Dimer Interpretations in Cluster Algebras

Piriyakorn Piriyatamwong and Caledonia Wilson

September 17, 2018

Abstract

We give a new combinatorial interpretation of the cluster variables in cluster algebras of finite types using mixed dimers.

1 Introduction

We consider all possible F-polynomials generated by a source-inducing mutation sequence of a quiver whose underlying graph is a Dynkin diagram of type D_n. Each of these F-polynomials contains the term 1 and so can be described using a hexagon-square model, where the term 1 designates a minimal matching of a graph from our model. We then define a transformation of the hexagon-square model called a *flip* and assign each hexagon and square a unique *accounting factor* that keeps track of flips applied to that hexagon and square. The F-polynomial is interpreted as the sum of the accounting factors from all the possible flip sequences. We show that there is a correspondence between the quiver and the hexagon-square graph.

By using a source-inducing mutation sequence on D_n-type quiver, we guarantee that all of the resulting quivers are also D_n-type, which allows us to create a table of hexagon-square graphs with minimal matchings. This table tracks the correspondence between quivers resulting from a repeated mutation sequence, and their respective F-polynomials, with our hexagon-square model. A diamond relation is found in this table: the product of F-polynomials on the diagonal of the "diamond" differs from the product of F-polynomials on the anti-diagonal by one term. In terms of the hexagon-square model, this means that the smash of two hexagon-square graphs on the diagonal and all of their possible matchings can be split into the two hexagon-square graphs on the anti-diagonal and all but one of their possible matchings. We prove this diamond relation by induction.

By better understanding the ways two hexagon-square graphs with perfect matchings can be combined (or, as we prefer, "smashed") together to produce hexagon-square graphs with mixed dimers, we claim that given any acyclic quiver with underlying graph of type D_n, we can successfully generate a corresponding hexagon-square graph and its minimal matching.

2 Background

2.1 Preliminary Definitions: The Cluster Algebra

Definition 2.1. A *cluster algebra*, denoted as A, is a subalgebra of $\mathbb{Q}(x_0, \ldots, x_{n-1})$ defined by generators and relations, starting with the initial cluster $\{x_0, \ldots, x_{n-1}\}$ and a mutation in direction j, denoted as μ_j:

$$\{x_0, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_{n-1}\} \xrightarrow[\mu_j]{\text{mutation in direction } j} \{x_0, \ldots, x_{j-1}, x_j', x_{j+1}, \ldots, x_{n-1}\}$$

generates a new cluster. We call x_i an initial *cluster variable* of i.

Theorem 2.1 (Fomin-Zelevensky 2001)**.** *In a cluster algebra, any cluster variable can be written as a Laurent polynomial in the initial cluster over* $\mathbb{ZP}$.

Theorem 2.2 (Lee-Schiffler 2013)**.** *In a* skew-symmetric *cluster algebra, any cluster variable can be written as a Laurent polynomial in the initial cluster with* nonnegative integer linear combinations *of elements in* $\mathbb{ZP}$.

Definition 2.2. A cluster algebra is of *finite type* if its set of cluster variables is finite.

Definition 2.3. A *quiver* is an oriented graph. An *acyclic* quiver refers to a quiver that contains no cycles.

Throughout our work, we only consider an unweighted, finite quiver without self-loops or 2-cycles. Such a quiver Q of n vertices $\{0, \ldots, n-1\}$ defines a cluster algebra $A(Q) = A$ contained in $\mathbb{Q}(x_0, \ldots, x_{n-1})$ with the initial cluster $\{x_0, \ldots, x_{n-1}\}$.

Definition 2.4. Given a quiver Q of n vertices $\{0, \ldots, n-1\}$ without self-loops or 2-cycles, a *mutation in direction* j, μ_j, transforms it into a new quiver Q' according to the following rules:

- For every 2-path passing j such that $k \to j \to i$ in Q, add a new edge $k \to i$ in Q'.
- Reverse direction of all edges incident to j in Q to form Q'.
- Delete all 2-cycles in Q'.

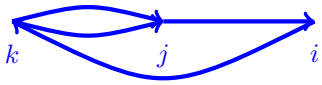
Furthermore, a new cluster variable at j, x'_j, is related to the initial x_j with this expression:

$$x_j x'_j = \Pi_{i \to j} x_i + \Pi_{j \to k} x_k$$

where the multiplication term is 1 if no edges i, j suit these conditions and edges directed the same way between two vertices are accounted for with multiplicity.

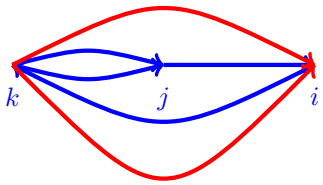
We provide Example 2.1 to illustrate how mutation works.

Example 2.1. Suppose we have the following quiver

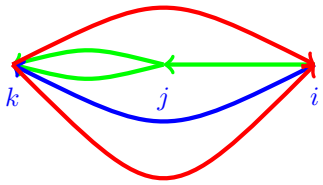


Mutation at direction j would give us the following sequence of transformations:

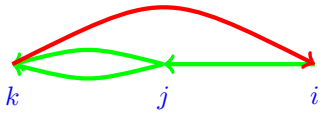
- First, *For every 2-path passing j i.e. $k \to j \to i$ in Q, add a new edge $k \to i$ in Q'.*



- Next, *Reverse direction of all edges incident to j in Q'.*



- Finally, *Delete all 2-cycles in Q'.*



We have the following relation for a new cluster variable x'_j: $x_j x'_j = x_i^2 + x_k$. In the end, our new cluster becomes $\{x_0, \ldots, x_{j-1}, x'_j = \frac{x_i^2+x_k}{x_j}, x_{j+1}, \ldots, x_{n-1}\}$.

Lemma 2.3. *Applying mutation twice at the same vertex yields the original quiver:* $\mu_j(\mu_j(Q)) = Q$ *for every* $j \in \{0, \ldots, n-1\}$.

Proof. The only edges affected by mutation at j are those incident to j and those $i \to k$ that are generated from the initial $i \to j$ and $j \to k$ edges. Mutating twice flips the direction of those incident to j twice, yielding the original directions. Mutating the first time adds directed edges $i \to k$ to $\mu_j(Q)$ from the initial $i \to j$ and $j \to k$ in Q, while $i \to j$ and $j \to k$ in Q are also reversed to $j \to i$ and $k \to j$ in $\mu_j(Q)$. Mutating the second time then adds directed edges $k \to i$ into $\mu_j(\mu_j(Q))$ which form two-cycles with the $i \to k$ edges from the first mutation. These two-cycles have to be removed, resulting in $\mu_j(\mu_j(Q)) = Q$. □

Theorem 2.4 (Fomin-Zelevinsky 2002)**.** *A finite, connected, unweighted quiver is mutation equivalent to an oriented Dynkin diagram of the following types:* A_n, D_n, E_6, E_7 *or* E_8*. In other words, a quiver generates a finite cluster algebra if and only if there exists a sequence of mutations which transforms the quiver into an oriented Dynkin diagram of type* A_n, D_n, E_6, E_7 *or* E_8*.*

The subscript of the type name denotes the number of vertices. Our convention is to start labeling from 0 on the bottom right and increment the index as we go leftward and upward.

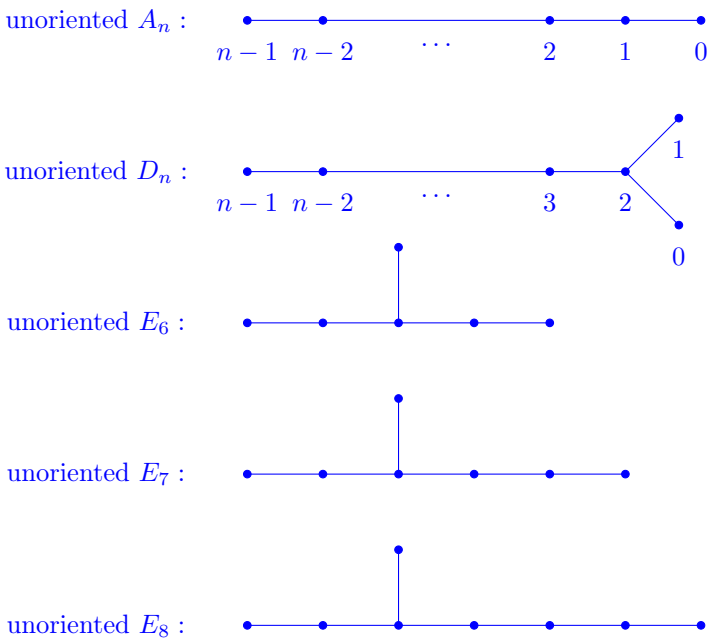
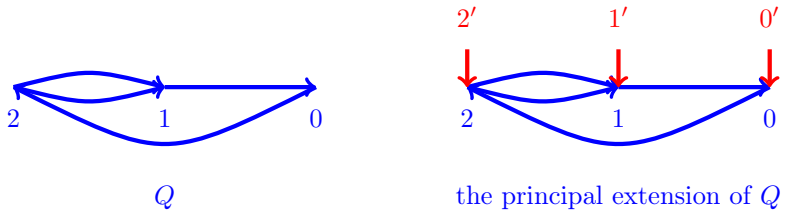


Figure 1: unoriented Dynkin diagrams of type A_n, D_n, E_6, E_7 and E_8

Definition 2.5. The *principal extension* of a quiver Q is formed by adding shadow vertices i' for every vertex i of Q and shadow edges $i' \to i$. Each shadow vertex i' is assigned a shadow cluster variable y_i.

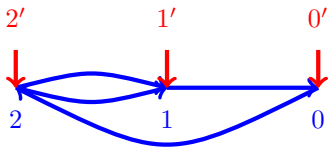
We treat those shadow vertices, edges, and cluster variables as true shadows, i.e. it is not possible to mutate at shadow vertices.



Definition 2.6. Given a quiver Q and its principal extension, the *F-polynomial* of a vertex i is obtained from the current cluster variable of i by setting every initial cluster variable x_j to 1.

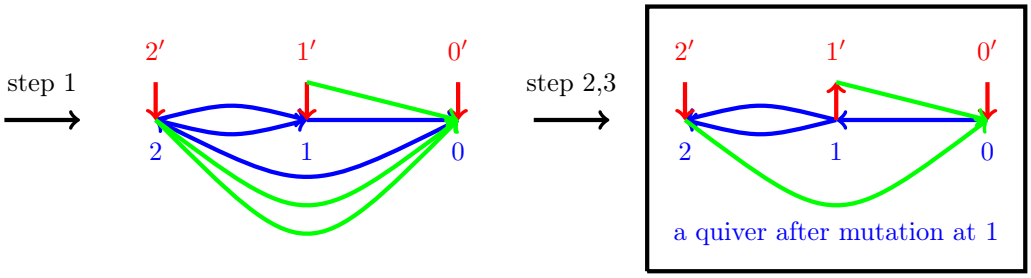
The important point is that we obtain the F-polynomial from the *current* cluster variable, not the initial one. By setting every initial cluster variable x_j to 1, the F-polynomial is always written in terms of the shadow cluster $(y_0, \ldots, y_{n-1})$. We give Example 2.2 to illustrate how to obtain the F-polynomial.

Example 2.2. Suppose we have the following quiver Q and its principal extension:



The initial cluster is $\{x_0, x_1, x_2\}$. Setting every x_j to 1, the initial F-polynomial for every vertex is simply 1.

Now, we proceed to mutate at vertex 1 (noting we cannot mutate at shadow vertices). We obtain the following quiver:



This mutation gives us the expression: $x_1 x_1' = \Pi_{i \to 1} x_i + \Pi_{1 \to k} x_k = x_2^2 y_1 + x_0$ i.e. $x_1' = \frac{x_2^2 y_1 + x_0}{x_1}$. By setting every x_i to 1, we have that the F-polynomial of vertex 1 is now $\frac{1^2 y_1 + 1}{1} = y_1 + 1$.

Corollary 1. *In a cluster algebra, an F-polynomial can be written as a polynomial in terms of the shadow cluster $(y_0, \ldots, y_{n-1})$. Moreover, if the cluster algebra is skew-symmetric, the F-polynomial only has coefficients as nonnegative integer linear combinations of elements in $\mathbb{ZP}$.*

Proof. This statement follows from Theorem 2.1 of Fomin-Zelevensky 2002 and Theorem 2.2 of Lee-Schiffler. Since mutation at shadow vertices is not allowed, by setting every initial cluster variable x_j in the current cluster variable to 1, the terms with powers of x_j vanish, leaving a polynomial in terms of the shadow cluster $(y_0, \ldots, y_{n-1})$. □

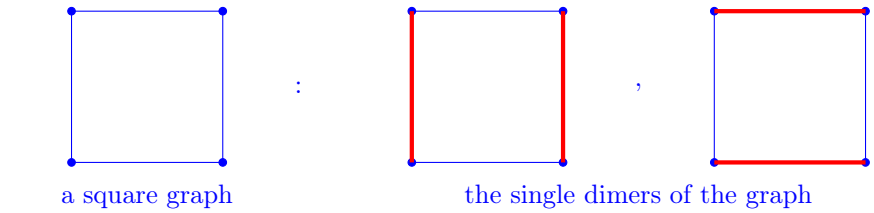
3 Main Results

3.1 The Single Dimer and the Square-Free F-Polynomials

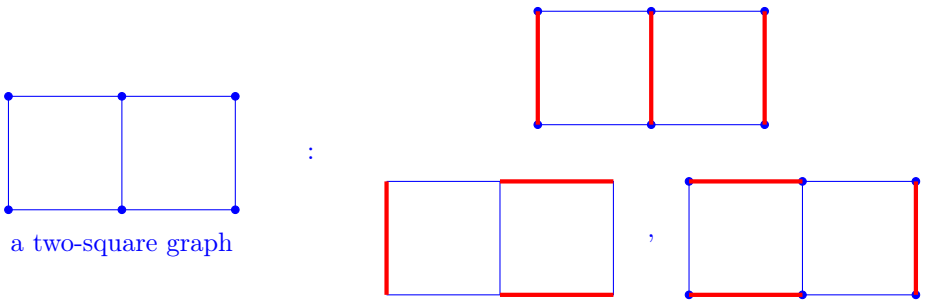
Following previous work, we are able to find a bijection between a quiver of finite type and its F-polynomials (with no squared terms) with the hexagon-square model, whose edges have a single dimer covering. This bijection maps the F-polynomial of a quiver, a result of some mutation sequence, to a specific transformation of the matching of the hexagon-square model.

