On surface cluster algebras: Snake graph calculus and dreaded torus

İlke Çanakçı

1Department of Mathematics
University of Leicester

joint work with Ralf Schiffler

Geometry Seminar, University of Bath
March 25, 2014
Outline of Topics

1. Surface cluster algebras
2. Abstract Snake Graphs
3. Relation to Cluster Algebras
4. Self-crossing snake graphs
5. Application
Cluster algebras were introduced by Fomin and Zelevinsky [FZ1] with the desire of creating an algebraic framework for the study of (dual) canonical bases in Lie theory.

Cluster algebras are defined by generators and relations, and the set of generators is constructed recursively from some initial data \((x, Q)\) called seed, where \(x = (x_1, \cdots, x_n)\) and \(Q\) is a quiver.

Cluster algebras form a class of combinatorially defined commutative algebras, and the set of generators of a cluster algebra, cluster variables, is obtained by an iterative process called seed mutation.

The cluster variables are rational functions in several variables \(x_1, x_2, \cdots, x_n\) by construction.

However, by a well-known result in [FZ1] they can be expressed as Laurent polynomials in \(x_1, x_2, \cdots, x_n\) with integer coefficients.
Overview

- Cluster algebras were introduced by Fomin and Zelevinsky [FZ1] with the desire of creating an algebraic framework for the study of (dual) canonical bases in Lie theory.
- Cluster algebras are defined by generators and relations, and the set of generators is constructed recursively from some initial data \((x, Q)\) called seed, where \(x = (x_1, \cdots, x_n)\) and \(Q\) is a quiver.
- Cluster algebras form a class of combinatorially defined commutative algebras, and the set of generators of a cluster algebra, cluster variables, is obtained by an iterative process called seed mutation.
- The cluster variables are rational functions in several variables \(x_1, x_2, \cdots, x_n\) by construction.
- However, by a well-known result in [FZ1] they can be expressed as Laurent polynomials in \(x_1, x_2, \cdots, x_n\) with integer coefficients.
Overview

- Cluster algebras were introduced by Fomin and Zelevinsky [FZ1] with the desire of creating an algebraic framework for the study of (dual) canonical bases in Lie theory.
- Cluster algebras are defined by generators and relations, and the set of generators is constructed recursively from some initial data \((x, Q)\) called seed, where \(x = (x_1, \ldots, x_n)\) and \(Q\) is a quiver.
- Cluster algebras form a class of combinatorially defined commutative algebras, and the set of generators of a cluster algebra, cluster variables, is obtained by an iterative process called seed mutation.
  - The cluster variables are rational functions in several variables \(x_1, x_2, \ldots, x_n\) by construction.
  - However, by a well-known result in [FZ1] they can be expressed as Laurent polynomials in \(x_1, x_2, \ldots, x_n\) with integer coefficients.
Overview

- Cluster algebras were introduced by Fomin and Zelevinsky [FZ1] with the desire of creating an algebraic framework for the study of (dual) canonical bases in Lie theory.

- Cluster algebras are defined by generators and relations, and the set of generators is constructed recursively from some initial data \((x, Q)\) called seed, where \(x = (x_1, \cdots, x_n)\) and \(Q\) is a quiver.

- Cluster algebras form a class of combinatorially defined commutative algebras, and the set of generators of a cluster algebra, cluster variables, is obtained by an iterative process called seed mutation.

- The cluster variables are rational functions in several variables \(x_1, x_2, \cdots, x_n\) by construction.

- However, by a well-known result in [FZ1] they can be expressed as Laurent polynomials in \(x_1, x_2, \cdots, x_n\) with integer coefficients.
Overview

- Cluster algebras were introduced by Fomin and Zelevinsky [FZ1] with the desire of creating an algebraic framework for the study of (dual) canonical bases in Lie theory.
- Cluster algebras are defined by generators and relations, and the set of generators is constructed recursively from some initial data \((x, Q)\) called seed, where \(x = (x_1, \ldots, x_n)\) and \(Q\) is a quiver.
- Cluster algebras form a class of combinatorially defined commutative algebras, and the set of generators of a cluster algebra, cluster variables, is obtained by an iterative process called seed mutation.
- The cluster variables are rational functions in several variables \(x_1, x_2, \ldots, x_n\) by construction.
- However, by a well-known result in [FZ1] they can be expressed as Laurent polynomials in \(x_1, x_2, \ldots, x_n\) with integer coefficients.
Overview

- Cluster algebras were introduced by Fomin and Zelevinsky [FZ1] with the desire of creating an algebraic framework for the study of (dual) canonical bases in Lie theory.

- Cluster algebras are defined by **generators** and **relations**, and the set of generators is constructed recursively from some **initial data** \((x, Q)\) called **seed**, where \(x = (x_1, \cdots, x_n)\) and \(Q\) is a quiver.

- Cluster algebras form a class of combinatorially defined commutative algebras, and the set of generators of a cluster algebra, **cluster variables**, is obtained by an iterative process called **seed mutation**.

- The cluster variables are **rational functions in several variables** \(x_1, x_2, \cdots, x_n\) by construction.

- However, by a well-known result in [FZ1] they can be expressed as **Laurent polynomials** in \(x_1, x_2, \cdots, x_n\) with integer coefficients.
Overview

- **Cluster algebras from surfaces**, introduced in [FST], have a geometric interpretation in surfaces.

- A surface cluster algebra $\mathcal{A}$ is associated to a surface $S$ with boundary that has finitely many marked points.

- Cluster variables are in bijection with certain curves [FST], called arcs. Two crossing arcs satisfy the skein relations, [MW].

- The authors in [MSW] associate a connected graph, called the snake graph to each arc in the surface to obtain a direct formula, the expansion formula, for cluster variables of surface cluster algebras.

\[
x_\gamma = \frac{1}{\text{cross}(\gamma, T)} \sum_{P \in G_\gamma} x(P)y(P)
\]
Overview

- **Cluster algebras from surfaces**, introduced in [FST], have a geometric interpretation in surfaces.
- A **surface cluster algebra** $\mathcal{A}$ is associated to a surface $S$ with boundary that has finitely many marked points.

- Cluster variables are in bijection with certain curves [FST], called arcs. Two crossing arcs satisfy the skein relations, [MW].
- The authors in [MSW] associate a connected graph, called the **snake graph** to each arc in the surface to obtain a direct formula, the **expansion formula**, for cluster variables of surface cluster algebras.

$$x_\gamma = \frac{1}{\text{cross} (\gamma, T)} \sum_{P \in G_\gamma} x(P)y(P)$$
Overview

- **Cluster algebras from surfaces**, introduced in [FST], have a geometric interpretation in surfaces.
- A **surface cluster algebra** $\mathcal{A}$ is associated to a surface $S$ with boundary that has finitely many marked points.

Cluster variables are in bijection with certain curves [FST], called arcs. Two crossing arcs satisfy the skein relations [MW].

The authors in [MSW] associate a connected graph, called the **snake graph** to each arc in the surface to obtain a direct formula, the **expansion formula**, for cluster variables of surface cluster algebras.

$$x_\gamma = \frac{1}{\text{cross } (\gamma, T)} \sum_{P \perp G_\gamma} x(P)y(P)$$
Overview

- **Cluster algebras from surfaces**, introduced in [FST], have a geometric interpretation in surfaces.
- A **surface cluster algebra** $\mathcal{A}$ is associated to a surface $S$ with boundary that has finitely many marked points.

- **Cluster variables** are in bijection with certain curves [FST], called arcs. Two crossing arcs satisfy the skein relations, [MW].
- The authors in [MSW] associate a connected graph, called the **snake graph** to each arc in the surface to obtain a direct formula, the **expansion formula**, for cluster variables of surface cluster algebras.

$$x_\gamma = \frac{1}{\text{cross}(\gamma, T)} \sum_{P \vdash G_\gamma} x(P)y(P)$$
Overview

- **Cluster algebras from surfaces**, introduced in [FST], have a geometric interpretation in surfaces.

- A **surface cluster algebra** $\mathcal{A}$ is associated to a surface $S$ with boundary that has finitely many marked points.

- Cluster variables are in bijection with certain curves [FST], called arcs. Two crossing arcs satisfy the skein relations, [MW].

- The authors in [MSW] associate a connected graph, called the snake graph to each arc in the surface to obtain a direct formula, the expansion formula, for cluster variables of surface cluster algebras.

$$ x_\gamma = \frac{1}{\text{cross}(\gamma, T)} \sum_{P \in G_\gamma} x(P) y(P) $$
Cluster algebras from surfaces, introduced in [FST], have a geometric interpretation in surfaces.

A surface cluster algebra $\mathcal{A}$ is associated to a surface $S$ with boundary that has finitely many marked points.

Cluster variables are in bijection with certain curves [FST], called arcs. Two crossing arcs satisfy the skein relations, [MW].

The authors in [MSW] associate a connected graph, called the snake graph to each arc in the surface to obtain a direct formula, the expansion formula, for cluster variables of surface cluster algebras.

$$x_\gamma = \frac{1}{\text{cross}(\gamma, T)} \sum_{P \in G_\gamma} x(P)y(P)$$
Overview

- **Cluster algebras from surfaces**, introduced in [FST], have a geometric interpretation in surfaces.
- A **surface cluster algebra** \( \mathcal{A} \) is associated to a surface \( S \) with boundary that has finitely many marked points.

\[
x_\gamma = \frac{1}{\text{cross } (\gamma, T)} \sum_{P \in G_\gamma} x(P)y(P)
\]

- **Cluster variables** are in bijection with certain curves [FST], called **arcs**. Two crossing arcs satisfy the **skein relations**, [MW].
- The authors in [MSW] associate a connected graph, called the **snake graph** to each arc in the surface to obtain a direct formula, the **expansion formula**, for cluster variables of surface cluster algebras.
Overview

- **Cluster algebras from surfaces**, introduced in [FST], have a geometric interpretation in surfaces.
- A **surface cluster algebra** $\mathcal{A}$ is associated to a surface $S$ with boundary that has finitely many marked points.

