BIPARTITE GRAPHS, QUIVERS, AND CLUSTER VARIABLES

IN-JEE JEONG

Abstract. We explore connections between formulas for certain combinatorial and algebraic objects. In particular, given a planar bipartite graph G, we consider the cluster algebra $\mathcal{A}$ corresponding to a quiver obtained from its dual graph. We then obtain formulas for certain cluster variables in $\mathcal{A}$ in terms of perfect matchings of subgraphs of G. Such subgraphs look like trees; locally they look like snakes with bridges connecting them. In the case of a spine snake with some trees attached to its edges we can obtain combinatorial formulas through superpositions.

1. Introduction

Fomin and Zelevinsky, in [3, 4], created a new mathematical object called a cluster algebra. Following their definitions, we start with an initial cluster (set of variables) $\{x_1, x_2, ..., x_n\}$ of a cluster algebra $\mathcal{A}$ and mutate it to obtain more variables. It turns out that in certain types of cluster algebras, cluster variables can be obtained by weights of graphs, as demonstrated in [8, 9, 10].

As described in [2], there is a graphical way of formulating the rules for mutation. For each cluster $\mathcal{C}$ of $\mathcal{A}$, we consider a quiver (directed graph) $Q_{\mathcal{C}}$ such that mutation of $\mathcal{C}$ corresponds to mutation of $Q_{\mathcal{C}}$. In particular, it was shown in [12] that if we start with x_i at the ith vertex in the quiver below and mutate at 1,3,2,4,1,3,2,4,..., we get the weights of Aztec diamond graphs with an appropriate weighting scheme. Gregg Musiker conjectured that if we mutate at 1,4,3,4,3,4,..., we get the weights of a family of graphs shown below. In particular, if we begin with ones instead of variables and mutate in that sequence, we will get the number of perfect matchings of these graphs. When we were proving this, we realized that we could get families of graphs which have more squares attached to both branches from the horizontally unfolded version of this quiver. All of these graphs looks like a spine with branches attached. With this motivation, we studied 1) which graphs qualify as a spine and 2) what is the most general type of branch we can attach to such a spine.

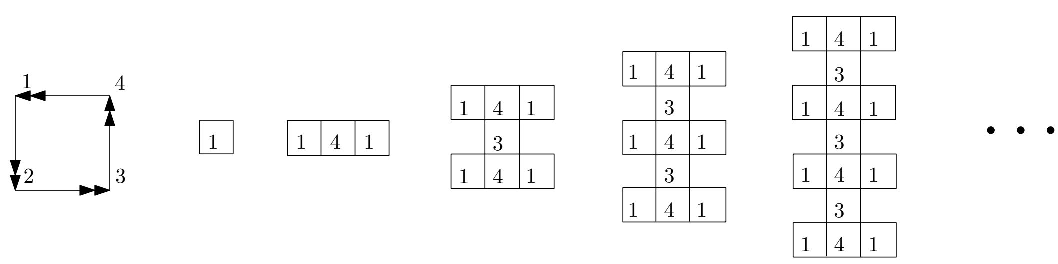


Figure 1.1. Musiker's Conjecture

It turns out that when we have a planar bipartite graph, its dual graph can be turned into a quiver where variables we get from certain mutation sequences have combinatorial interpretations in our original graph. These interpretations can be merged or superimposed for mutation sequences that meet the required conditions. In this paper, we describe simple instances when those combinations of combinatorial interpretations work well.

Our first main result is Theorem 3.2, which using the definitions in Section 2, says.

Theorem 3.2. *Any natural mutation sequence associated with a tree $\mathcal{T}$ carries $\mathcal{T}$.*

It turns out that in certain cluster algebras, all noninitial variables represent weights of trees embedded in a 2-connected planar bipartite graph. In other cases, trees cover only a few variables. This has intersections with Musiker's results in [8] although we do not recover all of his results. After developing more definitions and notations in the next section, we state and prove our second result. This is a generalization of the above conjecture (see Theorem 3.5).

The outline of this paper is as follows. In Section 2, we develop and discuss several concepts which we will need throughout the paper. Then we prove two phenomena in Section 3. In the last section, we first apply our results to infinite grid quivers. After that we consider various finite quivers; in particular, we show that given a tree quiver, there is a corresponding planar bipartite graph. Quivers of type A_n and D_n from [8] fit in this family. Also, superposition recovers a result from [9]. Finally, we attempt to combinatorially represent Markoff polynomials studied in [11] within our framework.

2. Preliminaries

We now define some terminology that we will use throughout this paper.

Definition 2.1. If G is a 2-connected planar bipartite graph with its vertex set $V = E_1 \cup E_2$, consider its dual graph H. We turn H into a quiver so that, near every vertex of E_1, we have arrows rotating clockwise around that vertex, and counterclockwise for E_2. This is possible since G is bipartite. After killing all resulting 2-cycles one by one, we call the resulting quiver the **dual quiver** of G, $\bar{G}$.

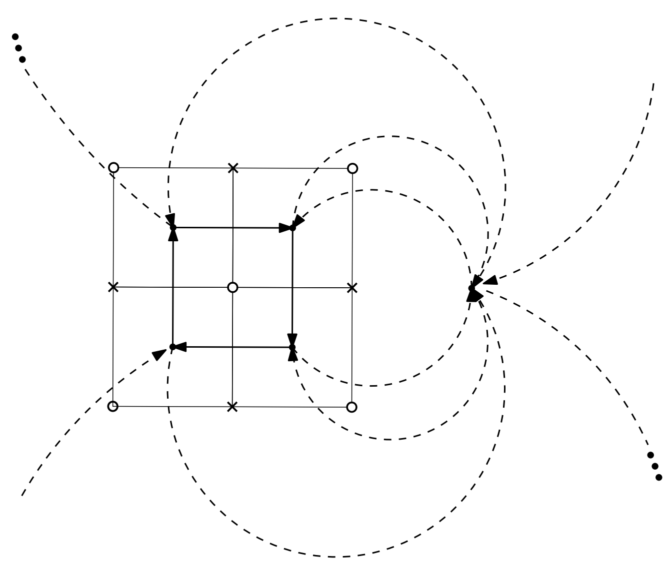


Figure 2.1. Here, G equals the union of four squares. All above dashed arrows are involved in 2-cycles; hence $\bar{G}$ equals a four cycle plus one disconnected vertex.

Here, 2-connectedness means G is connected and remains connected if any edge were to be removed. This will make every face including the possible infinite faces of G to be well-defined.

So when we consider its dual, these faces of G will be vertices of $\bar{G}$, and edges will be arrows connecting these vertices. One can observe that in $\bar{G}$, (before killing 2-cycles) faces are well defined and arrows are cyclic in every face. Also every edge of G will be shared by two distinct faces of G, and a 2-connected graph will be bipartite if and only if each face has even number of edges. Throughout this paper, given a graph G, we will be looking at various subgraphs of G and it is indeed enough that such subgraphs are 2-connected, bipartite, and can be embedded into a plane so that their dual quivers are well defined. However, it is certainly convenient to work with G that satisfies all such properties.

Given G, we attach a cluster variable x_α to every vertex α of $\bar{G}$ (which is a face of G), obtaining a cluster algebra $\mathcal{A}_G = \{x_\alpha\}_{\alpha \in \bar{G}}$. When a vertex α is mutated for the first time, we name new cluster variable y_α. By a neighborhood (or neighbor) of a subgraph H of G (which is a union of faces of G) we mean the set of all faces in G that is not a face of H and share at least one edge with that subgraph. We use Kenyon's edge weighting from [5]; $w(e) = 1/x_\alpha x_\beta$ where α, β are vertices correspond to two faces in G sharing e. Then the weight of a perfect matching will be the product of weights of all the edges in that matching, and the weight of a graph is the sum of the weights over all possible matchings.

To **localize** at a set of vertices of a dual quiver means that we will be looking at only those vertices and arrows attached to them. We adopt a convention that unless otherwise specified, we let the face in G corresponding to a vertex v of $\bar{G}$ be F_v. Similarly, if we have a face F in G, we call the corresponding vertex in $\bar{G}$ v_F and the variable initially attached to this vertex x_{v_F}. When we mutate at v_F for the first time, we will call the new variable y_{v_F}.

Definition 2.2. Here we define a collection of subgraphs of G which we call **tree**s, as follows:

- A tree $\mathcal{T}$ is a finite union of partially ordered faces of G where no two faces in $\mathcal{T}$ share more than one edge.
- This partial order on $\mathcal{T}$ satisfies the following conditions:
  - there is a unique maximal face T
  - a face A directly covers (or directly covered by) B if and only if A and B share an edge.
  - any non maximal face of $\mathcal{T}$ is directly covered by exactly one other face, and when it directly covers several faces, it is called a bridge face.
- If $A < B$ are two faces in $\mathcal{T}$, then A and B can share at most one vertex if for every C such that $A < B < C$, C has that vertex. If two faces are not comparable, they are disjoint with each other.

We define the **order** of a tree $\mathcal{T}$ as the number of faces in $\mathcal{T}$. We allow a single edge of G as an order zero tree. If we have a tree $\mathcal{T}$, we can consider the set of vertices in $\bar{G}$ which corresponds to all the faces in $\mathcal{T}$. Then we say a mutation sequence in $\bar{G}$ is **natural** with respect to $\mathcal{T}$ if this sequence contains all vertices in that set exactly once and whenever $A < B$ (two faces of $\mathcal{T}$), v_B appears before v_A in the sequence. That is, a natural sequence respects the ordering on the tree. We say two trees are **commuting** if they are disjoint except possibly at one vertex of their maximal faces. We call $\mathcal{T}$ a **snake** if it does not have any bridge faces. Then we can also visualize a tree as a subgraph consisting of bridge faces and snakes connecting them.

When we say mutate at a tree $\mathcal{T}$ or a subgraph of $\mathcal{T}$ we mean we mutate in any natural mutation sequence or its subsequence in $\bar{G}$. From our definition of a tree, the set of variables we obtain when we mutate in any natural sequence should be the same. In particular, y_{v_T} (the last variable) is the same. Also if $\mathcal{T}_1$ and $\mathcal{T}_2$ commutes then mutation at $\mathcal{T}_1$ and $\mathcal{T}_2$

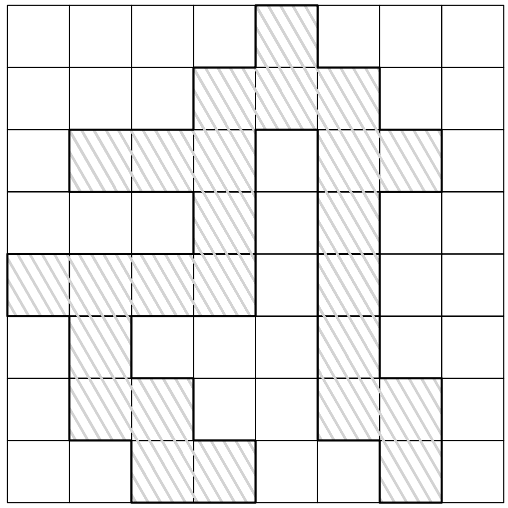


FIGURE 2.2. A tree living inside a finite square grid; any face qualifies as the maximal face when we order other faces with the distance from that face.

commutes simply because they do not share any edges so there is no arrow between them in $\bar{G}$. Similarly, mutating at a face F of $\mathcal{T}$ will mean mutating at the corresponding vertex v_T with the assumption that we have mutated at all faces A of $\mathcal{T}$ such that $A < F$.

Definition 2.3. (separation and merging) Let $\mathcal{T}$ be a tree and U be a neighbor face. If U shares multiple edges $f_1, f_2, ...f_t$ with $\mathcal{T}$ we may **separate** U into U' (which we will usually identify as U) and $V_1, V_2, ...V_t$ where all V_i are small quadrilaterals with one edge f_i and other three edges completely embedded in U'. We will refer to V_i as **parts** of U. This procedure is as follows: 1) We recover all 2-cycles killed before separation, so an arrow between U and $\mathcal{T}$ will exist for each edge of $\mathcal{T}$ shared with U. As soon as we draw V_1, we have three new arrows between U (or U') and V_1 in our dual quiver, so two of them kill each other. The arrow between U and $\mathcal{T}$ is now split into two arrows of the same direction, one between U and V_1 and the other between V_1 and $\mathcal{T}$. All other arrows remain the same. After we draw all $V_1, ..., V_t$, U does not share any edges with $\mathcal{T}$, V_i shares f_i with $\mathcal{T}$, and in our dual quiver all arrows are preserved except those described above. We will identify $x_{v_U} = x_{v_{U'}} = x_{v_{V_1}} = ... = x_{v_{V_t}}$ for convenience. **Merging** is the inverse process. So when we say we merge V back to U it is implicitly assumed that U used to contain V.

Definition 2.4. If $\mathcal{T} = \cup_{1\leq i\leq n} F_i$ is a tree in G, we define its **covering monomial** to be the product

$$m(\mathcal{T}) = \prod_{i=1}^{n} x_i^{f_i} \prod_{U\in\mathcal{U}} x_{v_U}^{g_U},$$

where $\mathcal{U}$ is the neighbor of $\mathcal{T}$, and f_i =(number of edges of F_i−2)/2.

