Posets for cluster variables in cluster algebras from surfaces

Nathan Reading
NC State University

Duke Combinatorics Seminar
September 27, 2024

Reporting on joint work with Vincent Pilaud and Sibylle Schroll

Prologue: Now I can tell nonmathematicians what I do

One (small) reason I love working with the marked surfaces model: I now have a way to tell my non-mathematician friends and family what I do.

Prologue: Now I can tell nonmathematicians what I do

One (small) reason I love working with the marked surfaces model: I now have a way to tell my non-mathematician friends and family what I do.

"I work on algebraic and geometric combinatorics, particularly the combinatorics of Coxeter groups and cluster algebras."

Prologue: Now I can tell nonmathematicians what I do

One (small) reason I love working with the marked surfaces model: I now have a way to tell my non-mathematician friends and family what I do.

~~"I work on algebraic and geometric combinatorics, particularly the combinatorics of Coxeter groups and cluster algebras."~~

Prologue: Now I can tell nonmathematicians what I do

One (small) reason I love working with the marked surfaces model: I now have a way to tell my non-mathematician friends and family what I do.

~~"I work on algebraic and geometric combinatorics, particularly the combinatorics of Coxeter groups and cluster algebras."~~

"I think about how to draw pictures to avoid having to do algebra."

Prologue: Now I can tell nonmathematicians what I do

One (small) reason I love working with the marked surfaces model: I now have a way to tell my non-mathematician friends and family what I do.

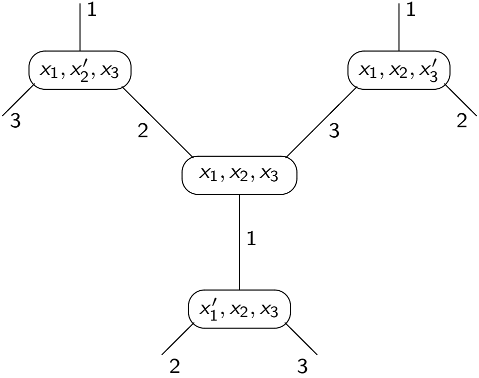
~~"I work on algebraic and geometric combinatorics, particularly the combinatorics of Coxeter groups and cluster algebras."~~

"I think about how to draw pictures to avoid having to do algebra."

I don't tell them that I use hyperbolic geometry to show that the pictures give the right answer.

Section 1: Cluster algebras

Cluster algebras (Fomin and Zelevinsky, 1999)



Cluster algebras (Fomin and Zelevinsky, 1999)

Given:

- initial cluster variables $x_1, \ldots, x_n$
- combinatorial data B

Mutation: a multidirectional recursion defining cluster variables.

Cluster algebras (Fomin and Zelevinsky, 1999)

Given:

- initial cluster variables $x_1, \ldots, x_n$
- combinatorial data B

Mutation: a multidirectional recursion defining cluster variables.

Some details:

Combinatorial data: B is a skew-symmetric integer matrix.

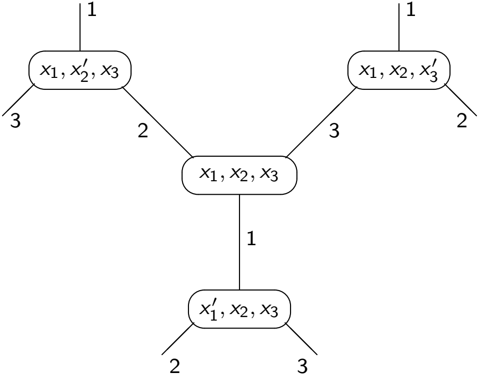
Mutation happens in n "directions". It is an involution that

- switches out one cluster variable, replaces it with a new one;
- changes B by matrix mutation.

Cluster variables are *a priori* rational functions in $x_1, \ldots, x_n$, but they turn out to be Laurent polynomials.

Collect all cluster variables coming from sequences of mutations. The cluster algebra is the ring generated by these cluster variables.

Cluster algebras (Fomin and Zelevinsky, 1999)



The mutation of B in direction k is the matrix $B' = \mu_k(B)$ with

$$b'_{ij} = \begin{cases} -b_{ij} & \text{if } k \in \{i,j\}; \\ b_{ij} + \operatorname{sgn}(b_{kj})[b_{ik}b_{kj}]_+ & \text{otherwise.} \end{cases}$$

where $[a]_+$ means $\max(a, 0)$.

Mutating the cluster variables $x_1, \ldots, x_n$ in direction k means keeping x_i for $i \neq k$ and replacing x_k by x'_k according to the exchange relations

$$x_k x'_k = \prod_{i=1}^{n} x_i^{[b_{ik}]_+} + \prod_{i=1}^{n} x_i^{[-b_{ik}]_+}.$$

Mutation example

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \overset{\mu_1}{\longleftrightarrow} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \overset{\mu_2}{\longleftrightarrow} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$\Big\updownarrow \mu_2$$

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \overset{\mu_1}{\longleftrightarrow} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \overset{\mu_2}{\longleftrightarrow} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Mutation example

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \overset{\mu_1}{\longleftrightarrow} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \overset{\mu_2}{\longleftrightarrow} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix}$$

$$\Big\updownarrow \mu_2$$

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \overset{\mu_1}{\longleftrightarrow} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \overset{\mu_2}{\longleftrightarrow} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Mutation example

$$
\begin{array}{ccccc}
\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} & \overset{\mu_1}{\longleftrightarrow} & \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} & \overset{\mu_2}{\longleftrightarrow} & \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \\
\begin{bmatrix} x_1 & x_2 \end{bmatrix} & & \begin{bmatrix} \frac{x_2+1}{x_1} & x_2 \end{bmatrix} & & \begin{bmatrix} \frac{x_2+1}{x_1} & \frac{x_1+x_2+1}{x_1 x_2} \end{bmatrix} \\
\Big\updownarrow \mu_2 & & & & \\
\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} & \overset{\mu_1}{\longleftrightarrow} & \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} & \overset{\mu_2}{\longleftrightarrow} & \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \\
\begin{bmatrix} x_1 & \frac{1+x_1}{x_2} \end{bmatrix} & & \begin{bmatrix} \frac{x_1+x_2+1}{x_1 x_2} & \frac{1+x_1}{x_2} \end{bmatrix} & & \begin{bmatrix} \frac{x_1+x_2+1}{x_1 x_2} & \frac{x_2+1}{x_1} \end{bmatrix}
\end{array}
$$

Mutation example

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \overset{\mu_1}{\longleftrightarrow} \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad \overset{\mu_2}{\longleftrightarrow} \quad \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \longleftarrow$$

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \qquad \begin{bmatrix} \frac{x_2+1}{x_1} & x_2 \end{bmatrix} \qquad \begin{bmatrix} \frac{x_2+1}{x_1} & \frac{x_1+x_2+1}{x_1 x_2} \end{bmatrix}$$

$$\Big\updownarrow \mu_2$$

identify these

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad \overset{\mu_1}{\longleftrightarrow} \quad \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \overset{\mu_2}{\longleftrightarrow} \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \longleftarrow$$

$$\begin{bmatrix} x_1 & \frac{1+x_1}{x_2} \end{bmatrix} \qquad \begin{bmatrix} \frac{x_1+x_2+1}{x_1 x_2} & \frac{1+x_1}{x_2} \end{bmatrix} \qquad \begin{bmatrix} \frac{x_1+x_2+1}{x_1 x_2} & \frac{x_2+1}{x_1} \end{bmatrix}$$

Mutation example

$$
\begin{array}{ccccc}
\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} & \stackrel{\mu_1}{\longleftrightarrow} & \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} & \stackrel{\mu_2}{\longleftrightarrow} & \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \\
\begin{bmatrix} x_1 & x_2 \end{bmatrix} & & \begin{bmatrix} \frac{x_2+1}{x_1} & x_2 \end{bmatrix} & & \begin{bmatrix} \frac{x_2+1}{x_1} & \frac{x_1+x_2+1}{x_1 x_2} \end{bmatrix} \longleftarrow
\end{array}
$$

$$
\Big\updownarrow \mu_2
$$

identify these

$$
\begin{array}{ccccc}
\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} & \stackrel{\mu_1}{\longleftrightarrow} & \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} & \stackrel{\mu_2}{\longleftrightarrow} & \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \\
\begin{bmatrix} x_1 & \frac{1+x_1}{x_2} \end{bmatrix} & & \begin{bmatrix} \frac{x_1+x_2+1}{x_1 x_2} & \frac{1+x_1}{x_2} \end{bmatrix} & & \begin{bmatrix} \frac{x_1+x_2+1}{x_1 x_2} & \frac{x_2+1}{x_1} \end{bmatrix} \longleftarrow
\end{array}
$$

Upshot: *A priori*, mutations live on an infinite path for $n = 2$.
Here, they live on a pentagon (5-cycle).

Solving the recursion that defines cluster variables

Big question: Find a formula for each cluster variable in terms of the initial cluster variables $x_1, \ldots, x_n$.

Before we can solve the recursion, we need to know the combinatorics that underlies the solution.