![Diagram](image)

- **Cluster variables** are in bijection with certain curves [FST], called *arcs*. Two crossing arcs satisfy the **skein relations**, [MW].
- The authors in [MSW] associate a connected graph, called the **snake graph** to each arc in the surface to obtain a direct formula, the **expansion formula**, for cluster variables of surface cluster algebras.

$$x_\gamma = \frac{1}{\text{cross}(\gamma, T)} \sum_{P \in G_\gamma} x(P)y(P)$$
Overview

- **Cluster algebras from surfaces**, introduced in [FST], have a geometric interpretation in surfaces.
- A **surface cluster algebra** $\mathcal{A}$ is associated to a surface $S$ with boundary that has finitely many marked points.

\[
x_{\gamma_1} x_{\gamma_2} = *x_{\gamma_3} x_{\gamma_4} + *x_{\gamma_5} x_{\gamma_6}
\]
Skein relation ([MW])

- **Cluster variables** are in bijection with certain curves [FST], called **arcs**. Two crossing arcs satisfy the **skein relations**, [MW].
- The authors in [MSW] associate a connected graph, called the **snake graph** to each arc in the surface to obtain a direct formula, the **expansion formula**, for cluster variables of surface cluster algebras.

\[
x_\gamma = \frac{1}{\text{cross}(\gamma, T)} \sum_{P \in G_\gamma} x(P)y(P)
\]
Overview

- **Cluster algebras from surfaces**, introduced in [FST], have a geometric interpretation in surfaces.

- A **surface cluster algebra** $\mathcal{A}$ is associated to a surface $S$ with boundary that has finitely many marked points.

- **Cluster variables** are in bijection with certain curves [FST], called arcs. Two crossing arcs satisfy the skein relations, [MW].

- The authors in [MSW] associate a connected graph, called the **snake graph** to each arc in the surface to obtain a direct formula, the **expansion formula**, for cluster variables of surface cluster algebras.

\[ x_{\gamma} = \frac{1}{\text{cross}(\gamma, T)} \sum_{P \perp G_{\gamma}} x(P)y(P) \]
Motivation

Let $\mathcal{A}(S, M)$ cluster algebra associated to a surface $(S, M)$.

We have the following situation:

Question

“How much can we recover from snake graphs themselves?”

In particular,

- When do the two arcs corresponding to two snake graphs cross?
- What are the snake graphs corresponding to the skein relations?
Let $\mathcal{A}(S, M)$ cluster algebra associated to a surface $(S, M)$.

We have the following situation:

cluster variable $\leftrightarrow_{[FST]}$ arc

**Question**

“How much can we recover from snake graphs themselves?”

In particular,

- When do the two arcs corresponding to two snake graphs cross?
- What are the snake graphs corresponding to the skein relations?
Motivation

Let \( \mathcal{A}(S, M) \) cluster algebra associated to a surface \((S, M)\).

We have the following situation:

\[
\begin{array}{c}
\text{cluster variable} \\ \xleftarrow{[\text{FST}]} \\
\text{arc} \\ \xrightarrow{[\text{MSW}]} \\
\text{snake graph}
\end{array}
\]

Question

“How much can we recover from snake graphs themselves?”

In particular,

- When do the two arcs corresponding to two snake graphs cross?
- What are the snake graphs corresponding to the skein relations?
Let $\mathcal{A}(S, M)$ cluster algebra associated to a surface $(S, M)$.

We have the following situation:

| cluster variable | $\xrightarrow{\text{[FST]}}$ | arc | $\xrightarrow{\text{[MSW]}}$ | snake graph |

**Question**

“How much can we recover from snake graphs themselves?”

In particular,

- When do the two arcs corresponding to two snake graphs cross?
- What are the snake graphs corresponding to the skein relations?
Let $\mathcal{A}(S, M)$ cluster algebra associated to a surface $(S, M)$.

We have the following situation:

| cluster variable | $\xleftarrow{[\text{FST}]}$ | arc | $\xrightarrow{[\text{MSW}]}$ | snake graph |

**Question**

"How much can we recover from snake graphs themselves?"

In particular,

- When do the two arcs corresponding to two snake graphs cross?
- What are the snake graphs corresponding to the skein relations?
Let \( \mathcal{A}(S, M) \) cluster algebra associated to a surface \((S, M)\).

We have the following situation:

\[
\begin{array}{c}
\text{cluster variable} & \xrightarrow{[\text{FST}]} & \text{arc} & \xrightarrow{[\text{MSW}]} & \text{snake graph}
\end{array}
\]

**Question**

"How much can we recover from snake graphs themselves?"

In particular,

- When do the two arcs corresponding to two snake graphs cross?
- What are the snake graphs corresponding to the skein relations?
Motivation

Let $\mathcal{A}(S, M)$ cluster algebra associated to a surface $(S, M)$.

We have the following situation:

| cluster variable | $\leftarrow\rightarrow_{\text{[FST]}}$ | arc | $\rightarrow_{\text{[MSW]}}$ | snake graph |

Question

“How much can we recover from snake graphs themselves?”

In particular,

- When do the two arcs corresponding to two snake graphs cross?
- What are the snake graphs corresponding to the skein relations?
Surface Cluster Algebras

- Let $S$ be a connected oriented 2-dimensional Riemann surface with nonempty boundary, and let $M$ be a nonempty finite subset of the boundary of $S$, such that each boundary component of $S$ contains at least one point of $M$. The elements of $M$ are called marked points. The pair $(S, M)$ is called a bordered surface with marked points.
Surface Cluster Algebras

- Let $S$ be a connected oriented 2-dimensional Riemann surface with nonempty boundary, and let $M$ be a nonempty finite subset of the boundary of $S$, such that each boundary component of $S$ contains at least one point of $M$. The elements of $M$ are called marked points. The pair $(S, M)$ is called a bordered surface with marked points.
Surface Cluster Algebras

Definition
An arc γ in (S, M) is a curve in S, considered up to isotopy, such that:
- the endpoints of γ are in M;
- γ does not cross itself;
- except for the endpoints, γ is disjoint from the boundary of S;
- and
- γ does not cut out a monogon or a bigon.

Remark
Curves that connect two marked points and lie entirely on the boundary of S without passing through a third marked point are boundary segments. Note that boundary segments are not arcs.
Surface Cluster Algebras

Definition
An arc $\gamma$ in $(S, M)$ is a curve in $S$, considered up to isotopy, such that:

- the endpoints of $\gamma$ are in $M$;
- $\gamma$ does not cross itself;
- except for the endpoints, $\gamma$ is disjoint from the boundary of $S$; and
- $\gamma$ does not cut out a monogon or a bigon.

Remark
Curves that connect two marked points and lie entirely on the boundary of $S$ without passing through a third marked point are boundary segments. Note that boundary segments are not arcs.
Surface Cluster Algebras

Definition
For any two arcs $\gamma, \gamma'$ in $S$, let $e(\gamma, \gamma')$ be the minimal number of crossings of arcs $\alpha$ and $\alpha'$, where $\alpha$ and $\alpha'$ range over all arcs isotopic to $\gamma$ and $\gamma'$, respectively. We say that arcs $\gamma$ and $\gamma'$ are compatible if $e(\gamma, \gamma') = 0$.

Definition
A triangulation is a maximal collection of pairwise compatible arcs (together with all boundary segments).
Definition
For any two arcs $\gamma, \gamma'$ in $S$, let $e(\gamma, \gamma')$ be the minimal number of crossings of arcs $\alpha$ and $\alpha'$, where $\alpha$ and $\alpha'$ range over all arcs isotopic to $\gamma$ and $\gamma'$, respectively. We say that arcs $\gamma$ and $\gamma'$ are compatible if $e(\gamma, \gamma') = 0$.

Definition
A triangulation is a maximal collection of pairwise compatible arcs (together with all boundary segments).
Surface Cluster Algebras

Definition
For any two arcs $\gamma, \gamma'$ in $S$, let $e(\gamma, \gamma')$ be the minimal number of crossings of arcs $\alpha$ and $\alpha'$, where $\alpha$ and $\alpha'$ range over all arcs isotopic to $\gamma$ and $\gamma'$, respectively. We say that arcs $\gamma$ and $\gamma'$ are compatible if $e(\gamma, \gamma') = 0$.

