j \in \{3,4,5\}$:

$$M_{E(3,4)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha b & -\alpha a \\ 0 & -\alpha b & 0 & -\alpha^2 \\ 0 & \alpha a & \alpha^2 & 0 \end{bmatrix}, \qquad \mathfrak{M}_{E(3,4)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha b & -\alpha a & 0 \\ 0 & -\alpha b & 0 & -\alpha^2 & 0 \\ 0 & \alpha a & \alpha^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$M_{E(3,5)} = \begin{bmatrix} 0 & -ac & -\alpha c & 0 \\ ac & 0 & 0 & -\alpha a \\ \alpha c & 0 & 0 & -\alpha^2 \\ 0 & \alpha a & \alpha^2 & 0 \end{bmatrix}, \qquad \mathfrak{M}_{E(3,5)} = \begin{bmatrix} 0 & -ac & -\alpha c & 0 & 0 \\ ac & 0 & 0 & 0 & -\alpha a \\ \alpha c & 0 & 0 & 0 & -\alpha^2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha a & \alpha^2 & 0 & 0 \end{bmatrix},$$

$$M_{E(4,5)} = \begin{bmatrix} 0 & -bc & -\alpha c & 0 \\ bc & 0 & 0 & -\alpha b \\ \alpha c & 0 & 0 & -\alpha^2 \\ 0 & \alpha b & \alpha^2 & 0 \end{bmatrix}, \qquad \mathfrak{M}_{E(4,5)} = \begin{bmatrix} 0 & -bc & 0 & -\alpha c & 0 \\ bc & 0 & 0 & 0 & -\alpha b \\ 0 & 0 & 0 & 0 & 0 \\ \alpha c & 0 & 0 & 0 & -\alpha^2 \\ 0 & \alpha b & 0 & \alpha^2 & 0 \end{bmatrix}.$$

Here we highlighted the added 0-rows/-columns in gray. By considering the lower right 3×3 matrices of $\mathfrak{M}_{E(3,4)}, \mathfrak{M}_{E(3,5)}, \mathfrak{M}_{E(4,5)}$ we observe that these matrices are linearly independent. This we generalise in the theorem below.

Theorem 3.2.11. *Let $A \in \mathrm{Mat}_{m\times n}(\mathbb{Z})$ as above. Then for distinct $i, j \in R$ we have*

$$A^T \cdot \mathfrak{M}_{E(i,j)} = 0_{n\times m}.$$

Furthermore, if A is of full rank and N is chosen such that the submatrix A_N yields the rank, then the matrices $\mathfrak{M}_{E(i,j)}$ are linearly independent.

Proof. By construction, for $s \in R\backslash\{i, j\}$ the s-th column of $\mathfrak{M}_{E(i,j)}$ contains nothing but zeros. Hence for arbitrary $r \in \tilde{I}$, we have

$$\left(A^T \cdot \mathfrak{M}_{E(i,j)}\right)_{r,s} = 0. \tag{3.12}$$

Now let $s \in E(i, j)$. Then

$$\sum_{k=1}^{m} a_{k,r}\mathfrak{m}_{k,s} = \sum_{k\in E(i,j)} a_{k,r}\, m_{k,s} = 0,$$

by Lemma 3.2.8, completing the first statement.
Without loss of generality, assume $i < j$ and $N = I$. Then by assumption on the rank, $\beta := (-1)^{i+j}\det(A_I) \neq 0$ and by construction, $\mathfrak{M}_{E(i,j)}$ is of the form as in Figure 3.2. Then $\pm\beta$ is the only entry of the submatrix of $\mathfrak{M}_{E(i,j)}$ indexed by $F \times F$. This immediately provides

the linear independence. $\square$

$$
\begin{array}{c|c|ccccccc}
 & 1\cdots n & n+1 & \cdots & i & \cdots & j & \cdots & m \\
1 & & & & & & & & \\
\vdots & * & & & & * & & & \\
n & & & & & & & & \\
\hline
n+1 & & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
\vdots & & \vdots & \ddots & & & & & \vdots \\
i & & 0 & & 0 & & \beta & & 0 \\
\vdots & * & \vdots & & & \ddots & & & \vdots \\
j & & 0 & & -\beta & & 0 & & 0 \\
\vdots & & \vdots & & & & & \ddots & \vdots \\
m & & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0
\end{array} .
$$

Figure 3.2: An example of the form of enhanced solution matrices

As an immediate consequence we obtain that there are at least $\binom{m-n}{2}$ many $m \times m$ integer matrices M satisfying

$$A^T \cdot M = \begin{bmatrix} 0_{n\times n} & 0_{n\times(m-n)} \end{bmatrix}.$$

Together with the final remark of the proof for Lemma 3.2.7, we thus conclude that the above constructed matrices form a basis for the space of solutions Λ of the homogeneous equation (3.9).

Recall that every skew-symmetriser is an $\mathbb{N}^+$-linear combination of the fundamental skew-symmetrisers of the connected components of B. This allows us to rephrase the findings above as follows.

Corollary 3.2.12. *Let $\tilde{B} = \left[\begin{smallmatrix} B \\ C \end{smallmatrix}\right]$ be an $m \times n$ extended exchange matrix of full rank and r the number of connected components of the quiver associated to the principal part B (or to DB if B is skew-symmetrisable and not skew-symmetric). Then the solution space of matrices Λ satisfying the compatibility equation $\tilde{B}^T \Lambda = \begin{bmatrix} D' & 0 \end{bmatrix}$ to a given skew-symmetriser D' is a vector space over the rational numbers of dimension $\binom{m-n}{2}$.*

In particular, the set of all quantisations lies in a rational vector space of dimension $r + \binom{m-n}{2}$.

By Theorem 3.2.11 the vector space of matrix solutions in the first statement of the above corollary can be explicitly constructed:

Corollary 3.2.13. *(a) All solutions of the compatibility equation $\tilde{B}^T \Lambda = \begin{bmatrix} D' & 0 \end{bmatrix}$ to a fixed skew-symmetriser D' can be constructed as the sum of a solution Λ_0 and a linear combination of all $\mathfrak{M}_{E(i,j)}$ for $i, j \in \tilde{I}$.*

(b) *In the special case where the principal part of $\tilde{B}$ is already invertible, quantisations of full subquivers with all mutable and two frozen vertices yield a basis of the homogeneous solution space.*

In Appendix C.1 we provide Sage-code with methods for the class `ClusterQuiver` that can be used to construct both the compatible matrix Λ from Theorem 3.2.4, as well as the enhanced solution matrices of Theorem 3.2.11.

4

Green sequences

Since the notion of green and red vertices have been defined in [Kel11], sequences of mutations at thus coloured mutable vertices are intensely studied. Restricting to mutations at green vertices alone yields an orientation of the ordinary exchange graph as considered in [BY13; KY14; KQ15].

The existence of a maximal green sequence for a given particular quiver without loops and 2-cycles is of special interest. As proven in [BDP14], all such quivers which are additionally acyclic admit a maximal green sequence. This result has since been extended to all finite-type cluster algebras except those which are of mutation type $\mathbb{X}_7$, cf. [Mil16] for further details. The known cases of cyclic quivers which do not admit a maximal green sequence are rather limited, see [Sev14; Mul16] for examples, and we pursue the question which simply-laced quiver possesses this property in Theorem 4.3.9.

Our considerations are predicated on recent advances in the study of green sequences from [Br17] which we first recall in Section 4.1. We continue by combinatorially discussing particular quivers on up to four mutable vertices and reveal in Section 4.3 the smallest — with respect to the number of mutable vertices — simply-laced quiver which does not admit a maximal green sequence. We use the same combinatorial techniques to further provide a new inifinite class of quivers of the same characteristic.

In Section 4.4 we place the preceding combinatorial results in the context of periodicities in the oriented exchange graph and study the case of extended Dynkin type $\tilde{A}_{n-1}$ quivers in detail.

4.1 Definitions and fundamental results

We recall definitions and results for green sequences from [BDP14].

For this chapter let Q be a finite quiver without loops and 2-cycles on $n \geq 1$ vertices and $\tilde{Q}$ its principal extension. Let the set of vertices of $\tilde{Q}$ be given by $\tilde{I} := \{1, 2, \dots, 2n\} = \{1, \dots, n\} \cup \{n+1, \dots, 2n\}$, where $I := \{1, \dots, n\}$ is the set of mutable vertices and $F := \{n+1, \dots, 2n\}$ that of frozen ones. Further let $\tilde{Q}'$ be some quiver which is mutation equivalent to $\tilde{Q}$. We may refer to the subquiver Q of $\tilde{Q}$ on the mutable vertices I as the *mutable part*. In addition, let $\tilde{B} = \left[\begin{smallmatrix} B \\ \mathbb{I}_n \end{smallmatrix}\right]$ $\left(\text{resp. } \tilde{B}'' = \left[\begin{smallmatrix} B' \\ C' \end{smallmatrix}\right]\right)$ be the associated extended exchange matrix of $\tilde{Q}$ $\left(\text{resp. } \tilde{Q}'\right)$.

Definition 4.1.1. A mutable vertex v in $\tilde{Q}'$ is called *green* if there exists an arrow from F to v, or in other words, there exists $i \in F$ such that the arrow $i \to v$ is contained in the arrow set of $\tilde{Q}'$. If on the other hand there exists an arrow from v to F in $\tilde{Q}'$ then the vertex v is called *red*.

It is not obvious and highly non-trivial that a mutable vertex v cannot be simultaneously green and red. Following [BDP14, Theorem 2.6], this property ensured by the sign-coherence of **c**-vectors as detailed in Theorem 3.1.11. This conclusion makes use of the correspondence between principally extended quivers without loops and 2-cycles and extended exchange matrices as discussed in Section 3.1, which we will repeatedly employ without explicitly pointing it out.

Another not immediately obvious property is given in the following theorem.

Theorem 4.1.2 [BDP14, Theorem 2.6]**.** *Any mutable vertex in $\tilde{Q}'$ is either green or red.*

Remark 4.1.3. The definition of green respectively red vertices goes back to [Kel11] in the context of stability conditions and quantum dilogarithm identities. Yet the initial definition reversed green respectively red vertices, i.e. it started out with a quiver Q with associated C-matrix $-I_n$ instead of I_n in the case of principal extensions.
Of these two variants of the definition of green and red vertices the one used here seems to have become dominant in the literature.

Definition 4.1.4. A sequence of mutations $\mathbf{i} = (i_1, \dots, i_r)$ for $r \in \mathbb{N}$ starting in $\tilde{Q}'$ is called a *green sequence* (resp. *red sequence*) if i_j is a green (resp. red) mutable vertex in

$$\mu_{(i_1,\dots,i_{j-1})}\left(\tilde{Q}'\right) = \mu_{i_{j-1}} \circ \cdots \circ \mu_{i_1}\left(\tilde{Q}'\right),$$

for all $1 \leq j \leq r$. If in addition $\mu_{\mathbf{i}}\left(\tilde{Q}'\right)$ has no green vertex then $\mathbf{i}$ is called a *maximal green sequence*. In this situation $\mu_{\mathbf{i}}\left(\tilde{Q}'\right)$ is said to be *all-red*.

When drawing quivers, we colour mutable vertices according to Definition 4.1.1. See Figure 4.2 for examples. One particular question regarding maximal green sequences is whether the quiver at the end of such a sequence has a particular form. The answer turns out to be rather nice, but we need to introduce additional notion before declaring it.

Definition 4.1.5 [BDP14, Section 1.2]**.** Let $\tilde{Q}_1, \tilde{Q}_2$ be mutation equivalent to $\tilde{Q}$ and $\tilde{B}_1 = \left[\begin{smallmatrix} B_1 \\ C_1 \end{smallmatrix}\right]$, $\tilde{B}_2 = \left[\begin{smallmatrix} B_2 \\ C_2 \end{smallmatrix}\right]$ the associated respective extended exchange matrices. The matrices $\tilde{B}_1$ and $\tilde{B}_2$ are said to be *isomorphic* if there exists a permutation $\sigma \in S_n$ such that

$$
\begin{aligned}
B_2 &= P_\sigma^T B_1 P_\sigma, \\
C_2 &= C_1 P_\sigma,
\end{aligned}
$$

where we denote by P_σ the permutation matrix associated to σ.
In the language of quivers, this isomorphism of exchange matrices amounts to permuting the labels of mutable vertices while keeping those of frozen once fixed. We denote the isomorphism on extended exchange matrices and quivers by $\tilde{B}_1 \cong \tilde{B}_2$ and $\tilde{Q}_1 \cong \tilde{Q}_2$.

Definition 4.1.6 [BDP14, Definition 1.10]**.** The *oriented exchange graph* $\mathbf{EG}(Q)$ of Q is the (possibly infinite) directed graph whose vertices are the isomorphism classes of quivers mutation equivalent to $\tilde{Q}$ and there exists an arrow $[\tilde{Q}_1] \to [\tilde{Q}_2]$ between two such isomorphism classes if and only if there exists a green vertex k in $\tilde{Q}_1$ such that $\mu_k(\tilde{Q}_1) \cong \tilde{Q}_2$.

With these notions in place, we are now in a position to formulate the form of a quiver at the end of a maximal green sequence.

Proposition 4.1.7 [BDP14, Proposition 2.10 (2)]**.** *Assume there exists a maximal green sequence* $\mathbf{i} = (i_1, \ldots, i_r)$ *for* $\tilde{Q}$ *whose extended exchange matrix is given by* $\left[\begin{smallmatrix} B \\ I_n \end{smallmatrix}\right]$*. Denote by* $\mu_{\mathbf{i}}(\tilde{B}) = \tilde{B}_\Omega = \left[\begin{smallmatrix} B_\Omega \\ C_\Omega \end{smallmatrix}\right]$ *the extended exchange matrix at the end of the maximal green sequence. Then* $\tilde{B}_\Omega$ *is isomorphic to* $\left[\begin{smallmatrix} B \\ -I_n \end{smallmatrix}\right]$.

In terms of the oriented exchange graph, the above proposition yields that if a sink in this particular digraph exists, it is unique.Another important property of the oriented exchange is the following.

Theorem 4.1.8 [BDP14, Proposition 2.14]**.** *The oriented exchange graph* $\mathbf{EG}(Q)$ *has no oriented cycles.*

The existence of a maximal green sequence given an arbitrary quiver Q without loops and 2-cycles is still an open problem. A large family of examples for which such sequences do occur has been established in [BDP14] which we state in the following Lemma.

Lemma 4.1.9 [BDP14, Lem. 2.20]**.** *If* Q *is acyclic then any sink order yields a maximal green sequence in* $\tilde{Q}$*, where an order* $v_1 < v_2 < \ldots < v_n$ *on the vertices of* Q *is called a* sink order *if* v_{i+1} *is a sink in* $\mu_{(v_1, v_2, \ldots, v_i)}(Q)$.

Example 4.1.10. Consider the principal extension $\tilde{Q}$ of the orientation of $\tilde{A}_2$ as shown in Figure 4.1.

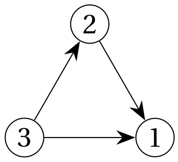


Figure 4.1: Quiver Q of type $\tilde{A}_2$

Then a sink order of $\tilde{Q}$ is given by $(1,2,3)$ and we compute the associated mutation sequence in Figure 4.2.

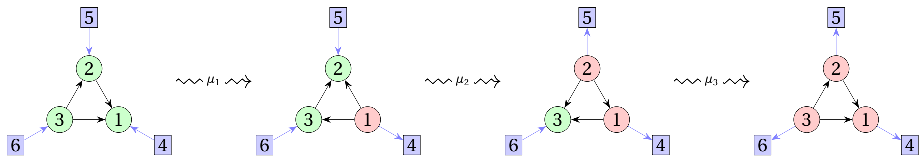


Figure 4.2: Mutation along a sink order of a quiver of type $\tilde{A}_2$

A variety of properties — such as being of finite type — of cluster algebras has been proven by showing that these do not depend on a particular choice of a seed but rather on its mutation equivalence class. Such properties thus constitute invariants of the mutation equivalence class and of the cluster algebra itself. Unfortunately such behaviour has been shown to not hold for the existence of maximal green sequences.

Theorem 4.1.11 [Mul16, Cor. 2.3.3]**.** *Let $\tilde{Q}''$ be mutation equivalent to $\tilde{Q}'$ and assume that $\tilde{Q}'$ admits a maximal green sequence. Then there does not necessarily exist a maximal green sequence for $\tilde{Q}''$, i.e. the existence of a maximal green sequence is not an invariant of the mutation equivalence class.*

Example 4.1.12. Let us shortly give the prime example of [Mul16] providing a counterexample of the invariance of the existence of maximal green sequences. Take Q_M to be the quiver in Figure 4.3.

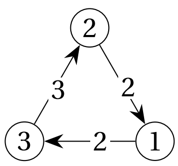


Figure 4.3: Quiver Q_M providing a counterexample for the invariance of the existence of maximal green sequences under mutation

Then $\mu_{(1,3)}(Q_M)$ is acyclic and by Lemma 4.1.9 there exists a maximal green sequence turning its principal extension all-red. For the unmutated quiver Q_M on the other hand, Theorem 4.2.4 of the subsequent section asserts that there does not exist a maximal green sequence for the principal extension of Q_M.

Nevertheless, there do exist further classes of quivers than acyclic ones for which the existence or non-existence of maximal green sequences has been proven. We refer to [Mil16] for a recent overview. There it is also shown that any quiver in the mutation equivalence class of a mutation finite quiver admits a maximal green sequence, except the quivers inside the class of the exceptional type $\mathbb{X}_7$. The latter has been shown to not admit such sequences in [Sev14] by making explicit use of the fact that its mutation equivalence class consists of two members only.

4.2 Permissible vertices

In the recent work [Br17] new techniques have been developed in the study of green and red sequences. We shortly present some of the definitions and results which we will subsequently use, keeping the notation and assumptions of the preceding section.

Definition 4.2.1 [Br17, Definition 2.1.6]**.** For $k \in 1,2,\dots n$ and $\varepsilon \in \{+,-\}$ let H^{ε} be the set of C-matrices of an extended exchange matrix mutation equivalent to $\tilde{B}$ such that the k-th row of the associated G-matrix has non-negative (resp. non-positive) entries corresponding to $\varepsilon = +$ (resp. $\varepsilon = -$). These sets are called *k-hemispheres*.

By the sign-coherence of g-vectors, k-hemispheres are well-defined. The most important feature of this notion is that C-matrices change their respective k-hemisphere only on very particular occasions. We denote by $\mathbf{e}_i$ the i-th standard basis vector of $\mathbb{Q}^n$ as before.

Lemma 4.2.2 [Br17, Lemma 2.1.7]**.** *Let C' and G' be the associated C- and G-matrix of the extended exchange matrix $\tilde{B}'$. Then the following properties hold for $j,k \in \{1,\dots,n\}$:*

1. *If the j-th column in C' equals $\mathbf{e}_k$ (resp. $-\mathbf{e}_k$) then the k-th row of G is non-negative (resp. non-positive), yielding that C' is an element of H_k^+ (resp. of H_k^-).*
2. *The C-matrices C' and $\mu_j(C')$ are in different k-hemispheres if and only if the j-th column in C' equals $\pm\mathbf{e}_k$*

The above lemma is built upon the relation (3.5) and expressing the mutation of C-matrices by matrix multiplications as in (3.4). The central result in [Br17] is the so-called *rotation lemma*, which states that the existence of a maximal green sequence can be moved along this very sequence in the following fashion.

Theorem 4.2.3 [Br17, Thm. 2.2.4]**.** *If $\mathbf{i} = (i_1,\dots,i_r)$ is a maximal green sequence for $\tilde{Q}$ and $\sigma \in S_n$ is the associated permutation, then $(i_2,\dots,i_r,\sigma(i_1))$ is a maximal green sequence for $\mu_{i_1}(\tilde{Q})$. In particular, any quiver that appears along a maximal green sequence itself admits such a sequence.*

One particular consequence of the rotation lemma asserts what kind of mutations can be ignored when searching for maximal green sequences inside the oriented exchange graph.

Theorem 4.2.4 [Br17, Cor. 3.3.2]**.** *Assume $\mathbf{i} = (i_1,\dots,i_r)$ is a maximal green sequence for $\tilde{Q}$. Then for every index $1 \le t \le r$ there does not exist a vertex v such that there is more than one arrow from $i_t \to v$ in $\mu_{(i_1,i_2,\dots,i_{t-1})}(\tilde{Q})$.*

As i_t is assumed to be green for any $1 \le t \le r$ inside $\tilde{Q}_{t-1} = \mu_{(i_1,i_2,\dots,i_{t-1})}(Q)$ from Theorem 4.2.4, there exist arrows from frozen vertices into i_t in $\tilde{Q}_{t-1}$ but not from i_t to frozen ones. Thus the above theorem implies that in a maximal green sequence there exists no multiedge leaving the vertex at which a green mutation is performed at.

Definition 4.2.5. We call a mutable vertex v in $\tilde{Q}'$ *permissible* if there does not exist another mutable vertex w such that there are multiple arrows $v \to w$ in $\tilde{Q}'$. If all green vertices of a given quiver are not permissible, we call it a *green dead end*.
If $\mathbf{i} = (i_1, i_2, \ldots, i_r)$ is a sequence of mutations starting in $\tilde{Q}'$, we call this sequence *permissible* if each vertex i_s is in $\mu_{(i_1,\ldots,i_{s-1})}(\tilde{Q}')$ for all $1 \leq s \leq r$. If $\mathbf{i}$ is both green and permissible, we simply call it *green permissible*.

Theorem 4.2.4 says that whenever one reaches a green dead end, no green sequence can exist which yields an all-red quiver. As we use green dead ends and Theorem 4.2.4 extensively in the subsequent discussion, let us consider one essential observation in some detail.
When drawing quivers, we now colour green and nonpermissible vertices orange, see Figure 4.4 for examples of this convention.

Example 4.2.6. Assume we are in a situation where mutable vertices v, w and a frozen vertex f exist, such that v is green but not permissible and there exist $a > 1$ many arrows from v to w and $b > 1$ many from f to v. Then mutation first at v and then at w is shown in Figure 4.4.

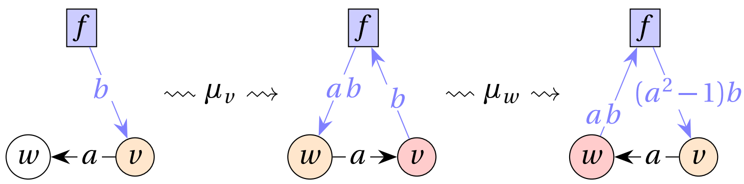


Figure 4.4: Mutation of green and nonpermissible vertices

Since the number of arrows between the frozen vertex f and the mutable vertices is strictly increasing, induction shows that green and nonpermissible vertices in a situation as in the left arrow of Figure 4.4 allow for an infinite green sequence. Note that green-ness of w in the second quiver of Figure 4.4 is ensured by the sign-coherence of c-vectors if there existed further frozen vertices. One may even assume there exist $c \leq \frac{ab}{2}$ many arrows from w to f in the initial quiver and still observe the same behaviour.

For a cluster algebra $\mathcal{A}$ of finite type and principal extension, no green dead ends exist as no multiple arrows occur in the mutable parts of any quiver of the associated isomorphism classes. This is not the case for quivers of extended Dynkin type for which multiple arrows inside the mutation equivalence class do occur. The occurrence of green dead ends for such quivers is part of the discussion in the subsequent section in which green dead ends play a dominant rôle.

4.3 The oriented pentatope

In this section we present the smallest — by the number of vertices — simply-laced quiver without a maximal green sequence in Theorem 4.3.9. The proof builds upon Theorem 4.2.4 and combinatorial investogations of particular quivers with up to four mutable vertices. The same techniques are then used to provide in Theorem 4.3.11 a new infinite class of quivers on 5 mutable vertices that do not admit maximal green sequences.

First, let us motivate the subsequent combinatorial discussion by returning to a particular example from the preceeding section.

Example 4.3.1. Let $\tilde{Q}$ be as in Example 4.1.10 the principal extension of a quiver of type $\tilde{A}_2$. Then by [BDP14, Thm. 5.4] there are only finitely many maximal green sequences for $\tilde{Q}$. Five ones are easy to determine: $(1,2,3)$, $(1,3,2,3)$, $(2,1,2,3)$, $(2,1,3,2,3)$ and $(3,1,3,2,1)$. But do there exist any more and if not, how can we prove the non-existence? We will return to this question later.

Let Q_{tri} be the quiver with three mutable and one frozen vertex as in Figure 4.5 with positive integers $a, b, c > 0$.

Proposition 4.3.2. *If $a > b > c > 0$, then there exists no green permissible sequence turning Q_{tri} all-red.*

Proof. We observe that mutating the mutable part of Q_{tri} at the unique source 3, we once again obtain an unoriented 3-cycle in which 3 takes the place of 1, 2 of 3 and 1 of 2. Induction shows that the sequence $(3,2,1,3,2,1,3,\ldots)$ always mutates at sources, there exists at each mutation step precisely one sink, one source and one transit vertex — a vertex being neither sink nor source — and no multiple arrows occur.
The C-matrix of $\mu_3(Q_{\text{tri}})$ has entries $a-b$, $a+c$, $-a$ and since $a > b > c > 0$, the vertex 1 is green, as is the vertex 2 and 3 is red. Then induction on the length of the mutation sequence $(3,2,1,3,2,1,3,\ldots)$ together with the initial assumption $a > b > c > 0$ yields that this sequence is indeed a green permissible sequence.
Fix $\mathbf{i} = (3,2,1,3,2,1,3,\ldots)$ of length $r > 0$. If we mutate at the unique transit vertex in $\mu_{\mathbf{i}}(Q_{\text{tri}})$

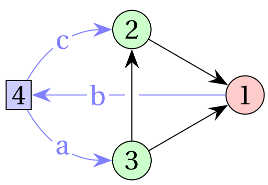


Figure 4.5: Quiver Q_{tri} whose mutable part is of size 3

— the only green vertex besides the unique sink — a second arrow from the unique source to the unique sink of $\mu_{\mathbf{i}}(Q_{\text{tri}})$ gets created. The initial condition $a > b > c > 0$ guarantees that the sink remains red even after mutation at the transit vertex, hence this mutation yields a green dead end.
This shows that sequences of the form $(3,2,1,3,2,1,3,\ldots)$ are the only green permissible mutation sequences starting in Q_{tri}, none of which terminates in an all-red quiver. □

Remark 4.3.3. Following analogous arguments as in the proof above, for integers

- $g \geq f > 1$ and $h \geq 0$ or
- $g > f \geq 1$ and $h \geq 0$

the mutable vertex v in the quiver of Figure 4.6 remains an infinite source under mutation sequences of the form $(3,2,1,3,2,1,3,\ldots)$.

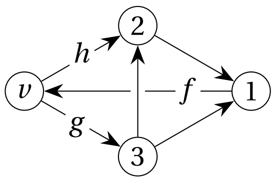


Figure 4.6: 4-point quiver extending Figure 4.5

In particular, if the mutable part of the quiver Q_{tri} from Proposition 4.3.2 is extended by a single mutable vertex such that the new node v is connected in the above fashion, the sequences $(3,2,1,3,2,1,3,\ldots)$ are the only ones which are both green and permissible.
What is more, if $g > f > h > 0$ then the vertex v acts in the very same way as the frozen vertex in Proposition 4.3.2.

Slightly extending the quiver considered in Proposition 4.3.2, let Q'_{tri} be the quiver as in Figure 4.7 with positive integers $a, b, c, d > 0$.

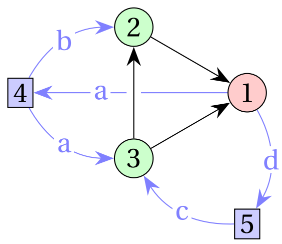


Figure 4.7: Quiver Q'_{tri} whose mutable part is of size 3

Corollary 4.3.4. *If $a > b > 0$ and $c > d > 0$, then there exists no green permissible sequence turning Q'_{tri} all-red.*

Proof. Utilising the same mutation sequence as in the proof of Proposition 4.3.2 and induction yields the statement. □

Now we are in a position to return to the motivating example of the combinatorial discussion above.

Example 4.3.5. Let $\tilde{Q}$ be as in Example 4.3.1 and Q its mutable part. One can easily compute that in the oriented exchange graph $\mathbf{EG}(Q)$, all possible green permissible directions of the branches starting in (1),(2) and (3,1) have been exploited, i.e. $(1,2,3)$, $(1,3,2,3)$, $(2,1,2,3)$, $(2,1,3,2,3)$ and $(3,1,3,2,1)$ are the only maximal green sequence in these parts of $\mathbf{EG}(Q)$. Thus, we only need to further consider the green permissible sequences starting with $(3,2)$. The quiver $\mu_{(3,2,3)}(\tilde{Q})$ is a green dead end, hence it is sufficient to concentrate on the branch of $\mathbf{EG}(Q)$ following the green permissible sequence $(3,2,1)$. The quiver which results from applying $(3,2,1)$ to $\tilde{Q}$ is shown in Figure 4.8.

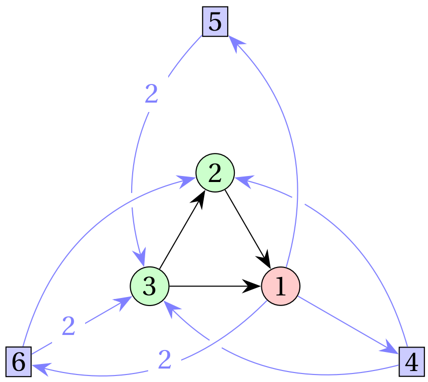
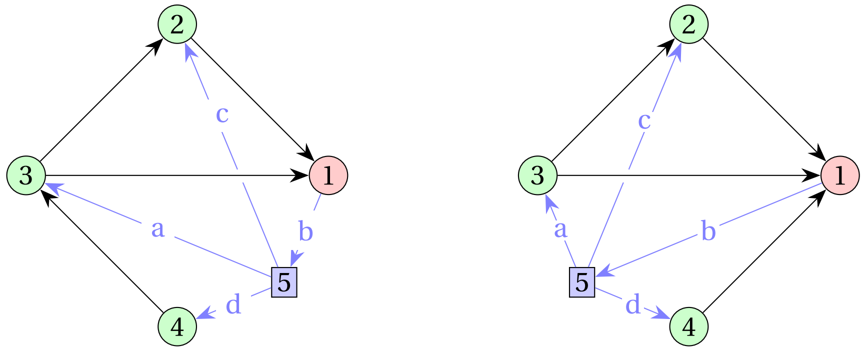


Figure 4.8: Principally extended quiver of type $\tilde{A}_2$ after mutation sequence $(3,2,1)$

Corollary 4.3.4 applied to the frozen vertices 5 and 6 together with the sign-coherence of c-vectors yields $(3,2,1)$ cannot be continued to form a maximal green sequence for $\tilde{Q}$. Hence, we have answered the question posed in Example 4.3.1: the principal extension of the quiver from Figure 4.1 of type $\tilde{A}_2$ admits only five maximal green sequences.

For later use we extend the above results to particular quivers on four mutable vertices as follows. Let $Q_{\text{tri}}^{\text{source}}$ be the quiver in Figure 4.9a and $Q_{\text{tri}}^{\text{sink}}$ the one in Figure 4.9b. We apply the results on quivers with three mutable vertices to these two quivers with four mutable vertices in the following lemma.

Note that we slightly misuse the definition of *oriented exchange graphs* which we apply to general quivers with frozen vertices in the following discussion, not necessarily implying a principal extension.



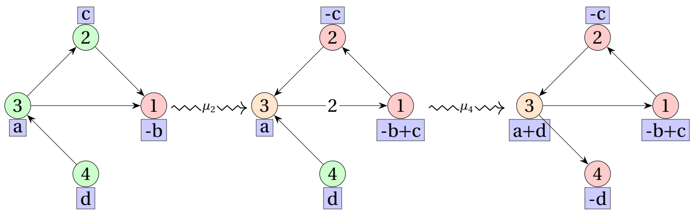
(a) Quiver $Q_{\text{tri}}^{\text{source}}$ (b) Quiver $Q_{\text{tri}}^{\text{sink}}$

Figure 4.9: Quivers whose mutable parts are of size 4

Lemma 4.3.6. *If $a > b > c > 0$, $b > d > 0$, there exists no green permissible sequence turning either Q_{tri}^{source} or Q_{tri}^{sink} all-red.*

Proof. We make explicit computations on how the quivers evolve under green permissible mutations. In this proof we typeset the entry $(5, i)$ of the associated extended exchange matrix for $1 \leq i \leq 4$ next to vertex i and colour this number light blue, instead of drawing the frozen vertex 5. For the convenience of the reader, a complete list of associated extended exchange matrices of the quivers in this proof is provided in Appendix B.1.

Let us start by considering $Q_{\text{tri}}^{\text{source}}$ and mutate at 2:



Thus, we see that the only green permissible vertex after mutating at 2 is 4 and after having mutated at the latter, the only green vertex — namely 3 — is not permissible, giving a green dead end.

Let us fix the following notation for the remainder of this proof: for a mutation sequence $\mathbf{i}$, denote by $\mathbf{c_i} = (c_{\mathbf{i},1}\ c_{\mathbf{i},2}\ \cdots\ c_{\mathbf{i},5})$ the single row of the associated C-matrix of $\mu_{\mathbf{i}}(Q_{\text{tri}}^{\text{source}})$.

The part of the oriented exchange graph of $Q_{\text{tri}}^{\text{source}}$ starting with the mutation at 4 is drawn in Figure 4.10.

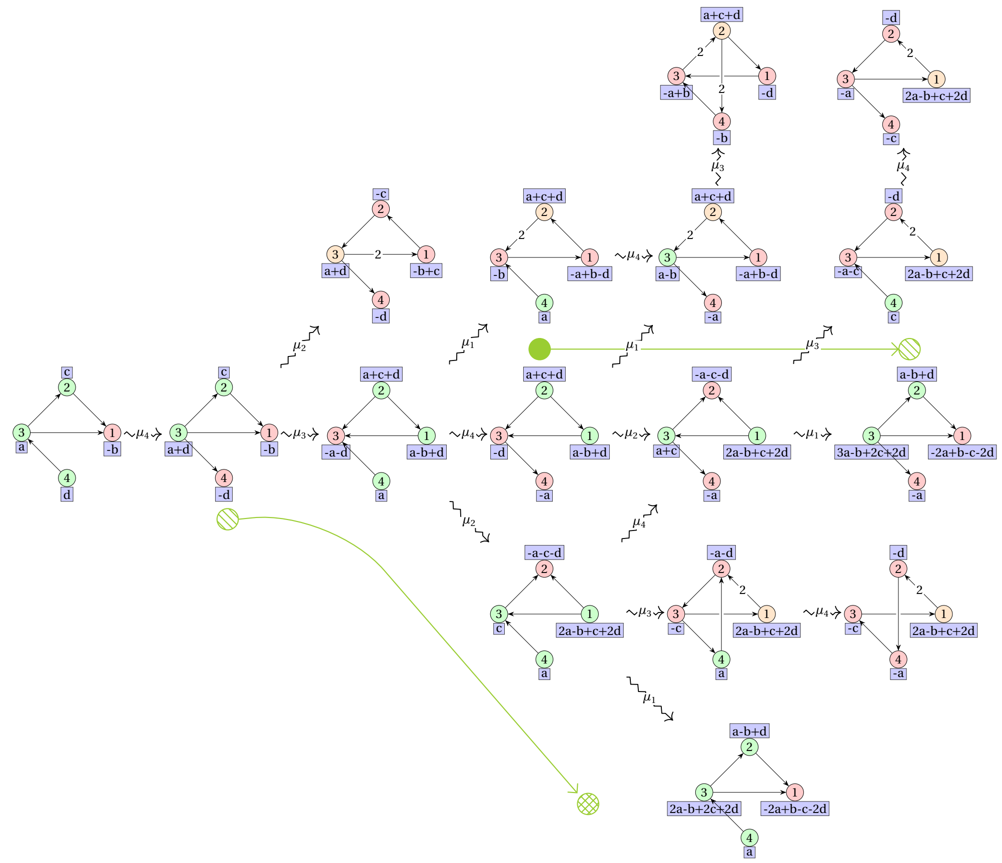


Figure 4.10: Mutations of $Q_{\text{tri}}^{\text{source}}$ starting with 4

We observe that all green permissible sequences reach a green dead end, except for the two sequences $(4,3,2,1)$ and $(4,3,2,4,1)$ resp. $(4,3,4,2,1)$. By assumption, we have

$$2a-b+2c+2d > 2a-b+c+2d > a > a-b+d > 0$$

and we thus obtain

$$\begin{aligned}
&c_{(4,3,2,1),3} > -c_{(4,3,2,1),1} > c_{(4,3,2,1),2} > 0, \; -c_{(4,3,2,1),1} > c_{(4,3,2,1),4} > 0,\\
&c_{(4),3} > -c_{(4),1} > c_{(4),2} > 0, \; -c_{(4),1} > -c_{(4),4} > 0,\\
&c_{(4,3,2,4,1),3} > -c_{(4,3,2,4,1),1} > c_{(4,3,2,4,1),2} > 0, \; -c_{(4,3,2,4,1),1} > -c_{(4,3,2,4,1),4} > 0.
\end{aligned}$$

Hence, the entries of the C-matrix associated to $\mu_{(4,3,2,1)}\left(Q_{\text{tri}}^{\text{source}}\right)$ satisfy the same inequalities as the ones of the C-matrix associated to $Q_{\text{tri}}^{\text{source}}$. Analogously, the entries of the C-matrices associated to $\mu_{(4,3,2,4,1)}\left(Q_{\text{tri}}^{\text{source}}\right)$ and $\mu_4\left(Q_{\text{tri}}^{\text{source}}\right)$ respect identical inequalities. Induction on the length of green permissible sequence applied to $Q_{\text{tri}}^{\text{source}}$ starting with 4 yields that no such green permissible sequence can end in an all-red quiver.