Definition 3.1. Given a graph G, a *single dimer*, or equivalently a *perfect matching*, on G is a set of edges such that each vertex is touched exactly once.

A single dimer is not unique; there might be more than one single dimer for a given graph. See the following illustration:



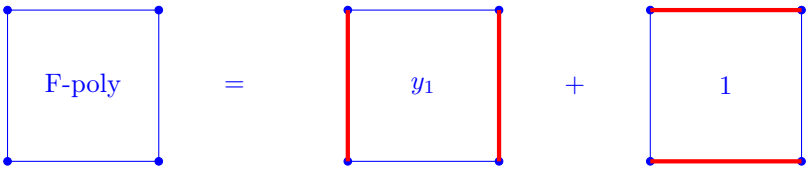
a square graph the single dimers of the graph



a two-square graph the single dimers of the two-square graph

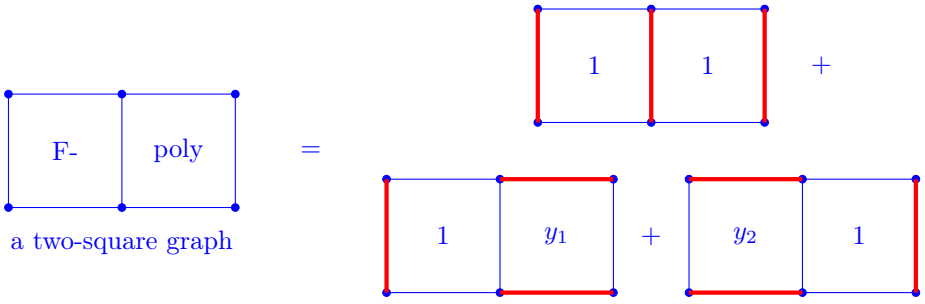
Recall from Example 2.2 that the F-polynomial of vertex 1 is calculated to be $y_1 + 1$. Combinatorially, we can associate this polynomial to a square graph, as it has two perfect matchings where two is exactly the number of terms in the polynomial. The value 1 is assigned to the minimal matching of the square, which we can think of informally as out starting point for doing flips on a dimer matching.

The term y_1 can be viewed as a factor that takes into account the result of this transformation. Overall, the F-polynomial is the sum of the accounting factors for all possible perfect matchings.



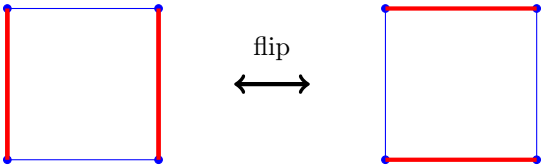
a square graph the single dimers of the square graph

For instance, a two-square graph could represent the F-polynomial of $y_2 + y_1 + 1$. In this graph, y_1 is the accounting factor of the transformation on the right square, and y_2 is the accounting factor for the flip of the left square. Notice that the accounting factor of each square is multiplied together to generate a term in F-polynomial.



a two-square graph the single dimers of the two-square graph

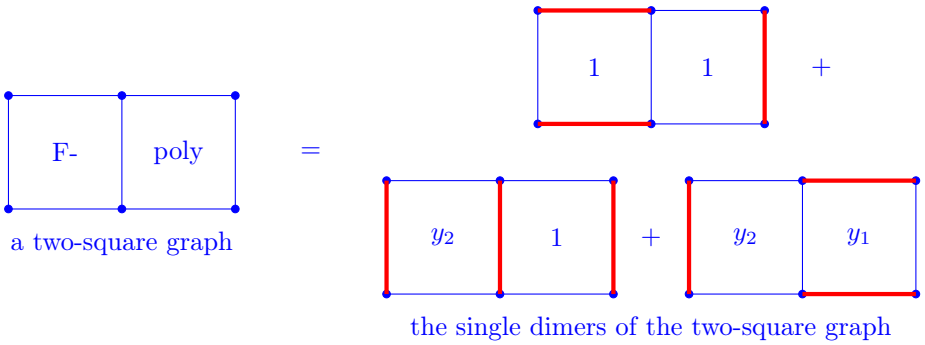
Definition 3.2. Given a square graph and its perfect matching, the *flip* of the square is a transformation that changes the initial perfect matching to another perfect matching.



the two perfect matchings of the square graph

Definition 3.3. An *accounting factor* is a way to record a flip at square i; this flip gives rise to the accounting factor y_i.

It is important to note that different minimal matchings usually represent different F-polynomials, even if both matchings are created on the same hexagon-square model. Compare the following example of a two-square graph to the above example. The F-polynomial of $1+y_2+y_2y_1$ is represented instead of y_2+y_1+1. This implies that the minimal matching is instrumental for determining the corresponding F-polynomial.

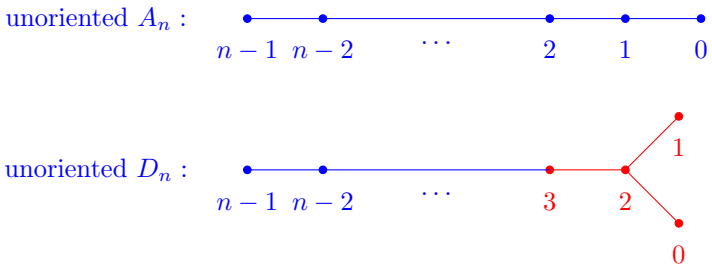


Lemma 3.1. *Flipping a single dimer square twice yields the original perfect matching.*

This lemma implies that a graph comprising of squares and its perfect matchings *cannot* represent F-polynomials with square terms. For example, y_1^2 would be the identity map in a single dimer model. We will introduce a new matching rule later to account for F-polynomials with terms that are not square-free. With the new mathing rule, flipping a square twice may not yield the original matching anymore.

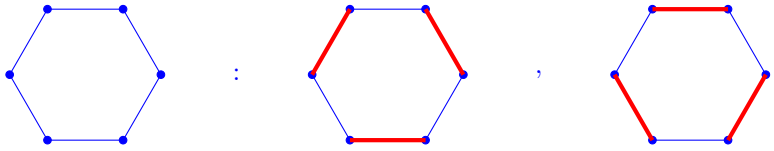
Based on our current examples, it might be tempting to think that a minimal matching is not unique; that many matchings can be considered minimal. However, this is not the case when we extend our consideration into more complicated graphs and matching rules. The more complicated the graphs and the matching rules, the more restrictive the choice of minimal matching. In some cases, the minimal matching is unique, but this is not guaranteed in all cases.

Now we consider examples for Dynkin diagram of types A_n and D_n. The A_n diagram is basically a line diagram, so the square model should still work well with quivers whose underlying graph is type A_n. Indeed, this graph can be formed by attaching the squares of vertices together based on the vertex order on the line. However, we run into a complication with the D_n diagram, where vertex 2 is incident to three vertices: 0, 1, and 3. It feels unnatural to attach squares to three of the four sides of the vertex 2 square.



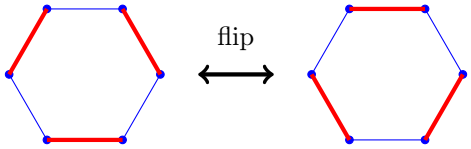
We could try to represent the vertex 2 with a three-sided triangle graph, as vertex 2 is incident to three other vertices. However, this idea fails to take into account the direction of the edges incident to vertex 2. Recall that the square model works well when each vertex is incident to at most two other vertices; with two choices of direction, there are at most *four* possible orientations of edges incident to a vertex, which is why the square graph, with *four* sides, can represent such a vertex. We are motivated to represent the vertex 2 with a six-sided graph – a hexagon graph.

Now, we need to extend the definitions of a *flip* and a *minimal matching* to the hexagon graph. Fortunately, the hexagon graph also has two perfect matchings, so it is simple to modify the original definitions as following:



a hexagon graph the single dimers of the hexagon graph

Definition 3.4. Given a hexagon graph and its perfect matching, the *flip* of the hexagon is a transformation that changes the initial perfect matching to another perfect matching.



the two perfect matchings of the hexagon graph

Lemma 3.2. *Flipping a single dimer hexagon twice yields the original perfect matching.*

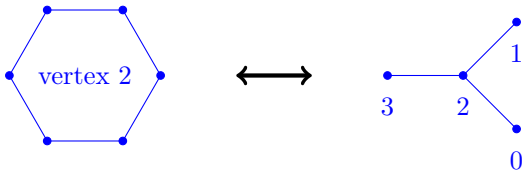
Definition 3.5. Given a graph comprising of squares and hexagons, a *minimal matching* is the dimer matching of this graph that allows us to flip in a specific sequence in order to reach all possible dimer coverings of that graph. This sequence does not contain flips that are involutions of previous flips. Within the poset of possible flip sequences, any path should lead to one option ordered the highest, denoted as the maximal matching. We use the subscript *min* to refer to the minimal matching of a given dimer.

3.2 The Attachment Rule for the Hexagon-Square Model

So far, we have casually referred to the "attachment" rule as a description for how we attach squares and hexagons together to represent a quiver. This rule is partly a personal convention and partly a requirement that arises from the directions of the edges of the quiver. This section will provide the attachment rule for the hexagon-square model D_n quiver and the intuition for how it was developed.

We would like to make it clear that a single graph of the hexagon-square model of the quiver cannot represent *every possible F-polynomial* generated from any sequence of mutations on a given quiver, as there is a large variation between F-polynomials, where some of them are very small, e.g. $1+y_1$, while others are very big, e.g. $y_2y_3y_4y_5+y_3y_4y_5+y_4y_5+y_5+1$. Though we can create graphs and matchings to represent every F-polynomial generated by a given quiver, our goal is to find the graph and matchings for the largest F-polynomial (the F-polynomial with highest degree). The largest F-polynomial amalgamates all the information of possible sequences of mutations in the quiver.

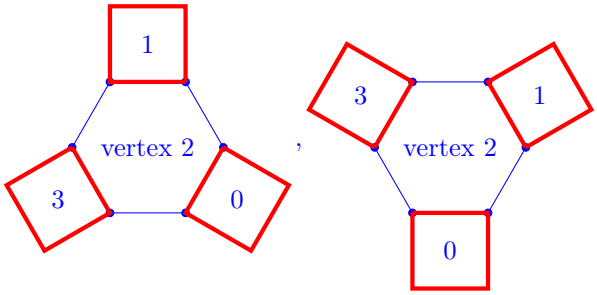
First, we observe a hexagon graph, which represents vertex 2 in the D_n quiver. Pay particular attention to the neighbors of vertex 2: vertices 0, 1, and 3. You can think of this neighborhood around vertex 2 as a D_4 subgraph within the D_n graph.



the hexagon graph the vertex 2 and its neighbors

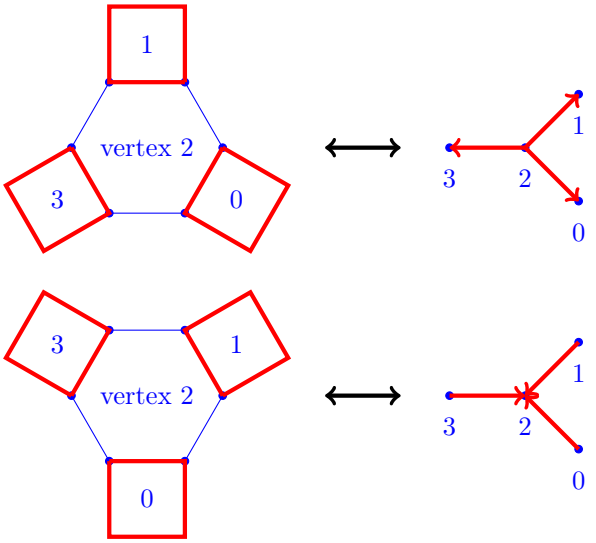
The most natural, and symmetric, choices of edge direction for the D_4 subgraph are to have all the edges from 0, 1, and 3 point either inward to 2 or outward from 2. Combinatorially, we would like to attach the three squares representing 0, 1, and 3 on three edges of the hexagon, and the only two symmetric choices of position are to distant each square by one edge. See the illustration below:

the two symmetric sub D_4 quiver

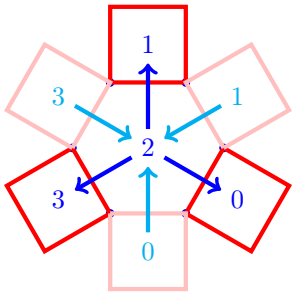


the two symmetric attachments of three squares on the sides of the hexagon

Our attachment convention is depicted below:

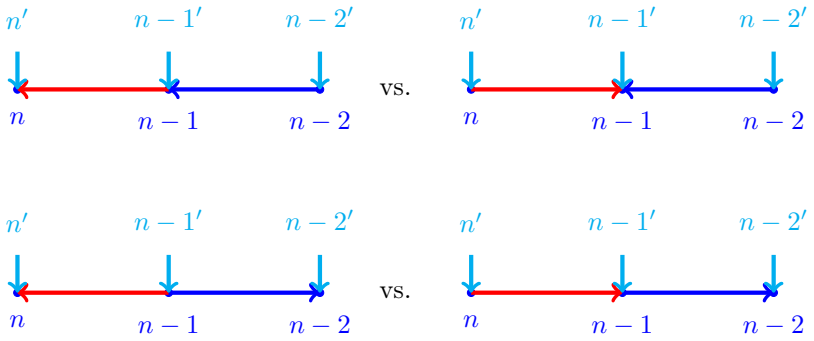


In general, we attach the square represent a vertex in one of two places depending on the direction of the arrow incident to vertex 2:



At this point, we agree on the attachment rule for the "head" of a D_n quiver; the neighborhood of vertex 2. Now we need to provide the attachment rule for the "tail" of the D_n quiver. In particular, we extend the tail by one vertex, mutate at the penultimate vertex, and observe the difference in the F-polynomial of the penultimate vertex in two cases.