Definition
A triangulation is a maximal collection of pairwise compatible arcs (together with all boundary segments).
Definition
Triangulations are connected to each other by sequences of flips. Each flip replaces a single arc $\gamma$ in a triangulation $T$ by a (unique) arc $\gamma' \neq \gamma$ that, together with the remaining arcs in $T$, forms a new triangulation.
**Surface Cluster Algebras**

### Definition

Triangulations are connected to each other by sequences of **flips**. Each flip replaces a single arc $\gamma$ in a triangulation $T$ by a (unique) arc $\gamma' \neq \gamma$ that, together with the remaining arcs in $T$, forms a new triangulation.
Surface Cluster Algebras

Definition

Triangulations are connected to each other by sequences of \textbf{flips}. Each flip replaces a single arc $\gamma$ in a triangulation $T$ by a (unique) arc $\gamma' \neq \gamma$ that, together with the remaining arcs in $T$, forms a new triangulation.
Definition

Triangulations are connected to each other by sequences of \textit{flips}. Each flip replaces a single arc $\gamma$ in a triangulation $T$ by a (unique) arc $\gamma' \neq \gamma$ that, together with the remaining arcs in $T$, forms a new triangulation.
Definition
Triangulations are connected to each other by sequences of \textbf{flips}. Each flip replaces a single arc $\gamma$ in a triangulation $T$ by a (unique) arc $\gamma' \neq \gamma$ that, together with the remaining arcs in $T$, forms a new triangulation.
Surface Cluster Algebras

Definition

Triangulations are connected to each other by sequences of flips. Each flip replaces a single arc $\gamma$ in a triangulation $T$ by a (unique) arc $\gamma' \neq \gamma$ that, together with the remaining arcs in $T$, forms a new triangulation.
**Definition**

Triangulations are connected to each other by sequences of **flips**. Each flip replaces a single arc $\gamma$ in a triangulation $T$ by a (unique) arc $\gamma' \neq \gamma$ that, together with the remaining arcs in $T$, forms a new triangulation.
Surface Cluster Algebras

Definition

Triangulations are connected to each other by sequences of \textbf{flips}. Each flip replaces a single arc $\gamma$ in a triangulation $T$ by a (unique) arc $\gamma' \neq \gamma$ that, together with the remaining arcs in $T$, forms a new triangulation.
Surface Cluster Algebras

Theorem (FST,FT)

For cluster algebras from surfaces
- there are bijections

\[
\begin{align*}
\{ \text{arcs} \} & \quad \longrightarrow \quad \{ \text{cluster variables} \} \\
\gamma & \quad \mapsto \quad x_\gamma \\
\{ \text{triangulations} \} & \quad \longrightarrow \quad \{ \text{clusters} \} \\
T = \{\tau_1, \ldots, \tau_n\} & \quad \mapsto \quad x_T = \{x_{\tau_1}, \ldots, x_{\tau_n}\}
\end{align*}
\]

- The triangulation \( T \setminus \{\tau_k\} \cup \{\tau'_k\} \) obtained by flipping the arc \( \tau_k \)
corresponds to the mutation \( \mu_k(x_T) = x_T \setminus \{x_{\tau_k}\} \cup \{x_{\tau'_k}\} \).

Definition

The surface cluster algebra \( \mathcal{A} = \mathcal{A}(S, M) \) associated to a surface \((S, M)\) is a \(\mathbb{Z}\)-subalgebra of \(\mathbb{Q}(x_1, \ldots, x_n)\) generated by all cluster variables \(x_\gamma\).
Snake graphs and perfect matchings

For each arc $\gamma$ in a surface $(S, M, T)$, we associate a weighted graph $G_\gamma$, called snake graph, from $\gamma$ and $T$. A perfect matching $P$ of a graph $G$ is a subset of the set of edges of $G$ such that each vertex of $G$ is incident to exactly one edge in $P$. 
Snake graphs and perfect matchings

For each arc $\gamma$ in a surface $(S, M, T)$, we associate a weighted graph $G_\gamma$, called **snake graph**, from $\gamma$ and $T$.

A perfect matching $P$ of a graph $G$ is a subset of the set of edges of $G$ such that each vertex of $G$ is incident to exactly one edge in $P$. 
Snake graphs and perfect matchings

For each arc $\gamma$ in a surface $(S, M, T)$, we associate a weighted graph $G_\gamma$, called **snake graph**, from $\gamma$ and $T$.

A perfect matching $P$ of a graph $G$ is a subset of the set of edges of $G$ such that each vertex of $G$ is incident to exactly one edge in $P$. 
Snake graphs and perfect matchings

For each arc $\gamma$ in a surface $(S, M, T)$, we associate a weighted graph $G_\gamma$, called snake graph, from $\gamma$ and $T$.

A perfect matching $P$ of a graph $G$ is a subset of the set of edges of $G$ such that each vertex of $G$ is incident to exactly one edge in $P$. 
Snake graphs and perfect matchings

For each arc $\gamma$ in a surface $(S, M, T)$, we associate a weighted graph $G_\gamma$, called snake graph, from $\gamma$ and $T$.

A perfect matching $P$ of a graph $G$ is a subset of the set of edges of $G$ such that each vertex of $G$ is incident to exactly one edge in $P$. 
Snake graphs and perfect matchings

For each arc $\gamma$ in a surface $(S, M, T)$, we associate a weighted graph $G_\gamma$, called **snake graph**, from $\gamma$ and $T$.

A perfect matching $P$ of a graph $G$ is a subset of the set of edges of $G$ such that each vertex of $G$ is incident to exactly one edge in $P$. 
Snake graphs and perfect matchings

For each arc $\gamma$ in a surface $(S, M, T)$, we associate a weighted graph $G_\gamma$, called snake graph, from $\gamma$ and $T$.

A perfect matching $P$ of a graph $G$ is a subset of the set of edges of $G$ such that each vertex of $G$ is incident to exactly one edge in $P$. 
Snake graphs and perfect matchings

For each arc $\gamma$ in a surface $(S, M, T)$, we associate a weighted graph $G_\gamma$, called **snake graph**, from $\gamma$ and $T$.

A **perfect matching** $P$ of a graph $G$ is a subset of the set of edges of $G$ such that each vertex of $G$ is incident to exactly one edge in $P$. 
Snake graphs and perfect matchings

For each arc $\gamma$ in a surface $(S, M, T)$, we associate a weighted graph $G_\gamma$, called **snake graph**, from $\gamma$ and $T$.

A **perfect matching** $P$ of a graph $G$ is a subset of the set of edges of $G$ such that each vertex of $G$ is incident to exactly one edge in $P$.
Snake graphs and perfect matchings

For each arc $\gamma$ in a surface $(S, M, T)$, we associate a weighted graph $G_\gamma$, called **snake graph**, from $\gamma$ and $T$.

A **perfect matching** $P$ of a graph $G$ is a subset of the set of edges of $G$ such that each vertex of $G$ is incident to exactly one edge in $P$. 
The authors in [MSW] gives an explicit formula, called **expansion formula**, for cluster variables. The formula is given by

\[
x_\gamma = \frac{1}{\mathrm{cross}(\gamma, T)} \sum_{P \vdash G_\gamma} x(P)y(P)
\]

where the sum is over all perfect matchings \( P \) of \( G_\gamma \).
The authors in [MSW] give an explicit formula, called **expansion formula**, for cluster variables. The formula is given by

$$x_{\gamma} = \frac{1}{\text{cross}(\gamma, T)} \sum_{P \perp G_{\gamma}} x(P)y(P)$$

where the sum is over all perfect matchings $P$ of $G_{\gamma}$. 

![Diagram](attachment:image.png)
Expansion formula

The authors in [MSW] gives an explicit formula, called expansion formula, for cluster variables. The formula is given by

$$x_\gamma = \frac{1}{\text{cross}(\gamma, T)} \sum_{P \sqsubset G_\gamma} x(P)y(P)$$

where the sum is over all perfect matchings $P$ of $G_\gamma$.

$$x(P) = x_3x_4x_7$$
Expansion formula

The authors in [MSW] give an explicit formula, called expansion formula, for cluster variables. The formula is given by

\[ x_{\gamma} = \frac{1}{\text{cross}(\gamma, T)} \sum_{P \in G_\gamma} x(P)y(P) \]

where the sum is over all perfect matchings \( P \) of \( G_\gamma \).

\[ x(P) = x_3x_4x_7 \quad \text{and} \quad x(P) = x_2x_3x_4^2x_6x_7 \]
Applying the formula, the cluster variable corresponding to the arc $\gamma$ is given by

$$x_\gamma = \frac{1}{x_1 x_2 x_3 x_4 x_5 x_6 x_7} (x_1 x_2 x_3 x_5^2 x_6 + y_4 x_1 x_2 x_5 x_6 + y_7 x_1 x_2 x_3 x_5^2 + y_3 y_4 x_1 x_4 x_5 x_6 + y_4 y_7 x_1 x_2 x_5 + y_6 y_7 x_1 x_2 x_3 x_5 x_7 + y_2 y_3 y_4 x_3 x_4 x_5 x_6 + y_3 y_4 y_7 x_1 x_4 x_5 + y_4 y_6 y_7 x_1 x_2 x_7 + y_1 y_2 y_3 y_4 x_2 x_3 x_4 x_5 x_6 + y_2 y_3 y_4 y_7 x_3 x_4 x_5 + y_3 y_4 y_6 y_7 x_1 x_4 x_7 + y_4 y_5 y_6 y_7 x_1 x_2 x_4 x_6 x_7 + y_1 y_2 y_3 y_4 y_7 x_2 x_3 x_4 x_5 + y_2 y_3 y_4 y_6 y_7 x_3 x_4 x_7 + y_3 y_4 y_5 y_6 y_7 x_1 x_4^2 x_5 x_7 + y_1 y_2 y_3 y_4 y_5 y_6 y_7 x_2 x_3 x_4 x_7 + y_2 y_3 y_4 y_5 y_6 y_7 x_3 x_4^2 x_6 x_7 + y_1 y_2 y_3 y_4 y_5 y_6 y_7 x_2 x_3 x_4^2 x_6 x_7).$$
Our results

- We introduce the notion of an abstract snake graph, which is not necessarily related to an arc in a surface.
- We define what it means for two abstract snake graphs to cross.
- Given two crossing snake graphs, we construct the resolution of the crossing as two pairs of snake graphs from the original pair of crossing snake graphs.
- We then prove that there is a bijection \( \varphi \) between the set of perfect matchings of the two crossing snake graphs and the set of perfect matchings of the resolution.
- We then apply our constructions to snake graphs arising from unpunctured surfaces.
- We then extend our results to self-crossing snake graphs associated to self-crossing arcs in a surface.
Our results

- We introduce the notion of an abstract snake graph, which is not necessarily related to an arc in a surface.
- We define what it means for two abstract snake graphs to cross.
- Given two crossing snake graphs, we construct the resolution of the crossing as two pairs of snake graphs from the original pair of crossing snake graphs.
- We then prove that there is a bijection $\varphi$ between the set of perfect matchings of the two crossing snake graphs and the set of perfect matchings of the resolution.
- We then apply our constructions to snake graphs arising from unpunctured surfaces.
- We then extend our results to self-crossing snake graphs associated to self-crossing arcs in a surface.
On surface cluster algebras

İlke Çanakçı

Surface cluster algebras
Abstract Snake Graphs
Relation to Cluster Algebras
Self-crossing snake graphs
Application

Our results

• We introduce the notion of an abstract snake graph, which is not necessarily related to an arc in a surface.