To determine g_U, we localize at the set $\{1, 2, ..., n, v_U\}$ of vertices of $\bar{G}$ which corresponds to the set of faces $\{F_1, F_2, ...F_n, U\}$ in G and recover all 2-cycles. A general cycle at this localization will look like

$$C_i : U \to F_{i_1} \to F_{i_2} \to ... \to F_{i_{k(i)}} \to U$$

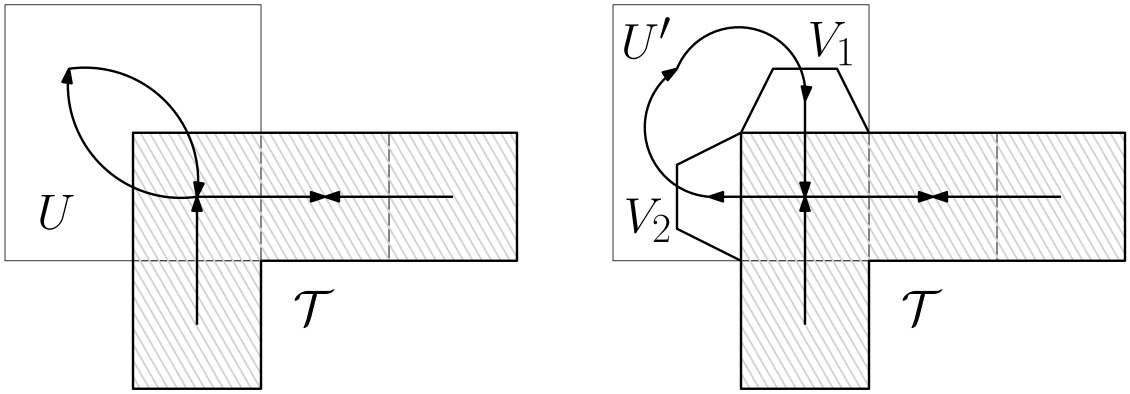


FIGURE 2.3. An example of separation

(or all arrows in the opposite direction). We give a partial ordering on these cycles by declaring the order of C_i to be the order of $\max_{1\le l\le k(i)} F_{i_l}$. Assume that there are two maximal faces, say F_{i_q} and F_{i_p}. $(q < p)$ Then $F_{i_{q+1}} \lessdot F_{i_q}$ (F_{i_q} directly covers $F_{i_{q+1}}$) and $F_{i_{p-1}} \lessdot F_{i_p}$ since these pairs share an edge respectively. And $F_{i_{q+2}} \lessdot F_{i_{q+1}}$ since $F_{i_{q+1}}$ cannot be directly covered by multiple faces. Likewise $F_{i_{p-2}} \lessdot F_{i_{p-1}}$. Continuing this way, we will have some face F_{i_k} $(q < k < p)$ directly covered by two faces $F_{i_{k-1}}$ and $F_{i_{k+1}}$; this is a contradiction. So $\max_{1\le l\le k(i)} F_{i_l}$ is well-defined.

Kill all 2-cycles one by one. After that, pick one of the minimal cycles and get rid of two opposite arrows (killing) between U and $\mathcal{T}$. If we still see cycles, repeat this until we have no cycles. Then g_U equals the sum of the number of all cycles killed (including 2-cycles) and the number of remaining arrows between U and $\mathcal{T}$. (not between faces of $\mathcal{T}$) Equivalently g_U equals the number of edges shared by U and $\mathcal{T}$ minus the number of cycles killed.

We will refer to those cycles (including 2-cycles) killed in this procedure as **U-cycles**. So from the definition of U-cycles, we have a specific sequence of killing minimal cycles one by one. We will not show here that the number of U-cycles for each neighborhood U of $\mathcal{T}$ is independent on the choice of killing sequence but it will be obvious once we prove $y_{v_T} = m(\mathcal{T})w(\mathcal{T})$ since neither $w(\mathcal{T})$ nor y_{v_T} depends on the choice of the killing sequence.

Remark 2.5. More general covering monomials

We can extend the definition this monomial to cover more general family of subgraphs other than trees. Assume that we have a subgraph $H = \cup_{i\in I} F_i$ in G and a partial ordering on F_i such that for any neighboring face U, whenever we have a cycle $U \to F_{i_1} \to F_{i_2} \to ... \to F_{i_{k(i)}} \to U$ in $\bar{G}$, $\max_{1\le l\le k(i)} F_{i_l}$ is well-defined. Then we can define its covering monomial in the same way.

We will say that a sequence of vertices $v_1, v_2, ...v_n$ of $\bar{G}$ **carries** a tree $\mathcal{T} = \cup_{1\le i\le m} F_i$ if $w(\mathcal{T})m(\mathcal{T})$ equals the nth variable we get after we mutate the quiver in order of that sequence. When the mutation sequence is understood, we will say v_n (the last term of the sequence) or F_m (maximal face of $\mathcal{T}$) carries $\mathcal{T}$. When $\mathcal{T}$ is not a tree but if its covering monomial is well-defined and there exists a mutation sequence in $\bar{G}$ such that the last variable equals $w(\mathcal{T})m(\mathcal{T})$, we will also say that this sequence carries $\mathcal{T}$.

Remark 2.6. Speyer's weighting

There is an earlier face weighting scheme which appeared in [12]. Consider a connected subgraph H of G and name the edges of the outer face of H as $e_1, e_2, ...e_t$ clockwise. To

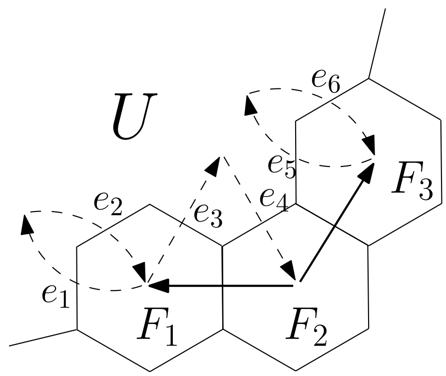


FIGURE 2.4. U shares edges from e_1 to e_6 with a tree; hence we have three U-cycles

apply Speyer's weighting, it is required that this outer face is "partitioned" by neighbor faces; that is, if a neighbor face U contains e_i and e_j then U contains all e_k which lies in either the counterclockwise or clockwise path from e_i to e_j. There are no such restrictions in our original setting. With this assumption, let M be a matching of $H = \cup_{i=1}^n F_i$. For each i, let b be the number of edges of F_i not in M and a be the number of edges of F_i in M. For a neighbor face U of H, let b be the number of edges of $U \cap H$ not in M and a be the number of edges of $U \cap H$ in M. Then Speyer's weighting equals

$$\bar{w}(M) = \prod_{i=1}^{n} x_i^{h_i} \prod_{U \in \mathcal{U}} x_{v_U}^{d_U}$$

where $h_i = \lceil \frac{b-a}{2} \rceil - 1$ and $d_U = \lceil \frac{b-a}{2} \rceil$.

Then we set $\bar{w}(H) = \sum_M \bar{w}(M)$ where this sum is over all matchings of H. Then we claim $\bar{w}(M) = w(M)m(H)$ for a given matching M. First we compare exponents of x_i for $i = 1...n$. We have x_i^{-a} in $w(M)$ and $x_i^{(a+b)/2-1}$ in $m(H)$ so $w(M)m(H)$ contains $x_i^{(b-a)/2-1} = x_i^{h_i}$. For a neighbor face U, due to our assumption, all U-cycles are short: If U shares edges $e_i, ...e_j$ then for each e_k, $(k = i...j-1)$ a cycle crossing e_k and e_{k+1} is formed. Hence we observe $\lfloor (b+a)/2 \rfloor$ U-cycles and $g_U = b + a - \lfloor (b+a)/2 \rfloor = \lceil (b+a)/2 \rceil$. And we have $x_{v_U}^{-a}$ in $w(M)$ so we also have $x_{v_U}^{d_U}$ on the right hand side. This is true for all M, so $\bar{w}(H) = w(H)m(H)$. Therefore, whenever an equality like $y = w(H)m(H)$ holds where the outer face of H is partitioned, it implies $y = \bar{w}(H)$.

3. RESULTS

3.1. **Tree Phenomenon.** Our first goal is to show that any natural mutation sequence associated to a tree $\mathcal{T}$ carries $\mathcal{T}$ with the covering monomial defined above. We make some notations to better deal with trees.

We say $\mathcal{T}$ is **extendable** to a face F of G if $\mathcal{T} \cup F$ qualifies as a tree with F being the new maximal face and all other order relations being preserved. In this setting, let a and b be two vertices shared by F and $\mathcal{T}$ and let $\mathcal{T}^* = \mathcal{T} \backslash \{a, b\}$ denote the subgraph of $\mathcal{T}$ obtained from $\mathcal{T}$ by getting rid of a, b and all edges incident to them ($\mathcal{T}$ **star** with respect to F). The weight $w(\mathcal{T}^*)$ is well-defined; when $\mathcal{T}$ is an order zero tree, $\mathcal{T}^*$ is empty so its weight is 1. We mutate at $\mathcal{T} \backslash T$ and localize at T. We always have an arrow between v_F and v_T; assume $v_F \to v_T$. Then consider all arrows $v_T \to v_\alpha$ and define $y_{v_T}^*$ be the product of all variables carried by

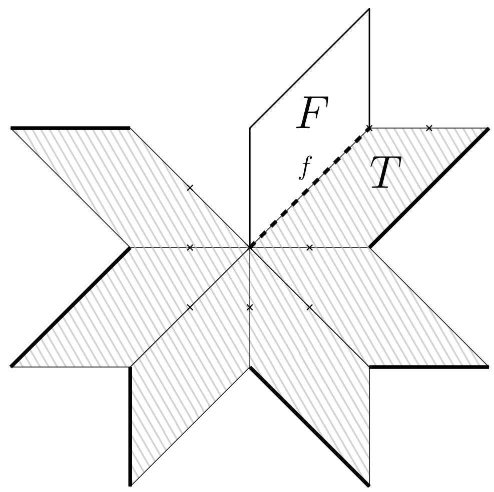


FIGURE 3.1. This tree $\mathcal{T}$ (shaded region) with maximal face T is extendable to F. Notice that there is only one matching of $\mathcal{T}^*$ (thicker edges) so $w(\mathcal{T}^*)$ equals the product of weights of six floating edges.

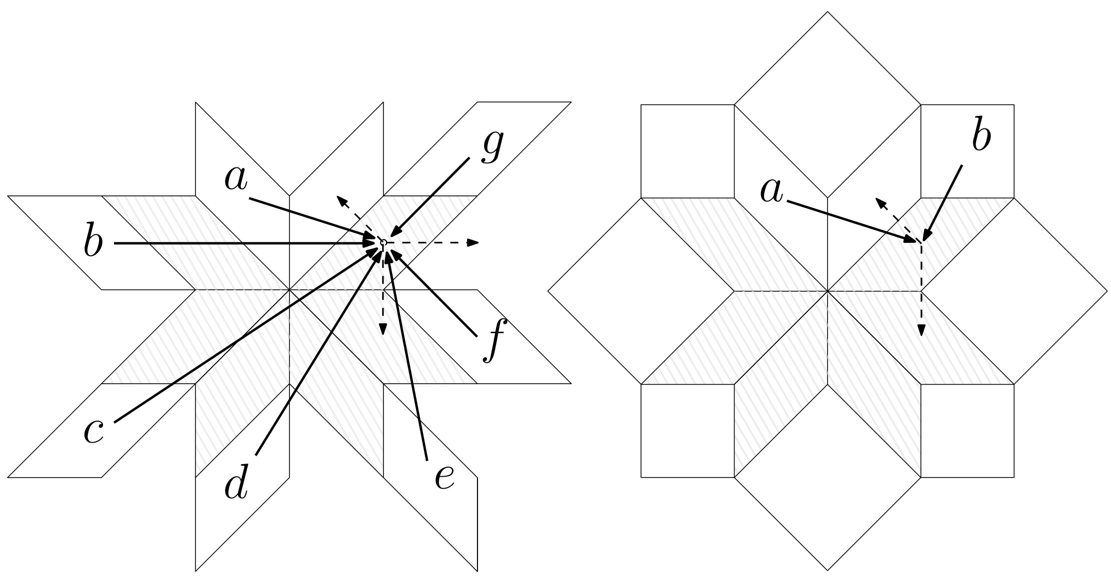


FIGURE 3.2. Examples of localization at T: $y^*_{v_T} = x_a x_b x_c x_d x_e x_f x_g$, $y^*_{v_T} = x_a x_b$ (left, right)

such v_α with correct multiplicities. Now mutate at T and localize at F; we will have arrows $v_F \to v_\alpha$ from all such v_α. Together with the arrow $v_F \leftarrow v_T$ (reversed after mutating at T), we say these arrows are "originated" from $\mathcal{T}$. As a special case, if $\mathcal{T}$ is an order zero tree e (an edge), set the other face containing e to be T and define $y^*_{v_T} = x_{v_T}$ and $y_{v_T} = 1$. This convention makes sense since we have not mutated at T. Notice that $w(\mathcal{T}^*)$ is not affected by the neighbor of $\mathcal{T}$ but $y^*_{v_T}$ is highly affected by it. Naively, at this localization we expect arrows from neighbor faces of $\mathcal{T}$ which respectively contains a prohibited edge of $\mathcal{T}^*$. This is no longer true when a neighbor face contains several edges of $\mathcal{T}^*$.