Regarding $Q_{\text{tri}}^{\text{source}}$ we are left with starting the mutation at the vertex 3, see Figure 4.11 for the appropriate part of the oriented exchange graph. Then $\mu_{(3,4)}(Q_{\text{tri}}^{\text{source}})$ is isomorphic to $\mu_{(4,3,4)}(Q_{\text{tri}}^{\text{source}})$, Remark 4.3.3 can be applied to $\mu_{(3,2,1)}(Q_{\text{tri}}^{\text{source}})$ and $\mu_{(3,1,4)}(Q_{\text{tri}}^{\text{source}})$ has no green permissible vertices, nor does $\mu_{(3,2,3)}(Q_{\text{tri}}^{\text{source}})$. Hence this case gets reduced to already known situations discussed above.

Now consider $Q_{\text{tri}}^{\text{sink}}$, for which the first mutation steps of the associated oriented exchange graph are given in Figure 4.12. Following the notation above, denote the single row of the associated C-matrix of $\mu_{\mathbf{i}}\left(Q_{\text{tri}}^{\text{sink}}\right)$ by $\mathbf{c}_{\mathbf{i}}^{\text{sink}} = (c_{\mathbf{i},1}^{\text{sink}}\ c_{\mathbf{i},2}^{\text{sink}}\ \cdots\ c_{\mathbf{i},5}^{\text{sink}})$.

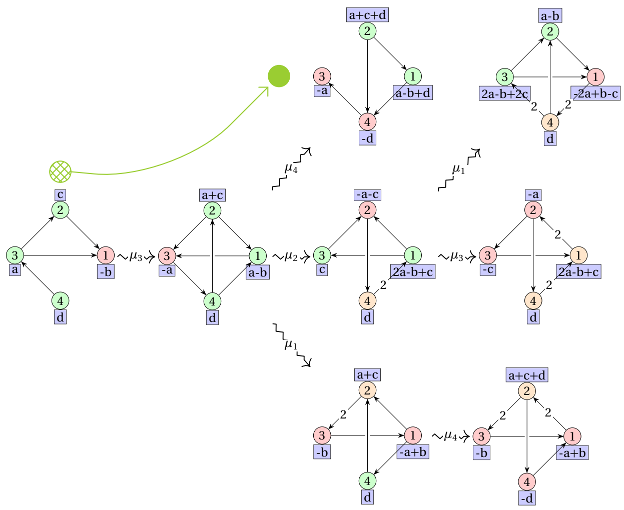


Figure 4.11: Mutations of $Q_{\text{tri}}^{\text{source}}$ starting with 3

We observe that $\mu_{(2,4)}(Q_{\text{tri}}^{\text{sink}}) = \mu_{(4,2)}(Q_{\text{tri}}^{\text{sink}})$ either does not have any green permissible vertex or 1 is the unique green permissible one and mutation at it leads to a green dead end (hence we do not colour the vertex 1 in Figure 4.12). What is more, we see that $\mu_{(3,1,4)}(Q_{\text{tri}}^{\text{sink}})$ is a green dead end as well.
By the initial conditions, we get

$$c_{(4,3,4,2),1} > -c_{(4,3,4,2),2}, \quad -c_{(4,3,4,2),2} > c_{(4,3,4,2),3} > -c_{(4,3,4,2),4} > 0, \tag{4.1}$$

$$c_{(3,4),2}^{\text{sink}} > -c_{(3,4),3}^{\text{sink}}, \quad -c_{(3,4),3}^{\text{sink}} > c_{(3,4),1}^{\text{sink}} > -c_{(3,4),4}^{\text{sink}} > 0. \tag{4.2}$$

Careful inspection of the part of the oriented exchange graph of $Q_{\text{tri}}^{\text{source}}$ which follows the mutation sequence $(4,3,4,2)$ yields that inequalities (4.1) are sufficient for the red-green-colouring in this branch. As (4.2) shows the the entries of the C-matrix associated to $\mu_{(3,4)}(Q_{\text{tri}}^{\text{sink}})$ fulfil the very same inequalities up to relabelling the mutable part, we may conclude there does not exist a green permissible sequence turning it all-red. We are left

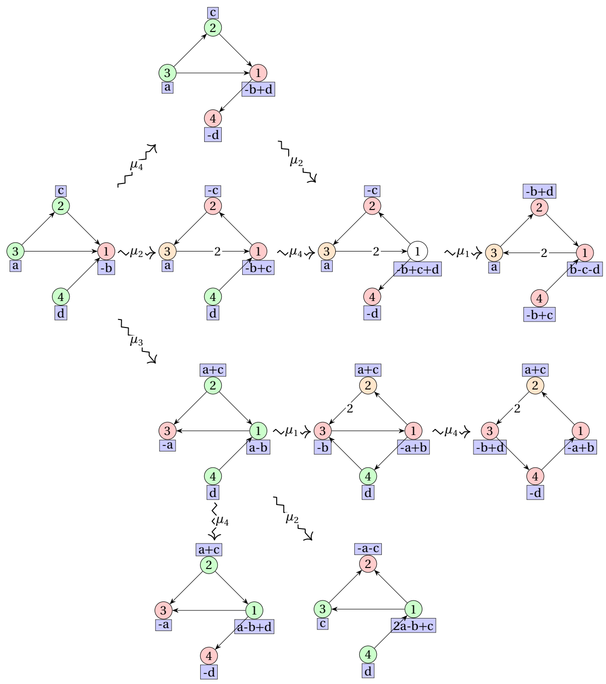


Figure 4.12: Mutations of $Q_{\text{tri}}^{\text{sink}}$

considering $\mu_{(3,2)}(Q_{\text{tri}}^{\text{sink}})$, for which

$$c_{(3,2),1}^{\text{sink}} > -c_{(3,2),2}^{\text{sink}} > c_{(3,2),3}^{\text{sink}} > 0, \quad -c_{(3,2),2}^{\text{sink}} > c_{(3,2),4}^{\text{sink}} > 0.$$

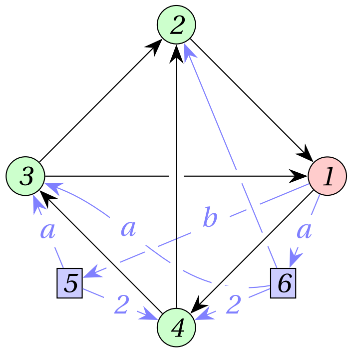
Thus, $\mu_{(3,2)}(Q_{\text{tri}}^{\text{sink}})$ can be reduced to the case $Q_{\text{tri}}^{\text{source}}$ and again induction on the length of green permissible sequences applied to $Q_{\text{tri}}^{\text{source}}$ yields that it does not admit a maximal green sequence either.

In the discussion above, we have covered all possible green permissible directions in the oriented exchange graphs of $Q_{\text{tri}}^{\text{source}}$ and $Q_{\text{tri}}^{\text{sink}}$. It has been shown that none of the branches in either graph can ever lead to an all-red quiver, yielding the claimed result. □

Remark 4.3.7. In Figures 4.10 and 4.11 we highlight a particular iterating path in the oriented exchange graph of $Q_{\text{tri}}^{\text{source}}$ in light green.
This path has length 7 and starts in $\mu_{(4)}\left(Q_{\text{tri}}^{\text{source}}\right)$. After mutating consecutively at vertices $3,2,1$, the resulting quiver is identified with $Q_{\text{tri}}^{\text{source}}$ in Figure 4.11. Using the aforementioned isormorphism between $\mu_{(3,4)}\left(Q_{\text{tri}}^{\text{source}}\right)$ and $\mu_{(4,3,4)}\left(Q_{\text{tri}}^{\text{source}}\right)$, the path continues in Figure 4.10 along the mutation directions $2,1$ and reaches $\mu_{(4,3,4,2,1)}\left(Q_{\text{tri}}^{\text{source}}\right)$. The latter can be identified with $\mu_{(4)}\left(Q_{\text{tri}}^{\text{source}}\right)$ and in this fashion the path repeats itself.
The inductions in the proof of Lemma 4.3.6 assert that none of the quivers along this path admits a maximal green sequence.

One of the quivers which lies on the path described in Remark 4.3.7 is of particular interest to us, albeit in a slightly extended form.

Corollary 4.3.8. *Let Q be given by*



with $a > b > 2$. Then Q does not admit a maximal green sequence.

Proof. Direct computations show that the oriented exchange graph of Q is identical to the one of $\mu_3\left(Q_{\text{tri}}^{\text{source}}\right)$. The claim hence follows from the proof of Lemma 4.3.6. □

The corollary above marks the last combinatorial result which is needed for the proofs of the two main theorems in this section. First, we present the smallest — with respect to the number of vertices — simply-laced quiver whose principal extension does not admit a maximal green sequence.

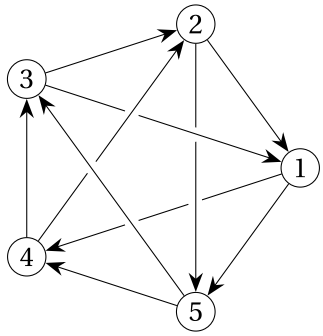


Figure 4.13: Particular orientation Q_{pent} of the pentatope graph

Theorem 4.3.9. *The orientation Q_{pent} of the pentatope graph presented in Figure 4.13 is the only simply-laced quiver with up to 5 vertices whose principal extension does not admit a maximal green sequence.*

Proof. According to [Har57], the number of orientations of connected quivers without loops and 2-cycles with n vertices is 5 for $n = 3$, 34 for $n = 4$ and 535 for $n = 5$. Of these orientations, only 1, 10 and 268 respectively are cyclic[1] to which we may restrict since a sink order would yield a maximal green sequence for the respective principal extension otherwise. Using the software *Sage*, the package *cluster quiver* therein and self-written code, maximal green sequences could be determined for the principal extensions of all 379 quivers except for Q_{pent}. A list of these 378 quivers together with one maximal green sequence for the respective principal extension is provided in Appendix A.

In the subsequent discussion, When displaying certain parts of the oriented exchange graph $\mathbf{EG}\big(Q_{\text{pent}}\big)$, we make the following notational choices: we omit frozen variables for better readability and only display one representative for each isomorphism class. The B- and C-matrices associated to the quivers shown in Figures 4.14 and 4.15 are enlisted in Appendix B.2 so that the reader can easily verify the red-green-colouring of the mutable vertices.
As our aim is to investigate the existence of maximal green sequences, we may disregard those directions in $\mathbf{EG}\big(Q_{\text{pent}}\big)$ which are given by mutations at green yet nonpermissible vertices by Theorem 4.2.4. In other words, leaves in $\mathbf{EG}\big(Q_{\text{pent}}\big)$ which form green dead ends may be dismissed in the search of maximal green sequences.

Due to the symmetry of Q_{pent}, we may restrict to green permissible sequences starting in vertex 1. Let $\tilde{Q}_{\text{pent}}$ be the principal extension of Q_{pent}. The first few layers of $\mathbf{EG}\big(Q_{\text{pent}}\big)$ starting with mutating at vertex 1 are shown in Figure 4.14. We immediately obtain that the green permissible sequences $(1,3,4)$, $(1,3,5,1)$, $(1,4,3)$, $(1,4,1,2)$, $(1,4,1,3,1)$ and $(1,5,1,3,1)$ terminate in green dead ends and can therefore not be extended by green permissible mutations to form a maximal green sequence.

The C-matrices corresponding to $\tilde{Q}_{(1,3,5,4)} := \mu_{(1,3,5,4)}\big(\tilde{Q}_{\text{pent}}\big)$ and $\tilde{Q}_{(1,4,1,3,2)} := \mu_{(1,4,1,3,2)}\big(\tilde{Q}_{\text{pent}}\big)$

[1] See https://oeis.org/A101228.

are

$$\begin{bmatrix} 2 & 0 & 0 & -2 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 1 \\ 2 & 0 & 0 & -1 & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & 0 & 0 & -1 & 1 \\ 1 & -1 & 1 & 0 & 0 \\ 2 & -1 & 0 & 0 & 0 \\ 2 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

We observe that in $\tilde{Q}_{(1,3,5,4)}$ the vertex 3 can only be set green again if a mutation at 2 takes place, due to the third row of the associated C-matrix and the sign-coherence of c-vectors. But Remark 4.3.3 gives that the latter always remains a nonpermissible vertex when green permissible mutations at the vertices of the undirected cycle on the vertices 1, 4 and 5 take place. What is more, this unoriented cycle together with the first and last row of the associated C-matrix satisfies the assumptions of Corollary 4.3.4. Hence the initial green permissible sequence $(1,3,5,4)$ cannot be extended by a green permissible sequence to yield an all-red quiver.

Analogous arguments show that the same holds true for $(1,4,1,3,2)$ where the vertices of the unoriented cycle are 1, 2, 3 and the third and fourth row of the associated C-matrix exhibit the conditions of Corollary 4.3.4.

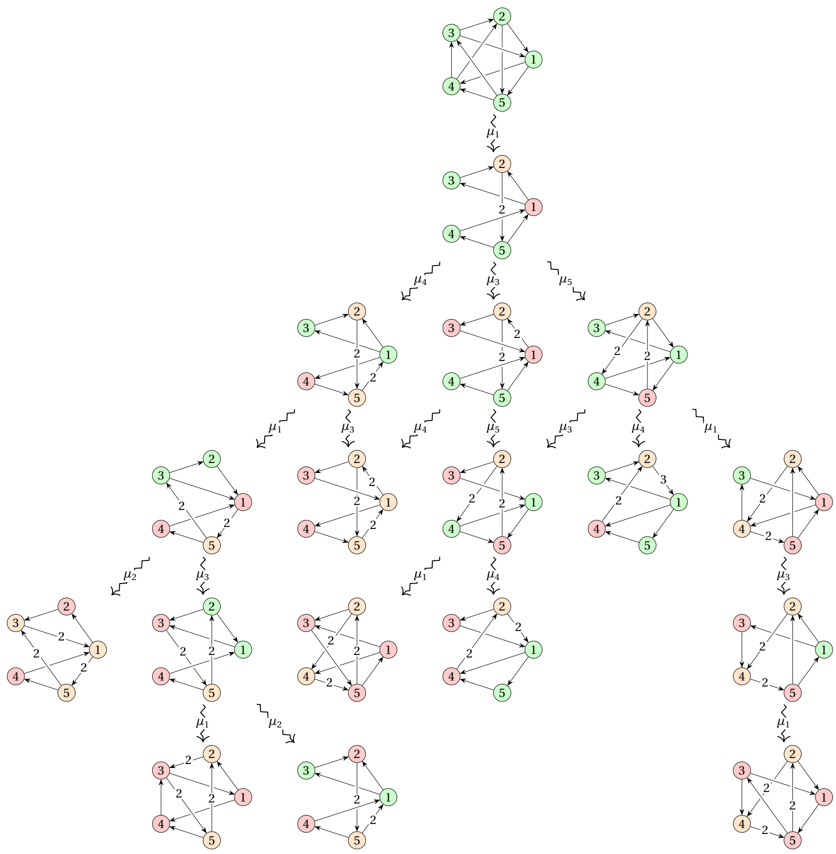


Figure 4.14: Mutations of $\tilde{Q}_{\text{pent}}$ starting with 1

Thus, we are left with the part of $\mathbf{EG}(Q_{\text{pent}})$ which follows the mutation sequence $(1,5,4)$. Note that the quiver $\mu_{(1,5,4,3)}(\tilde{Q}_{\text{pent}})$ is isomorphic to $\mu_{(1,3,5,4)}(\tilde{Q}_{\text{pent}})$ which has already been discussed. In addition, the mutation sequences $(1,3,5,4,5)$ and $(1,5,4,5,3)$, $(1,3,5,4,1)$ and $(1,5,4,1,3,1)$ as well as $(1,3,5,4,1,5)$ and $(1,5,4,1,3,5,1,5)$ yield pairwise isomorphic quivers such that there is nothing to show for these green permissible continuations of $(1,5,4)$ either. We can further identify $\mu_{(1,5,4,1,3,5,1,3)}(\tilde{Q}_{\text{pent}})$ and $\mu_{(1,5,4,1,5,1,3)}(\tilde{Q}_{\text{pent}})$.

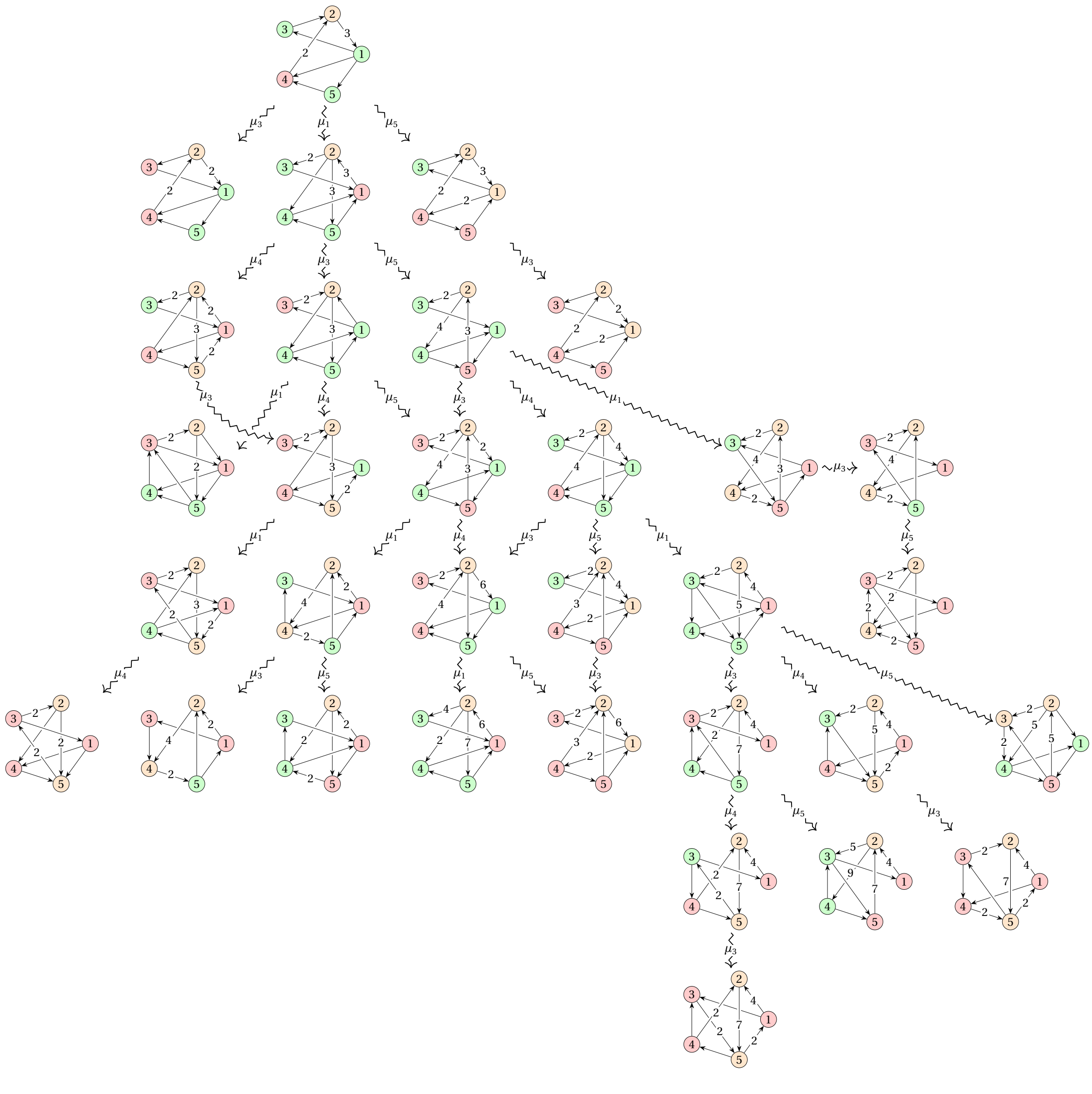


Figure 4.15: Mutations of $\tilde{Q}_{\text{pent}}$ starting after having applied $(1,5,4)$

We obtain that the only remaining leaves of the part of $\mathbf{EG}(Q_{\text{pent}})$ as shown in Figure 4.15

which are not green dead ends are

$$^{1}\tilde{Q} := \mu_{(1,5,4,1,3,5,4,1)}\left(\tilde{Q}_{\text{pent}}\right),\ ^{2}\tilde{Q} := \mu_{(1,5,4,1,5,4,1,3,5)}\left(\tilde{Q}_{\text{pent}}\right),\ ^{3}\tilde{Q} := \mu_{(1,5,4,1,5,4,1,5)}\left(\tilde{Q}_{\text{pent}}\right).$$

The C-matrices associated to $^{1}\tilde{Q}$ and $^{3}\tilde{Q}$ respectively are

$$\begin{bmatrix} -6 & 0 & 4 & 3 & 6 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 \\ -4 & 0 & 3 & 2 & 4 \\ -5 & 0 & 3 & 2 & 6 \end{bmatrix}, \qquad \begin{bmatrix} 0 & 0 & 2 & 5 & -4 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & -3 \\ 1 & 0 & 2 & 4 & -4 \end{bmatrix}.$$

It follows that the mutable part of the full subquiver of $^{1}\tilde{Q}$ given by the vertices $1,3,4$ and 5 along with the last row of the C-matrix satisfies the conditions of Lemma 4.3.6. Additionally, the vertex 2 acts just as the frozen vertex ibid. such that we may conclude that there does not exist a green permissible sequence starting in $^{1}\tilde{Q}$ leading to an all-red quiver.
It is easily verifyable that $^{2}\tilde{Q}$ is isomorphic to $\mu_{(5,3)}\left(^{1}\tilde{Q}\right)$, thus there is nothing left to show in this case.
Regarding $^{3}\tilde{Q}$, we apply Remark 4.3.3 to conclude that the vertices 2 and 3 remain nonpermissible vertices when mutating at the undirected cycle on the vertices 1, 4 and 5. This shows that $^{3}\tilde{Q}$ restricts to the quiver of Proposition 4.3.2 yet again.
We conclude that there exists no maximal green sequence of $\tilde{Q}_{\text{pent}}$ starting with $(1,5,4)$.

This finishes the proof as all possible green permissible directions in the oriented exchange graph $\mathbf{EG}\left(Q_{\text{pent}}\right)$ have been shown to not admit a maximal green sequence. □

Remark 4.3.10. In [BDP14, Example 8.2] a closely related quiver is studied, namely the one which is obtained from replacing all single arrows on the inside of the oriented cycle of Q_{pent} by double arrows. This particular quiver is commonly referred to as the *McKay quiver*. Using involved representation theoric arguments developed in [**DeThanhofferdeVolcsey2013**], it is shown that the principal extension of the McKay quiver does not admit a maximal green sequence. This is not surprising in hindsight of Theorem 4.2.4 and we may regard this example as yet another indication of the strength of restricting to green permissible vertices as induced by the results of [Br17].

Now that we have established the *smallest* — with respect to the number of vertices — simply-laced quiver whose principal extension has no maximal green sequence, we can apply the same techniques to provide a new infinite family of quivers with the same property.

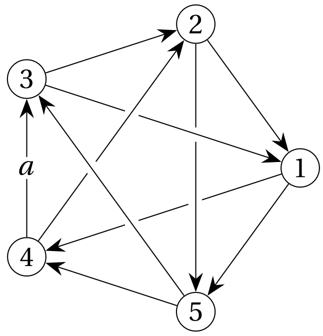


Figure 4.16: The quiver Q^a_{pent}, obtained from Q_{pent} by introducing one multiedge

Theorem 4.3.11. *For an arbitrary integer $a > 0$, the principal extension of the quiver in Figure 4.16 does not admit a maximal green sequence.*

Proof. Let Q^a_{pent} denote the quiver in Figure 4.16 and $\tilde{Q}^a_{\text{pent}}$ be its principal extension.

Here, we restrict to the case $a \geq 5$. The cases $1 < a < 5$ can be proven analogously applying the same techniques to the respective oriented exchange graph $\mathbf{EG}\left(Q^a_{\text{pent}}\right)$, which for a thus chosen parameter a exhibits more branches than those we discuss here. For $a = 1$ the claim is proven in Theorem 4.3.9.

Since $a > 1$ the symmetry of Q_{pent} is broken and we do indeed need to investigate all possible initial directions of mutation. We follow the same notational conventions as in the previous proof and provide in Appendix B.3 the associated B- and C-matrices to all quivers mutation-equivalent to $\tilde{Q}^a_{\text{pent}}$ which are involved in the following discussion.

Let us first consider the branch of the oriented exchange graph following the mutation $\mu_1\left(\tilde{Q}^a_{\text{pent}}\right)$, shown in Figure 4.17.

We observe that all branches of this part of $\mathbf{EG}\left(Q^a_{\text{pent}}\right)$ terminate in leaves which are green dead ends, except for $\mu_{(1,3,5,2,4)}\left(\tilde{Q}^a_{\text{pent}}\right)$. The associated C-matrix is

$$\begin{bmatrix} 2 & 0 & 0 & -2 & 1 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 1 \\ 2 & 0 & 0 & -1 & 0 \end{bmatrix}.$$

The second row of this C-matrix implies that the vertex 2 in $\mu_{(1,3,5,2,4)}\left(\tilde{Q}^a_{\text{pent}}\right)$ can only be set green if mutation at 3 occurs. But according to Remark 4.3.3 the vertex 3 remains a nonpermissible vertex under green permissible mutations at vertices of the undirected cycle on $1, 4, 5$. Applying Corollary 4.3.4 yields the non-existence of a green permissible sequence following $(1, 3, 5, 2, 4)$ which turns $\tilde{Q}^a_{\text{pent}}$ all-red.

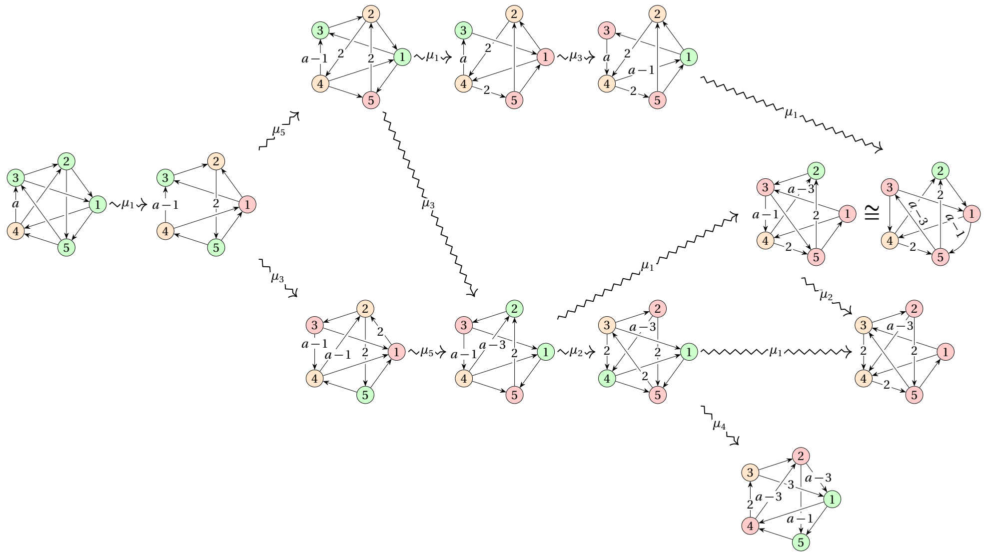


Figure 4.17: Mutations of $\tilde{Q}^a_{\text{pent}}$ after having applied (1)

An analogous argument for the undirected cycle on the vertices $1, 2, 5$ in $\mu_{(5,3,5,2,1,5)}\left(\tilde{Q}^a_{\text{pent}}\right)$ shows that there exists no maximal green sequence starting in mutation direction 5 either. The associated branch of $\mathbf{EG}\left(Q^a_{\text{pent}}\right)$ is displayed in Figure 4.18.

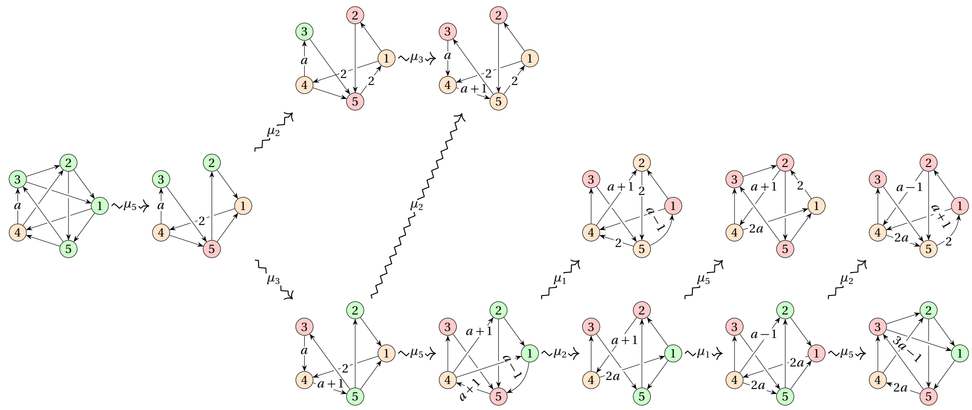


Figure 4.18: Mutations of $\tilde{Q}^a_{\text{pent}}$ after having applied (5)

The second to last vertex of $\tilde{Q}^a_{\text{pent}}$ we have to investigate is 2. The first mutations of this part of $\mathbf{EG}\left(Q^a_{\text{pent}}\right)$ are shown in Figure 4.19.
We observe that we only need to consider $\mu_{(2,1,5,2,1,5,2)}\left(\tilde{Q}^a_{\text{pent}}\right)$ as all other leaves are again green dead ends. The vertex 3 in $\mu_{(2,1,5,2,1,5,2)}\left(\tilde{Q}^a_{\text{pent}}\right)$ plays the same rôle as the additional vertex v in Remark 4.3.3. This argument not only yields that 3 remains nonpermissible after consecutive mutations along the unoriented cycle on the vertices 1, 2 and 5, but it also shows

that the same applies to vertex 4. The C-matrix associated to $\mu_{(2,1,5,2,1,5,2)}\left(\tilde{Q}^a_{\text{pent}}\right)$ is given by

$$\begin{bmatrix} 4 & -3 & 0 & 2 & 0 \\ 4 & -4 & 0 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 3 & -3 & 0 & 1 & 1 \end{bmatrix},$$

whose first row shows that Proposition 4.3.2 can be applied to the aforementioned unoriented cycle. Thus, there is nothing left to show for this branch of $\mathbf{EG}\left(Q^a_{\text{pent}}\right)$.

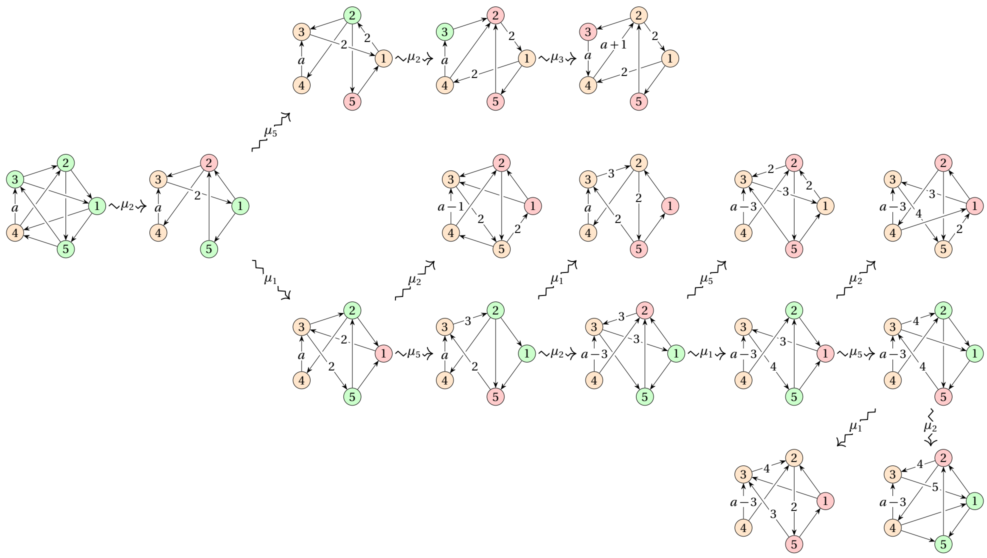


Figure 4.19: Mutations of Q^a_{pent} after having applied (2)

Before we present the part of $\mathbf{EG}\left(Q^a_{\text{pent}}\right)$ after initial mutation at 3, it is easy to verify the following equalities and isomorphisms to already known cases:

$$\begin{array}{ccccc}
\mu_{(3,1,3)}\left(\tilde{Q}^a_{\text{pent}}\right) & \cong & \mu_{(1,3)}\left(\tilde{Q}^a_{\text{pent}}\right), & & \\
\mu_{(3,5)}\left(\tilde{Q}^a_{\text{pent}}\right) & \cong & \mu_{(5,3,5)}\left(\tilde{Q}^a_{\text{pent}}\right), & & \\
\mu_{(3,2,5)}\left(\tilde{Q}^a_{\text{pent}}\right) & = & \mu_{(3,5,2)}\left(\tilde{Q}^a_{\text{pent}}\right) & \cong & \mu_{(5,3,5,2)}\left(\tilde{Q}^a_{\text{pent}}\right), \\
\mu_{(3,1,5)}\left(\tilde{Q}^a_{\text{pent}}\right) & = & \mu_{(3,5,1)}\left(\tilde{Q}^a_{\text{pent}}\right) & \cong & \mu_{(5,3,5,1)}\left(\tilde{Q}^a_{\text{pent}}\right), \\
\mu_{(3,2,3,5,3)}\left(\tilde{Q}^a_{\text{pent}}\right) & \cong & \mu_{(3,5,2,3)}\left(\tilde{Q}^a_{\text{pent}}\right) & \cong & \mu_{(5,3,5,2,3)}\left(\tilde{Q}^a_{\text{pent}}\right), \\
\mu_{(3,2,1,5)}\left(\tilde{Q}^a_{\text{pent}}\right) & = & \mu_{(3,2,5,1)}\left(\tilde{Q}^a_{\text{pent}}\right) & \cong & \mu_{(5,3,5,2,1)}\left(\tilde{Q}^a_{\text{pent}}\right), \\
\mu_{(3,2,1,2,5)}\left(\tilde{Q}^a_{\text{pent}}\right) & = & \mu_{(3,5,2,1,2)}\left(\tilde{Q}^a_{\text{pent}}\right) & \cong & \mu_{(5,3,5,2,1,2)}\left(\tilde{Q}^a_{\text{pent}}\right), \\
\mu_{(3,2,1,3,5,3)}\left(\tilde{Q}^a_{\text{pent}}\right) & \cong & \mu_{(3,5,2,1,3)}\left(\tilde{Q}^a_{\text{pent}}\right) & \cong & \mu_{(5,3,5,2,1,3)}\left(\tilde{Q}^a_{\text{pent}}\right), \\
\mu_{(3,2,1,3,2,5,3,2)}\left(\tilde{Q}^a_{\text{pent}}\right) & = & \mu_{(3,2,1,5,3,2)}\left(\tilde{Q}^a_{\text{pent}}\right) & \cong & \mu_{(5,3,5,2,1,3,2)}\left(\tilde{Q}^a_{\text{pent}}\right),
\end{array}$$

$$\mu_{(3,2,1,3,2,3,5,2)}\left(\tilde{Q}^a_{\text{pent}}\right) = \mu_{(3,2,1,5,3,2,3)}\left(\tilde{Q}^a_{\text{pent}}\right) \cong \mu_{(5,3,5,2,1,3,2,3)}\left(\tilde{Q}^a_{\text{pent}}\right),$$
$$\mu_{(3,2,1,3,2,1,2,5)}\left(\tilde{Q}^a_{\text{pent}}\right) = \mu_{(3,2,1,3,2,1,5,2)}\left(\tilde{Q}^a_{\text{pent}}\right).$$

Thus, in the search for maximal green sequences we may omit the cases above and what remains of the branch of $\mathbf{EG}\left(Q^a_{\text{pent}}\right)$ starting with the mutation at 3 is shown in Figure 4.20.