• We define what it means for two abstract snake graphs to cross.

• Given two crossing snake graphs, we construct the resolution of the crossing as two pairs of snake graphs from the original pair of crossing snake graphs.

• We then prove that there is a bijection $\varphi$ between the set of perfect matchings of the two crossing snake graphs and the set of perfect matchings of the resolution.

• We then apply our constructions to snake graphs arising from unpunctured surfaces.

• We then extend our results to self-crossing snake graphs associated to self-crossing arcs in a surface.
Our results

- We introduce the notion of an **abstract snake graph**, which is not necessarily related to an arc in a surface.
- We define what it means for two abstract **snake graphs to cross**.
- Given two crossing snake graphs, we construct the **resolution** of the crossing as two pairs of snake graphs from the original pair of crossing snake graphs.
- We then prove that there is a **bijection** \( \varphi \) between the set of perfect matchings of the two crossing snake graphs and the set of perfect matchings of the resolution.
- We then apply our constructions to snake graphs arising from unpunctured surfaces.
- We then extend our results to **self-crossing snake graphs** associated to self-crossing arcs in a surface.
Our results

- We introduce the notion of an **abstract snake graph**, which is not necessarily related to an arc in a surface.

- We define what it means for two abstract **snake graphs to cross**.

- Given two crossing snake graphs, we construct the **resolution** of the crossing as two pairs of snake graphs from the original pair of crossing snake graphs.

- We then prove that there is a **bijection** $\varphi$ between the set of perfect matchings of the two crossing snake graphs and the set of perfect matchings of the resolution.

- We then apply our constructions to snake graphs arising from **unpunctured surfaces**.

- We then extend our results to **self-crossing snake graphs** associated to self-crossing arcs in a surface.
Our results

- We introduce the notion of an **abstract snake graph**, which is not necessarily related to an arc in a surface.
- We define what it means for two abstract **snake graphs to cross**.
- Given two crossing snake graphs, we construct the **resolution** of the crossing as two pairs of snake graphs from the original pair of crossing snake graphs.
- We then prove that there is a **bijection** $\varphi$ between the set of perfect matchings of the two crossing snake graphs and the set of perfect matchings of the resolution.
- We then apply our constructions to snake graphs arising from **unpunctured surfaces**.
- We then extend our results to **self-crossing snake graphs** associated to self-crossing arcs in a surface.
Our results

- We introduce the notion of an **abstract snake graph**, which is not necessarily related to an arc in a surface.
- We define what it means for two abstract **snake graphs to cross**.
- Given two crossing snake graphs, we construct the **resolution** of the crossing as two pairs of snake graphs from the original pair of crossing snake graphs.
- We then prove that there is a **bijection** $\varphi$ between the set of perfect matchings of the two crossing snake graphs and the set of perfect matchings of the resolution.
- We then apply our constructions to snake graphs arising from **unpunctured surfaces**.
- We then extend our results to **self-crossing snake graphs** associated to self-crossing arcs in a surface.
Abstract Snake Graphs

Definition

A **snake graph** \( G \) is a connected graph in \( \mathbb{R}^2 \) consisting of a finite sequence of tiles \( G_1, G_2, \ldots, G_d \) with \( d \geq 1 \), such that for each \( i = 1, \ldots, d - 1 \)

(i) \( G_i \) and \( G_{i+1} \) share exactly one edge \( e_i \) and this edge is either the north edge of \( G_i \) and the south edge of \( G_{i+1} \) or the east edge of \( G_i \) and the west edge of \( G_{i+1} \).

(ii) \( G_i \) and \( G_j \) have no edge in common whenever \( |i - j| \geq 2 \).

(iii) \( G_i \) and \( G_j \) are disjoint whenever \( |i - j| \geq 3 \).

Example
Abstract Snake Graphs

Definition
A snake graph $G$ is a connected graph in $\mathbb{R}^2$ consisting of a finite sequence of tiles $G_1, G_2, \ldots, G_d$ with $d \geq 1$, such that for each $i = 1, \ldots, d - 1$

(i) $G_i$ and $G_{i+1}$ share exactly one edge $e_i$ and this edge is either the north edge of $G_i$ and the south edge of $G_{i+1}$ or the east edge of $G_i$ and the west edge of $G_{i+1}$.

(ii) $G_i$ and $G_j$ have no edge in common whenever $|i - j| \geq 2$.

(iii) $G_i$ and $G_j$ are disjoint whenever $|i - j| \geq 3$.

Example

\[ \text{G} \]
Example

Notation

- $G = (G_1, G_2, \ldots, G_d)$
- $G[i:i+t] = (G_i, G_{i+1}, \ldots, G_{i+t})$
- We denote by $e_i$ the interior edge between the tiles $G_i$ and $G_{i+1}$.
Example

Notation

- $G = (G_1, G_2, \ldots, G_d)$
- $G[i, i+t] = (G_i, G_{i+1}, \ldots, G_{i+t})$
- We denote by $e_i$ the interior edge between the tiles $G_i$ and $G_{i+1}$.
Example

\[ G \]

\[ G_1 \]

\[ G_2 \]

Notation

- \( G = (G_1, G_2, \ldots, G_d) \)
- \( G[i, i + t] = (G_i, G_{i+1}, \ldots, G_{i+t}) \)
- We denote by \( e_i \) the interior edge between the tiles \( G_i \) and \( G_{i+1} \).
Example

Notation

- $\mathcal{G} = (G_1, G_2, \ldots, G_d)$
- $\mathcal{G}[i, i+t] = (G_i, G_{i+1}, \ldots, G_{i+t})$
- We denote by $e_i$ the interior edge between the tiles $G_i$ and $G_{i+1}$. 
Example

Notation

- $\mathcal{G} = (G_1, G_2, \ldots, G_d)$
- $\mathcal{G}[i, i + t] = (G_i, G_{i+1}, \ldots, G_{i+t})$
- We denote by $e_i$ the interior edge between the tiles $G_i$ and $G_{i+1}$. 
Local Overlaps

Definition
We say two snake graphs \( G_1 \) and \( G_2 \) have a **local overlap** \( G \) if \( G \) is a maximal subgraph contained in both \( G_1 \) and \( G_2 \).

**Notation:** \( G \cong G_1[s, \cdots, t] \cong G_2[s', \cdots, t'] \).

Example

Therefore \( G \) is a local overlap of \( G_1 \) and \( G_2 \).

Note that two snake graphs may have several overlaps.
Local Overlaps

Definition
We say two snake graphs $G_1$ and $G_2$ have a **local overlap** $G$ if $G$ is a maximal subgraph contained in both $G_1$ and $G_2$.

**Notation:** $G \simeq G_1[s, \cdots, t] \simeq G_2[s', \cdots, t']$.

Example

Therefore $G$ is a local overlap of $G_1$ and $G_2$.

Note that two snake graphs may have several overlaps.
Local Overlaps

Definition
We say two snake graphs $G_1$ and $G_2$ have a local overlap $G$ if $G$ is a maximal subgraph contained in both $G_1$ and $G_2$.

Notation: $G \cong G_1[s, \ldots, t] \cong G_2[s', \ldots, t']$.

Example

Therefore $G$ is a local overlap of $G_1$ and $G_2$.

Note that two snake graphs may have several overlaps.
Local Overlaps

Definition
We say two snake graphs $G_1$ and $G_2$ have a **local overlap** $G$ if $G$ is a maximal subgraph contained in both $G_1$ and $G_2$.

**Notation:** $G \cong G_1[s, \cdots, t] \cong G_2[s', \cdots, t'].$

Example

Therefore $G$ is a local overlap of $G_1$ and $G_2$.

- Note that two snake graphs may have several overlaps.
Local Overlaps

**Definition**

We say two snake graphs $G_1$ and $G_2$ have a **local overlap** $G$ if $G$ is a maximal subgraph contained in both $G_1$ and $G_2$.

**Notation:** $G \cong G_1[s, \cdots, t] \cong G_2[s', \cdots, t']$.

**Example**

\[ G_1 \]
\[ G_2 \]

Therefore $G$ is a local overlap of $G_1$ and $G_2$.

Note that two snake graphs may have several overlaps.
Local Overlaps

Definition

We say two snake graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ have a **local overlap** $\mathcal{G}$ if $\mathcal{G}$ is a maximal subgraph contained in both $\mathcal{G}_1$ and $\mathcal{G}_2$.

**Notation:** $\mathcal{G} \cong \mathcal{G}_1[s, \cdots, t] \cong \mathcal{G}_2[s', \cdots, t']$.

Example

![Snake Graphs]

Therefore $\mathcal{G}$ is a local overlap of $\mathcal{G}_1$ and $\mathcal{G}_2$.

- Note that two snake graphs may have several overlaps.
Local Overlaps

Definition
We say two snake graphs $G_1$ and $G_2$ have a **local overlap** $G$ if $G$ is a maximal subgraph contained in both $G_1$ and $G_2$.

**Notation:** $G \cong G_1 [s, \cdots, t] \cong G_2 [s', \cdots, t']$.

Example

Therefore $G$ is a local overlap of $G_1$ and $G_2$.