We choose to mutate at the penultimate vertex because it links the newly added vertex to the original D_n. See a figure below for illustration:



We then mutate at vertex $n-1$, since. In the case where the edge at the end of the tail (red) has the same direction as the penultimate edge (blue), we obtain the expression:

$$F_{n-1}F'_{n-1} = y_{n-1}F_{n-2} + F_n$$

from the top left figure or

$$F_{n-1}F'_{n-1} = y_{n-1}F_n + F_{n-2}$$

from the bottom right figure. In the other case where the edges at the end of the tail has the opposite direction to the penultimate edge, we obtain the expression $F_{n-1}F'_{n-1} = F_nF_{n-2} + y_{n-1}$ from the bottom left figure or $F_{n-1}F'_{n-1} = F_nF_{n-2}y_{n-1} + 1$ from the top right figure.

Now, we count the number of terms in F'_{n-1}. For the case where the edge at the end of the tail (red) has the same direction as the penultimate edge (blue), we have

$$\#\text{ terms of } F_{n-1} = \#\text{ terms of } F_n + \#\text{ terms of } F_{n-2}$$

For the other case where the edges at the end of the tail has the opposite direction to the penultimate edge, we obtain the expression

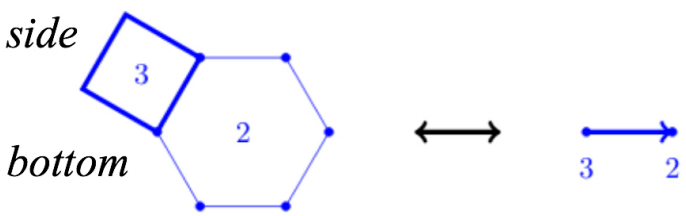
$$\#\text{ terms of } F_{n-1} = (\#\text{ terms of } F_n) \times (\#\text{ terms of } F_{n-2}) + 1$$

Obviously, the case where the edges at the end of the tail has the opposite direction to the penultimate edge yields more number of terms than the case where the edge at the end of the tail has the same direction as the penultimate edge.

How does the number of terms of the F-polynomial influence our hexagon-square model? The more terms in F'_{n-1}, the more possible flip-sequences we need to ensure exist in the hexagon-square model in order to accommodate more perfect matchings corresponding to terms in the F-polynomial.

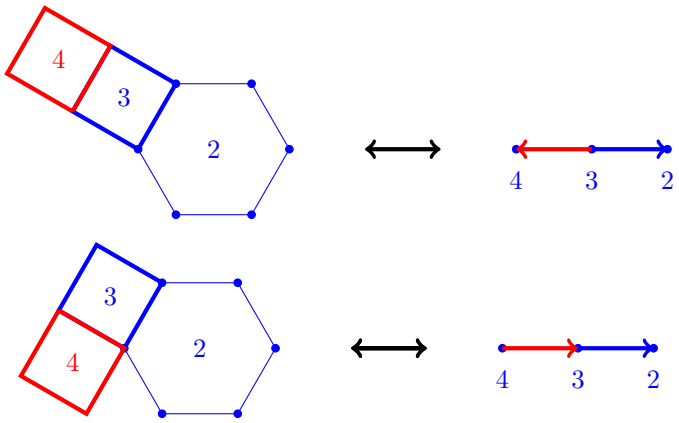
We look at a simple case where we have the D_4 quiver and its hexagon-square model. We would like to add vertex 4 at the end of the tail together with another square in the model. Formally, we refer to an *A-type tail* (or simply *tail*) as the portion of a dimer matching that corresponds to indices greater than or equal to 3 in a given quiver (with our standard labeling. The *length* of a tail refers to the highest indexed vertex in the quiver.

Suppose that the edge from vertex 3 points inward to vertex 2. The direction of edges from vertices 0 and 1 are irrelevant here, so we omit their parts.

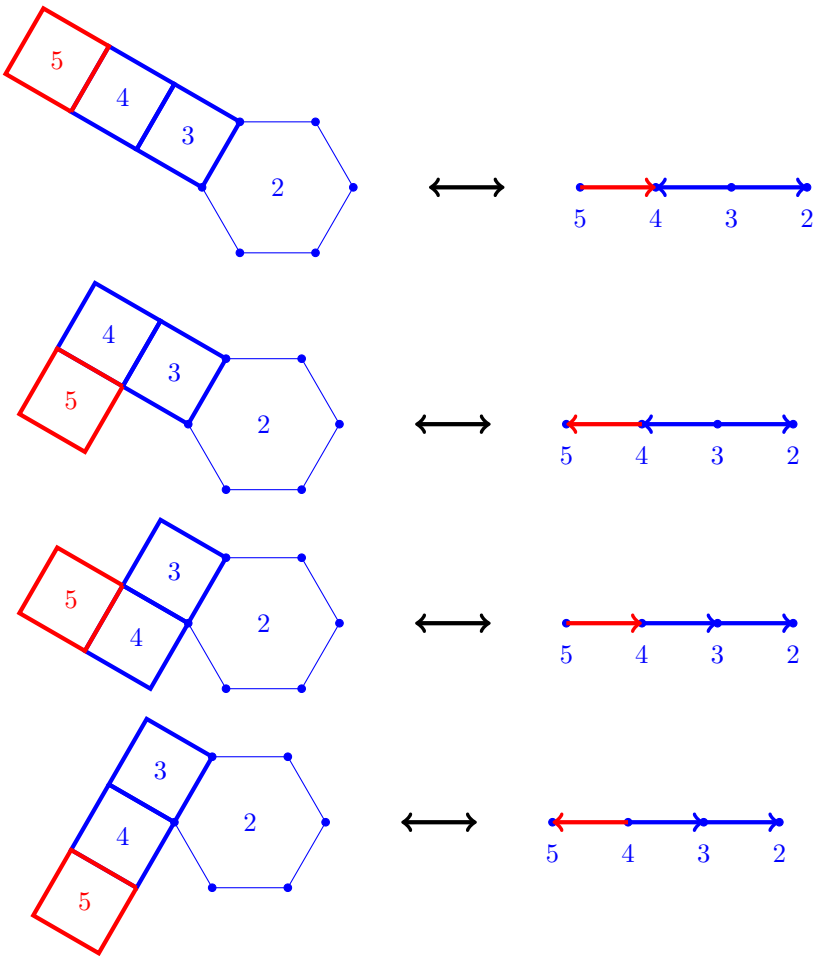


Naturally, we would like to attach square 4 to one of the three available edges of square 3. Two locations available for placing square 4: on the side of square 3 and on the bottom of square 3. We see that the less restrictive placement choice is the side of square 3, as square 4 would

be completely independent to hexagon 2. On the other hand, placing square 4 on the bottom of square 3 results in one vertex incident to three polygons, a more restrictive placement option. We match the quiver with the hexagon-square model based on this measure of restrictiveness.



This rule continues to hold for the later part of the tail of D_n, by placing a square graph to the most restrictive location when the edge "continues the trend" i.e. has the same direction as earlier edges that are already taken into account in the hexagon-square model. In summary, a part of the tail where the edge directions alternate corresponds to a sequence of square graphs in line, while a part of the tail where the edge directions are the same corresponds to a sequence of square graphs in zig-zag path. The D_5 case is depicted below for completion.



3.3 Introducing Mixed Dimers

Lemmas 3.1 and 3.2 implies that a graph comprising of squares and hexagons and its perfect matchings *cannot* represent F-polynomials with square terms, like y_1^2. We need a new matching

rule to take such squared terms into account, so that flipping twice at the same index is no longer an involution.

We find the answer to the problem of F-polynomials with squared terms in the form of mixed dimers, which is defined as following:

Definition 3.6. A *double dimer* is a matching in which each vertex is touched by exactly two edges. A term y_i^2 in the F-polynomial corresponds to a double dimer matching of i, where i is the index of a hexagon or square in our hexagon-square model.

Since not every term is squared in the highest degree term of F-polynomials (for example, $y_1y_2^2y_3$ has unsquared y_2), we need to use both single and double dimers, giving rise to a *mixed dimer* for the D_n case.

Lemma 3.3. *No cycles arise in a minimal matching of a mixed dimer.*

Proof. Let S be a mixed dimer of length j and let l^2 be its highest indexed squared term.

Suppose S has a minimal matching with a cycle at index g, for $2 \leq g < l$. Then g has two pairs of edges that we can flip at. Let's say we can flip at edge set A or edge set B. See Figure 2.

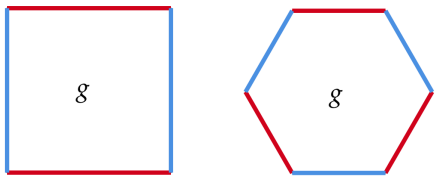


Figure 2: Let edge set A be the set of edges in red, and edge set B be the set in blue

Suppose without loss of generality, we flip at edge set A. Then we end up with doubled edges overlaying edge set B. But note that after performing this flip, the only way we can get doubled edges overlaying set A is to flip at g again to undo our first action, and then flip once more at g. We've just performed an involution by flipping twice at index g, which violates our definition of minimal matching. Therefore the minimal matching of S cannot contain any cycles.

□

Now that we've defined mixed dimers, the next logical question is if there are any restrictions we need to implement when we incorporate them into our hexagon-square model. We find that, in order to preserve the correspondence between our hexagon-square model and the F-polynomials, we need to introduce connectivity requirements for our matchings.

Our mixed dimer model has to obey **connectivity rules**: the vertices on the outer edges of squares 0 and 1 must have a path of edges in the matching in order to travel between them, as seen in Figure 3:

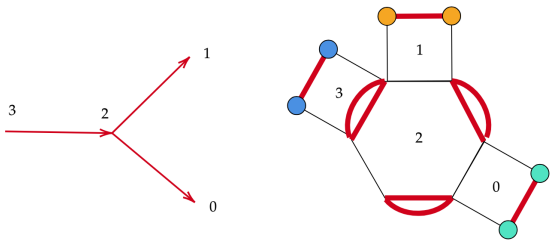


Figure 3: We have a path of edges between the orange colored vertices, and another path between the green vertices, and these paths preserve connectivity of the dimer

3.4 Working with the model

A *revolution* is a source-inducing mutation sequence through every vertex in the quiver. This mutation sequence ensures that going through n revolutions of the quiver produces all possible F-polynomials (R. Schiffler, A geometric model for cluster categories of type Dn).

A *batch* is the set of F-polynomials obtained after one revolution, numbered in order. For instance, after n revolutions, we will have n batches, where the first batch is generated by one revolution, and the kth batch is produced by the kth revolution.

We refer to the *global matching* as the minimum matching of the dimer configuration corresponding to the largest F-polynomial of all n batches. A *local matching* is the minimum matching of the dimer configuration corresponding to the largest F-polynomial in a batch (but not necessarily globally).

Within an *A-type tail* tail of a hexagon-square graph, we consider *interior edges*, which are edges in the perfect matching of the A-type tail that occur between two squares in a snake graph. All other edges in the perfect matching of the tail are deemed *boundary edges*. (Rabideau)

An *empty box* refers to a tile of the A-type tail that has no interior or boundary edges in its perfect matching.

Definition 3.7. *Smashing* two dimers A and B together refers to overlaying their perfect matchings on top of each other, such that if both dimers have a specific edge included in their perfect matching, this edge becomes doubled in the smash. We denote the resulting dimer of this smash $A\#B$.

Definition 3.8. *Splitting* a dimer refers to taking at least a double dimer matching and considering what two dimers could be smashed to result in this dimer. We say our original dimer has been split into these two dimers.

Consider some mixed dimer S, and suppose S has length j with l^2 as its highest indexed squared term, where $j \geq l+1$. We would like to split S into single dimers M and N, where we can consider N a single dimer matching of S, such that smashing with M adds any necessary edges to make S a proper mixed dimer matching. To define M and N formally:

Definition 3.9. To split S, we make the following requirements:

- The length of M's tail is l.
- The length of N's tail is j.
- N has a graph and matching including squares 0 and 1, where M does not.

See Figure 4.

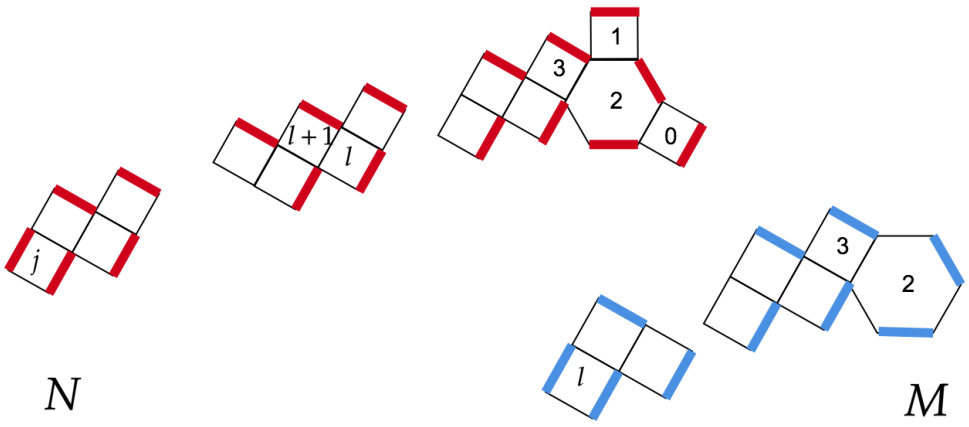


Figure 4: These are the minimal matchings for N and M for the D_n quiver with all arrows pointing to the right, where $j \geq l+1$

We assign the minimal matching in Figure 4 based on Definition 3.5. For a quiver in the all-right orientation, we note that the F-polynomial always contains the monomial y_k, where k is the highest indexed square. This arises from our source-inducing mutation sequence; in the all-right orientation, we always mutate at vertex k first to get a source there. This means we must be able to flip initially at square k. However, y_{k-1} does not appear as a monomial, so we are not able to flip at square $k-1$ initially. Therefore, we must choose the parallel edges on the boundary edges of square k for our minimal matching. In the all-right case, this behavior at the end of the tail forces the rest of the matching in order to create a perfect matching.