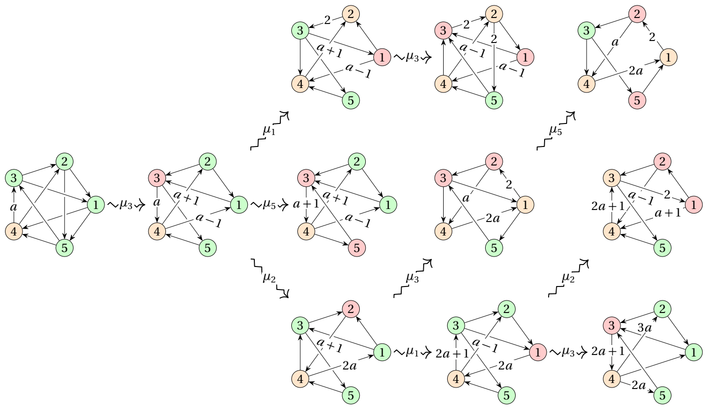


Figure 4.20: Mutations of Q^a_{pent} after having applied (3)

We obtain from Figure 4.20 that $\mu_{(3,2,1,3)}\left(\tilde{Q}^a_{\text{pent}}\right)$ is the only non-green dead end leave in this part of $\mathbf{EG}\left(Q^a_{\text{pent}}\right)$. The continuation of this particular branch of $\mathbf{EG}\left(Q^a_{\text{pent}}\right)$ is presented in Figure 4.21.

As remarked above, the green permissible continutations of the leaves $\mu_{(3,2,1,3,5)}\left(\tilde{Q}^a_{\text{pent}}\right)$, $\mu_{(3,2,1,3,2,1,2)}\left(\tilde{Q}^a_{\text{pent}}\right)$, $\mu_{(3,2,1,3,2,3,5)}\left(\tilde{Q}^a_{\text{pent}}\right)$ and $\mu_{(3,2,1,3,2,5,3)}\left(\tilde{Q}^a_{\text{pent}}\right)$ we have not drawn in Figure 4.21 are already covered by isomorphic cases. Hence, only two more leaves have to be considered. One of these remaining cases is $\mu_{(3,2,1,3,2,1,5)}\left(\tilde{Q}^a_{\text{pent}}\right)$, whose C-matrix is given by

$$\begin{bmatrix} -2 & 0 & 4 & 0 & -1 \\ -3 & 1 & 5 & 0 & -2 \\ -3 & 0 & 6 & 0 & -2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix}.$$

Using the same argument as for $\mu_{(2,1,5,2,1,5,2)}\left(\tilde{Q}^a_{\text{pent}}\right)$ for the unoriented cycle on vertices $1,2,3$ in $\mu_{(3,2,1,3,2,1,5)}\left(\tilde{Q}^a_{\text{pent}}\right)$ in conjunction with the first row of the C-matrix and the two nonpermissible vertices $4,5$ in the same quiver, we obtain that no green permissible sequence continues $(3,2,1,3,2,1,5)$ to form a maximal green sequence.

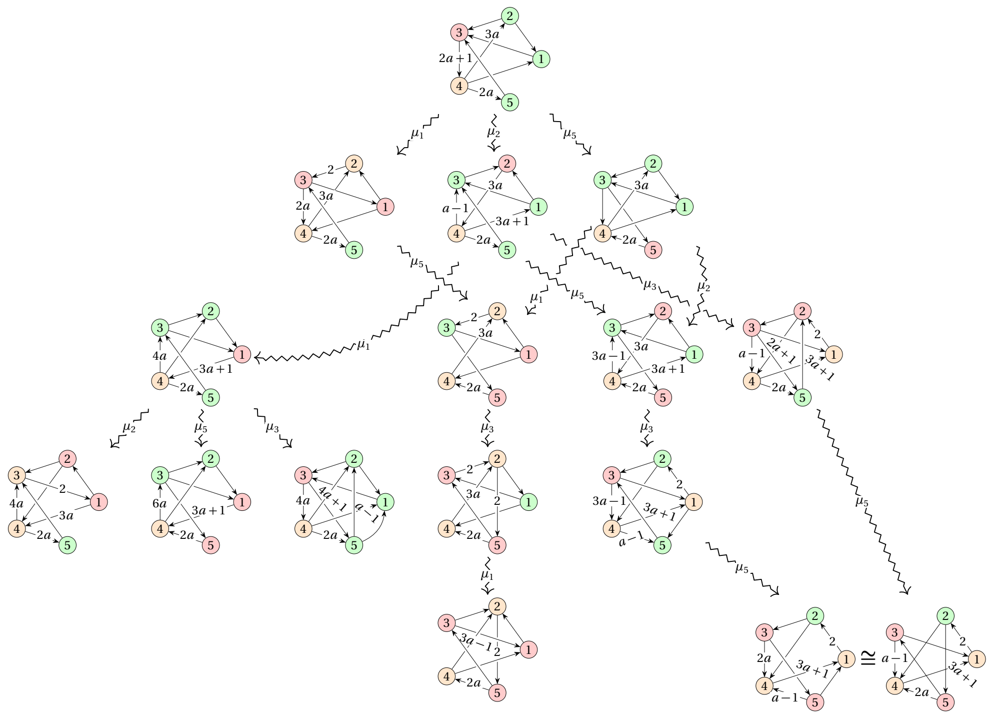


Figure 4.21: Mutations of Q^a_{pent} after having applied $(3,2,1,3)$

The only leave of this branch of $\mathbf{EG}\left(Q^a_{\text{pent}}\right)$ we still have to discuss is $\mu_{(3,2,1,3,2,1,3)}\left(\tilde{Q}^a_{\text{pent}}\right)$, see Figure 4.21. Since $4a+1 > 4a > a-1 > 0$ and $4a > 2a > 0$, the vertex 4 in the associated quiver satisfies the same inequalities with respect to the number of in- and outgoing arrows to the full subquiver on the vertices $1,2,3,5$ as the frozen vertex in $\mu_3\left(Q^{\text{source}}_{\text{tri}}\right)$ in the proof of Lemma 4.3.6. This shows that 4 remains a nonpermissible vertex under greeen permissible mutations of the remaining vertices $1,2,3,5$. The associated C-matrix of $\mu_{(3,2,1,3,2,1,3)}\left(\tilde{Q}^a_{\text{pent}}\right)$ is

$$\begin{bmatrix} 1 & 3 & -3 & 0 & 1 \\ 0 & 4 & -3 & 0 & 2 \\ 1 & 4 & -4 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The second and third row fulfil the condition of Corollary 4.3.8 for the full subquiver on the vertices $1,2,3,5$. We thus conclude that no maximal green sequence lies in the part of $\mathbf{EG}\left(Q^a_{\text{pent}}\right)$ following the mutation sequence $(3,2,1,3,2,1,3)$ and we have exhausted all branches of the oriented exchange graph $\mathbf{EG}\left(Q^a_{\text{pent}}\right)$.

□

Remark 4.3.12. (i) In the proof above, the restriction to green permissible vertices allows us to cover all cases for $a \geq 5$ at the same time. As only single arrows are allowed to leave a green vertex at which a mutation is performed at, the parameter a does not contribute to the associated C-matrices.

(ii) In Figures 4.17–4.21 we can easily localise at which leaves of the branches of $\mathbf{EG}\left(Q^a_{\text{pent}}\right)$ additional mutations have to be considered for $1 < a < 5$. For instance, the vertex 4 in both $\mu_{(1)}\left(\tilde{Q}^a_{\text{pent}}\right)$ and $\mu_{(1,3,5)}\left(\tilde{Q}^a_{\text{pent}}\right)$ is green permissible for $a = 2$ and $1 < a < 5$ respectively.

To summarise our findings, the orientation of the pentatope given in Figure 4.13 is the smallest — with respect to the number of mutable vertices — simply-laced quiver whose principal extension does not admit a maximal green sequence. Replacing one of the arrows on the outer directed 5-cycle by $a > 1$ multiple arrows gives a quiver as shown in Figure 4.16, providing a new infinite family of quivers with the same property.

4.4 Periodicities in the oriented exchange graph

The explicit computations in and the combinatorial preparations of the proofs of Theorems 4.3.9 and 4.3.11 we now embed in a broader context. We review the notion of periodicities in cluster algebras, cf. Section 4.4.1, and ascertain that these existing definitions do neither cover the phenomena of the preceeding section nor admit advances in the search of maximal green sequences. This leads us to develop and define *green periods* and *green permissible periods* in the oriented exchange graph, see Section 4.4.2. After investigating general properties of these new kinds of periods, we study their appearences in the case of the extended Dynkin type $\tilde{A}_{n-1}$ in more detail.

4.4.1 Periodicities in cluster theory

For the review of periodicities in cluster algebras we follow [Nak11].

Definition 4.4.1 [Nak11, Definition 2.4]**.** For a cluster algebra $\mathcal{A} = \mathcal{A}(B, \mathbf{x}, \mathbf{y})$ and any seed $(B', \mathbf{x}', \mathbf{y}')$ in $\mathcal{A}$, let $\sigma : I \to I$ be a bijection of the associated index set, $\mathbf{i} = (i_1, \ldots, i_r)$ a sequence of mutations and denote $(B'', \mathbf{x}'', \mathbf{y}'') = \mu_{\mathbf{i}}((B', \mathbf{x}', \mathbf{y}'))$. Then

- (P_1) $\mathbf{i}$ is called a *σ-period of the exchange matrix B'* if $b''_{\sigma(i),\sigma(j)} = b'_{i,j}$ holds for all $i, j \in I$ and
- (P_2) it is called a *σ-period of the seed* $(B', \mathbf{x}', \mathbf{y}')$ if in addition $x''_{\sigma(i)} = x'_i$ and $y''_{\sigma(i)} = y_i$ for all $i \in I$.

In light of Definition 4.1.5 we can reformulate (P_1)–(P_2) for principally extended, skew-symmetric cluster algebras as follows: Assume $\mathcal{A}$ has a principally extended seed $(B, \mathbf{x}, \mathbf{y})$ and $\tilde{Q}$ is the principally extended quiver associated to B. Let $\tilde{Q}_A$ be mutation equivalent to $\tilde{Q}$ and for a mutation sequence $\mathbf{i} = (i_1, \ldots, i_r)$ denote $\tilde{Q}_\Omega = \mu_{\mathbf{i}}\big(\tilde{Q}_A\big)$. Then

- (MP_1) $\mathbf{i}$ is called a *σ-period of the mutable part of* $\tilde{Q}_A$ if the mutable parts of $\tilde{Q}_A$ and $\tilde{Q}_\Omega$ are isomorphic with respect to σ and
- (MP_2) $\mathbf{i}$ is called a *σ-period of* $\tilde{Q}_A$ if $\tilde{Q}_A$ and $\tilde{Q}_\Omega$ are isomorphic with respect to σ.

Such σ-periods of (extended) exchange matrices and quivers without loops and 2-cycles have appeared in numerous contexts, an overview can be found in [Nak11, Sections 3.1 and 3.2]. A particular family of examples for periodicities of exchange matrices can be found in [FM11], where the authors study quivers which allow single rotations as the bijections in Definition 4.4.1. Let us consider one of these in more detail, namely the orientations of $\tilde{A}_{n-1}$ with one unique source and one sink.

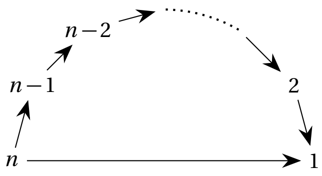


Figure 4.22: Orientation of $\tilde{A}_{n-1}$

Example 4.4.2. For $n > 1$ let Q be the quiver of Figure 4.22. Denote $Q' = \mu_1(Q)$ and $Q'' = \mu_n(Q)$, the resulting quivers of mutations of Q at the unique sink and unique source respectively. In Q' the vertex 2 is the unique sink, 1 is the unique source and we observe that the mutation sequence (1) satisfies Definition 4.4.1 for the clockwise rotation $(n\ 1\ 2\ \cdots\ n-1)$. Similarly, (n) is a period for the anti-clockwise rotation $(2\ 3\ \cdots\ n-1\ n\ 1)$.
Considering the principal extension $\tilde{Q}$ of Q, neither μ_1 nor μ_n is a period of seeds.

Let us further take into account one example of periodic seeds, following [Nak11, Example 3.4 (a)].

Example 4.4.3. For n an odd positive integer, let Q be the quiver from Figure 4.23 of type A_n and $\tilde{Q}$ its principal extension. Then let $\mathbf{i} = (1, 3, \dots, n, 2, 4, \dots, n-1)$ and we can compute that $\mathbf{i}$ repeated $\frac{n+3}{2}$ times is a period for $\tilde{Q}$ with respect to $\sigma = (n\ n-1\ \cdots\ 2\ 1)$. In particular, we observe that repeating $\mathbf{i}$ only $\frac{n+1}{2}$ many times yields a maximal green sequence. Denote

$$\tilde{Q}_{\frac{n+1}{2}} = \mu_{\mathbf{i}^{\frac{n+1}{2}}}(\tilde{Q}) = \underbrace{\mu_{\mathbf{i}} \circ \cdots \circ \mu_{\mathbf{i}}}_{\frac{n+1}{2} \text{ many times}}(\tilde{Q}).$$

Applying $\mathbf{i}$ once more to $\tilde{Q}_{\frac{n+1}{2}}$ mutates at red vertices only. For n even, similar observations can be made, see [Nak11, Example 3.4 (b)].

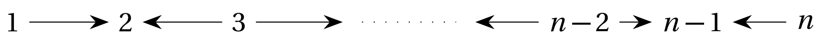


Figure 4.23: Quiver of type A_n with alternating orientation

The observation in the previous example that a period of an extended exchange matrix as given in Definition 4.4.1 mutates both at green and red vertices is no coincidence as the following corollary shows.

Lemma 4.4.4. *Let $\mathcal{A}$ be a skew-symmetric principally extended cluster algebra, i.e. $\mathcal{A} = \mathcal{A}(\left[\begin{smallmatrix} B \\ I_n \end{smallmatrix}\right])$ for some $n \times n$ skew-symmetrizable exchange matrix B. Let $\tilde{B}' = \left[\begin{smallmatrix} B' \\ C' \end{smallmatrix}\right]$ be a seed in $\mathcal{A}$ and assume $\mathbf{i} = (i_1, \dots, i_r)$ is a period for $\tilde{B}'$ with respect to some permutation $\sigma \in S_n$. Under these conditions $\mathbf{i}$ is not a green sequence.*

Proof. Theorem 4.1.8 asserts that the oriented exchange graph $\mathbf{EG}(Q)$ has no oriented cycles, where Q is the quiver associated to the principal part B. If $\mathbf{i}$ was at the same time a green sequence and a σ-period, this would amount to a cycle in $\mathbf{EG}(Q)$ providing a contradiction. $\square$

4.4.2 Green permissible periods

As discussed in the previous subsection, the existing notions of periodicities in cluster algebras have one major drawback when it comes to green sequences: the periodicity of B-matrices is indifferent to the colouring of vertices and the periodicity of seeds mutates both at green and red vertices as shown in Lemma 4.4.4. In order to only accommodate green mutation directions, we develop and definea new kind of periodicity in principally extended cluster algebras which we call *green σ-periods.*

First let us return to the initial example of Section 4.3.

Example 4.4.5. Let $\tilde{Q}$ be as in Example 4.3.1. We have explicitly computed in Proposition 4.3.2 and Corollary 4.3.4 what happens when $\mu_{(3,2,1)}(\tilde{Q})$ is further mutated in green permissible directions. Of particular interest is the mutation sequence $(3,2,1,3,2,1,3,\ldots)$, thus mutating in each step at the respective source of the mutable part. Such mutation sequences yield periods of the B-matrix as in Definition 4.4.1. The first four steps of such green permissible continuations of the sequence $(3,2,1)$ are displayed in Figure 4.24, where we only draw the mutable part and write out the respective C-matrix underneath the quiver.

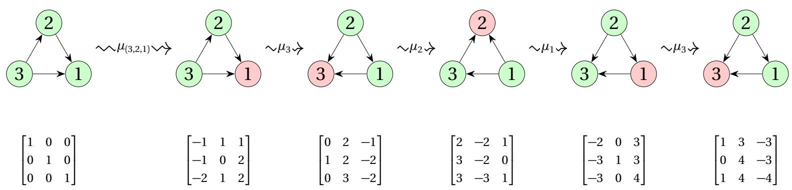


Figure 4.24: Repeated mutation of a quiver of type $\tilde{A}_2$ along the source order

Comparing $\mu_{(3,2,1)}(\tilde{Q})$ with $\mu_{(3,2,1,3,2,1,3)}(\tilde{Q})$, we observe in Figure 4.24 that the C-matrix does not meet the criterion for the σ-periodicity of the extended exchange matrix for any $\sigma \in S_3$. Instead, a different phenomenon becomes apparent in this comparison:while the mutable part of the extended exchange matrix stays fixed with respect to rotating the quiver anti-clockwise by one position, the total amounts of the entries in the C-matrix increase — not necessarily strictly monotonic — with respect to the same permutation whilst keeping their respective signs.

In view of the example above, let us give matrices satisfying the growth behaviour just observed a name.

Definition 4.4.6. Let M and M' be two matrices of dimension $n \times n$ and $\sigma \in S_n$ a permutation. We say that M' is *sign-coherently bigger* than M with respect to σ (or *sign-coherently* *increasing* in short) if

$$\left|(M')_{\sigma(i),\sigma(j)}\right| \geq \left|(M)_{i,j}\right| \text{ and } \operatorname{sgn}(M')_{\sigma(i),\sigma(j)} = \operatorname{sgn}(M)_{i,j} \text{ for all } i,j \in \{1,2,\ldots,n\}. \tag{4.3}$$

We call M' *sign-coherently bigger with respect to its columns* and σ (or *sign-coherently column-increasing* in short) if

$$\left|(M')_{i,\sigma(j)}\right| \geq \left|(M)_{i,j}\right| \text{ and } \operatorname{sgn}(M')_{i,\sigma(j)} = \operatorname{sgn}(M)_{i,j} \text{ for all } i,j \in \{1,2,\ldots,n\}. \tag{4.4}$$

If in addition to (4.3) or (4.4) zero entries of M' and M'' are required to agree with respect to the permuted indices, we say that such matrices are*zero invariant*.

If we want to accommodate the iterating behaviour of quivers as in Proposition 4.3.2 in a new form of σ-periodicity, Example 4.4.5 shows that sign-coherently column-increasing C-matrices are a necessary assumption in any such definition. But careful attention is needed: no infinite green permissible sequence in cluster algebras of finite type exist by Theorem 3.1.7. Yet, there exist periodic B-matrices with sign-coherently column-increasing C-matrices in these cases as the following example shows.

Example 4.4.7. Let $\tilde{Q}$ be the principal extension of type D_4 quiver with the orientation as given in Figure 4.25.

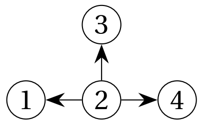


Figure 4.25: Star-shaped D_4 with outward facing arrows

We can compute that the mutable part of $\mu_2(\tilde{Q})$ admits a period as in Definition 4.4.1 given by the green permissible sequence $(1,3,4,2)$ with respect to the permutation (3 2 1 4). The associated C-matrices at the start and end of this period are

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & -1 & 1 & 1 \\ 1 & -2 & 1 & 1 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & 1 & 0 \end{bmatrix}.$$

Since Q is of finite representation type, its oriented exchange graph is a finite digraph and thus every green sequence can be extended to a maximal green sequence; in this case we may subsequently mutate along the green sequence $(1,3,4,2,1,3,4)$ to obtain an all-red quiver.

Having considered the examples of respective type $\tilde{A}_{n-1}$ and D_n above, let us now give a new definition of periodicity in principally extended cluster algebras, which can be regarded to be more strict than (MP_1) but less restrictive than (MP_2).

Definition 4.4.8. Let $\tilde{Q}$ be a principally extended quiver with n mutable vertices and $\tilde{B}_A = \left[\begin{smallmatrix} B_A \\ C_A \end{smallmatrix}\right]$ the extended exchange matrix of a quiver mutation equivalent to $\tilde{Q}$. For a green permissible sequence $\mathbf{i} = (i_1, \ldots, i_r)$ starting in $\tilde{B}_A$, denote by $I(\mathbf{i}) := \{i_1, \ldots, i_r\} \subseteq I$ those indices which contribute to the sequence and by $J(\mathbf{i}) := I \setminus I_{\mathbf{i}}$ those which do not. Then $\mathbf{i}$ is called a *green σ-period* for $\sigma \in S_I$ if the following conditions on $\left[\begin{smallmatrix} B_\Omega \\ C_\Omega \end{smallmatrix}\right] := \mu_{\mathbf{i}}(\left[\begin{smallmatrix} B_A \\ C_A \end{smallmatrix}\right])$ are satisfied:

(GP_1) $\sigma(I(\mathbf{i})) = I(\mathbf{i})$ and $\sigma(J(\mathbf{i})) = J(\mathbf{i})$,

(GP_2) B_Ω is sign-coherently bigger than B_A and these two matrices are zero invariant with respect to σ,

(GP_3) C_Ω sign-coherently bigger with respect to its columns than C_A and these two matrices are zero invariant with respect to σ,

(GP_4) $(B_\Omega)_{\sigma(i),\sigma(j)} = (B_A)_{i,j}$ for all $i, j \in I(\mathbf{i})$ and

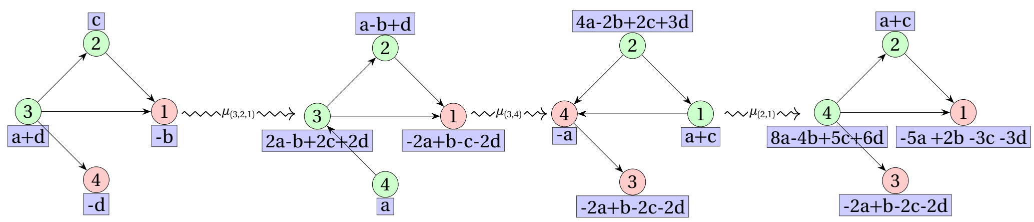
(GP_5) $(C_\Omega)_{i,\sigma(j)} = (C_1)_{i,j}$ for all $i \in I,\ j \in J(\mathbf{i})$.

If $\mathbf{i}$ is in addition permissible, it is called a *green permissible σ-period.*

In the above definition, the submatrices of B and C whose rows are indexed by $I(\mathbf{i})$ and $J(\mathbf{i})$ respectively, satisfy the requirements of Definition 4.4.1. The other entries of the extended exchange matrix $\tilde{B}_A$ are allowed to grow with respect to their respective sign and the permutation σ.

Example 4.4.9. We exhibit that the green permissible sequence in Example 4.4.5 fulfills the conditions of a green permissible period with respect to the permutation (2 3 1). The sequence in Example 4.4.7 on the other hand does not comply with (GP_3).

Example 4.4.10. All quivers in the proof of Lemma 4.3.6 which are proven to recur by induction satisfy Definition 4.4.8. Let us review the green permissible sequence of length 7 discussed in Remark 4.3.7:



Comparing the extended exchange matrices of $\mu_4\left(Q_{\text{tri}}^{\text{source}}\right)$ and $\mu_{(4,3,2,1,3,4,2,1)}\left(Q_{\text{tri}}^{\text{source}}\right)$ gives that this green permissible sequence also satisfies (GP_1)–(GP_5) with respect to the permutation interchanging the vertices 3 and 4.

The definition of a green (permissible) period builds heavily on Theorem 4.2.4 as the follwing example demonstrates.

Example 4.4.11. Let $\tilde{Q}_{\mathrm{Kron}}$ be the principally extended Kronecker quiver of Figure 4.26.

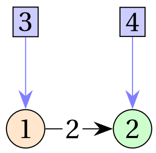
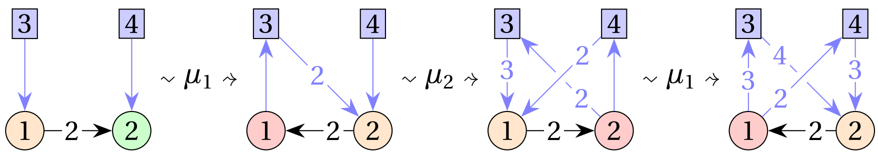


Figure 4.26: Principally extended Kronecker quiver

Then mutating along the green sequence $(1,2,1)$ gives:



We observe that (1) is a green period starting in $\mu_{(1,2)}\left(\tilde{Q}_{\mathrm{Kron}}\right)$ which ends after a single mutation in $\mu_{(1,2,1)}\left(\tilde{Q}_{\mathrm{Kron}}\right)$. In particular, any nonpermissible vertex v for which a neighbouring vertex w and a frozen vertex f exists as in Example 4.2.6 yields a green period of length 1.

We can extend the Kronecker example in the following way.

Lemma 4.4.12. *If* $\mathbf{i}$ *is permissible then the cardinality of* $I(\mathbf{i})$ *is bigger than 2.*

Proof. Assume first $\mathbf{i} = (i_1)$. By (GP_1) and (GP_5) there necessarily exists a red vertex $j \in \tilde{Q}_A$ with $\sigma(i_1) = j$. In particular, j is green after mutation of $\tilde{Q}_A$ at i_1 and one arrow from i_1 to j exists in $\tilde{Q}_A$. Then for any frozen vertex $f \in F$ with $\left(\tilde{B}_A\right)_{f,j} < 0$ it follows by the sign-coherence of c-vectors that

$$\left(\tilde{B}_A\right)_{f,i_1} \geq -\left(\tilde{B}_A\right)_{f,j}.$$

For at least one frozen vertex f_1 this inequality is strict. But in $\tilde{Q}_\Omega$ we have

$$\left(\tilde{B}_\Omega\right)_{\sigma(i_1),f_1} = \left(\tilde{B}_\Omega\right)_{j,f_1} = \left(\tilde{B}_A\right)_{f_1,i_1} + \left(\tilde{B}_\Omega\right)_{f_1,j} \not\geq \left(\tilde{B}_A\right)_{f_1,i_1}$$

and $\left(\tilde{B}_\Omega\right)_{\sigma(i_1),f_1}, \left(\tilde{B}_A\right)_{f_1,i_1}$ are both positive integers. Thus we obtain a contradiction to (GP_3).

Next, assume $\mathbf{i} = (i_1, i_2, i_1, i_2, \ldots)$ for distinct $i_1, i_2 \in I$. By (GP_1) and (GP_3) we have that either $\sigma = \mathrm{id}$ and $i_r = i_2$ or $\sigma(i_1) = i_2$, $\sigma(i_2) = i_1$ and $i_r = i_1$. In both of these cases i_1 is green and i_2 is red in $\tilde{B}_A$. Since i_2 is assumed to be green permissible in $\mu_{i_1}\left(\tilde{B}_A\right)$, there exists a frozen vertex $f \in F$ with

$$\left(\tilde{B}_A\right)_{f,i_1} \geq -\left(\tilde{B}_A\right)_{f,i_2} > 0.$$

This in turn yields that

$$\left(\mu_{(i_1,i_2)}\left(\tilde{B}_A\right)\right)_{f,i_1} = \left(\tilde{B}_A\right)_{f,i_2}$$

and thus both i_1 and i_2 are red in $\mu_{(i_1,i_2)}\left(\tilde{B}_A\right)$, a contradiction to $\mathbf{i}$ being a green permissible sequence. □

For the rest of this section assume that $\tilde{Q}$ is a principally extended quiver with n mutable vertices, $I = \{1,\dots,n\}$ and $\tilde{B}_A = \left[\begin{smallmatrix} B_A \\ C_A \end{smallmatrix}\right]$ is the associated extended exchange matrix of a quiver mutation equivalent to $\tilde{Q}$. Further let $\mathbf{i} = (i_1,\dots,i_r)$ be a green permissible period with respect to some permutation $\sigma \in S_n$ starting in the quiver $\tilde{Q}_A$ associated to $\tilde{B}_A$. Denote $\tilde{Q}_\Omega := \mu_{\mathbf{i}}\left(\tilde{Q}_A\right)$ with associated extended exchange matrix $\tilde{B}_\Omega = \left[\begin{smallmatrix} B_\Omega \\ C_\Omega \end{smallmatrix}\right]$.
After having defined green permissible periods, let us collect further properties of these.

Proposition 4.4.13. *At each mutation step k of $\mathbf{i}$ there exists no arrow from $i_k \to j$ with $j \in J(\mathbf{i})$ and we have*

$$(B_\Omega)_{\sigma(i),\sigma(j)} = (B_A)_{i,j} \tag{4.5}$$

for all $i, j \in J(\mathbf{i})$.

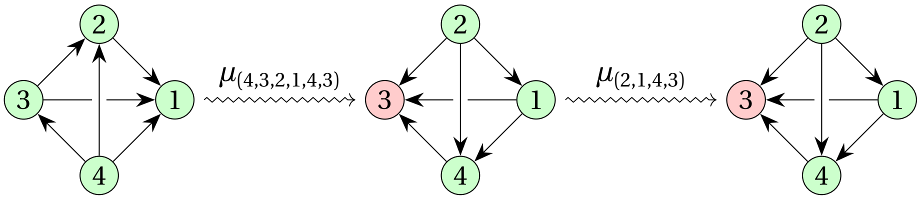
Proof. (GP_4) together with (3.4) immediately yields the desired property. □

Proposition 4.4.14. *The green permissible period $\mathbf{i}$ restricts to the full subquiver of the mutable part of $\tilde{Q}_A$ indexed by $I(\mathbf{i})$ with neighbouring frozen vertices.*

Proof. Follows immediately from Definition 4.4.8. □

The *length* of the green permissible period $\mathbf{i}$ in Example 4.4.5, denoted as $\ell(\mathbf{i})$, is bigger than the number of mutable vertices of the quiver itself. It is natural to ask if the length of such a period is in any form bounded. It is clearly at least as big as the cardinality of $I(\mathbf{i})$ and this lower bound can indeed be met as the following example shows.

Example 4.4.15. Let Q be the quiver on the very left of Figure 4.27 and $\tilde{Q}$ its principal extension. The mutation sequences along with the associated C-matrices of $\tilde{Q}$ are indicated in Figure 4.27 and we observe in particular that $\ell(\mathbf{i}) = I(\mathbf{i}) = 4$.



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 2 & 4 & -2 & 1 \\ 5 & 6 & -4 & 2 \\ 8 & 11 & -6 & 2 \\ 13 & 19 & -11 & 5 \end{bmatrix} \qquad \begin{bmatrix} 24 & 32 & -19 & 8 \\ 40 & 56 & -32 & 13 \\ 69 & 96 & -56 & 24 \\ 120 & 165 & -96 & 40 \end{bmatrix}$$

Figure 4.27: Quiver with green permissible period of minimal length

Thus we have shown the following proposition.

Proposition 4.4.16. *The length* $\mathbf{i}$ *is at least as big as the cardinality of* $I(\mathbf{i})$ *and this bound is sharp.*

As introduced in Section 4.2, k-hemispheres play an important rôle in the proof of Theorem 4.2.4 and thus in our considerations of the oriented pentatope quivers in Section 4.3. Let us thus consider how k-hemispheres are influenced by green permissible σ-periods.

Lemma 4.4.17. *Under a green permissible* σ*-period* $\mathbf{i}$ *no* k*-hemisphere gets changed.*

Proof. Assume there occurs a change of a k-hemispheres at some mutable index $\ell \in I$. By Lemma 4.2.2 changes of k-hemispheres can only occur if mutation at a c-vector which equals $\pm e_k$ takes place. Since $\mathbf{i}$ is a green sequence, there exists an index $1 \le s \le r$ for which the i_s-th column of $\mu_{(i_1,i_2,\dots,i_{s-1})}(C_A)$ equals the positive basis vector e_ℓ.
As no other frozen vertex other than $\ell + n$ is neighbouring i_s in $\mu_{(i_1,i_2,\dots,i_{s-1})}\big(\tilde{Q}_A\big)$ and $\mathbf{i}$ is green, the i_s-th column of C_Ω is $-e_\ell$. By (GP_3) there exists some column j of C_A which equals $-e_\ell$ and $\sigma(i_s) = j$. For the smallest index $1 \le t \le r$ with $i_t = j$ the same reasoning as above gives that the j-th column of $\mu_{(i_1,i_2,\dots,i_{t-1})}\big(\tilde{Q}_A\big)$ is $-e_\ell$, contradicting the greenness of $\mathbf{i}$. □

Another property of green permissible σ-periods concerns certain sinks along these mutation sequences.

Lemma 4.4.18. *For any* $1 \le t \le r$*, the vertex* i_t *in* $\mu_{(i_1,\dots,i_{t-1})}(Q_A)$ *is not a sink in the full subquiver of the mutable part indexed by* $I(\mathbf{i})$*.*

Proof. Assuming the converse, let $1 \le t \le r$ be an index such that i_t is a sink in the mutable part of $\mu_{(i_1,\dots,i_{t-1})}(Q_A)$ indexed by $I(\mathbf{i})$. Due to Theorem 3.1.14 and the mutation of C-matrices in (3.4) we obtain that mutating the associated G-matrix G_{t-1} at i_t negates its i_t-th column. Two cases need to be considered.

Case 1: There exist indices $j_1, j_2 \in I$ such that $(G_{t-1})_{j_1,i_t} \neq 0$ and $(G_{t-1})_{j_1,j_2} \neq 0$. Then mutation of G_{t-1} at i_t induces a contradiction to the sign-coherence of g-vectors.

Case 2: No indices as in Case 1 exist. As G_{t-1} has full rank n, the i_t-th column (resp. i_t-th row) of this matrix necessarily equals $\lambda\mathbf{e}_k$ (resp. $\lambda(\mathbf{e}_k)^T$) for some index $k \in I$ and a non-zero integer $\lambda \in \mathbb{Z}$. As G_{t-1} is an integer matrix with determinant 1, we obtain $\lambda = \pm 1$. By the relation 3.5 between C- and G-matrices, the i_t-th column of $\mu_{(i_1,\ldots,i_{t-1})}(C_A)$ also equals $\pm\mathbf{e}_k$. This in conjunction with Lemma 4.2.2 yields a contradiction to the invariance of k-hemispheres under green permissible periods of Lemma 4.4.17.

□

This marks the end of properties of green permissible periods we are able to prove in full generality. Computations of principally extended quivers for $n = 3,4,5,6$ give us hope that further features of these periods can be established in future research. The most desirable properties of green permissible periods we embrace in the following conjectures.

Conjecture 4.4.19. *The sequence $(i_2, i_3, \ldots, i_r, \sigma(i_1))$ is a green permissible period for $\mu_{i_1}(\tilde{Q}_A)$ with respect to σ.*

As a consequence one would obtain that each green permissible period sparks an infinite green permissible sequence. This in turn would raise the question in which component of the associated cluster category — an object which we have and will not define — these sequences are located in. One interesting example is shown in Theorem 4.4.24 below.

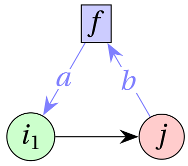
In regards to finding maximal green sequences, our computations also suggest the following conjecture.

Conjecture 4.4.20. *Any maximal green sequence for $\tilde{Q}_\Omega$ gives a maximal green sequence for $\tilde{Q}_A$ and vice versa.*

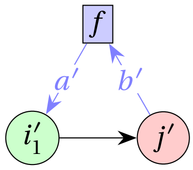
As the quivers at hand are assumed to be finite, the statement of the last conjecture would induce that only finitely many steps in a search algorithm for finding maximal green sequences are necessary. Hence, the question of when a principally extended quiver without loops and 2-cycles admits such a sequence would be determinable by a computer program.

In Appendix C.2 we present an implementation of green permissible sequences in *Sage*, along with a new object class and examples how to use the methods involved. The author is determined to make the code publicly available via https://pub.uni-bielefeld.de, the institutional repository of Bielefeld University.

Remark 4.4.21. Attempts to prove Conjecture 4.4.19 by combinatorial means have so far failed for the following reason: assume in $\tilde{Q}_A$ exists a mutable vertex j and a frozen vertex f such that the full subquiver on the vertices i_1, j, f is given by



with $a > b > 0$ and $j \in I(\mathbf{i})$. By Definition 4.4.8 there exists $\sigma \in S_I$ and integers $a' > a$, $b' > b$ such that the full subquiver of $\tilde{Q}_\Omega$ on the vertices $i_1' = \sigma^{-1}(i_1)$, $j' = \sigma^{-1}(j)$ and f is



But the inequality $a > b > 0$ together with (GP_3) does not yield $a' > b'$. In particular, the sign-coherent increase of the C-matrices associated to $\mu_{i_1}\left(\tilde{Q}_A\right)$ and $\mu_{\sigma^{-1}(i_1)}\left(\tilde{Q}_\Omega\right)$ cannot be shown to hold in this fashion.