- Note that two snake graphs may have several overlaps.
Local Overlaps

**Definition**
We say two snake graphs $G_1$ and $G_2$ have a **local overlap** $G$ if $G$ is a maximal subgraph contained in both $G_1$ and $G_2$.

**Notation:** $G \cong G_1[s, \cdots, t] \cong G_2[s', \cdots, t']$.

**Example**

![Diagram of snake graphs]

Therefore $G$ is a local overlap of $G_1$ and $G_2$.

- Note that two snake graphs may have several overlaps.
Local Overlaps

Definition
We say two snake graphs $G_1$ and $G_2$ have a **local overlap** $G$ if $G$ is a maximal subgraph contained in both $G_1$ and $G_2$.

**Notation:** $G \cong G_1[s, \cdots, t] \cong G_2[s', \cdots, t']$.

Example

Therefore $G$ is a local overlap of $G_1$ and $G_2$.

- Note that two snake graphs may have several overlaps.
Local Overlaps

Definition
We say two snake graphs $G_1$ and $G_2$ have a **local overlap** $G$ if $G$ is a maximal subgraph contained in both $G_1$ and $G_2$.

**Notation:** $G \cong G_1[s, \cdots, t] \cong G_2[s', \cdots, t']$.

Example

Therefore $G$ is a local overlap of $G_1$ and $G_2$.

- Note that two snake graphs may have several overlaps.
Local Overlaps

**Definition**
We say two snake graphs $G_1$ and $G_2$ have a **local overlap** $G$ if $G$ is a maximal subgraph contained in both $G_1$ and $G_2$.

**Notation:** $G \cong G_1[s, \ldots, t] \cong G_2[s', \ldots, t']$.

**Example**

Therefore $G$ is a local overlap of $G_1$ and $G_2$.

- Note that two snake graphs may have several overlaps.
Sign Function

Definition
A sign function $f$ on a snake graph $G$ is a map $f$ from the set of edges of $G$ to $\{+, -\}$ such that on every tile in $G$ the north and the west edge have the same sign, the south and the east edge have the same sign and the sign on the north edge is opposite to the sign on the south edge.

Example
A sign function on $G_1$ and $G_2$
Sign Function

Definition
A sign function $f$ on a snake graph $G$ is a map $f$ from the set of edges of $G$ to $\{+,-\}$ such that on every tile in $G$ the north and the west edge have the same sign, the south and the east edge have the same sign and the sign on the north edge is opposite to the sign on the south edge.

Example
A sign function on $G_1$ and $G_2$
Definition
A sign function $f$ on a snake graph $G$ is a map $f$ from the set of edges of $G$ to $\{+,-\}$ such that on every tile in $G$ the north and the west edge have the same sign, the south and the east edge have the same sign and the sign on the north edge is opposite to the sign on the south edge.

Example
A sign function on $G_1$ and $G_2$
Sign Function

Definition

A **sign function** $f$ on a snake graph $G$ is a map $f$ from the set of edges of $G$ to $\{+,-\}$ such that on every tile in $G$ the north and the west edge have the same sign, the south and the east edge have the same sign and the sign on the north edge is opposite to the sign on the south edge.

Example

A sign function on $G_1$ and $G_2$
Sign Function

Definition
A **sign function** $f$ on a snake graph $G$ is a map $f$ from the set of edges of $G$ to $\{+, -\}$ such that on every tile in $G$ the north and the west edge have the same sign, the south and the east edge have the same sign and the sign on the north edge is opposite to the sign on the south edge.

Example
A sign function on $G_1$ and $G_2$
Sign Function

Definition
A sign function $f$ on a snake graph $G$ is a map $f$ from the set of edges of $G$ to $\{+,-\}$ such that on every tile in $G$ the north and the west edge have the same sign, the south and the east edge have the same sign and the sign on the north edge is opposite to the sign on the south edge.

Example
A sign function on $G_1$ and $G_2$
Sign Function

Definition

A **sign function** $f$ on a snake graph $G$ is a map $f$ from the set of edges of $G$ to $\{+,-\}$ such that on every tile in $G$ the north and the west edge have the same sign, the south and the east edge have the same sign and the sign on the north edge is opposite to the sign on the south edge.

Example

A sign function on $G_1$ and $G_2$
Sign Function

Definition

A sign function \( f \) on a snake graph \( G \) is a map \( f \) from the set of edges of \( G \) to \( \{+,-\} \) such that on every tile in \( G \) the north and the west edge have the same sign, the south and the east edge have the same sign and the sign on the north edge is opposite to the sign on the south edge.

Example

A sign function on \( G_1 \) and \( G_2 \)
**Sign Function**

**Definition**

A **sign function** $f$ on a snake graph $G$ is a map $f$ from the set of edges of $G$ to $\{+,-\}$ such that on every tile in $G$ the north and the west edge have the same sign, the south and the east edge have the same sign and the sign on the north edge is opposite to the sign on the south edge.

**Example**

A sign function on $G_1$ and $G_2$
Sign Function

Definition
A sign function $f$ on a snake graph $G$ is a map $f$ from the set of edges of $G$ to \{+,-\} such that on every tile in $G$ the north and the west edge have the same sign, the south and the east edge have the same sign and the sign on the north edge is opposite to the sign on the south edge.

Example
A sign function on $G_1$ and $G_2$
Sign Function

Definition
A sign function $f$ on a snake graph $G$ is a map $f$ from the set of edges of $G$ to $\{+,-\}$ such that on every tile in $G$ the north and the west edge have the same sign, the south and the east edge have the same sign and the sign on the north edge is opposite to the sign on the south edge.

Example
A sign function on $G_1$ and $G_2$
Definition
We say that $G_1$ and $G_2$ cross in a local overlap $G$ if one of the following conditions hold.

- $f_1(e_{s-1}) = -f_1(e_t)$ if $s > 1$, $t < d$
- $f_1(e_{s-1}) = f_2(e_{t'})$ if $s > 1$, $t < d$, $s' = 1$, $t' < d'$

Example
$G_1$ and $G_2$ cross at the overlap $G$. 
Definition
We say that $G_1$ and $G_2$ cross in a local overlap $G$ if one of the following conditions hold.

- $f_1(e_{s-1}) = -f_1(e_t)$ if $s > 1$, $t < d$
- $f_1(e_{s-1}) = f_2(e'_{t'})$ if $s > 1$, $t < d$, $s' = 1$, $t' < d'$

Example
$G_1$ and $G_2$ cross at the overlap $G$. 
Definition
We say that $G_1$ and $G_2$ cross in a local overlap $G$ if one of the following conditions hold.

- $f_1(e_{s-1}) = -f_1(e_t)$ if $s > 1$, $t < d$
- $f_1(e_{s-1}) = f_2(e'_{t'})$ if $s > 1$, $t < d$, $s' = 1$, $t' < d'$

Example
$G_1$ and $G_2$ cross at the overlap $G$.
**Definition**

We say that $G_1$ and $G_2$ **cross in a local overlap** $G$ if one of the following conditions hold.

- $f_1(e_{s-1}) = -f_1(e_t)$ if $s > 1$, $t < d$
- $f_1(e_{s-1}) = f_2(e'_{t'})$ if $s > 1$, $t < d$, $s' = 1$, $t' < d'$

**Example**

$G_1$ and $G_2$ cross at the overlap $G$. 
Definition
We say that $G_1$ and $G_2$ **cross in a local overlap** $G$ if one of the following conditions hold.

- $f_1(e_{s-1}) = -f_1(e_t)$ if $s > 1$, $t < d$
- $f_1(e_{s-1}) = f_2(e'_{t'})$ if $s > 1$, $t < d$, $s' = 1$, $t' < d'$

Example
$G_1$ and $G_2$ cross at the overlap $G$. 

![Diagram showing crossing of $G_1$ and $G_2$ at overlap $G$.](image)
Crossing

Definition
We say that \( G_1 \) and \( G_2 \) cross in a local overlap \( G \) if one of the following conditions hold.

- \( f_1(e_{s-1}) = -f_1(e_t) \) if \( s > 1, \ t < d \)
- \( f_1(e_{s-1}) = f_2(e'_{t'}) \) if \( s > 1, \ t < d, \ s' = 1, \ t' < d' \)

Example
\( G_1 \) and \( G_2 \) cross at the overlap \( G \).
**Definition**

We say that $G_1$ and $G_2$ **cross in a local overlap** $G$ if one of the following conditions hold.

- $f_1(e_{s-1}) = -f_1(e_t)$ if $s > 1$, $t < d$
- $f_1(e_{s-1}) = f_2(e'_{t'})$ if $s > 1$, $t < d$, $s' = 1$, $t' < d'$

**Example**

$G_1$ and $G_2$ cross at the overlap $G$. 

![Crossing Diagram]
Crossing

Definition
We say that $G_1$ and $G_2$ cross in a local overlap $G$ if one of the following conditions hold.

- $f_1(e_{s-1}) = -f_1(e_t)$ if $s > 1$, $t < d$
- $f_1(e_{s-1}) = f_2(e_{t'})$ if $s > 1$, $t < d$, $s' = 1$, $t' < d'$

Example
$G_1$ and $G_2$ cross at the overlap $G$. 
Example: Resolution $\text{Res}_G(G_1, G_2)$
Example: Resolution $\text{Res}_G(G_1, G_2)$
Example: Resolution $\text{Res}_{\mathcal{G}}(\mathcal{G}_1, \mathcal{G}_2)$
Example: Resolution $\text{Res}_G(G_1, G_2)$

$G_1$

$G_2$

$G_3$
Example: Resolution $\text{Res}_G(G_1, G_2)$
Example: Resolution $\text{Res}_G(G_1, G_2)$
Example: Resolution $\text{Res}_G(G_1, G_2)$
Example: Resolution $\text{Res}_G(G_1, G_2)$
Example: Resolution $\text{Res}_G(G_1, G_2)$
Example: Resolution $\text{Res}_G(G_1, G_2)$
Example: Resolution (Continued)
Example: Resolution (Continued)
Example: Resolution (Continued)
Resolution: Definition

**Assumption:** We will assume that $s > 1$, $t < d$, $s' = 1$ and $t' < d'$. For all other cases, see [CS].

We define four connected snake graphs as follows.

