Fundamenta Informaticae 149 (2016) 241–261
DOI 10.3233/FI-2016-1448
IOS Press

Graph Theoretical and Algorithmic Characterizations of Positive Definite Symmetric Quasi-Cartan Matrices

M. Abarca, D. Rivera[*†]

Instituto de Investigación en Ciencias Básicas y Aplicadas
Centro de Investigación en Ciencias
Universidad Autónoma del Estado de Morelos
Av. Universidad 1001, Cuernavaca, Mor. Mexico.
{*asma,darivera*}@*uaem.mx*

Abstract. A well known constructive proof for the $\mathbb{ADE}$-classification of many mathematical objects, such as positive unit forms and their associated quasi-Cartan matrices, has lead to an *Inflations Algorithm*. However, this algorithm is not known to run in polynomial time. In this paper we use a so called *flation* transformation and show how its invariants can be used to characterize the Dynkin types $\mathbb{A}$ and $\mathbb{D}$ in the language of graph theory. Also, a polynomial-time algorithm for computing the Dynkin type is suggested.

Keywords: Positive unit forms, Dynkin-type, edge-bipartite graphs, combinatorial algorithms

1. Introduction

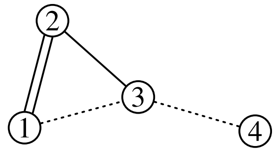
Quasi-Cartan matrices and unit forms play an important role in theory of representations of finite dimensional algebras (see [1, sec. II.6], [2] and [3]). These are square integer matrices $\boldsymbol{A}$ with diagonal entries $(\boldsymbol{A})_{ii} = 2$ with the property that there exists a diagonal matrix $\boldsymbol{D}$ with positive integer diagonal entries such that $\boldsymbol{D}\,\boldsymbol{A}$ is symmetric. In this paper we are focused in the set of symmetric quasi-Cartan matrices only, which we denote here as **sqC**.

*Address for correspondence: Centro de Investigación en Ciencias, Universidad Autónoma del Estado de Morelos; Av. Universidad 1001, Cuernavaca, Mor. Mexico.
†The authors gratefully acknowledge the support of CONACyT (Grant 156667).

Received August 2015; revised September 2016.

To each $\boldsymbol{A} \in \mathbf{sqC}$ we may associate a quadratic *unit form* $\mathbf{q}_{\boldsymbol{A}} : \mathbf{Z}^n \to \mathbf{Z}$ given by $\mathbf{q}_{\boldsymbol{A}} := \frac{1}{2}\,\boldsymbol{x}^{\mathrm{T}} \boldsymbol{A}\, \boldsymbol{x}$, and an edge-bipartite graph (*bigraph* for short) $\mathbf{bigr}\,(\boldsymbol{A})$ with vertices $\{1, \ldots, n\}$ and bundles of $\left|(\boldsymbol{A})_{i\,j}\right|$ *solid* (respectively *dotted*) edges between each pair of vertices $i \neq j$ such that $(\boldsymbol{A})_{i\,j} < 0$ (resp. $(\boldsymbol{A})_{i\,j} > 0$).

$\boldsymbol{A}$	1	2	3	4
1	2	−2	1	0
2	−2	2	−1	0
3	1	−1	2	1
4	0	0	1	2



$\mathbf{bigr}\,(\boldsymbol{A})$

$$\mathbf{q}_{\boldsymbol{A}}\,(x_1, x_2, x_3, x_4) = x_1^2 + x_2^2 + x_3^2 + x_4^2 - 2\,x_1\,x_2 + x_1\,x_3 - x_2\,x_3 + x_3\,x_4$$

Figure 1. A symmetric quasi-Cartan matrix and its associated unit form and bigraph.

We say that a bigraph G is *proper* if between each pair of vertices there are not both solid and dotted edges. To *simplify* a non-proper bigraph is to delete all pairs of solid and dotted parallel edges. Let us denote by $\mathbf{biadj}\,(G)$ the quasi-Cartan matrix associated to the simplified version of G.

Remark 1.1. The terms *unit form*, *symmetric quasi-Cartan matrix* and *proper bigraph* may be used interchangeably in many contexts; they are essentially the same thing by association.

1.1. The $\mathbb{ADE}$-classification

We say that $\boldsymbol{A} \in \mathbf{sqC}$ is *positive definite* if $\mathbf{q}_{\boldsymbol{A}}\,(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in \mathbf{Z}^n \setminus \{\mathbf{0}\}$. Also, $\boldsymbol{A}, \boldsymbol{B} \in \mathbf{sqC}$ are said to be $\mathbf{Z}$-*equivalent* if there exists $\boldsymbol{M} \in \mathcal{M}_n\,(\mathbf{Z})$ with $\boldsymbol{M}^{-1} \in \mathcal{M}_n\,(\mathbf{Z})$ such that $\boldsymbol{B} = \boldsymbol{M}^{\mathrm{T}}\,\boldsymbol{A}\,\boldsymbol{M}$. The following classification Theorem is considered classical (see [4], [5, sec. 1.2] and [6, sec. 6.2]):

Theorem 1.2. Any positive definite bigraph G is $\mathbf{Z}$-equivalent to a graph Δ (i.e. with no dotted edges), which is uniquely determined up to isomorphism of graphs, and is the disjoint union of Dynkin graphs.

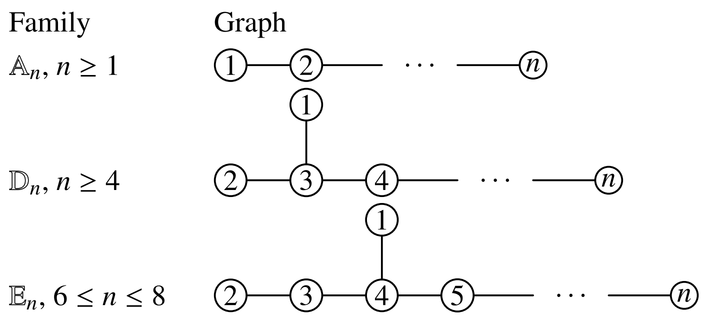


Figure 2. Dynkin graphs.

The Δ graph which Theorem 1.2 refers to is called the *Dynkin type* of G. It follows that two objects with the same Dynkin type are $\mathbf{Z}$-equivalent; therefore, it is important to have efficient criteria to decide when do two given objects share the same Dynkin type.

1.2. The inflations algorithm

There is a well known constructive proof of Theorem 1.2 introduced by Ovsienko [4]. Here we outline the basic ideas as presented in [6, sec. 6.2]; these form the basis of the proof technique used here. The reader is referred to [1, sec. II.7] and [7] for a detailed explanation.

Consider the elementary matrix $\boldsymbol{T}^{\sigma}_{s\,r} := \boldsymbol{I} + \sigma\, \boldsymbol{e}_s\, \boldsymbol{e}_r{}^{\mathrm{T}}$ where $\boldsymbol{I}$ is the identity matrix of size n and $\boldsymbol{e}_j$ the j-th column vector of $\boldsymbol{I}$. For $s \neq r$ it is the identity matrix but with a σ in the (s, r) position, and it is called an *inflation* if $\sigma = -1$ and *deflation* if $\sigma = 1$. Clearly $\boldsymbol{T}^{1}_{s\,r}$ is the inverse matrix of $\boldsymbol{T}^{-1}_{s\,r}$; then, for any $\boldsymbol{A} \in \mathbf{sqC}$, $\boldsymbol{B} := \left(\boldsymbol{T}^{-1}_{s\,r}\right)^{\mathrm{T}} \boldsymbol{A} \left(\boldsymbol{T}^{-1}_{s\,r}\right)$ is $\mathbf{Z}$-equivalent to $\boldsymbol{A}$.

A bigraph is *simple* if it does not contains parallel edges between any two vertices. It is easy to check that if $\boldsymbol{A} \in \mathbf{sqC}$ is positive-definite then $\mathbf{bigr}(\boldsymbol{A})$ is simple. Short proof: assume $(\boldsymbol{A})_{i\,j} \neq 0$; then for $\boldsymbol{x} := \boldsymbol{e}_i - \frac{(\boldsymbol{A})_{i\,j}}{|(\boldsymbol{A})_{i\,j}|}\boldsymbol{e}_j$ we get $\mathbf{q}_{\boldsymbol{A}}(\boldsymbol{x}) = 2 - \left|(\boldsymbol{A})_{i\,j}\right|$ which is positive if and only if $\left|(\boldsymbol{A})_{i\,j}\right| < 2$.

The idea behind Algorithm 1.1 below may be formulated as follows: we are given a positive definite quasi-Cartan matrix $\boldsymbol{A}$ with associated bigraph G; we wish to remove all dotted edges from G while preserving $\mathbf{Z}$-equivalence; to this end we apply inflations over dotted edges repeatedly until all of them are solid. It can be proved that this algorithm achieves this goal in a finite number of iterations which is bounded by $|\{\boldsymbol{x} \in \mathbf{Z}^n | \mathbf{q}_{\boldsymbol{A}}(\boldsymbol{x}) = 1\}|$ (more on this in Section 5). Also, it is not too difficult to convince oneself that the Dynkin graphs are the only positive definite connected bigraphs with no dotted edges; consequently, the algorithm must end with a disjoint union of Dynkin graphs.