Unless $\mathcal{T}$ is an order zero tree, let T be the maximal face of $\mathcal{T}$ with edges $e_1, ...e_{2m}$ (clockwise) and consider $\mathcal{T}\backslash T$. Let $\mathcal{T}_i$ be the maximal subtree of $\mathcal{T}\backslash T$ attached to e_i. If we set $F_i \neq T$ be the face containing e_i, then the maximal face of $\mathcal{T}_i$ (unless it equals e_i) is F_i. Two faces $F_i = F_j$ for $i \neq j$ is possible only when $\mathcal{T}_i = e_i$ and $\mathcal{T}_j = e_j$. For convenience, we set the vertex corresponding to F_i in $\bar{G}$ i (so it carries x_i and y_i). We also have $\mathcal{T} = T \cup (\cup_{i=1}^{2m} \mathcal{T}_i)$ and from our definition of a tree, we observe that $\mathcal{T}_i$ are all commuting with each other. Also each subtree $\mathcal{T}_i$ is extendable to T. Let v_i and v_{i+1} (modulo $2m$) be two vertices (clockwise) of e_i. Set $\mathcal{T}_i^* = \mathcal{T}_i \backslash \{v_i, v_{i+1}\}$ for all i. Then we have the following lemma.

Lemma 3.1. *Let $\mathcal{T}$ be a tree of positive order. Then a matching of $\mathcal{T}$ either comes from a set of matchings of $\{\mathcal{T}_i, \mathcal{T}_j^*\}$ or $\{\mathcal{T}_j, \mathcal{T}_i^*\}$ where i ranges over all even and j all odd between 1 and $2m$. Therefore,*

$$w(\mathcal{T}) = \prod_{j=1}^{m} w(\mathcal{T}_{2j}) w(\mathcal{T}_{2j-1}^*) + \prod_{i=1}^{m} w(\mathcal{T}_{2i-1}) w(\mathcal{T}_{2i}^*). \tag{3.1}$$

Proof. In a matching of $\mathcal{T}$, if the edge containing v_1 is from $\mathcal{T}_1$, the same holds for v_2 because $\mathcal{T}_1$ has an even number of vertices. So this matching of $\mathcal{T}$ contains a matching of $\mathcal{T}_1$. Then again by parity, edges containing v_2 and v_3 cannot come from $\mathcal{T}_2$, so this matching contains a matching of $\mathcal{T}_2^*$. This pattern continues; we see that this matching equals the union of a matching of $\mathcal{T}_i$ for i odd and a matching of $\mathcal{T}_j^*$ for j odd. We conclude that a matching of $\mathcal{T}$ is either from a set of matchings of $\{\mathcal{T}_i, \mathcal{T}_j^*\}$ or from $\{\mathcal{T}_i^*, \mathcal{T}_j\}$ for all i even and j odd. □

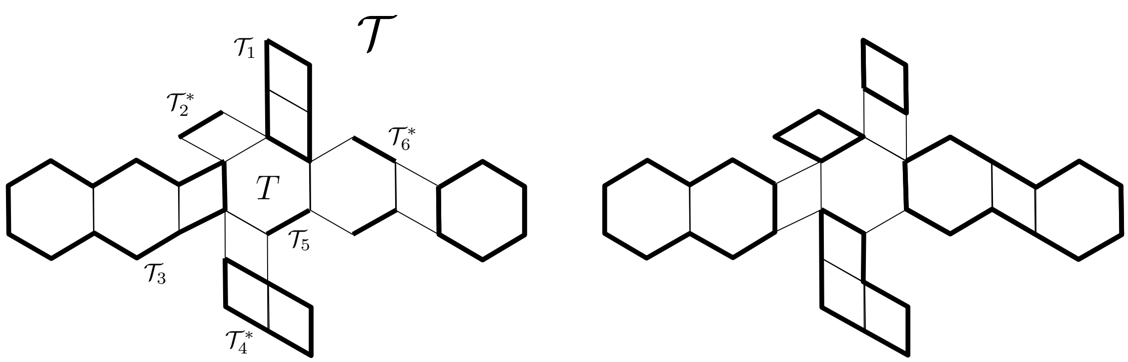


FIGURE 3.3. Two possible sets of matchings of $\mathcal{T} = T \cup (\cup_{i=1}^{6} \mathcal{T}_i)$

The simplest example of above lemma is when $\mathcal{T} = T$. This order one tree T has two matchings; one contains all e_i for i even and the other contains all e_j for j odd. Let us define $w(\mathcal{T}^{ev}) = \prod_{j=1}^{m} w(\mathcal{T}_{2j}) w(\mathcal{T}_{2j-1}^*)$ and $w(\mathcal{T}^{od}) = \prod_{i=1}^{m} w(\mathcal{T}_{2i-1}) w(\mathcal{T}_{2i}^*)$ so $w(\mathcal{T}) = w(\mathcal{T}^{ev}) + w(\mathcal{T}^{od})$.

Theorem 3.2. *Any natural mutation sequence associated with a tree $\mathcal{T}$ carries $\mathcal{T}$.*

Proof. We follow the notations from above discussion. We claim something stronger: If we have a tree $\mathcal{T}$, (after following a natural mutation sequence) $y_{v_{\mathcal{T}}} = m(\mathcal{T})w(\mathcal{T})$ holds and if it is extendable to F then $y_{v_{\mathcal{T}}}^* x_{v_F} = m(\mathcal{T})w(\mathcal{T}^*)$ holds. We use induction on the order of a tree. If $\mathcal{T}$ is an edge e, then $1 = m(e)w(e)$ and $x_{v_{\mathcal{T}}} x_{v_F} = m(e)$ so our claim holds where F can be one of two faces containing e. Now let $\mathcal{T}$ to be a tree of positive order, and assume that this claim is true for all trees that have order less than $\mathcal{T}$. For each $1 \leq i \leq 2m$, $\mathcal{T}_i$ is extendable to $\mathcal{T}$ and have order less than that of $\mathcal{T}$, so $y_i = m(\mathcal{T}_i)w(\mathcal{T}_i)$ and $y_i^* x_{v_T} = m(\mathcal{T}_i)w(\mathcal{T}_i^*)$. We mutate at all subtrees and localize at T. Here we consider the well-split case first:

(i) Well-split around T: This means a neighbor face U of $\mathcal{T}$ shares edges with only one of $\mathcal{T}_i$. In this case, we follow the definition of the covering monomial to obtain $m(\mathcal{T}) = (\prod_{i=1}^{2m} m(\mathcal{T}_i))/x_{v_T}^{m+1}$. Also, when we localize at T, there are no cancellations of arrows. Therefore, all arrows in y_i^* for each i are preserved. Then we clearly have

$$y_{v_T} x_{v_T} = \prod_{j=1}^{m} y_{2j} y_{2j-1}^* + \prod_{i=1}^{m} y_{2i-1} y_{2i}^*. \tag{3.2}$$

Define $y_{v_T}^{ev} = \prod_{j=1}^{m} y_{2j} y_{2j-1}^*$ and $y_{v_T}^{od} = \prod_{i=1}^{m} y_{2i-1} y_{2i}^*$. We multiply both sides of the formula for $w(\mathcal{T})$ by $m(\mathcal{T})$ and we are left with $y_{v_T} = m(\mathcal{T})w(\mathcal{T})$. Now without loss of generality, assume that $\mathcal{T}$ is extendable to F_1 , which would imply that $\mathcal{T}_1 = e_1$. Then $m(\mathcal{T}) = x_1(\prod_{i=2}^{2m} m(\mathcal{T}_i))/x_{v_T}^{m}$ and $w(\mathcal{T}^*) = \prod_{i=1}^{m} w(\mathcal{T}_{2i}^*) \prod_{j=1}^{m-1} w(\mathcal{T}_{2j+1})$. Also it is easy to see $y_{v_T}^* = \prod_{i=1}^{m} y_{2i}^* \prod_{j=1}^{m-1} y_{2j+1}$. Combining these, we get

$$y_{v_T}^* x_1 = m(\mathcal{T})w(\mathcal{T}^*).$$

(ii) General case: We can use above separation scheme to make the neighbor of $\mathcal{T}$ to be well-split. That is, each neighbor face U is split into $V_1, V_2, ...V_t$ such that V_i (can be empty) only shares one edge with $\mathcal{T}$. For each U, we are looking for a specific merging sequence such that we have $y_{v_T} = m(\mathcal{T})w(\mathcal{T})$ and $y_{v_T}^* x_1 = m(\mathcal{T})w(\mathcal{T}^*)$ throughout the whole process. In $\bar{G}$, we localize at vertices that correspond to faces in $\mathcal{T}$ and v_U and look for cycles containing U and its parts. There are four types of such cycles:

$$A : U \to V_{a_1} \to \mathcal{T}_{i_a} \to V_{a_2} \to U,$$

$$B : U \to V_{b_1} \to \mathcal{T}_{i_b} \to T \to V_{b_2} \to U,$$

$$C : U \to V_{c_1} \to \mathcal{T}_{i_c} \to T \to \mathcal{T}_{j_c} \to V_{c_2} \to U,$$

and

$$D : U \to V_{d_1} \to T \to V_{d_2} \to U,$$

(or all arrows in the opposite direction) where "$\to \mathcal{T}_i \to$" means this cycle is passing through some faces of $\mathcal{T}_i$. Cycles of types B, C, and D have the same (maximal) order since T is the maximal face. Cycles of type A have strictly less order than cycles of other types. Pick one of minimal cycles of type A and merge corresponding V_{a_1} and V_{a_2} back to U. As we merge parts back, we should kill the corresponding A cycle to be coherent with our covering monomial rule. We repeat this until we see no more A cycles. Some cycles of type B, C, and D will be killed in this process. When we merge parts from A cycles back to U, $\mathcal{T}$ is still well-split around T, so $y_{v_T} = m(\mathcal{T})w(\mathcal{T})$ and $y_{v_T}^* x_1 = m(\mathcal{T})w(\mathcal{T}^*)$ holds.

If we have a D cycle left, merge V_{d_1} and V_{d_2} back to U. Then we have a cancellation of arrows $U \to T$ and $U \leftarrow T$ so y_{v_T} has lost one factor of x_{v_U}. But since we observe a 2-cycle at T, when we are determining g_U of $m(\mathcal{T})$, we have lost one factor of x_{v_U}, so $y_{v_T} = m(\mathcal{T})w(\mathcal{T})$ is still true. Also both of $y_{v_T}^{ev}$ and $y_{v_T}^{od}$ has lost one factor of x_{v_U}. We are assuming that $\mathcal{T} \cup F_1$ is a tree, so it is not possible that $F_1 = U$. Hence $x_{v_U} \neq x_1 = y_1^*$ and if shows $y_{v_T}^*$ has lost one factor of x_{v_U}. That is, $y_{v_T}^* x_1 = m(\mathcal{T})w(\mathcal{T}^*)$ also holds. And we kill this 2-cycle. We repeat this until we see no more D cycles.

Next we look at a B or C cycle, if any. If we have a B cycle, we merge back V_{b_1} and V_{b_2} to U. After merging, this cycle looks like

$$U(\to V_{b_1}) \to B_1 \to B_2 \to ... \to B_k = F_{i_b} \to T(\to V_{b_2}) \to U$$

where $B_1, ...B_k$ are faces of $\mathcal{T}_{i_b}$. Then $m(\mathcal{T})$ has lost one x_{v_U} since we now have one more U-cycle. When we mutate at B_1 in a natural mutation sequence, the arrow $U \to B_1$ gets transported to an arrow $U \to B_2$. If $2 \neq k$, assume that at B_2 there are more or equal number of arrows of $B_2 \to U$ than the number of $U \to B_2$ so that this arrow $U \to B_2$ originated from $V_{b_1} \to B_1$ is killed. It is a contradiction since we then have an "unexpected" cycle $U(\to V_{b_1}) \to B_1 \to B_2 \to U$ which should have been killed when considering this B cycle. That is, the arrow $U \to B_2$ "survives" at the localization at B_2 and gets transported to B_3 when we mutate at B_2. By the same argument, this pattern continues and the arrow $U \to B_1$ finally gets transported to an arrow $U \to T$ and kills the arrow $T(\to V_{b_2}) \to U$. Therefore, y_{v_T} has lost one factor of x_{v_U}. We kill this cycle, and look for a B or C cycle.

Assume there is a C cycle left. Then we merge back V_{c_1} and V_{c_2}; It looks like

$$U \to B_1 \to ... \to B_k \to T \to A_l \to ... \to A_1 \to U$$

where $B_1, ...B_k$ belong to $\mathcal{T}_{i_c}$ and $A_1, ...A_l$ belong to $\mathcal{T}_{j_c}$ for some i_c and j_c. Again, $m(\mathcal{T})$ lose one factor of x_{v_U} and as we mutate at $\mathcal{T}_{i_c}$ the arrow $U \to B_1$ is transported to an arrow $U \to T$ and $A_1 \to U$ becomes $T \to U$. Hence y_{v_T} also lose one factor of x_{v_U}. And kill this cycle. Repeat this procedure until we do not see any B or C cycles.

We have seen that $y_{v_T} = m(\mathcal{T})w(\mathcal{T})$ holds when we merge back parts that belonged to such cycles. Likewise we see that $y^*_{v_T}x_1 = m(\mathcal{T})w(\mathcal{T}^*)$ throughout this whole procedure. Now we merge back all parts left which does not involved in any U-cycles. First, this does not affect $m(\mathcal{T})$ since there are no cycles at this stage. This also does not affect y_{v_T}; if it is affected, it means there are two "new" opposite arrows (after merging) between T and U when we mutate at all subtrees and localize at T. This requires the existence of a cycle of type B, C, or D. Therefore, we have successfully merged back all parts of U and two equations still hold. We repeat this procedure for all neighbor face of $\mathcal{T}$ and we are done. □

Remark 3.3. We have defined $y^{ev}_{v_T}$ and $y^{od}_{v_T}$ in the well-split case so that $y_{v_T}x_{v_T} = y^{ev}_{v_T} + y^{od}_{v_T}$. In non well-split cases, $y^{ev}_{v_T} = \prod_{j=1}^{m} y_{2j}y^*_{2j-1}$ is no longer true (we will miss some factors due to cancellations), but we can still define $y^{ev}_{v_T}$ and $y^{od}_{v_T}$ such that $y_{v_T}x_{v_T} = y^{ev}_{v_T} + y^{od}_{v_T}$ still holds.