Given any acyclic D_n quiver, we assign its hexagon-square graph a mixed dimer minimal matching (corresponding to the highest-degree F-polynomial generated) by assigning squares corresponding with vertices that are sources parallel boundary edges, and with squares corresponding to sinks getting empty boundary edges in the matching. Section 3.7 gives a more thorough description of how to draw a minimal matching given an acyclic quiver with any orientation.

Lemma 3.4. *Splitting the minimal matching of a mixed dimer into minimal matchings of M and N is unique.*

Proof. First we assign edges that appear on squares in N that don't exist for M. This forces our hand- we must assign the tail (for $l < g \leq j$) to be the same matching as S_{min} at those indices. Similarly, N_{min} must have the same matching as S_{min} at squares 0 and 1, since M does not contain these squares.

Then the minimal matching for N is forced- there is only one way to create a single dimer matching given that the ending of both the tail and the behavior at 0 and 1 are determined. Note that we only fill in boundary edges from S onto the 0 and 1 squares for N, to ensure that N is a single dimer (and not a double). We fill in the rest of the matching for N. Note that this applies for any acyclic quiver, regardless of whether it is in the all-right orientation. Once we are given the behavior for the end of the tail and at 0 and 1 for N, and we can only pick edges in our matching from the matching of S, there is only one possible result for N's minimal matching.

Now, we know M has a length l. When we look at the square l in S_{min}, we will find either an interior edge paired with a doubled boundary edge, or we will find a pair of doubled boundary edges.

In the case of the interior edge, we assign M the matching that would give rise to this interior edge, and then we assign M any edges that were doubled in the matching of S_{min}. In the case of the doubled edges at l, we assign M the pair of edges at l that result in these doubled edges when smashed with N_{min}. Then we assign M any edges that were doubled in the matching of S_{min}. Assigning these edges will force our hand as to what the matching has to be for the hexagon. Thus we have matched both M and N without any choice in the way we matched them.

□

Lemma 3.5. *A single dimer snake graph has minimal matching that contains no interior edges.*

Proof. This proof is given by I. Canakci and R. Schiffler in "Cluster algebras and continued fractions". □

Lemma 3.6. *Suppose S is a mixed dimer of length j with l^2 as its highest indexed squared term, where $j \geq l+1$. There are k cycles in some matching of S if and only if there are 2^k ways to split this matching of S into distinct matchings of M and N.*

Proof.

$\Rightarrow$ Suppose there are k cycles in a matching of a mixed dimer S. We denote this matching S_γ, where γ denotes some flip sequence from the minimal matching of S that results in k cycles.

By Lemma 3.6, we can split the minimal matching of S into minimal matchings of N and M, denoted N_{min} and M_{min}. Then we note that squares 1 and 0 cannot contain cycles, since their outermost edges can be touched by only one edge in order for a perfect matching to exist. We also note that squares t such that $l+1 \leq t \leq j$ cannot contain cycles since these are covered by single dimers in S, and a cycle requires at least a double dimer matching.

Then we see that the only portions of the dimer S that may include cycles are portions g with indices labeled $2 \leq g \leq l$. Note that this graph in contained in both M and N. Note also that in the all-right case, we must always start flipping at the end of the tail for both M and N (see Lemma 3.11 in a later section), so the tail of S_γ for indices greater than or equal to l must reflect this

maximal matching. In any case, we are performing an allowed flip sequence in order to generate the cycles.

Consider some cycle (spanning a single hexagon or square) at index $2 \leq g \leq l$, where g could be the hexagon or a square. Then either M had a minimal matching of g and N had the flipped matching of g, or vice versa. We can make this assumption since S_{min} has no cycles, meaning $M_{min} \# N_{min}$ resulted in no cycles, so either M or N must flip at g to introduce a cycle at g, but not both.

Now consider a cycle at indices $2 < g < l$, where the cycle surrounds multiple indexed squares. In this region of the tail, we get that the matching of M_{min} and N_{min} is exactly the same. Then we need only flip at either N or M in sequence γ and smash with M_{min} or N_{min}, respectively, to obtain this cycle over multiple indexed squares (and possibly the hexagon). We can include indices 2 and l as part of a multiple index spanned cycle by noting that even though the matchings of N and M may not be exactly the same at 2 or l, S_{min} has no cycles, so we must have flipped at N or M to obtain a (mutiple index spanning) cycle.

Now, if S_γ has k cycles, for each cycle we choose to flip squares or the hexagon at either M or N. This results in 2^k options for smashing matches of N and M to get S_γ.

$\Leftarrow$ Suppose there are 2^k ways to split S_γ into distinct matchings of M and N. By Lemma 3.6, we know that there is one unique way to split S_{min} into M_{min} or N_{min}.

Since M does not have squares indexed greater than l, matchings on squares t such that $l+1 \leq t \leq j$ must be a result of some flip sequence on N. Similarly, since M does not have squares at 0 or 1, the matching at these squares is exactly the same as the matching for some N. We pick the matching of N that fulfills these criteria.

Now, for graph indices $2 \leq g < l$, we must get 2^k ways to split this mixed dimer. Note that every pair of interior edges in this portion of the tail of S_γ introduces a choice: we could have flipped M at g to get this interior edge pair, or we could have flipped N at g to get this interior edge pair. Note that every time we flip once at M or N, we introduce two interior edges.

We know that there is only one way to produce a doubled interior edge at g for $2 \leq g \leq l$: either we have an interior edge that was introduced from the minimal matching of M, and so we must flip N at the same square to double this edge, or, in an alternating portion of the tail, we must flip both M and N at g to produce this doubled edge (which results in a pair of doubled interior edges). Doubled interior edges do not give rise to multiple splitting options.

Suppose there were no cycles in S_γ. Then we can have no pair of interior edges that has a complete set of boundary edges between them. Then each time we flip at g of M (or N), we must be certain that our matching of N (or M) does not consist of the complementary boundary edges at g. However, remember that M and N have the same matching for graph indices $2 < g < l$, and that even if their matchings differ at indices l and 2, we have no cycles in S_{min} including at these vertices. Then if we are able to flip at g on M, we can either flip at g for N, or we can choose not to flip at g. Note that if flipping N at g results in doubled interior edges in the smash of M and N, these do not give rise to splitting options. Then suppose we flip at M but not at N. Since M and N have the same initial matching in this region of the tail for $2 < g < l$, flipping at g for M but not N (or for N but not M) means that N necessarily contains the complementary boundary edges, and so there is a cycle in the smashing of these matchings of M and N. This is a contradiction, as we assumed S_γ had no cycles.

Now we know that there must be some cycles in S_γ, and we want to show that there are exactly k cycles. We argue that interior edges in S_γ only give rise to splitting options when they are part of cycles. We know there are no interior edges in M_{min} or N_{min} by Lemma 3.5. Depending on the quiver in question, we may get an interior edge between l and $l+1$ in $M_{min} \# N_{min}$ based on the tail ending of M_{min}. Then further flipping at M results in an odd number of interior edges: in the alternating case, flipping M at g but not N results in a cycle. In the all-right case, flipping from the end of the tail at M but using N_{min} only results in one interior edge in S_γ, and we know this edge has to be from M since we must flip from the end of the tail for M or N.

Therefore, interior edges only give rise to splitting options of S_γ when they are paired, but not doubled. Since M and N have the same matching for graph indices $2 < g < l$, and S_{min} has no cycles including indices 2 and l, pairs of interior edges at any indices $2 \leq g <\leq l$ must result in creating a cycle for each pair. Then 2^k ways to split S_γ implies that this matching of S must contain exactly k cycles. $\square$

3.5 Previous Work

Our goal is to build on the work of Thao Tran, as well as restating Theorem 4.1 of Kenyon-Pemantle's for taut dimer configurations to get the following statement:

Proposition 3.7. *For any acylic (without oriented cycle) D_n-quiver, there exists a graph made up of squares and one hexagon such that mixed dimer/double dimer configurations satisfying certain connectivity of 1-valent (outside) vertices have F-polynomial as $\sum_D y(D)2^{\#\text{ cycle components}}$.*

Our results prove this statement (see Lemma 3.6 and Theorem 3.19).

3.6 Building a Table

We want to know all possible dimer configurations for any F-polynomial for this D_n "all right" quiver.

Figure 5: D_n quiver with all arrows pointing right

By looking at the quiver, we see that if we want to mutate in a source-inducing sequence, we have to start by mutating at vertex $n-1$. Performing this mutation results in a source at $n-2$, and we continue mutating in order of decreasing vertex index (where the order we mutate at 0 and 1 doesn't matter as long as we are consistent with each revolution).

This source-inducing mutation sequence produces the following F-polynomials:

Batch \ Vertex	$n-1$	$n-2$	$n-3$
1	$y_{n-1}+1$	$y_{n-2}*y_{n-1}+y_{n-1}+1$	$y_{n-3}*y_{n-2}*y_{n-1}$ $+y_{n-2}*y_{n-1}+y_{n-1}+1$
2	$y_{n-2}+1$	$y_{n-3}*y_{n-2}+y_{n-2}+1$	
3	$y_{n-3}+1$	$y_{n-4}*y_{n-3}+y_{n-3}+1$	

Figure 6: D_n The entries for F-polynomials for part of our table

Lemma 3.8. *Given an acyclic D_n quiver in the "all right" orientation, source-inducing mutation requires we mutate first at vertex $n-1$ and mutate at each vertex consecutively down the tail (and where we can mutate at either 0 or 1 first, as long as we use the same convention for each revolution).*

Proof. We proceed by induction. For the D_4 "all-right" case, we see that vertex 3 is the only initial source. After mutating at 3, vertex 2 becomes a source. After mutating at 2, vertices 1 and 0 both become sources, so we can pick which one we would like to mutate at first.

Now, we assume that for D_k, $k \geq 4$, a source inducing sequence requires us to begin mutation and flip along the A-type tail at decreasing vertices until we mutate at vertex 2.

Consider D_{k+1}, where we add a vertex k with arrow pointing towards vertex k-1. Since k-1 now has one arrow pointing away from it, but one pointing towards it, k-1 is no longer a source. From our inductive hypothesis, we know that in the D_k case, k-1 was our only initial source, so we can't mutate anywhere else in the quiver. But vertex k is a source- when we mutate at k, we get that $k-1$ is a source. Then by our inductive hypothesis, we are forced to continue to mutate down the tail until we get to vertex 2. □

Lemma 3.9. *Let F_j^n denote the F-polynomial of vertex j for a quiver that results from performing a source-inducing mutation sequence n times at vertex j. For the all-right case, mutating in a source-inducing sequence yields the relation*

$$F_j^n \cdot F_j^{n+1} = F_{j+1}^{n+1} \cdot F_{j-1}^n + Y,$$

where Y is a monomial of the shadow cluster variables, for vertices j labeled n-1 through 3.

Proof. By Lemma 3.8, we see that we must mutate beginning at $n-1$ and mutating down the tail. Note that, for our initial revolution, all shadow vertices i' have one arrow pointing towards vertex i, and so there is a path of arrows from the vertex i' to vertex $i-1$.

When we begin by mutating at $n-1$, this means we must add an edge from $n-1'$ to $n-2$. Now $n-2$ is a source with arrows pointing away from it to $n-1$ and $n-3$, and one arrow pointing towards it from $n-1'$. By our cluster variable relation, we see that we add the product of arrows pointing to $n-1$ with arrows pointing away from $n-1$. After mutating at $n-2$, we get that we have the arrow from $n-2'$ pointing towards $n-1$, and an arrow pointing from $n-1$ to $n-2$.

When we mutate the rest of the quiver after $n-2$, we are ensured that we can't get another vertex i' pointing towards $n-1$. This is a result of the arrow from $n-1$ to $n-2$. It ensures there can be no paths from some other vertex to vertex $n-1$. Similar to what happened to vertex $n-1$, we get that at the end of this revolution, a vertex j (where $2 \leq j \leq n-1$) has only one shadow vertex pointing to it : vertex $j-1'$.

When we repeat this process for the second round of mutation, we decrease by one the index of the shadow variables pointing at vertex j. After k revolutions are complete, we get that $j-k'$ has an arrow pointing to vertex j.

Base Case:

For D_6 all right For the first revolution:

$F_5 * F_5^1 = F_4 + y_5$

$F_4 * F_4^1 = F_5^1 * F_3 + y_4 * y_5$

$F_3 * F_3^1 = F_4^1 * F_2 + y_3 * y_4 * y_5$

For the second revolution:

$F_5^1 * F_5^2 = F_4^1 + y_4$

$F_4^1 * F_4^2 = F_5^2 * F_3^1 + y_3 * y_4$

$F_3^1 * F_3^2 = F_4^2 * F_2^1 + y_2 * y_3 * y_4$

We see that $F_j^n \cdot F_j^{n+1} = F_{j+1}^{n+1} \cdot F_{j-1}^n + Y$ in cases in the middle of the tail. We get a slight modification for vertex n-1, which we can clear up by noting that we can imagine $F_n^k = 1$, where $0 \leq k \leq n-1$.