Algorithm 1.1: Inflations Algorithm
Input: a positive definite $\boldsymbol{A} \in \mathbf{sqC}$
while *there is some non-diagonal entry* $(\boldsymbol{A})_{s\,r} = 1$ **do**
    $\boldsymbol{A} := \left(\boldsymbol{T}^{-1}_{s\,r}\right)^{\mathrm{T}} \boldsymbol{A} \left(\boldsymbol{T}^{-1}_{s\,r}\right)$
return bigr $(\boldsymbol{A})$

2. The flation transformation

In this section we study the transformation given by $\boldsymbol{A}' := (\boldsymbol{T}^{\sigma}_{s\,r})^{\mathrm{T}} \boldsymbol{A}\, (\boldsymbol{T}^{\sigma}_{s\,r})$. A direct computation reveals that the entries of $\boldsymbol{A}'$ are as follows:

$$(\boldsymbol{A}')_{i\,j} = \begin{cases} (\boldsymbol{A})_{i\,j} & \text{if } i \neq r \text{ and } j \neq r \\ (\boldsymbol{A})_{r\,j} + \sigma\, (\boldsymbol{A})_{s\,j} & \text{if } i = r \text{ and } j \neq r \\ (\boldsymbol{A})_{i\,r} + \sigma\, (\boldsymbol{A})_{i\,s} & \text{if } i \neq r \text{ and } j = r \\ (\boldsymbol{A})_{r\,r} + \sigma\, ((\boldsymbol{A})_{s\,r} + (\boldsymbol{A})_{r\,s}) + \sigma^2\, (\boldsymbol{A})_{s\,s} & \text{if } i = r \text{ and } j = r \end{cases} \tag{1}$$

If we assume that $\boldsymbol{A} \in \mathbf{sqC}$ and demand $\boldsymbol{A}' \in \mathbf{sqC}$ then the diagonal entry $(\boldsymbol{A}')_{r\,r} = 2 + 2\,\sigma\, (\boldsymbol{A})_{s\,r} + 2\,\sigma^2 = 2$. Hence $\sigma = 0$ or $\sigma = -(\boldsymbol{A})_{s\,r}$. Clearly $\boldsymbol{T}^{0}_{s\,r}$ is the identity matrix. Following [8], we call *flation* the transformation given by[1]:

$$T_{s\,r}(\boldsymbol{A}) := \left(\boldsymbol{T}^{-(\boldsymbol{A})_{s\,r}}_{s\,r}\right)^{\mathrm{T}} \boldsymbol{A} \left(\boldsymbol{T}^{-(\boldsymbol{A})_{s\,r}}_{s\,r}\right) \tag{2}$$

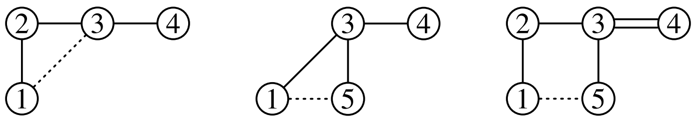
[1] In [8] a *flation* is either an inflation or a deflation

Remark 2.1. Notice that $T_{s_2\,r_2}\left(T_{s_1\,r_1}\left(\boldsymbol{A}\right)\right) = \left(\boldsymbol{T}_{s_2\,r_2}^{-(A')_{s_2\,r_2}}\right)^{\mathrm{T}} \boldsymbol{A}' \left(\boldsymbol{T}_{s_2\,r_2}^{-(A')_{s_2\,r_2}}\right)$ where $\boldsymbol{A}' := T_{s_1\,r_1}\left(\boldsymbol{A}\right)$.

2.1. Sum of bigraphs

For the sake of definiteness, let us define a *sum* operator on bigraphs; this operator will be analogue to the sum of quasi-Cartan matrices (save for the diagonal entries) and also to the union of graphs.

For any multiset E of edges, let $\mathbf{simp}\,(E)$ denote the multiset obtained after removing all pairs of solid/dotted parallel edges; e.g. $E := \{1—2, 1—2, 1\cdots\cdots 2, 1\cdots\cdots 3\}$, $\mathbf{simp}\,(E) = \{1—2, 1\cdots\cdots 3\}$. Recall that the sum of multisets $X + Y$ is given by adding the multiplicity of each element in either X or Y (see [9]). We write $G + H = (V, E)$ where $V := V_G \cup V_H$ and $E := \mathbf{simp}\,(E_G + E_H)$.



G $\qquad$ H $\qquad$ $G + H$

Figure 3. The sum of two bigraphs

2.2. The flation transformation on bigraphs

Equation (1) can be implemented for symmetric quasi-Cartan matrices in $\mathrm{O}\,(n)$ arithmetic operations by updating only the entries of the r-th row and column; nonetheless, we think it is convenient to understand the effect of $T_{s\,r}$ on bigraphs. To this end, let us the define the following $n \times n$ matrix:

$$\boldsymbol{Edge}\,(i, j) := \begin{cases} \boldsymbol{e}_i\,\boldsymbol{e}_j{}^{\mathrm{T}} + \boldsymbol{e}_j\,\boldsymbol{e}_i{}^{\mathrm{T}} & \text{if } i \neq j \\ \boldsymbol{0} & \text{if } i = j \end{cases}$$

Notice that, for any $\boldsymbol{A} \in \mathbf{sqC}$, the sum $\boldsymbol{A} + k\,\boldsymbol{Edge}\,(u, v)$ conveniently translates in bigraph terminology to *add and simplify* k parallel edges between u and v with dotted line-style if $k > 0$ or solid otherwise. Then Equation (1) can be rewritten as the following matrix summation for $\sigma = -\left(\boldsymbol{A}\right)_{s\,r}$, and $\boldsymbol{A} \in \mathbf{sqC}$:

$$T_{s\,r}\left(\boldsymbol{A}\right) = \boldsymbol{A} + \sum_{i \in \{1,\ldots,n\} \setminus \{r\}} \left(-\left(\boldsymbol{A}\right)_{i\,s}\left(\boldsymbol{A}\right)_{s\,r}\right) \boldsymbol{Edge}\,(i, r) \tag{3}$$

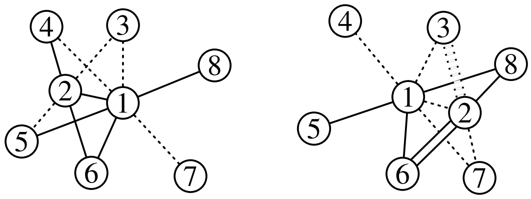
$$= \boldsymbol{A} + \left(-2\left(\boldsymbol{A}\right)_{s\,r}\right) \boldsymbol{Edge}\,(s, r) + \sum_{i \in \{1,\ldots,n\} \setminus \{r,s\}} \left(-\left(\boldsymbol{A}\right)_{i\,s}\left(\boldsymbol{A}\right)_{s\,r}\right) \boldsymbol{Edge}\,(i, r) \tag{4}$$

In the following, let u——v (with greyed line-style) denote any edge with endpoints $\{u, v\}$. Notice that $\left(\boldsymbol{A}\right)_{i\,s}\left(\boldsymbol{A}\right)_{s\,r}$ is the number of distinct simple paths i——s——r. Equation (4) yields Algorithm 2.1 (below) which is intended to be easy to read.

2.3. Flation properties

Let $G - x$ denote the bigraph obtained from G by removing a vertex x and all its incident edges. The following Lemma says that G differs from $T_{s,r}\,(G)$ only in the edges that end in r, that two consecutive

Algorithm 2.1: A procedure for updating $G := T_{sr}(G)$
Input: A bigraph G and two vertices r, s
if $s = r$ **then**
  for each *edge* $i \overset{e}{\text{——}} r$ **do**
    swap the dotted/solid line style of e
else
  for each *path* $i \overset{e}{\text{——}} s \overset{f}{\text{——}} r$ *with* $i \neq r$ **do**
    if *e and f have the same line style* **then**
      add a new solid edge $i \overset{g}{\text{——}} r$
    else
      add a new dotted edge $i \overset{g}{\cdots\cdots} r$
    if *there exists an edge* $i \overset{h}{\text{——}} r$ *with opposite line style as* g **then** erase g and h
  for each *edge* $s \overset{f}{\text{——}} r$ **do** swap the dotted/solid line style of f



G $\qquad$ $T_{1\,2}(G)$

Figure 4. A flation example. Vertex 2 moves a little to the right and carries along the edges incident on vertex 1

flations T_{sr} (with the same s and r) nullify each other, and that T_{sr} has null effect when there is no edge $s \text{——} r$.

Lemma 2.2. Let G be any bigraph with two distinct vertices s, r; then

1. $T_{sr}(G) - r = G - r$
2. $T_{sr}(T_{sr}(G)) = G$
3. If $s \neq r$ and there is no edge between s and r, then $T_{sr}(G) = G$

Proof:
Notice that the flation adds, removes or swaps the line style of edges ending in r only. This proves property 1.

T_{rr} swaps the line style of edges ending in r, and a second application of T_{rr} just swaps them back to their original line style. For $s \neq r$, let $\boldsymbol{A} := \mathbf{biadj}(G)$ and $\boldsymbol{A}' := T_{sr}(\boldsymbol{A})$. By equation (1) follows that $(\boldsymbol{A}')_{sr} = -(\boldsymbol{A})_{sr}$; thus, by remark 2.1, $T_{sr} \circ T_{sr}$ has the same effect as $\boldsymbol{T}_{sr}^{(\boldsymbol{A})_{sr}} \boldsymbol{T}_{sr}^{-(\boldsymbol{A})_{sr}} = \boldsymbol{T}_{sr}^{0} = \boldsymbol{I}$.

Finally, if there is no edge between s and r then $(\boldsymbol{A})_{sr} = 0$ and $\left(\boldsymbol{T}_{sr}^{-(\boldsymbol{A})_{sr}}\right)$ is the identity matrix. $\square$

We now show how to express $T_{s\,s}$ as a composition of many $T_{s\,r_i}$ where $s \neq r_i$ for all i. Thus, whenever we want to study the effect of any flation $T_{s\,r}$, we may assume without any loss of generality that $s \neq r$.