3.2. **Superposition.** Until this point, we have never mutated at previously mutated vertices. Here we describe an instant where we still get combinatorial interpretations in G when we repeatedly mutate at some sequences of vertices in $\bar{G}$.

We introduce some notations. For each odd n, initial cluster variable of F_n will be called x_n as usual, and further variables we get as we successively mutate at n will be named $y_n = y_{[n,n]}$, $y_{[n-2,n+2]}$, $y_{[n-4,n+4]}$,... and so on. For even n, variables will be called x_n, $y_n = y_{[n-1,n+1]}$, $y_{[n-3,n+3]}$, $y_{[n-5,n+5]}$,... and so on.

A snake was defined as a tree without bridge faces. We will write faces of a snake S as $F_1, ...F_n$ where $F_i < F_j$ if and only if $i < j$ so that F_n is the maximal face. We let the vertex in $\bar{G}$ corresponding to F_i be i. When we have a snake $S = \cup_{i=1}^{n} F_i$, we say S is **straight** at $1 < j < n$ if in $\bar{G}$ with respect to j (vertex corresponding to F_j) arrows $j \leftrightarrow j+1$ and $j-1 \leftrightarrow j$ have the opposite direction. Otherwise we say it is **bent** at j. In this situation, let e_i be the unique edge shared by F_i and F_{i+1} for $i = 1...n-1$. Let e_0 (resp. e_n) be the second edge different from e_1 (e_{n-1}) in F_1 (F_n) counted counterclockwise (clockwise) from e_1 (e_n). In F_i, we label edges of F_i counterclockwise from e_i as $e_{i,-j_i-1}, e_{i,-j_i}, ...e_{i,-2}, e_{i,-1}, e_{i,0}, e_{i,1}, ...e_{i,k_i}$

where $e_{i-1} = e_{i,0}$, $e_i = e_{i,k_i+1} = e_{i,-j_i-1}$. Here, k_i and j_i will have the same parity. An edge of S will be called **even** if it is $e_{i,q}$ for some nonzero even q and $1 \leq i \leq n$. An edge which is not one of e_p is called **odd** otherwise. As two exceptions, $e_0 = e_{1,0}$ and $e_n = e_{n,-2}$ are considered even. Parity of edge $e_{i,q}$ will mean parity of q. Parity of e_p for $1 \leq p < n$ is not defined. An infinite snake S is simply a union of distinct faces of G, $S = \cup_{i \in \mathbb{Z}} F_i$ (or $\cup_{i \in \mathbb{N}} F_i$) such that any subunion of the form $\cup_{j \leq i \leq k} F_i$ for any pair of integers $j \leq k$ is a snake of order $k - j + 1$ (or for a pair of integers $1 \leq j \leq k$ for $S = \cup_{i \in \mathbb{N}} F_i$).

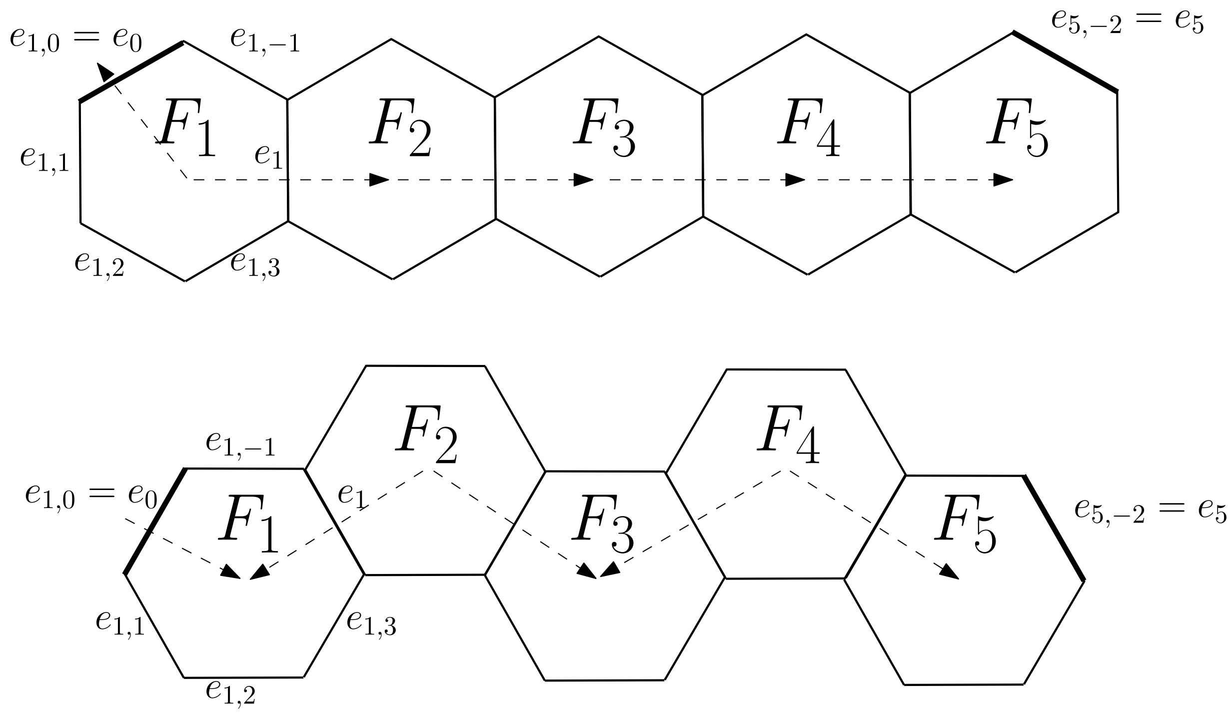


FIGURE 3.4. Two hexagonal snakes; At vertices 2, 3, and 4, above one is bent and below one is straight

Definition 3.4. A **spine** will mean a (possibly infinite) snake that is straight at all faces. Assume $S = \cup_{a \leq i \leq b} F_i$ is a spine. ($a = -\infty$, $b = \infty$, or both is allowed) We let $\mathcal{T}_{i,k}$ be the tree attached to the edge $e_{i,k}$ whenever $e_{i,k}$ is well-defined. Define $\mathcal{T}_i = F_i \cup (\cup \mathcal{T}_{i,k})$ for all i and k defined, and $\mathcal{T}_{[c,d]} = \cup_{c \leq k \leq d} \mathcal{T}_k$ where we require $\mathcal{T}_k$ to be well-defined for all $c \leq k \leq d$.

In a spine, mutation at F_i and F_j commutes if $|i-j| \geq 2$. So consider a procedure "o" where we simultaneously mutate at all odd faces of S and "e" where we mutate at all even. When we do "o" (or "e") arrows $F_i \to F_{i+1}$ still alternate for all i; that is, we again get a spine no matter how many times we repeat this simultaneous mutation. Motivated from this, we do not define the ordering on S in the usual way but we impose $F_{odd} < F_{even}$ when we consider the sequence *oeoeoe*....We assume that each face F_i have trees (all commuting) attached and they are already mutated (which will imply that $T_{i,l} \lessdot F_i$ for all i and l where $T_{i,l}$ is the maximal face of $\mathcal{T}_{i,l}$). Mutation sequences *eoeoeo*... or *oeoeoe*... respects 2-periodicity of our rough picture, so it is enough to localize at one even face of S after *eo*...*eoeo* or *o*...*eoeo* and one odd face after *oe*...*oeoe* or *e*...*oeoe*. To be more precise, if at a certain stage F_i carries $\mathcal{T}_{[a,b]}$ then it automatically means F_{i+2k} carries $\mathcal{T}_{[a+2k,b+2k]}$ for any integer k, as long as F_{i+2k} and $\mathcal{T}_{[a+2k,b+2k]}$ are well-defined.

One thing we should note is that $\mathcal{T}_{[a,b]}$ is not a tree since it has many maximal faces (namely all F_{even} in $\mathcal{T}_{[a,b]}$) but we can define its monomial in the exact same way, since it is well-ordered among faces that contain a U-cycle; that is, any U-cycles cannot pass through several maximal faces (Remark 2.5).

Theorem 3.5. *Assume we have a spine S in G whose faces are indexed by integers, $\cup_{a\leq i\leq b}F_i$. After mutating at all commuting trees $\mathcal{T}_{i,l}$, we consider the mutation sequence oeoeoeoeoeoe.... At kth stage, F_{k+1+2t} carries $\mathcal{T}_{[1+2t,2k+1+2t]}$ for any integer t whenever $\mathcal{T}_{[1+2t,2k+1+2t]}$ is well-defined. That is, $y_{[1+2t,2k+1+2t]} = w(\mathcal{T}_{[1+2t,2k+1+2t]})m(\mathcal{T}_{[1+2t,2k+1+2t]})$.*

Proof. It is enough to show $y_{[1,2k+1]} = w(\mathcal{T}_{[1,2k+1]})m(\mathcal{T}_{[1,2k+1]})$, assuming that $\mathcal{T}_{[1,2k+1]}$ is well-defined. This is achieved by comparing the cluster variable formula from mutated quiver with the weight formula of the graph. To get the cluster variable formula, we need an explicit description of all the arrows connected to faces of S at each stage of our mutation sequence; which can be done by induction. On the other hand, we will use Kuo's condensation to get a formula for the weights. Once we have all the formulas, it is a direct computation.

Without loss of generality, assume that we have an arrow $F_0 \to F_1$ in $\bar{G}$, which will determine directions of all other arrows in $\bar{G}$. Localize at F_1, after mutating at all its attached trees, $\mathcal{T}_{1,l}$. Since $\mathcal{T}_1 = F_1 \cup (\cup \mathcal{T}_{1,l})$ is a tree with maximal face F_1, $y_1x_1 = y_{[1,1]}x_1 = y_1^{od} + y_1^{ev}$. We did not mutate at F_0 and F_2 yet, so variable y_1^{od} contains arrows from F_0 and F_2; we now want to keep track of these arrows separately. We define $x_0x_2y_1^{tod} = y_1^{od}$, $y_1^{tev} = y_1^{ev}$ so that $y_{[1,1]}x_1 = x_0x_2y_1^{tod} + y_1^{tev}$.

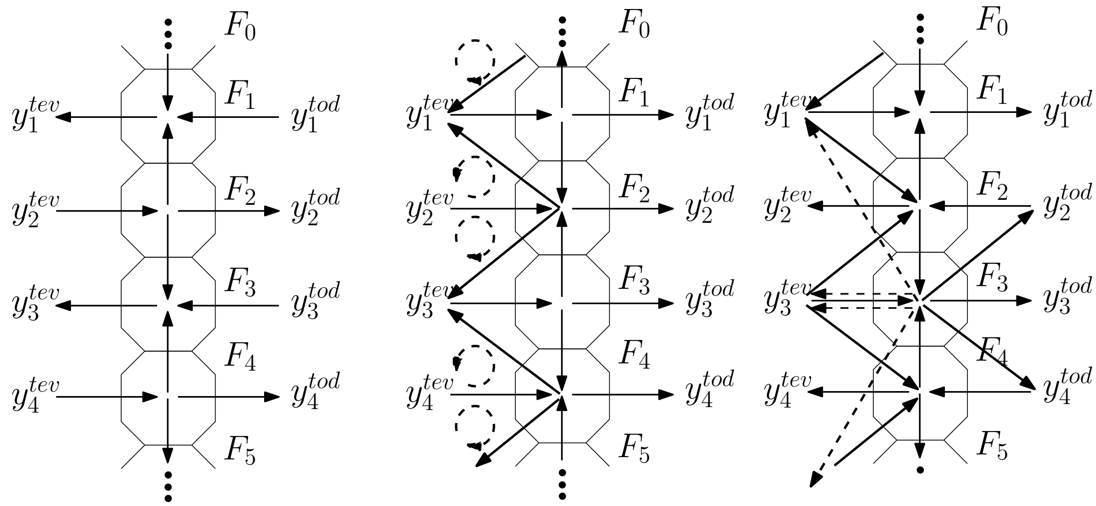


FIGURE 3.5. First two mutations; Dashed lines in the rightmost figure now carries $y_1^{tev}/P_{12}, y_3^{tev}/P_{23}, y_3^{tev}/P_{34}$, and y_5^{tev}/P_{45} respectively from top to bottom.