- $G_3 = G_1[1, t] \cup G_2[t' + 1, d']$,
- $G_4 = G_2[1, t'] \cup G_1[t + 1, d]$,
- $G_5 = G_1[1, k]$ where $k < s - 1$ is the largest integer such that the sign on the interior edge between tiles $k$ and $k + 1$ is the same as the sign on the interior edge of tiles $s - 1$ and $s$,
- $G_6 = G_2[d', t' + 1] \cup G_1[t + 1, d]$ where the two subgraphs are glued along the south $G_2[t+1]$ and the north of $G_1[t'+1]$ if $G_2[t+1]$ is north of $G_1$ in $G$.

**Definition**

The resolution of the crossing of $G_1$ and $G_2$ in $G$ is defined to be $(G_3 \sqcup G_4, G_5 \sqcup G_6)$ and is denoted by $\text{Res}_G(G_1, G_2)$. 
Resolution: Definition

**Assumption:** We will assume that \( s > 1, \ t < d, \ s' = 1 \) and \( t' < d' \). For all other cases, see [CS].

We define four connected snakegraphs as follows.

- \( \mathcal{G}_3 = \mathcal{G}_1[1, t] \cup \mathcal{G}_2[t' + 1, d'] \),
- \( \mathcal{G}_4 = \mathcal{G}_2[1, t'] \cup \mathcal{G}_1[t + 1, d] \),
- \( \mathcal{G}_5 = \mathcal{G}_1[1, k] \) where \( k < s - 1 \) is the largest integer such that the sign on the interior edge between tiles \( k \) and \( k + 1 \) is the same as the sign on the interior edge of tiles \( s - 1 \) and \( s \),
- \( \mathcal{G}_6 = \overline{\mathcal{G}}_2[d', t' + 1] \cup \mathcal{G}_1[t + 1, d] \) where the two subgraphs are glued along the south \( \mathcal{G}_{t+1} \) and the north of \( \mathcal{G}'_{t'+1} \) if \( \mathcal{G}_{t+1} \) is north of \( \mathcal{G}_t \) in \( \mathcal{G}_1 \).

**Definition**

The resolution of the crossing of \( \mathcal{G}_1 \) and \( \mathcal{G}_2 \) in \( \mathcal{G} \) is defined to be \( (\mathcal{G}_3 \sqcup \mathcal{G}_4, \mathcal{G}_5 \sqcup \mathcal{G}_6) \) and is denoted by \( \text{Res}_g(\mathcal{G}_1, \mathcal{G}_2) \).
Assumption: We will assume that $s > 1$, $t < d$, $s' = 1$ and $t' < d'$. For all other cases, see [CS].

We define four connected snake graphs as follows.

- $G_3 = G_1[1, t] \cup G_2[t' + 1, d']$,
- $G_4 = G_2[1, t'] \cup G_1[t + 1, d]$,
- $G_5 = G_1[1, k]$ where $k < s - 1$ is the largest integer such that the sign on the interior edge between tiles $k$ and $k + 1$ is the same as the sign on the interior edge of tiles $s - 1$ and $s$,
- $G_6 = G_2[d', t' + 1] \cup G_1[t + 1, d]$ where the two subgraphs are glued along the south $G_{t+1}$ and the north of $G'_{t'+1}$ if $G_{t+1}$ is north of $G_t$ in $G_1$.

Definition
The resolution of the crossing of $G_1$ and $G_2$ in $G$ is defined to be $(G_3 \sqcup G_4, G_5 \sqcup G_6)$ and is denoted by $\text{Res}_G(G_1, G_2)$. 
Bijection of Perfect Matchings

- Let $\text{Match}(G)$ denote the set of all perfect matchings of the graph $G$ and
  $\text{Match}(\text{Res}_G(G_1, G_2)) = \text{Match}(G_3 \sqcup G_4) \cup \text{Match}(G_5 \sqcup G_6)$.

**Theorem (CS)**

Let $G_1, G_2$ be two snake graphs. Then there is a bijection

\[ \text{Match}(G_1 \sqcup G_2) \rightarrow \text{Match}(\text{Res}_G(G_1, G_2)) \]

- Note that we construct the bijection map and its inverse map explicitly.
Bijection of Perfect Matchings

- Let $\text{Match}(G)$ denote the set of all perfect matchings of the graph $G$ and
  $\text{Match}(\text{Res}_G(G_1, G_2)) = \text{Match}(G_3 \sqcup G_4) \cup \text{Match}(G_5 \sqcup G_6)$.

Theorem (CS)

Let $G_1, G_2$ be two snake graphs. Then there is a bijection

$$\text{Match}(G_1 \sqcup G_2) \longrightarrow \text{Match}(\text{Res}_G(G_1, G_2))$$

- Note that we construct the bijection map and its inverse map explicitly.
Bijection of Perfect Matchings

- Let \( \text{Match}(G) \) denote the set of all perfect matchings of the graph \( G \) and
  \[
  \text{Match}(\text{Res}_G(G_1, G_2)) = \text{Match}(G_3 \sqcup G_4) \cup \text{Match}(G_5 \sqcup G_6).
  \]

**Theorem (CS)**

Let \( G_1, G_2 \) be two snake graphs. Then there is a bijection

\[
\text{Match}(G_1 \sqcup G_2) \longrightarrow \text{Match}(\text{Res}_G(G_1, G_2))
\]

- Note that we construct the bijection map and its inverse map explicitly.
Bijection of Perfect Matchings

- Let Match($G$) denote the set of all perfect matchings of the graph $G$ and
  $\text{Match}(\text{Res}_G(G_1, G_2)) = \text{Match}(G_3 \sqcup G_4) \cup \text{Match}(G_5 \sqcup G_6)$.

**Theorem (CS)**

*Let $G_1, G_2$ be two snake graphs. Then there is a bijection

$$\text{Match}(G_1 \sqcup G_2) \longrightarrow \text{Match}(\text{Res}_G(G_1, G_2))$$

- Note that we construct the bijection map and its inverse map explicitly.*
'Idea' of proof

\[ G_1 \]

\[ G_2 \]

\[ G_3 \]

\[ G_4 \]

\[ G_5 \]

\[ G_6 \]
'Idea' of proof
'Idea' of proof

\[ G_1 \]
\[ G_2 \]
\[ G_3 \]
\[ G_4 \]
\[ G_5 \]
\[ G_6 \]
'Idea' of proof

\[ G_1 \]

\[ G_2 \]

\[ G_3 \]

\[ G_4 \]

\[ G_5 \]

\[ G_6 \]
'Idea' of proof
On surface cluster algebras

Surface cluster algebras

Abstract Snake Graphs

Relation to Cluster Algebras

Self-crossing snake graphs

Application

I like Čanakçı (U. Leicester)

On surface cluster algebras

Geometry Seminar (U. Bath) 26 / 35

‘Idea’ of proof

$G_1$ $G_2$

$G_3$ $G_4$ $G_6$

$G_5$
‘Idea’ of proof
‘Idea’ of proof
‘Idea’ of proof

\[ G_1 \]

\[ G_2 \]

\[ G_3 \]

\[ G_4 \]

\[ G_5 \]

\[ G_6 \]
On surface cluster algebras

İlke Çanakçı

Surface cluster algebras
Abstract Snake Graphs
Relation to Cluster Algebras
Self-crossing snake graphs
Application

‘Idea’ of proof

$G_1$ $G_2$ $G_3$ $G_4$ $G_5$ $G_6$
On surface cluster algebras

İlke Çanakçı

Surface cluster algebras

Abstract Snake Graphs

Relation to Cluster Algebras

Self-crossing snake graphs

Application

‘Idea’ of proof

\[ G_1 \]

\[ G_2 \]

\[ G_3 \]

\[ G_4 \]

\[ G_5 \]

\[ G_6 \]
On surface cluster algebras

˙Ilke Çanakçi

Surface cluster algebras

Abstract Snake Graphs

Relation to Cluster Algebras

Self-crossing snake graphs

Application

‘Idea’ of proof

\[ G_1 \]

\[ G_2 \]

\[ G_3 \]

\[ G_4 \]

\[ G_5 \]

\[ G_6 \]
I](c)Ilke Canakci (U. Leicester)

On surface cluster algebras

Abstract Snake Graphs

Relation to Cluster Algebras

Self-crossing snake graphs

Application

‘Idea’ of proof
On surface cluster algebras

Ilke Cənəkçı

Surface cluster algebras

Abstract Snake Graphs

Relation to Cluster Algebras

Self-crossing snake graphs

Application

'Idea' of proof

\[ G_1 \]

\[ G_2 \]

\[ G_3 \]

\[ G_4 \]

\[ G_5 \]

\[ G_6 \]
On surface cluster algebras

İlke Çanakçı

Surface cluster algebras

Abstract Snake Graphs

Relation to Cluster Algebras

Self-crossing snake graphs

Application

‘Idea’ of proof

\[ G_1 \]

\[ G_2 \]

\[ G_3 \]

\[ G_4 \]

\[ G_5 \]

\[ G_6 \]
Surface cluster algebras

Abstract Snake Graphs

Relation to Cluster Algebras

Self-crossing snake graphs

Application

‘Idea’ of proof
‘Idea’ of proof
Surface Example
Surface Example
Surface Example
On surface cluster algebras

İlke Çanakçı

Surface cluster algebras
Abstract Snake Graphs
Relation to Cluster Algebras
Self-crossing snake graphs
Application

Surface Example

\[ G_1 \]

\[ G_2 \]
Surface Example
Surface Example
Surface Example
Surface Example
Surface Example

Surface cluster algebras

Abstract Snake Graphs

Relation to Cluster Algebras

Self-crossing snake graphs

Application
Surface Example

$\mathcal{G}_3$

$\mathcal{G}_4$

$\mathcal{G}_5$

$\mathcal{G}_6$
Surface Example

$\gamma_1$

$G_1$

$G_2$

$G_3$

$G_4$

$G_5$

$G_6$
Surface Example
Relation to Cluster Algebras

Let $\gamma_1$ and $\gamma_2$ be two arcs and $G_1$ and $G_2$ their corresponding snake graphs.