Lemma 2.3. Let G be any bigraph with some vertex s, and let $r_1, r_2, \ldots, r_k$ be the neighbours of s. Then $T_{s\,s}(G) = T_{s\,r_k} \circ \cdots \circ T_{s\,r_2} \circ T_{s\,r_1}(G)$.

Proof:
Let $\boldsymbol{A} := \mathbf{biadj}(G)$, $T := T_{s\,r_k} \circ \cdots \circ T_{s\,r_2} \circ T_{s\,r_1}$ and $G' := T(G)$. Lemma 2.2 implies that each $T_{s\,r_i}$ does not affect the other edges s—r_j; hence, we infer the following facts:

- T is given by the matrix $\boldsymbol{T} := \boldsymbol{T}_{s\,r_k}^{-(\boldsymbol{A})_{s\,r_k}} \cdots \boldsymbol{T}_{s\,r_2}^{-(\boldsymbol{A})_{s\,r_2}}\, \boldsymbol{T}_{s\,r_1}^{-(\boldsymbol{A})_{s\,r_1}}$, where

$$(\boldsymbol{T})_{i\,j} = \begin{cases} 1 & \text{if } i = j \\ -(\boldsymbol{A})_{s\,j} & \text{if } i = s \text{ and } j \text{ is a neighbour of } s \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

- All edges s—r_i swap their line style exactly once (see line 11 of algorithm 2.1).

All that is left, is to prove that the all other edges (i. e. the ones not ending in s) are not affected by T. Let $\boldsymbol{B} := \boldsymbol{T}^{\mathrm{T}}\,\boldsymbol{A}\,\boldsymbol{T}$. Then for all i, j, s ($i \neq j, i \neq s, j \neq s$):

$$\begin{aligned}
(\boldsymbol{B})_{i\,j} &= \sum_{\ell=1}^{n}\sum_{k=1}^{n} (\boldsymbol{T})_{\ell\,i}\,(\boldsymbol{A})_{\ell\,k}\,(\boldsymbol{T})_{k\,j} \\
&= (\boldsymbol{T})_{s\,i}\,(\boldsymbol{A})_{s\,j}\,(\boldsymbol{T})_{j\,j} + (\boldsymbol{T})_{s\,i}\,(\boldsymbol{A})_{s\,s}\,(\boldsymbol{T})_{s\,j} + (\boldsymbol{T})_{i\,i}\,(\boldsymbol{A})_{i\,s}\,(\boldsymbol{T})_{s\,j} + (\boldsymbol{T})_{i\,i}\,(\boldsymbol{A})_{i\,j}\,(\boldsymbol{T})_{j\,j} \\
&= -(\boldsymbol{A})_{s\,i}\,(\boldsymbol{A})_{s\,j} + (\boldsymbol{A})_{s\,i}\,(\boldsymbol{A})_{s\,s}\,(\boldsymbol{A})_{s\,j} - (\boldsymbol{A})_{i\,s}\,(\boldsymbol{A})_{s\,j} + (\boldsymbol{A})_{i\,j} \\
&= (\boldsymbol{A})_{i\,j}
\end{aligned}$$

The first equation is given by direct computation; then we use the fact that only the diagonal and s-th row of $\boldsymbol{T}$ contain non-zero elements; then substitute equation (5); and the rest follows from $\boldsymbol{A}$ being a symmetric quasi-Cartan matrix. $\square$

2.4. Flation invariants

Consider any property P about graphs, such as "G is connected". We say that P is a *flation invariant* if $P(G)$ implies $P(T_{s\,r}(G))$ for any bigraph G and any selection of vertices s and r.

Example 2.4. The property "G is connected" is a flation invariant. To prove it, let G be any connected bigraph and $G' := T_{s\,r}(G)$. We may assume $s \neq r$ because of Lemma 2.3. By Lemma 2.2, only vertex r can be disconnected in G'; but this is not the case because G' contains an edge s—r.

Example 2.5. Following [8], we say that G satisfies the *cycle condition* if every chordless cycle (i.e. vertex-induced cycle) has an odd number of dotted edges. The cycle condition is not a flation invariant. Consider the counterexample given in Figure 5: G satisfies the cycle condition, but $T_{5\,1}(G)$ has a chordles cycle 1—3—2—4—1 with no dotted edges.

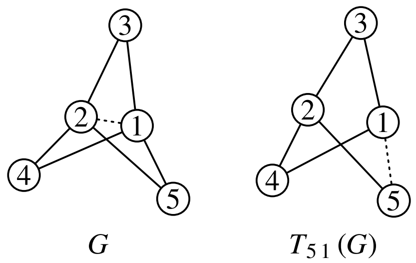


Figure 5. The cycle condition is not a flation invariant.

Lemma 2.6. Let Δ be a disjoint union of Dynkin graphs and let P be a flation invariant such that $P(\Delta)$. Then $P(G)$ holds for all bigraphs G of Dynkin type Δ.

Proof:
Assume that G is any positive definite bigraph; then we may apply the Inflations Algorithm to it and obtain a chain of flations $T_{s_\ell,r_\ell} \circ \cdots \circ T_{s_2,r_2} \circ T_{s_1\,r_1}(G) = \Delta$ where Δ is the Dynkin type of G. By Lemma 2.2 we may rewrite this chain as $G = T_{s_1\,r_1} \circ T_{s_2\,r_2} \circ \cdots \circ T_{s_\ell\,r_\ell}(\Delta)$. If P is a flation invariant and $P(\Delta)$ holds, then P also holds for all $G_k := T_{s_k\,r_k} \circ T_{s_{k+1}\,r_{k+1}} \circ \cdots \circ T_{s_\ell\,r_\ell}(\Delta)$ with $k = \ell, \ell-1, \ldots, 2, 1$. $\square$

3. Bigraphs of Dynkin type $\mathbb{A}$

In this section we use flation invariants to characterize the Dynkin type $\mathbb{A}$, i.e. the set of all bigraphs whose Dynkin type is $\mathbb{A}_n$ for some $n \geq 1$. Our characterization is based on [8], written in the language of graph theory and the flations method alone. We compare these characterizations at the end of this section.

3.1. Main result for $\mathbb{A}$

Following [10, sec. 5.2], a *separating vertex* of a connected bigraph G is a vertex x whose deletion results in a disconnected bigraph; a non-separating vertex is called an *internal vertex*. Also, for any two edges e, e' we say that e and e' are equivalent, denoted $e \sim e'$, if either $e = e'$ or there is a simple cycle containing both e and e'. It is easy to see that $\sim$ is an equivalence relation on the edges. Each equivalence class E_i induces a sub-bigraph $B_i := (V_i, E_i)$ whose vertex set V_i consists of all ends of edges of E_i. Each B_i is a nonseparabe sub-bigraph (i.e. with no separating vertex) and maximal with respect to this property; consequently, they are called *blocks* of G. By convention, any singleton graph is a block on its own. If we collapse each block into a single vertex while retaining all separating vertices, so that each separating vertex s_j is joined to the meta-vertex B_i if and only if s_j belongs to B_i, the result is a tree graph $\mathrm{BT}(G)$ called the *block tree* of G. See Figure 6 for an example.

Let X and Y be disjoint vertex sets. We denote by $\mathbf{F}[X, Y]$ the nonseparable bigraph obtained by joining each pair of vertices x, y with $x \in X$ and $y \in Y$ by a solid edge, and all other pairs of vertices by a dotted edge; such bigraph is called an $\mathbb{A}$-*block* (examples in Figs. 6 and 7). Let $\mathcal{F}$ denote the class of all $\mathbb{A}$-blocks. We say that any block tree is an $\mathbb{A}$-*block tree* if all blocks are $\mathbb{A}$-blocks and each separating vertex s has degree 2 in $\mathrm{BT}(G)$.

Theorem 3.1. The bigraph G has Dynkin type $\mathbb{A}$ if and only if G has $n \geq 1$ vertices and the block tree of G is an $\mathbb{A}$-block tree.

An example of Theorem 3.1 is shown in Figure 6 below. All blocks are $\mathbb{A}$-blocks (notice that $B_1 = \mathbf{F}[\{3,4\},\{5,9\}]$, $B_2 = \mathbf{F}[\{1\},\{2,3\}]$, $B_3 = \mathbf{F}[\{5,6,8\},\emptyset]$, and $B_4 = \mathbf{F}[\{4\},\{7\}]$) and each separation vertex belongs to exactly two blocks; hence, $\mathrm{BT}(G)$ is an $\mathbb{A}$-block tree and G has Dynkin type $\mathbb{A}$.

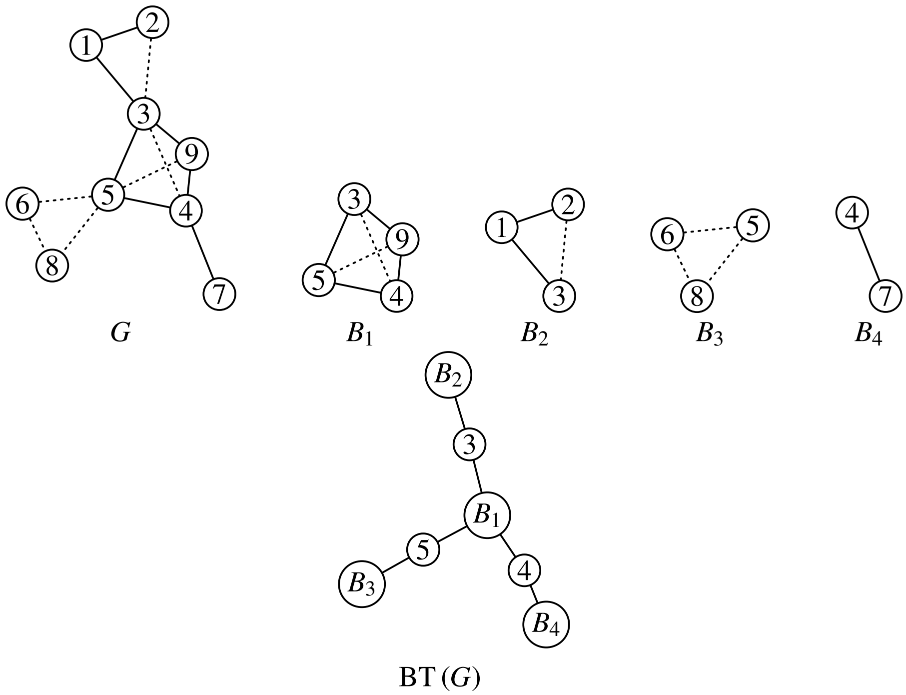


Figure 6. A bigraph and the decomposition into its block tree.