Define y_n^{ted} and y_n^{tod} similarly, such that $y_nx_n = y_{[n,n]}x_n = x_{n-1}x_{n+1}y_n^{tod} + y_n^{tev}$ for all n. Applying (3.2) to the tree $\mathcal{T}_1$ with maximal face F_1 gives us $y_1^{ev} = w(\mathcal{T}_1^{ev})m(\mathcal{T}_1)x_1$ and $y_1^{od} = w(\mathcal{T}_1^{od})m(\mathcal{T}_1)x_1$. Here, notice that $w(\mathcal{T}_1^{ev})$ contains $w(e_0)$ and $w(e_1)$; define $w(\mathcal{T}_1^{tev})w(e_0)w(e_1) = w(\mathcal{T}_1^{ev})$ and $w(\mathcal{T}_1^{tod}) = w(\mathcal{T}_1^{od})$. From these, we now have formulas

$$y_n^{tev} = w(\mathcal{T}_n^{tev})m(\mathcal{T}_n)/x_{n-1}x_nx_{n+1} \tag{3.3}$$

and

$$y_n^{tod} = w(\mathcal{T}_n^{tod})m(\mathcal{T}_n)x_n/x_{n-1}x_{n+1} \tag{3.4}$$

for any n.

After we get y_i for all odd i, we do "o" and look at even snake faces; by periodicity, we only need to look at one of them, say F_2. This is another tree phenomenon at work with maximal face F_2, where the tree attached to e_1 is $\mathcal{T}_1$ and the tree attached to e_2 is $\mathcal{T}_3$ and we have original trees of F_2, namely $\mathcal{T}_{2,l}$. We will actually verify F_2 carries $\mathcal{T}_{[1,3]}$ with our new notations. For a moment assume that it is a *tree well-split* case; that is, a neighborhood of $\mathcal{T}_{[1,3]}$ do not share edges with both of $\mathcal{T}_1$ and $\mathcal{T}_2$ or both of $\mathcal{T}_3$ and $\mathcal{T}_2$. Sharing edges with both $\mathcal{T}_1$ and $\mathcal{T}_3$ does not matter; first, there cannot be any cycle (the monomial is not affected) going through F_1, F_2, and F_3 as our snake is straight, and also at this stage, arrows originated from $\mathcal{T}_1$ and $\mathcal{T}_3$ cannot kill each other as they have the same direction (cluster variable formula is also unaffected). In this setting, we easily verify by comparing exponents that

$$m(\mathcal{T}_{[1,3]}) = m(\mathcal{T}_1)m(\mathcal{T}_2)m(\mathcal{T}_3)/x_1x_2^2x_3,$$

and from Equation (3.1), it is direct to see

$$w(\mathcal{T}_{[1,3]}) = w(\mathcal{T}_1)w(\mathcal{T}_2^{tev})w(\mathcal{T}_3) + w(\mathcal{T}_1^{tev})w(\mathcal{T}_2^{tod})w(\mathcal{T}_3^{tev}).$$

Multiply both sides by $m(\mathcal{T}_{[1,3]})$ and compare the resulting equation with $y_2x_2 = y_{[1,3]}x_2 = y_1y_2^{tev}y_3+y_1^{tev}y_2^{tod}y_3^{tev}$, which we get from the quiver. Then we see $y_2 = m(\mathcal{T}_{[1,3]})w(\mathcal{T}_{[1,3]})$ holds. In the general case, for each neighbor face U of $\mathcal{T}_{[1,3]}$, we might observe cycles $U \leftrightarrow \mathcal{T}_{[1,3]}$ going through F_1 and F_2 or F_2 and F_3. That is, they all have the same order and hence can be taken care of (as before, we are assuming that we first separate neighbor faces to reduce to a tree well-split case, then merge parts from U-cycles of this particular form, and finally merge back parts left - which does not change anything indeed) one by one. In the former case, a component of y_1^{tev} is reacting with a component of y_2^{tev} and in the latter case, a component of y_3^{tev} is reacting with a component of y_2^{tev}. If we look at localization of y_2, it is clear that at this stage these are only two possibilities of cancellations, since cancellations of (sets of) arrows y_2^{tod} and y_2^{tev} are assumed to be taken care of. Such cancellations are shown as circular arrows in Figure (3.5). So when we pick one of these cycles and merge corresponding parts back to U, $m(\mathcal{T}_{[1,3]})$ lose one x_{v_U}; on the other side, y_2^{tev} always lose one x_{v_U} and either y_1^{tev} or y_3^{tev} (not both) lose one x_{v_U}. That is, y_2 lose one x_{v_U} in any cases; $y_2 = m(\mathcal{T}_{[1,3]})w(\mathcal{T}_{[1,3]})$ remains true. After we are done with merging, all U-cycles going through F_2 are determined for all neighbor faces U and now y_1^{tev} and y_3^{tev} has lost some monomials respectively. Call them P_{12} and P_{23} respectively so that the former y_1^{tev}, y_2^{tev}, and y_3^{tev} now carries y_1^{tev}/P_{12}, $y_2^{tev}/P_{12}P_{23}$, and y_3^{tev}/P_{23}, respectively. Same thing is happening at every F_{2k} due to 2-periodicity. In general, U-cycles are "short" (only formed within $U \leftrightarrow \mathcal{T}_i \leftrightarrow \mathcal{T}_{i+1}$ for some i) so we can merge every parts back after taking care of such short cycles. Our monomial formula now reads

$$m(\mathcal{T}_{[2k-1,2k+1]}) = m(\mathcal{T}_{2k-1})m(\mathcal{T}_{2k})m(\mathcal{T}_{2k+1})/x_{2k-1}x_{2k}^2x_{2k+1}P_{2k-1\,2k}P_{2k\,2k+1}.$$

After "e" we localize at F_3. We have arrows going out to faces that have variables y_2^{tod}, y_3^{tod}, y_4^{tod}, y_1^{tev}/P_{12}, y_5^{tev}/P_{45}, y_3^{tev}/P_{23}, and y_3^{tev}/P_{34} and arrows coming in from faces of y_3^{tev}, y_2, and y_4. The arrow y_3^{tev} is totally canceled by two arrows y_3^{tev}/P_{23} and y_3^{tev}/P_{34} and we are left with the arrow $y_3^{tev}/P_{23}P_{34}$ going out. So we have only two arrows coming in faces F_2 and F_4, so no new cancellations can occur at this step (this must be happening since we have merged all our faces back). We have $y_{[1,5]}y_3 = y_2y_4 + (y_1^{tev}/P_{12})(y_3^{tev}/P_{23}P_{34})(y_5^{tev}/P_{45})y_2^{tod}y_3^{tod}y_4^{tod}$.

We claim and prove by induction that after we mutate for k times and look at F_{k+1}, after cancellations, we have arrows going out to y_i^{tod} where $2 \le i \le 2k$, y_1^{tev}/P_{12}, $y_{2k+1}^{tev}/P_{2k\,2k+1}$, and $y_j^{tev}/P_{j-1\,j}P_{j\,j+1}$ for $3 \le j \le 2k-1$ and arrows coming in from F_k and F_{k+2}. That is,

$$y_{[1,2k+1]}y_{[3,2k-1]} = y_{[1,2k-1]}y_{[3,2k+1]} + \prod_{i=2}^{2k} y_i^{tod} \prod_{j=3}^{2k-1} (y_j^{tev}\frac{1}{P_{j-1\,j}P_{j\,j+1}})\frac{y_1^{tev}y_{2k+1}^{tev}}{P_{12}P_{2k\,2k+1}}. \; (k \ge 2) \tag{3.5}$$

Again, no further cancellations of arrows can occur. Next we consider the monomials. It is another simple computation to check that $m(\mathcal{T}_{[1,2k+1]})m(\mathcal{T}_{[3,2k-1]}) = m(\mathcal{T}_{[1,2k-1]})m(\mathcal{T}_{[3,2k+1]})$ holds. Also it is straightforward to see

$$m(\mathcal{T}_{[1,n]}) = x_1 x_n \prod_{i=1}^{n}(\frac{m(\mathcal{T}_i)}{x_i^2})\prod_{j=1}^{n-1}(\frac{1}{P_{j\,j+1}}). \tag{3.6}$$

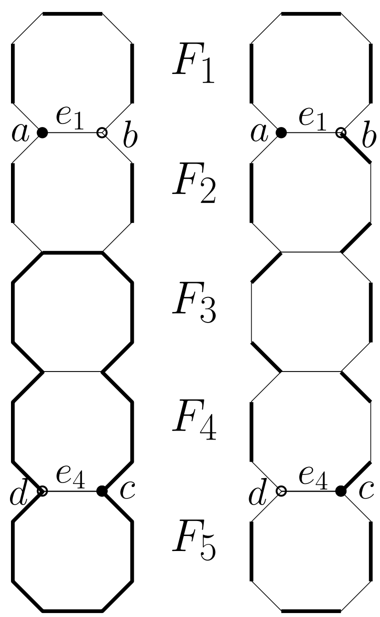


FIGURE 3.6. A simple example of $\mathcal{T}_{[1,5]}\backslash\{a,b\}$ and $\mathcal{T}_{[1,5]}\backslash\{a,d\}$. (left, right)

The final piece of the puzzle is a corresponding formula for weights. Let two vertices of e_1 be a and b and two vertices of e_{2k} be c and d such that a, b, c, and d are located in the cyclic order in a face of S. We apply the condensation formula in [7] to $H = \mathcal{T}_{[1,2k+1]}$ with these four vertices. S is straight so a, c are in the same vertex set and b, d are in the other vertex set. Hence

$$w(H)w(H\backslash\{a,b,c,d\}) = w(H\backslash\{a,b\})w(H\backslash\{c,d\}) + w(H\backslash\{a,d\})w(H\backslash\{b,c\}).$$

We see what happens if we get rid of a and b (and all incident edges to them). In $\mathcal{T}_1$, we will have full even positions (including e_0) and star odd positions, which together have weight $w(\mathcal{T}_1^{tev})/x_0x_2$. On the other hand, in $\mathcal{T}_2$ we have full odd positions and star even positions, leaving $\mathcal{T}_{[3,2k+1]}$ free. Therefore, $w(\mathcal{T}_{[1,2k+1]}\backslash\{a,b\}) = w(\mathcal{T}_1^{tev})w(\mathcal{T}_2^{tev})w(\mathcal{T}_{[3,2k+1]})/x_0x_2$. Two terms $w(\mathcal{T}_{[1,2k+1]}\backslash\{c,d\})$ and $w(\mathcal{T}_{[1,2k+1]}\backslash\{a,b,c,d\})$ are analogously determined. When we get rid of a, d, what happens is that in $\mathcal{T}_1$we still have $w(\mathcal{T}_1^{tev})/x_0x_2$. But in $\mathcal{T}_2$, we have a mixed pattern; for positive l, when we consider $\mathcal{T}_{2,l}$ then we have full odd and star even, but for negative l we have full even and star odd. This pattern is reversed in $\mathcal{T}_3$; for positive l we have full even and star odd, and for negative l we have full odd and star even. This continues to alternate until we reach the end where we are left with the weight $w(\mathcal{T}_{2k+1}^{tev})/x_{2k+1}x_{2k+2}$. By symmetry, we have the opposite pattern when we get rid of b, c and it shows $w(\mathcal{T}_{[1,2k+1]}\backslash\{a,d\})w(\mathcal{T}_{[1,2k+1]}\backslash\{b,c\})$ equals $(w(\mathcal{T}_1^{tev})/x_0x_2)^2(w(\mathcal{T}_{2k+1}^{tev})/x_{2k+1}x_{2k+2})^2$ times $\prod_{2\leq i\leq 2k}(w(\mathcal{T}_i^{tev})w(\mathcal{T}_i^{tod}))$. When we plug everything in the condensation formula we are left with

$$w(\mathcal{T}_{[1,2k+1]})w(\mathcal{T}_{[3,2k-1]}) = w(\mathcal{T}_{[1,2k-1]})w(\mathcal{T}_{[3,2k+1]}) + \prod_{i=2}^{2k} w(\mathcal{T}_i^{tod}) \prod_{j=3}^{2k-1} w(\mathcal{T}_j^{tev}) \frac{w(\mathcal{T}_1^{tev})w(\mathcal{T}_{2k+1}^{tev})}{x_0x_2x_{2k+1}x_{2k+2}},$$

where $k \geq 2$. Multiply both sides of this equation with $m(\mathcal{T}_{[1,2k+1]})m(\mathcal{T}_{[3,2k-1]})$, which is equal to $m(\mathcal{T}_{[1,2k-1]})m(\mathcal{T}_{[3,2k+1]})$. We first use induction assumptions ($w(\mathcal{T}_{[3,2k-1]})m(\mathcal{T}_{[3,2k-1]}) = y_{[3,2k-1]}$, $w(\mathcal{T}_{[1,2k-1]})m(\mathcal{T}_{[1,2k-1]}) = y_{[1,2k-1]}$, and $w(\mathcal{T}_{[3,2k+1]})m(\mathcal{T}_{[3,2k+1]}) = y_{[3,2k+1]}$) to simplify. To take care of the last term, we use (3.6) to write $m(\mathcal{T}_{[1,2k+1]})$ and $m(\mathcal{T}_{[3,2k-1]})$ as products of $m(\mathcal{T}_j)$ and other monomials. Then we can again simplify by equations (3.3) and (3.4). Finally, we compare the resulting equation with (3.5); we are left with

$$y_{[1,2k+1]} = w(\mathcal{T}_{[1,2k+1]})m(\mathcal{T}_{[1,2k+1]}).$$

□

4. Applications

4.1. **Infinite Grids.**

4.1.1. *The Square Grid.* Consider the infinite unit square grid G in which squares are centered at (i, j) where i and j ranges over all integers. Naturally, each square which we will denote as $s(i, j)$ will have variable $x_{(i,j)}$ indexed by its center. Its dual quiver will be another infinite square grid where arrows are cyclic and vertices are indexed by two integers.