In the previous example: Cluster variables are indexed by vertices of a pentagon (not by vertices of an infinite path).

In general, cluster variables are indexed, *a priori* by the vertices of an (infinite!) n-regular tree. But in specific examples, the tree might collapse a polytope (or something between an infinite tree and a polytope).

Today: A special class of cluster algebra where the natural combinatorial model is triangulations of surfaces.

Solving the recursion that defines cluster variables

Big question: Find a formula for each cluster variable in terms of the initial cluster variables $x_1, \ldots, x_n$.

Before we can solve the recursion, we need to know the combinatorics that underlies the solution.

In the previous example: Cluster variables are indexed by vertices of a pentagon (not by vertices of an infinite path).

In general, cluster variables are indexed, *a priori* by the vertices of an (infinite!) n-regular tree. But in specific examples, the tree might collapse a polytope (or something between an infinite tree and a polytope).

Today: A special class of cluster algebra where the natural combinatorial model is triangulations of surfaces.

But first, why do this at all?

Why cluster algebras?

Cluster algebras were invented by Fomin and Zelevinsky to study total positivity of matrices.

Since then, cluster algebras have turned out to be connected to many areas of mathematics, including combinatorics, representation theory, topology, symplectic geometry, dynamical systems, algebraic geometry (mirror symmetry), number theory, etc.

Why cluster algebras?

Cluster algebras were invented by Fomin and Zelevinsky to study total positivity of matrices.

Since then, cluster algebras have turned out to be connected to many areas of mathematics, including combinatorics, representation theory, topology, symplectic geometry, dynamical systems, algebraic geometry (mirror symmetry), number theory, etc.

Cluster algebras are also of interest in mathematical physics...

Cluster algebras in the service of Physics

Scattering amplitude: Empirically measurable probability density in n-space, describing possible interactions between particles.

In principle, these are sums/integrals over Feynman diagrams.
In practice, the computations are often forbiddingly complex.

Cluster algebras are a machine that makes functions of $x_1, \ldots, x_n$.

Sometimes, using these functions, we can skip Feynman diagrams and directly write down formulas for scattering amplitudes.

In some cases, the building blocks of the formulas are observed to be cluster variables in some cluster algebra.

The combinatorics of clusters of cluster variables also plays a role here.

Section 2: The marked surfaces model

Triangulated marked surfaces (Fomin, Shapiro, Thurston, 2008.)

Some cluster algebras are modeled by marked surfaces.

Surface: A compact orientable surface **S**, possibly with boundary.

Marked: A collection **M** of marked points.

Marked points in the interior are called punctures.

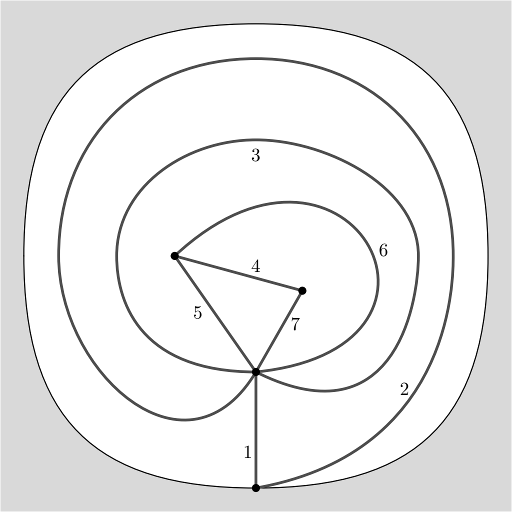
Combinatorial data: A triangulation of **S** with vertices at **M**.

This means a set T of arcs (curves connecting the marked points) that cut **S** into triangles.

From T, you can read off combinatorially, a skew-symmetric matrix B.

Flips of arcs: Remove one arc from T and insert the unique different arc that makes a triangulation.

Marked surface example

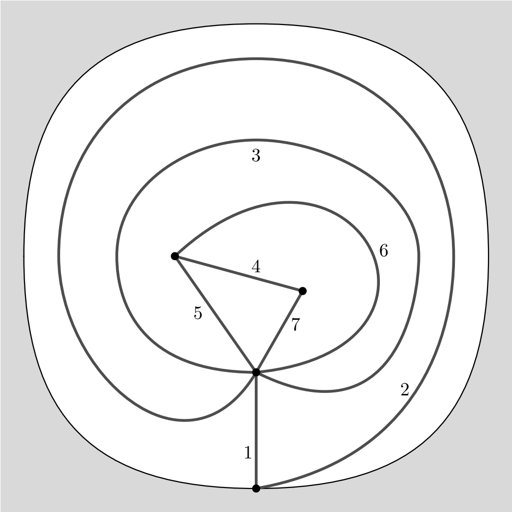


There are 4 marked points

The triangulation has 7 arcs.

Flip: Remove any arc and then fill in the other diagonal of the quadrilateral

Marked surface example



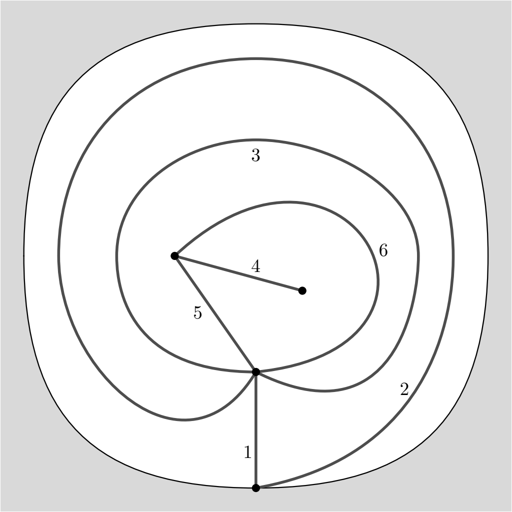
There are 4 marked points

The triangulation has 7 arcs.

Flip: Remove any arc and then fill in the other diagonal of the quadrilateral

Let's try flipping arc 7

Marked surface example



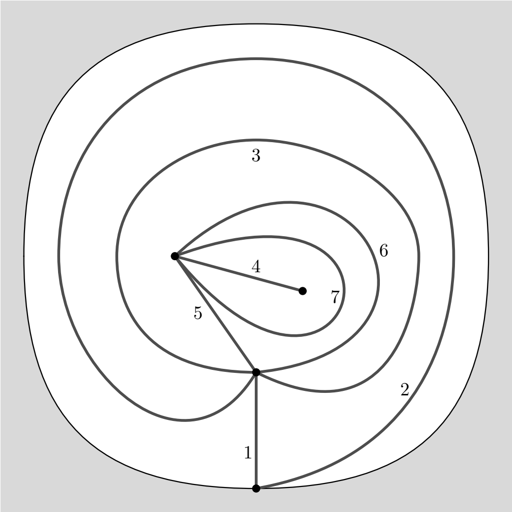
There are 4 marked points

The triangulation has 7 arcs.

Flip: Remove any arc and then fill in the other diagonal of the quadrilateral

Let's try flipping arc 7

Marked surface example



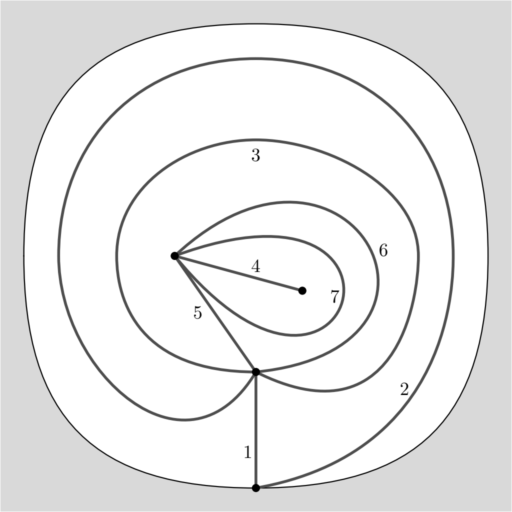
There are 4 marked points

The triangulation has 7 arcs.

Flip: Remove any arc and then fill in the other diagonal of the quadrilateral

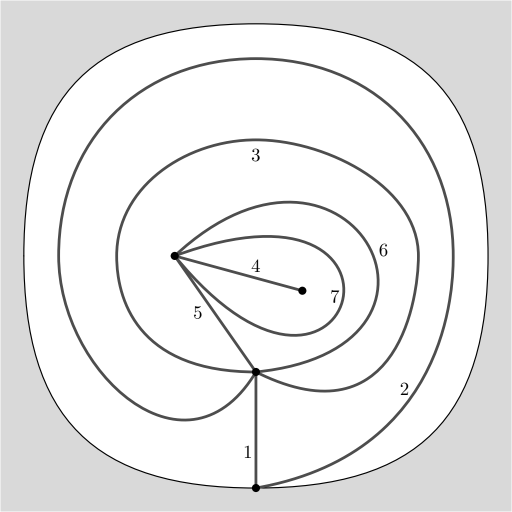
Let's try flipping arc 7

Cluster variables and hyperbolic lengths



Put a constant curvature -1 hyperbolic metric on **S**, marked points “at infinity”.