4.4.3 The extended Dynkin case $\tilde{A}_{n-1}$

At the end of the last subsection we have conjectured additional properties of green permissible periods. We now restrict to a particular orientation of type $\tilde{A}_{n-1}$ quivers and prove some of these properties for such cases.

Proposition 4.4.22. *Let Q be the quiver as in Figure 4.22 of type $\tilde{A}_{n-1}$ for $n > 2$ and $\tilde{Q}$ its principal extension. Then $\tilde{Q}$ admits green permissible σ-periods for some $\sigma \in S_n$.*

Proof. The proof is done by constructing an intuitive green permissible σ-period involving all mutable vertices.

We first observe that mutating $\tilde{Q}$ along the sequence $\mathbf{i} = (n, n-1, \ldots, 2, 1, n, n-1, \ldots)$ of arbitrary positive length is both green and permissible as mutation always takes place at the respective unique source and affects only neighbouring vertices. Secondly, each mutation yields a period of the mutable part in the sense of Definition 4.4.1 with respect to rotating the vertices once clockwise. What is more, the mutable part of $\mu_{(n,n-1,\ldots,1)}(\tilde{Q})$ is identical to the one of $\tilde{Q}$ and in particular, any sequence $\mathbf{i}$ as above whose length is at least n satisfies (GP_1) for an appropriate permutation σ.

By induction on n and definition of mutation (3.4), the C-matrices of $\mu_{(n,n-1,\ldots,4,3)}(\tilde{Q})$ and

$\mu_{(n,n-1,\dots,2,1)}(\tilde{Q})$ are given by

$$C' := \begin{bmatrix} 1 & 0 & 0 & | & | & & | \\ 0 & 1 & 0 & | & | & & | \\ 0 & 1 & -1 & | & | & & | \\ \vdots & \vdots & \vdots & \mathbf{e}_3 & \mathbf{e}_4 & \cdots & \mathbf{e}_{n-1} \\ 0 & 1 & -1 & | & | & & | \\ 1 & 1 & -1 & | & | & & | \end{bmatrix} \quad \text{and} \quad C'' := \begin{bmatrix} -1 & 1 & | & | & & | & 1 \\ -1 & 0 & | & | & & | & 1 \\ \vdots & \vdots & & & & & \vdots \\ -1 & 0 & \mathbf{e}_3 & \mathbf{e}_4 & \cdots & \mathbf{e}_{n-2} & 1 \\ -1 & 0 & | & | & & | & 2 \\ -2 & 1 & | & | & & | & 2 \end{bmatrix}.$$

The matrix product $C'' \cdot C'$ is

$$\begin{bmatrix} | & 2 & -1 & | & | & & | \\ | & 2 & -2 & | & | & & | \\ & \vdots & \vdots & & & & \\ \mathbf{e}_{n-1} & 2 & -2 & \mathbf{e}_2 & \mathbf{e}_3 & \cdots & \mathbf{e}_{n-4} \\ | & 2 & -2 & | & | & & | \\ | & 3 & -2 & | & | & & | \end{bmatrix}$$

and we choose σ to be the permutation that maps $(1, 2, \dots, n-3, n-2, n-1, n)$ to $(3, 4, \dots, n-1, n, 1)$. Denote $\tilde{Q}'' = \mu_{(n,n-1,\dots,1)}(\tilde{Q})$. By (3.4) the matrix $C'' \cdot C'$ equals the C-matrix of $\mu_{\mathbf{j}}(\tilde{Q}'')$ for the mutation sequence $\mathbf{j} = (n, n-1, \dots, 4, 3)$ with $\ell(\mathbf{j}) = n-2$. Then C'' and $C'' \cdot C'$ satisfy (GP_3) with respect to σ. We further observe that (GP_1) and (GP_5) hold since $J(\mathbf{j}) = \varnothing$. In addition, σ is a permutation which identifies the mutable parts of $\tilde{Q}''$ and $\mu_{\mathbf{j}}(\tilde{Q}'')$, yielding (GP_2) and (GP_4).
Thus, $(n, n-1, \dots, 4, 3)$ is a green permissible σ-period starting in $\tilde{Q}''$ for the permutation σ as chosen above. □

In order to study the mutation sequence of Proposition 4.4.22 further, we use a recent combinatorial model called *periodic trees*, developed in [ITW14], which deals with the root poset of $\tilde{A}_{n-1}$ quivers in a graphical way. As we only utilise this technique shortly and do not make any new contributions to it, we refer to [ITW14] for the necessary definitions. The same applies to the well-known theory of *cluster categories* and *cluster tilting objects* inside such categories. A thorough introduction to these notions can be found in [Ami11] and [Kel10; Kel12].

Theorem 4.4.23 [ITW14, Corollary 2.4.2]**.** *Let Γ be an n-periodic tree and $M = \bigoplus_{i=1}^{n} M_i$ the corresponding cluster tilting object. If we denote by ℓ_i the edge of Γ associated to each summand of M, then*

(1) M_i is regular if and only if ℓ_i does not lie on the periodic infinite path of Γ,

(2) M_i is preprojective if and only if ℓ_i lies on the periodic infinite path of Γ and either Γ has positive slope or it possess zero slope and ℓ_i has positive slope,

(3) *M_i is preinjective if and only if ℓ_i lies on the periodic infinite path of Γ and either Γ has negative slope or it possess zero slope and ℓ_i has negative slope.*

Theorem 4.4.24. *For a mutation sequence $\mathbf{i} = (n, n-1, \ldots, 2, 1, n, n-1, \ldots)$ of finite positive length applied to the principal extension of the orientation of $\tilde{A}_{n-1}$ as given in Figure 4.22, the associated cluster tilting object in the cluster category lies in the preinjective component.*

Proof. Let Q be the quiver as shown in Figure 4.22 and Γ the associated n-periodic tree. The i-th edge in Γ connects the points p_i and p_{i+1} with $p_i > p_{i+1}$. By definition of infinite period trees in [ITW14], all edges in Γ have negative slope and Γ is an infinite path.

For a positive integer $r = an + s$ with $0 \leq s < r$ denote by

$$\begin{aligned}\mathbf{i}_r &= (n, n-1, \ldots, 2, 1)^a \mid (n, n-1, \ldots, n-(s-1)) \\ &= (\underbrace{n, n-1, \ldots, 2, 1, \ldots, n, n-1, \ldots, 2, 1}_{(n, n-1, \ldots, 2, 1) \text{ repeated } a \text{ times}}, n, n-1, \ldots, n-(s-1))\end{aligned}$$

a particular mutation sequence of length r. We claim that $\mu_{i_r}(\Gamma)$ is an infinite path with negative slope of the form as depicted in Figure 4.28, there exists precisely one edge with positive slope and the $(n-s) \mod n$-th edge of $\mu_{i_r}(\Gamma)$ has negative slope.

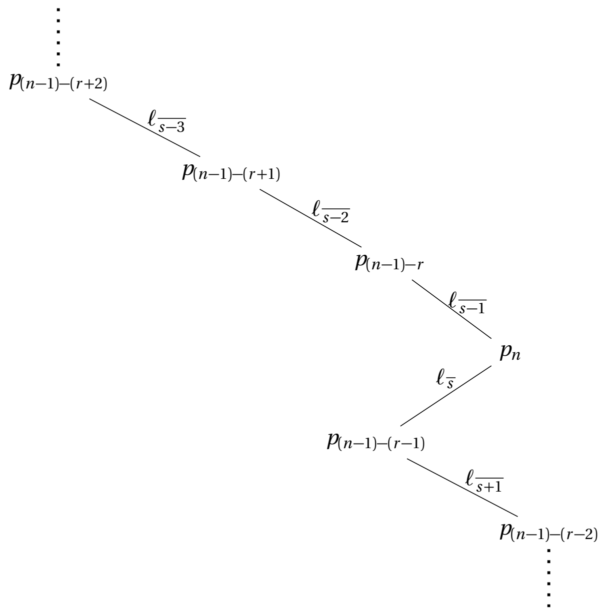


Figure 4.28: Infinite periodic tree obtained by applying the sequence of mutations $\mathbf{i}_r$ to Γ

If $r = s = 1$ then by [ITW14, Definition 1.6.1] the only changes to Γ when mutating at the n-th edge are given by reversing the orientation of $p_{n-1} - p_n$, replacing $p_{n-2} - p_{n-1}$ with

$p_{n-2} - p_n$ and $p_n - p_{n+1}$ with $p_{n-1} - p_{n+1}$. T his mutation in its graphical form is shown in Figure 4.29.

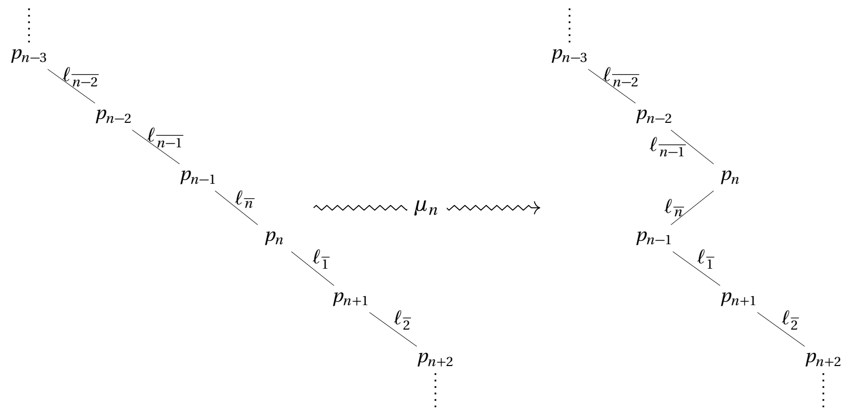


Figure 4.29: Mutation of the infinite periodic tree $\mu_{\mathbf{i}_r}(\Gamma)$

Thus $\mu_n(\Gamma)$ is again an infinite path with negative slope and $p_{n-1} - p_n$ is the only edge with positive slope. What is more, the $(n-1)$-th edge of $\mu_n(\Gamma)$ is by the above argument $p_{n-2} - p_n$.

Now assume the claim holds for some $r > 1$. Then the by assumption and [ITW14, Definition 1.6.1] the n-periodic tree mutation from $\mu_{\mathbf{i}_r}(\Gamma)$ to $\mu_{\mathbf{i}_{r+1}}(\Gamma)$ is given in Figure 4.30.

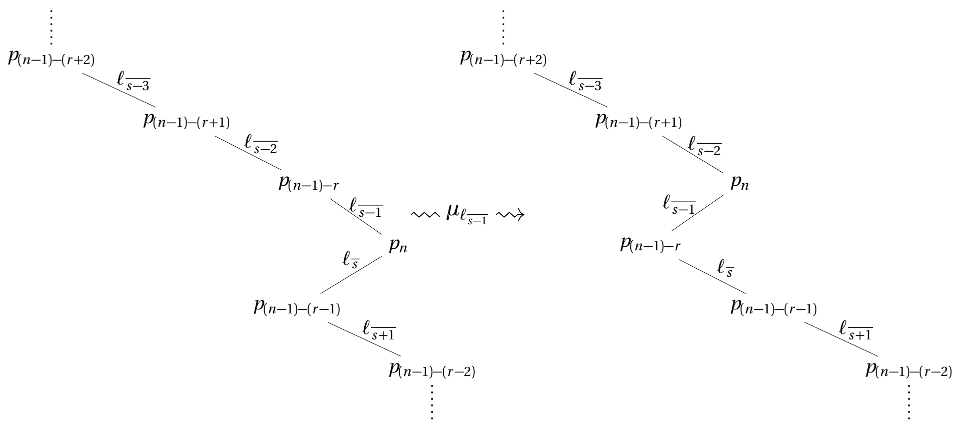
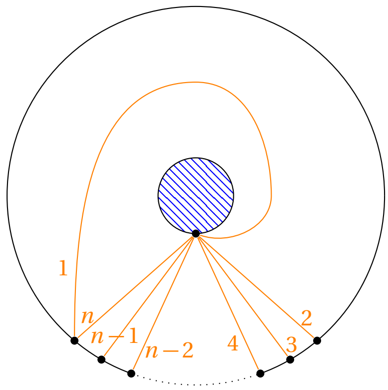


Figure 4.30: Mutation of the infinite periodic tree $\mu_{\mathbf{i}_r}(\Gamma)$ at the edge $\ell_{\overline{s-1}}$

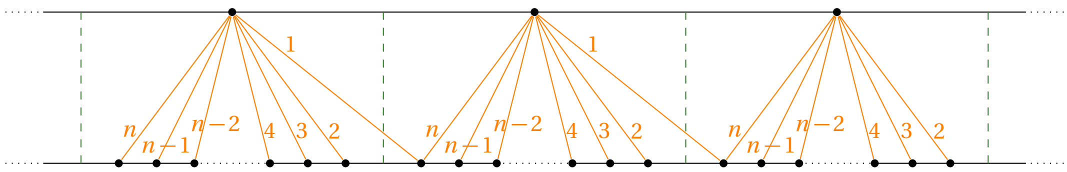
In particular we obain that $\mu_{\mathbf{i}_{r+1}}(\Gamma)$ satisfies all of the properties of the claim and induction yields that it holds for any $r > 1$.

Now that we know the particular shape of any $\mu_{\mathbf{i}_r}(\Gamma)$ we can use Theorem 4.4.23 and deduce that any cluster tilting module along the green permissible sequence $\mathbf{i}_r$ is indeed contained in the preinjective component of the associated cluster category. □

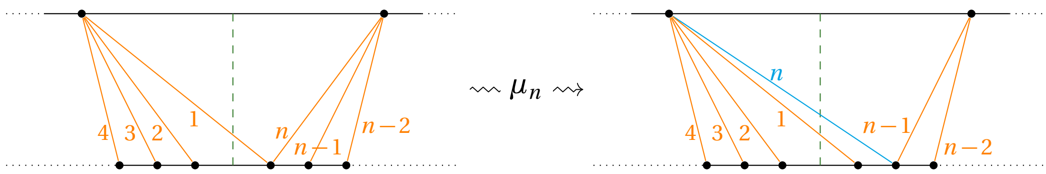
Remark 4.4.25. An alternative proof of Theorem 4.4.24 has been suggested to us by Philipp Lampe: following the well-developed connection between cluster algebras and triangulated surfaces, orientations of a quiver of type $\tilde{A}_{n-1}$ can be realised by triangulations of an annulus with 1 marked point on the inner and $n-1$ marked points on the outer boundary component, cf. [FST08, Section 4]. In this setting, there exists a bijection between the arcs of a triangulation and cluster variables. Following our notation from above, the triangulation associated to the quiver as given in Figure 4.22 can be drawn as follows when labelling the arcs by $1, 2, \dots, n$:



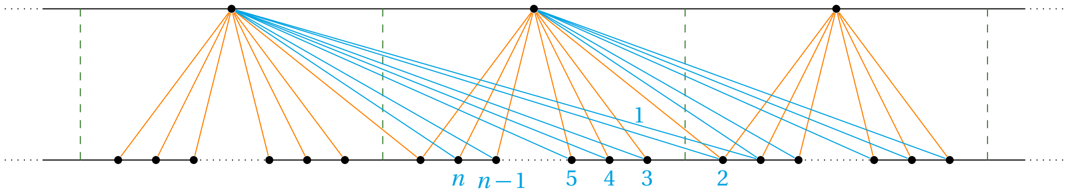
Simplifying the depiction, the universal cover of this particular surface together with the chosen triangulation is given by the following diagram:



In this context, mutation is equivalent to the so-called *flip*. Flipping the triangulation above at the n-th arc amounts to



The dimension vector of the j-th summand of the associated cluster tilting object can be read off the transformed universal cover by counting the number of crossings of the j-th arc with the initial triangulation, see for example [BZ11]. Hence if we overlay the initial triangulation with the one obtain by flipping the arcs $n, n-1, n-2, \ldots, 2$ and 1 consecutively which we colour blue, we have



The associated summands of the cluster tilting complex have dimension vectors

$$\begin{pmatrix}1\\1\\1\\1\\\vdots\\1\\2\end{pmatrix}, \begin{pmatrix}0\\1\\1\\1\\\vdots\\1\\1\end{pmatrix}, \begin{pmatrix}0\\0\\1\\1\\\vdots\\1\\1\end{pmatrix}, \ldots, \begin{pmatrix}0\\0\\0\\0\\\vdots\\0\\1\end{pmatrix}.$$

Using the so-called *defect* of indecomposable representations as defined in [DR76, p. 11] and the results of [Kra08, Proposition 5.2.1], we may conclude that all of the above dimension vectors belong to indecomposable preinjective representations.
In the fashion outlined above, a similar induction as the one used in proving Theorem 4.4.24 provides the same conclusions as previously established.

5

Antichains in posets of quiver representations

Recently, antichains within the root poset of quivers of Dynkin type have been studied in [Rin13] when the partial ordering is given by $x \leq y$ if $y - x$ is a non-negative linear combination of elements in the basis of simple roots. For such posets, the maximal cardinality of any antichain clearly equals the number of vertices of the Dynkin diagram in question and thus antichains of cardinality $n-1$ are considered.

Rather than following the above approach by studying the dimension vectors of indecomposable representations, we investigate the poset which is given by inclusion of indecomposable representations of Dynkin diagrams. This idea is inspired by so-called *monomorphism categories* which are in itself an intense object of study, see the introduction in [Zha11] for a short overview of this topic.

5.1 Poset properties

Let $n \geq 1$ be a natural number and Q be a Dynkin qiver of type A_n. For natural numbers $1 \leq a \leq b \leq n$ use the shorthand notation $[a, b]$ for the representation with vector spaces $V_i = k$ for $a \leq i \leq b$ and $V_i = 0$ elsewhere, and linear maps $V_\alpha = (V_i \to V_j) = \mathrm{id}_k$ for arrows $\alpha\colon i \to j$ with $a \leq i, j \leq b$ and $V_\alpha = 0$ for all others. In the case where $a = b$ we simply write $[a]$ instead of $[a, a]$.

As a consequence of Gabriel's theorem, every indecomposable representation of Q is isomorphic to $[a,b]$ for appropriate choices of $1 \leq a \leq b \leq n$.

Example 5.1.1. Let Q be the quiver $1 \longleftarrow 2 \longrightarrow 3$. The set of isomorphism classes of indecomposable representations contains 6 elements: [1], [2], [3], [1,2], [2,3], [1,3]. These representations are visualized in Figure 5.1. To simplify the notation, whenever we draw an arrow between one-dimensional vector spaces, we assume that the associated map is the identity.

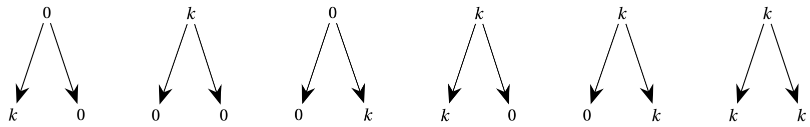


Figure 5.1: Indecomposable representations of alternating A_3

For the rest of this chapter, let $(\mathcal{P}_Q, \leq)$ be the poset with set $\mathcal{P}_Q = \{[a,b] : 1 \leq a \leq b \leq n\}$ and $[a,b] \leq [a',b']$ whenever there exists a monomorphism $[a,b] \hookrightarrow [a',b']$. Equivalently, we have $[a,b] \leq [a',b']$ if and only if $[a,b] \subseteq [a',b']$.

Example 5.1.2. Consider again the quiver as in Example 5.1.1. Then $[1] \leq [1,2]$ and $[1] \leq [1,3]$, but $[1,2] \not\leq [1,3]$. The Hasse diagram of this poset is shown in Figure 5.2.

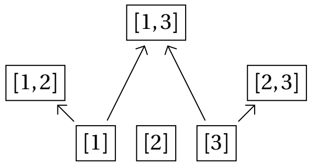


Figure 5.2: Hasse diagram of indecomposable representations of alternating A_3

Remark 5.1.3. The poset $(\mathcal{P}_Q, \leq)$ for Q a Dynkin quiver of type A_n may be graded as in Example 5.1.2, but this is not generally the case as we observe later on in Example 5.2.3. What is more, for $n > 1$ these posets are never bounded and thus never Sperner as the simple representations supported on a source yield isolated vertices in the poset.

5.2 Linear orientation

In this section, consider the linear orientation with unique sink at 1 and unique source at n, see Figure 5.3.

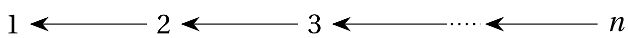


Figure 5.3: Linear orientation of type A_n

Then $[a,b] \leq [a',b']$ if and only if $a = a'$ and $b \leq b'$. Hence the poset $(\mathcal{P}_Q, \leq)$ decomposes into n disjoint chains $C_i = ([i] \leq [i, i+1] \leq [i, i+2] \leq \ldots \leq [i, n])$ for all $1 \leq i \leq n$ and we immediately obtain the following result.

Proposition 5.2.1. *A maximum antichain of* $(\mathcal{P}_Q, \leq)$ *for linearly oriented* A_n *consists of exactly* n *elements.*

5.2.1 Simple zigzag

Let $1 \leq s \leq n$ and consider the orientation of the path of length n with a unique source at position s. For $s = 1$ or $s = n$ the case degenerates to the linear orientation (up to reordering the vertices) of Subsection 5.2. To simplify the notation, denote $\ell = s - 1$ and $r = n - s$ so that the quiver is of the form as shown in Figure 5.4.

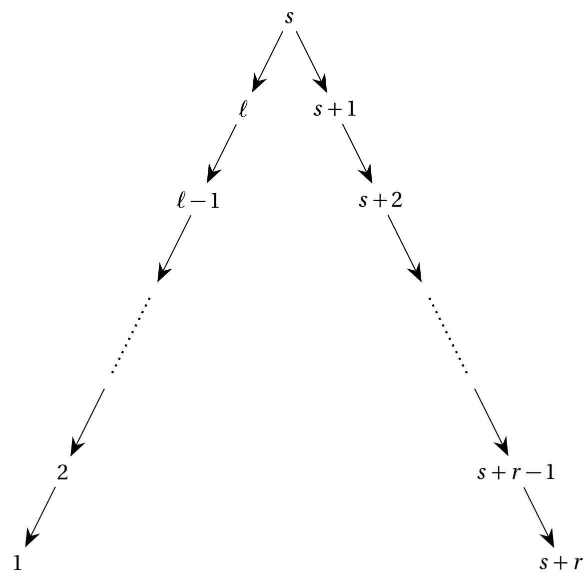


Figure 5.4: Simple zigzag orientation of type A_n

Theorem 5.2.2. *The set* $\mathcal{F} = \{[a, b] \colon a \leq s \leq b\}$ *is a maximum antichain of size* $(\ell+1)(r+1)$ *in* $(\mathcal{P}_Q, \leq)$.

Proof. Let $[a, b]$ and $[a', b']$ be distinct elements from the set $\mathcal{F}$ and suppose there exists a monomorphism $\phi \colon [a, b] \hookrightarrow [a', b']$. This implies $a' \leq a$ and $b \leq b'$. Since these elements are distinct, $a' < a$ or $b < b'$. Without loss of generality we may constrict to $a' < a$. Then the linear map corresponding to $\alpha \colon a \to (a-1)$ is zero in $[a, b]$ whereas in $[a', b']$ it is the identity. But $\mathrm{id}_k \circ \phi_a \neq \phi_{a-1} \circ 0$ yields a contradiction. Hence $\mathcal{F}$ is an antichain.

To show that $\mathcal{F}$ is a maximum antichain, we describe a particular chain decomposition of $(\mathcal{P}_Q, \leq)$ and observe that $\mathcal{F}$ contains precisely one element of each chain.

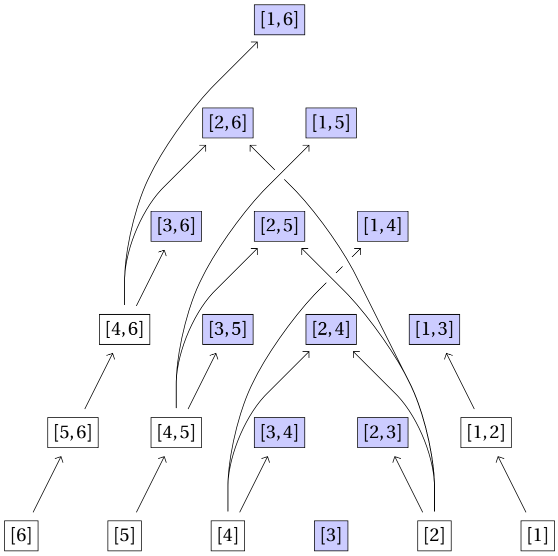


Figure 5.5: Hasse diagram of the monomorphism poset of a quiver of type A_6

For $1 \leq i \leq \ell$ and $1 \leq j \leq r$ let

$$L_i = ([i] \leq [i, i+1] \leq \ldots \leq [i, s]),$$
$$R_j = \big([s+j] \leq [s+j-1, s+j] \leq \ldots \leq [s, s+j]\big).$$

We then obtain a chain decomposition of the entire poset $\mathcal{P}$ as follows:

$$\mathcal{P} = \left(\bigsqcup_{1 \leq i \leq \ell} L_i \right) \sqcup \left(\bigsqcup_{1 \leq j \leq r} R_j \right) \sqcup \left(\bigsqcup_{a < s < b} \{[a, b]\} \right) \sqcup \{[s]\}. \tag{5.1}$$

Every element of $\mathcal{F}$ which is contained in L_i or R_j lies in exactly one chain in (5.1) as the maximal element. The number of elements in an antichain is thus bounded above by the number of chains in the decomposition in (5.1), which coincides with the cardinality of $\mathcal{F}$ by construction. This implies that the decomposition in (5.1) is a Dilworth decomposition and that $\mathcal{F}$ is a maximum antichain. □

Example 5.2.3. Let $n = 6$ and $s = 3$, hence $\ell = 2$ and $r = 3$. The Hasse diagram drawn horizontally and $[a, b] \leq [a', b']$ indicated by arrows $[a, b] \to [a', b']$ is shown in Figure 5.5. The elements of the set $\mathcal{F}$ are highlighted.

5.3 Alternating orientation

For this subsection, let $m \geq 1$ be a natural number and $n = 2m + 1$. Then we consider the quiver Q which arises from the alternating orientation of the path of length n starting and ending with a sink, see Figure 5.6.

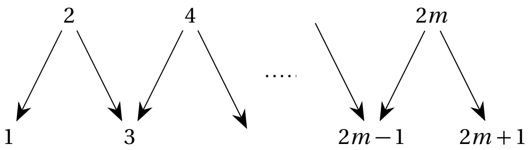


Figure 5.6: Alternating orientation of path of length n with m sources

Theorem 5.3.1. *Let* $\mathcal{F} = \mathcal{F}_{src} \sqcup \mathcal{F}_{1-src} \sqcup \mathcal{F}_{src-n} \sqcup \mathcal{F}_{1-n}$ *be the union of the following four sets*

$$\mathcal{F}_{src} = \{[a,b] : a \le b \text{ sources}\}, \qquad \mathcal{F}_{src-n} = \{[a,n] : a \text{ source}\},$$
$$\mathcal{F}_{1-src} = \{[1,b] : b \text{ source}\}, \qquad \mathcal{F}_{1-n} = \{[1,n]\}.$$

Then $\mathcal{F}$ *is a maximum antichain of cardinality* $\frac{1}{2}m(m+1) + 2m + 1 = \binom{m+2}{2} + m$.

Proof. Using the same arguments as in the proof of Theorem 5.2.2, we see that $\mathcal{F}$ is an antichain. And as before, we provide a chain decomposition of the entire poset to show that it is also maximum.

Let a and b be sources. Then consider the chains in Table 5.1 depending on a choice of one or two sources.

Chain			
Name	Description	Condition	Cardinality
$C_{[a]}$	$([a])$		1
$C_{[a,n]}$	$([a,n])$	$a \neq 2$	1
$C_{[1,a]}$	$([a-1,a] \le [a-3,a] \le \ldots \le [3,a] \le [1,a])$		$a/2$
$C_{[a,b]}$	$([a+1,b-1] \le [a,b-1] \le [a,b])$	$a < b$	3

Table 5.1: Chains depending on a choice of one or two sources

Furthermore, we also consider two chains $C_{[1,n]}$ and $C_{[2,n]}$ not depending on a choice of sources, both of cardinality $m+1$:

$$C_{[1,n]} = ([1] \le [1,3] \le [1,5] \le \ldots \le [1,n]),$$
$$C_{[2,n]} = ([n] \le [n-2,n] \le [n-4,n] \le \ldots \le [5,n] \le [3,n] \le [2,n]).$$

By construction, all of the chains above are pairwise disjoint. Altogether these chains exhaust all elements of the poset since

$$\begin{aligned}\frac{n(n+1)}{2} = \frac{(2m+1)(2m+2)}{2} &= 2m^2+3m+1 \\ &= m\cdot 1+(m-1)\cdot 1+\frac{m(m+1)}{2}+\binom{m}{2}\cdot 3+2\cdot(m+1).\end{aligned}$$

Thus they form a chain decomposition of $(\mathcal{P}_Q, \leq)$ and every element of $\mathcal{F}$ lies in exactly one chain as the maximal element. □

We can apply this construction to a particular subquiver of Q. Let Q' be the full subquiver of Q with the vertex n deleted, i. e. the alternating orientation of the path of length $n-1$ starting with a sink and ending with a source. Then we notate $n' = n-1$ and observe that Q' has indeed $n' = 2m$ vertices and s sources.

Corollary 5.3.2. *Let* $\mathcal{F}'_{src} = \mathcal{F}_{src} \setminus \{[2, n']\}$ *and* $\mathcal{F}'_{1-src} = \mathcal{F}_{1-src} \setminus \{[1, n']\}$. *Then*

$$\mathcal{F}' = \mathcal{F}'_{src} \sqcup \mathcal{F}'_{1-src} \sqcup \{[1, n'-1]\} \sqcup \{[2, n'-1]\} \sqcup \{[3, n']\}$$

is a maximum antichain of $(\mathcal{P}_{Q'}, \leq)$ *of cardinality* $\frac{1}{2}(m+1)(m+2)$.

Proof. The Dilworth decomposition in the proof of Theorem 5.3.1 degenerates to a Dilworth decomposition of $\mathcal{P}_{Q'}$ if one removes all those elements supported at vertex n. □

Example 5.3.3. Let us consider the case for $m = 3$, hence $n = 7$ and $n' = 6$. The Hasse diagrams of the posets $(\mathcal{P}_Q, \leq)$ and $(\mathcal{P}_{Q'}, \leq)$ are shown in Figure 5.7. Those nodes contained in $\mathcal{P}_Q$ but not in $\mathcal{P}_{Q'}$ are shaded gray above the downward diagonal. The elements in $\mathcal{F}$ are highlighted in blue below the downward diagonal and those of $\mathcal{F}'$ in red above the downward diagonal.

Remark 5.3.4. For the only case where the alternating orientation of this section coincides with the simple zigzag of Subsection 5.2.1, the maximum antichains of Theorem 5.2.2 and Theorem 5.3.1 are identical.

Although only few cases of orientations of A_n quivers have been in the above discussion, these consideration should be extended to arbitrary orientations type A_n quivers. Also the Dynkin types D_n and $E_{6,7,8}$ are desirable, yet computational experiments suggest that the combinatorics of general Dynkin cases are much more intricate than what we have just seen.

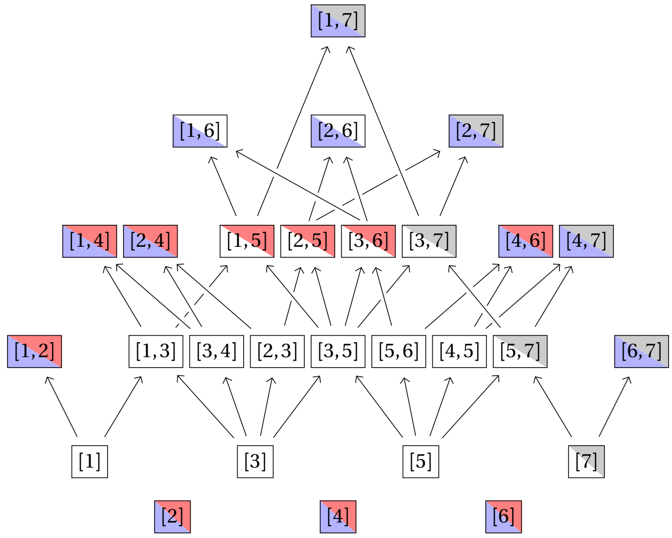


Figure 5.7: Hasse diagrams of alternating orientations of A_6 and A_7

6

Conclusion and Outlook

In this thesis we have discussed a range of questions in cluster algebras and representation theory of quivers. All of these concerne sequences of some sort: the construction of matrices $\mathfrak{M}_{E(i,j)}$ in Theorem 3.2.11 follows a sequential process, maximal green sequences and green permissible periods of Chapter 4 are sequences themselves and finally, in Chapter 5 maximum antichains can be regarded as sequences of objects. Various results for these sequential structures have been shown, in turn raising new questions.

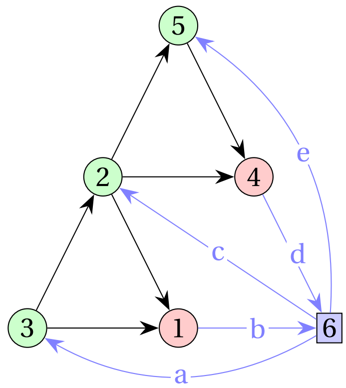
The issue of finding all quantisations of cluster algebras we solve in full completeness, Corollaries 3.2.12 and 3.2.13 providing a summary of what has been accomplished. Yet the structural nature of the constructed matrices constitutes an open problem: when defining quantum seeds, it is necessary to have integer solutions Λ for the compatibility equation (3.7). Both Λ from Theorem 3.2.4 and the enhanced solution matrices $\mathfrak{M}_{E(i,j)}$ from Theorem 3.2.11 are integer matrices. General integer solutions Λ to (3.7) form a semigroup with respect to addition: assume Λ_1 and Λ_2 are skew-symmetric $m \times m$ integer matrices satisfying

$$\tilde{B}^T \Lambda_1 = \begin{bmatrix} D_1' & 0 \end{bmatrix} \quad \text{and} \quad \tilde{B}^T \Lambda_2 = \begin{bmatrix} D_2' & 0 \end{bmatrix},$$

where D_1' and D_2' are diagonal matrices with positive diagonal entries. Then $D_1' + D_2'$ also possesses positive diagonal entries and $\tilde{B}^T(\Lambda_1 + \Lambda_2) = \begin{bmatrix} (D_1' + D_2') & 0 \end{bmatrix}$. However, the constructed matrices do not generate the semigroup of all integer quantisations and it would be interesting to further investigate the generators of this semigroup.

In the discourse of (maximal) green sequences in Chapter 4, we develope a combinatorial machinery to study certain quivers with up to four mutable vertices. These relatively easy considerations enable us to present the smallest simply laced quiver whose principal extension does not admit a maximal green sequence. What is more, the same techniques can be used to also provide a new infinite class of quivers on five vertices with the same property. As well as providing new examples of quivers without maximal green sequences, we embed the combinatorial discussion into the framework of periodicities in the oriented exchange graph. Specialising to quivers of type $\tilde{A}_{n-1}$ in Section 4.4.3, we show that one of these new periods yields an infinite sequence of mutations inside the preinjective component of the associated cluster category. A multitude of questions is imposed by the results established so far, alongside the conjectures stated at the end of Section 4.4.2:

1. Can the quivers of the combinatorial discussion at the start of Section 4.3 be generalised? In other words, is it possible to define a combinatorial construction of quivers with frozen vertices which do not admit maximal green sequences?
   As an example of how such a construction could be started, two copies of the quiver Q_{tri} of Proposition 4.3.2 can be put together in the following way:

   with positive integers $a > b > c > 0$ and $a - c > d > e > 0$. The mutation sequence $(3,2,1,5,4)$ yields a quiver of the same form and it is both green and permissible. If we wanted to show that this quiver does not admit a maximal green sequence, all possible mutations would have to be checked.

2. Do the combinatorial results of Section 4.3 suffice to reprove the statement of [Sev14] that the principal extension of the quiver $\mathbb{X}_7$ of exceptional type does not admit a maximal green sequence?

3. Let Q be a quiver without loops and 2-cycles on n vertices. Further let $\mathbf{i} = (i_1, i_2, \ldots, i_r)$ be a green permissible σ-period for the principal exstension $\tilde{Q}$ of Q. Denote by $\tilde{Q}_A$ the quiver at which $\mathbf{i}$ starts and $\tilde{Q}_\Omega = \mu_{\mathbf{i}}(\tilde{Q}_A)$.

   (a) Do all summands of the cluster tilting object associated to $\mu_{(i_1,\ldots,i_s)}(\tilde{Q}_A)$ for $1 \leq s \leq r-1$ lie in the preinjective component as in Theorem 4.4.24?