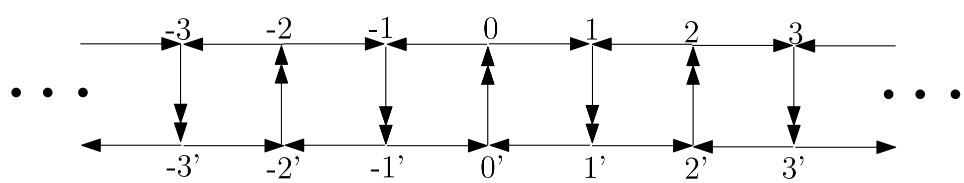


Figure 4.1. $2 \times \infty$ Rectangular Quiver

To approach the original conjecture, we define an equivalence relation on the set $\mathbb{Z} \times \mathbb{Z}$: $(a, b) \sim (c, d)$ if and only if $a = c$ and b and d have the same parity. We set n and n' to be the equivalence classes containing $(n, 1)$ and $(n, 0)$ respectively. With this relation, our infinite grid quiver is folded into 2 by ∞ quiver with vertices n and n'. In this setting, our infinite straight snake S will be the union of all squares with its first coordinate zero. These squares are colored by 0 and $0'$, alternatively.

On the infinite grid G, consider the following three families of graphs $\{P(j,k)_n\}_{n=1}^{\infty}$,$\{Q(j,k)_n\}_{n=1}^{\infty}$, and $\{R(j,k)_n\}_{n=1}^{\infty}$ for each pair of non negative integers (j, k), where

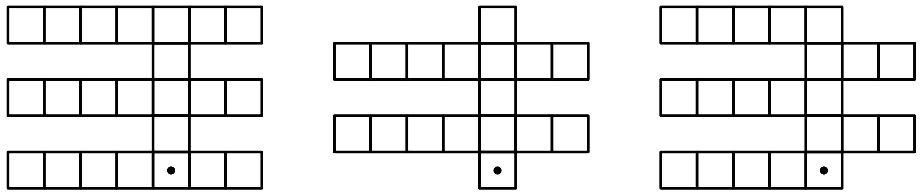


FIGURE 4.2. Graphs of $P(4,2)_3$, $R(4,2)_3$, and $A(4,2)_3$; the squre $s(0,0)$ is marked with a dot

$$P(j,k)_n = \Big(\bigcup_{\substack{-j\le i\le k\\ 0\le l\le n-1}} s(i,2l)\Big) \cup \Big(\bigcup_{m=1}^{2n-1} s(0,m)\Big),$$

$$R(j,k)_n = \Big(\bigcup_{\substack{-j\le i\le k\\ 0\le l\le n-2}} s(i,2l+1)\Big) \cup \Big(\bigcup_{m=1}^{2n-1} s(0,m)\Big),$$

and

$$A(j,k)_n = \Big(\bigcup_{\substack{0<i\le k\\ 0\le l\le n-2}} s(i,2l+1)\Big) \cup \Big(\bigcup_{\substack{-j\le i<0\\ 0\le l\le n-1}} s(i,2l)\Big) \cup \Big(\bigcup_{m=1}^{n-1} s(0,2m-1)\Big).$$

By using Theorem (3.5), we obtain the following result.

Corollary 4.1. *We first mutate in the order $-j,-j-1,...,-2-1,k,k-1,k-2,...2,1$ (first mutate at the branches) and then mutate in the sequence $0,0',0,0',0,....$ then $y_{[0,2n-2]} = w(P(j,k)_n)m(P(j,k)_n)$. When we mutate at the branches and mutate at $0',0,0',0,0',...,$ we have $y_{[0,2n-2]} = w(R(j,k)_n)m(R(j,k)_n)$. Finally, when we mutate in the order $-j,-j-1,...,-2-1,k',(k-1)',...2',1'$ and mutate at $0,0',0,0',0,...,$ $y_{[0,2n-2]} = w(A(j,k)_n)m(A(j,k)_n)$.*

Proof. We observe that the action of folding the infinite quiver commutes with simultaneous mutation at all vertices of the same equivalence class. Hence mutating at $-j$ in our folded quiver corresponds to simultaneously mutating at all vertices $(-j,2r)$ for all integer r. Therefore we are really mutating at all the branches when we do $-j,-j-1,...,-2-1,k,k-1,k-2,...2,1$. Our spine (infinite straight snake) is $S=\cup_{i\in\mathbb{Z}} s(0,i)$ and superposition applies. Similarly we treat R and A. □

Remark 4.2. The graphs $P(1,1)_n$ correspond to those appearing in Figure 1.1 and the conjecture that inspired this project. When $j=k=1$, vertices -1,1 and $-1',1'$ become indistinguishable, so we can strengthen the equivalence relation by defining $\mathbb{Z}\times\mathbb{Z}$: $(a,b)\sim(c,d)$ if and only if a,c have the same parity and b,d have the same parity. Then this corresponds to further folding our $2\times\infty$ quiver to 2×2 quiver and we get the original conjecture back with the correct weighting if we name four equivalence classes as $1,2,3$, and 4 appropriately. Also $R(1,1)$ "lives inside" this 2×2 quiver but $A(1,1)$ does not. The smallest quiver containing $A(1,1)$ is 2×4.

From the formula

$$y_{[1,2k+1]}y_{[3,2k-1]} = y_{[1,2k-1]}y_{[3,2k+1]} + \prod_{i=2}^{2k} y_i^{tod} \prod_{j=3}^{2k-1} \Big(y_j^{tev}\frac{1}{P_{j-1\,j}P_{j\,j+1}}\Big)\frac{y_1^{tev}y_{2k+1}^{tev}}{P_{12}P_{2k\,2k+1}}, (n\ge 2)$$

we get recurrence relations on the number of matchings. We set all initial variables to be ones, then $P_{n\,n+1} = 1$ for any n, and if T used to carry a subgraph H of G, y_T equals the number of matchings of H. As an example, we can apply this to $P(j,k)$. For i even, there are no trees attached to $F_i = s(0,i)$ so $y_i^{ted} = y_i^{tod} = 1$. For odd i, we count $y_i^{tod} = f_{j+2}f_{k+2}$ and $y_i^{ted} = f_{j+1}f_{k+1}$ where f_v is the vth Fibonacci number with $f_1 = f_2 = 1$. Finally $y_{[1,2n+1]} = \#\text{matching}(P(j,k)_{n+1})$, so we are left with the recurrence

$$p(j,k)_{n+1}p(j,k)_{n-1} = p(j,k)_n^2 + (f_{j+2}f_{k+2})^{n-1}(f_{j+1}f_{k+1})^{n+1}$$

for $n \geq 2$, where $p(j,k)_i = \#\text{matching}(P(j,k)_i)$. Similarly,

$$r(j,k)_{n+1}r(j,k)_{n-1} = r(j,k)_n^2 + (f_{j+2}f_{k+2})^{n+1}(f_{j+1}f_{k+1})^{n-1}$$

for $n \geq 2$, where $r(j,k)_i = \#\text{matching}(R(j,k)_i)$. Likewise, whenever we have a superposition situation, we can compute y_i^{tod} and y_i^{tev} explicitly for all i to get similar recurrences as above. All of families P, R, and A have vertical period 2. We can unfold $2\times\infty$ quiver to get skeleton types of graphs of period 4, 6, ... living inside quivers $4\times\infty$, $6\times\infty$, and so on.

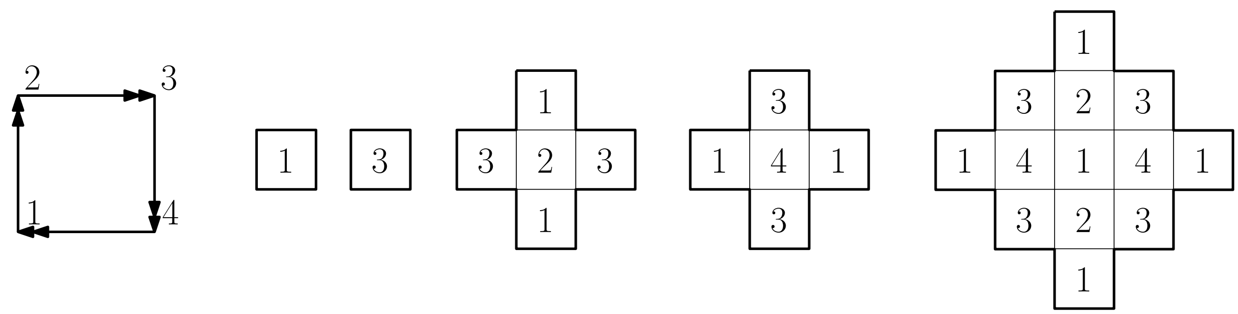


FIGURE 4.3. Fully folded rectangular quiver and first five Aztec diamonds

Here we rephrase the Aztec diamond theorem. Consider the fully folded rectangular quiver and the infinite square grid with the corresponding equivalence relation, so that each square is colored either 1, 2, 3, or 4. We consider the partial order $3 < 2 < 4 < 1$ on the Aztec diamonds living inside this square grids; then Remark 2.5, we have $g_U = 1$ for every neighbor face U of each Aztec diamond. We consider the mutation sequence 1, 3, 2, 4, 1, 3, 2, 4, ... on our quiver. When we mutate at 1 (resp. 3) for the kth time, the cluster variable we get carries the order $2k-1$ Aztec diamond centered at a face colored 1 (resp. 3). Likewise, when we mutate at 2 (resp. 4) for the k th time, the cluster variable we get carries the order $2k$ Aztec diamond centered at a face colored 2 (resp. 4). A proof can be found in [12].

4.1.2. *Tilings of the triangular grid.* If we have the triangular grid, we can first consider its rhombus tilings. Any rhombus tiling is bipartite, so we can consider its dual quiver and apply previous results. This contains the example of infinite square grid as a special case. We can define a tile in this triangular grid more generally to be a union $t = \cup_{i=1}^{2m} f_i$ of an even number of triangles such that for any i, exist $j \neq i$, f_i and f_j share an edge. Then, any tiling with such tiles is again bipartite so we consider its dual quiver. An important example of this tiling is the infinite hexagonal grid. We see that its dual graph is the triangular grid with arrows being cyclic in every face.

In the hexagonal grid, we can consider similar types of graphs as in the square grid case, with the infinite hexagonal spine. For an example, the infinite snake ...$BGBGBG$... in Figure 4.5 is straight, so it is a spine. We can attach commuting hexagonal trees to edges of this spine. On

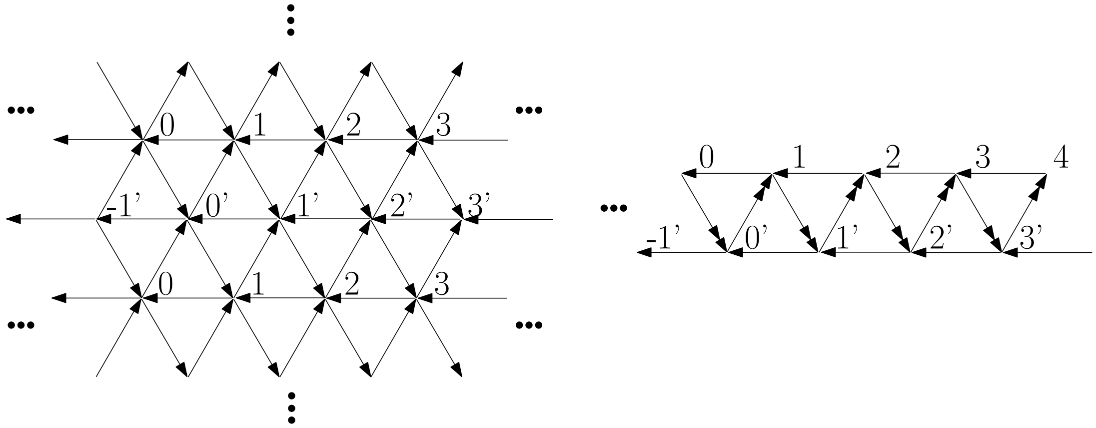


FIGURE 4.4. Vertical folding of the infinite triangular quiver into the $2 \times \infty$ triangular quiver

the quiver side, we can also fold its dual infinite triangular quiver into a $2\times\infty$ triangular quiver and fold this further in the other direction, too. (This equivalence relation is shown in Figure 4.4.) In this $2 \times \infty$ triangular quiver, (or equivalently in its further-folded version) we will get the weights of period 2 hexagonal skeleton-type graphs by following appropriate mutation sequences. An example is shown in Figure 4.5. Again, we can fold the infinite triangular quiver to be a $2m \times \infty$ triangular quiver and obtain weights of hexagonal skeleton-type graphs of period $2m$.

4.2. **Embedding and Dualizing Quivers.** Assume we have a planar quiver Q and want to find combinatorial interpretations for cluster variables from mutation sequences in V (set of vertices of Q). We will set $x_w = 1$ for all vertices $w \notin V$, so we only need to worry about arrows between vertices in V. Then we can dualize Q around V to obtain a planar, 2-connected, and bipartite graph G such that the dual quiver of G is a subquiver of Q whose vertex set contains V. To obtain such G, we clearly need for all $v \in V$ to be **cyclic**; the direction of arrows attached to v should alternate. This is also sufficient. Whenever there exists a vertex in V of degree 2, we attach two (or any even number) new alternating arrows to v. After that, draw a $2n$-gon around each $v \in V$ where $\deg v = 2n$ in a way that if $v, w \in V$ are connected by an arrow, their corresponding polygons will share an edge. There can be issues on where and how many should we add such arrows, given a vertex of degree 2. It does not matter; all planar bipartite graphs we obtain in this way will be matchingwise equivalent. That is, their weights $(w(G)m(G))$ are the same. Assume we have obtained two dual graphs, G_1 and G_2, from a quiver Q. Consider a tree in G_1, $\cup_{i=1}^{n} F_i$; this corresponds to a set of vertices in Q, and then consider the union of faces in Q corresponds to those vertices. We can see that such union of faces is a tree, $\cup_{i=1}^{n} F_i^{'}$, and has the same weight as $\cup_{i=1}^{n} F_i$. To prove this, it is enough to show that the action of adding two opposite arrows to a vertex does not affect the weight of its dual.