Cluster var’s: lengths of arcs.

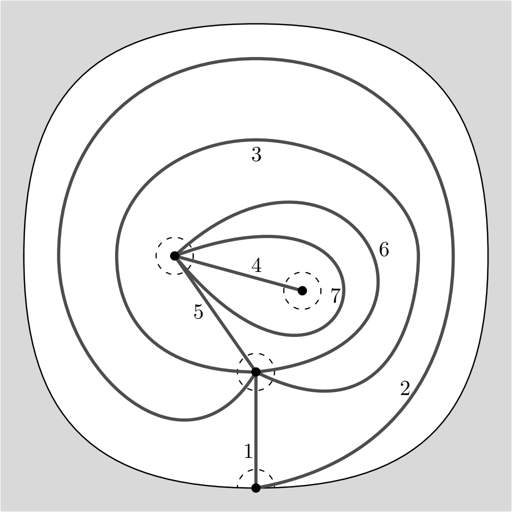


Put a constant curvature -1 hyperbolic metric on **S**, marked points "at infinity".

Cluster var's: lengths of arcs.

Problem: Arcs have infinite length.

Cluster variables and hyperbolic lengths



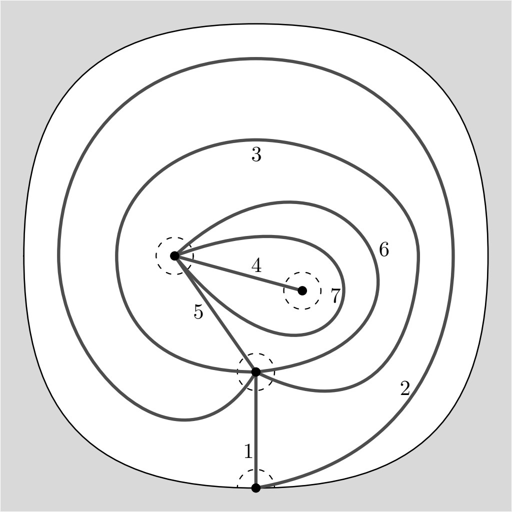
Put a constant curvature -1 hyperbolic metric on **S**, marked points "at infinity".

Cluster var's: lengths of arcs.

Problem: Arcs have infinite length.

Solution: Horocycles about marked points. Cluster variables are lengths between horocycles.

Cluster variables and hyperbolic lengths



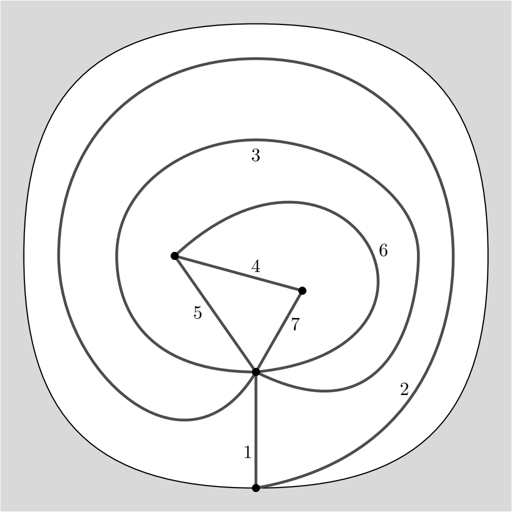
Put a constant curvature -1 hyperbolic metric on **S**, marked points "at infinity".

Cluster var's: lengths of arcs.

Problem: Arcs have infinite length.

Solution: Horocycles about marked points. Cluster variables are exponentials of lengths between horocycles.

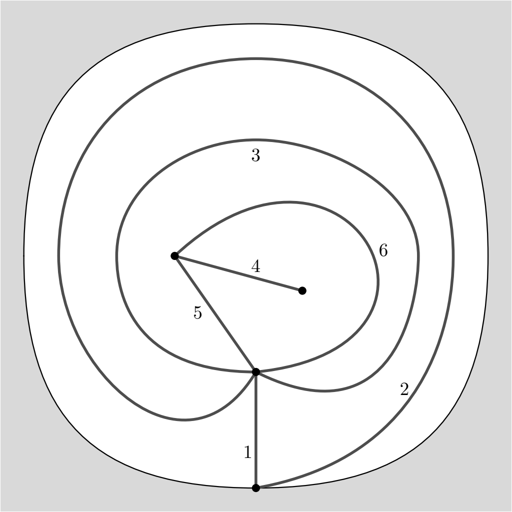
One more complication: Arcs are not enough



Problem:

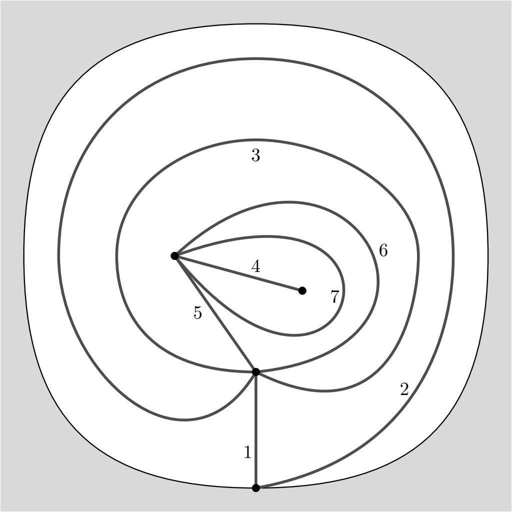
After we flip arc 7,

One more complication: Arcs are not enough



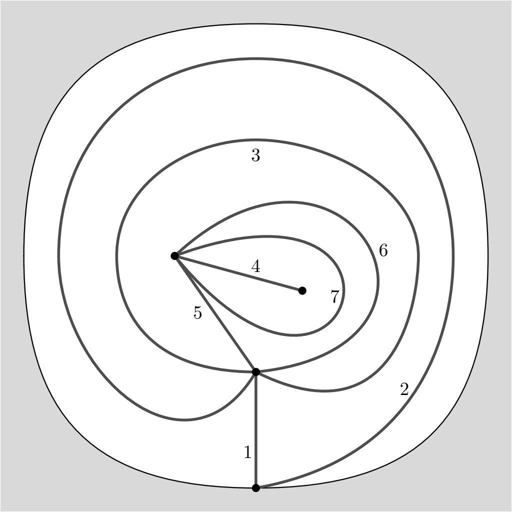
Problem:
After we flip arc 7,

One more complication: Arcs are not enough



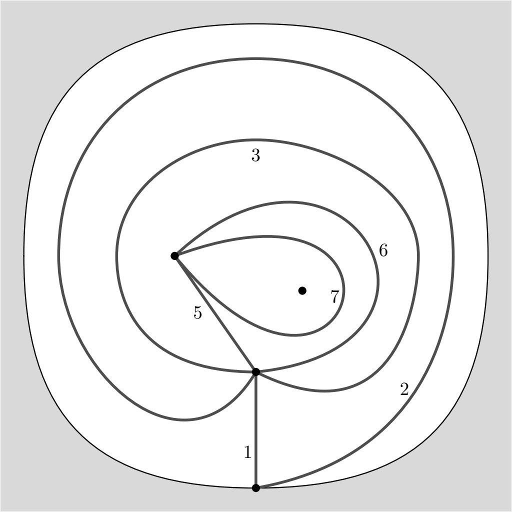
Problem:
After we flip arc 7,

One more complication: Arcs are not enough



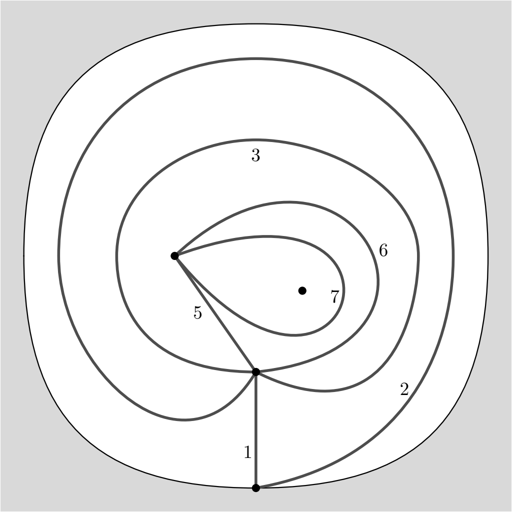
Problem:
After we flip arc 7,
we can't flip arc 4.

One more complication: Arcs are not enough



Problem:
After we flip arc 7,
we can't flip arc 4.

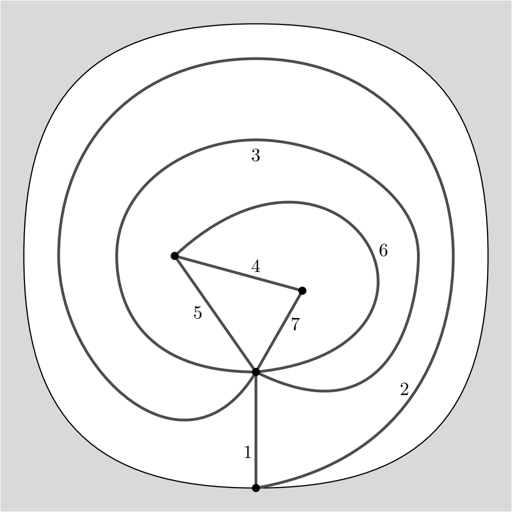
One more complication: Arcs are not enough



Problem:
After we flip arc 7,
we can't flip arc 4.