**Theorem (CS)**

$\gamma_1$ and $\gamma_2$ cross if and only if $G_1$ and $G_2$ cross as snake graphs.

**Theorem (CS)**

If $\gamma_1$ and $\gamma_2$ cross, then the snake graphs of the four arcs obtained by smoothing the crossing are given by the resolution $\text{Res}_G(G_1, G_2)$ of the crossing of the snake graphs $G_1$ and $G_2$ at the overlap $G$.

**Remark**

We do not assume that $\gamma_1$ and $\gamma_2$ cross only once. If the arcs cross multiple times the theorem can be used to resolve any of the crossings.
Relation to Cluster Algebras

Let $\gamma_1$ and $\gamma_2$ be two arcs and $G_1$ and $G_2$ their corresponding snake graphs.

**Theorem (CS)**

$\gamma_1$ and $\gamma_2$ cross if and only if $G_1$ and $G_2$ cross as snake graphs.

**Theorem (CS)**

If $\gamma_1$ and $\gamma_2$ cross, then the snake graphs of the four arcs obtained by smoothing the crossing are given by the resolution $\text{Res}_g(G_1, G_2)$ of the crossing of the snake graphs $G_1$ and $G_2$ at the overlap $G$.

**Remark**

We do not assume that $\gamma_1$ and $\gamma_2$ cross only once. If the arcs cross multiple times the theorem can be used to resolve any of the crossings.
Let $\gamma_1$ and $\gamma_2$ be two arcs and $G_1$ and $G_2$ their corresponding snake graphs.

**Theorem (CS)**

$\gamma_1$ and $\gamma_2$ cross if and only if $G_1$ and $G_2$ cross as snake graphs.

**Theorem (CS)**

If $\gamma_1$ and $\gamma_2$ cross, then the snake graphs of the four arcs obtained by smoothing the crossing are given by the resolution $\text{Res}_G(G_1, G_2)$ of the crossing of the snake graphs $G_1$ and $G_2$ at the overlap $G$.

**Remark**

We do not assume that $\gamma_1$ and $\gamma_2$ cross only once. If the arcs cross multiple times the theorem can be used to resolve any of the crossings.
 Relation to Cluster Algebras

Let $\gamma_1$ and $\gamma_2$ be two arcs and $G_1$ and $G_2$ their corresponding snake graphs.

**Theorem (CS)**

$\gamma_1$ and $\gamma_2$ cross if and only if $G_1$ and $G_2$ cross as snake graphs.

**Theorem (CS)**

If $\gamma_1$ and $\gamma_2$ cross, then the snake graphs of the four arcs obtained by smoothing the crossing are given by the resolution $\text{Res}_G(G_1, G_2)$ of the crossing of the snake graphs $G_1$ and $G_2$ at the overlap $G$.

**Remark**

We do not assume that $\gamma_1$ and $\gamma_2$ cross only once. If the arcs cross multiple times the theorem can be used to resolve any of the crossings.
Skein Relations

As a corollary we obtain a new proof of the skein relations [MW].

Corollary (CS)

Let $\gamma_1$ and $\gamma_2$ be two arcs which cross and let $(\gamma_3, \gamma_4)$ and $(\gamma_5, \gamma_6)$ be the two pairs of arcs obtained by smoothing the crossing. Then

$$x_{\gamma_1} x_{\gamma_2} = x_{\gamma_3} x_{\gamma_4} + y(\check{G}) x_{\gamma_5} x_{\gamma_6}$$

where $\check{G} = (G_3 \cup G_4) \setminus (G_5 \cup G_6)$ and $y(\check{G}) = \prod_{G_i \text{ a tile in } \check{G}} y_i$.

Remark

- Note that Musiker and Williams in [MW] use hyperbolic geometry to prove the skein relations.
- Our proof is purely combinatorial. The key ingredient to our proof is Theorem 17 where we show the bijection between the perfect matchings.
Skein Relations

As a corollary we obtain a new proof of the skein relations [MW].

Corollary (CS)

Let $\gamma_1$ and $\gamma_2$ be two arcs which cross and let $(\gamma_3, \gamma_4)$ and $(\gamma_5, \gamma_6)$ be the two pairs of arcs obtained by smoothing the crossing. Then

$$x_{\gamma_1} x_{\gamma_2} = x_{\gamma_3} x_{\gamma_4} + y(\tilde{G}) x_{\gamma_5} x_{\gamma_6}$$

where $\tilde{G} = (G_3 \cup G_4) \setminus (G_5 \cup G_6)$ and $y(\tilde{G}) = \prod_{G_i \text{ a tile in } \tilde{G}} y_i$.

Remark

- Note that Musiker and Williams in [MW] use hyperbolic geometry to prove the skein relations.
- Our proof is purely combinatorial. The key ingredient to our proof is Theorem 17 where we show the bijection between the perfect matchings.
Skein Relations

As a corollary we obtain a new proof of the skein relations [MW].

Corollary (CS)

Let $\gamma_1$ and $\gamma_2$ be two arcs which cross and let $(\gamma_3, \gamma_4)$ and $(\gamma_5, \gamma_6)$ be the two pairs of arcs obtained by smoothing the crossing. Then

$$x_{\gamma_1} x_{\gamma_2} = x_{\gamma_3} x_{\gamma_4} + y(\tilde{G}) x_{\gamma_5} x_{\gamma_6}$$

where $\tilde{G} = (G_3 \cup G_4) \setminus (G_5 \cup G_6)$ and $y(\tilde{G}) = \prod_{G_i \text{ a tile in } \tilde{G}} y_i$.

Remark

- Note that Musiker and Williams in [MW] use hyperbolic geometry to prove the skein relations.
- Our proof is purely combinatorial. The key ingredient to our proof is Theorem 17 where we show the bijection between the perfect matchings.
Skein Relations

As a corollary we obtain a new proof of the skein relations [MW].

Corollary (CS)

Let $\gamma_1$ and $\gamma_2$ be two arcs which cross and let $(\gamma_3, \gamma_4)$ and $(\gamma_5, \gamma_6)$ be the two pairs of arcs obtained by smoothing the crossing. Then

$$x_{\gamma_1} x_{\gamma_2} = x_{\gamma_3} x_{\gamma_4} + y(\tilde{G}) x_{\gamma_5} x_{\gamma_6}$$

where $\tilde{G} = (G_3 \cup G_4) \setminus (G_5 \cup G_6)$ and $y(\tilde{G}) = \prod_{G_i \text{ a tile in } \tilde{G}} y_i$.

Remark

- Note that Musiker and Williams in [MW] use hyperbolic geometry to prove the skein relations.

- Our proof is purely combinatorial. The key ingredient to our proof is Theorem 17 where we show the bijection between the perfect matchings.
Skein Relations

As a corollary we obtain a new proof of the skein relations [MW].

Corollary (CS)

Let $\gamma_1$ and $\gamma_2$ be two arcs which cross and let $(\gamma_3, \gamma_4)$ and $(\gamma_5, \gamma_6)$ be the two pairs of arcs obtained by smoothing the crossing. Then

$$x_{\gamma_1} x_{\gamma_2} = x_{\gamma_3} x_{\gamma_4} + y(\tilde{G}) x_{\gamma_5} x_{\gamma_6}$$

where $\tilde{G} = (G_3 \cup G_4) \setminus (G_5 \cup G_6)$ and $y(\tilde{G}) = \prod_{G_i \text{ a tile in } \tilde{G}} y_i$.

Remark

- Note that Musiker and Williams in [MW] use hyperbolic geometry to prove the skein relations.
- Our proof is purely combinatorial. The key ingredient to our proof is Theorem 17 where we show the bijection between the perfect matchings.
Self-crossing snake graphs and band graphs

- Self-crossing arcs and closed loops appear naturally in the process of smoothing crossings. Consider the following example.

Example

In this example we resolve two crossings of the following arcs.
Self-crossing snake graphs and band graphs

- Self-crossing arcs and closed loops appear naturally in the process of smoothing crossings. Consider the following example.

Example
In this example we resolve two crossings of the following arcs.

Example (Band graph)
Self-crossing snake graphs and band graphs

- Self-crossing arcs and closed loops appear naturally in the process of smoothing crossings. Consider the following example.

**Example**
In this example we resolve two crossings of the following arcs.

**Example (Band graph)**
Self-crossing snake graphs and band graphs

- Self-crossing arcs and closed loops appear naturally in the process of smoothing crossings. Consider the following example.

Example

In this example we resolve two crossings of the following arcs.

Example (Band graph)
Self-crossing snake graphs and band graphs

- Self-crossing arcs and closed loops appear naturally in the process of smoothing crossings. Consider the following example.

**Example**

In this example we resolve two crossings of the following arcs.

Example (Band graph)
Self-crossing snake graphs and band graphs

- Self-crossing arcs and closed loops appear naturally in the process of smoothing crossings. Consider the following example.

Example

In this example we resolve two crossings of the following arcs.