Assume we have a cyclic vertex v_E with degree 4 (Figure 4.6). We draw a quadrilateral E around v_E such that each arrow crosses exactly one side of this quadrilateral. If we add a 2-cycle between v_A and v_E, v_E is still cyclic, now with degree 6 - so we get a hexagon E' around v_E. Notice that any matching of E' comes from a matching of E with one of two new edges. These two new edges have the same weight of $1/x_{v_E} x_{v_A}$; hence $w(E') = w(E)/x_{v_E} x_{v_A}$. On the other hand, in the monomial formula, $m(E')$ has one more factor of x_{v_E} than $m(E)$

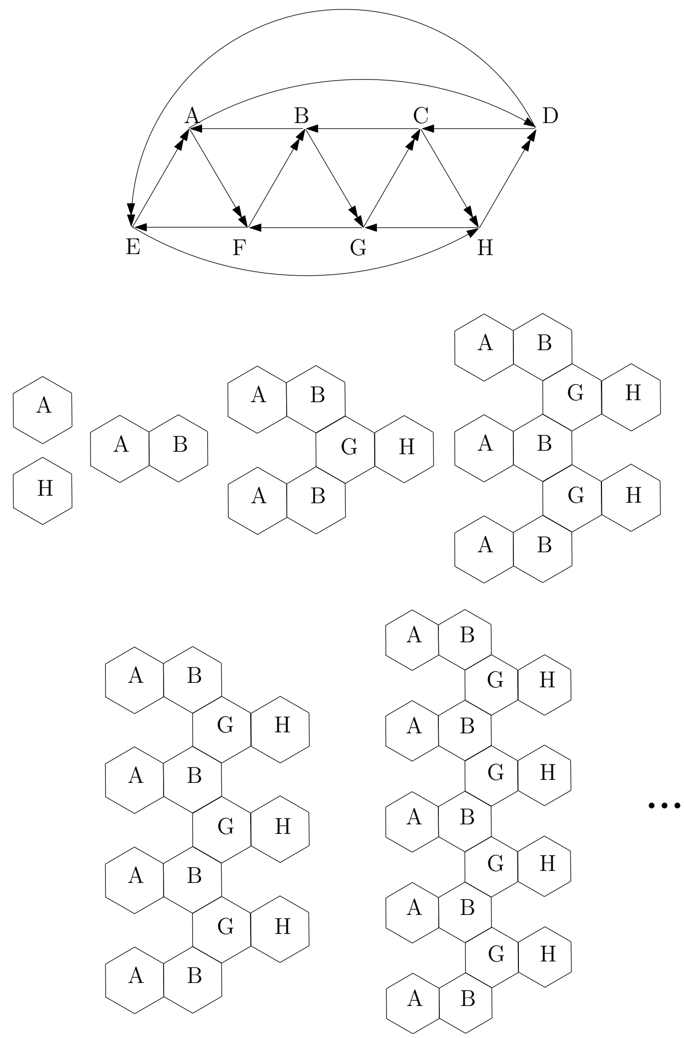


FIGURE 4.5. The mutation sequence $A, H, B, G, B, G, B, G, \ldots$ carries this family of hexagonal graphs (embedded in the hexagonal grid)

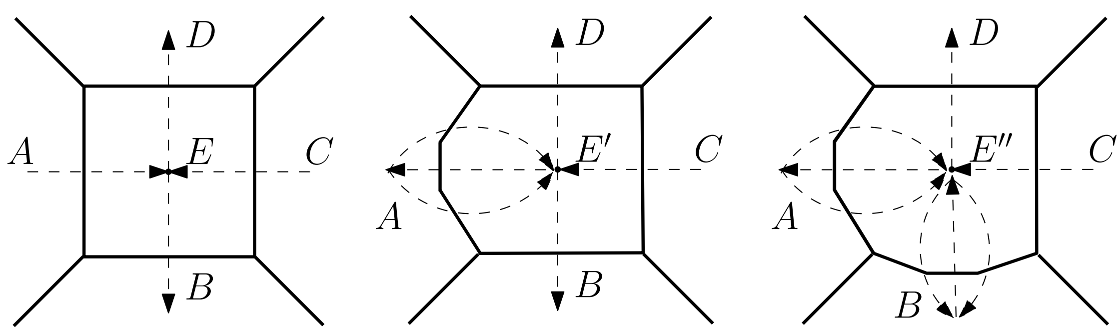


FIGURE 4.6. Matchingwise equivalent graphs

since its number of edges has increased by two. It also has one more factor of x_{v_A}, as the neighbor face A now shares two more edges with E' and one additional A-cycle is formed.

Hence $m(E') = m(E)x_{v_E}x_{v_A}$ and we have $w(E')m(E') = w(E)m(E)$. Alternatively, if we use Speyer's weighting, it is more direct to see that $\bar{w}(E) = \bar{w}(E')$. This argument applies equally well when we consider a union of faces containing E: for example, $w(D \cup E)m(D \cup E) = w(D \cup E')m(D \cup E')$.

As an example, we dualize cyclic quivers of n vertices. In such a quiver, each vertex has degree 2, we add two (or any even number) arrows to each vertex and dualize around all vertices. From above discussion, weights of the trees living inside the dual graph does not depend on the location and number of the (pairs of opposite) arrows added. Moreover, we do not care to which vertices such dashed arrows are connected on the other side.

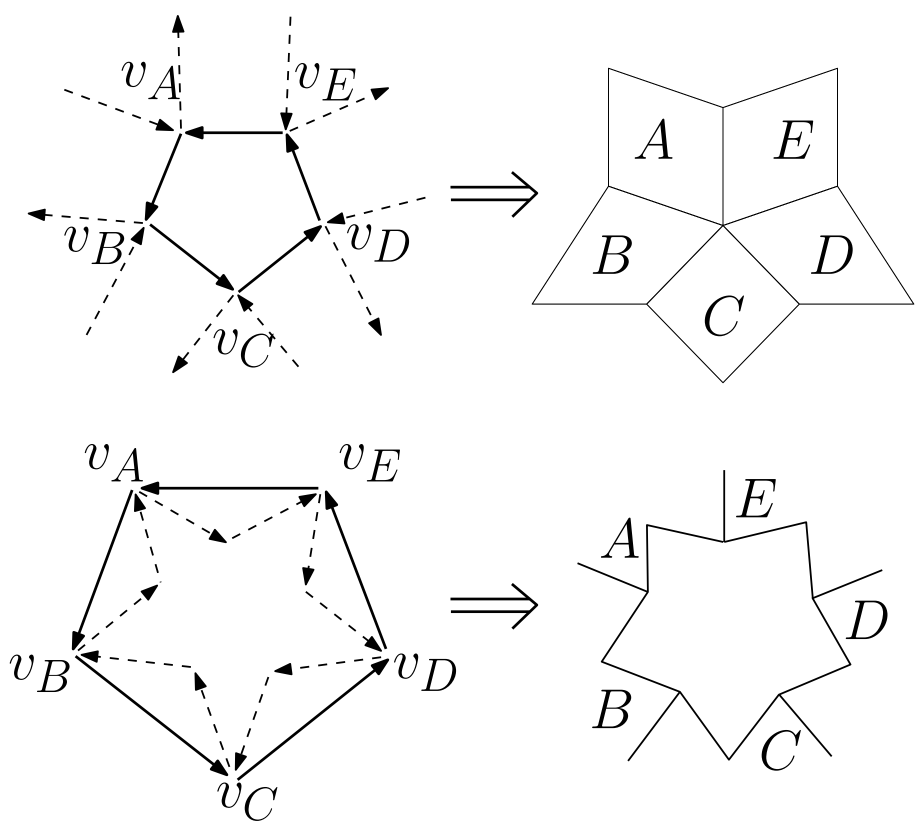


FIGURE 4.7. Dualizing the cyclic quiver on five vertices. We can add extra arrows on the outer face of the cycle (above) or do the opposite (below). Two resulting planar bipartite graphs look quite differently. Above graph has five finite faces sharing a vertex. Below graph consists of five infinite faces, and a "hollow" in the center. One can observe that a snake in one of them has the same weight as its corresponding snake on the other planar bipartite graph (matchingwise equivalent).

In [1], it was shown that these cyclic quivers are mutation equivalent to the quiver associated with the cluster algebra of type D_n. We also know there are n^2 distinct cluster variables, from [4]. When we count the number of snakes in the corresponding dual planar bipartite graphs, we see there are n snakes of order i for any $1 \leq i \leq n-1$. An observation is that when we have a tree $\mathcal{T}$ with maximal face T and mutate in any natural sequence, the denominator of y_{v_T} equals $\prod_{F_\alpha \in \mathcal{T}} x_\alpha$. One can also observe that this denominator never cancels out with

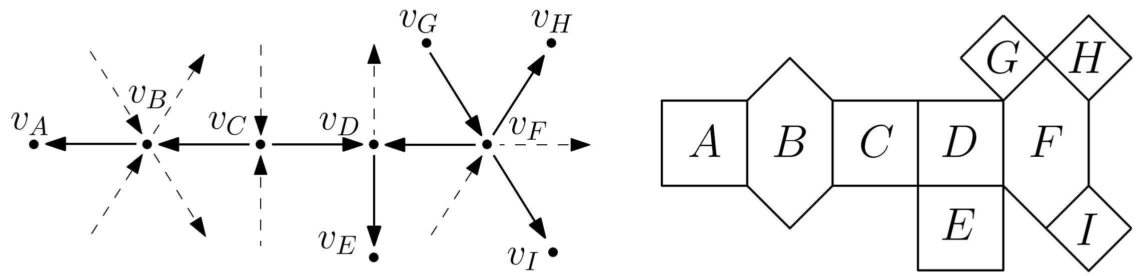


FIGURE 4.8. A tree quiver and its dual tree graph

the numerator. Therefore, y_{v_T} is distinct for all $n(n-1)$ snakes. This shows that every non initial cluster variable in this form of D_n represents the weight of a snake. This is one of a few instances where all non initial variables can be represented by snakes in the dual graph.

Now, if some vertices in V were not cyclic, we can embed Q into a quiver which is cyclic at vertices in V; simply force each vertex in V to be cyclic by adding arrows of appropriate directions. Again, we cannot add arrows between vertices in V. For each vertex, there is a minimal number of mandatory arrows to make it cyclic. After that, we can add pairs of opposite arrows to each vertices, which will not affect the weight of its dual. That is, chosen a set of vertices V in Q, its dualization after embedding into a cyclic quiver is matchingwise unique.

As an example of this embedding, we can consider a particular family of quivers, which will be dual to trees. We will say a quiver is a tree if when viewed as an undirected graph, it is connected with no cycles (usual definition of a tree in graph theory). Given a tree quiver Q, we can embed it to a cyclic quiver by adding arrows and dualize "around" vertices of Q to obtain a planar 2-connected bipartite graph, $\mathcal{T}_Q$, which is indeed a tree we considered in the previous section, without prescribed partial order. When we dualize $\mathcal{T}_Q$, we recover Q (with uninteresting vertices w with $x_w = 1$). That is, there is a bijective correspondence between tree quivers and (matchingwise) equivalence classes of trees, if we forget about partial order.

Now we impose a partial order on $\mathcal{T}_Q$. We have many choices as any face of $\mathcal{T}_Q$ can be chosen as the maximal face. Then it induces a partial order on the quiver Q and naturally we consider natural mutation sequences correspond to this particular order. In this situation, all neighbor faces have variables 1 associated to them, so in particular, $m(\mathcal{T}_Q)$ is independent of the partial order on $\mathcal{T}_Q$. The weight $w(\mathcal{T}_Q)$ is also independent, so $m(\mathcal{T}_Q)w(\mathcal{T}_Q) = y_{v_T}$ (T is the maximal face) is independent. Equivalently we can use Speyer's weighting, $\bar{w}(\mathcal{T}_Q) = y_{v_T}$. That is, the cluster variable y_{v_T} is indeed something independent on the choice of T but it depends only on the shape of Q. Hence we can define a variable y_Q associated with Q.

Corollary 4.3. *Assume Q is a tree quiver. If we follow any natural mutation sequence on Q with respect to any partial ordering on $\mathcal{T}_Q$, we get the same variable; $y_Q = m(\mathcal{T}_Q)w(\mathcal{T}_Q) = \bar{w}(\mathcal{T}_Q)$.*

A particular example of a tree quiver is the one corresponds to the cluster algebra of type D_5. We add arrows to turn it into a cyclic quiver, which is shown on the right. Dualize this cyclic quiver and obtain a planar graph G. Look at the faces of G which correspond to the vertices from Q. Then we count that this shaded area has seventeen trees. Therefore, three noninitial cluster variables are not covered by trees. This example was motivated from Musiker's previous work; one can find out in [8] that the other three graphs have two central hexagons and extra arcs. Such extra arcs are also found in [9].