(b) Can the starting point $\tilde{Q}_A$ of $\mathbf{i}$ be characterized either combinatorially or via cluster categories and special objects (silting, τ-tilting etc.) therein?

(c) Another open problem in cluster theory is whether the upper cluster algebras as introduced in [BFZ05] coincides with the cluster algebra itself. For cluster algebras of finite type this has been shown to be true ibid. and various authors have extended this result to include further types, see [Mul14] and [CLS15] for instance.
In this context, could the cluster variables associated to either $\tilde{Q}_A$ or $\tilde{Q}_\Omega$ be used to show that these two structures do not coincide when no maximal green sequences exist?

Lastly, for certain orientations of A_n quivers we inspect the poset induced by monomorphisms of indecomposable quiver representations in Chapter 5. Other orientations of the diagram A_n than the ones discussed are less symmetric and thus allow for higher combinatorial complexity. The constructions of Dilworth decompositions do not generalise to arbitrary orientations and computational experiments suggest that the combinatorics of those cases are much more intricate.Nevertheless, it should be feasible to obtain similar results with the same techniques as above for particular orientations of quivers of type D_n. As orientations of Dynkin quivers of type E_6, E_7 and E_8 form a finite family, studying the associated monomorphism posets presents a finite problem which might prove insightful. In addition, it would be interesting to see for which orientation of the diagram A_n the poset $(\mathcal{P}, \leq)$ has largest width, and whether this maximum is obtained by a simple zigzag with the unique source in the middle of the quiver.

Bibliography

[Ami09] Claire Amiot. “Cluster categories for algebras of global dimension 2 and quivers with potential”. In: *Ann. Inst. Fourier (Grenoble)* 59.6 (2009), pp. 2525–2590.

[Ami11] Claire Amiot. “On generalized cluster categories”. In: *Representations of algebras and related topics*. EMS Ser. Congr. Rep. Eur. Math. Soc., Zürich, 2011, pp. 1–53.

[ASS06] Ibrahim Assem, Daniel Simson, and Andrzej Skowroński. *Elements of the representation theory of associative algebras. Vol. 1*. Vol. 65. London Mathematical Society Student Texts. Techniques of representation theory. Cambridge University Press, Cambridge, 2006, pp. x+458.

[Aus74] Maurice Auslander. “Representation theory of Artin algebras. I, II”. In: *Comm. Algebra* 1 (1974), 177–268; ibid. 1 (1974), 269–310.

[BDP14] Thomas Brüstle, Grégoire Dupont, and Matthieu Pérotin. “On maximal green sequences”. In: *Int. Math. Res. Not. IMRN* 16 (2014), pp. 4547–4586.

[BFZ05] Arkady Berenstein, Sergey Fomin, and Andrei Zelevinsky. “Cluster algebras. III. Upper bounds and double Bruhat cells”. In: *Duke Math. J.* 126.1 (2005), pp. 1–52.

[BMR07] Aslak Bakke Buan, Robert J. Marsh, and Idun Reiten. “Cluster-tilted algebras”. In: *Trans. Amer. Math. Soc.* 359.1 (2007), pp. 323–332.

[BMR08] Aslak Bakke Buan, Robert J. Marsh, and Idun Reiten. “Cluster mutation via quiver representations”. In: *Comment. Math. Helv.* 83.1 (2008), pp. 143–177.

[Bou68] Nicolas Bourbaki. *Elements of mathematics. Theory of sets*. Translated from the French. Hermann, Publishers in Arts and Science, Paris; Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1968, pp. viii+414.

[Br17] Thomas Brüstle, Stephen Hermes, Kiyoshi Igusa, and Gordana Todorov. “Semi-invariant pictures and two conjectures on maximal green sequences”. In: *J. Algebra* 473 (2017), pp. 80–109.

[Bua+06] Aslak Bakke Buan, Robert J. Marsh, Markus Reineke, Idun Reiten, and Gordana Todorov. “Tilting theory and cluster combinatorics”. In: *Adv. Math.* 204.2 (2006), pp. 572–618.

[BY13] Thomas Brüstle and Dong Yang. "Ordered exchange graphs". In: *Advances in representation theory of algebras*. EMS Ser. Congr. Rep. Eur. Math. Soc., Zürich, 2013, pp. 135–193.

[BZ05] Arkady Berenstein and Andrei Zelevinsky. "Quantum cluster algebras". In: *Adv. Math.* 195.2 (2005), pp. 405–455.

[BZ11] Thomas Brüstle and Jie Zhang. "On the cluster category of a marked surface without punctures". In: *Algebra Number Theory* 5.4 (2011), pp. 529–566.

[Cay49] A. Cayley. "Sur les déterminants gauches. (Suite du Mémoire T. XXXII. p. 119)". In: *J. Reine Angew. Math.* 38 (1849), pp. 93–96.

[CC06] Philippe Caldero and Frédéric Chapoton. "Cluster algebras as Hall algebras of quiver representations". In: *Comment. Math. Helv.* 81.3 (2006), pp. 595–616.

[CLS15] Ilke Canakci, Kyungyong Lee, and Ralf Schiffler. "On cluster algebras from unpunctured surfaces with one marked point". In: *Proc. Amer. Math. Soc. Ser. B* 2 (2015), pp. 35–49.

[DR76] Vlastimil Dlab and Claus Michael Ringel. "Indecomposable representations of graphs and algebras". In: *Mem. Amer. Math. Soc.* 6.173 (1976), pp. v+57.

[DWZ10] Harm Derksen, Jerzy Weyman, and Andrei Zelevinsky. "Quivers with potentials and their representations II: applications to cluster algebras". In: *J. Amer. Math. Soc.* 23.3 (2010), pp. 749–790.

[FM11] Allan P. Fordy and Robert J. Marsh. "Cluster mutation-periodic quivers and associated Laurent sequences". In: *J. Algebraic Combin.* 34.1 (2011), pp. 19–66.

[FST08] Sergey Fomin, Michael Shapiro, and Dylan Thurston. "Cluster algebras and triangulated surfaces. I. Cluster complexes". In: *Acta Math.* 201.1 (2008), pp. 83–146.

[FZ02] Sergey Fomin and Andrei Zelevinsky. "Cluster algebras. I. Foundations". In: *J. Amer. Math. Soc.* 15.2 (2002), pp. 497–529.

[FZ03] Sergey Fomin and Andrei Zelevinsky. "Cluster algebras. II. Finite type classification". In: *Invent. Math.* 154.1 (2003), pp. 63–121.

[FZ07] Sergey Fomin and Andrei Zelevinsky. "Cluster algebras. IV. Coefficients". In: *Compos. Math.* 143.1 (2007), pp. 112–164.

[Gab72] Peter Gabriel. "Unzerlegbare Darstellungen. I". In: *Manuscripta Math.* 6 (1972), 71–103; correction, ibid. 6 (1972), 309.

[GL14] Florian Gellert and Philipp Lampe. "Quantisation Spaces of Cluster Algebras". In: (2014), pp. 1–11. arXiv: 1402.1094.

[GL16] Florian Gellert and Philipp Lampe. "Maximum antichains in posets of quiver representations". In: (2016), pp. 1–11. arXiv: 1608.03446.

[GLS13] C. Geiß, B. Leclerc, and J. Schröer. "Cluster structures on quantum coordinate rings". In: *Selecta Math. (N.S.)* 19.2 (2013), pp. 337–397.

[Gro+14] Mark Gross, Paul Hacking, Sean Keel, and Maxim Kontsevich. “Canonical bases for cluster algebras”. In: (2014), pp. 1–136. arXiv: 1411.1394.

[GSV03] Michael Gekhtman, Michael Shapiro, and Alek Vainshtein. “Cluster algebras and Poisson geometry”. In: *Mosc. Math. J.* 3.3 (2003). {Dedicated to Vladimir Igorevich Arnold on the occasion of his 65th birthday}, pp. 899–934, 1199.

[GY14] Kenneth R. Goodearl and Milen T. Yakimov. “Quantum cluster algebras and quantum nilpotent algebras”. In: *Proc. Natl. Acad. Sci. USA* 111.27 (2014), pp. 9696–9703.

[Har57] Frank Harary. “The number of oriented graphs”. In: *Michigan Math. J.* 4 (1957), pp. 221–224.

[HL10] David Hernandez and Bernard Leclerc. “Cluster algebras and quantum affine algebras”. In: *Duke Math. J.* 154.2 (2010), pp. 265–341.

[Hum72] James E. Humphreys. *Introduction to Lie algebras and representation theory.* Graduate Texts in Mathematics, Vol. 9. Springer-Verlag, New York-Berlin, 1972, pp. xii+169.

[ITW14] Kiyoshi Igusa, Gordana Todorov, and Jerzy Weyman. “Periodic trees and semi-invariants”. In: (2014), pp. 1–33. arXiv: 1407.0619.

[Kel05] Bernhard Keller. “On triangulated orbit categories”. In: *Doc. Math.* 10 (2005), pp. 551–581.

[Kel10] Bernhard Keller. “Cluster algebras, quiver representations and triangulated categories”. In: *Triangulated categories.* Vol. 375. London Math. Soc. Lecture Note Ser. Cambridge Univ. Press, Cambridge, 2010, pp. 76–160.

[Kel11] Bernhard Keller. “On cluster theory and quantum dilogarithm identities”. In: *Representations of algebras and related topics.* EMS Ser. Congr. Rep. Eur. Math. Soc., Zürich, 2011, pp. 85–116.

[Kel12] Bernhard Keller. “Cluster algebras and derived categories”. In: *Derived categories in algebraic geometry.* EMS Ser. Congr. Rep. Eur. Math. Soc., Zürich, 2012, pp. 123–183.

[Knu96] Donald E. Knuth. “Overlapping Pfaffians”. In: *Electron. J. Combin.* 3.2 (1996). The Foata Festschrift, Research Paper 5, approx. 13 pp.

[KQ15] Alastair King and Yu Qiu. “Exchange graphs and Ext quivers”. In: *Adv. Math.* 285 (2015), pp. 1106–1154.

[Kra08] Henning Krause. “Representations of quivers via reflection functors”. In: (2008), pp. 1–42. arXiv: 0804.1428.

[KV02] Bernhard Korte and Jens Vygen. *Combinatorial optimization.* Second. Vol. 21. Algorithms and Combinatorics. Theory and algorithms. Springer-Verlag, Berlin, 2002, pp. xiv+530.

[KY14] Steffen König and Dong Yang. "Silting objects, simple-minded collections, t-structures and co-t-structures for finite-dimensional algebras". In: *Doc. Math.* 19 (2014), pp. 403–438.

[Lam11] Philipp Lampe. "A quantum cluster algebra of Kronecker type and the dual canonical basis". In: *Int. Math. Res. Not. IMRN* 13 (2011), pp. 2970–3005.

[Lam14] Philipp Lampe. "Quantum cluster algebras of type A and the dual canonical basis". In: *Proc. Lond. Math. Soc. (3)* 108.1 (2014), pp. 1–43.

[Lec04] Bernard Leclerc. "Dual canonical bases, quantum shuffles and q-characters". In: *Math. Z.* 246.4 (2004), pp. 691–732.

[Lem12] Joel Lemay. "Valued Graphs and the Representation Theory of Lie Algebras". In: *Axioms* 1.3 (2012), pp. 111–148.

[Lus93] George Lusztig. *Introduction to quantum groups.* Vol. 110. Progress in Mathematics. Birkhäuser Boston, Inc., Boston, MA, 1993, pp. xii+341.

[Mil16] Matthew R. Mills. "Maximal green sequences for quivers of finite mutation type". In: (2016), pp. 1–27. arXiv: 1606.03799.

[Mul14] Greg Muller. "A=U for locally acyclic cluster algebras". In: *SIGMA Symmetry Integrability Geom. Methods Appl.* 10 (2014), Paper 094, 8.

[Mul16] Greg Muller. "The existence of a maximal green sequence is not invariant under quiver mutation". In: *Electron. J. Combin.* 23.2 (2016), Paper 2.47, 23.

[Nak11] Tomoki Nakanishi. "Periodicities in cluster algebras and dilogarithm identities". In: *Representations of algebras and related topics.* EMS Ser. Congr. Rep. Eur. Math. Soc., Zürich, 2011, pp. 407–443.

[NZ12] Tomoki Nakanishi and Andrei Zelevinsky. "On tropical dualities in cluster algebras". In: *Algebraic groups and quantum groups.* Vol. 565. Contemp. Math. Amer. Math. Soc., Providence, RI, 2012, pp. 217–226.

[Pla11] Pierre-Guy Plamondon. "Cluster characters for cluster categories with infinite-dimensional morphism spaces". In: *Adv. Math.* 227.1 (2011), pp. 1–39.

[Rin13] Claus Michael Ringel. "The (n-1)-antichains in a root poset of width n". In: (2013), pp. 1–16. arXiv: 1306.1593.

[Sev14] Ahmet I. Seven. "Maximal green sequences of exceptional finite mutation type quivers". In: *SIGMA Symmetry Integrability Geom. Methods Appl.* 10 (2014), Paper 089, 5.

[YZ08] Shih-Wei Yang and Andrei Zelevinsky. "Cluster algebras of finite type via Coxeter elements and principal minors". In: *Transform. Groups* 13.3-4 (2008), pp. 855–895.

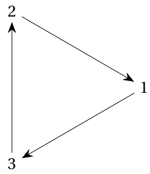
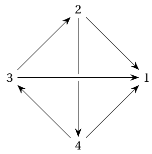
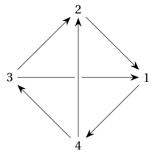
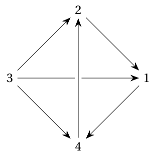
[Zha11] Pu Zhang. "Monomorphism categories, cotilting theory, and Gorenstein-projective modules". In: *J. Algebra* 339 (2011), pp. 181–202.

Appendices

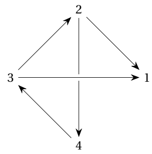
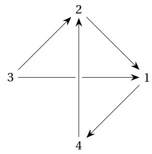
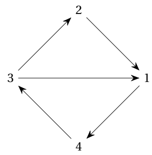
A — C

A

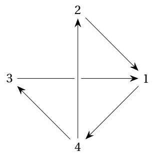
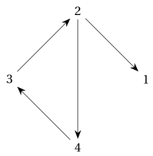
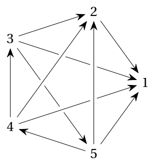
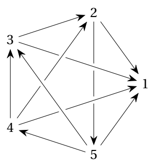
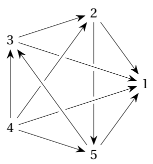
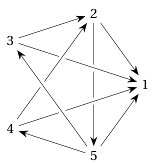
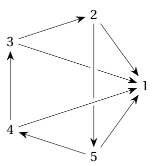
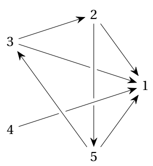
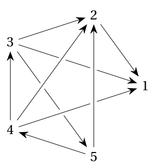
Cyclic quivers with maximal green sequences

Quiver	Maximal green sequence
	(1, 2, 3, 1)
	(1, 2, 3, 4, 2)
	(1, 2, 3, 4, 1)
	(1, 2, 4, 1, 3)

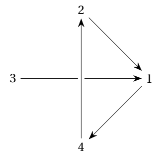
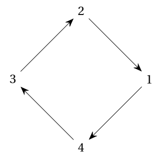
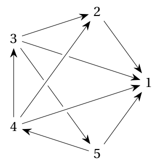
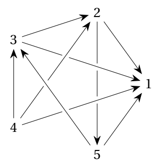
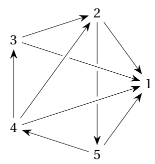
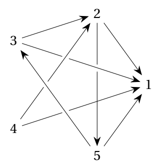
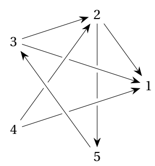
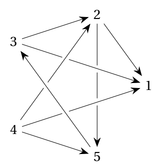
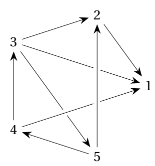
Continued in next column.

Quiver	Maximal green sequence
	(1, 2, 3, 4, 2)
	(1, 2, 4, 1, 3)
	(1, 4, 2, 3, 1)

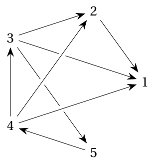
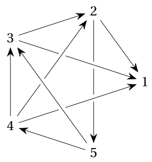
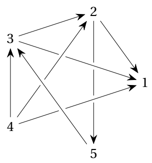
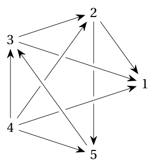
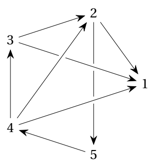
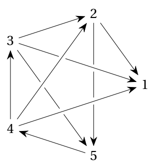
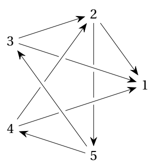
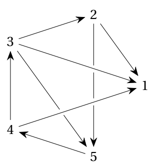
Continued on next page.

Quiver	Maximal green sequence
	(1, 2, 3, 4, 1)
	(1, 2, 3, 4, 2)
	(1, 2, 3, 4, 5, 3)
	(1, 2, 3, 4, 5, 2)
	(1, 2, 3, 5, 2, 4)
	(1, 2, 3, 4, 5, 2)
	(1, 2, 3, 4, 5, 3, 2)
	(1, 2, 3, 4, 5, 2)
	(1, 2, 3, 4, 5, 3)

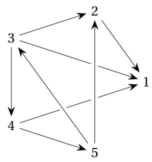
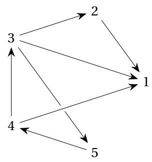
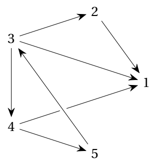
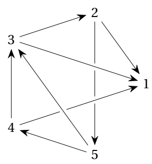
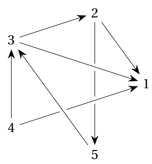
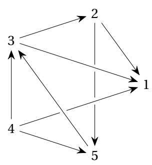
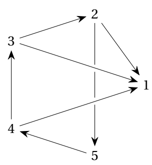
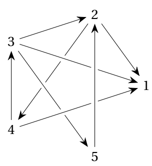
Continued in next column.

Quiver	Maximal green sequence
	(1, 2, 4, 1, 3)
	(1, 2, 3, 4, 2, 1)
	(1, 2, 3, 4, 5, 3)
	(1, 2, 3, 5, 2, 4)
	(1, 2, 5, 3, 4, 2)
	(1, 2, 3, 5, 2, 4)
	(1, 2, 3, 5, 2, 4)
	(1, 2, 3, 5, 2, 4)
	(1, 2, 3, 4, 5, 3)

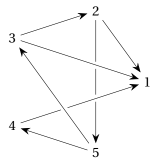
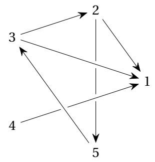
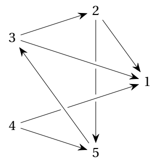
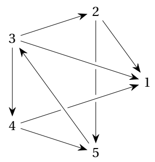
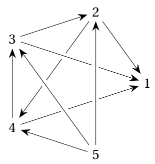
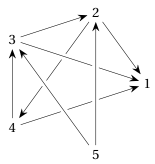
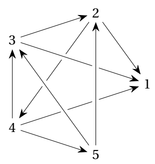
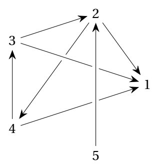
Continued on next page.

Quiver	Maximal green sequence
	(1, 2, 3, 4, 5, 3)
	(1, 2, 3, 4, 5, 2)
	(1, 2, 3, 5, 2, 4)
	(1, 2, 3, 5, 2, 4)
	(1, 2, 5, 3, 4, 2)
	(1, 5, 2, 3, 4, 5)
	(1, 2, 3, 4, 5, 2)
	(1, 4, 5, 2, 3, 4)

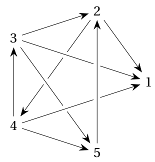
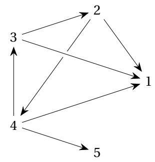
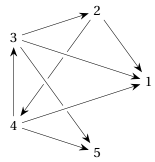
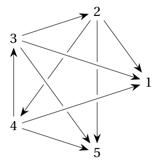
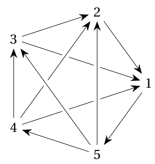
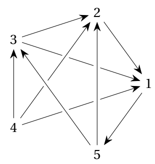
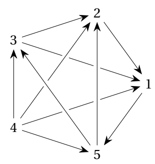
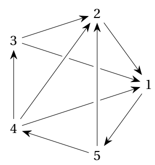
Continued in next column.

Quiver	Maximal green sequence
	(1, 2, 3, 4, 5, 3)
	(1, 2, 3, 4, 5, 3)
	(1, 2, 3, 4, 5, 3)
	(1, 2, 3, 4, 5, 2)
	(1, 2, 3, 4, 5, 2)
	(1, 2, 3, 5, 2, 4)
	(1, 2, 3, 4, 5, 3, 2)
	(1, 2, 4, 5, 3, 2)

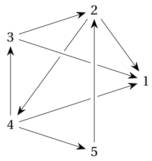
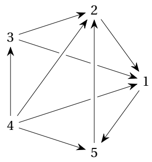
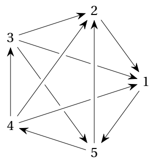
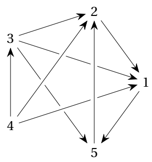
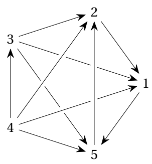
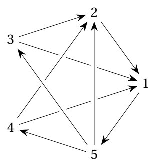
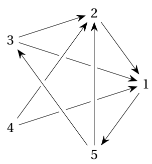
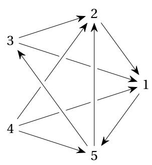
Continued on next page.

Quiver	Maximal green sequence
	(1, 2, 3, 4, 5, 2)
	(1, 2, 3, 4, 5, 2)
	(1, 2, 3, 5, 2, 4)
	(1, 5, 2, 3, 4, 5)
	(1, 2, 3, 4, 2, 5)
	(1, 2, 3, 4, 2, 5)
	(1, 2, 3, 4, 5, 2)
	(1, 2, 3, 4, 2, 5)

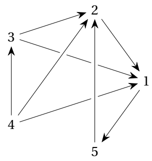
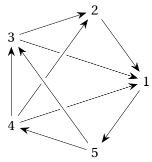
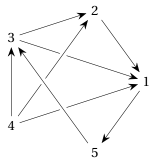
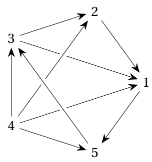
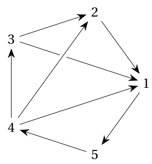
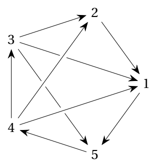
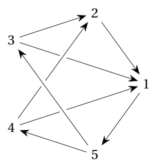
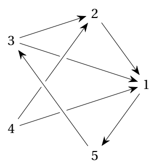
Continued in next column.

Quiver	Maximal green sequence
	(1, 2, 4, 5, 3, 2)
	(1, 2, 3, 5, 4, 2)
	(1, 2, 5, 3, 4, 2)
	(1, 5, 2, 3, 4, 2)
	(1, 2, 3, 4, 5, 1)
	(1, 2, 3, 5, 1, 4)
	(1, 2, 3, 5, 1, 4)
	(1, 2, 5, 3, 4, 1)

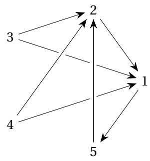
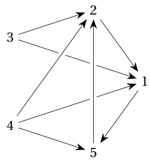
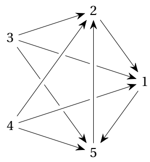
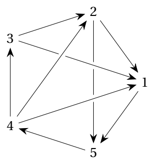
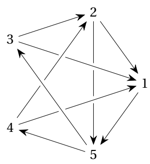
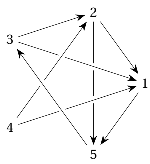
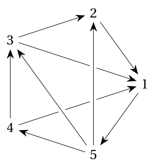
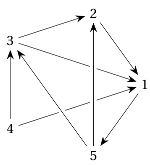
Continued on next page.

Quiver	Maximal green sequence
	(1, 2, 3, 4, 5, 2)
	(1, 2, 5, 1, 3, 4)
	(1, 2, 4, 5, 1, 3, 4)
	(1, 2, 5, 1, 3, 4)
	(1, 2, 5, 1, 3, 4)
	(1, 2, 3, 4, 5, 1)
	(1, 2, 3, 5, 1, 4)
	(1, 2, 3, 5, 1, 4)

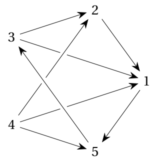
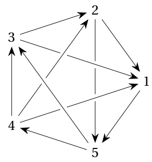
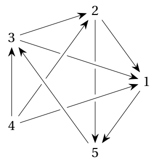
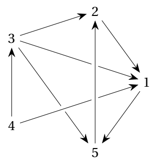
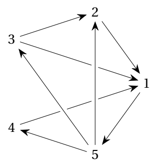
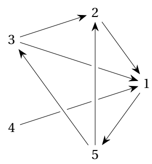
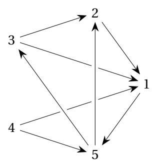
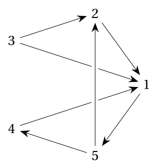
Continued in next column.

Quiver	Maximal green sequence
	(1, 2, 5, 1, 3, 4)
	(1, 5, 2, 3, 4, 1)
	(1, 5, 2, 3, 1, 4)
	(1, 5, 2, 3, 1, 4)
	(1, 5, 2, 3, 4, 1)
	(5, 1, 2, 3, 4, 5)
	(1, 5, 2, 3, 4, 1)
	(1, 5, 2, 3, 1, 4)

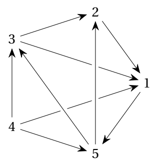
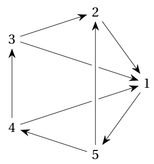
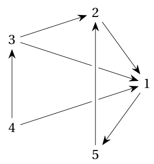
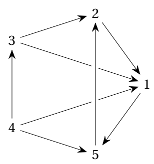
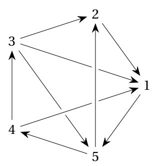
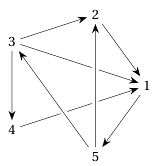
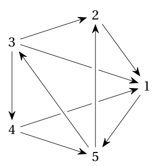
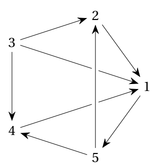
Continued on next page.

Quiver	Maximal green sequence
	(1, 2, 5, 1, 3, 4)
	(1, 2, 5, 1, 3, 4)
	(1, 2, 5, 1, 3, 4)
	(5, 1, 2, 3, 4, 5)
	(1, 5, 2, 1, 3, 4, 2)
	(5, 1, 2, 3, 4, 5)
	(1, 2, 3, 4, 5, 1)
	(1, 2, 3, 5, 1, 4)

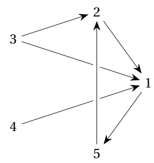
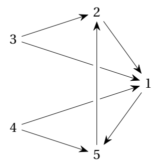
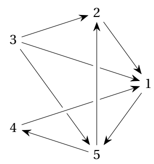
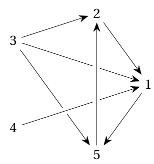
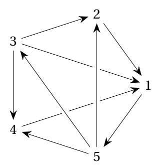
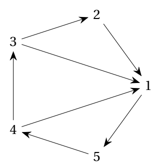
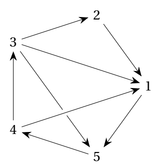
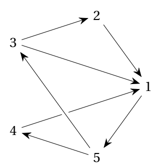
Continued in next column.

Quiver	Maximal green sequence
	(1, 5, 2, 3, 1, 4)
	(1, 5, 2, 1, 3, 4, 2)
	(5, 1, 2, 3, 5, 4)
	(1, 2, 5, 1, 3, 4)
	(1, 2, 3, 4, 5, 1)
	(1, 2, 3, 5, 1, 4)
	(1, 2, 3, 5, 1, 4)
	(1, 2, 4, 5, 1, 3)

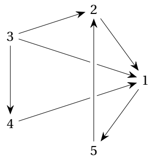
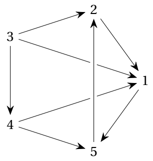
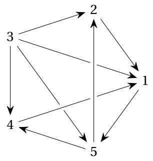
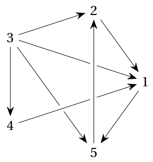
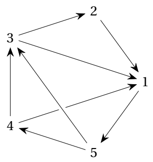
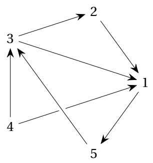
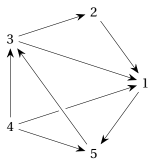
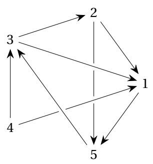
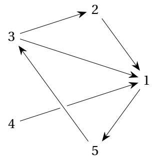
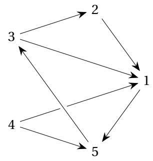
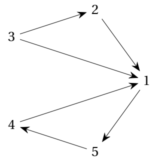
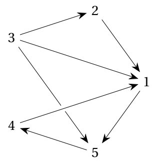
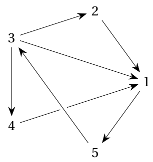
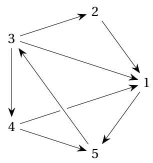
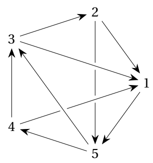
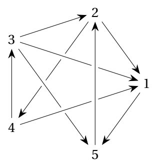
Continued on next page.

Quiver	Maximal green sequence
	(1, 2, 3, 5, 1, 4)
	(1, 2, 5, 3, 4, 1)
	(1, 2, 5, 1, 3, 4)
	(1, 2, 5, 1, 3, 4)
	(1, 2, 4, 5, 1, 3, 4)
	(1, 2, 5, 4, 3, 1)
	(1, 2, 3, 5, 1, 4, 3)
	(1, 2, 4, 5, 1, 3)

Continued in next column.

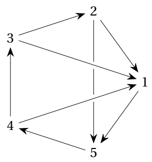
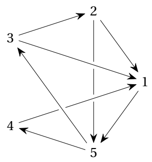
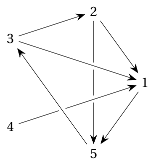
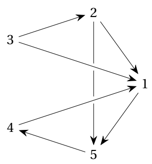
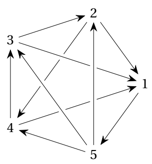
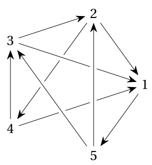
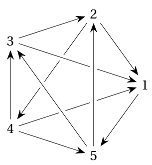
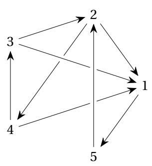
Quiver	Maximal green sequence
	(1, 2, 5, 1, 3, 4)
	(1, 2, 5, 1, 3, 4)
	(1, 2, 4, 5, 1, 3)
	(1, 2, 5, 1, 3, 4)
	(1, 2, 4, 3, 5, 1)
	(1, 5, 2, 3, 4, 1)
	(5, 1, 2, 3, 4, 5)
	(1, 4, 5, 2, 3, 1)

Continued on next page.

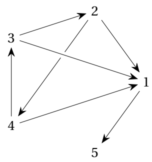
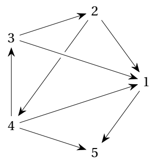
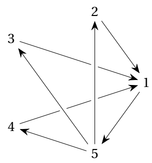
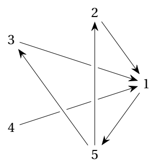
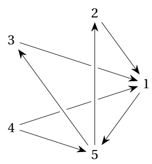
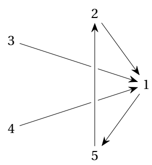
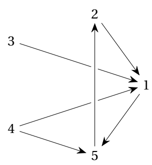
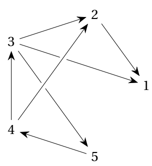
Quiver	Maximal green sequence
	(1, 2, 5, 1, 4, 3)
	(1, 2, 5, 1, 4, 3)
	(1, 2, 4, 5, 1, 3)
	(1, 2, 5, 1, 4, 3)
	(1, 5, 2, 3, 4, 1)
	(1, 5, 2, 3, 1, 4)
	(1, 5, 2, 3, 1, 4)
	(5, 1, 2, 3, 5, 4)
	(1, 5, 2, 3, 1, 4)
	(1, 5, 2, 3, 1, 4)
	(1, 4, 5, 1, 2, 3)
	(1, 4, 5, 1, 2, 3)
	(1, 5, 2, 4, 3, 1)
	(5, 1, 2, 3, 4, 5)
	(1, 3, 5, 2, 4, 1, 3)
	(1, 2, 5, 4, 1, 3, 2)

Continued in next column.

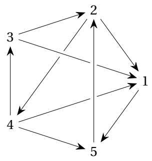
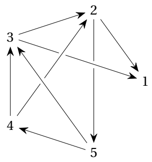
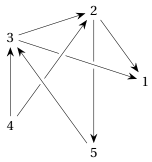
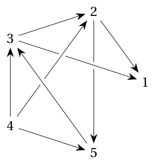
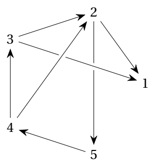
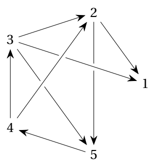
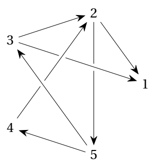
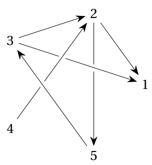
Continued on next page.

Quiver	Maximal green sequence
	(5, 1, 2, 3, 4, 2, 5)
	(1, 3, 4, 5, 1, 2, 3)
	(5, 1, 2, 3, 4, 5)
	(1, 4, 5, 1, 2, 3)
	(1, 2, 3, 4, 2, 5, 1)
	(1, 2, 3, 5, 4, 1, 2)
	(1, 2, 3, 5, 1, 4, 2)
	(1, 2, 5, 3, 4, 1, 2)

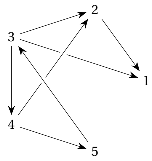
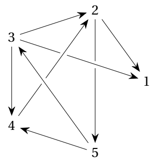
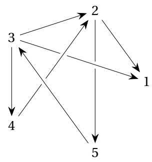
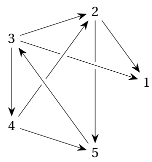
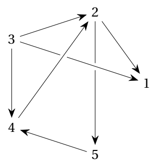
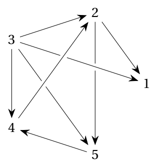
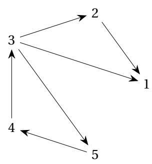
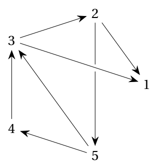
Continued in next column.

Quiver	Maximal green sequence
	(5, 1, 2, 3, 4, 2)
	(5, 1, 2, 3, 4, 2)
	(1, 2, 3, 4, 5, 1)
	(1, 2, 3, 5, 1, 4)
	(1, 2, 3, 5, 1, 4)
	(1, 2, 5, 1, 3, 4)
	(1, 2, 5, 1, 3, 4)
	(1, 2, 3, 4, 5, 3)

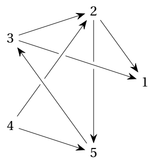
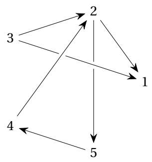
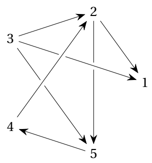
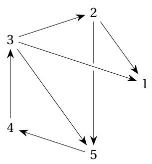
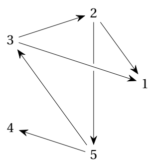
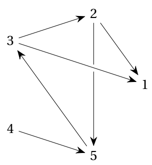
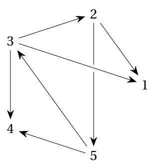
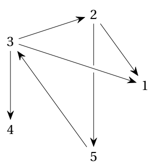
Continued on next page.

Quiver	Maximal green sequence
	(1, 2, 5, 1, 3, 4, 2)
	(1, 2, 3, 4, 5, 2)
	(1, 2, 3, 5, 2, 4)
	(1, 2, 3, 5, 2, 4)
	(1, 2, 5, 3, 4, 2)
	(1, 5, 2, 3, 4, 5)
	(1, 2, 3, 4, 5, 2)
	(1, 2, 3, 5, 2, 4)

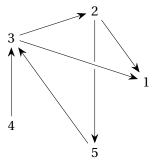
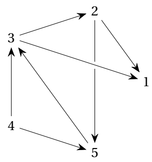
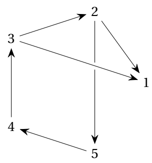
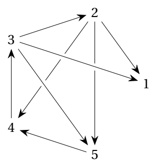
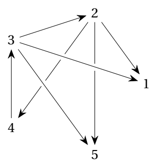
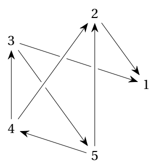
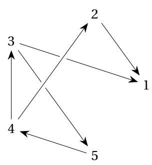
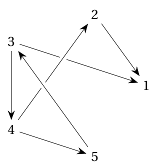
Continued in next column.