Solution:
Replace arcs that loop around a point with tagged arcs.

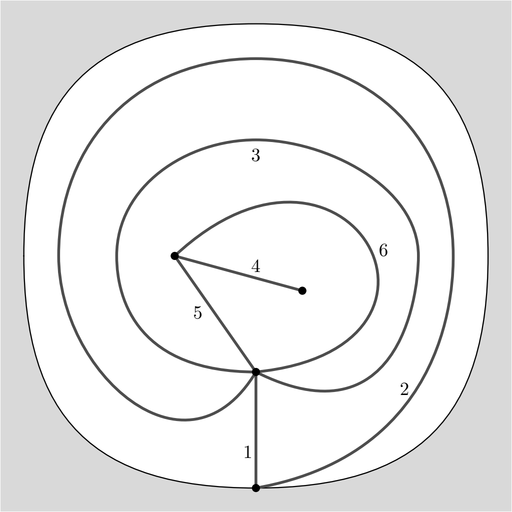
One more complication: Arcs are not enough



Problem:
After we flip arc 7,
we can't flip arc 4.

Solution:
Replace arcs that loop around a point with tagged arcs.

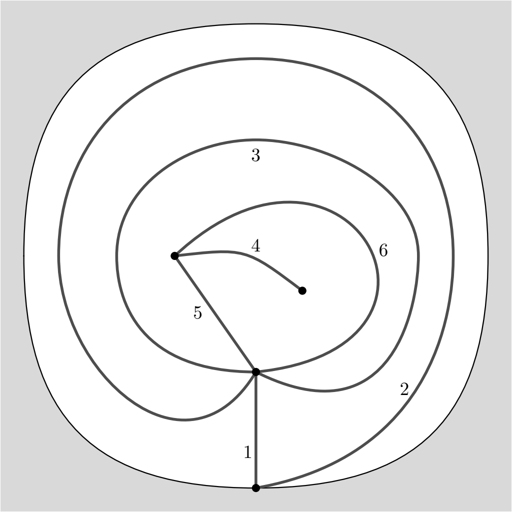
One more complication: Arcs are not enough



Problem:
After we flip arc 7,
we can't flip arc 4.

Solution:
Replace arcs that loop around a point with tagged arcs.

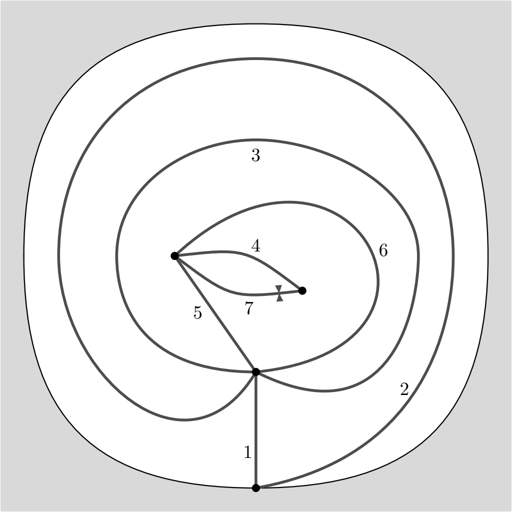
One more complication: Arcs are not enough



Problem:
After we flip arc 7,
we can't flip arc 4.

Solution:
Replace arcs that loop around a point with tagged arcs.

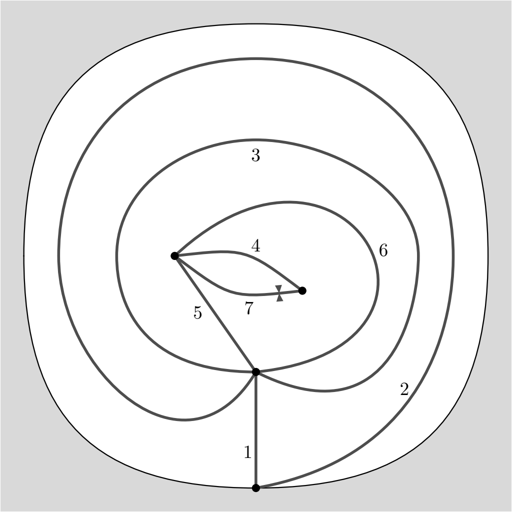
One more complication: Arcs are not enough



Problem:
After we flip arc 7, we can't flip arc 4.

Solution:
Replace arcs that loop around a point with tagged arcs.

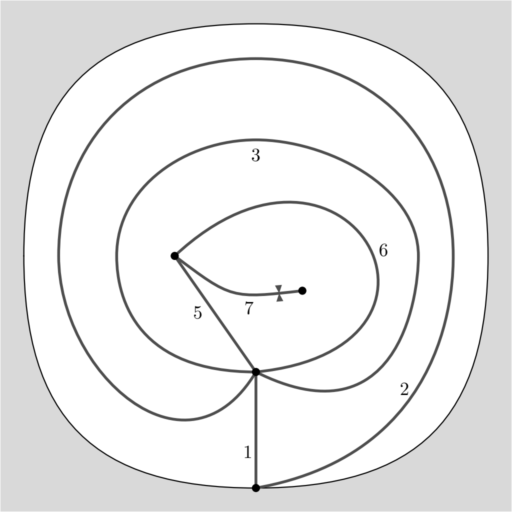
Tagged arcs



Cluster variables $\longleftrightarrow$ tagged arcs (arcs that might be notched at any puncture).

$\exists$ rules for how tagged arcs can be together in a tagged triangulation.

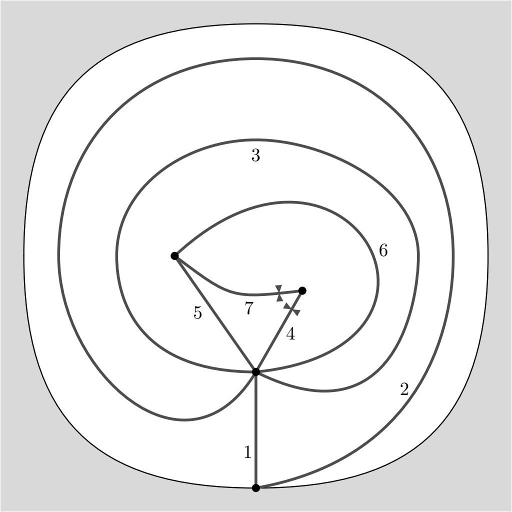
Tagged arcs



Cluster variables $\longleftrightarrow$ tagged arcs (arcs that might be notched at any puncture).

$\exists$ rules for how tagged arcs can be together in a tagged triangulation.

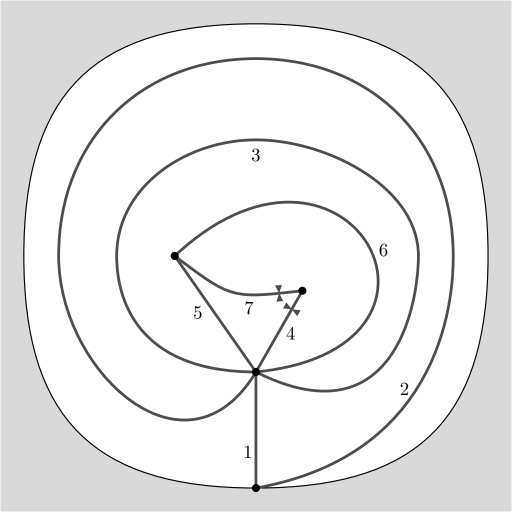
Tagged arcs



Cluster variables $\longleftrightarrow$ tagged arcs (arcs that might be notched at any puncture).

$\exists$ rules for how tagged arcs can be together in a tagged triangulation.

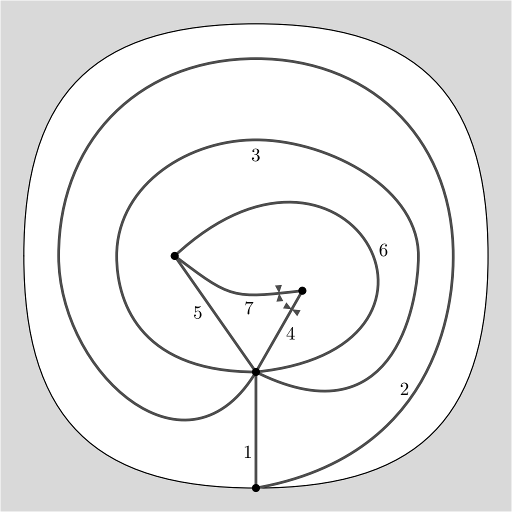
Tagged arcs



Cluster variables $\longleftrightarrow$ tagged arcs (arcs that might be notched at any puncture).