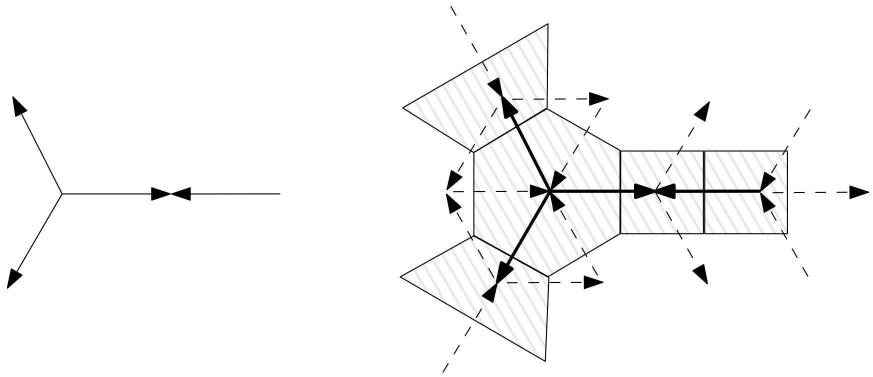


FIGURE 4.9

For another example, we analyze the case of A_n. For each n, consider the linear quiver which correspond to the cluster algebra of type A_n where arrows are alternating. As an example, when $n = 5$ we have

$$1 \longrightarrow 2 \longleftarrow 3 \longrightarrow 4 \longleftarrow 5.$$

Let us name this quiver Q_n. Vertices are named $1, 2, ..., n$ and their corresponding faces will be named $F_1, ..., F_n$. We embed all of Q_n into the infinite square grid quiver. Its dual is the infinite square grid, and we consider trees consisting of squares $F_1, F_2, ..., F_n$. These are snakes of the form $\cup_{i=j}^{k} F_i$ for all pairs of integers (j, k) where $1 \leq j < k \leq n$ and hence we precisely recover the corresponding result in [8]. We can repeat this analysis for any linear quiver which is mutation equivalent to Q_n. This interpretation is not new; see [10].

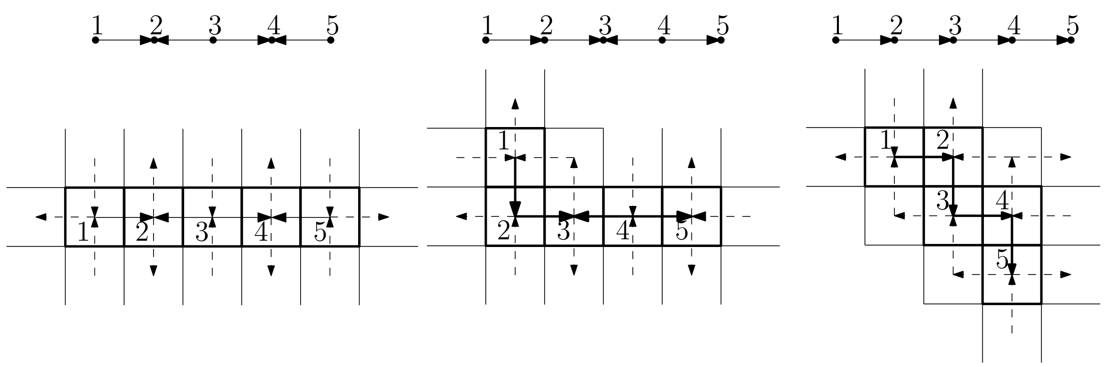
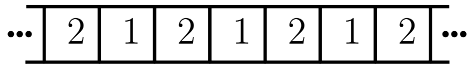
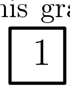
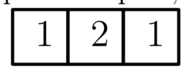
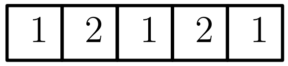


FIGURE 4.10. Three linear quivers of the type A_5 and their embeddings

4.3. **Unfolding.** If we have a quiver with multiple arrows, it is more efficient to unfold first and dualize, since we might observe a spine and then superposition applies. In particular, if we have a doubled arrow between two vertices of Q, it unfolds to be an infinite spine. The simplest possible example of this type is the following: $1 \bullet \Rightarrow \bullet 2$. This was covered in [9], with Speyer's weighting (which will be equivalent to Kenyon's weighting). We can fold it to be the infinite linear quiver $\cdots \longrightarrow 2 \longleftarrow 1 \longrightarrow 2 \longleftarrow 1 \longrightarrow 2 \longleftarrow \cdots$. This is a subquiver of the square grid quiver, so we naturally consider its dual living inside the square grid:



Recall that a spine is defined as a (possibly infinite) snake which is straight at all its faces. Equivalently, a snake graph is a spine if we have alternating arrows along vertices in its dual quiver. So this graph is a spine, and we see the mutation sequence 1,2,1,2,... corresponds to subgraphs , , , and so on. Alternatively, one can observe that this quiver corresponds to $\{P(0,0)_n\}_{n\geq 1}$ considered in 4.1.1.

As a slightly more complicated example, consider the cyclic triangular quiver with doubled arrows. Cluster variables from this quiver are called Markoff polynomials in [11]. We can unfold this quivers to be a subquiver of the infinite triangular quiver. Hence we consider the corresponding hexagonal subgraph of the dual infinite hexagonal grid. This $2 \times \infty$ hexagonal graph (embedded in the hexagonal grid) is shown on the right. In Figure 4.11, the infinite snake *...acacac...* qualifies as a spine, since the direction of arrows $a \leftarrow c$ and $c \rightarrow a$ alternates. Therefore, superposition applies to mutation sequences $a, c, a, c, ...$ and $c, a, c, a, ...$ That is, if we never mutate at b, all cluster variables are represented as snakes embedded in this $2 \times \infty$ hexagonal graph. Following the results from [11], we can attempt to represent all cluster variables as snakes living inside this graph. The following is merely a re-interpretation, so we will omit the proofs.

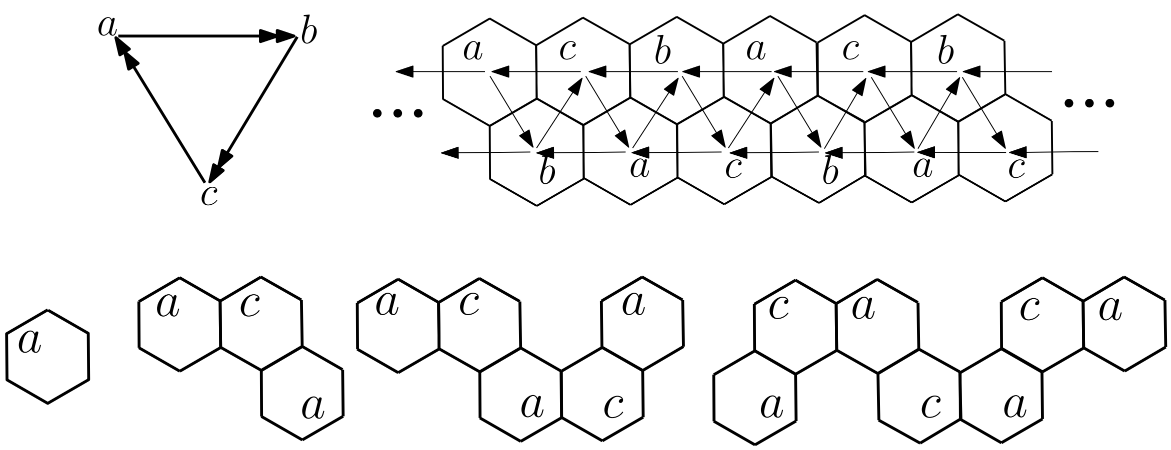
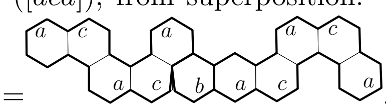


FIGURE 4.11. Unfolding of a triangular quiver and first few superpositions along *...acacac...*, namely $[a]$, $[ac]$, $[aca]$, and $[acac]$.

We will illustrate the procedure through an example. Given a finite mutation sequence $a_1...a_n$, let us name the snake it carries as $[a_1a_2...a_n]$. We only consider reduced sequences; that is, $a_i \neq a_{i+1}$ for all i. Consider mutation sequence $acab$. The mutation sequence aca carries the straight snake of length five ($[aca]$), from superposition. When we mutate at b, we claim that it carries the snake $[acab] =$ , where its monomial is the square of the monomial of $[aca]$. This snake is obtained by joining two $[aca]$s through a face of b and "splitting" the vertical edge shared by faces c and b. This snake is symmetric, and at each ends it contains snakes $[ac]$ and $[aca]$ which are the snakes carried by vertices c and a respectively at this stage. Now when we mutate at c (resp. a), c (resp. a) carries the snake obtained by combining two above snakes in a way that their intersection is $[ac]$

(resp. $[aca]$). In general, when we have a sequence $a_1 a_2 ... a_n$, where a_n is a previously mutated vertex, (say a_t is the latest) the snake $[a_1 ... a_n]$ is obtained by combining two snakes $[a_1 ... a_{n-1}]$ in a way that their intersection is precisely $[a_1, ... a_t]$. This works since we can prove by induction that any snake $[a_1 ... a_n]$ contains the snake $[a_1 ... a_{n-1}]$ at each ends (which would contain $[a_1 ... a_{n-2}]$ and so ons. We need a separate definition for $m([a_1 ... a_n])$, and one can verify that it equals $m([a_1 ... a_{n-1}])^2 / m([a_1 ... a_t])$. If a_n were not previously mutated, connect two $[a_1 ... a_{n-1}]$s through a face of a_n to obtain $[a_1 ... a_n]$, and $m([a_1 ... a_n]) = m([a_1 ... a_{n-1}])^2$.

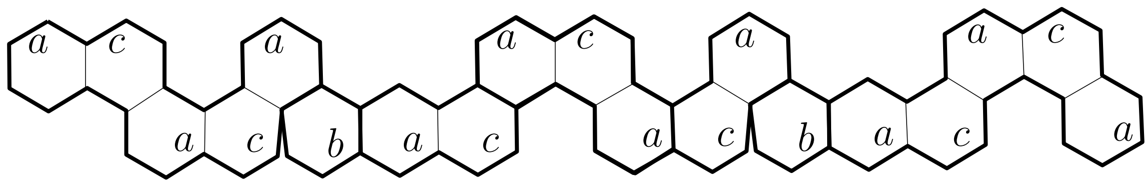


FIGURE 4.12. $[acabc]$

Remark 4.4. We have observed connections between a planar bipartite graph and the cluster algebra its dual quiver generates, through tree and superposition phenomena. Although they are of limited use, it is likely that such phenomena are special instances of more general phenomenon. Still, the dual planar graph of a quiver is not "big" enough to cover all cluster variables from its dual quiver, as the need for a split of edges in the above example or extra arcs from [8] suggests. It is possible that there exists a higher-dimensional construction containing more information about the cluster algebra from its corresponding quiver.

Acknowledgments

This research was done in the University of Minnesota REU program, which was supported by NSF grant number DMS-1001933. I would like to thank Vic Reiner, Gregg Musiker, and Pavlo Pylyavskyy for their guidance and support. I owe my deepest gratitude to Gregg Musiker; he not only proposed the original conjecture and the cyclic grid quiver but also he gave me countless insightful suggestions throughout the REU program. It should be mentioned that I relied on his Sage package with Christian Stump [6] to test conjectures. I also thank Shiyu Li and Gregg Musiker for reading through and commenting on the drafts.

References

[1] Aslak Bakke Buan and Hermund André Torkildsen. The number of elements in the mutation class of a quiver of type D_n. *Electron. J. Combin.*, 16(1):Research Paper 49, 23, 2009.
[2] P. Caldero, F. Chapoton, and R. Schiffler. Quivers with relations arising from clusters (A_n case). *Trans. Amer. Math. Soc.*, 358(3):1347–1364, 2006.
[3] Sergey Fomin and Andrei Zelevinsky. Cluster algebras. I. Foundations. *J. Amer. Math. Soc.*, 15(2):497–529 (electronic), 2002.
[4] Sergey Fomin and Andrei Zelevinsky. Cluster algebras. II. Finite type classification. *Invent. Math.*, 154(1):63–121, 2003.
[5] R. Goncharov, A. B.; Kenyon. Dimers and cluster integrable systems. *eprint arXiv:1107.5588*, 2011.
[6] Christian Stump Gregg Musiker. A compendium on the cluster algebra and quiver package in sage. *arXiv:1102.4844v2*, 2011.
[7] Eric H. Kuo. Applications of graphical condensation for enumerating matchings and tilings. *Theoret. Comput. Sci.*, 319(1-3):29–57, 2004.
[8] G. Musiker. A graph theoretic expansion formula for cluster algebras of classical type. *Annals of Combinatorics*, 15:147–184, 2011.

[9] Gregg Musiker and James Propp. Combinatorial interpretations for rank-two cluster algebras of affine type. *Electron. J. Combin.*, 14(1):Research Paper 15, 23 pp. (electronic), 2007.

[10] Gregg Musiker and Ralf Schiffler. Cluster expansion formulas and perfect matchings. *J. Algebraic Combin.*, 32(2):187–209, 2010.

[11] James Propp. The combinatorics of frieze patterns and markoff numbers. *arXiv:math/0511633v4*, 2005.

[12] David E. Speyer. Perfect matchings and the octahedron recurrence. *J. Algebraic Combin.*, 25(3):309–348, 2007.