Quiver	Maximal green sequence
	(1, 2, 3, 4, 5, 3)
	(1, 2, 4, 3, 5, 2)
	(1, 2, 5, 4, 3, 2)
	(1, 5, 2, 3, 4, 5)
	(1, 2, 4, 5, 2, 3)
	(1, 2, 4, 5, 2, 3)
	(1, 2, 3, 4, 5, 3)
	(1, 2, 3, 4, 5, 2)

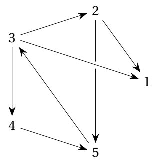
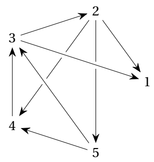
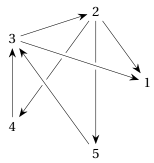
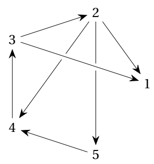
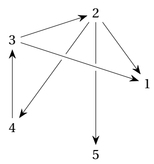
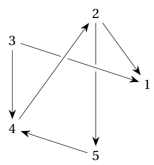
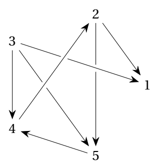
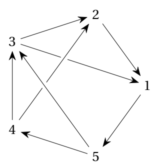
Continued on next page.

Quiver	Maximal green sequence
	(1, 2, 3, 5, 2, 4)
	(1, 2, 4, 5, 2, 3)
	(1, 2, 4, 5, 2, 3)
	(1, 4, 5, 2, 3, 4)
	(1, 2, 3, 4, 5, 2)
	(1, 2, 3, 5, 2, 4)
	(1, 2, 4, 3, 5, 2)
	(1, 2, 4, 3, 5, 2)

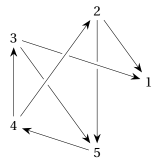
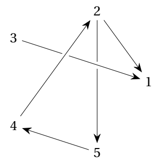
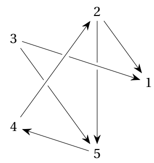
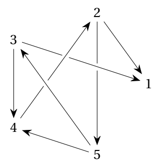
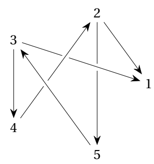
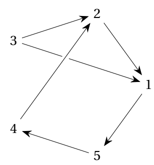
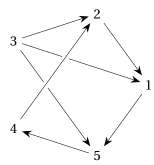
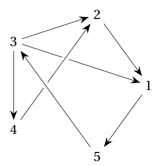
Continued in next column.

Quiver	Maximal green sequence
	(1, 2, 3, 4, 5, 2)
	(1, 2, 3, 5, 2, 4)
	(1, 2, 3, 4, 5, 3, 2)
	(1, 4, 3, 5, 2, 4)
	(1, 4, 5, 2, 3, 4)
	(1, 2, 3, 4, 5, 3)
	(1, 2, 3, 4, 5, 3)
	(1, 2, 3, 4, 5, 3)

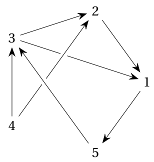
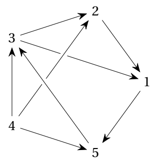
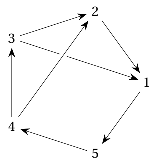
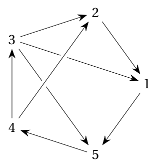
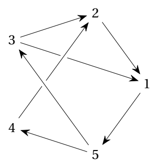
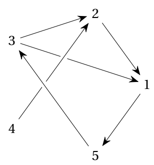
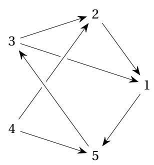
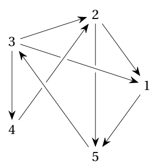
Continued on next page.

Quiver	Maximal green sequence
	(1, 5, 2, 3, 4, 5)
	(1, 3, 2, 4, 5, 3)
	(1, 3, 2, 4, 5, 3)
	(1, 3, 4, 5, 2, 3)
	(1, 3, 4, 5, 2, 3)
	(1, 2, 4, 3, 5, 2)
	(1, 2, 4, 5, 2, 3)
	(1, 2, 5, 3, 4, 2, 1)

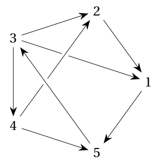
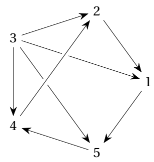
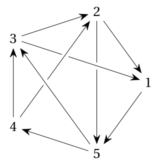
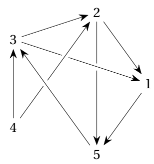
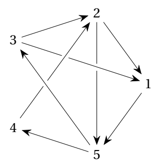
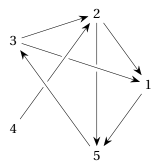
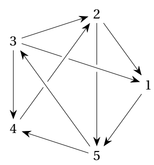
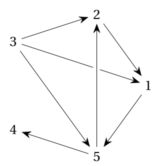
Continued in next column.

Quiver	Maximal green sequence
	(1, 5, 2, 3, 4, 5)
	(1, 2, 3, 4, 5, 2)
	(1, 2, 4, 5, 2, 3)
	(1, 2, 4, 3, 5, 2)
	(1, 2, 3, 4, 5, 2, 3)
	(1, 2, 4, 5, 2, 1, 3)
	(1, 2, 4, 5, 2, 1, 3)
	(1, 5, 2, 4, 3, 1)

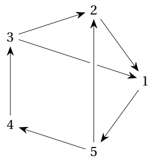
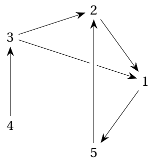
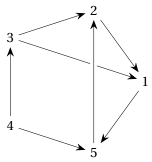
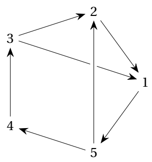
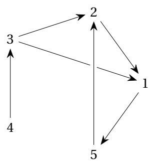
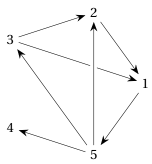
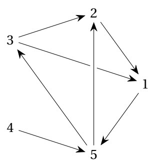
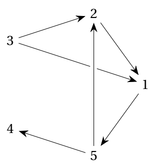
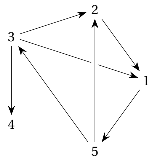
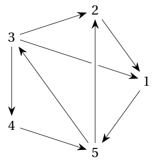
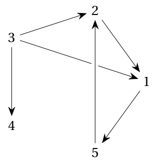
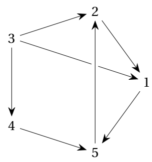
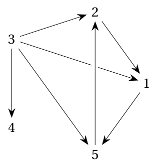
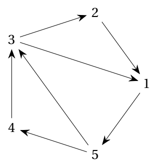
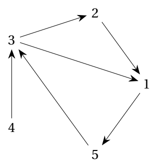
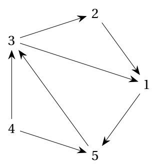
Continued on next page.

Quiver	Maximal green sequence
	(1, 5, 2, 3, 1, 4)
	(1, 5, 2, 3, 1, 4)
	(1, 5, 2, 3, 4, 5, 1)
	(2, 5, 1, 4, 2, 3, 5)
	(1, 2, 4, 5, 2, 3, 1)
	(1, 5, 2, 3, 1, 4)
	(1, 5, 2, 3, 1, 4)
	(5, 1, 2, 4, 3, 5)

Continued in next column.

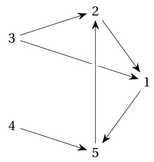
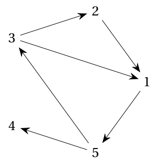
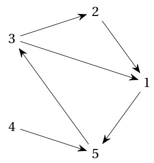
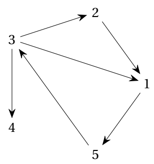
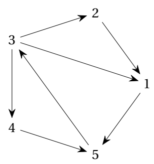
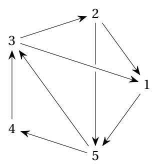
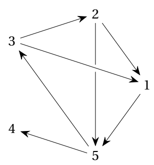
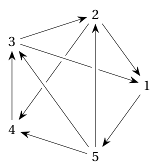
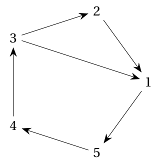
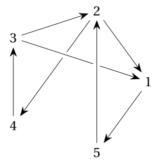
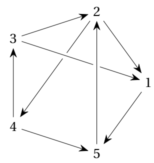
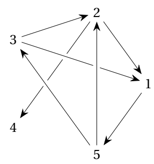
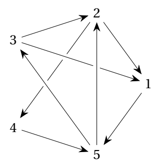
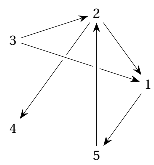
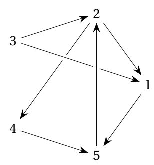
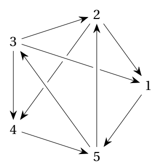
Quiver	Maximal green sequence
	(5, 1, 2, 3, 4, 5)
	(1, 2, 4, 5, 2, 1, 3)
	(1, 3, 5, 1, 2, 4, 3, 5, 1)
	(5, 1, 2, 3, 5, 4)
	(4, 5, 1, 2, 3, 4, 5)
	(5, 1, 2, 3, 4, 5)
	(4, 5, 1, 2, 3, 4, 5)
	(1, 2, 4, 5, 1, 3)

Continued on next page.

Quiver	Maximal green sequence
	(1, 2, 3, 4, 5, 3, 1)
	(1, 2, 5, 1, 3, 4)
	(1, 2, 5, 1, 3, 4)
	(1, 2, 4, 5, 1, 3, 4)
	(1, 2, 5, 1, 3, 4)
	(1, 2, 3, 4, 5, 1)
	(1, 2, 3, 5, 1, 4)
	(1, 2, 4, 5, 1, 3)
	(1, 2, 4, 3, 5, 1)
	(1, 2, 3, 5, 1, 4, 3)
	(1, 2, 4, 5, 1, 3)
	(1, 2, 5, 1, 4, 3)
	(1, 2, 4, 5, 1, 3)
	(1, 2, 3, 4, 5, 3, 2, 1)
	(1, 5, 2, 3, 1, 4)
	(1, 5, 2, 3, 1, 4)

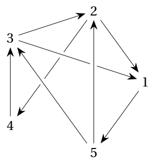
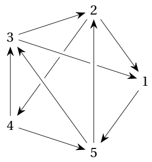
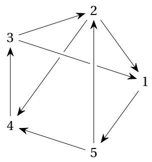
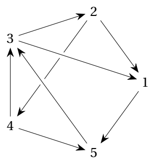
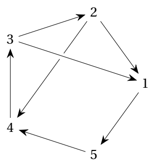
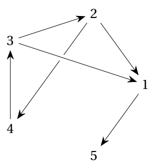
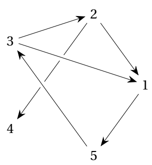
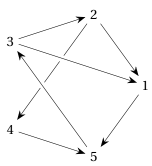
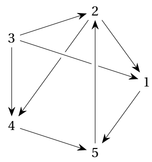
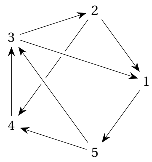
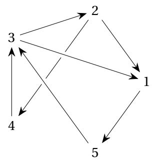
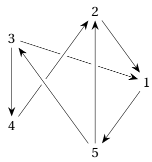
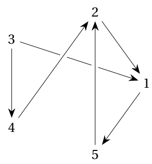
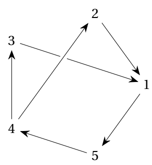
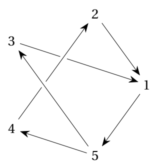
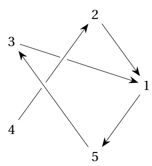
Continued in next column.

Continued on next page.

Quiver	Maximal green sequence
	(1, 2, 5, 1, 3, 4)
	(1, 4, 5, 2, 3, 1)
	(1, 5, 2, 3, 1, 4)
	(1, 4, 5, 2, 3, 1)
	(5, 1, 2, 3, 4, 5)
	(1, 3, 4, 5, 2, 1, 4, 3)
	(4, 5, 1, 2, 3, 5)
	(1, 2, 3, 4, 2, 5, 1)
	(1, 4, 5, 2, 3, 1, 4)
	(1, 2, 4, 5, 1, 3, 2)
	(1, 2, 5, 1, 3, 4, 2)
	(1, 4, 2, 3, 5, 1)
	(1, 2, 5, 4, 3, 1, 2)
	(1, 4, 2, 5, 1, 3)
	(5, 1, 2, 4, 5, 3)
	(1, 5, 4, 1, 2, 3, 4)

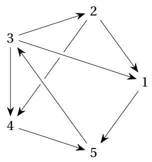
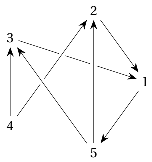
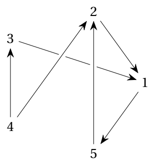
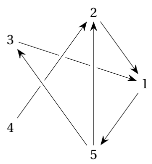
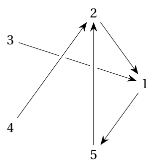
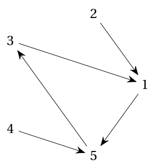
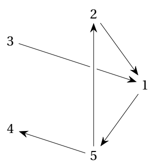
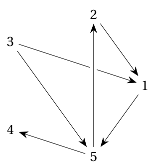
Continued in next column.

Continued on next page.

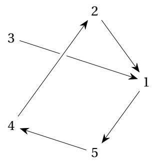
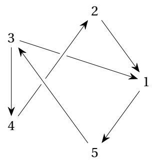
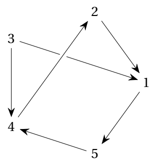
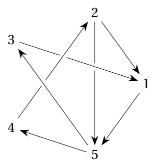
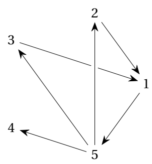
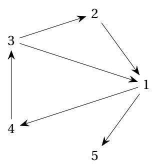
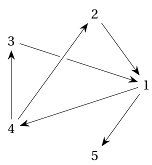
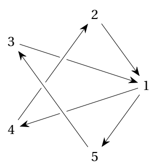
Quiver	Maximal green sequence
	(1, 2, 3, 4, 2, 5, 1)
	(1, 2, 3, 5, 1, 4, 2)
	(1, 4, 2, 3, 5, 1, 4)
	(1, 3, 5, 2, 1, 4, 3)
	(1, 4, 3, 5, 1, 2, 4)
	(4, 5, 1, 2, 3, 4)
	(1, 4, 5, 2, 3, 1)
	(5, 1, 2, 3, 4, 2, 5)
	(5, 1, 2, 4, 5, 3)
	(1, 3, 4, 5, 2, 1, 3)
	(1, 3, 4, 5, 2, 1, 3)
	(1, 2, 4, 3, 5, 1)
	(1, 2, 4, 5, 1, 3)
	(1, 5, 2, 3, 4, 5, 1)
	(1, 2, 3, 4, 5, 2, 1)
	(1, 3, 5, 1, 2, 4)

Continued in next column.

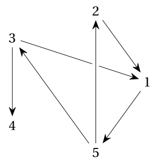
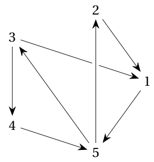
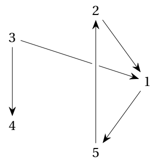
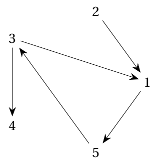
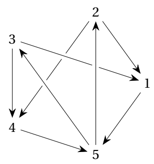
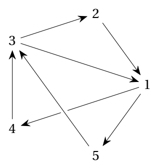
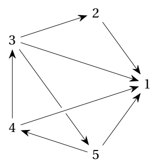
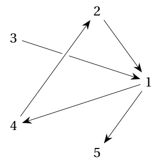
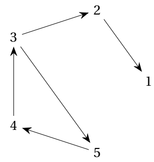
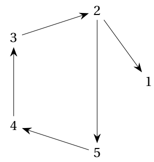
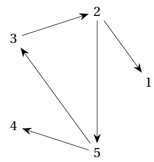
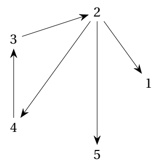
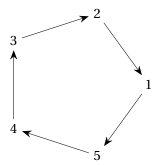
Continued on next page.

Quiver	Maximal green sequence
	(5, 1, 4, 2, 3, 5)
	(1, 2, 3, 4, 5, 1)
	(1, 2, 5, 1, 3, 4)
	(1, 2, 3, 4, 5, 1)
	(1, 2, 4, 5, 1, 3)
	(1, 2, 3, 5, 1, 4)
	(1, 2, 4, 5, 1, 3)
	(1, 2, 4, 5, 1, 3)

Continued in next column.

Quiver	Maximal green sequence
	(1, 2, 4, 5, 2, 1, 3)
	(1, 5, 2, 4, 3, 1)
	(1, 2, 4, 5, 2, 1, 3)
	(1, 3, 4, 5, 1, 2, 4)
	(1, 2, 3, 4, 5, 1)
	(4, 5, 1, 2, 3, 4)
	(5, 1, 2, 3, 4, 1)
	(2, 4, 5, 1, 2, 3, 5)

Continued on next page.

Quiver	Maximal green sequence
	(1, 2, 4, 3, 5, 1)
	(1, 2, 3, 5, 1, 4, 3)
	(1, 2, 4, 5, 1, 3)
	(1, 4, 3, 5, 1, 2)
	(1, 5, 4, 1, 2, 3, 4)
	(4, 1, 2, 3, 5, 2, 4, 1)
	(1, 2, 3, 4, 5, 3)
	(2, 4, 5, 1, 2, 3)
	(1, 2, 3, 4, 5, 3)
	(1, 2, 3, 4, 5, 3, 2)
	(1, 2, 3, 4, 5, 2)
	(1, 3, 4, 5, 2, 3)
	(1, 2, 3, 4, 5, 3, 2, 1)

Continued in next column.

Table finished.

B

Extended exchange matrices for proofs in Section 4.3

B.1 Extended exchange matrices for the proof of Lemma 4.3.6

B.1.1 Extended exchange matrices for mutations of $Q_{\mathrm{tri}}^{\mathrm{source}}$

Mutation sequence	Extended exchange matrix after application to $Q_{\mathrm{tri}}^{\mathrm{source}}$
()	$\begin{bmatrix} 0 & -1 & -1 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ -b & c & a & d \end{bmatrix}$
(2)	$\begin{bmatrix} 0 & 1 & -2 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ -b+c & -c & a & d \end{bmatrix}$

Continued on next page.

Mutation sequence	Extended exchange matrix after application to $Q_{\text{tri}}^{\text{source}}$
(2, 4)	$\begin{bmatrix} 0 & 1 & -1 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ -b+c & -c & a+d & -d \end{bmatrix}$
(4)	$\begin{bmatrix} 0 & -1 & -1 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ -b & c & a+d & -d \end{bmatrix}$
(4, 2)	$\begin{bmatrix} 0 & 1 & -2 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ -b+c & -c & a+d & -d \end{bmatrix}$
(4, 3)	$\begin{bmatrix} 0 & -1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & -1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ a-b+d & a+c+d & -a-d & a \end{bmatrix}$
(4, 3, 1)	$\begin{bmatrix} 0 & 1 & -1 & 0 \\ -1 & 0 & 2 & 0 \\ 1 & -2 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ -a+b-d & a+c+d & -b & a \end{bmatrix}$
(4, 3, 1, 4)	$\begin{bmatrix} 0 & 1 & -1 & 0 \\ -1 & 0 & 2 & 0 \\ 1 & -2 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ -a+b-d & a+c+d & a-b & -a \end{bmatrix}$

Continued on next page.

Mutation sequence	Extended exchange matrix after application to $Q_{\text{tri}}^{\text{source}}$
(4, 3, 1, 4, 3)	$\begin{bmatrix} 0 & -1 & 1 & 0 \\ 1 & 0 & -2 & 2 \\ -1 & 2 & 0 & -1 \\ 0 & -2 & 1 & 0 \\ -d & a+c+d & -a+b & -b \end{bmatrix}$
(4, 3, 2)	$\begin{bmatrix} 0 & 1 & 1 & 0 \\ -1 & 0 & -1 & 0 \\ -1 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 2a-b+c+2d & -a-c-d & c & a \end{bmatrix}$
(4, 3, 2, 1)	$\begin{bmatrix} 0 & -1 & -1 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ -2a+b-c-2d & a-b+d & 2a-b+2c+2d & a \end{bmatrix}$
(4, 3, 2, 3)	$\begin{bmatrix} 0 & 2 & -1 & 0 \\ -2 & 0 & 1 & -1 \\ 1 & -1 & 0 & 1 \\ 0 & 1 & -1 & 0 \\ 2a-b+c+2d & -a-d & -c & a \end{bmatrix}$
(4, 3, 2, 3, 4)	$\begin{bmatrix} 0 & 2 & -1 & 0 \\ -2 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 \\ 2a-b+c+2d & -d & -c & -a \end{bmatrix}$
(4, 3, 2, 4)	$\begin{bmatrix} 0 & 1 & 1 & 0 \\ -1 & 0 & -1 & 0 \\ -1 & 1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 2a-b+c+2d & -a-c-d & a+c & -a \end{bmatrix}$

Continued on next page.

Mutation sequence	Extended exchange matrix after application to $Q_{\text{tri}}^{\text{source}}$
(4, 3, 2, 4, 1)	$\begin{bmatrix} 0 & -1 & -1 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ -2a+b-c-2d & a-b+d & 3a-b+2c+2d & -a \end{bmatrix}$
(4, 3, 2, 4, 3)	$\begin{bmatrix} 0 & 2 & -1 & 0 \\ -2 & 0 & 1 & 0 \\ 1 & -1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 2a-b+c+2d & -d & -a-c & c \end{bmatrix}$
(4, 3, 2, 4, 3, 4)	$\begin{bmatrix} 0 & 2 & -1 & 0 \\ -2 & 0 & 1 & 0 \\ 1 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 2a-b+c+2d & -d & -a & -c \end{bmatrix}$
(4, 3, 4)	$\begin{bmatrix} 0 & -1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ a-b+d & a+c+d & -d & -a \end{bmatrix}$
(3)	$\begin{bmatrix} 0 & -1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ -1 & -1 & 0 & 1 \\ 1 & 1 & -1 & 0 \\ a-b & a+c & -a & d \end{bmatrix}$
(3, 1)	$\begin{bmatrix} 0 & 1 & -1 & 1 \\ -1 & 0 & 2 & -1 \\ 1 & -2 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -a+b & a+c & -b & d \end{bmatrix}$

Continued on next page.

Mutation sequence	Extended exchange matrix after application to $Q_{\text{tri}}^{\text{source}}$
(3, 1, 4)	$\begin{bmatrix} 0 & 2 & -1 & -1 \\ -2 & 0 & 2 & 1 \\ 1 & -2 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ -a+b & a+c+d & -b & -d \end{bmatrix}$
(3, 2)	$\begin{bmatrix} 0 & 1 & 1 & -2 \\ -1 & 0 & -1 & 1 \\ -1 & 1 & 0 & 0 \\ 2 & -1 & 0 & 0 \\ 2a-b+c & -a-c & c & d \end{bmatrix}$
(3, 2, 1)	$\begin{bmatrix} 0 & -1 & -1 & 2 \\ 1 & 0 & -1 & -1 \\ 1 & 1 & 0 & -2 \\ -2 & 1 & 2 & 0 \\ -2a+b-c & a-b & 2a-b+2c & d \end{bmatrix}$
(3, 2, 3)	$\begin{bmatrix} 0 & 2 & -1 & -2 \\ -2 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 2 & -1 & 0 & 0 \\ 2a-b+c & -a & -c & d \end{bmatrix}$
(3, 4)	$\begin{bmatrix} 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \\ -1 & -1 & 1 & 0 \\ a-b+d & a+c+d & -a & -d \end{bmatrix}$

Table finished.

B.1.2 Extended exchange matrices for mutations of $Q_{\text{tri}}^{\text{sink}}$

Mutation sequence	Extended exchange matrix after application to $Q_{\text{tri}}^{\text{sink}}$
()	$\begin{bmatrix} 0 & -1 & -1 & -1 \\ 1 & 0 & -1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ -b & c & a & d \end{bmatrix}$
(2)	$\begin{bmatrix} 0 & 1 & -2 & -1 \\ -1 & 0 & 1 & 0 \\ 2 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ -b+c & -c & a & d \end{bmatrix}$
(2, 4)	$\begin{bmatrix} 0 & 1 & -2 & 1 \\ -1 & 0 & 1 & 0 \\ 2 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -b+c+d & -c & a & -d \end{bmatrix}$
(2, 4, 1)	$\begin{bmatrix} 0 & -1 & 2 & -1 \\ 1 & 0 & -1 & 0 \\ -2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ b-c-d & -b+d & a & -b+c \end{bmatrix}$
(3)	$\begin{bmatrix} 0 & -1 & 1 & -1 \\ 1 & 0 & 1 & 0 \\ -1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ a-b & a+c & -a & d \end{bmatrix}$
(3, 1)	$\begin{bmatrix} 0 & 1 & -1 & 1 \\ -1 & 0 & 2 & 0 \\ 1 & -2 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ -a+b & a+c & -b & d \end{bmatrix}$

Continued on next page.

Mutation sequence	Extended exchange matrix after application to $Q_{\text{tri}}^{\text{sink}}$
(3, 1, 4)	$\begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & 2 & 0 \\ 0 & -2 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ -a+b & a+c & -b+d & -d \end{bmatrix}$
(3, 2)	$\begin{bmatrix} 0 & 1 & 1 & -1 \\ -1 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 2a-b+c & -a-c & c & d \end{bmatrix}$
(3, 4)	$\begin{bmatrix} 0 & -1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ -1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ a-b+d & a+c & -a & -d \end{bmatrix}$
(4)	$\begin{bmatrix} 0 & -1 & -1 & 1 \\ 1 & 0 & -1 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -b+d & c & a & -d \end{bmatrix}$

Table finished.

B.2 B- and C-matrices for the proof of Theorem 4.3.9

Mutation sequence	B-matrix after application to Q_{pent}	C-matrix after application to Q_{pent}
()	$\begin{bmatrix} 0 & -1 & -1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 \\ 1 & 1 & 0 & -1 & -1 \\ -1 & 1 & 1 & 0 & -1 \\ -1 & -1 & 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(1)	$\begin{bmatrix} 0 & 1 & 1 & -1 & -1 \\ -1 & 0 & -1 & 0 & 2 \\ -1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 1 & -2 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(1, 3)	$\begin{bmatrix} 0 & 2 & -1 & -1 & -1 \\ -2 & 0 & 1 & 0 & 2 \\ 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 1 & -2 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(1, 3, 4)	$\begin{bmatrix} 0 & 2 & -1 & 1 & -2 \\ -2 & 0 & 1 & 0 & 2 \\ 1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 2 & -2 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(1, 3, 5)	$\begin{bmatrix} 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 1 & 2 & -2 \\ 1 & -1 & 0 & 0 & 0 \\ 1 & -2 & 0 & 0 & 1 \\ -1 & 2 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 2 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & -1 \end{bmatrix}$
(1, 3, 5, 1)	$\begin{bmatrix} 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & 1 & 2 & -2 \\ -1 & -1 & 0 & 0 & 1 \\ -1 & -2 & 0 & 0 & 2 \\ 1 & 2 & -1 & -2 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 2 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 & 0 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q_{pent}	C-matrix after application to Q_{pent}
(1, 3, 5, 4)	$\begin{bmatrix} 0 & -2 & -1 & 1 & 1 \\ 2 & 0 & 1 & -2 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 & 0 & -2 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 1 \\ 2 & 0 & 0 & -1 & 0 \end{bmatrix}$
(1, 4)	$\begin{bmatrix} 0 & 1 & 1 & 1 & -2 \\ -1 & 0 & -1 & 0 & 2 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 2 & -2 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(1, 4, 1)	$\begin{bmatrix} 0 & -1 & -1 & -1 & 2 \\ 1 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & -2 \\ 1 & 0 & 0 & 0 & -1 \\ -2 & 0 & 2 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(1, 4, 1, 2)	$\begin{bmatrix} 0 & 1 & -2 & -1 & 2 \\ -1 & 0 & 1 & 0 & 0 \\ 2 & -1 & 0 & 0 & -2 \\ 1 & 0 & 0 & 0 & -1 \\ -2 & 0 & 2 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(1, 4, 1, 3)	$\begin{bmatrix} 0 & -1 & 1 & -1 & 0 \\ 1 & 0 & 1 & 0 & -2 \\ -1 & -1 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 2 & -2 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 \\ 0 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(1, 4, 1, 3, 1)	$\begin{bmatrix} 0 & 1 & -1 & 1 & 0 \\ -1 & 0 & 2 & 0 & -2 \\ 1 & -2 & 0 & -1 & 2 \\ -1 & 0 & 1 & 0 & -1 \\ 0 & 2 & -2 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q_{pent}	C-matrix after application to Q_{pent}
(1, 4, 1, 3, 2)	$\begin{bmatrix} 0 & 1 & 1 & -1 & -2 \\ -1 & 0 & -1 & 0 & 2 \\ -1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 2 & -2 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & -1 & 1 \\ 1 & -1 & 1 & 0 & 0 \\ 2 & -1 & 0 & 0 & 0 \\ 2 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(1, 5)	$\begin{bmatrix} 0 & -1 & 1 & -1 & 1 \\ 1 & 0 & -1 & 2 & -2 \\ -1 & 1 & 0 & 0 & 0 \\ 1 & -2 & 0 & 0 & 1 \\ -1 & 2 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 2 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & -1 \end{bmatrix}$
(1, 5, 1)	$\begin{bmatrix} 0 & 1 & -1 & 1 & -1 \\ -1 & 0 & 0 & 2 & -1 \\ 1 & 0 & 0 & -1 & 0 \\ -1 & -2 & 1 & 0 & 2 \\ 1 & 1 & 0 & -2 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 2 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & 1 & 0 \end{bmatrix}$
(1, 5, 1, 3)	$\begin{bmatrix} 0 & 1 & 1 & 0 & -1 \\ -1 & 0 & 0 & 2 & -1 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & -2 & -1 & 0 & 2 \\ 1 & 1 & 0 & -2 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 2 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$
(1, 5, 1, 3, 1)	$\begin{bmatrix} 0 & -1 & -1 & 0 & 1 \\ 1 & 0 & 0 & 2 & -2 \\ 1 & 0 & 0 & 1 & -1 \\ 0 & -2 & -1 & 0 & 2 \\ -1 & 2 & 1 & -2 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 2 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$
(1, 5, 4)	$\begin{bmatrix} 0 & -3 & 1 & 1 & 1 \\ 3 & 0 & -1 & -2 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 & 0 & -2 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 1 \\ 2 & 0 & 0 & -1 & 0 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q_{pent}	C-matrix after application to Q_{pent}
(1, 5, 4, 1)	$\begin{bmatrix} 0 & 3 & -1 & -1 & -1 \\ -3 & 0 & 2 & 1 & 3 \\ 1 & -2 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 \\ 1 & -3 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -2 & 0 & 2 & 0 & 3 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 2 \\ -2 & 0 & 2 & 1 & 2 \end{bmatrix}$
(1, 5, 4, 3)	$\begin{bmatrix} 0 & -2 & -1 & 1 & 1 \\ 2 & 0 & 1 & -2 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 & 0 & -2 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 1 \\ 2 & 0 & 0 & -1 & 0 \end{bmatrix}$
(1, 5, 4, 5)	$\begin{bmatrix} 0 & -3 & 1 & 2 & -1 \\ 3 & 0 & -1 & -2 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ -2 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 2 & 0 & 0 & -1 & 0 \end{bmatrix}$
(1, 5, 4, 1, 3)	$\begin{bmatrix} 0 & 1 & 1 & -1 & -1 \\ -1 & 0 & -2 & 1 & 3 \\ -1 & 2 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 \\ 1 & -3 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & -2 & 0 & 3 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 2 \\ 0 & 0 & -2 & 1 & 2 \end{bmatrix}$
(1, 5, 4, 1, 4)	$\begin{bmatrix} 0 & 2 & -1 & 1 & -2 \\ -2 & 0 & 2 & -1 & 3 \\ 1 & -2 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 \\ 2 & -3 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -2 & 0 & 2 & 0 & 3 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 2 \\ -1 & 0 & 2 & -1 & 2 \end{bmatrix}$
(1, 5, 4, 1, 5)	$\begin{bmatrix} 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 2 & 4 & -3 \\ 1 & -2 & 0 & 0 & 0 \\ 1 & -4 & 0 & 0 & 1 \\ -1 & 3 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 2 & 3 & -3 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 2 & -2 \\ 0 & 0 & 2 & 3 & -2 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q_{pent}	C-matrix after application to Q_{pent}
$(1,5,4,5,3)$	$\begin{bmatrix} 0 & -2 & -1 & 2 & -1 \\ 2 & 0 & 1 & -2 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ -2 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 2 & 0 & 0 & -1 & 0 \end{bmatrix}$
$(1,5,4,1,3,1)$	$\begin{bmatrix} 0 & -1 & -1 & 1 & 1 \\ 1 & 0 & -2 & 0 & 2 \\ 1 & 2 & 0 & -1 & -1 \\ -1 & 0 & 1 & 0 & -1 \\ -1 & -2 & 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & -2 & 0 & 3 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 2 \\ 0 & 0 & -2 & 1 & 2 \end{bmatrix}$
$(1,5,4,1,3,4)$	$\begin{bmatrix} 0 & 0 & 1 & 1 & -2 \\ 0 & 0 & -2 & -1 & 3 \\ -1 & 2 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 \\ 2 & -3 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & -2 & 0 & 3 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 2 \\ 1 & 0 & -2 & -1 & 2 \end{bmatrix}$
$(1,5,4,1,3,5)$	$\begin{bmatrix} 0 & -2 & 1 & -1 & 1 \\ 2 & 0 & -2 & 4 & -3 \\ -1 & 2 & 0 & 0 & 0 \\ 1 & -4 & 0 & 0 & 1 \\ -1 & 3 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 3 & 0 & -2 & 3 & -3 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 2 & 0 & -1 & 2 & -2 \\ 2 & 0 & -2 & 3 & -2 \end{bmatrix}$
$(1,5,4,1,5,1)$	$\begin{bmatrix} 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & 2 & 4 & -3 \\ -1 & -2 & 0 & 0 & 1 \\ -1 & -4 & 0 & 0 & 2 \\ 1 & 3 & -1 & -2 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 2 & 3 & -2 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 2 & -1 \\ 0 & 0 & 2 & 3 & -2 \end{bmatrix}$
$(1,5,4,1,5,4)$	$\begin{bmatrix} 0 & -4 & -1 & 1 & 1 \\ 4 & 0 & 2 & -4 & 1 \\ 1 & -2 & 0 & 0 & 0 \\ -1 & 4 & 0 & 0 & -1 \\ -1 & -1 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 4 & 0 & 2 & -3 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 1 & -2 & 0 \\ 3 & 0 & 2 & -3 & 1 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q_{pent}	C-matrix after application to Q_{pent}
(1, 5, 4, 1, 3, 4, 1)	$\begin{bmatrix} 0 & 0 & -1 & -1 & 2 \\ 0 & 0 & -2 & -1 & 3 \\ 1 & 2 & 0 & 0 & -2 \\ 1 & 1 & 0 & 0 & -1 \\ -2 & -3 & 2 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & -2 & 0 & 3 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 2 \\ -1 & 0 & -1 & 0 & 2 \end{bmatrix}$
(1, 5, 4, 1, 3, 5, 1)	$\begin{bmatrix} 0 & 2 & -1 & 1 & -1 \\ -2 & 0 & 0 & 4 & -1 \\ 1 & 0 & 0 & -1 & 0 \\ -1 & -4 & 1 & 0 & 2 \\ 1 & 1 & 0 & -2 & 0 \end{bmatrix}$	$\begin{bmatrix} -3 & 0 & 1 & 3 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ -2 & 0 & 1 & 2 & 0 \\ -2 & 0 & 0 & 3 & 0 \end{bmatrix}$
(1, 5, 4, 1, 3, 5, 4)	$\begin{bmatrix} 0 & -6 & 1 & 1 & 1 \\ 6 & 0 & -2 & -4 & 1 \\ -1 & 2 & 0 & 0 & 0 \\ -1 & 4 & 0 & 0 & -1 \\ -1 & -1 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 6 & 0 & -2 & -3 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 4 & 0 & -1 & -2 & 0 \\ 5 & 0 & -2 & -3 & 1 \end{bmatrix}$
(1, 5, 4, 1, 5, 1, 3)	$\begin{bmatrix} 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -2 & 4 & -1 \\ 1 & 2 & 0 & 0 & -1 \\ -1 & -4 & 0 & 0 & 2 \\ 0 & 1 & 1 & -2 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & -2 & 3 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ -1 & 0 & -1 & 2 & 0 \\ 0 & 0 & -2 & 3 & 0 \end{bmatrix}$
(1, 5, 4, 1, 5, 4, 1)	$\begin{bmatrix} 0 & 4 & 1 & -1 & -1 \\ -4 & 0 & 2 & 0 & 5 \\ -1 & -2 & 0 & 1 & 1 \\ 1 & 0 & -1 & 0 & -1 \\ 1 & -5 & -1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -4 & 0 & 2 & 1 & 4 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -3 & 0 & 1 & 1 & 3 \\ -3 & 0 & 2 & 0 & 4 \end{bmatrix}$
(1, 5, 4, 1, 5, 4, 5)	$\begin{bmatrix} 0 & -4 & -1 & 2 & -1 \\ 4 & 0 & 2 & -3 & -1 \\ 1 & -2 & 0 & 0 & 0 \\ -2 & 3 & 0 & 0 & 1 \\ 1 & 1 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 4 & 0 & 2 & -3 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 1 & -2 & 0 \\ 3 & 0 & 2 & -2 & -1 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q_{pent}	C-matrix after application to Q_{pent}
(1, 5, 4, 1, 3, 4, 1, 4)	$\begin{bmatrix} 0 & 0 & -1 & 1 & 1 \\ 0 & 0 & -2 & 1 & 2 \\ 1 & 2 & 0 & 0 & -2 \\ -1 & -1 & 0 & 0 & 1 \\ -1 & -2 & 2 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & -2 & 0 & 3 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 2 \\ -1 & 0 & -1 & 0 & 2 \end{bmatrix}$
(1, 5, 4, 1, 3, 5, 1, 3)	$\begin{bmatrix} 0 & 2 & 1 & 0 & -1 \\ -2 & 0 & 0 & 4 & -1 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & -4 & -1 & 0 & 2 \\ 1 & 1 & 0 & -2 & 0 \end{bmatrix}$	$\begin{bmatrix} -2 & 0 & -1 & 3 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ -1 & 0 & -1 & 2 & 0 \\ -2 & 0 & 0 & 3 & 0 \end{bmatrix}$
(1, 5, 4, 1, 3, 5, 1, 5)	$\begin{bmatrix} 0 & 2 & -1 & -1 & 1 \\ -2 & 0 & 0 & 2 & 1 \\ 1 & 0 & 0 & -1 & 0 \\ 1 & -2 & 1 & 0 & -2 \\ -1 & -1 & 0 & 2 & 0 \end{bmatrix}$	$\begin{bmatrix} -3 & 0 & 1 & 3 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ -2 & 0 & 1 & 2 & 0 \\ -2 & 0 & 0 & 3 & 0 \end{bmatrix}$
(1, 5, 4, 1, 3, 5, 4, 1)	$\begin{bmatrix} 0 & 6 & -1 & -1 & -1 \\ -6 & 0 & 4 & 2 & 7 \\ 1 & -4 & 0 & 0 & 0 \\ 1 & -2 & 0 & 0 & -1 \\ 1 & -7 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -6 & 0 & 4 & 3 & 6 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 \\ -4 & 0 & 3 & 2 & 4 \\ -5 & 0 & 3 & 2 & 6 \end{bmatrix}$
(1, 5, 4, 1, 3, 5, 4, 5)	$\begin{bmatrix} 0 & -6 & 1 & 2 & -1 \\ 6 & 0 & -2 & -3 & -1 \\ -1 & 2 & 0 & 0 & 0 \\ -2 & 3 & 0 & 0 & 1 \\ 1 & 1 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 6 & 0 & -2 & -3 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 4 & 0 & -1 & -2 & 0 \\ 5 & 0 & -2 & -2 & -1 \end{bmatrix}$
(1, 5, 4, 1, 5, 1, 3, 5)	$\begin{bmatrix} 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -2 & 2 & 1 \\ 1 & 2 & 0 & -2 & 1 \\ -1 & -2 & 2 & 0 & -2 \\ 0 & -1 & -1 & 2 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & -2 & 3 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & -1 & 2 & 0 \\ 0 & 0 & -2 & 3 & 0 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q_{pent}	C-matrix after application to Q_{pent}
(1, 5, 4, 1, 5, 4, 1, 3)	$\begin{bmatrix} 0 & 4 & -1 & 0 & 0 \\ -4 & 0 & -2 & 2 & 7 \\ 1 & 2 & 0 & -1 & -1 \\ 0 & -2 & 1 & 0 & -1 \\ 0 & -7 & 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -4 & 0 & -2 & 3 & 6 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 \\ -3 & 0 & -1 & 2 & 4 \\ -3 & 0 & -2 & 2 & 6 \end{bmatrix}$
(1, 5, 4, 1, 5, 4, 1, 4)	$\begin{bmatrix} 0 & 4 & 0 & 1 & -2 \\ -4 & 0 & 2 & 0 & 5 \\ 0 & -2 & 0 & -1 & 1 \\ -1 & 0 & 1 & 0 & 1 \\ 2 & -5 & -1 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -3 & 0 & 2 & -1 & 4 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -2 & 0 & 1 & -1 & 3 \\ -3 & 0 & 2 & 0 & 4 \end{bmatrix}$
(1, 5, 4, 1, 5, 4, 1, 5)	$\begin{bmatrix} 0 & -1 & 0 & -1 & 1 \\ 1 & 0 & 2 & 5 & -5 \\ 0 & -2 & 0 & 2 & -1 \\ 1 & -5 & -2 & 0 & 1 \\ -1 & 5 & 1 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 2 & 5 & -4 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & -3 \\ 1 & 0 & 2 & 4 & -4 \end{bmatrix}$
(1, 5, 4, 1, 5, 4, 1, 3, 4)	$\begin{bmatrix} 0 & 4 & -1 & 0 & 0 \\ -4 & 0 & 0 & -2 & 7 \\ 1 & 0 & 0 & 1 & -2 \\ 0 & 2 & -1 & 0 & 1 \\ 0 & -7 & 2 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -4 & 0 & 1 & -3 & 6 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ -3 & 0 & 1 & -2 & 4 \\ -3 & 0 & 0 & -2 & 6 \end{bmatrix}$
(1, 5, 4, 1, 5, 4, 1, 3, 5)	$\begin{bmatrix} 0 & 4 & -1 & 0 & 0 \\ -4 & 0 & 5 & 9 & -7 \\ 1 & -5 & 0 & -1 & 1 \\ 0 & -9 & 1 & 0 & 1 \\ 0 & 7 & -1 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -4 & 0 & 4 & 9 & -6 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & -1 \\ -3 & 0 & 3 & 6 & -4 \\ -3 & 0 & 4 & 8 & -6 \end{bmatrix}$
(1, 5, 4, 1, 5, 4, 1, 4, 3)	$\begin{bmatrix} 0 & 4 & 0 & 1 & -2 \\ -4 & 0 & -2 & 0 & 7 \\ 0 & 2 & 0 & 1 & -1 \\ -1 & 0 & -1 & 0 & 2 \\ 2 & -7 & 1 & -2 & 0 \end{bmatrix}$	$\begin{bmatrix} -3 & 0 & -2 & -1 & 6 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ -2 & 0 & -1 & -1 & 4 \\ -3 & 0 & -2 & 0 & 6 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q_{pent}	C-matrix after application to Q_{pent}
(1, 5, 4, 1, 5, 4, 1, 3, 4, 3)	$\begin{bmatrix} 0 & 4 & 1 & 0 & -2 \\ -4 & 0 & 0 & -2 & 7 \\ -1 & 0 & 0 & -1 & 2 \\ 0 & 2 & 1 & 0 & -1 \\ 2 & -7 & -2 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -3 & 0 & -1 & -2 & 6 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ -2 & 0 & -1 & -1 & 4 \\ -3 & 0 & 0 & -2 & 6 \end{bmatrix}$