Example (Band graph)
Self-crossing snake graphs

- Similar to the definition of a local overlap for two snake graphs, we define the notion of **self-overlap** for abstract snake graphs. Here we have two subcases.
  - Self-overlap in the same direction
  - without intersection
  - with intersection
  - Self-overlap in the opposite direction
- We then define what it means for a snake graph to self-cross in a self-overlap.
- We give the resolution of a self-crossing snake graph which consists of two snake graphs and a band graph.
- Finally, we show a bijection between perfect matchings of a self-crossing snake graph with perfect matchings of its resolution.
Self-crossing snake graphs

- Similar to the definition of a local overlap for two snake graphs, we define the notion of **self-overlap** for abstract snake graphs. Here we have two subcases.
  - Self-overlap in the same direction
    - without intersection
    - with intersection
  - Self-overlap in the opposite direction

We then define what it means for a snake graph to self-cross in a self-overlap.

We give the resolution of a self-crossing snake graph which consists of two snake graphs and a band graph.

Finally, we show a bijection between perfect matchings of a self-crossing snake graph and perfect matchings of its resolution.
Self-crossing snake graphs

• Similar to the definition of a local overlap for two snake graphs, we define the notion of **self-overlap** for abstract snake graphs. Here we have two subcases.
  
  • Self-overlap in the same direction
    
    • without intersection
    
    • with intersection
  
  • Self-overlap in the opposite direction

• We then define what it means for a snake graph to self-cross in a self-overlap.

• We give the resolution of a self-crossing snake graph which consists of two snake graphs and a band graph.

• Finally, we show a bijection between perfect matchings of a self-crossing snake graph with perfect matchings of its resolution.
Self-crossing snake graphs

• Similar to the definition of a local overlap for two snake graphs, we define the notion of self-overlap for abstract snake graphs. Here we have two subcases.
  • Self-overlap in the same direction
    • without intersection
    • with intersection
  • Self-overlap in the opposite direction
• We then define what it means for a snake graph to self-cross in a self-overlap.
• We give the resolution of a self-crossing snake graph which consists of two snake graphs and a band graph.
• Finally, we show a bijection between perfect matchings of a self-crossing snake graph with perfect matchings of its resolution.
Self-crossing snake graphs

- Similar to the definition of a local overlap for two snake graphs, we define the notion of **self-overlap** for abstract snake graphs. Here we have two subcases.
  - Self-overlap in the same direction
    - without intersection
    - with intersection
  - Self-overlap in the opposite direction

- We then define what it means for a snake graph to **self-cross** in a self-overlap.

- We give the **resolution** of a self-crossing snake graph which consists of two snake graphs and a band graph.

- Finally, we show a **bijection** between perfect matchings of a self-crossing snake graph with perfect matchings of its resolution.
Self-crossing snake graphs

• Similar to the definition of a local overlap for two snake graphs, we define the notion of **self-overlap** for abstract snake graphs. Here we have two subcases.
  • Self-overlap in the same direction
    • without intersection
    • with intersection
  • Self-overlap in the opposite direction

• We then define what it means for a snake graph to **self-cross** in a self-overlap.

• We give the **resolution** of a self-crossing snake graph which consists of **two snake graphs** and a **band graph**.

• Finally, we show a **bijection** between perfect matchings of a self-crossing snake graph with perfect matchings of its resolution.
Self-crossing snake graphs

• Similar to the definition of a local overlap for two snake graphs, we define the notion of **self-overlap** for abstract snake graphs. Here we have two subcases.
  • Self-overlap in the same direction
    • without intersection
    • with intersection
  • Self-overlap in the opposite direction

• We then define what it means for a snake graph to **self-cross** in a self-overlap.

• We give the **resolution** of a self-crossing snake graph which consists of two snake graphs and a **band graph**.

• Finally, we show a **bijection** between perfect matchings of a self-crossing snake graph with perfect matchings of its resolution.
Self-crossing snake graphs

- Similar to the definition of a local overlap for two snake graphs, we define the notion of **self-overlap** for abstract snake graphs. Here we have two subcases.
  - Self-overlap in the same direction
    - without intersection
    - with intersection
  - Self-overlap in the opposite direction
- We then define what it means for a snake graph to **self-cross** in a self-overlap.
- We give the **resolution** of a self-crossing snake graph which consists of **two snake graphs** and a **band graph**.
- Finally, we show a **bijection** between perfect matchings of a self-crossing snake graph with perfect matchings of its resolution.
Self-crossing snake graphs

- Similar to the definition of a local overlap for two snake graphs, we define the notion of **self-overlap** for abstract snake graphs. Here we have two subcases.
  - Self-overlap in the same direction
    - without intersection
    - with intersection
  - Self-overlap in the opposite direction
- We then define what it means for a snake graph to **self-cross** in a self-overlap.
- We give the **resolution** of a self-crossing snake graph which consists of **two snake graphs** and a **band graph**.
- Finally, we show a **bijection** between perfect matchings of a self-crossing snake graph with perfect matchings of its resolution.
Self-crossing snake graphs

- Similar to the definition of a local overlap for two snake graphs, we define the notion of **self-overlap** for abstract snake graphs. Here we have two subcases.
  - Self-overlap in the same direction
    - without intersection
    - with intersection
  - Self-overlap in the opposite direction
- We then define what it means for a snake graph to **self-cross** in a self-overlap.
- We give the **resolution** of a self-crossing snake graph which consists of **two snake graphs** and a **band graph**.
- Finally, we show a **bijection** between perfect matchings of a self-crossing snake graph with perfect matchings of its resolution.
Self-crossing snake graphs

- Similar to the definition of a local overlap for two snake graphs, we define the notion of **self-overlap** for abstract snake graphs. Here we have two subcases.
  - Self-overlap in the same direction
    - without intersection
    - with intersection
  - Self-overlap in the opposite direction
- We then define what it means for a snake graph to **self-cross** in a self-overlap.
- We give the **resolution** of a self-crossing snake graph which consists of **two snake graphs** and a **band graph**.
- Finally, we show a **bijection** between perfect matchings of a self-crossing snake graph with perfect matchings of its resolution.
Figure: Example of resolution of selfcrossing when $s' < t$ and $s = 1$ together with geometric realization on the annulus. Here the snake graph $G_{56}$ is a single edge and the corresponding arc in the surface is a boundary segment.
On surface cluster algebras

Ilke Canakci

Surface cluster algebras

Abstract Snake Graphs

Relation to Cluster Algebras

Self-crossing snake graphs

Application

Figure: Example of resolution of selfcrossing when $s' < t$ together with geometric realization on the punctured disk
Dreaded torus

Definition (Upper cluster algebra)

\[ U = \bigcap_{x \text{ seed}} \mathbb{Z}[x]. \]

Theorem (C, Kyungyong Lee, S)

Let \( A \) be the cluster algebra associated to the dreaded torus and \( U \) be its upper cluster algebra. Then \( A = U \).

Sketch of proof. By [MM], it suffices to show that three particular Laurent polynomials given by the band graphs of three loops \( X, Y, Z \) belong to the cluster algebra.
Dreaded torus

Definition (Upper cluster algebra)

\[ U = \bigcap_{x \text{ seed}} \mathbb{Z}[x]. \]

Theorem (C, Kyungyong Lee, S)

Let \( A \) be the cluster algebra associated to the dreaded torus and \( U \) be its upper cluster algebra. Then \( A = U \).

Sketch of proof. By [MM], it suffices to show that three particular Laurent polynomials given by the band graphs of three loops \( X, Y, Z \) belong to the cluster algebra.

\[ X = \]

\[ \begin{array}{c}
\text{1} \\
\text{2} \\
\text{1} \\
\text{2} \\
\text{3} \\
\text{4} \\
\text{2} \\
\text{3} \\
\end{array} \]
Dreaded torus

Definition (Upper cluster algebra)

\[ U = \bigcap_{x \text{ seed}} \mathbb{Z}[x]. \]

Theorem (C, Kyungyong Lee, S)

Let \( A \) be the cluster algebra associated to the dreaded torus and \( U \) be its upper cluster algebra. Then \( A = U \).

Sketch of proof. By [MM], it suffices to show that three particular Laurent polynomials given by the band graphs of three loops \( X, Y, Z \) belong to the cluster algebra.
Dreaded torus

Definition (Upper cluster algebra)

\[ U = \bigcap_{x \text{ seed}} \mathbb{Z}[x]. \]

Theorem (C, Kyungyong Lee, S)

Let \( A \) be the cluster algebra associated to the dreaded torus and \( U \) be its upper cluster algebra. Then \( A = U \).

Sketch of proof. By [MM], it suffices to show that three particular Laurent polynomials given by the band graphs of three loops \( X, Y, Z \) belong to the cluster algebra.
Dreaded torus

**Definition (Upper cluster algebra)**

\[ \mathcal{U} = \bigcap_{x \text{ seed}} \mathbb{Z}[x]. \]

**Theorem (C, Kyungyong Lee, S)**

Let \( A \) be the cluster algebra associated to the dreaded torus and \( \mathcal{U} \) be its upper cluster algebra. Then \( A = \mathcal{U} \).

**Sketch of proof.** By [MM], it suffices to show that three particular Laurent polynomials given by the band graphs of three loops \( X, Y, Z \) belong to the cluster algebra.
Dreaded torus

Definition (Upper cluster algebra)

\[ U = \bigcap_{x \text{ seed}} \mathbb{Z}[x]. \]

Theorem (C, Kyungyong Lee, S)

Let \( A \) be the cluster algebra associated to the dreaded torus and \( U \) be its upper cluster algebra. Then \( A = U \).

Sketch of proof. By [MM], it suffices to show that three particular Laurent polynomials given by the band graphs of three loops \( X, Y, Z \) belong to the cluster algebra.