$\exists$ rules for how tagged arcs can be together in a tagged triangulation. But we won't go farther in that direction.

Tagged arcs



Cluster variables $\longleftrightarrow$ tagged arcs (arcs that might be notched at any puncture).

$\exists$ rules for how tagged arcs can be together in a tagged triangulation. But we won't go farther in that direction.

Cluster variables for tagged arcs are also hyperbolic lengths.
(At notched endpoint, measure length to the conjugate horocycle.)

Soft Reboot: Rephrase the big question as geometry

Soft Reboot: Rephrase the big question as geometry

Start with a marked surface $(\mathbf{S}, \mathbf{M})$ and a triangulation T.

Soft Reboot: Rephrase the big question as geometry

Start with a marked surface $(\mathbf{S}, \mathbf{M})$ and a triangulation T.

The lambda length of a tagged arc α is the exponential of its hyperbolic length, measured between horocycles, (using conjugate horocycles where there is a notch on α).

Soft Reboot: Rephrase the big question as geometry

Start with a marked surface $(\mathbf{S}, \mathbf{M})$ and a triangulation T.

The lambda length of a tagged arc α is the exponential of its hyperbolic length, measured between horocycles, (using conjugate horocycles where there is a notch on α).

For each arc γ in T, choose a lambda length x_γ.

Soft Reboot: Rephrase the big question as geometry

Start with a marked surface $(\mathbf{S}, \mathbf{M})$ and a triangulation T.

The lambda length of a tagged arc α is the exponential of its hyperbolic length, measured between horocycles, (using conjugate horocycles where there is a notch on α).

For each arc γ in T, choose a lambda length x_γ.

There is a unique way to put a hyperbolic metric and horocycles on $\mathbf{S}$ so that each arc $\gamma \in T$ is a geodesic with lambda length x_γ.

Soft Reboot: Rephrase the big question as geometry

Start with a marked surface $(\mathbf{S}, \mathbf{M})$ and a triangulation T.

The lambda length of a tagged arc α is the exponential of its hyperbolic length, measured between horocycles, (using conjugate horocycles where there is a notch on α).

For each arc γ in T, choose a lambda length x_γ.

There is a unique way to put a hyperbolic metric and horocycles on $\mathbf{S}$ so that each arc $\gamma \in T$ is a geodesic with lambda length x_γ.

The fine print: Constant curvature -1, marked points at infinity, and boundary segments are geodesics with lambda length 1.

Soft Reboot: Rephrase the big question as geometry

Start with a marked surface $(\mathbf{S}, \mathbf{M})$ and a triangulation T.

The lambda length of a tagged arc α is the exponential of its hyperbolic length, measured between horocycles, (using conjugate horocycles where there is a notch on α).

For each arc γ in T, choose a lambda length x_γ.

There is a unique way to put a hyperbolic metric and horocycles on $\mathbf{S}$ so that each arc $\gamma \in T$ is a geodesic with lambda length x_γ.

Soft Reboot: Rephrase the big question as geometry

Start with a marked surface $(\mathbf{S}, \mathbf{M})$ and a triangulation T.

The lambda length of a tagged arc α is the exponential of its hyperbolic length, measured between horocycles, (using conjugate horocycles where there is a notch on α).

For each arc γ in T, choose a lambda length x_γ.

There is a unique way to put a hyperbolic metric and horocycles on $\mathbf{S}$ so that each arc $\gamma \in T$ is a geodesic with lambda length x_γ.

Now, every tagged arc α has a lambda length.

Soft Reboot: Rephrase the big question as geometry

Start with a marked surface $(\mathbf{S}, \mathbf{M})$ and a triangulation T.

The lambda length of a tagged arc α is the exponential of its hyperbolic length, measured between horocycles, (using conjugate horocycles where there is a notch on α).

For each arc γ in T, choose a lambda length x_γ.

There is a unique way to put a hyperbolic metric and horocycles on $\mathbf{S}$ so that each arc $\gamma \in T$ is a geodesic with lambda length x_γ.

Now, every tagged arc α has a lambda length.

The fine print: α is isotopic to a unique (tagged) geodesic. Take the lambda length of that geodesic.

Soft Reboot: Rephrase the big question as geometry

Start with a marked surface $(\mathbf{S}, \mathbf{M})$ and a triangulation T.

The lambda length of a tagged arc α is the exponential of its hyperbolic length, measured between horocycles, (using conjugate horocycles where there is a notch on α).

For each arc γ in T, choose a lambda length x_γ.

There is a unique way to put a hyperbolic metric and horocycles on $\mathbf{S}$ so that each arc $\gamma \in T$ is a geodesic with lambda length x_γ.

Now, every tagged arc α has a lambda length.

Soft Reboot: Rephrase the big question as geometry

Start with a marked surface $(\mathbf{S}, \mathbf{M})$ and a triangulation T.

The lambda length of a tagged arc α is the exponential of its hyperbolic length, measured between horocycles, (using conjugate horocycles where there is a notch on α).

For each arc γ in T, choose a lambda length x_γ.

There is a unique way to put a hyperbolic metric and horocycles on $\mathbf{S}$ so that each arc $\gamma \in T$ is a geodesic with lambda length x_γ.

Now, every tagged arc α has a lambda length.

Big question: Find a formula for the lambda length of α in terms of the lambda lengths x_γ for $\gamma \in T$.

Soft Reboot: Rephrase the big question as geometry

Start with a marked surface $(\mathbf{S}, \mathbf{M})$ and a triangulation T.

The lambda length of a tagged arc α is the exponential of its hyperbolic length, measured between horocycles, (using conjugate horocycles where there is a notch on α).

For each arc γ in T, choose a lambda length x_γ.

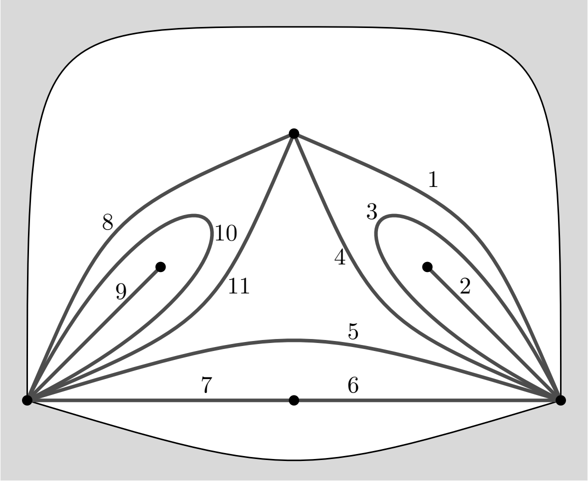
There is a unique way to put a hyperbolic metric and horocycles on $\mathbf{S}$ so that each arc $\gamma \in T$ is a geodesic with lambda length x_γ.

Now, every tagged arc α has a lambda length.

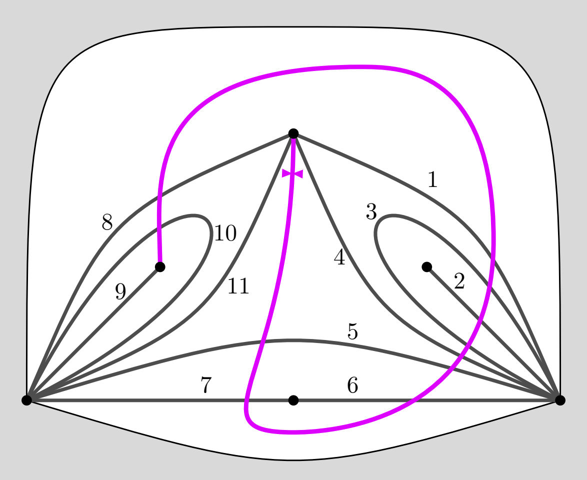
Big question: Find a formula for the lambda length of α in terms of the lambda lengths x_γ for $\gamma \in T$.

This is the same big question as before!
(Formulas for cluster variables in terms of the initial cluster.)

Example



Example



Find the lambda length of this tagged arc.

The previous* state of the art

So far, everything I've explained is work of Fomin, Shapiro, and Thurston (FST 2006, FT 2012).

The other main work on the big question for surfaces: Musiker, Schiffler, and Williams (MS 2008, MSW 2009&2011, MW 2011). They give a formula for cluster variables as a weighted sum of perfect matchings on snake graphs.

The case where α has no notches is simplest: Sum over perfect matchings, with each matching weighted by a Laurent monomial.

Our starting point is an insight about the simplest case, already in the MSW work: There is a distributive lattice structure on the set of perfect matchings of a graph (Propp 2002).

The previous* state of the art

So far, everything I've explained is work of Fomin, Shapiro, and Thurston (FST 2006, FT 2012).

The other main work on the big question for surfaces: Musiker, Schiffler, and Williams (MS 2008, MSW 2009&2011, MW 2011). They give a formula for cluster variables as a weighted sum of perfect matchings on snake graphs.

The case where α has no notches is simplest: Sum over perfect matchings, with each matching weighted by a Laurent monomial.