The rest of this section is devoted to prove Theorem 3.1.

3.2. The $\mathbb{A}$-block tree flation invariant

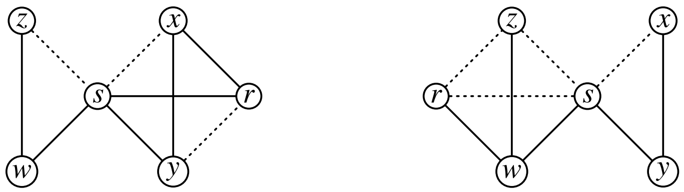
Let us begin by studding the effect of T_{sr} on $\mathbb{A}$-blocks. The following Lemma states that T_{sr} “moves” r from the block H containing s—r to the contiguous $\mathbb{A}$-block F separated by s (see Figure 7 below). We can always assume the existence of such F by setting $F := \mathbf{F}[\{s\},\emptyset]$ if needed. Notice, however, that F is not necessarily a block of the whole bigraph, and neither H' mentioned below.

Lemma 3.2. Let s—r be an edge of some $H \in \mathcal{F}$, and let $F \in \mathcal{F}$ such that $V_F \cap V_H = \{s\}$; then, $T_{sr}(F+H) = F' + H'$ where:

1. F' and H' are members of $\mathcal{F}$ such that $V_{F'} \cap V_{H'} = \{s\}$
2. $F' - r = F$ and $H - r = H'$

Proof:
Let $H = \mathbf{F}[X, Y]$ and $F = \mathbf{F}[X', Y']$. Without any loss of generality assume that $s \in X \cap X'$ (otherwise reverse the roles of X and Y or X' and Y' by noting that $\mathbf{F}[X, Y] = \mathbf{F}[Y, X]$). There are two cases: $r \in Y$ or $r \in X \setminus \{s\}$. In either case we use algorithm 2.1 (from line 5 onwards) to compute $G' := T_{s\,r}\, G$.

First assume $r \in Y$. Notice that for each $i \in X \setminus \{s\}$ there is a dotted edge i······s that cancels out the solid edge i——r and for each $i \in Y$ there is a solid edge i——s that cancels out the dotted edge i······r, so that the vertex r is not adjacent in G' to any other vertex of X nor Y, except for s, which is connected to r by a dotted edge. Also, since r is not adjacent to any vertex of X' nor Y' in G, then for each $i \in X'$ (resp. $i \in Y'$) we add a dotted edge i······r (resp. solid edge i——r). By construction we get $G' = F' + H'$ with $F' = \mathbf{F}[X, Y] \setminus \{r\}$ and $H' = \mathbf{F}[X' \cup \{r\}, Y']$. The case $r \in X \setminus \{s\}$ is analogous and yields $F' = \mathbf{F}[X \setminus \{r\}, Y]$ and $H' = \mathbf{F}[X', Y' \cup \{r\}]$. □



$$\underbrace{\mathbf{F}[\{s, z\}, \{w\}]}_{F} + \underbrace{\mathbf{F}[\{s, x\}, \{r, y\}]}_{H} \qquad \underbrace{\mathbf{F}[\{r, s, z\}, \{w\}]}_{F'} + \underbrace{\mathbf{F}[\{s, x\}, \{y\}]}_{H'}$$

Figure 7. A minimal nontrivial example of Lemma 3.2

Lemma 3.3. The property of having an $\mathbb{A}$-block tree is flation invariant.

Proof:
Let G be a bigraph such that $\mathrm{BT}(G)$ is an $\mathbb{A}$-block tree and let $G' := T_{s\,r}(G)$. Without any loss of generality assume that s══r is an edge of G; then, it is contained in some block $B \in \mathcal{F}$. Consider the sub-bigraph $\bar{B}$ of G induced by s and all of its neighbors that do not belong to B. Notice that either $\bar{B}$ is the singleton bigraph $\mathbf{F}[\{s\}, \emptyset]$ or s is a separation vertex of G. By Lemma 3.2, $T_{s\,r}$ moves r from B to $\bar{B}$ yielding $B' = B - r$, and $\bar{B}' \in \mathcal{F}$ where $\bar{B}' - r = \bar{B}$. Then $\bar{B}' \in \mathcal{F}$ is a block of G' since it is a maximal nonseparable bigraph, and contains at least one edge (namely s══r). On the other hand B' is a block of G' if and only if B contains some vertex other than s and r (otherwise $B' = \mathbf{F}[\{s\}, \emptyset] \subseteq \bar{B}'$); and in this case the vertex s happens to separate B' and $\bar{B}'$only. By using the fact that $T_{s\,r}$ only affects the subgraph of G induced by s and its neighbors (Lemma 2.2), all blocks of G' are $\mathbb{A}$-blocks and all of the separation vertices of G' separates exactly two blocks. Therefore $\mathrm{BT}(G')$ is an $\mathbb{A}$-block tree. □

Lemma 3.4. If G is a bigraph of Dynkin type $\mathbb{A}$, then $\mathrm{BT}(G)$ is an $\mathbb{A}$-block tree.

Proof:
The block tree of the graph $\mathbb{A}_n$ is given in Figure 8. The separation vertices are $2, 3, \ldots, n-1$, and the blocks are $B_i = \mathbf{F}[\{i\}, \{i+1\}]$ for $i = 1, 2, \ldots, n-1$. Clearly $\mathrm{BT}(\mathbb{A}_n)$ is an $\mathbb{A}$-block tree; then by Lemma 2.6 it follows that all bigraphs G of Dynkin type $\mathbb{A}$ are such that $\mathrm{BT}(G)$ is an $\mathbb{A}$-block tree. □

$$B_1 — 2 — B_2 — 3 — \cdots — k — B_k$$

Figure 8. The block tree of $\mathbb{A}_n$, where $k := n - 1$.

3.3. Proof of main result for $\mathbb{A}$

Lemma 3.4 proves one of the two implications of the "if and only if" mentioned in Theorem 3.1. Here we prove the reciprocal.

Lemma 3.5. If $F := \mathbf{F}[X, Y]$ has at least one vertex, then it has Dynkin type $\mathbb{A}$.

Proof:
At least one of X or Y is not empty, and X and Y are interchangeable; thus, we may assume without any loss of generality that $X \neq \emptyset$ and let $X = \{x_1, \ldots, x_m\}$. By applying algorithm 2.1 we see that:

$$T_{x_2\,x_1}(F) = x_1 —\!\!— x_2 + \mathbf{F}[\{x_2, \ldots, x_m\}, Y]$$

This can be generalized by induction to:

$$T_{x_m\,x_{m-1}} \circ T_{x_3\,x_2} \circ T_{x_2\,x_1}(F) = x_1 —\!\!— x_2 —\!\!— \cdots —\!\!— x_m + \mathbf{F}[\{x_m\}, Y]$$

If $Y = \emptyset$ then we are done, since $\mathbf{F}[\{x_m\}, Y]$ is the single vertex x_m. Otherwise, let $Y = \{y_1, y_2, \ldots, y_{m'}\}$. By analogous argument, $T_{y_1\,y_2} \circ \cdots \circ T_{y_{m'-1}\,y_{m'}} \circ T_{x_m\,x_{m-1}} \circ T_{x_3\,x_2} \circ T_{x_2\,x_1}(F)$ yields the following bigraph:

$$x_1 —\!\!— x_2 —\!\!— \cdots —\!\!— x_m + \mathbf{F}[\{x_m\}, \{y_1\}] + y_1 —\!\!— y_2 —\!\!— \cdots —\!\!— y_{m'}$$

Notice that $\mathbf{F}[\{x_m\}, \{y_1\}] = x_m —\!\!— y_1$. Therefore $\mathbf{F}[X, Y]$ has Dynkin type $\mathbb{A}$. □

Lemma 3.6. If G is any bigraph with at least one vertex, such that $\mathrm{BT}(G)$ is an $\mathbb{A}$-block tree, then G has Dynkin type $\mathbb{A}$.

Proof:
If G consists of a single $\mathbb{A}$-block then we are done by previous Lemma. Otherwise let s be the separating vertex of some blocks F and H, and let $\{s, r_1, r_2, \ldots, r_k\}$ be the vertex set of F. Then by repeated use of Lemma 3.2 we see that $T_{s\,r_k} \circ \cdots \circ T_{s\,r_2} \circ T_{s\,r_1}$ "merges" F and H into a single $\mathbb{A}$-block. This shows that it is always possible to transform any bigraph G which block tree is an $\mathbb{A}$-block tree into a single $\mathbb{A}$-block; then by the previous Lemma, transform it further into the graph $\mathbb{A}$. □

3.4. Comparison with previous results

As mentioned earlier, Barot [8] gave a characterization for unit forms of Dynkin type $\mathbb{A}$ of which this work is a reinterpretation.