Table finished

B.3 B- and C-matrices for the proof of Theorem 4.3.11

Mutation sequence	B-matrix after application to Q^a_{pent}	C-matrix after application to Q^a_{pent}
()	$\begin{bmatrix} 0 & -1 & -1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 \\ 1 & 1 & 0 & -a & -1 \\ -1 & 1 & a & 0 & -1 \\ -1 & -1 & 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(1)	$\begin{bmatrix} 0 & 1 & 1 & -1 & -1 \\ -1 & 0 & -1 & 0 & 2 \\ -1 & 1 & 0 & -a+1 & 0 \\ 1 & 0 & a-1 & 0 & -1 \\ 1 & -2 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(1, 3)	$\begin{bmatrix} 0 & 2 & -1 & -1 & -1 \\ -2 & 0 & 1 & -a+1 & 2 \\ 1 & -1 & 0 & a-1 & 0 \\ 1 & a-1 & -a+1 & 0 & -1 \\ 1 & -2 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(1, 3, 5)	$\begin{bmatrix} 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 1 & -a+3 & -2 \\ 1 & -1 & 0 & a-1 & 0 \\ 1 & a-3 & -a+1 & 0 & 1 \\ -1 & 2 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 2 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & -1 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q^a_{pent}	C-matrix after application to Q^a_{pent}
(1, 3, 5, 1)	$\begin{bmatrix} 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & 1 & -a+3 & -2 \\ -1 & -1 & 0 & a-1 & 1 \\ -1 & a-3 & -a+1 & 0 & 2 \\ 1 & 2 & -1 & -2 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 2 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 & 0 \end{bmatrix}$
(1, 3, 5, 2)	$\begin{bmatrix} 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & -1 & a-3 & 2 \\ 1 & 1 & 0 & 2 & -2 \\ 1 & -a+3 & -2 & 0 & 1 \\ -1 & -2 & 2 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 2 & -1 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & -1 \end{bmatrix}$
(1, 3, 5, 2, 1)	$\begin{bmatrix} 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & -1 & a-3 & 2 \\ -1 & 1 & 0 & 2 & -1 \\ -1 & -a+3 & -2 & 0 & 2 \\ 1 & -2 & 1 & -2 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 2 & -1 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 & 0 \end{bmatrix}$
(1, 3, 5, 2, 4)	$\begin{bmatrix} 0 & -a+3 & -3 & 1 & 1 \\ a-3 & 0 & -1 & -a+3 & a-1 \\ 3 & 1 & 0 & -2 & 0 \\ -1 & a-3 & 2 & 0 & -1 \\ -1 & -a+1 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 & 0 & -2 & 1 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 1 \\ 2 & 0 & 0 & -1 & 0 \end{bmatrix}$
(1, 5)	$\begin{bmatrix} 0 & -1 & 1 & -1 & 1 \\ 1 & 0 & -1 & 2 & -2 \\ -1 & 1 & 0 & -a+1 & 0 \\ 1 & -2 & a-1 & 0 & 1 \\ -1 & 2 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 2 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & -1 \end{bmatrix}$
(1, 5, 1)	$\begin{bmatrix} 0 & 1 & -1 & 1 & -1 \\ -1 & 0 & 0 & 2 & -1 \\ 1 & 0 & 0 & -a & 0 \\ -1 & -2 & a & 0 & 2 \\ 1 & 1 & 0 & -2 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 2 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & 1 & 0 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q^a_{pent}	C-matrix after application to Q^a_{pent}
(1, 5, 1, 3)	$\begin{bmatrix} 0 & 1 & 1 & -a+1 & -1 \\ -1 & 0 & 0 & 2 & -1 \\ -1 & 0 & 0 & a & 0 \\ a-1 & -2 & -a & 0 & 2 \\ 1 & 1 & 0 & -2 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 2 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$
(5)	$\begin{bmatrix} 0 & -1 & 0 & 2 & -1 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -a & 1 \\ -2 & 0 & a & 0 & 1 \\ 1 & 1 & -1 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & -1 \end{bmatrix}$
(5, 2)	$\begin{bmatrix} 0 & 1 & 0 & 2 & -2 \\ -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -a & 1 \\ -2 & 0 & a & 0 & 1 \\ 2 & -1 & -1 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & -1 \end{bmatrix}$
(5, 2, 3)	$\begin{bmatrix} 0 & 1 & 0 & 2 & -2 \\ -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a & -1 \\ -2 & 0 & -a & 0 & a+1 \\ 2 & -1 & 1 & -a-1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$
(5, 3)	$\begin{bmatrix} 0 & -1 & 0 & 2 & -1 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & a & -1 \\ -2 & 0 & -a & 0 & a+1 \\ 1 & 1 & 1 & -a-1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$
(5, 3, 5)	$\begin{bmatrix} 0 & -1 & 0 & -a+1 & 1 \\ 1 & 0 & 0 & -a-1 & 1 \\ 0 & 0 & 0 & -1 & 1 \\ a-1 & a+1 & 1 & 0 & -a-1 \\ -1 & -1 & -1 & a+1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q^a_{pent}	C-matrix after application to Q^a_{pent}
(5, 3, 5, 1)	$\begin{bmatrix} 0 & 1 & 0 & a-1 & -1 \\ -1 & 0 & 0 & -a-1 & 2 \\ 0 & 0 & 0 & -1 & 1 \\ -a+1 & a+1 & 1 & 0 & -2 \\ 1 & -2 & -1 & 2 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$
(5, 3, 5, 2)	$\begin{bmatrix} 0 & 1 & 0 & -2a & 1 \\ -1 & 0 & 0 & a+1 & -1 \\ 0 & 0 & 0 & -1 & 1 \\ 2a & -a-1 & 1 & 0 & 0 \\ -1 & 1 & -1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 \\ 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$
(5, 3, 5, 2, 1)	$\begin{bmatrix} 0 & -1 & 0 & 2a & -1 \\ 1 & 0 & 0 & -a+1 & -1 \\ 0 & 0 & 0 & -1 & 1 \\ -2a & a-1 & 1 & 0 & 2a \\ 1 & 1 & -1 & -2a & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 2 \\ -2 & 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$
(5, 3, 5, 2, 5)	$\begin{bmatrix} 0 & 2 & 0 & -2a & -1 \\ -2 & 0 & -1 & a+1 & 1 \\ 0 & 1 & 0 & -1 & -1 \\ 2a & -a-1 & 1 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$
(5, 3, 5, 2, 1, 2)	$\begin{bmatrix} 0 & 1 & 0 & a+1 & -2 \\ -1 & 0 & 0 & a-1 & 1 \\ 0 & 0 & 0 & -1 & 1 \\ -a-1 & -a+1 & 1 & 0 & 2a \\ 2 & -1 & -1 & -2a & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 2 \\ -1 & -1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$
(5, 3, 5, 2, 1, 5)	$\begin{bmatrix} 0 & -1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -3a+1 & 1 \\ 1 & 1 & 0 & -1 & -1 \\ 0 & 3a-1 & 1 & 0 & -2a \\ -1 & -1 & 1 & 2a & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 2 & 0 & 0 & -1 \\ 1 & 2 & 0 & 0 & -2 \\ 0 & 3 & 0 & 0 & -2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q^a_{pent}	C-matrix after application to Q^a_{pent}
(2)	$\begin{bmatrix} 0 & 1 & -2 & 0 & 1 \\ -1 & 0 & 1 & 1 & -1 \\ 2 & -1 & 0 & -a & 0 \\ 0 & -1 & a & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(2, 1)	$\begin{bmatrix} 0 & -1 & 2 & 0 & -1 \\ 1 & 0 & -1 & 1 & -1 \\ -2 & 1 & 0 & -a & 2 \\ 0 & -1 & a & 0 & 0 \\ 1 & 1 & -2 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(2, 1, 2)	$\begin{bmatrix} 0 & 1 & 1 & 0 & -2 \\ -1 & 0 & 1 & -1 & 1 \\ -1 & -1 & 0 & -a+1 & 2 \\ 0 & 1 & a-1 & 0 & -1 \\ 2 & -1 & -2 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 & 1 & 1 \\ -1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(2, 1, 5)	$\begin{bmatrix} 0 & -1 & 0 & 0 & 1 \\ 1 & 0 & -3 & 1 & 1 \\ 0 & 3 & 0 & -a & -2 \\ 0 & -1 & a & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 2 & 0 & 0 & -1 \\ 1 & 2 & 0 & 0 & -2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & -1 \end{bmatrix}$
(2, 1, 5, 1)	$\begin{bmatrix} 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & -3 & 1 & 2 \\ 0 & 3 & 0 & -a & -2 \\ 0 & -1 & a & 0 & 0 \\ 1 & -2 & 2 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 2 & 0 & 0 & -1 \\ -1 & 2 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 \end{bmatrix}$
(2, 1, 5, 2)	$\begin{bmatrix} 0 & 1 & -3 & 0 & 1 \\ -1 & 0 & 3 & -1 & -1 \\ 3 & -3 & 0 & -a+3 & 1 \\ 0 & 1 & a-3 & 0 & 0 \\ -1 & 1 & -1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & -2 & 0 & 2 & 1 \\ 3 & -2 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 2 & -1 & 0 & 1 & 0 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q^a_{pent}	C-matrix after application to Q^a_{pent}
(2, 1, 5, 2, 1)	$\begin{bmatrix} 0 & -1 & 3 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 \\ -3 & 0 & 0 & -a+3 & 4 \\ 0 & 1 & a-3 & 0 & 0 \\ 1 & 1 & -4 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -2 & 0 & 0 & 2 & 3 \\ -3 & 1 & 0 & 2 & 3 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -2 & 1 & 0 & 1 & 2 \end{bmatrix}$
(2, 1, 5, 2, 5)	$\begin{bmatrix} 0 & 2 & -3 & 0 & -1 \\ -2 & 0 & 2 & -1 & 1 \\ 3 & -2 & 0 & -a+3 & -1 \\ 0 & 1 & a-3 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & -1 & 0 & 2 & -1 \\ 3 & -2 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 2 & -1 & 0 & 1 & 0 \end{bmatrix}$
(2, 1, 5, 2, 1, 2)	$\begin{bmatrix} 0 & 1 & 3 & -1 & -2 \\ -1 & 0 & 0 & 1 & 1 \\ -3 & 0 & 0 & -a+3 & 4 \\ 1 & -1 & a-3 & 0 & 0 \\ 2 & -1 & -4 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -2 & 0 & 0 & 2 & 3 \\ -2 & -1 & 0 & 2 & 3 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & -1 & 0 & 1 & 2 \end{bmatrix}$
(2, 1, 5, 2, 1, 5)	$\begin{bmatrix} 0 & -1 & -1 & 0 & 1 \\ 1 & 0 & -4 & -1 & 1 \\ 1 & 4 & 0 & -a+3 & -4 \\ 0 & 1 & a-3 & 0 & 0 \\ -1 & -1 & 4 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 3 & 0 & 2 & -3 \\ 0 & 4 & 0 & 2 & -3 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 3 & 0 & 1 & -2 \end{bmatrix}$
(2, 1, 5, 2, 1, 5, 1)	$\begin{bmatrix} 0 & 1 & 1 & 0 & -1 \\ -1 & 0 & -4 & -1 & 2 \\ -1 & 4 & 0 & -a+3 & -3 \\ 0 & 1 & a-3 & 0 & 0 \\ 1 & -2 & 3 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 3 & 0 & 2 & -2 \\ 0 & 4 & 0 & 2 & -3 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 3 & 0 & 1 & -2 \end{bmatrix}$
(2, 1, 5, 2, 1, 5, 2)	$\begin{bmatrix} 0 & 1 & -5 & -1 & 1 \\ -1 & 0 & 4 & 1 & -1 \\ 5 & -4 & 0 & -a+3 & 0 \\ 1 & -1 & a-3 & 0 & 1 \\ -1 & 1 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 4 & -3 & 0 & 2 & 0 \\ 4 & -4 & 0 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 3 & -3 & 0 & 1 & 1 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q^a_{pent}	C-matrix after application to Q^a_{pent}
(2, 5)	$\begin{bmatrix} 0 & 2 & -2 & 0 & -1 \\ -2 & 0 & 1 & 1 & 1 \\ 2 & -1 & 0 & -a & 0 \\ 0 & -1 & a & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 \end{bmatrix}$
(2, 5, 2)	$\begin{bmatrix} 0 & -2 & 0 & 2 & 1 \\ 2 & 0 & -1 & -1 & -1 \\ 0 & 1 & 0 & -a & 0 \\ -2 & 1 & a & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 1 & 0 \end{bmatrix}$
(2, 5, 2, 3)	$\begin{bmatrix} 0 & -2 & 0 & 2 & 1 \\ 2 & 0 & 1 & -a-1 & -1 \\ 0 & -1 & 0 & a & 0 \\ -2 & a+1 & -a & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$
(3)	$\begin{bmatrix} 0 & -1 & 1 & -a+1 & 0 \\ 1 & 0 & 1 & -a-1 & 0 \\ -1 & -1 & 0 & a & 1 \\ a-1 & a+1 & -a & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(3, 1)	$\begin{bmatrix} 0 & 1 & -1 & a-1 & 0 \\ -1 & 0 & 2 & -a-1 & 0 \\ 1 & -2 & 0 & 1 & 1 \\ -a+1 & a+1 & -1 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(3, 1, 3)	$\begin{bmatrix} 0 & -1 & 1 & a-1 & 0 \\ 1 & 0 & -2 & -a+1 & 2 \\ -1 & 2 & 0 & -1 & -1 \\ -a+1 & a-1 & 1 & 0 & -1 \\ 0 & -2 & 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & -1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q^a_{pent}	C-matrix after application to Q^a_{pent}
(3, 5)	$\begin{bmatrix} 0 & -1 & 1 & -a+1 & 0 \\ 1 & 0 & 1 & -a-1 & 0 \\ -1 & -1 & 0 & a+1 & -1 \\ a-1 & a+1 & -a-1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$
(3, 2)	$\begin{bmatrix} 0 & 1 & 1 & -2a & 0 \\ -1 & 0 & -1 & a+1 & 0 \\ -1 & 1 & 0 & -1 & 1 \\ 2a & -a-1 & 1 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 \\ 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(3, 2, 3)	$\begin{bmatrix} 0 & 2 & -1 & -2a & 1 \\ -2 & 0 & 1 & a & 0 \\ 1 & -1 & 0 & 1 & -1 \\ 2a & -a & -1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 \\ 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(3, 2, 3, 5)	$\begin{bmatrix} 0 & 2 & 0 & -2a & -1 \\ -2 & 0 & 1 & a & 0 \\ 0 & -1 & 0 & 1 & 1 \\ 2a & -a & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix}$
(3, 2, 1)	$\begin{bmatrix} 0 & -1 & -1 & 2a & 0 \\ 1 & 0 & -1 & -a+1 & 0 \\ 1 & 1 & 0 & -2a-1 & 1 \\ -2a & a-1 & 2a+1 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 1 & 1 & 0 & 0 \\ -1 & 0 & 2 & 0 & 0 \\ -2 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(3, 2, 1, 2)	$\begin{bmatrix} 0 & 1 & -2 & a+1 & 0 \\ -1 & 0 & 1 & a-1 & 0 \\ 2 & -1 & 0 & -2a-1 & 1 \\ -a-1 & -a+1 & 2a+1 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 1 & 0 & 0 \\ -1 & 0 & 2 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q^a_{pent}	C-matrix after application to Q^a_{pent}
(3, 2, 1, 3)	$\begin{bmatrix} 0 & -1 & 1 & -1 & 0 \\ 1 & 0 & 1 & -3a & 0 \\ -1 & -1 & 0 & 2a+1 & -1 \\ 1 & 3a & -2a-1 & 0 & 2a \\ 0 & 0 & 1 & -2a & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 2 & -1 & 0 & 1 \\ 1 & 2 & -2 & 0 & 2 \\ 0 & 3 & -2 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(3, 2, 1, 3, 1)	$\begin{bmatrix} 0 & 1 & -1 & 1 & 0 \\ -1 & 0 & 2 & -3a & 0 \\ 1 & -2 & 0 & 2a & -1 \\ -1 & 3a & -2a & 0 & 2a \\ 0 & 0 & 1 & -2a & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 2 & -1 & 0 & 1 \\ -1 & 2 & -1 & 0 & 2 \\ 0 & 3 & -2 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(3, 2, 1, 3, 2)	$\begin{bmatrix} 0 & 1 & 1 & -3a-1 & 0 \\ -1 & 0 & -1 & 3a & 0 \\ -1 & 1 & 0 & -4 & -1 \\ 3a+1 & -3a & 4 & 0 & 2a \\ 0 & 0 & 1 & -2a & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & -2 & 1 & 0 & 1 \\ 3 & -2 & 0 & 0 & 2 \\ 3 & -3 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(3, 2, 1, 3, 5)	$\begin{bmatrix} 0 & -1 & 1 & -1 & 0 \\ 1 & 0 & 1 & -3a & 0 \\ -1 & -1 & 0 & 1 & 1 \\ 1 & 3a & -1 & 0 & -2a \\ 0 & 0 & -1 & 2a & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 2 & 0 & 0 & -1 \\ 1 & 2 & 0 & 0 & -2 \\ 0 & 3 & 0 & 0 & -2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix}$
(3, 2, 1, 3, 1, 5)	$\begin{bmatrix} 0 & 1 & -1 & 1 & 0 \\ -1 & 0 & 2 & -3a & 0 \\ 1 & -2 & 0 & 0 & 1 \\ -1 & 3a & 0 & 0 & -2a \\ 0 & 0 & -1 & 2a & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 2 & 0 & 0 & -1 \\ -1 & 2 & 1 & 0 & -2 \\ 0 & 3 & 0 & 0 & -2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix}$
(3, 2, 1, 3, 2, 1)	$\begin{bmatrix} 0 & -1 & -1 & 3a+1 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 1 & 1 & 0 & -4a & -1 \\ -3a-1 & 1 & 4a & 0 & 2a \\ 0 & 0 & 1 & -2a & 0 \end{bmatrix}$	$\begin{bmatrix} -2 & 0 & 3 & 0 & 1 \\ -3 & 1 & 3 & 0 & 2 \\ -3 & 0 & 4 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q^a_{pent}	C-matrix after application to Q^a_{pent}
(3, 2, 1, 3, 2, 3)	$\begin{bmatrix} 0 & 2 & -1 & -3a-1 & 0 \\ -2 & 0 & 1 & 2a+1 & -1 \\ 1 & -1 & 0 & a-1 & 1 \\ 3a+1 & -2a-1 & -a+1 & 0 & 2a \\ 0 & 1 & -1 & -2a & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & -1 & -1 & 0 & 1 \\ 3 & -2 & 0 & 0 & 2 \\ 3 & -2 & -1 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(3, 2, 1, 3, 2, 5)	$\begin{bmatrix} 0 & 1 & 1 & -3a-1 & 0 \\ -1 & 0 & -1 & 3a & 0 \\ -1 & 1 & 0 & -3a+1 & 1 \\ 3a+1 & -3a & 3a-1 & 0 & -2a \\ 0 & 0 & -1 & 2a & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & -2 & 2 & 0 & -1 \\ 3 & -2 & 2 & 0 & -2 \\ 3 & -3 & 3 & 0 & -2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix}$
(3, 2, 1, 3, 1, 5, 3)	$\begin{bmatrix} 0 & -1 & 1 & 1 & 0 \\ 1 & 0 & -2 & -3a & 2 \\ -1 & 2 & 0 & 0 & -1 \\ -1 & 3a & 0 & 0 & -2a \\ 0 & -2 & 1 & 2a & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 2 & 0 & 0 & -1 \\ 0 & 2 & -1 & 0 & -1 \\ 0 & 3 & 0 & 0 & -2 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & 0 \end{bmatrix}$
(3, 2, 1, 3, 2, 1, 2)	$\begin{bmatrix} 0 & 1 & -2 & 3a & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 2 & -1 & 0 & -4a & -1 \\ -3a & -1 & 4a & 0 & 2a \\ 0 & 0 & 1 & -2a & 0 \end{bmatrix}$	$\begin{bmatrix} -2 & 0 & 3 & 0 & 1 \\ -2 & -1 & 3 & 0 & 2 \\ -3 & 0 & 4 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(3, 2, 1, 3, 2, 1, 3)	$\begin{bmatrix} 0 & -1 & 1 & -a+1 & -1 \\ 1 & 0 & 1 & -4a-1 & -1 \\ -1 & -1 & 0 & 4a & 1 \\ a-1 & 4a+1 & -4a & 0 & 2a \\ 1 & 1 & -1 & -2a & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 3 & -3 & 0 & 1 \\ 0 & 4 & -3 & 0 & 2 \\ 1 & 4 & -4 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
(3, 2, 1, 3, 2, 1, 5)	$\begin{bmatrix} 0 & -1 & -1 & 3a+1 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 1 & 1 & 0 & -6a & 1 \\ -3a-1 & 1 & 6a & 0 & -2a \\ 0 & 0 & -1 & 2a & 0 \end{bmatrix}$	$\begin{bmatrix} -2 & 0 & 4 & 0 & -1 \\ -3 & 1 & 5 & 0 & -2 \\ -3 & 0 & 6 & 0 & -2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix}$

Continued on next page

Mutation sequence	B-matrix after application to Q^a_{pent}	C-matrix after application to Q^a_{pent}
(3, 2, 1, 3, 2, 3, 5)	$\begin{bmatrix} 0 & 2 & -1 & -3a-1 & 0 \\ -2 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & a-1 & -1 \\ 3a+1 & -1 & -a+1 & 0 & -2a \\ 0 & -1 & 1 & 2a & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 & -1 & 0 & -1 \\ 3 & 0 & 0 & 0 & -2 \\ 3 & 0 & -1 & 0 & -2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 \end{bmatrix}$
(3, 2, 1, 3, 2, 5, 3)	$\begin{bmatrix} 0 & 2 & -1 & -3a-1 & 1 \\ -2 & 0 & 1 & 1 & 0 \\ 1 & -1 & 0 & 3a-1 & -1 \\ 3a+1 & -1 & -3a+1 & 0 & a-1 \\ -1 & 0 & 1 & -a+1 & 0 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 & -2 & 0 & 1 \\ 3 & 0 & -2 & 0 & 0 \\ 3 & 0 & -3 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 \end{bmatrix}$
(3, 2, 1, 3, 1, 5, 3, 1)	$\begin{bmatrix} 0 & 1 & -1 & -1 & 0 \\ -1 & 0 & -1 & -3a+1 & 2 \\ 1 & 1 & 0 & 0 & -1 \\ 1 & 3a-1 & 0 & 0 & -2a \\ 0 & -2 & 1 & 2a & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 2 & 0 & 0 & -1 \\ 0 & 2 & -1 & 0 & -1 \\ 0 & 3 & 0 & 0 & -2 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 & 0 \end{bmatrix}$

Table finished

C

Code

The computations that led to the results of this work have almost exclusively been carried out in *Sage*[1] (or *SageMath*). Since we fully support this particular open-source innitiative many researchers have taken and are actively taking part in, we provide working code for quantisations as discussed in Section 3.2 and green permissible periods of Section 4.4.2. All functions are based on Sage 7.3 and are not included in the current distribution.[2] In both cases we further show how these functions can be used in their respective description headers and hope they may serve others in future research.

C.1 Quantisation

We provide three main and two auxiliary functions as computational methods for quantisations of Section 3.2. These functions are based on the class `ClusterQuiver` and their respective objectives are:

- `is_quantizable`: check if a `ClusterQuiver` is quantisable, i.e. compare the rank and the number of columns of the associated (extended) exchange matrix as imposed by Theorem 3.2.4,
- `quantization_partner`: produce an integer matrix Λ as in Theorem 3.2.4 for a

[1] See the official website for more information: http://www.sagemath.org
[2] The current stable version on 31 March 2017 is Sage 7.6.

given `ClusterQuiver`,

- `_block_two_frozen`: auxiliary function to produce matrices M as in Lemma 3.2.8 for extended exchange matrices of dimension $n+2$,
- `_blocks_skewsymmetric`: auxiliary function for constructing matrices $\mathfrak{M}_{E(i,j)}$ as in Theorem 3.2.11 for extended exchange matrices whose principal part is skew-symmetric,
- `quantization_homogeneous_blocks`: for a given `ClusterQuiver`, create a list of matrices satisfying the homogeneous equation (3.9).

```
def is_quantizable(self):
    r"""
    Returns true if ``self`` is quantizable.

    EXAMPLES::

        sage: ClusterQuiver(['A',4]).is_quantizable()
        True

        sage: ClusterQuiver(['A',[2,1],1]).is_quantizable()
        False

        sage: ClusterQuiver(['A',[2,1],1]).principal_extension().
           is_quantizable()
        True
    """
    return self._M.rank() == self._n

def quantization_partner(self):
    r"""
    Returns the compatible matrix Lambda if ``self`` is quantizable.

    EXAMPLES::

        sage: ClusterQuiver(['A',4]).quantization_partner()
        [ 0  1  0 -1]
        [-1  0  0  0]
        [ 0  0  0  1]
        [ 1  0 -1  0]

        sage: ClusterQuiver(['A',[2,1],1]).principal_extension().
           quantization_partner()
        [ 0  0  0 -1  0  0]
        [ 0  0  0  0 -1  0]
        [ 0  0  0  0  0 -1]
        [ 1  0  0  0  1  1]
        [ 0  1  0 -1  0 -1]
        [ 0  0  1 -1  1  0]
    """
    if not self.is_quantizable():
        raise ValueError("The cluster quiver is not quantizable.")
    else:
        Mrows = self._m + self._n
        Mcols = self._n
        L = matrix( QQ, Mrows, Mrows )
```

```
        M = matrix( QQ, Mrows, Mrows )
        B = VectorSpace( QQ, Mrows ).basis()
        maxInd = Combinations( range( Mrows ), Mrows - Mcols ).list()
        D = diagonal_matrix(QQ, self._M[:Mcols,:Mcols].is_skew_symmetrizable(return_diag=True, positive=True))
        M[:, :Mcols] = self._M
        for i in range( len( maxInd ) ):
            for j in range( Mrows - Mcols ):
                M[ :, range( Mcols, Mrows )[ j ] ] = B[ maxInd[ i ][ j ] ]
            if M.rank() ==  M.ncols():
                break
        L[:Mcols,:Mcols] = D*self._M[:Mcols,:Mcols]
        L[:Mcols,Mcols:Mrows] = D*M[:Mcols,Mcols:Mrows]
        L[Mcols:Mrows, :Mcols] = -M[:Mcols,Mcols:Mrows].transpose() * D
        L = M.transpose().inverse() * L * M.inverse()
        if L not in MatrixSpace(ZZ,Mrows,Mrows):
            DenomRationalEntries = list()
            for k in range(Mrows):
                for l in range(Mrows):
                    if L[k][l] not in ZZ:
                        DenomRationalEntries.append(L[k][l].denominator())
            L *= lcm(DenomRationalEntries)
        return L

def _block_two_frozen( self, L ):
    r"""
    Returns a matrix A for which N.transpose()*A is the zero matrix. The input parameter L is required to be a list of row indices of length n+2 and N is the submatrix of ``self``._M whose rows are indexed by L.

    EXAMPLES::

        sage: ClusterQuiver(['A',2]).principal_extension()._block_two_frozen( range(4) )
        [ 0 -1 -1  0]
        [ 1  0  0 -1]
        [ 1  0  0 -1]
        [ 0  1  1  0]

        sage: ClusterQuiver(['A',[2,1],1]).principal_extension()._block_two_frozen(range(5))
        [ 0  0  1 -1  1]
        [ 0  0  1 -1  1]
        [-1 -1  0  1 -1]
        [ 1  1 -1  0  0]
        [-1 -1  1  0  0]
    """
    if len( L ) <> (self._n+2) :
        raise ValueError("The list of given row indices is not of length n+2.")
    elif not self.is_quantizable():
        raise ValueError("The cluster quiver is not quantizable.")
    else:
        mat = self._M[ L , :]
        minorList = mat.minors( self._n )
        minorBlock = matrix( QQ, self._n+2, self._n+2 )
        k = len( minorList )-1
        for i in range( self._n + 1 ):
            for j in range( self._n + 2 ):
                if ( i < j and k > -1 ):
```

```
                    minorBlock[i,j] = (-1)**(i+j) * minorList[k]
                    minorBlock[j,i] = -minorBlock[i,j]
                    k -= 1
        return minorBlock

def _blocks_skewsymmetric( self ):
    r"""
    Returns a list of linearly independent matrices A for which ``self``._M.transpose()*A equals a zero matrix, given the principal part ``self`` is skew-symmetric.

    EXAMPLES::

        sage: ClusterQuiver(['A',3]).principal_extension()._blocks_skewsymmetric()
        [
        [ 0  0 -1  0  1  0]  [ 0  1  0  1  0  1]  [ 0 -1  0 -1  0 -1]
        [ 0  0  0  0  0  0]  [-1  0  1  0  0  0]  [ 1  0  0  0 -1  0]
        [ 1  0  0  0 -1  0]  [ 0 -1  0 -1  0 -1]  [ 0  0  0  0  0  0]
        [ 0  0  0  0  0  0]  [-1  0  1  0  0  0]  [ 1  0  0  0 -1  0]
        [-1  0  1  0  0  0]  [ 0  0  0  0  0  0]  [ 0  1  0  1  0  1]
        [ 0  0  0  0  0  0], [-1  0  1  0  0  0], [ 1  0  0  0 -1  0]
        ]

        sage: ClusterQuiver(['A',[2,1],1]).principal_extension()._blocks_skewsymmetric()
        [
        [ 0  0  1 -1  1  0]  [ 0 -1  0  1  0  1]  [ 0 -1  0  1  0  1]
        [ 0  0  1 -1  1  0]  [ 1  0 -1  1  0  1]  [ 1  0  0  0  1  1]
        [-1 -1  0  1 -1  0]  [ 0  1  0 -1  0 -1]  [ 0  0  0  0  0  0]
        [ 1  1 -1  0  0  0]  [-1 -1  1  0  0  0]  [-1  0  0  0 -1 -1]
        [-1 -1  1  0  0  0]  [ 0  0  0  0  0  0]  [ 0 -1  0  1  0  1]
        [ 0  0  0  0  0  0], [-1 -1  1  0  0  0], [-1 -1  0  1 -1  0]
        ]
    """
    matInd = Combinations( range( self._n+self._m ), self._n ).list()
    for i in range( len( matInd ) ):
      if self._M[matInd[i],:].rank() == self._n:
          matInd = set( matInd[i] )
          break
    Eij = Combinations( set( range( self._M.nrows() ) ).difference( matInd ), 2 ).list()
    compList = list( matrix(QQ, self._n+self._m, self._n+self._m ) for r in range( len( Eij ) ) )
    for r in range( len( Eij ) ):
        compList[r][ list( matInd.union( set( Eij[r] ) ) ), list( matInd.union( set( Eij[r] ) ) ) ] = self._block_two_frozen( list( matInd.union( set( Eij[r] ) ) ) )
    return( compList )


def quantization_homogeneous_blocks(self):
    r"""
    Returns a list of linearly independent matrices A for which ``self``._M.transpose()*A equals a zero matrix.

    EXAMPLES::

        sage: ClusterQuiver(['A',3]).principal_extension().quantization_homogeneous_blocks()
```

```
        [
        [ 0  0 -1  0  1  0]  [ 0  1  0  1  0  1]  [ 0 -1  0 -1  0 -1]
        [ 0  0  0  0  0  0]  [-1  0  1  0  0  0]  [ 1  0  0  0 -1  0]
        [ 1  0  0  0 -1  0]  [ 0 -1  0 -1  0 -1]  [ 0  0  0  0  0  0]
        [ 0  0  0  0  0  0]  [-1  0  1  0  0  0]  [ 1  0  0  0 -1  0]
        [-1  0  1  0  0  0]  [ 0  0  0  0  0  0]  [ 0  1  0  1  0  1]
        [ 0  0  0  0  0  0], [-1  0  1  0  0  0], [ 1  0  0  0 -1  0]
        ]

        sage: M = copy(ClusterQuiver(['A',3])._M)
        sage: M[0,1]=2
        sage: ClusterQuiver(M).principal_extension().
           quantization_homogeneous_blocks()
        [
        [ 0  0  1 -1  1  0]  [ 0 -1  0  1  0  1]  [ 0 -1  0  1  0  1]
        [ 0  0  1 -1  1  0]  [ 1  0 -1  1  0  1]  [ 1  0  0  0  1  1]
        [-1 -1  0  1 -1  0]  [ 0  1  0 -1  0 -1]  [ 0  0  0  0  0  0]
        [ 1  1 -1  0  0  0]  [-1 -1  1  0  0  0]  [-1  0  0  0 -1 -1]
        [-1 -1  1  0  0  0]  [ 0  0  0  0  0  0]  [ 0 -1  0  1  0  1]
        [ 0  0  0  0  0  0], [-1 -1  1  0  0  0], [-1 -1  0  1 -1  0]
        ]
    """
    if not self.is_quantizable():
        raise ValueError("The cluster quiver is not quantizable.")
    elif self._M.nrows() - self._n == 1:
        raise ValueError("The quantization for this cluster quiver is
           essentially unique.")
    elif self._M.nrows() - self._n == 2:
        return self._block_two_frozen( range(self._n+2) )
    elif self._M[:self._n,:self._n].is_skew_symmetric():
        return( self._blocks_skewsymmetric() )
    # If the principal part B is not skew-symmetric, construct results for
      DB and modify accordingly
    elif self._M[:self._n,:self._n].is_skew_symmetrizable(positive=True):
        D = diagonal_matrix( QQ, self._M[:self._n,:self._n].
           is_skew_symmetrizable(return_diag=True, positive=True) )
        skewB = D * self._M[ :self._n,:]
        skewCQ = ClusterQuiver( skewB.stack( self._M[ self._n:, : ] ) )
        blockList = skewCQ._blocks_skewsymmetric()
        for i in range( len( blockList ) ):
            blockList[ i ][:self._n,:] = D.transpose() * blockList[ i ][:
               self._n,:]
        return( blockList )
```