Our starting point is an insight about the simplest case, already in the MSW work: There is a distributive lattice structure on the set of perfect matchings of a graph (Propp 2002).

FTFDL!

Section 3: FTFDL

The Fundamental Theorem of Finite Distributive Lattices

Lattice: A partially ordered set with a meet (greatest lower bound) operation and a join (least upper bound) operation.

Distributive: Meet distributes over the join and vice versa.

Prototypical example: The lattice of all subsets of a given set, ordered by inclusion.

"meet" = intersection, "join" = union (usual distributive laws)

The Fundamental Theorem of Finite Distributive Lattices

Lattice: A partially ordered set with a meet (greatest lower bound) operation and a join (least upper bound) operation.

Distributive: Meet distributes over the join and vice versa.

Prototypical example: The lattice of all subsets of a given set, ordered by inclusion.

$$\text{“meet”} = \text{intersection}, \quad \text{“join”} = \text{union} \quad \text{(usual distributive laws)}$$

More general example: The lattice of all downsets (AKA order ideals) of a given partially ordered set (“poset”), ordered by inclusion. Again, meet is intersection and join is union.

The Fundamental Theorem of Finite Distributive Lattices

Lattice: A partially ordered set with a meet (greatest lower bound) operation and a join (least upper bound) operation.

Distributive: Meet distributes over the join and vice versa.

Prototypical example: The lattice of all subsets of a given set, ordered by inclusion.

$$\text{"meet"} = \text{intersection}, \quad \text{"join"} = \text{union} \quad (\text{usual distributive laws})$$

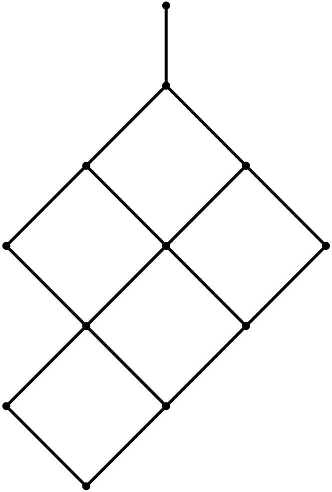
More general example: The lattice of all downsets (AKA order ideals) of a given partially ordered set ("poset"), ordered by inclusion. Again, meet is intersection and join is union.

FTFDL (Birkhoff, 1937):
A finite lattice L is distributive if and only if it is the lattice of downsets of a finite poset P. (The poset P is isomorphic to the subposet of L consisting of join-irreducible elements.)

FTFDL example

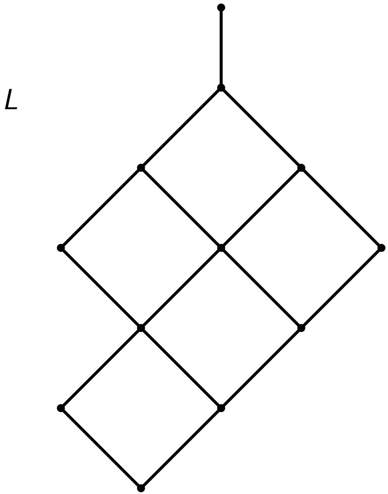
L is a distributive lattice.

L



FTFDL example

L is a distributive lattice.

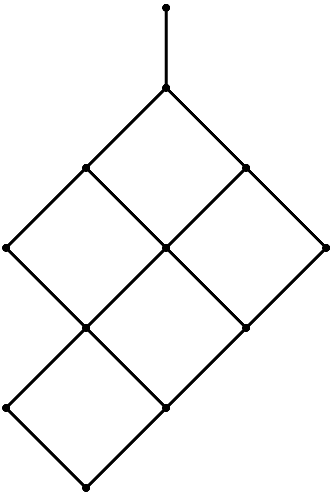


FTFDL says: There exists P such that L is the lattice of downsets of P.

FTFDL example

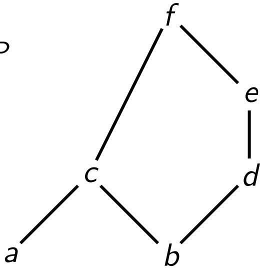
L is a distributive lattice.

L



FTFDL says: There exists *P* such that *L* is the lattice of downsets of *P*.

P

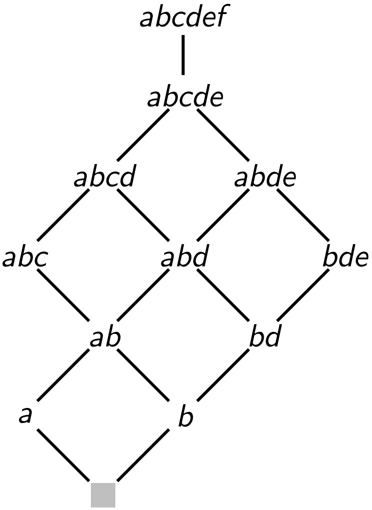


FTFDL example

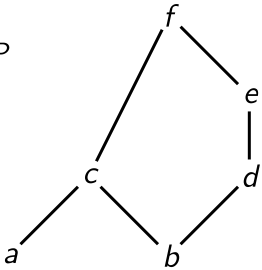
L is a distributive lattice.

FTFDL says: There exists *P* such that *L* is the lattice of downsets of *P*.

L



P



Philosophy

Philosophy

If you see a distributive lattice, you must use FTFDL.

If you see a distributive lattice, you must use FTFDL

Back to the cluster variables/hyperbolic geometry questions:

In the simplest case of the MSW work, each cluster variable is a sum of Laurent monomials indexed by the elements of a finite distributive lattice.

A finite distributive lattice is the set of downsets in a finite poset.

Q: What is the nicest possible way to get monomials from downsets in a finite poset?

If you see a distributive lattice, you must use FTFDL

Back to the cluster variables/hyperbolic geometry questions:

In the simplest case of the MSW work, each cluster variable is a sum of Laurent monomials indexed by the elements of a finite distributive lattice.

A finite distributive lattice is the set of downsets in a finite poset.

Q: What is the nicest possible way to get monomials from downsets in a finite poset?
A: Give a monomial weight to each element of the poset. The weight of a downset is the product of the weights of its elements.

If you see a distributive lattice, you must use FTFDL

Back to the cluster variables/hyperbolic geometry questions:

In the simplest case of the MSW work, each cluster variable is a sum of Laurent monomials indexed by the elements of a finite distributive lattice.

A finite distributive lattice is the set of downsets in a finite poset.

Q: What is the nicest possible way to get monomials from downsets in a finite poset?
A: Give a monomial weight to each element of the poset. The weight of a downset is the product of the weights of its elements.

Main result (with Pilaud and Schroll): A simple, combinatorial way to construct a weighted poset P_α for any tagged arc α so that the weighted sum of downsets, times a global factor, is the cluster variable (lambda length).

If you see a distributive lattice, you must use FTFDL

Back to the cluster variables/hyperbolic geometry questions:

In the simplest case of the MSW work, each cluster variable is a sum of Laurent monomials indexed by the elements of a finite distributive lattice.

A finite distributive lattice is the set of downsets in a finite poset.

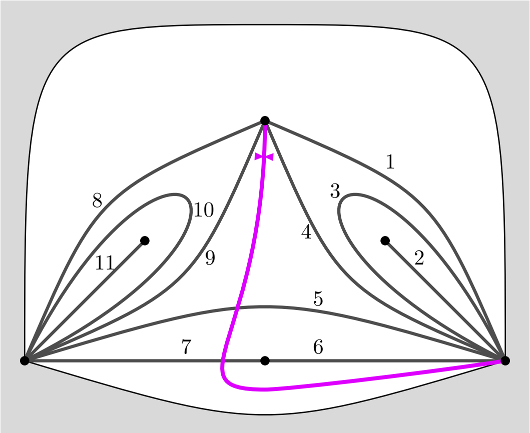
Q: What is the nicest possible way to get monomials from downsets in a finite poset?
A: Give a monomial weight to each element of the poset. The weight of a downset is the product of the weights of its elements.

Main result (with Pilaud and Schroll): A simple, combinatorial way to construct a weighted poset P_α for any tagged arc α so that the weighted sum of downsets, times a global factor, is the cluster variable (lambda length).

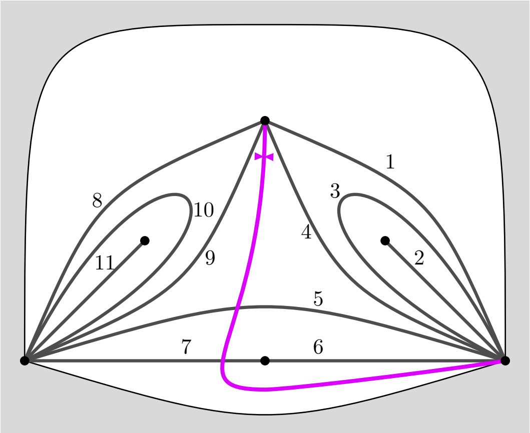
For the experts: This is the $\mathbf{g}$-vector times the F-polynomial.