Let $\delta_G(v)$ denote the set of all edges of G ending in some vertex v. A bigraph G is said to be an *assemblage* if it can be obtained by the disjoint union of some bigraphs $B_1, \ldots, B_n$ by identifying $\sigma_i(i —\!\!— j)$ and $\sigma_j(i —\!\!— j)$ for all $i —\!\!— j$ in some arbitrary graph Γ with vertices $\{1, 2, \ldots, t\}$, where all $\sigma_i : \delta_\Gamma(i) \to V(B_i)$ are injective functions. When Γ is a tree, it is called a *tree assemblage*.

Let $F_{m,m'} := \mathbf{F}[X, Y]$ where $X = \{(1,1), (1,2), \ldots, (1,m)\}$ and $Y = \{(2,1), (2,2), \ldots, (2,m')\}$. The main result of [8] states that a unit form $q : \mathbf{Z}^n \to \mathbf{Z}$ has Dynkin type $\mathbb{A}$ if and only if $\mathbf{bigr}(q)$ is a tree assemblage of bigraphs of the form $F_{m,m'}$ each.

Notice that both characterizations are equivalent:

- Let G be a tree assemblage of Γ and $B_1, \ldots, B_t$ and say that the disjoint union renames vertex (k_1, k_2) of B_i into (i, k_1, k_2) (so that the vertex sets are disjoint). The block tree of G can be obtained by replacing each edge i—j of Γ by B_i—(i, k_1, k_2)—B_j where $(k_1, k_2) := \sigma_i\,(i$—$j)$ and then identifying $(j, k_3, k_4) = (i, k_1, k_2)$ where $(k_3, k_4) := \sigma_j\,(i$—$j)$.

- Let $\mathrm{BT}\,(G)$ be an $\mathbb{A}$-block tree with $\mathbb{A}$-blocks $B_1, \ldots, B_t$, where each $B_i := \mathbf{F}\,[X_i, Y_i]$. Moreover, let $X_i = \{x_{i\,1}, x_{i\,2}, \ldots, x_{i\,m_i}\}$ and $Y_i = \left\{y_{i\,1}, y_{i\,2}, \ldots, y_{i\,m'_i}\right\}$ for all i. We define a labeling on the vertices of each B_i given by $L_i\,(v) := (1, k)$ if $v = x_{i\,k}$ and $L_i\,(v) := (2, k)$ if $v = y_{i\,k}$. Recall that for each separation vertex s there is a path B_i—s—B_j joining two $\mathbb{A}$-blocks B_i and B_j; for each one of them we may add an edge i—j to Γ and set $\sigma_i\,(i$—$j) := L_i\,(s)$ and $\sigma_j\,(i$—$j) := L_j\,(s)$. Then by replacing each vertex v in B_i by $L_i\,(v)$ we get the desired tree assemblage.

Notice however that the block tree is defined for all bigraphs and that it is easy to compute in linear time (see [11]), whereas only those bigraphs where each separation vertex joins exactly two blocks have a definite tree assemblage. Thus, our characterization can be easily translated into an efficient algorithm for testing Dynkin type $\mathbb{A}$.

4. Bigraphs of Dynkin type $\mathbb{D}$

For the Dynkin type $\mathbb{D}$ bigraphs two constructions were presented in [12]; the combination of which characterizes the positive unit forms of Dynkin type $\mathbb{D}$ up to the line style of the associated bigraph. We give a new construction which fully characterizes the Dynkin type $\mathbb{D}$ bigraphs and which proof follows the same pattern as the previous section. Again, a comparision is presented at the end of this section.

4.1. Main result for $\mathbb{D}$

Construction 4.1. ($\mathbb{D}$-core gluing)

Items needed:

- A cycle bigraph $H = x_1$—x_2—$\cdots$—x_h—x_1 (all x_i distinct for $1 \leq i \leq h$) which satisfies the cycle condition (recall Example 2.5 on page 246). By convention we demand $h \geq 2$ and allow H to collapse into a 2-bond when $h = 2$ (an n-bond is a bundle of n parallel edges). This bigraph H will be referred to as the $\mathbb{D}$-*core*.

- Bigraphs $F_1, F_2, \ldots, F_h$, such that:
  - each F_i has Dynkin type $\mathbb{A}$
  - F_i contains an edge x_i—x_{i+1} with the same line style as in H
  - x_i and x_{i+1} are internal vertices of F_i
  - All F_i are vertex-disjoint except for the vertices of the $\mathbb{D}$-core. Specifically, if $F_i = (V_i, E_i)$, then $V_i \cap V_{i+1} = \{x_{i+1}\}$, $V_h \cap V_1 = \{x_1\}$, and $V_i \cap V_j = \emptyset$ for all other combinations of i, j.

Procedure:
Compute the bigraph sum $\sum_{i=1}^{h} F_i$; the resulting bigraph is the $\mathbb{D}$-*core gluing of H and $F_1, F_2, \ldots, F_h$.*

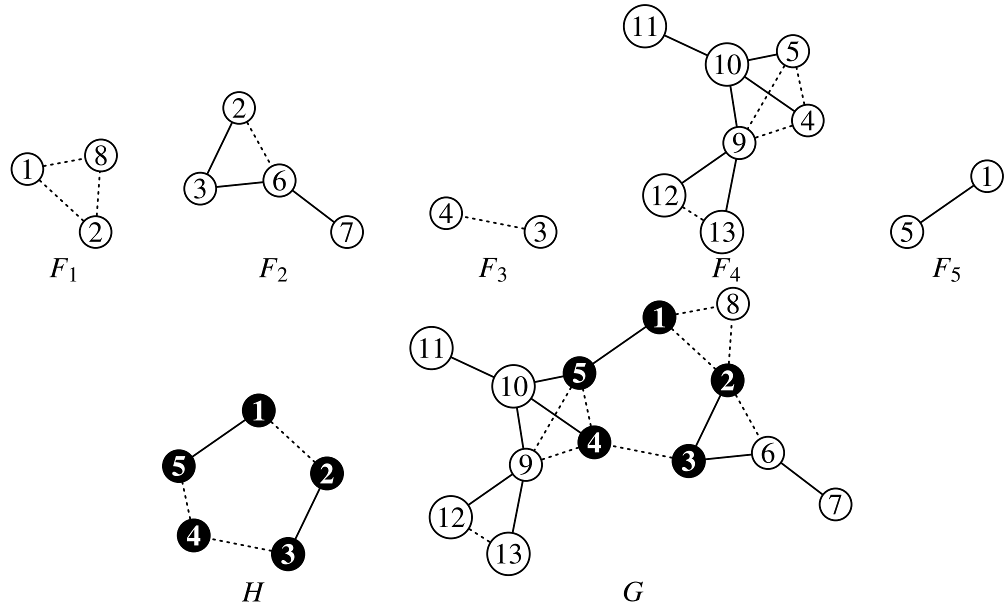


Figure 9. A $\mathbb{D}$-core gluing.

Theorem 4.2. Let G be any bigraph with $n \geq 4$ vertices. Then G has Dynkin type $\mathbb{D}$ if and only if G is a $\mathbb{D}$-core gluing.

4.2. The $\mathbb{D}$-core gluing flation invariant

Lemma 4.3. The $\mathbb{D}$-core gluing is a flation invariant; i.e. if G is a $\mathbb{D}$-core gluing, then so is $T_{sr}(G)$ for any selection of vertices s and r.

Proof:
Let G be the $\mathbb{D}$-core gluing of H with $F_1, F_2, \ldots, F_h$ as in Construction 4.1, and let $G' = T_{sr}(G)$. We may assume without any loss of generality that there exists an edge s—r in G. For $j = 1, \ldots, h$, let B_j denote the block of F_j which contains the edge x_j—x_{j+1}. Notice that, by construction, $F_{j-1} + F_j$ is a bigraph of Dynkin type $\mathbb{A}$ with separation vertex x_j for all $j = 1, \ldots, h$, ($F_0 := F_h$ and $x_0 := x_h$).

There are four cases depending on the location of s and r:

1. Neither s nor r are vertices of H. Let F_i be the sub-bigraph containing s—r. Clearly, s—r is not contained in H; therefore T_{sr} does not affect H and G' is the $\mathbb{D}$-core gluing of H with $F_1, \ldots, F_{i-1}, T_{sr}(F_i), F_{i+1}, \ldots, F_h$.

2. Only vertex s belongs to H. Let $s = x_i$; then s—r either belongs to B_i or to B_{i-1}. By Lemma 3.2, T_{sr} moves r between B_i and B_{i-1} without affecting H, yielding two bigraphs F'_{i-1} and F'_i separated by s; consequently, G' is the $\mathbb{D}$-core gluing of H with $F_1, \ldots, F_{i-2}, F'_{i-1}, F'_i, F_{i+1}, \ldots, F_h$.

3. Only vertex r belongs to H. We prove that T_{sr} make H to absorb s. Let $r = x_i$ and $t = x_{i+1}$. Assume without any loss of generality that s—r is an edge of B_i (the other possibility being

B_{i+1}). Let $\bar{B}_i$ be the subgraph of F_i induced by s and all neighbors of s that do not belong to B_i; then, $\bar{B}_i \in \mathcal{F}$ and either s is a separation vertex of F_i or $\bar{B}_i$ is a singleton graph. By Lemma 3.2, T_{sr} moves r from B_i to $\bar{B}_i$ turning s into a separation vertex of $T_{sr}(F_i)$ (if it was not already), and erasing edge $r \text{---} t$ in particular. Then we may split $T_{sr}(F_i)$ into two bigraphs F_i' and F_i'' separated by s; being F_i' the one which contains $r \text{---} s$ and F_i'' the one which contains $s \text{---} t$. Thus, $H' := x_1 \text{---} \cdots \text{---} x_{i-1} \text{---} x_i \text{---} s \text{---} x_{i+1} \text{---} \cdots \text{---} x_h \text{---} x_1$ is a chodrless cycle in G'. We now prove that H' has an odd number of dotted edges. By the structure of $\mathcal{F}$, its clear that B_i contains the triangle $s \text{---} r \text{---} t \text{---} s$ with exactly one or three dotted edges. Then by Algorithm 2.1 it is easy to see that the line style of edges $r \text{---} s$ and $s \text{---} t$ in G' are equal if and only if $r \text{---} t$ is a solid edge in G. Therefore, G' is the $\mathbb{D}$-core gluing of H' and $F_1, F_2, \ldots, F_{i-1}, F_i', F_i'', F_{i+1}, \ldots, F_h$.