In order to attach the above functions to the class `ClusterQuiver`, we simply use the following few lines:

```
from sage.combinat.cluster_algebra_quiver.quiver import ClusterQuiver
ClusterQuiver.is_quantizable = is_quantizable
ClusterQuiver.quantization_partner = quantization_partner
ClusterQuiver._block_two_frozen = _block_two_frozen
ClusterQuiver._blocks_skewsymmetric = _blocks_skewsymmetric
ClusterQuiver.quantization_homogeneous_blocks =
   quantization_homogeneous_blocks
```

C.2 Green permissible periods

Our implementation of methods used for experimenting with green permissible periods of Section 4.4.2 is rather extensive. It builds on features of the classes `ClusterQuiver` and `ClusterSeed`. In the course of our experiments, we came across time and space limitations when long mutation sequences were applied to objects of the class `ClusterSeed`. In particular, the currently present realisation of computing cluster variables and coefficients slows such efforts down. Hence we decided to translate the necessary features from `ClusterSeed` to `ClusterQuiver` and work exclusively with objects of the latter class.

First, we extend the class `ClusterQuiver` to `ClusterQuiverExtended` in order to encorporate three additional attributes: `_track_mut` which is a flag indicating whether mutation sequences are supposed to be tracked, `_mut_path` to store such sequences since the first initialisation onwoards and `_is_principal` to indicate if a given `ClusterQuiverExtended` stems from a principal extension.

Second, we create a new class `OrientedExchangeGraph`, derived from `DiGraph`, which stores not only the oriented exchange graph but also detected isomorphisms and green permissible periods therein.
The main method for this class is `find_green_permissible_periods` which runs from a given node index up the branch of this node until it reaches the unique source or detects the index of the start of a green permissible period.

Third, a number of functions are provided for the class `ClusterQuiverExtended`, in most parts translating and/or extending methods of the class `ClusterSeed`. The most advanced of these functions is `oriented_exchange_graph`, which produces an object of the class `OrientedExchangeGraph` and allows the user to specify a range of optional inputs.

Let us start by defining the class `OrientedExchangeGraph`.

```
from sage.combinat.cluster_algebra_quiver.quiver import ClusterQuiver
from sage.combinat.cluster_algebra_quiver.quiver_mutation_type import
   QuiverMutationType, QuiverMutationType_Irreducible,
   QuiverMutationType_Reducible, _edge_list_to_matrix
from sage.combinat.cluster_algebra_quiver.mutation_class import
   _principal_part, _digraph_mutate, _matrix_to_digraph, _dg_canonical_form,
    _mutation_class_iter, _digraph_to_dig6, _dig6_to_matrix

class ClusterQuiverExtended(ClusterQuiver):
    r"""
    The *quiver* associated to an *exchange matrix*.

    INPUT:

    - ``data`` -- can be any of the following::

        * any type of input data for a ClusterQuiver-object
        * ClusterQuiverExtended

    - ``frozen`` -- (default:``None``) sets the number of frozen variables
```

```
        if the input type is a DiGraph, it is ignored otherwise.


    EXAMPLES::

        sage: Q = ClusterQuiverExtended(['G',2, (3,1)]); Q
        Quiver on 4 vertices of type ['G', 2, [1, 3]]

        sage: Q = ClusterQuiverExtended( matrix
            ([[0,1,-1],[-1,0,1],[1,-1,0],[1,2,3]]) ); Q
        Quiver on 4 vertices with 1 frozen vertex
        sage: Q._mut_path; Q._is_principal
        []
        False

        sage: Q = ClusterQuiverExtended(['A',4]).principal_extension(); Q.
            _is_principal
        True
    """
    def __init__(self, data, frozen=None ):
        from sage.matrix.matrix import Matrix
        super(ClusterQuiverExtended,self).__init__(data,frozen=frozen)
        if isinstance(data, ClusterQuiverExtended):
            self._track_mut = data._track_mut
            self._mut_path = copy(data._mut_path)
            self._is_principal = data._is_principal
        if isinstance(data, ClusterQuiver) or isinstance(data, Matrix):
            self._track_mut = True
            self._mut_path = []

            if self._M[self._n:,:] == identity_matrix(self._n):
                self._is_principal = True
            else:
                self._is_principal = False
        elif isinstance(data, DiGraph):
            self._track_mut = True
            self._mut_path = []

            if self._M[self._n:self._M.nrows(),:] == identity_matrix(self._n
                ):
                self._is_principal = True
            else:
                self._is_principal = False
```

Next, we define the class `OrientedExchangeGraph` as a subclass of `DiGraph` and provide the following methods for objects of this class: `find_green_permissible_periods`, `get_isomorphisms`, `get_green_permissible_periods` and `get_number_isoclasses`:

```
from sage.graphs.generic_graph import GenericGraph
class OrientedExchangeGraph(DiGraph):
    r"""
    The *oriented exchange graph* associated to an *exchange matrix* that is
        mutation equivalent to a principally extended cluster seed.
    Primarily intended to be used in conjunction with the method ``
        oriented_exchange_graph`` of the class ClusterQuiverExtended.

    EXAMPLES::

```

```
        sage: CQ = ClusterQuiverExtended(['A',2]).principal_extension()
        sage: OEG = CQ.oriented_exchange_graph()
        sage: OEG. get_number_isoclasses()
        5
        sage: OEG.get_green_permissible_periods()
        []
        sage: OEG.get_isomorphisms()
        [(4, 5)]

        sage: OEG = CQ.oriented_exchange_graph(method="first_maximal_green_sequence")
        The first maximal green sequence has been found.
        sage: OEG. get_number_isoclasses()
        4

        sage: CQ = ClusterQuiverExtended(matrix([[0,2,-1],[-2,0,2],[1,-2,0]])).principal_extension()
        sage: OEG = CQ.oriented_exchange_graph(depth=10, exclude_non_permissibles = False)
        UserWarning: Running time and memory usage for infinite type can be enormous
        The maximal depth has been reached.
        sage: OEG.get_number_isoclasses()
        53
        sage: OEG.get_green_permissible_periods()
        [[7, 25], [13, 32], [21, 38], [23, 39], [27, 41], [34, 46], [42, 51]]
        sage: OEG.get_isomorphisms()
        [(11, 12), (5, 17), (16, 18), (21, 29), (28, 30), (36, 45), (43, 50), (52, 55)]
    """

    def __init__(self,data):
        msg = ''
        GenericGraph.__init__(self)
        from sage.structure.element import is_Matrix

        data_structure = "sparse"

        from sage.graphs.base.sparse_graph import SparseGraphBackend
        if data_structure in ["sparse", "static_sparse"]:
            CGB = SparseGraphBackend
        else:
            raise ValueError("data_structure must be equal to 'sparse', "
                             "'static_sparse' or 'dense'")
        self._backend = CGB(0, directed=True)

        if isinstance(data,DiGraph):
            if not data.is_directed_acyclic() or data.has_loops() or data.has_multiple_edges():
                raise ValueError("The input digraph may not have loops, cycles or multiedges.")
            self.add_vertices(data.vertex_iterator())
            self.add_edges(data.edge_iterator())
            self.name(data.name())
            self._isomorphism = []
            self._green_permissible_period = []

        elif isinstance(data,OrientedExchangeGraph):
            self.add_vertices(data.vertex_iterator())
```

```
            self.add_edges(data.edge_iterator())
            self.name(data.name())
            self._isomorphism = data._isomorphism
            self._green_permissible_period = data._green_permissible_period
        else:
            raise ValueError("The input data was not recognized.")

    def find_green_permissible_periods(self,end,use_isomorphism=True):
        r"""
        Applies a reordering of the mutable vertices of ``self`` either to itself or returns the resulting ClusterQuiver.
        Primarily intended to be used in conjunction with the method ``oriented_exchange_graph`` of the class ClusterQuiverExtended.

        INPUT:

        - ``end`` -- (fixed) index of a node that will be compared to all other nodes throughout the search
        - ``use_isomorphism`` -- (default: True) if this flag is set True, at each step all possible cluster quivers isomomorphic to the one at the current start-position are produced and test
        """
        if not all( [ isinstance(self.get_vertex(vertex_id), ClusterQuiverExtended) for vertex_id in self.vertex_iterator() ] ):
            raise TypeError('a ClusterQuiverExtended needs to be attached to each vertex of the OrientedExchangeGraph')
        import itertools

        end_cq = self.get_vertex(end)

        for path in self.all_paths_iterator( ending_vertices=[end], simple=True ):
            start = path[0]
            start_cq = self.get_vertex(start)

            # Determine the mutation path between the two ClusterQuiverExtended
            mut_path_difference = end_cq._mut_path[ len(start_cq._mut_path):len(end_cq._mut_path) ]
            set_direction_in_path = set(mut_path_difference)
            list_direction_in_path = set_direction_in_path.list()

            if len(list_direction_in_path) > 1:
                # Determine those indices which do not contribute to the in-between mutation path
                list_direction_not_in_path = list(set(range(end_cq._n)).difference(set_direction_in_path))

                # If isomorphism classes of ClusterQuiverExtended are to be considered
                if use_isomorphism:
                    # Construct all reorderings of the mutable part of the start ClusterQuiverExtended indexed by list_direction_in_path
                    reorder_list = list(itertools.permutations(list_direction_in_path,len(list_direction_in_path)))
                    reorder_dict_list = []
                    for i in range(len(reorder_list)):
```

```
                reorder_dict = {}
                for j in range(len(list_direction_in_path)):
                    reorder_dict[list_direction_in_path[j]] = reorder_list[i][j]
                reorder_dict_list.append(reorder_dict)

            start_subgraph = start_cq.digraph().subgraph(list_direction_in_path)
            start_subgraph_list = [start_subgraph.relabel(sigma, inplace=False) for sigma in reorder_dict_list]
        else:
            start_subgraph_list = [start_cq.digraph().subgraph(list_direction_in_path)]

        end_subgraph = end_cq.digraph().subgraph(list_direction_in_path)
        # Check if end ClusterQuiverExtended equals the start or, for use_isomorphism=True, if it equals any of the reorderings of start as constructed above
        check_list = [end_subgraph==start_subdig for start_subdig in start_subgraph_list]

        list_period_condition_flag = []
        check_indices = [j for j, entry in enumerate(check_list) if entry]

        # For all located indices of equal (resp. isomorphic) subquivers indexed by list_direction_in_path ...
        if check_indices:
            start_graph_dict = {}
            if use_isomorphism:
                for i in check_indices:
                    start_graph_dict[i] = start_cq.digraph().relabel(reorder_dict_list[i], inplace=False)
            else:
                start_graph_dict[0] = start_cq.digraph()
            for i in check_indices:
                period_condition_flag = True
                # Ignore case if subgraphs have different number of edges
                if len(start_graph_dict[i].edges()) <> len(end_cq.digraph().edges()):
                    period_condition_flag = False
                else:
                    # Check if the (multi-)edges increase 'sign-coherently'
                    for e in start_graph_dict[i].edge_iterator():
                        if end_cq.digraph().has_edge( (e[0],e[1]) ):
                            if (e[2][0] == 0 and end_cq.digraph().edge_label(e[0],e[1])[0] <> 0 ) or (e[2][0] > 0 and e[2][0] > end_cq.digraph().edge_label(e[0],e[1])[0]) or (e[2][0] < 0 and e[2][0] < end_cq.digraph().edge_label(e[0],e[1])[0]):
                                period_condition_flag = False
                                break
                        else:
                                period_condition_flag = False
                                break
                list_period_condition_flag.append(
```

```
                                    period_condition_flag)

                # If a periodic parent is found, append it to the attribute of the class
                if any(list_period_condition_flag):
                    self._green_permissible_period.append([start,end])
                    break

    def get_isomorphisms(self):
        r"""
        Returns a list of tuples of isomorphic ClusterQuiverExtended.
        """
        return self._isomorphism

    def get_green_permissible_periods(self):
        r"""
        Returns a list of green permissible periods detected in the oriented exchange graph.
        """
        return self._green_permissible_period

    def get_number_isoclasses(self):
        r"""
        Returns the number of isomorphism classes of the vertices in the oriented exchange graph.
        """
        return self.order()-len(self._isomorphism)
```

We subsequently need to merge an object of the class `DiGraph` to a second such object. Thus let us define a merge-function.

```
def merge_digraph(self, DiG, merge_index, inplace = True):
    if not is_instance(DiG,DiGraph):
        raise TypeError('The input data is no DiGraph.')
    if not merge_index in self.vertices():
        raise ValueError('The index for the merge hook could not be recognized')

    # Assume that all vertex-labels are integers
    setoff = max(self.vertices())+1
    copyDiG = copy(DiG)

    copyDiG.relabel(lambda i: i+setoff)
    list_outgoing_edge = copyDiG.outgoing_edges(setoff)
    list_outgoing_edge = [ (merge_index, edge[1], edge[2]) for edge in list_outgoing_edge ]
    copyDiG.delete_vertex(setoff)

    if inplace:
        self.add_vertices(copyDiG.vertices())
        self.add_edges(copyDiG.edges())
        self.add_edges(list_outgoing_edge)
    else:
        copySelf = copy(self)
        copySelf.add_vertices(copyDiG.vertices())
        copySelf.add_edges(copyDiG.edges())
        copySelf.add_edges(list_outgoing_edge)
        return copySelf
```

In order to append this method to the class DiGraph, we execute the following line.

```
DiGraph.merge_digraph = merge_digraph
```

And lastly, we define a number of methods for the class ClusterQuiverExtended:

- track_mutations: if the flag _track_mut was previously inactive, set it true an initialise the attribute _mut_path,
- mutations: return the value of _mut_path if mutation sequences are being tracked,
- mutate: mutate a ClusterQuiverExtended and, in particular, append the mutation to _mut_path,
- get_permissible_vertices: return a list of indices of permissible vertices,
- get_green_vertices: return a list of indices of green vertices,
- get_green_permissible_vertices: return a list of indices of green permissible vertices,
- reorder: applys a permutation to the mutable vertices,
- oriented_exchange_graph: compute the associated OrientedExchangeGraph object.

```

def mutate(self, data, inplace=True):
    r"""
    Extended version from ClusterQuiver.

    EXAMPLES::

        sage: CQ = ClusterQuiverExtended(['A',4]); CQ.b_matrix()
        [ 0  1  0  0]
        [-1  0 -1  0]
        [ 0  1  0  1]
        [ 0  0 -1  0]

        sage: CQ.mutate(0); CQ.b_matrix()
        [ 0 -1  0  0]
        [ 1  0 -1  0]
        [ 0  1  0  1]
        [ 0  0 -1  0]

        sage: T = CQ.mutate(0, inplace=False); T
        Quiver on 4 vertices of type ['A', 4]

        sage: CQ.mutate(0); CQ == T
        True
    """

    n = self._n
```

```
    m = self._m
    dg = self._digraph
    V = list(range(n))

    # If we get a string, execute as a function
    if isinstance(data, str):
        data = getattr(self, data)()

    # If we get a function, execute it
    if hasattr(data, '__call__'):
        # Function should return either integer or sequence
        data = data(self)

    if data is None:
        raise ValueError('Not mutating: No vertices given.')

    if data in V:
        seq = [data]
    else:
        seq = data
    if isinstance(seq, tuple):
        seq = list( seq )
    if not isinstance(seq, list):
        raise ValueError('The quiver can only be mutated at a vertex or at a
            sequence of vertices')
    if not isinstance(inplace, bool):
        raise ValueError('The second parameter must be boolean.  To mutate
          at a sequence of length 2, input it as a list.')
    if any( v not in V for v in seq ):
        v = filter( lambda v: v not in V, seq )[0]
        raise ValueError('The quiver cannot be mutated at the vertex %s'%v)

    # Utilize the mutation-function for DiGraphs
    for v in seq:
        dg = _digraph_mutate( dg, v, n, m )
    # Convert edge-list to adjacency matrix
    M = _edge_list_to_matrix( dg.edge_iterator(), n, m )

    if self._track_mut:
        mut_path = copy(self._mut_path)
        for v in seq:
            if len(mut_path) == 0 or mut_path[len(mut_path)-1] <> v:
                mut_path.append(v)
            else:
                mut_path.pop()

    if inplace:
        self._M = M
        self._M.set_immutable()
        self._digraph = dg
        if self._track_mut:
            self._mut_path = mut_path
    else:
        Q = ClusterQuiverExtended( M )
        Q._mutation_type = self._mutation_type
        if self._track_mut:
            Q._mut_path = mut_path
        return Q

def mutations(self):
```

```
    r"""
    Adopted from ClusterSeed

    EXAMPLES::

        sage: CQ = ClusterQuiverExtended(['A',3])
        sage: CQ.mutations()
        []
        sage: CQ.mutate([0,1,0,2])
        sage: CQ.mutations()
        [0, 1, 0, 2]
    """
    if self._track_mut:
        return copy(self._mut_path)
    else:
        raise ValueError("Not recording mutation sequence. Need to track mutations.")

def track_mutations(self, use=True):
    r"""
    Adopted from ClusterSeed.

    EXAMPLES::

        sage: CQ = ClusterQuiverExtended(['A',4]); CQ.track_mutations(False)
        sage: CQ.mutate(0)
        sage: CQ.mutations()
        Traceback (most recent call last):
        ...
        ValueError: Not recording mutation sequence.  Need to track mutations.
        sage: CQ.track_mutations(True)
        sage: CQ.mutations()
        []
        sage: CQ.mutate(2)
        sage: CQ.mutations()
        [2]
    """
    if self._track_mut <> use:
        self._track_mut = use
        if self._track_mut:
            self._mut_path = []
        else:
            self._mut_path = None

def principal_extension(self, inplace=False):
    r"""
    Adopted from ClusterSeed.

    EXAMPLES::

        sage: CQ = ClusterQuiverExtended([[0,1],[1,2],[2,3],[2,4]]); CQ
        Quiver on 5 vertices

        sage: T = CQ.principal_extension(); T
        Quiver on 10 vertices with 5 frozen vertices

        sage: CQ = ClusterQuiverExtended(['A',4]); CQ.principal_extension().b_matrix()
        [ 0  1  0  0]
```

```
            [-1  0 -1  0]
            [ 0  1  0  1]
            [ 0  0 -1  0]
            [ 1  0  0  0]
            [ 0  1  0  0]
            [ 0  0  1  0]
            [ 0  0  0  1]
        """
        dg = DiGraph( self._digraph )
        dg.add_edges( [(self._n+self._m+i,i) for i in range(self._n)] )
        Q = ClusterQuiverExtended( dg, frozen=self._m+self._n )
        Q._mutation_type = self._mutation_type
        Q._is_principal = True
        if inplace:
            self.__init__(Q)
        else:
            return Q

    def get_green_vertices(self):
        r"""
        Returns list of indices for which the respective column in the C-matrix of ``self`` is non-negative.

        EXAMPLES::

            sage: ClusterQuiverExtended(['A',4]).get_green_vertices()
            Traceback (most recent call last):
            ...
            TypeError: Green vertices are only implemented for principal extensions

            sage: CQ = ClusterQuiverExtended(['A',4]).principal_extension()
            sage: CQ.get_green_vertices()
            [0, 1, 2, 3]
            sage: CQ.mutate(0)
            sage: CQ.get_green_vertices()
            [1, 2, 3]
        """
        if not self._is_principal:
            raise TypeError('Green vertices are only implemented for principal extensions')
        return [ i for (i,v) in enumerate(self._M[self._n:,:].columns()) if any( x > 0 for x in v) ]

    def get_permissible_vertices(self):
        r"""
        Returns list of indices for which the respective row in the B-matrix of ``self`` has no entries bigger than 1.

        EXAMPLES::

            sage: ClusterQuiverExtended(['A',4]).get_permissible_vertices()
            [0, 1, 2, 3]

            sage: CQ = ClusterQuiverExtended(matrix([[0,2,-1],[-2,0,2],[1,-2,0]]))
            sage: CQ.get_permissible_vertices()
            [2]
        """
        return [ i for (i,v) in enumerate(self._M[:self._n,:].rows()) if all( x
```

```
            < 2 for x in v) ]

def get_green_permissible_vertices(self):
    r"""
    Returns list of indices for which the respective column in the C-matrix
       of ``self`` is non-negative and the respective row in the B-matrix of
        ``self`` has no entries bigger than 1.

    EXAMPLES::

        sage: CQ = ClusterQuiverExtended(['A',4]).principal_extension()
        sage: CQ.mutate(0); CQ.get_green_permissible_vertices()
        [1,2,3]

        sage: CQ = ClusterQuiverExtended(matrix
           ([[0,2,-1],[-2,0,2],[1,-2,0]])).principal_extension()
        sage: CQ.get_green_permissible_vertices()
        [2]
        sage: CQ.mutate(2)
        sage: CQ.get_green_permissible_vertices()
        [0, 1]
    """
    return list( set(self.get_green_vertices()).intersection(set(self.
       get_permissible_vertices())) )

def reorder(self, reordering, inplace=True):
    r"""
    Applies a reordering of the mutable vertices of ``self`` either to
       itself or returns the resulting ClusterQuiver.

    EXAMPLES::

        sage: CQ = ClusterQuiverExtended(['A',4])
        sage: CQ.mutate([0,2,3,1,2]); CQ.b_matrix()
        [ 0  1  0  0]
        [-1  0  1  0]
        [ 0 -1  0 -1]
        [ 0  0  1  0]
        sage: CQ.reorder([0,1,3,2])
        sage: CQ.b_matrix()
        [ 0  1  0  0]
        [-1  0  0  1]
        [ 0  0  0  1]
        [ 0 -1 -1  0]

        sage: CQ = ClusterQuiverExtended(matrix
           ([[0,2,-1],[-2,0,2],[1,-2,0]])).principal_extension()
        sage: CQ.mutate([0,1]); CQ.b_matrix()
        [ 0  2  1]
        [-2  0  0]
        [-1  0  0]
        [ 3 -2  0]
        [ 2 -1  0]
        [ 0  0  1]
        sage: T = CQ.reorder([2,1,0], inplace=False)
        sage: T.b_matrix()
        [ 0  0 -1]
        [ 0  0 -2]
        [ 1  2  0]
        [ 0 -2  3]
```

```
        [ 0 -1  2]
        [ 1  0  0]
    """
    quiver = ClusterQuiverExtended(self._digraph.relabel(reordering, inplace=False), frozen = self._m)
    quiver._track_mut = self._track_mut
    quiver._is_principal = self._is_principal
    quiver._mut_path = [ reordering[i] for i in self._mut_path ]
    if inplace:
        self._M = quiver._M
        self._M.set_immutable()
        self._digraph = quiver._digraph
        self._mut_path = quiver._mut_path
    else:
        return quiver

def oriented_exchange_graph(
    self,
    depth=2**10000,
    method="breadth",
    special_start = None,
    special_start_digraph = None,
    special_start_iteration_index = -1,
    use_isomorphism = True,
    show_timing_messages = False,
    exclude_red_mutations = True,
    exclude_non_permissibles = True,
    identify_green_permissible_periods = True
    ):

    r"""
    Return an object of the class ``OrientedExchangeGraph``.

    INPUT:

    - ``depth`` -- (default: 2**10000) maximal depth for the search algorithm which constructs the oriented exchange graphs
    - ``method`` -- (default: ``breadth``) type of search algorithm; can be any of "breadth", "depth" or "first_maximal_green_sequence"
    - ``special_start`` -- (default: None) sequence of mutations given as a list as initial direction
    - ``special_start_digraph`` -- (default: None) a previously constructed DiGraph with ClusterQuiverExtended as objects at vertices may be handed over for further use
    - ``special_start_iteration_index`` -- (default: -1) if special_start_digraph gets set, one may specify the vertex index at which the search should be continued; by default this is set to the biggest index of the vertex set
    - ``use_isomorphism`` -- (default: True) flag if isomorphisms of (iced) quivers are supposed to be used to reduce the number of cases in the search algorithm; if True, such isomorphisms are stored as tuples in ``_isomorphism`` parameter of the ``OrientedExchangeGraph`` object
    - ``show_timing_messages`` -- (default: False) if True, the running time of each depth within the search is printed
    - ``exclude_red_mutations`` -- (default: True) flag if red mutations are excluded as directions in the search algorithm
    - ``exclude_non_permissibles`` -- (default: True) flag if non-permissible vertices are excluded as directions in the search algorithm
    - ``identify_green_permissible_periods`` -- (default: True) flag to stop
```

```
        in a branch of the search tree if a green permissible period has
        been detected

        EXAMPLES::

            sage: CQ = ClusterQuiverExtended(['A',2]).principal_extension()
            sage: OEG = CQ.oriented_exchange_graph()
            sage: OEG. get_number_isoclasses()
            5
            sage: OEG.get_green_permissible_periods()
            []
            sage: OEG.get_isomorphisms()
            [(4, 5)]

            sage: OEG = CQ.oriented_exchange_graph(method="
               first_maximal_green_sequence")
            The first maximal green sequence has been found.
            sage: OEG. get_number_isoclasses()
            4

            sage: CQ = ClusterQuiverExtended(matrix
               ([[0,2,-1],[-2,0,2],[1,-2,0]])).principal_extension()
            sage: OEG = CQ.oriented_exchange_graph(depth=10,
               exclude_non_permissibles = False)
            UserWarning: Running time and memory usage for infinite type can
                be enormous
            The maximal depth has been reached.
            sage: OEG. get_number_isoclasses()
            53
            sage: OEG.get_green_permissible_periods()
            [[7, 25], [13, 32], [21, 38], [23, 39], [27, 41], [34, 46], [42,
                51]]
            sage: OEG.get_isomorphisms()
            [(11, 12), (5, 17), (16, 18), (21, 29), (28, 30), (36, 45), (43,
                50), (52, 55)]

            sage: OEG = CQ.oriented_exchange_graph(depth=10,
               exclude_non_permissibles = False, exclude_red_mutations =
               False)
            UserWarning: Running time and memory usage for infinite type can
                be enormous
            The maximal depth has been reached.
            sage: OEG. get_number_isoclasses()
            600
            sage: OEG.get_isomorphisms()[:10]
            [(8, 12), (11, 13), (5, 19), (18, 20), (5, 25), (9, 34), (24,
               40), (39, 41)]

    .. SEEALSO:: For an already implemented version based on seeds, see :
       meth:`~sage.combinat.cluster_algebra_quiver.cluster_seed.
       oriented_exchange_graph`.
    """
    from collections import deque
    import warnings

    if use_isomorphism:
        import itertools

    if show_timing_messages:
        import timeit
```

```
        from time import gmtime, strftime

    if not self.is_finite():
        warnings.warn('Running time and memory usage for infinite type can be enormous')

    if not self._is_principal:
        raise TypeError('Only works for principal coefficients')

    iteration_index = 0
    iteration_depth = 0

    # Case if a pre-computed DiGraph is present
    if isinstance(special_start_digraph,DiGraph):
        if not all( [ isinstance(vertex_object,ClusterQuiverExtended) for vertex_object in special_start_digraph.iterator() ] ):
            raise TypeError('An object of the class ClusterQuiverExtended needs to be attached to each vertex of the DiGraph')
        if special_start_iteration_index > 0 and special_start_iteration_index < DiG.order():
            iteration_index = special_start_iteration_index
        else:
            if special_start_iteration_index <> -1:
                warnings.warn('The given index for the sub-digraph could not be recognized and will be ignored.')
            iteration_index = DiG.order()-1

        # Create new branch starting at determined index
        SubDiG = special_start_digraph.get_vertex(iteration_index).oriented_exchange_graph(
            depth=depth,
            method=method,
            use_isomorphism = use_isomorphism,
            show_timing_messages = show_timing_messages,
            exclude_red_mutations = exclude_red_mutations,
            exclude_non_permissibles = exclude_non_permissibles,
            identify_green_permissible_periods = identify_green_permissible_periods
         )

        # Correct the deposited mutation paths
        for v in SubDiG.vertices():
            v._mut_path = special_start_digraph.get_vertex(iteration_index)._mut_path + v._mut_path

        # Return the merged DiGraphs
        return( special_start_digraph.merge_digraph(SubDiG, merge_index=iteration_index, inplace=False) )

    else:
        if not self._is_principal:
            raise TypeError('The flag ``principal_extension`` of the ClusterQuiverExtended has to be True')

        # Initialize empty DiGraph and add source
        DiG = OrientedExchangeGraph(DiGraph())
        DiG.add_vertex()
        DiG.set_vertex(iteration_index,copy(self))
        DiG.get_vertex(iteration_index)._mut_path = []
        DiG.get_vertex(iteration_index)._track_mut = True
```

```
        no_further_check_list = []

        if show_timing_messages:
            start = timeit.default_timer()

        stack = deque([iteration_index])

        if use_isomorphism:
            reorder_list = list(itertools.permutations(range(self._n),self._n))

        # While the stack of to-be-considered indices is not empty...
        while stack:
            if method=="breadth":
                iteration_index = stack.popleft()
            else:
                iteration_index = stack.pop()

            S = DiG.get_vertex(iteration_index)

            # Keep track of the current depth of the search algorithm
            if iteration_depth < len(S._mut_path):
                iteration_depth = len(S._mut_path)
                if show_timing_messages:
                    stop = timeit.default_timer()
                    print('The runtime for depth '+ str(iteration_depth-1) + ' was '+ str(stop - start) + 's at ' + strftime("%Y-%m-%d %H:%M:%S", gmtime()))
                    start = timeit.default_timer()

            # Stop if desired depth has been reached
            if iteration_depth >= depth:
                if show_timing_messages:
                    stop = timeit.default_timer()
                    print('The runtime for the last depth was '+ str(stop - start) + 's at ' + strftime("%Y-%m-%d %H:%M:%S", gmtime()))
                break

            # If isomorphism classes have to be considered
            if use_isomorphism and DiG.order()<>1:
                # Get all isomorphic quivers of current case
                S_reordered_list = [S.reorder(sigma, inplace=False) for sigma in reorder_list]
                equivalent_index = -1
                # Test if any of the already considered cases are contained in the isomorphism class
                for v in DiG.vertices()[:iteration_index]:
                    if v not in no_further_check_list:
                        # If current case is isomorphic to old one, set a particular equivalence-index
                        if any( [DiG.get_vertex(v) == S_reordered for S_reordered in S_reordered_list] ):
                            equivalent_index = v
                            break

                # If equivalence-index is set, add appropriate edges to DiGraph and go to next index in the stack
                if equivalent_index <> -1:
```

```
                    for edge_in in DiG.incoming_edge_iterator([equivalent_index]):
                        DiG.add_edge( (edge_in[0], iteration_index, edge_in[2]) )
                    no_further_check_list.append(iteration_index)
                    DiG._isomorphism.append((equivalent_index, iteration_index))
                    continue

            # If green permissible periods are to be checked for
            if identify_green_permissible_periods:
                # Look for periodic parents up the branch of the DiGraph
                DiG.find_green_permissible_periods(iteration_index, use_isomorphism=use_isomorphism)
                # In case a green permissible parent has been found, go to next index in the stack
                if DiG._green_permissible_period:
                    if DiG._green_permissible_period[-1][1]==iteration_index:
                        continue

            # Obtain next mutation directions
            if exclude_red_mutations and (not exclude_non_permissibles):
                next_directions = S.get_green_vertices()
            elif (not exclude_red_mutations) and exclude_non_permissibles:
                next_directions = S.get_permissible_vertices()
                if S._mut_path:
                    if S._mut_path[-1] in next_directions:
                        next_directions.remove(S._mut_path[-1])
            elif exclude_red_mutations and exclude_non_permissibles:
                next_directions = S.get_green_permissible_vertices()
            else:
                next_directions = range(S._n)
                if S._mut_path:
                    next_directions.remove(S._mut_path[-1])

            # Add appropriate mutation directions to the stack and enlarge the DiGraph
            for i in next_directions:
                if ( special_start and len(S._mut_seq) < len(special_start) and  S._mut_seq+[i] == special_start[:(len(curGreen._mut_seq)+1)] ) or ( special_start and len(S._mut_seq) >= len(special_start) ) or not special_start:
                    added_vertex = DiG.add_vertex()
                    mutated_S = S.mutate(i,inplace=False)
                    mutated_S._is_principal = True
                    DiG.set_vertex(added_vertex, mutated_S)
                    DiG.add_edge(iteration_index,added_vertex)
                    stack.append(added_vertex)
    return DiG
```