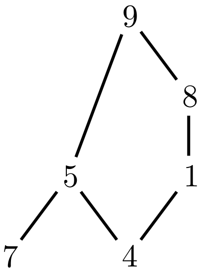
Section 4: The posets

Example (how to use the poset P_α, not yet how to make it)

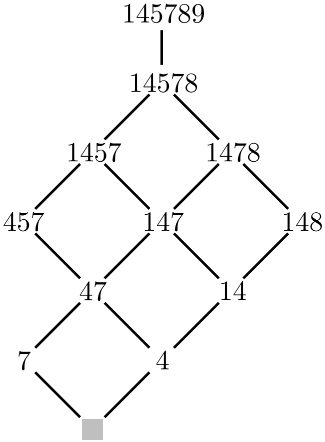


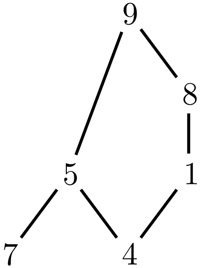
Example (how to use the poset P_α, not yet how to make it)



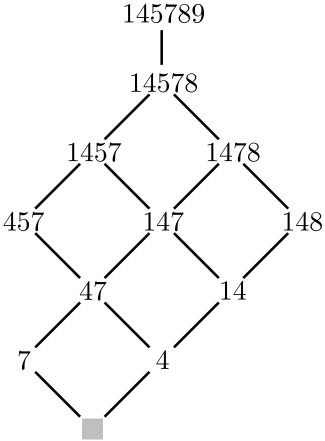


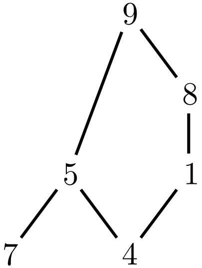
Example (how to use the poset P_α, not yet how to make it)





Example (how to use the poset P_α, not yet how to make it)

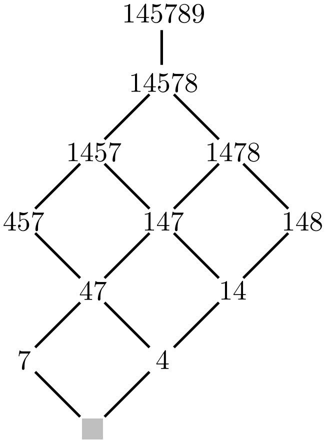




Cluster variable associated to α (i.e. lambda length of α):

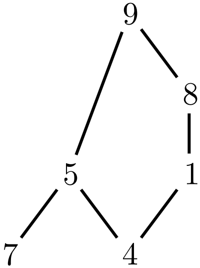
$$x_\alpha = \frac{x_5}{x_4}(1 + \hat{y}_4 + \hat{y}_7 + \hat{y}_1\hat{y}_4 + \hat{y}_4\hat{y}_7 + \hat{y}_1\hat{y}_4\hat{y}_7 + \hat{y}_4\hat{y}_5\hat{y}_7 + \hat{y}_1\hat{y}_4\hat{y}_8 + \hat{y}_1\hat{y}_4\hat{y}_5\hat{y}_7 + \hat{y}_1\hat{y}_4\hat{y}_7\hat{y}_8 + \hat{y}_1\hat{y}_4\hat{y}_5\hat{y}_7\hat{y}_8 + \hat{y}_1\hat{y}_4\hat{y}_5\hat{y}_7\hat{y}_8\hat{y}_9)$$

Example (how to use the poset P_α, not yet how to make it)



The $\hat{y}_i$ are certain monomials in the x's.

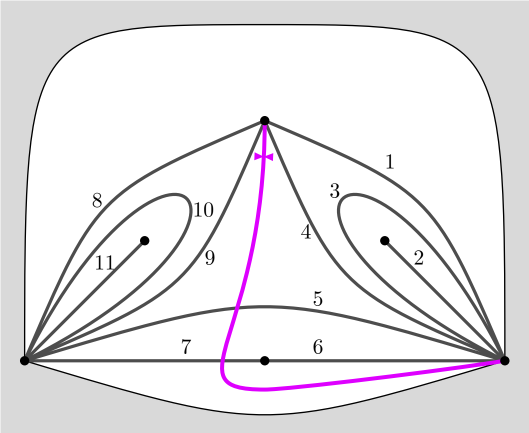
The monomial $\frac{x_5}{x_4}$ can be read off combinatorially.



Cluster variable associated to α (i.e. lambda length of α):

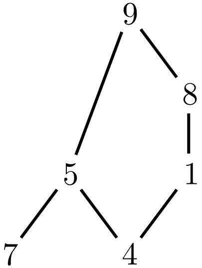
$$x_\alpha = \frac{x_5}{x_4}(1 + \hat{y}_4 + \hat{y}_7 + \hat{y}_1\hat{y}_4 + \hat{y}_4\hat{y}_7 + \hat{y}_1\hat{y}_4\hat{y}_7 + \hat{y}_4\hat{y}_5\hat{y}_7 + \hat{y}_1\hat{y}_4\hat{y}_8 + \hat{y}_1\hat{y}_4\hat{y}_5\hat{y}_7 + \hat{y}_1\hat{y}_4\hat{y}_7\hat{y}_8 + \hat{y}_1\hat{y}_4\hat{y}_5\hat{y}_7\hat{y}_8 + \hat{y}_1\hat{y}_4\hat{y}_5\hat{y}_7\hat{y}_8\hat{y}_9)$$

Example (how to use the poset P_α, not yet how to make it)



The $\hat{y}_i$ are certain monomials in the x's.

The monomial $\frac{x_5}{x_4}$ can be read off combinatorially.



Cluster variable associated to α (i.e. lambda length of α):

$$x_\alpha = \frac{x_5}{x_4}(1 + \hat{y}_4 + \hat{y}_7 + \hat{y}_1\hat{y}_4 + \hat{y}_4\hat{y}_7 + \hat{y}_1\hat{y}_4\hat{y}_7 + \hat{y}_4\hat{y}_5\hat{y}_7 + \hat{y}_1\hat{y}_4\hat{y}_8 + \hat{y}_1\hat{y}_4\hat{y}_5\hat{y}_7 + \hat{y}_1\hat{y}_4\hat{y}_7\hat{y}_8 + \hat{y}_1\hat{y}_4\hat{y}_5\hat{y}_7\hat{y}_8 + \hat{y}_1\hat{y}_4\hat{y}_5\hat{y}_7\hat{y}_8\hat{y}_9)$$

The weighted poset P_α ("non-degenerate" case)

Non-degenerate case: α is not (a tagged version of) an arc in T.

Follow α through T. Each time α crosses an arc γ of T, we get an element of P_α that is labeled (usually) with $\hat{y}_\gamma$.

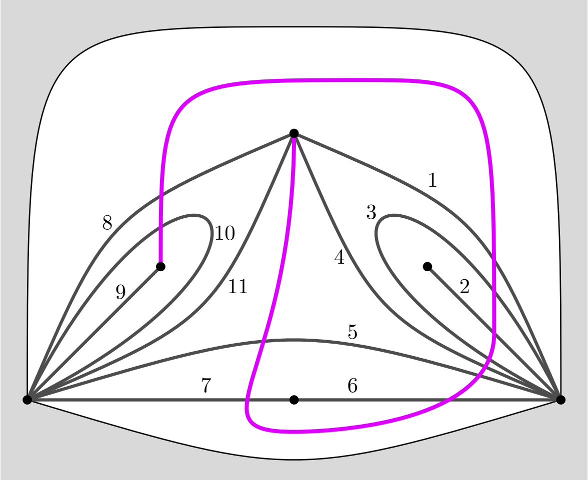
When γ is the interior edge of a self-folded triangle, the label is $\hat{y}_\gamma/\hat{y}_\beta$, where β is the exterior edge.

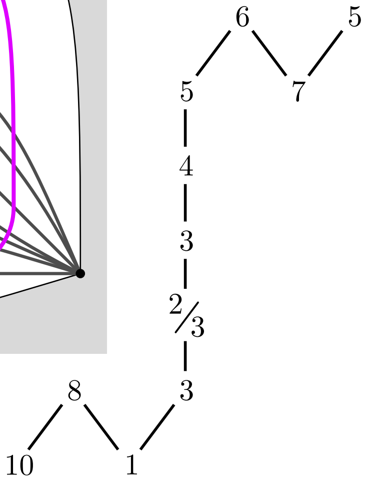
Each new element covers or is covered by the one before.
When we turn right in a triangle, we are going down in the poset.
When we turn left, we are going up.

When α is tagged notched at one or both endpoints, we add chains at those endpoints.

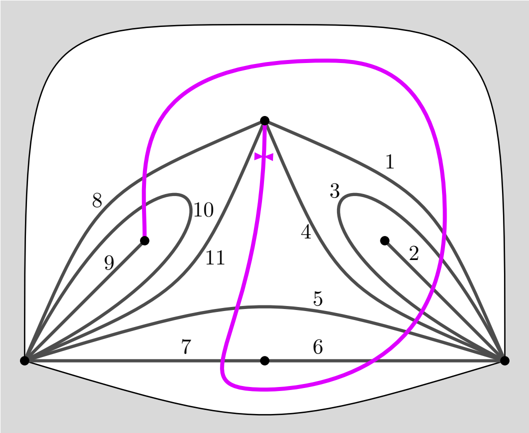
This case also done by Oğuz–Yıldırım. (See also MSW, Wilson).