4. Both s and r are in H. If s and r are the only vertices in H then edge $s \text{---} r$ has opposite line styles in F_1 and F_2, so $G = F_1 + F_2$ has no edge $s \text{---} r$; thus we may assume without any loss of generality that there are at least three vertices in H. This case has the opposite effect as the previous case (i.e. shrinking H by merging F_i and F_{i+1} and removing s from H). To see this, recall that two consecutive applications of the same T_{sr} nullify (Lemma 2.2); and notice that this case is precisely the result obtained after applying the T_{sr} as in the previous case with $r = x_i$ and $s = x_{i+1}$. □

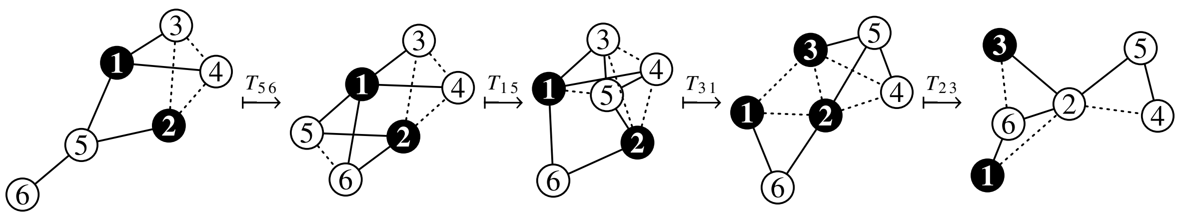


Figure 10. Example of the four cases mentioned in the proof of Lemma 4.3 (in the same given order). The vertices of the $\mathbb{D}$-core have been highlighted.

Lemma 4.4. If G is a bigraph of Dynkin type $\mathbb{D}$, then it is a $\mathbb{D}$-core gluing.

Proof:
The $\mathbb{D}$-core gluing of the bigraph $\mathbb{D}_n$ is given in Figure 11 below. Since the $\mathbb{D}$-core gluing is a flation invariant, by Lemma 2.6 all bigraphs of Dynkin type $\mathbb{D}$ are $\mathbb{D}$-core gluings. □

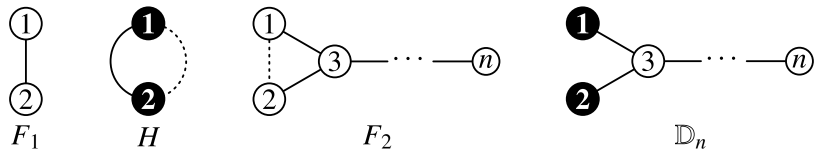


Figure 11. $\mathbb{D}_n$ is a $\mathbb{D}$-core gluing.

4.3. Proof of main result for $\mathbb{D}$

Lemma 4.5. Every $\mathbb{D}$-core on at least 4 vertices has Dynkin type $\mathbb{D}$.

Proof:
Let $C = 1$—2—$\cdots$—h—1 be a cycle with $h \geq 4$ vertices and an odd number of dotted edges. We may assume without any loss of generality that h⋯⋯1 is the only dotted edge; otherwise apply the following rules:

1. If h—1 is solid, apply $T_{h\,h}$; this guaranties that h—1 is dotted.
2. While there is a dotted edge other than h⋯⋯1, let $i < h$ be the minimum index such that $(i-1)$—i is dotted. Then $T_{i\,i}$ ensures that the path 1—2—$\cdots$—i has solid edges only; in addition, if i—$(i+1)$ was dotted, then this transformation yields a $\mathbb{D}$-core with 2 dotted edges less.

Let $h_i := h - i$ and recursively define $G_0 := C$, $G_{i+1} := T_{h_i\,1}(G_i)$ for $i = 1, \ldots, h-3$. Then each G_i is the $\mathbb{D}$-core gluing of $H_i := 1$—2—$\cdots$—h_i⋯⋯1 with $F_1^{(i)}, F_2^{(i)}, \ldots, F_{h_i}^{(i)}$ where $F_j^{(i)} := j$—$(j+1)$ for $j = 1, \ldots, h_i - 1$ and $F_{h_i}^{(i)} := \mathbf{F}[\{1, h_i\}, \{h_i + 1\}] + (h_i + 1)$—$\cdots$—$n$ (see Fig. 12 bellow). Notice that the $\mathbb{D}$-core gluing of G_{h-2} is the same as the $\mathbb{D}$-core gluing of $\mathbb{D}_n$ (see Fig. 11); therefore, $G_{h-2} = \mathbb{D}_n$ and C has Dynkin type $\mathbb{D}_n$. □

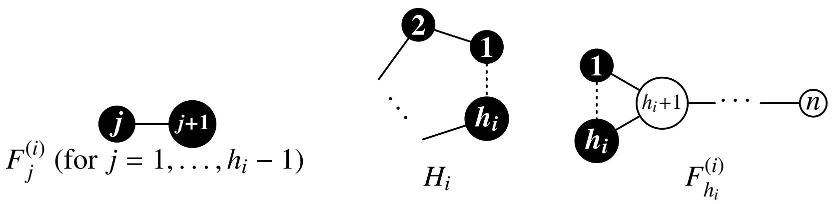


Figure 12. The $\mathbb{D}$-core gluings mentioned in the proof of Lemma 4.5. When $h_i = 2$ it is the same $\mathbb{D}$-core gluing as in Fig. 11.

Lemma 4.6. If G is a $\mathbb{D}$-core gluing with $n \geq 4$ vertices then it has Dynkin type $\mathbb{D}_n$.

Proof:
Assume that G is a $\mathbb{D}$-core gluing of H with $F_1, F_2, \ldots, F_h$. If $G = H$ then we are done by previous Lemma; otherwise we may grow H by selecting any edge r—s such that r is a vertex of H but s is not, and then applying $T_{s\,r}(G)$ (see case 3 given in proof of Lemma 4.3). Repeating this argument we see that G is equivalent to a $\mathbb{D}$-core of n vertices; therefore, it has Dynkin type $\mathbb{D}$. □

4.4. A simpler construction

As discussed earlier, construction 4.1 can be easily proved to yield bigraphs of Dynkin type $\mathbb{D}$. We suggest, however, another construction based on [12] with some minor differences. We will compare this constructions in the next subsection. Here, the length of a path or a cycle is the number of its edges.

Construction 4.7. ($\mathbb{A}$-cycling)

Items needed:

- A bigraph G of Dynkin type $\mathbb{A}$ with $n \geq 5$ vertices.

- Two internal vertices x, y of G such that the shortest path $x \rightsquigarrow y$ has length at least 2 and contains an odd number of dotted edges.

Procedure:
Identify x and y into a single vertex, and simplify by erasing pairs of dotted/solid parallel edges.

Before proving that the $\mathbb{A}$-cycling characterizes the Dynkin type $\mathbb{D}$ bigraphs, let us show first that this construction is well defined by proving that there is a unique shortest path between any pair of vertices in a bigraph of Dynkin type $\mathbb{D}$.

Lemma 4.8. Let G be a bigraph of Dynkin type $\mathbb{A}$, then the shortest path between any pair of vertices x and y is unique. Moreover, it is given by $x — s_1 — s_2 — \cdots — s_k — y$ where $s_1, s_2, \cdots, s_k$ are the separation vertices of G traversed by any simple $x \rightsquigarrow y$ path.

Proof:
Let P be any simple $x \rightsquigarrow y$ path and assume it traverses some vertex s; if s is a separation vertex, then all $x \rightsquigarrow y$ paths traverse s. Consequently, that all paths traverse the same separation vertices $s_1, s_2, \cdots, s_k$ as P in the same given order. Since each block of G is an $\mathbb{A}$-block (Theorem 3.1), then there exists unique edges $x — s_1$, $s_i — s_{i+1}$ for $i = 1, \ldots, k-1$, and $s_k — y$ which define the shortest path $x \rightsquigarrow y$. □

Theorem 4.9. A bigraph has Dynkin type $\mathbb{D}$ if and only if it is an $\mathbb{A}$-cycling.

Proof:
Let G' be the $\mathbb{A}$-cycling of some bigraph G of Dynkin type $\mathbb{A}$ with distinguished vertices u and v.

$\Rightarrow$) By Theorem 4.2 say that G is a $\mathbb{D}$-core gluing of $H = x_1 — x_2 — \cdots — x_h — x_1$ and $F_1, \ldots, F_h$ as in Construction 4.1. Let us replace x_1 by a new vertex x_{h+1} in F_h to obtain a new bigraph F'_h; then the bigraph sum $G := F_1 + F_2 + \cdots + F_{h-1} + F'_h$ has $n+1$ vertices, Dynkin type $\mathbb{A}$, and is such that x_1 and x_{h+1} are internal vertices while all other x_i are separation vertices. By Lemma 4.8, the shortest path $x_1 \rightsquigarrow x_{h+1}$ in G is precisely $x_1 — x_2 — \ldots — x_h — x_{h+1}$, which by definition of H has length at least 2, and an odd number of dotted edges. Finally, by construction of G, identifying $x_1 = x_{h+1}$ yields G'.