We also observe that the Y-monomial in each expression $F_j^k \cdot F_j^{k+1}$ corresponds to the shadow vertices pointing towards vertex j after we've mutated at vertex j+1 but before we mutate at j in the middle of the kth revolution.

Inductive Hypothesis: Assume that $F_j^k \cdot F_j^{k+1} = F_{j+1}^{k+1} \cdot F_{j-1}^k + y_{j-k+1} \cdots y_{n-k}$

where we note that $y_{j-k+1} \cdots y_{n-k} = \prod_{i=j-k+1}^{k} y_{n-i}$

for $j, k \geq 1$ where j denotes the vertex and k denotes the number of mutations performed on that vertex.

Inductive Step: Assume we know the following F-polynomials: F_j^{k+1}, F_{j-1}^k, and F_{j-2}^k. We want to find F_{j-1}^{k+1}. We need to show: $F_{j-1}^k \cdot F_{j-1}^{k+1} = F_j^{k+1} \cdot F_{j-2}^k + y_{j-k} \cdots y_{n-k-1}$.

Suppose we are partway through our *kth* revolution, and have this quiver, as in Figure 7:

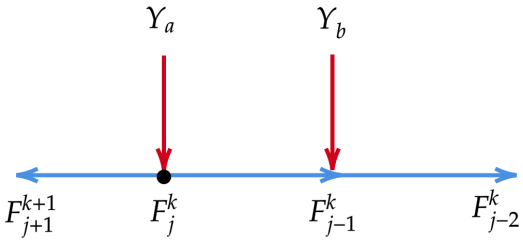


Figure 7:

We let Y_A represent the edges from the shadow vertices incident to F^{k+1}_j, and we let Y_B represent the edges that point to F^k_{j-1} from shadow vertices. Then we can mutate at vertex F^k_j to obtain F^{k+1}_j, as shown in Figure 8.

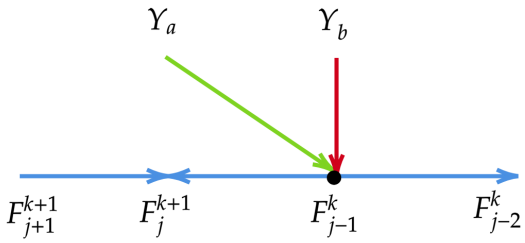


Figure 8:

We note now that mutation at vertex F^k_j means that all arrows from shadow vertices Y_A are now pointing towards F^k_{j-1}. However, we need to better define Y_A and Y_B. We want to make sure that we only count each arrow pointing to F^k_{j-1} once.

To check for overlap, we use our inductive hypothesis. Our Y-term in the recurrence always denotes the shadow vertices pointing into a vertex in the midst of that revolution. We see that F^k_j has a Y-term $y_{j-k}\cdots y_{n-k-1}$ which represents shadow edges emanating from vertices $\{j-k', j-k+1,\ldots,n-k-1\}$ incident to it during the kth revolution. F^k_{j-1} has Y-term $y_{j-k}\cdots y_{n-k}$, which represents arrows from shadow vertices $\{j-k', j-k+1,\ldots,n-k\}$ incident to it during the kth revolution.

After the kth revolution is complete, each vertex j has one shadow edge incident to it, whose arrow points from the shadow vertex j-k'. So since we are in the middle of the k+1 revolution, we get that $Y_A = \rightarrow\{$ j-k', j-k+1,...,n-k-1$\}$, where this set denotes the shadow vertices with arrows emanating from them. We get $Y_B = \rightarrow \{$n-k'$\}$, where this denotes the arrow emanating from vertex n-k', (since we haven't yet mutated at F^k_{j-1}, and after the kth revolution we are left with this lone arrow).

After we mutate at F^k_j, vertex F^k_{j-1} becomes a source and has edges emanating from it pointing towards F^{k+1}_j and F^k_{j-2}. We get arrows incident to it from shadow vertices equal to the set $Y_A + Y_B$

Therefore, $F^k_{j-1}\cdot F^{k+1}_{j-1} = F^{k+1}_j \cdot F^k_{j-2} + y_{j-k}\cdots y_{n-k-1}$.

□

Lemma 3.10. *For the all-right case, mutating in a source-inducing sequence yields: For k even,* $1 \le k < n-1$*:*

$F^{k-1}_0 \cdot F^k_0 = F^k_2 + y_1 * y_2 \cdots y_{n-k}$

For k odd:

$F^{k-1}_1 \cdot F^k_1 = F^k_2 + y_0 * y_2 \cdots y_{n-k}$

where Y is a monomial of the shadow cluster variables and for vertices j labeled 2, 1, and 0, and where $0 < k \le n-1$.

Proof. **Case 2:** Vertices 0, 1, 2:

Base Case:

For D_6 all right For the first revolution:

$F_2 \cdot F_2^1 = F_3^1 \cdot F_1 \cdot F_0 + y_2 \cdot y_3 \cdot y_4 \cdot y_5$

$F_1 \cdot F_1^1 = F_2^1 + y_1 \cdot y_2 \cdot y_3 \cdot y_4 \cdot y_5$

$F_0 \cdot F_0^1 = F_2^1 + y_0 \cdot y_2 \cdot y_3 \cdot y_4 \cdot y_5$

For the second revolution:

$F_2^1 \cdot F_2^2 = F_3^2 \cdot F_1^1 \cdot F_0^1 + y_0 \cdot y_1 \cdot y_2^2 \cdot y_3^2 \cdot y_4^2 \cdot y_5$

$F_1^1 \cdot F_1^2 = F_2^2 + y_0 \cdot y_2 \cdot y_3 \cdot y_4$

$F_0^1 \cdot F_0^2 = F_2^2 + y_1 \cdot y_2 \cdot y_3 \cdot y_4$

Inductive Hypothesis:

For k even, $1 \leq k < n-1$:

$F_0^{k-1} \cdot F_0^k = F_2^k + y_1 \cdot y_2 \cdots y_{n-k}$

For k odd:

$F_1^{k-1} \cdot F_1^k = F_2^k + y_0 \cdot y_2 \cdots y_{n-k}$

Inductive Step: Without loss of generality, we will prove this statement for k odd and assume that the proof follows for k even. By our inductive hypothesis, we have $F_1^{k-1} \cdot F_1^k = F_2^k + y_1 \cdot y_2 \cdots y_{n-k}$. Now suppose we are in the middle of the $k+1$ revolution: we have just mutated at vertex 2, but not yet mutated at vertex 1. We see that after mutation k, there was the set Y_B of shadow vertices with indices 1',...,n-k' with their arrows pointing towards vertex 1, and similarly we get the set Y_A of shadow vertices with indices 0',1',$2'^2$,...,$n-k-1'^2, n-k'$ with their arrows pointing towards vertex 2.

Now when we mutate at 2 in the middle of the k+1 revolution, we get that Y_A now has arrows incident to vertex 1, where some of these arrows cancel out with Y_B: now the only shadow vertices with arrows pointing to vertex 1 are 0',2',...,n-k-1'. Since vertex 1 is now a source, we also get an arrow pointing from 1 to 2. We then get: $F_1^k \cdot F_1^{k+1} = F_2^{k+1} + y_0 \cdot y_2 \cdots y_{n-k-1}$, as we hoped. □

Lemma 3.11. *For a single dimer N with tail length j (denoted N^j), the only possible flip sequences are $(j, j-1, \ldots, k)$ where $0 \leq k \leq j$ and $(j, j-1, \ldots, 3, 2, 0)$.*

Proof. We will use induction. For $j = 3$, the only first flip we can do is at 3. Once we flip at 3, we can only flip at 2. After flipping at 2, we can choose to flip at 0 or 1 or both. This yields the possible sequences $(3, 2, 1, 0), (3, 2, 1)$ and $(3, 2, 0)$ as desired.

Now, suppose that our statement is true for N_{j-1}, and we want to consider what happens when we add a square j to the end of the tail. The only first flip we can do is at j. Once we flip at j, we cannot flip at j again and the square $j-1$ obtains a parallel edge. So, the remaining of our flip sequence is completely determined by the sub N_{j-1} (our N_j excluding square j). By inductive hypothesis, we know that the possible flip sequences generated by the sub N_{j-1} are $(j-1, j-2, \ldots, k)$ where $0 \leq k \leq j-1$ and $(j-1, j-2, \ldots, 3, 2, 0)$. Adding the first flip to these sequences, we obtain the statement we desire. □

Lemma 3.12. *For $2 \leq i \leq n-1$, the F-polynomial of i after one revolution is $1 + y_{n-1} + y_{n-1}y_{n-2} + \ldots + y_{n-1}y_{n-2}\ldots y_i$. These terms are represented by a zig-zag pattern of squares (and potentially a hexagon) labeled $n-1, n-2, \ldots, i$, with minimal matching as follows in Figure 9:*

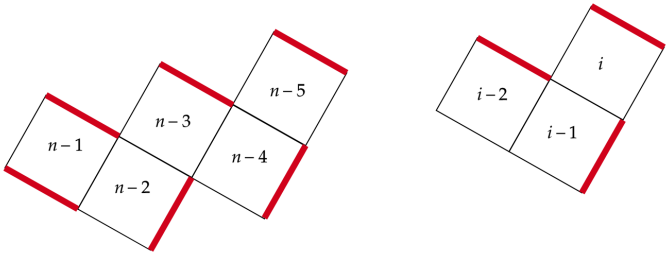


Figure 9:

Proof. Trying to prove the first row of the table inductively:

Base Case: For D_4 all right:

F_3^1 is $y_3 + 1$

F_2^1 is $y_2 \cdot y_3 + y_3 + 1$

F_1^1 is $y_1 \cdot y_2 \cdot y_3 + y_2 \cdot y_3 + y_3 + 1$

F_0^1 is $y_0 \cdot y_2 \cdot y_3 + y_2 \cdot y_3 + y_3 + 1$

F_3^1:We can choose either orientation for the y_3 square, but it will suit us later to choose the horizontal edges. With either choice, we get a possible flip which corresponds to our accounting factor y_3.

F_2^1: We need two squares, and since y_3 appears as a monomial in this expression, we need to be able to flip at y_3 before we can flip at y_2. Then we know we need the matching of this square and hexagon where we have horizontal parallel edges on y_3, and we connected the y_3 square according to our convention, since the arrow points from 3 to 2 in the quiver.

F_1^1: Now we need to build onto the squares we already have. We attach square y_1 in a zigzag according to convention in the quiver. The only labeling of these three squares is forced by the fact that we flip at y_3 first.

F_0^1: We follow the same process as for F_1^1 above but with square y_0 instead of y_1.

Inductive Hypothesis: Assume that for the D_{n-1} all-right quiver, the first row of the table looks like this:

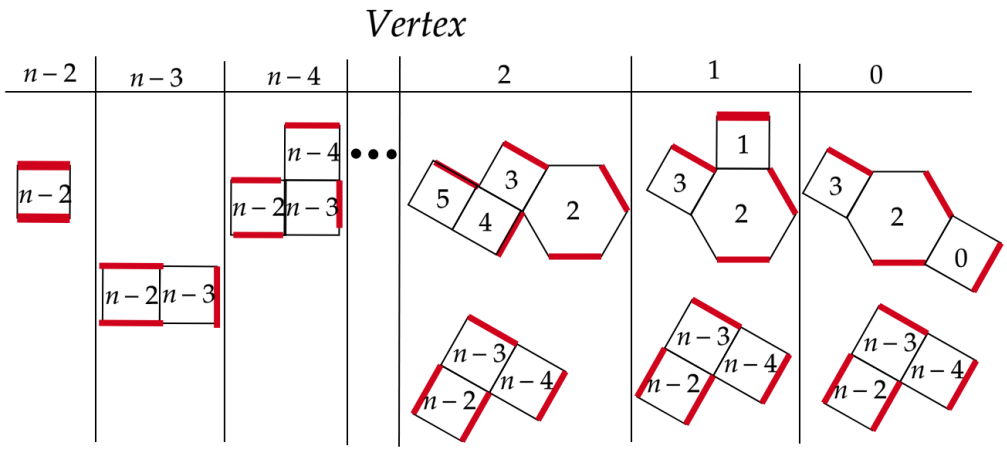


Figure 10: The first row of the D_{n-2} all-right table

Inductive Step: Suppose we add an arrow with label $n-1$ pointing towards vertex $n-2$ in our quiver. Then $n-2$ is no longer a source, but $n-1$ is. Since we always mutate in a source-inducing sequence, this orientation of the quiver forces us to mutate at vertex $n-1$ first. Then in our dimer model, we need to be able to flip at square y_{n-1} first, before we are able to flip at y_{n-2}, by Lemma 3.11. This means our first entry of the first row corresponds to a square y_{n-1}. Then we add the y_{n-1} square to the existing entries of the first row, keeping in mind that it must have parallel edges so we can flip at it first.

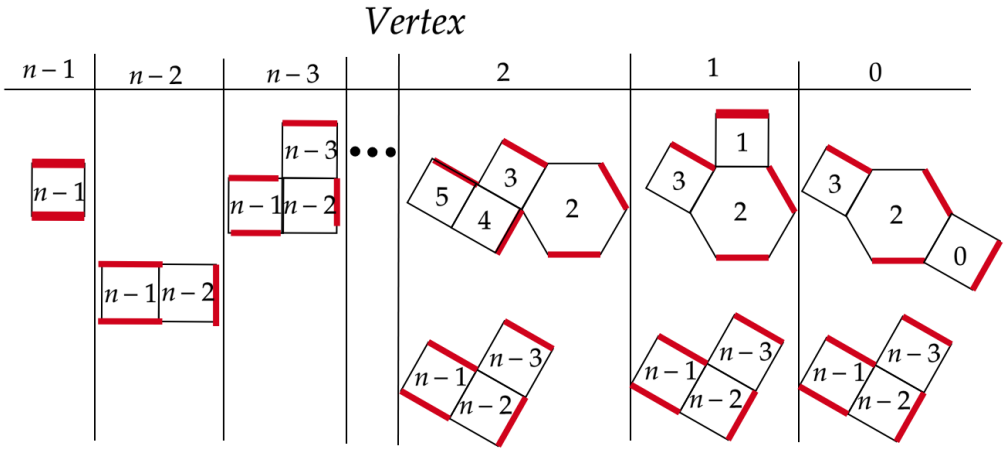


Figure 11: The first row of the D_{n-1} all-right table

□

Corollary 2. *The F-polynomial of $n-1$ after i revolutions ($1 \leq i \leq n-2$) is $y_{n-i}+1$. They are represented by square $n-i$.*

Proof. Consider the D_n all-right quiver. To obtain our first batch, we began source-inducing mutation at vertex $n-1$. By Lemma 3.8, we begin each revolution by flipping at vertex n-1. After one revolution, we get the same orientation of the quiver we had initially, but now we have that $n-2'$ points to $n-1$ instead of $n-1'$. We follow the same process from the first revolution in the second.

Then after the *kth* revolution, we see that we are left with an arrow from $n-k'$ pointing to vertex $n-1$. When we mutate for the $k+1$ revolution, we get arrow $n-k-1'$ pointing at vertex $n-1$. Every time we repeat this source inducing mutation sequence, we get the same behavior; the arrow pointing to the $n-1$ vertex comes from a shadow vertex whose index decreases by 1. Then the F-polynomial of $n-1$ after k revolutions ($1 \leq i \leq n-2$) is $y_{n-k}+1$. □

Corollary 3. *The kth row of the table looks like the last n-k entries of the first row where we delete any squares with labels greater than n-k.*

Proof. When we use a source-inducing mutation sequence on the quiver, we know that after the kth revolution, we are left with only the $j-k'$ arrow pointing to vertex j. Then the entries of row k that are square-free and without the 0 or 1 vertices can be found by calculating the F-polynomials where we are left with n-k as the smallest indexed label, but we build up inductively across the row in the same way we did for the first row. □

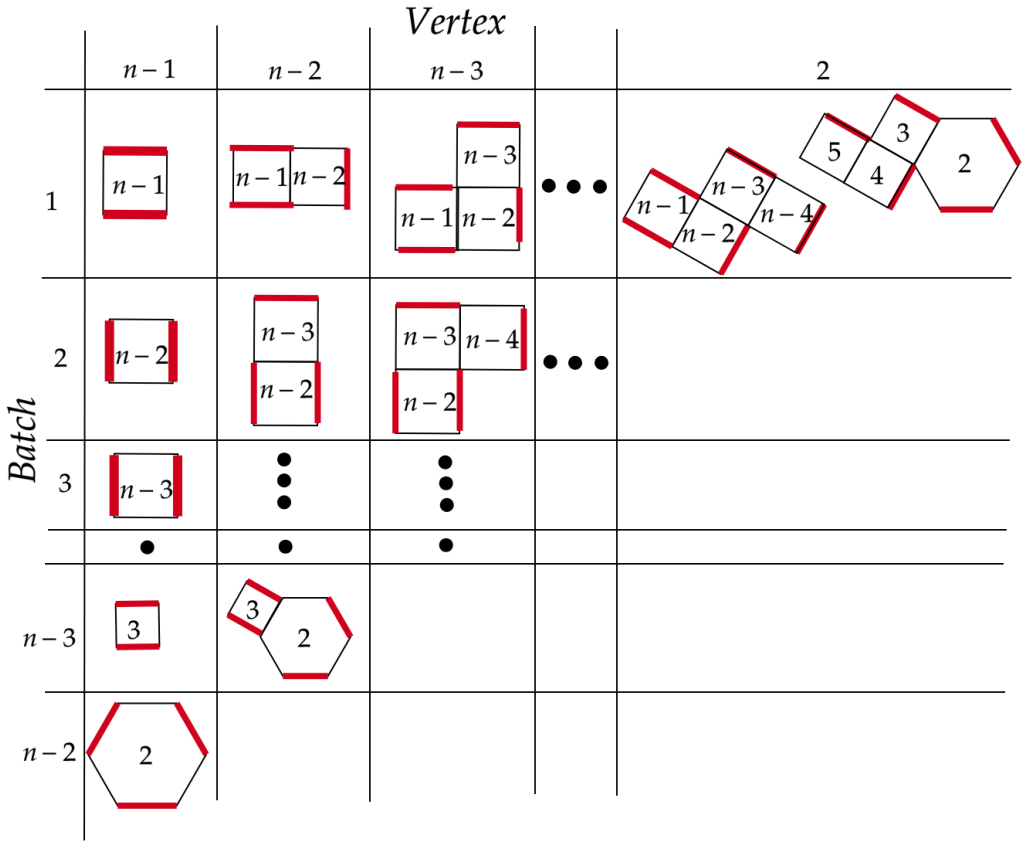


Figure 12:

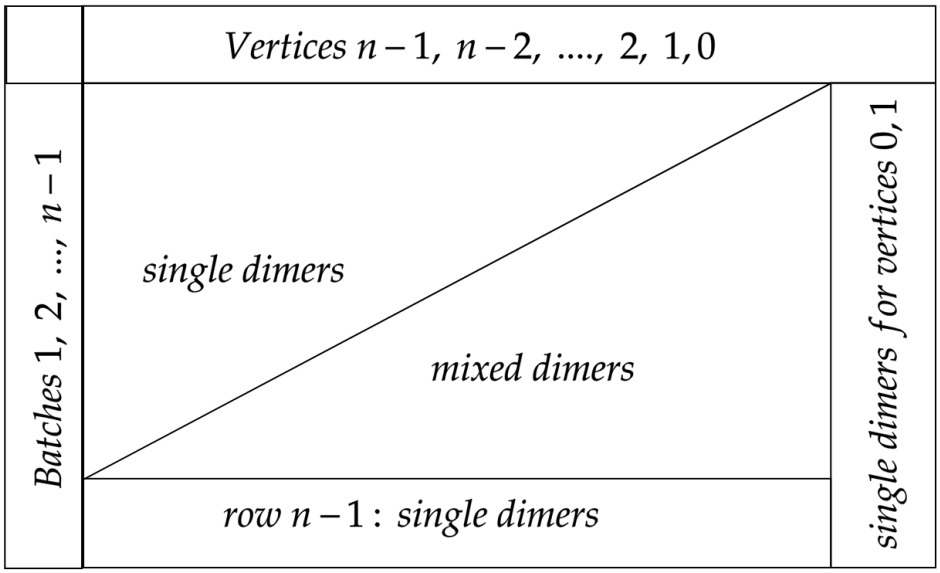


Figure 13:

Now that we have defined N and M, we want to be sure that single dimers matching their descriptions actually exist, and that we can define their associated F-polynomials. Let N^j refer to the minimal matching of the dimer configuration in the $n-1$ row of the table (for j>=2), and let M^l refer to the lth minimal matching (for $l \geq 2$) of the dimer configuration in the diagonal whose

first entry is F_{n-2}^{n-1}, where F_{n-2}^{n-1} corresponds to M^2. Let $N_j, ...k^j$ refer to the dimer configuration obtained by flipping N^j through $j, ..., k$ in the dimer matching, flipping in order starting with square j.

Now, we require j$\geq l+1$. Then smashing N^j and M^l, denoted $N^j \# M^l$, results in a dimer with doubled edges for the first lth entries of the tail, and single edges for all entries k such that $l < k \leq j$. Note that square l has a double dimer covering due to the interior edge at l, which is a result of M^l having a length of l.

We let $N^j \# M^l = D_{min}$, where D is a mixed dimer in our diamond relation in our table as labeled below in Figure 14, since we get a dimer configuration that has length j with l as the highest indexed double dimer in the tail. Through a similar process, we can explicitly break down A, B, and C into their M and N single dimer smashing pairs, so we get:

$$\begin{aligned} A_{min} &= M^{l+1} \# N^{j+1} \\ B_{min} &= M^{l+1} \# N^{j} \\ C_{min} &= M^{l} \# N^{j+1} \\ D_{min} &= N^{j} \# M^{l} \end{aligned}$$

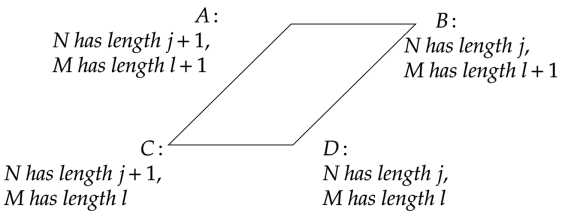


Figure 14: The diamond relation in our table in terms of M and N

Note here that if D has a tail of length j, and l is its highest-indexed squared term, then A has a length $j+1$ with $l+1$ as its maximum squared term, B has length j with $l+1$ as its maximum squared term, and C has length $j+1$ with l as its maximum squared term.

If we focus on the F-polynomial aspect of the recurrence, we see that if $Y_A * Y_D$ has a term $h * Y_k$ in its product F-polynomial, (for $h \geq 1$ a coefficient), then Y_k must also appear as a term in $Y_B * Y_C$, with h as its coefficient. We will address the term $Y_{D_{max}}$ at the end of this section, and show that $Y_{D_{max}}$ does not cancel out in our F-polynomial equation and corresponds to a dimer smashing of B and C that violates connectivity rules.

Lemma 3.13. *Consider the D_n quiver with all arrows pointing to the right. Let α and β refer to possible source-inducing flip-sequences for two single dimers, M_α^l and N_β^j respectively, where M_α^l has length l and N_β^j has length j, and $j \geq l+1$. The smashings $M_\alpha^l \# N_\beta^j$ violate connectivity only if $N_\beta^j = N_{max}^j$; when $\beta = \{j, ..., 1, 0\} = max$, for $\alpha \neq max$.*

Proof. Consider N_β^j and M_α^l. In the product $M_\alpha^l \# N_\beta^j$, we note that M doesn't have squares labeled 0,1. Since our connectivity rules are only related to the squares 0 and 1, we can consider cases of N for which these squares might change.

For $N_{j,...,k}^j$, where $k \geq 2$, we have connectivity preserved due to a single matched edge on the outermost side of the square parallel to the hexagon for both squares 0 and 1. See Figure 15.

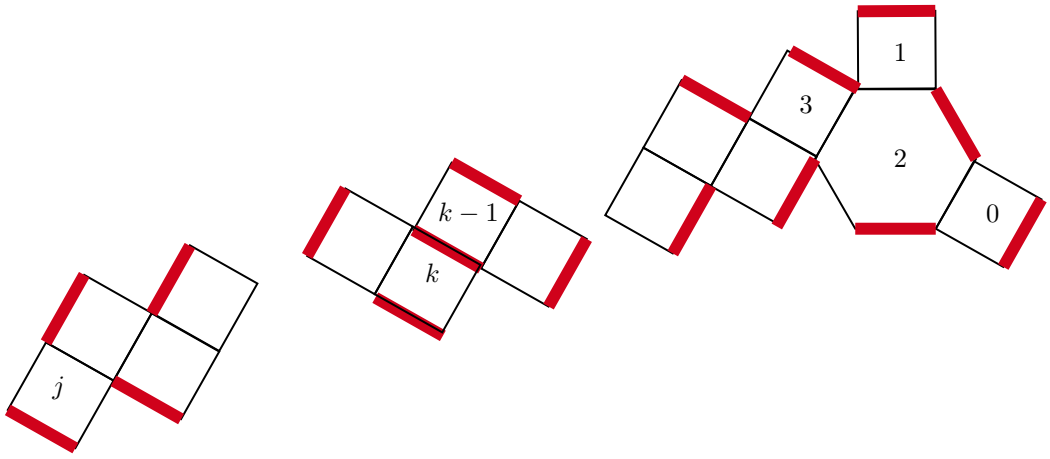


Figure 15: $N^j_{j,\ldots,k}$

Suppose we flip now at 0 to get $N^j_{j,\ldots,2,0}$. Then the single matched edge is converted to parallel edges along the sides of square 0. Consider M^l_α. For M^l_{max}, we get a matched edge on the hexagon at square 0, so connectivity is preserved. For M^l_{min}, we get a path down the tail of our smashed dimer, snaking to square l and then back again, so connectivity is preserved. Finally, for $M^l_{l,\ldots,k}$, for $2 < k < l$, our smashed dimer has a path that snakes to include square $k-1$, so connectivity is preserved. We can follow the same idea of proof for $N^j_{j,\ldots,2,1}$.