$\Leftarrow$) Let the shortest path P from u to v be $u = x_1 — x_2 — \cdots — x_h — x_{h+1} = v$. For $i = 1, \ldots, h$ let F_i denote the sub-bigraph of G induced by x_i, x_{i+1} and all vertices z such that there exists simple paths $x_i \rightsquigarrow z$ not containing x_{i+1}, and $x_{i+1} \rightsquigarrow z$ not containing x_i. By Lemma 4.8, each x_i $(2 \leq i \leq h)$ is a separation vertex of G and therefore, by identifying $x_1 = x_{h+1}$ we get a valid $\mathbb{D}$-core gluing of $H = x_1 — \cdots — x_h — x_1$ and $F_1, F_2, \ldots, F_h$. □

4.5. Comparison with previous results

We now briefly review the characterization of unit forms of Dynkin type $\mathbb{D}$ given in [12].

For any bigraph G let *the frame of* G, denoted as $\Phi(G)$, be the underlying graph of G i.e. the graph obtained by turning all dotted edges into solid ones. Barot's constructions characterize the frames of bigraphs of Dynkin type $\mathbb{D}$ as follows:

- A *mirror extension* of a graph Γ and a given vertex x is done by copying the adjacency list of x into a new vertex x^*; i.e. x^* is connected to the neighbours of x by the same number of edges as x.

- Barot's $\mathbb{A}$-*cycling* identifies two internal vertices x, y with distance $d(x, y) \geq 3$ selected from a tree assemblage Γ of complete graphs $K_{m_1}, \ldots, K_{m_t}$.

It is proven that any unit form $q : \mathbf{Z}^n \to \mathbf{Z}$ has Dynkin type $\mathbb{D}$ if and only if $\mathbf{bigr}(q)$ has $n \geq 4$ vertices, satisfies the cycle condition (defined in page 246), and its frame is a mirror extension or a Barot's $\mathbb{A}$-cycling. Notice that this characterization asks us to explicitly check the cycle condition on the given bigraph $G = \mathbf{bigr}(q)$. If done naively, one may enumerate all the chordless cycles of G, whereas our proposed characterizations ensure the cycle condition as a by-product and do not requiere such test.

Also, it is not difficult to see that the bigraphs obtained by a mirror extension are the same as those obtained by Construction 4.7 with $d(x, y) = 2$; or equivalently, where the $\mathbb{D}$-core H contains exactly 2 vertices (see examples on Figures 10 and 11). Consequently, there is no need to consider a version of the mirror extension which takes line style into account.

5. Computing the Dynkin type in polynomial time

In this section we are concerned on the computational complexity of computing the Dynkin type of a quasi-Cartan matrix and show that it can be solved in polynomial time. In the following, we study the time complexity of the Inflations Algorithm (Algorithm 1.1) and then suggest a polynomial time algorithm.

5.1. The complexity of the Inflations Algorithm

Let $\ell(\boldsymbol{A})$ denote the maximum length of a chain of inflations that can be applied to some positive definite $\boldsymbol{A} \in \mathbf{sqC}$, and let $\mathbf{N} = \{0, 1, 2, \ldots\}$ denote the set of natural numbers. Consider the set of *positive roots* of $\boldsymbol{A}$ given by $\mathcal{R}^+(\boldsymbol{A}) := \{\boldsymbol{x} \in \mathbf{N}^n | \mathbf{q}_{\boldsymbol{A}}(\boldsymbol{x}) = 1\}$ and the following facts:

(a). if $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ is a positive root of $\boldsymbol{A}$ then $0 \leq \boldsymbol{x} \leq 6$ (see [13] and [6, sec. 6.7])

(b). if $(\boldsymbol{A})_{sr} = 1$ then $\left|\mathcal{R}^+(\boldsymbol{A})\right| < \left|\mathcal{R}^+(T_{sr}(\boldsymbol{A}))\right|$ (see [6, sec. 6.2])

From (a) follows that $\left|\mathcal{R}^+(\boldsymbol{A})\right| \leq 7^n$ for all positive $\boldsymbol{A} \in \mathbf{sqC}$; and from (b) follows that the number of iterations done by the Inflations Algorithm is bounded by $\ell(\boldsymbol{A}) := \left|\mathcal{R}^+(\mathbf{biadj}(\Delta))\right| - \left|\mathcal{R}^+(\boldsymbol{A})\right| \leq 7^n$, where Δ is the Dynkin type of $\boldsymbol{A}$. Since the flation morphism can be implemented in $\mathrm{O}(n)$ operations, the time complexity of Algorithm 1.1 is at most $\mathrm{O}(n\,7^n)$. Notice that this complexity is defined only for inputs where $\boldsymbol{A}$ is positive definite.

Algorithm 5.1: A simple algorithm for computing the Dynkin type
Input: A matrix $\boldsymbol{A}$ with integer entries

```
test if A is positive definite symmetric quasi-Cartan matrix and exit if the test fails
for each connected component K of bigr(A) do
    if K has n ≤ 8 vertices then
        use the Inflations Algorithm on K to compute its Dynkin type and output it
    else
        test if K has Dynkin type A_n or D_n and output the result
```

5.2. Outline of the suggested algorithm

The Algorithm 5.1 is a suggested for computing the Dynkin type. The idea is that if we can guarantee that some connected bigraph K is positive definite, then it must have a Dynkin type $\mathbb{A}$, $\mathbb{D}$ or $\mathbb{E}$; more over, if K has $n > 9$ vertices, then we can exclude $\mathbb{E}$.

In [14] the authors showed that it is possible to decide if a (not necessarily positive definite) bigraph on n vertices and m edges has Dynkin type $\mathbb{A}$ in $\mathrm{O}(m+n)$ steps by using a Depth-First transversal. A similar test for $\mathbb{A}$ that uses the fact that $\boldsymbol{A}$ is positive definite is presented below. The rest of this section is devoted to fill in the gaps.

5.3. Testing for positive-definiteness of symmetric quasi-Cartan matrices

We begin addressing the task of checking if a symmetric $n \times n$ quasi-Cartan Matrix is positive definite. A common algorithm that works for any symmetric real matrix reduces the input matrix to its triangular form by Gaussian elimination and then announces whether or not all of the pivots where positive. The correctness of this algorithm can be found in many basic linear algebra texts such as [15, sec. 6.2]. The arithmetic complexity of Gaussian elimination is $\mathrm{O}\left(n^3\right)$ although a naive implementation can lead to an exponential bit-complexity.

Example 5.1. Apply the classic integer-preserving Gaussian elimination [16, eq. II.6] to **biadj** $(\mathbb{A}_n)$:

$$\begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & -1 & \ddots & \ddots & \\ & & & \ddots & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix} \mapsto \begin{bmatrix} 2 & -1 & & & & \\ & 4 & -2 & & & \\ & & 16 & -8 & & \\ & & & \ddots & \ddots & \\ & & & & 2^{2^{n-2}} & -2^{(2^{n-2}-1)} \\ & & & & & 2^{2^{n-1}} \end{bmatrix}$$

Notice how the bit-length of the matrix entries grow exponentially. Thus, arithmetical operations cannot be regarded as elementary and the algorithm takes exponential time.

Because of this, it is a good idea to work with the one-step fraction-free Gaussian eliminitation of Bareiss (see [16, eq. II.9]). The bit-complexity of this algorithm is $\mathrm{O}\left(n^5 (L + \log n)^2\right)$ where $2^L \geq \left|(\boldsymbol{A})_{ij}\right|$ for all i, j (see [17, sec. 10.2] for a proof). Algorithm 5.2 shows a natural adaptation to solve our problem. Since all of the absolute values of the entries of any positive-definite symmetric quasi-Cartan matrix are bounded by 2, we set $L = 1$ and get a bit-complexity of $\mathrm{O}\left(n^5 log^2 n\right)$. Note that this algorithm destroys the input matrix, so the input should be a *copy* of $\boldsymbol{A}$.

5.4. Bigraph decomposition by depth-first search

The depth-first search [11] is an elementary graph algorithm that traverses all the vertices of a graph in linear time using the following rule: *discover any vertex that is adjacent to the most recently discovered vertex provided that it have not been previously discovered.* Common applications include decomposing a graph into connected components, and connected graphs into blocks and separation vertices (see [11] and [10, sec. 6.1]). We review such applications briefly. The reader is referred to [18, sections 5.8 and 5.9] for implementation details.

Algorithm 5.2: Using Bareiss algorithm to decide whether a matrix is positive-definite

Input: A $n \times n$ symmetric matrix $\boldsymbol{Q}$ with integer entries

for $k = 1$ **to** n **do**
  for $i = k + 1$ **to** n **do**
    for $j = k + 1$ **to** n **do**
      if $(\boldsymbol{Q})_{k\,k} \leq 0$ **then return** `False`
      $(\boldsymbol{Q})_{i\,j} := \det \begin{bmatrix} (\boldsymbol{Q})_{k\,k} & (\boldsymbol{Q})_{k\,j} \\ (\boldsymbol{Q})_{i\,k} & (\boldsymbol{Q})_{i\,j} \end{bmatrix}$
      if $k > 1$ **then** $(\boldsymbol{Q})_{i\,j} := \frac{(\boldsymbol{Q})_{i\,j}}{(\boldsymbol{Q})_{k-1\,k-1}}$ `// This division is guaranted to be exact`

return `True`

Algorithm 5.3 below shows the basic structure of the depth-first search. Given a graph G and a vertex u, the Algorithm recursively marks all the vertices reachable from u as *visited*. Lines 2 and 7 are meant to define the operations to be executed when vertex u is discovered for the first time or when its exploration has finished. The recursion tree defines a rooted *depth-first tree* T with directed edges as given in line 5. Applications of depth-first search arise by properly defining the pre-process, edge process, and post-process operations.