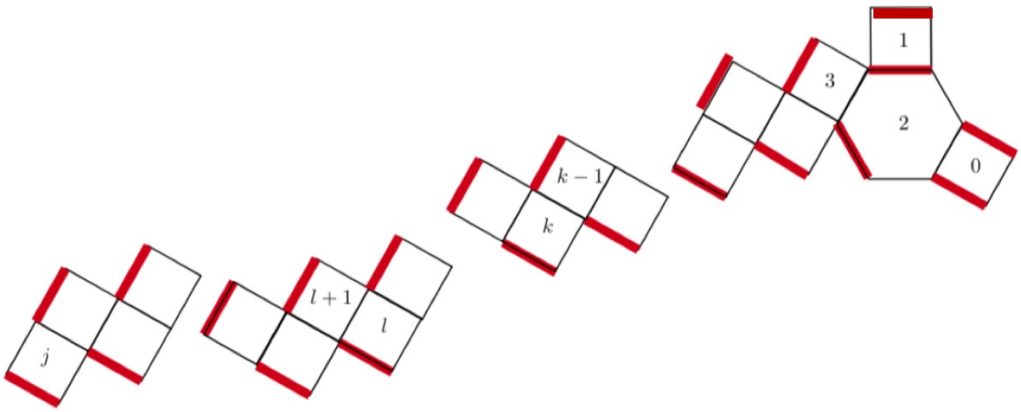


Figure 16: $N^j_{j,\ldots,2,1,0}$

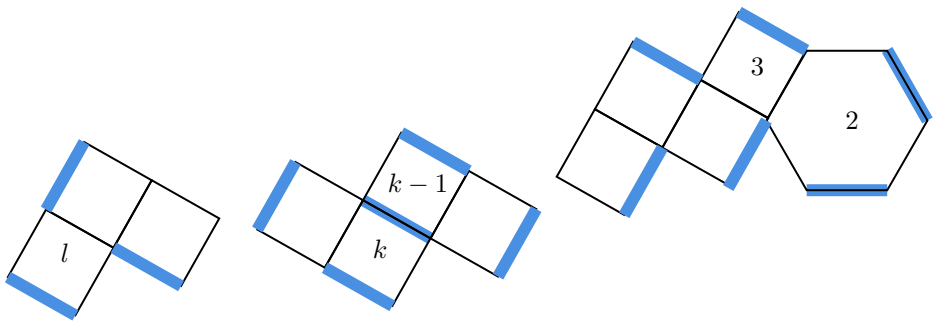


Figure 17: $M^l_{l,\ldots,k}$

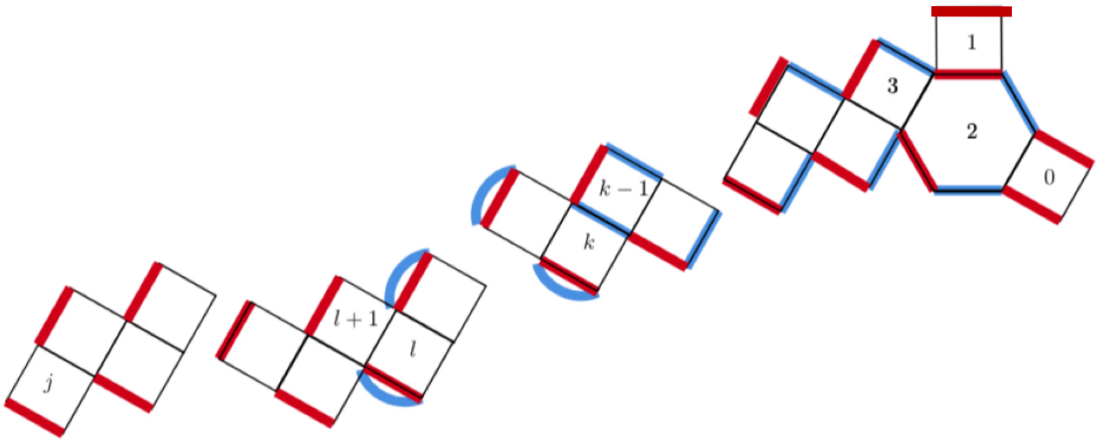


Figure 18: $M^l_{l,\ldots,k} \# N^j_{j,\ldots,2,0}$

Now consider $N^j_{j,\ldots,2,1,0}{=}N^j_{max}$. We have parallel edges along the sides of square 0, and along the side of square 1. The only way we can connect these edges is to smash with M^l_{max}. Smashing with $M^l_{l,\ldots,k}$, for $2 < k < l$, results in a path down the tail, but this path only connects the vertices of 0 and 1 "closest" to the tail. Therefore the only smashing for which connectivity fails can occur for N^j_{max}, and some M^l_α, where $\alpha \neq max$.

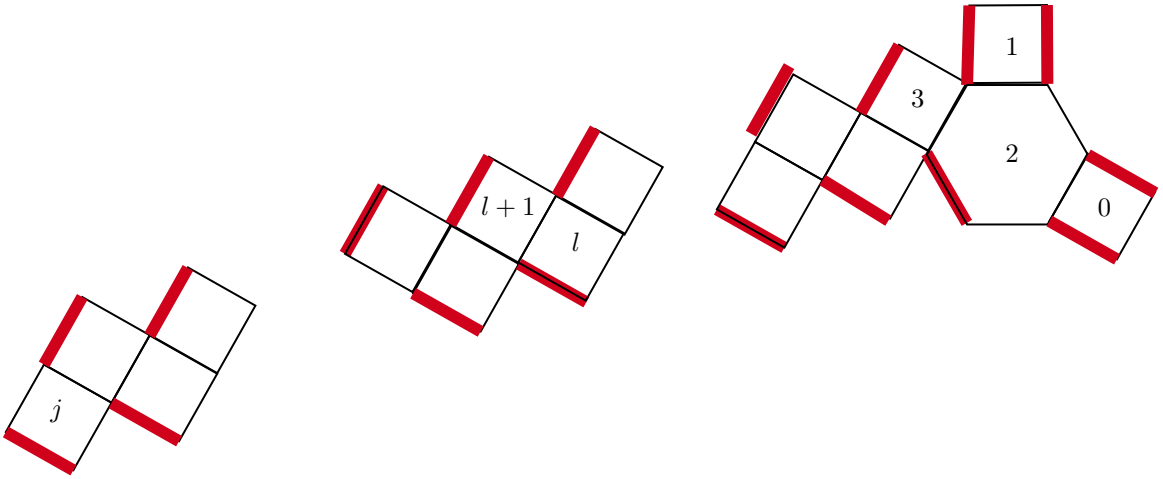


Figure 19: N^j_{max}

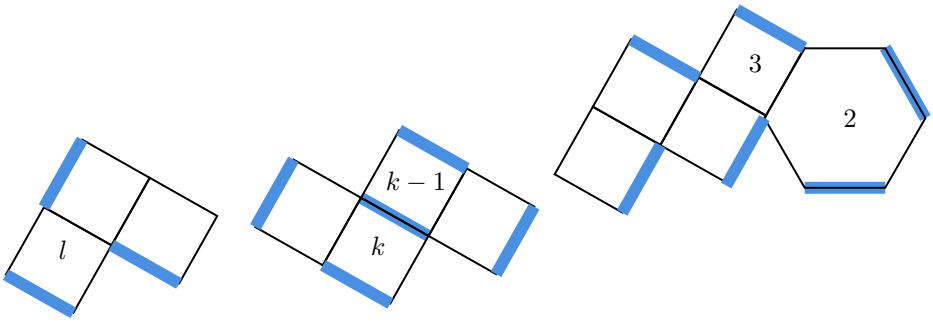


Figure 20: $M^l_{l,\ldots,k}$

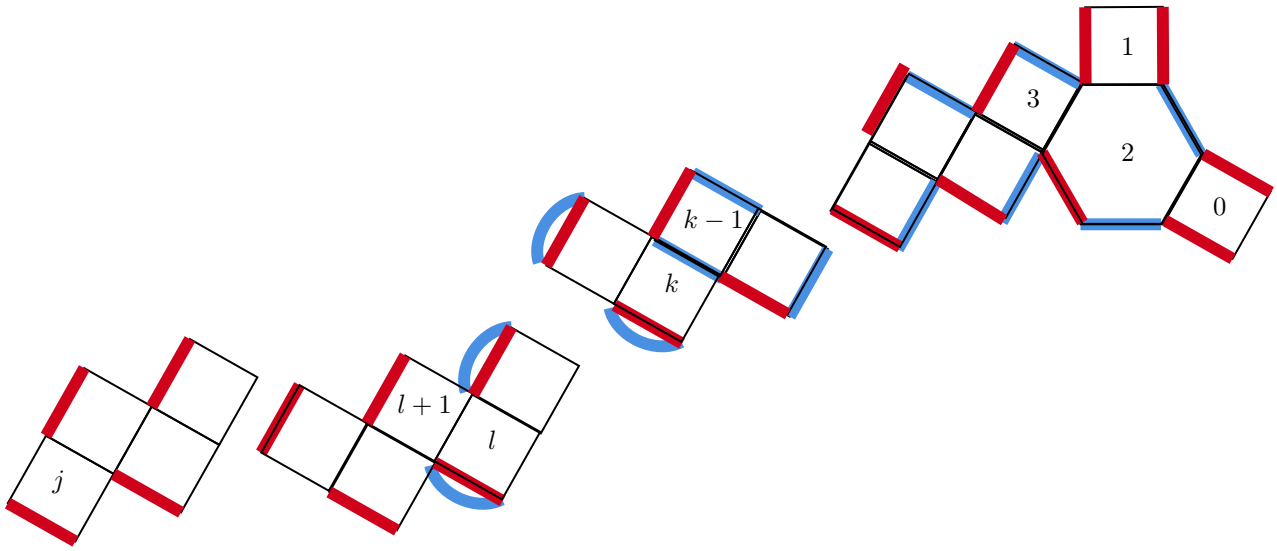


Figure 21: $M^l_{l,\dots,k}\#N^j_{max}$

□

Lemma 3.14. *The number of perfect matchings of N^j is $j+3$. Similarly, the number of perfect matchings of M^l is l.*

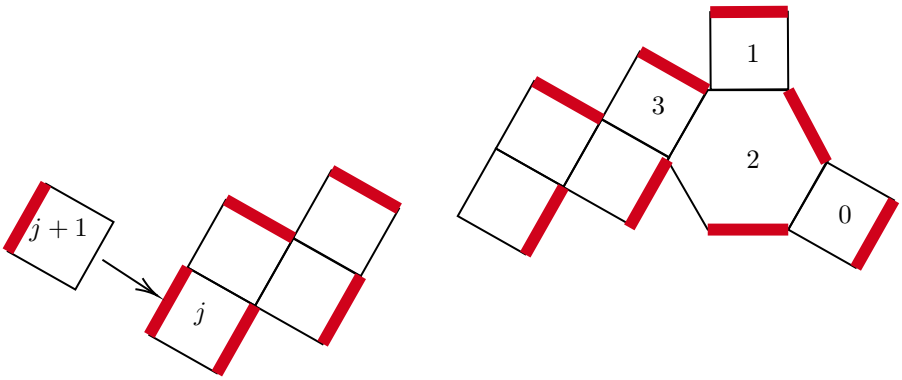


Figure 22: p_j with $j+1$ added

Proof. Let p_j be the number of perfect matchings of N_j. Now, we would like to find a recurrence relation of p_j. Consider N_{j+1} and in particular its square $j+1$ at the end of its tail. The two outer vertices of this square can be matched in two ways: one by a direct edge between them, and another by two parallel edges connecting them to the remaining vertices of the square. The earlier way tells us that all the remaining N_j part of this N_{j+1} have to be perfectly matched, meaning there are p_{j+1} perfect matchings in this case. The latter way forces us to match the remaining N_j part of this N_{j+1} in exactly one way. In the end, $p_{j+1} = p_j + 1$. We know the base case that $p_2 = 5$, so we get that the number of perfect matchings of N_j is $j+3$.

Using the same proof idea, but keeping in mind that M^l does not contain squares 0 or 1, the number of perfect matchings of M^l is l. □

Lemma 3.15. *(Considering dimers in our lower triangle for D_n all right): The number of terms in the product F-polynomial of two double dimers S and T is*

$$(M_S * N_S - \textit{invalid smashings of } M^{l_S}_\alpha \# N^{j_S}_\beta) * (M_T * N_T - \textit{invalid smashings of } M^{l_T}_\delta \# N^{j_T}_\gamma) \quad (1)$$

where M_S refers to the number of perfect matchings of $M^{l_S}_\alpha$, for S with l_S as its highest indexed box with a double dimer covering. N_S refers to the number of perfect matchings of $N^{j_S}_\beta$, where S

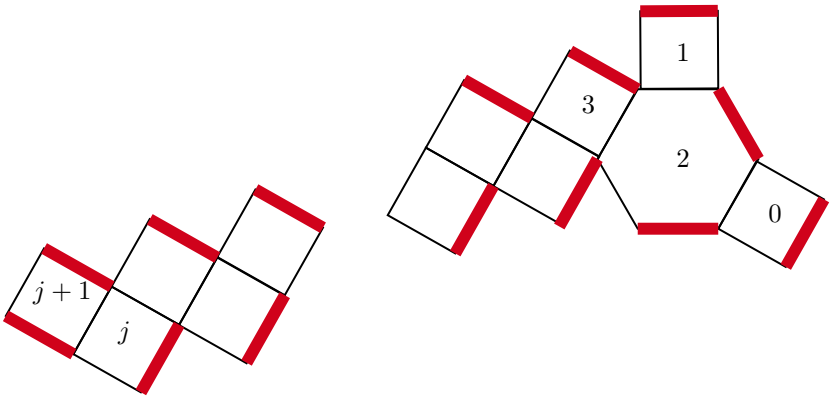


Figure 23: p_{j+1} with forced matching due to parallel edges on $j+1$

has j_S as its highest indexed term at the end of its A-type tail. Similarly, M_T and N_T refer to the number of possible perfect matchings of $M_\delta^{l_T}$ and $N_\gamma^{j_T}$ respectively, according to double dimer T with with l_T as its highest indexed box with a double dimer covering and j_T as its highest indexed term at the end of its A-type tail.

Proof. By our lemma on invalid connectivity, the only N single dimers that lead to invalid smashings with M single dimers is the matching of N_{max}. As long as we guarantee that our matchings of S and T both respect connectivity, we are guaranteed that $S\#T$ will as well, since smashing preserves all the original edges of S and T.

The expression $M_S * N_S -$ invalid smashings of $M_\alpha^{l_S}\#N_\beta^{j_S}$ gives the number of valid matchings for S, and the expression $M_T * N_T -$ invalid smashings of $M_\delta^{l_T}\#N_\gamma^{j_T}$ gives the number of valid matchings for T. Then smashing these valid matchings corresponds exactly to multiplying the F-polynomials for S and T.

Then we know that $j_S \geq l_S + 1$, and $j_T \geq l_T + 1$ for our double dimers (in this lower triangle of the table). By our lemma, we know $M_{l,\ldots,k}^l\#N_{max}^j$ violates connectivity unless $M_{l,\ldots,k}^l = M_{max}^l$. Since M_l has l perfect matchings by our lemma above, we get that there are $l_S - 1$ matchings that violate connectivity for S. Similarly, we get $l_T - 1$ matchings that violate connectivity for T.

Our product of F-polynomials for S#T includes only dimers that do not violate connectivity, since the F-polynomials of S and T have terms that correspond to valid double dimers only. By our above lemma, $M_S = l_S$, $N_S = j_S + 3$. Similarly, $M_T = l_T$, $N_T = j_T + 3$. So we know exactly how many terms the product F-polynomial $Y_{S\#T}$ has:

$$\begin{aligned}\text{number of terms of } Y_{S\#T} &= [l_S(j_S+3) - (l_S-1)] * [l_T(j_T+3) - (l_T-1)] \\ &= l_S j_S l_T j_T + 2l_S j_S l_T + 2l_S l_T j_T + l_S j_S + 4l_S l_T + l_T j_T + 2l_S + 2l_T + 1 \quad (2)\end{aligned}$$

□

Lemma 3.16. *AD has one extra matching that cannot be split into B and C. This matching is the one that corresponds to $A_{min}\#D_{max}$.*

Proof. By our above theorem, substituting $l_A = l+1$, $l_D = l$, $l_B = l+1$ and $l_C = l$, and substitute $j_A = j+1$, $j_D = j$, $j_B = j$ and $j_C = j+1$. Use a base case of: l=2, j=4. This is the smallest case we can get since A has its largest squared term as l+1, but has tail of length j, and we do not allow the end of the tail to be a doubled square.

Then by our above lemma: Number of matchings: A*D= ((l+1)*(j+4)-l)*((l)*(j+3)-l+1)

B*C= ((l+1)*(j+3)-l)*((l)*(j+4)-l+1)

Then AD-BC=1. There is one matching contained in AD that cannot be split into BC. This is the matching $A_{min}\#D_{max}$: this is a valid smashing in AD, but cannot be split into BC:

Note that $A_{min} = M_{min}^{l+1}\#N_{min}^{j+1}$ and note that $D_{max} = M_{max}^l\#N_{max}^j$. However, we get

$C_{split} = M_{max}^l\#N_{min}^{j+1}$ and $B_{split} = M_{min}^{l+1}\#N_{max}^j$. This is the only split possible, since C has a length of j+1 with l squared terms, and B has a length of j with l+1 squared terms. But we have a problem here: we have smashed $B_{split} = M_{min}^{l+1}\#N_{max}^j$. Since we smashed N_{max}^j with a

matching of M^l that is not M^l_{max}, by our lemma on connectivity, B_{split} violates connectivity. Thus $A_{min}\#D_{max}$ is the smashing that cannot be split into B and C.

Theorem 3.17. *The unsplittable matching $A_{min}\#D_{max}$ corresponds to the extra Y term in the F-polynomial recurrence.*

Proof. By the preceding lemma and the lemma that counts the number of terms of the product F-polynomials, we see that AD has an extra term that is not contained in BC, and we see that AD has an extra dimer matching that is not contained in BC. Since all the terms of the product F-polynomial correspond to valid dimer configurations, $A_{min}\#D_{max}$ must be the dimer configuration that produces this extra Y-term in the product F-polynomial. □

□

3.7 Given an Acyclic Quiver...

Given an acyclic D_n quiver, we can draw mixed dimer minimal matchings that correspond to all possible F-polynomials that could be generated by that quiver.

We can do this by considering subquivers and using our source-inducing mutation sequence.

Given an acyclic D_n quiver, consider a subquiver with highest indexed vertex j, where $0 < l < j \leq n-1$.

To draw N: Consider vertices $0, 1, ..., j$.

- Attach the squares together and to the hexagon based on the attachment rules
- Since sources can be flipped at first, we give the squares corresponding to sources in the quiver paired boundary edges so that we can flip at these squares first in our dimer model.
- This includes sources at the end of the tail; if the arrow incident to vertex $n-1$ is outgoing, give square $n-1$ a pair of boundary edges. If this arrow is incoming, cover the outermost boundary edge of square $n-1$. The same logic applies to vertices 1 and 0: we give them parallel edges that do not include a side of the hexagon if they are initially sources in the quiver so that this is a true minimal matching.
- Sinks in the quiver tail correspond to empty boxes in the dimer- we have to mutate/flip at both neighbors to get a source/pair of flippable edges.

To draw M to smash with N, consider the subquiver with vertices $2, ..., l.$:

- Follow the same rules as above, including behavior at the end of the tail but ignoring matchings on 0 and 1 since they are irrelevant to M.

Lemma 3.18. *If we switch the direction of each arrow in our initial quiver (but not the shadow arrows), then mutate in a source inducing sequence, we draw the same dimer configurations as for the original quiver but we invert the poset including the minimal and maximal matchings.*

Proof. Suppose we have drawn all dimer configurations in the poset for a quiver Q, and now we flip the direction of all the arrows to produce Q_{flip}. Suppose that when we draw the poset of dimer configurations of Q_{flip}, we get some dimer matching that we didn't for our Q poset. However, we have assumed that our poset for Q included all valid matchings of Q. Note that we could have produced the poset of Q_{flip} by using a sink-inducing mutation sequence on Q, which also guarantees we generate every possible F-polynomial (and hence every valid dimer matching). We have reached a contradiction; these posets must contain the same dimers and must be inversions of each other. □

Theorem 3.19. *By using the above minimal matchings for N and M and smashing them together, we can generate the largest terms of the largest F-polynomial. We can also describe the entire poset that results from flipping. This poset gives us each term of the largest F-polynomial.*

Proof. We start by considering the minimal matching of N, and suppose it has an associated F-polynomial Y_N. We can depict Y_N as a poset of perfect matchings, where the minimal value corresponds to N_{min} and the maximum value corresponds to N_{max}. We can do the same for M.

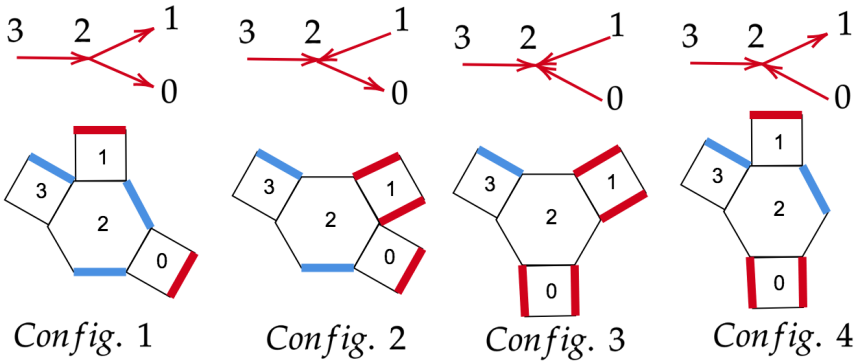


Figure 24: The possible configurations for N's minimal matching. Red edges represent labellings resulting from the quiver orientation at 1 and 0. Blue edges are forced from the red edges. Note that 3 has an (invisible) tail that is not represented here.

We can generate a lattice of smashed dimers by merging these two posets together. For every rank of the poset, we want to make sure it increases from the rank below it by only one flip, either in N or in M.

There can be redundancies in these smashings, which take the form of cycles forming. Rather than draw duplicate dimers, we assign dimers with cycles in them a coefficient (according to Lemma 3.6).

The only other problem we could run into is invalid smashings that do not preserve connectivity. The number and exact nature of these invalid smashings depends on the symmetry of the quiver around vertex 2.

We first note that smashings that violate connectivity at vertices 0 and 1 are the result of valid single dimer configurations. We must have the parallel edge "vertical" configuration to break connectivity. We note that for configurations 2 and 4, we can only get this configuration at either 1 or 0 but not both, since having both would not be a valid single dimer for N. By Lemma 3.18, we only need to consider a configuration and the configuration where we flip all of its arrows has the same poset (but upside down). See Figure 24.

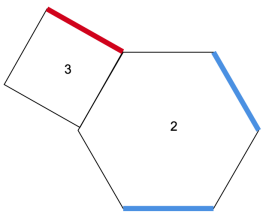


Figure 25: The possible configuration for M's minimal matching for configurations of N 1 through 4. Red edges represent labellings resulting from the quiver orientation at 3. Blue edges are forced from the red edges. Note that 3 has an (invisible) tail that is not represented here.

For configuration 1, we have proven that the exact configuration that breaks connectivity involves N_{max} and some configuration of M that is not maximal for an all-right tail case. Then we can pinpoint exactly where in the joint lattice these invalid smashings could arise: for ranks greater than or equal to j, but less than l+j+1, which is the rank that corresponds to the maximal matching. For each rank in this range, we will have a smashing of N_{max} with some non-maximal M that we will disregard for violating connectivity. Now we have successfully described the lattice for configuration 1 where the tail is in the all-right configuration.

We note that there can only be interior edges in a minimal matching of the smashed tail occurring adjacent to square l (in these cases, we can't get an interior edge on the square 3 side of the hexagon as a minimal matching where the arrow from 3 points towards 2 in the quiver, since

3 could be a source initially). Interior edges can give rise to invalid smashings in the symmetric (in 1 and 0) quiver case (configurations 1 and 3) as noted in the proof of Lemma 3.13, but not in the asymmetric case (configurations 2 and 4), since we know connectivity can only be broken at either 0 or 1 in those cases, and if we can snake through the tail to connect for 0 or 1, we have a valid smashing.

By carefully considering these behaviors in the tail and flipping at either N or M each rank in the poset, we can successfully construct the poset and be confident that it generates all terms of the F-polynomial under consideration. We find the term of the F-polynomial corresponding to each dimer in our poset by calculating the F-polynomials for M and N that we are smashing to get that dimer. We are guaranteed that this term M*N actually appears in the F-polynomial if the smash preserves connectivity at both 1 and 0.

□

Corollary 4. *By using the minimal matchings and smashing them together, we can generate the largest terms of any F-polynomial with squared terms and also describe the entire poset that results from flipping. This poset gives us each term of our chosen F-polynomial.*

Proof. We follow the same procedure as in the preceding theorem, noting that we may not be working under the condition that $j = l + 1$, but noting instead that $j \geq l + 1$. □

Corollary 5. *Special cases when we don't smash:*

- *We can consider the quiver from vertex $n-1$ to vertex 2, and draw M for any length of tail, as long as we don't smash with N.*
- *We can also draw N for any length of tail to stand alone.*
- *We can also include vertices 1 or 0 (but not both) for any length of tail for M, and then draw the single dimer for this subquiver.*

Each of these minimal matchings represents the largest term of a square-free F-polynomial, and we can describe the entire poset that results from flipping these single dimers.

Proof. These are valid single dimer configurations since they result from considering the quiver's orientation and flipping from the minimal matching when allowed. Since we are working only with single dimers and not smashing anything, we don't need to worry about invalid connectivity. Drawing the poset follows in a straightforward way once we have drawn the minimal matching.

For the minimal matching of M that includes vertices 1 or 0, we take into account the sink/source behavior at 1 or 0 to draw the minimal matching the same way we would for N, and consider that same behavior at 3 the same way we would for any other minimal matching of M. These conventions force a minimal matching of M with 1 or 0 attached in all cases. □

3.8 Results

For a D_n quiver, our indexing runs from 0 to $n-1$. We draw the quiver head on the right and extend the tail to the left. We label points starting from 0 on the right bottom, 1 on the right top, then increment the label by 1 as we go left.

Given an acyclic D_n-quiver, it is possible to draw the minimal matching of its highest degree F-polynomial in terms of a mixed dimer as follows:

- when the $n-1$ edge points to the right, the $n-1$ square has a 3-sided box in its minimum matching on the $n-1$ square
- when the $n-1$ edge points to the left, the $n-1$ square has only one side (the end of its tail) included in the minimum matching
- sources in the quiver at vertex i in the A-type tail (where $i \neq n-1$) correspond to double edges on i such that i has two sets of doubled edges on the boundary edges of the tail
- sinks in the quiver at vertex i in the A-type tail (where $i \neq n-1$) correspond to empty boxes on the minimal matching

- tails of a quiver with edges that all point in the same direction follow a pattern: each square $i \neq n-1$) in the minimal matching has one doubled edge, located on the boundary of the tail. By checking which way the $n-1$ edge points in this case by our first two statements above, we fill in these doubled edges such that each vertex of the matching has the allowed degree
- After addressing the sinks, sources, and the $n-1$ vertex of the quiver and which way the end of its dimer "tail" should be oriented, and then addressing portions of the tail with edges pointing in the same direction, the rest of the minimal matching in the dimer can be deduced by making sure each square's vertex has an allowed degree in the minimum matching

We can draw any other minimal matching for lower-degree F-polynomials by following the procedure outlined in Section 3.7.

3.9 References

Fomin-Zelevinsky 2001, *Cluster Algebras I: Foundations*, arXiV: 0104151.
Fomin-Zelevinsky 2002, *Cluster Algebras II: Finite Type Classification*, arXiV: 0208229.
Tran 2009, *F-polynomials in Quantum Cluster Algebras*, arXiV: 0904.3291.
Kenyon-Pemantle 2013, *Double-dimers, the Ising model and the hexahedron recurrence*, arXiV: 1308.2998.
Lee-Schiffler 2014, *Positivity for Cluster Algebras*, arXiV: 1306.2415.
Canakci-Schiffler 2016, *Cluster Algebras and Continued Fractions*, arXiV: 1608.06568.
Gross-Hacking-Keel-Kontsevich 2016, *Canonical Bases for Cluster Algebras*, arXiV: 1411.1394.

3.10 Acknowledgements

This research was carried out as part of the 2018 Combinatorics REU through the Mathematics department at the University of Minnesota, Twin Cities. The authors are grateful for the support of NSF RTG grant DMS-1745638, and would like to thank Gregg Musiker and Esther Banaian for their mentoring, intuition, and excitement while tackling this problem.