Algorithm 5.3: Depth-first visit

Input: A graph $G = (V, E)$ and a vertex u

1 mark u as visited
2 pre-process u
3 **for each** *edge* u—v *of* G **do**
4   **if** v *is not marked as visited* **then**
5   process the tree-edge $u \longrightarrow v$
6   recursively depth-first visit v in G
7 post-process u

In order to compute the connected components of a graph G use the fact that Algorithm 5.3 explores exactly the vertices that are in the same connected component as u. Consequently, it suffices to loop over all vertices and call Algorithm 5.3 on the ones that are not already marked as visited. Each call picks out an new connected component.

Next, we describe how blocks and separations vertices are computed from a connected graph. A vertex u is said to the *parent* of v in T if there exists an edge $u \longrightarrow v$ in T. The closure of the parent relation is the *ancestor* relation. *Children* and *successor* relations are defined in an analogous manner. Notice that all non-tree edges connect two related vertices (one is an ancestor of the other). A *back edge* $v \longrightarrow u$ is a non-tree edge connecting a vertex v with its ancestor u, as in Fig. 13.

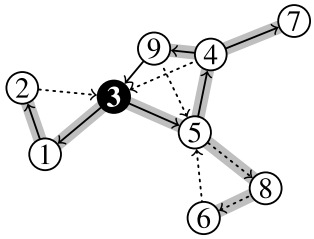


Figure 13. A depth-first tree on the bigraph of Figure 6. The root vertex and tree edges are highlighted.

As a consequence, u is a separation vertex if and only if one of the following conditions holds:

- u is the root vertex of T and has more than one child.
- u is a non-root vertex and has a child v none of whose descendants (including itself) has a back-edge to a proper ancestor of u.

Let $\texttt{pre}\,(u)$ denote a numbering of vertices in the same order as they were discovered, and let $\texttt{low}\,(u)$ be the smallest $\texttt{pre}\,(w)$ such that there is a non-proper descendant v of u with a back-edge $v \longrightarrow w$. Clearly, $\texttt{pre}\,(u)$ can be computed during the pre-processing of u, and $\texttt{low}\,(u)$ during its post-processing by choosing the minimum of $\texttt{pre}\,(u)$, $\texttt{low}\,(v)$ for all tree-edges $u \longrightarrow v$, and $\texttt{pre}\,(w)$ for all back-edges $u \longrightarrow w$. Thus, a non-root vertex u is a separation vertex if and only if it has a child v such that $\texttt{low}\,(v) \geq \texttt{pre}\,(u)$. Using stack of edges one can remove the blocks of G one at a time.

For n vertices and m edges, this procedures can be implemented in $\mathrm{O}\,(n+m)$ time [11].

5.5. Testing for Dynkin type $\mathbb{A}$

Theorem 3.1 can be directly implemented in three stages: decompose the bigraph G into blocks and separation vertices; check that all blocks are in $\mathcal{F}$; check that each separation vertex appears in exactly two blocks. However, this test is to be performed on positive definite bigraphs only; thus, we may simplify it as described below.

A *clique* in a graph G is a subgraph C such that every two distinct vertices of C are adjacent. A clique on k vertices is called a k-clique, and a *clique tree* is a type of graph in which every block is a clique. For the next lemma recall that $\Phi\,(G)$ denotes the underlying graph of any bigraph G.

Lemma 5.2. Let $\boldsymbol{A} \in \mathbf{sqC}$ be positive definite. Then $\boldsymbol{A}$ has Dynkin type $\mathbb{A}$ if and only if $\Phi\,(\mathbf{bigr}\,(\boldsymbol{A}))$ is a clique tree and each separation vertex separates exactly two blocks.

Proof:
$\Rightarrow$) Immediate by Theorem 3.1 and noticing that the underlying graph of $\mathbf{F}\,[X, Y]$ is complete.

$\Leftarrow$) It suffices to show that each clique of G is a member of $\mathcal{F}$; we do so by induction: Clearly, each edge u — v of G is the 2-clique $\mathbf{F}\,[\{u\},\{v\}]$ or $\mathbf{F}\,[\{u, v\}, \emptyset]$ For the induction step, let C be a $(k+1)$-clique of $\Phi\,(G)$ and let w be a vertex of C. Then $C' := C - w$ is a k-clique of $\Phi\,(G)$ and by I.H. the bigraph G restricted to C' is $\mathbf{F}\,[X, Y]$ for some X, Y. Recall that $\boldsymbol{A}$ is positive definite; then, by Silverter's Criterion [15, sec. 6.2] each principal minor of $\boldsymbol{A}$ must also be positive-definite. Notably, G cannot contain any triangle (3-clique) with an even number of dotted edges, for otherwise the submatrix of $\boldsymbol{A}$ given by such three vertices would not be positive-definite. In order to avoid triangles with an even number of dotted edges one of this conditions must hold:

(a). w is connected by dotted edges to each vertex of X and by solid edges to each vertex of Y; thus, $C = \mathbf{F}[X \cup \{w\}, Y]$.

(b). w is connected by solid edges to all vertices of X and by dotted edges to all vertices of Y; consequently, $C = \mathbf{F}[X, Y \cup \{w\}]$

□

Stage 2 is therefore reduced to checking that each block with k vertices has $\binom{k}{2}$ edges. Finally, stage 3 can be implemented by applying the following test to each separation vertex s:

- s is the root vertex and has exactly two children
- s is a non-root vertex and has exactly one children v such that $\mathtt{low}(v) \geq \mathtt{pre}(s)$.

These tests can be implemented during the post-processing of vertices in the depth-first search without increasing the running time.

5.6. Analysis of the suggested algorithm

Let us briefly analyse Algorithm 5.1. We have seen that the Inflations Algorithm runs in $\mathrm{O}(n\,7^n)$ steps. However, line 4 takes constant time since this algorithm is called for $n \leq 8$ only. As discussed above, line 1 of Algorithm 5.1 runs in $\mathrm{O}\left(n^5 log^2 n\right)$ steps. The rest of the algorithm can be implemented in $\mathrm{O}\left(n^2\right)$ steps by applying depth-first search on G two times: the first is to get the connected components and the second (if needed) to test for $\mathbb{A}$ in each component. Clearly, line 1 is the bottleneck of Algorithm 5.1; therefore, the running time is $\mathrm{O}\left(n^5 log^2 n\right)$.

References

[1] Knapp AW. Lie Groups Beyond an Introduction. vol. 140 of Progress in Mathematics. 2nd ed. Birkhäuser; 2002. Available from: http://www.springer.com/book/978-0-8176-4259-4.

[2] Barot M, Geiss C, Zelevinsky A. Cluster algebras of finite type and positive symmetrizable matrices. Journal of the London Mathematical Society. 2006;73(3):545–564. Available from: http://dx.doi.org/10.1112/S0024610706022769.

[3] Barot M, Rivera D. Generalized Serre relations for Lie algebras associated with positive unit forms. Journal of Pure and Applied Algebra. 2007;211:360–373. Available from: http://dx.doi.org/10.1016/j.jpaa.2007.01.008.

[4] Ovsienko SA. Integer weakly positive forms. Schurian Matrix problems and quadratic forms. 1978;p. 3 – 17.

[5] Ringel CM. Tame Algebras and Integral Quadratic Forms. vol. 1099 of Lecture Notes in Mathematics. Springer-Verlag Berlin Heidelberg; 1984. doi:10.1007/BFb0072870.

[6] Gabriel P, Roiter AV. Representations of Finite-Dimensional Algebras. vol. 73 of Encyclopaedia of Mathematical Sciences. Springer; 1997. Available from: http://www.springer.com/book/978-3-540-53732-8.

[7] Kosakowska J. Inflation Algorithms for Positive and Principal Edge-bipartite Graphs and Unit Quadratic Forms. Fundamenta Informaticae. 2012;119(2):149–162. Available from: http://content.iospress.com/articles/fundamenta-informaticae/fi119-2-02.

[8] Barot M. A characterization of positive unit forms. Boletín de la Sociedad Matemática Mexicana. 1999; 5:87–94.

[9] Multiset. Encyclopedia of Mathematics;. Available from: `http://www.encyclopediaofmath.org/index.php?title=Multiset&oldid=37512`.

[10] Bondy A, Murty USR. Graph Theory. vol. 244 of Graduate Texts in Mathematics. Springer; 2008. Available from: `http://www.springer.com/book/978-1-84628-969-9`.

[11] Hopcroft J, Tarjan R. Algorithm 447: Efficient algorithms for graph manipulation. Communications of the ACM. 1973;16(6):372–378. doi:10.1145/362248.362272.

[12] Barot M. A characterization of positive unit forms, part II. Boletín de la Sociedad Matemática Mexicana. 2001;7:13–22.

[13] Ovsienko SA. Boundedness of roots of integral weakly positive forms. In: Representations and quadratic forms (Russian). Akad. Nauk Ukrain. SSR, Inst. Mat., Kiev; 1979. p. 106–123, 155.

[14] Abarca M, Rivera D. Formas Cuadráticas Unitarias de tipo $\mathbf{A}_n$: Un Enfoque Combinatorio. Programación Matemática y Software. 2014;6(2):17–25. Available from: `http://www.progmat.uaem.mx:8080/vol6nu2ar3.html`.

[15] Strang G. Linear Algebra and Its Applications. Brooks Cole; 2005. ISSN: 0024-3795.

[16] Bareiss EH. Sylvester's Identity and Multistep Integer-Preserving Gaussian Elimination. Mathematics of Computation. 1968;22(103):565–578. Available from: `http://www.jstor.org/stable/2004533`.

[17] Yap C. Fundamental Problems in Algorithmic Algebra. Oxford University Press; 2000. Available from: `http://cs.nyu.edu/yap/book/berlin/`.

[18] Skiena SS. The algorithm design manual: Text. vol. 1. Springer Science & Business Media; 1998. ISBN: 0387948600, 9780387948607.