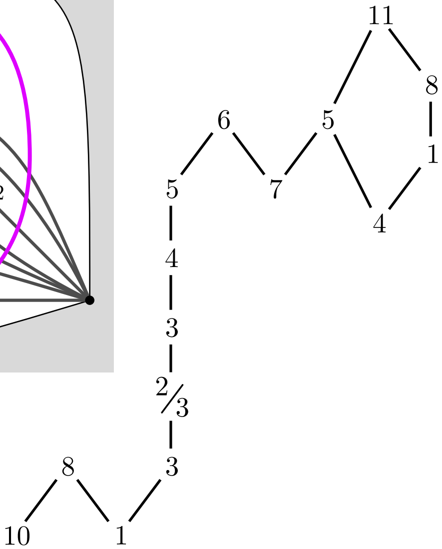
Non-degenerate case (α tagged plain)



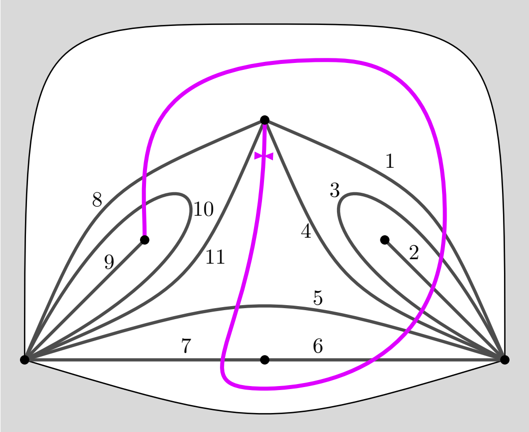


Non-degenerate case (add a chain at a tagged endpoint)

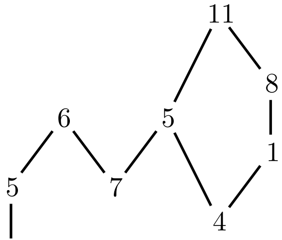


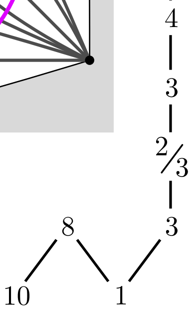


Non-degenerate case (add a chain at a tagged endpoint)



Make the chain that would correspond to an arc tracing around the endpoint. The top of the chain is above the last arc crossed. The bottom of the chain is below the last arc.

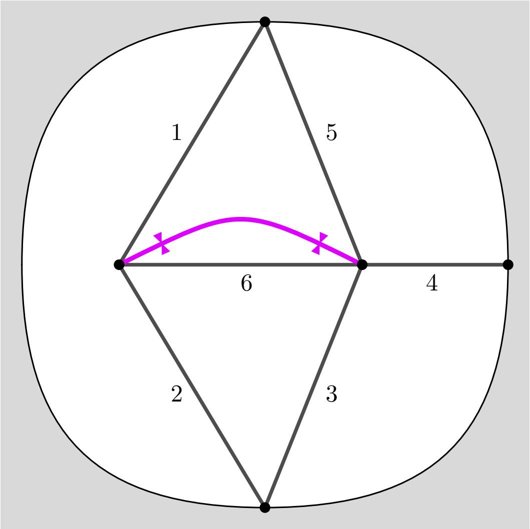


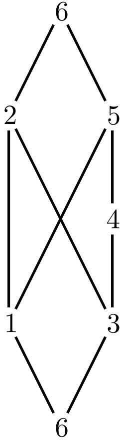


Here, the added chain is labeled $4 \lessdot 1 \lessdot 8 \lessdot 11$.

The last arc crossed is labeled 5.

A degenerate case (α notched at both ends)





Degenerate means α coincides, up to tagging, with an arc in T.

Section 5: Proof idea and comments

Proof idea (coefficient-free case)

Want to show:
The cluster variable (lambda length) for a tagged arc α is the weighted sum of downsets in P_α.

The tagged arc α can be reached by *some* sequence of flips from the initial triangulation.

If we could show that these weighted sums of downsets satisfy the exchange relations, we would be done.

In practice, it's difficult to have control over exchange relations. Why? Self-folded triangles, α may intersect the same arc in the triangulation many times, etc.

Proof idea (coefficient-free case)

Want to show:
The cluster variable (lambda length) for a tagged arc α is the weighted sum of downsets in P_α.

The tagged arc α can be reached by *some* sequence of flips from the initial triangulation.

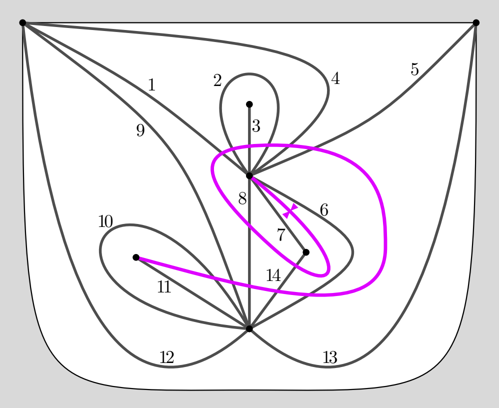
If we could show that these weighted sums of downsets satisfy the exchange relations, we would be done.

In practice, it's difficult to have control over exchange relations. Why? Self-folded triangles, α may intersect the same arc in the triangulation many times, etc.

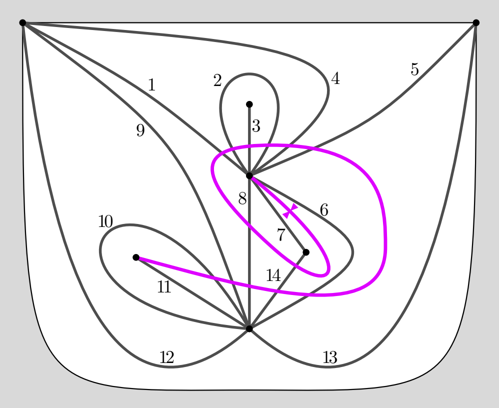
Instead, use the fact that the cluster variable is a hyperbolic length: Lift α to be a tagged arc α' in a surface without these complications, and lift the hyperbolic metric too. In the new surface, there are uncomplicated exchange relations.

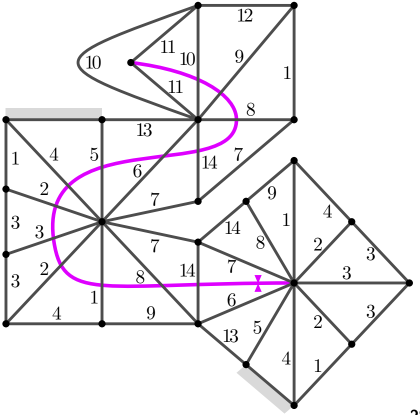
We can induct on the number of elements of P.

Picture of the lifting

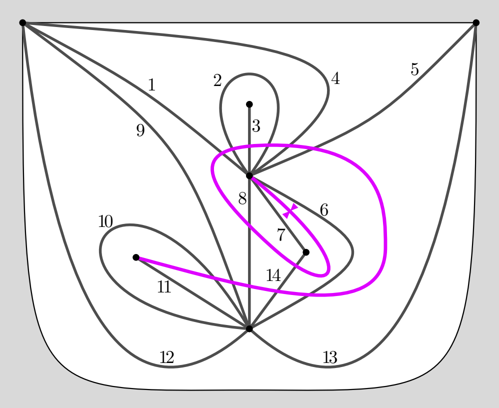


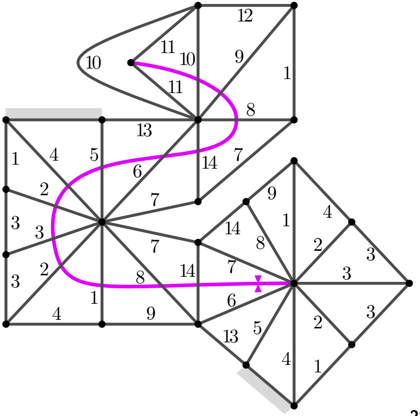
Picture of the lifting





Picture of the lifting





Key point: Lift the hyperbolic metric and the horocycles, not just the combinatorics.

So the arc α and the lift α' have the same lambda length.

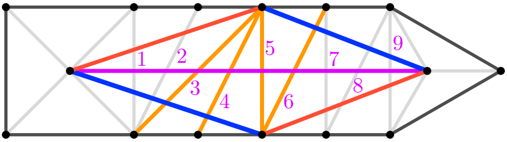
The exchange relation becomes simple combinatorics

Once we lift, $\exists$ many arcs $\gamma \in T'$ such that exchanging α' and γ is

$$F(P_{\alpha'}) = F(P_{\text{blue}}) + \hat{y}_{\text{orange}} \cdot F(P_{\text{red}}).$$

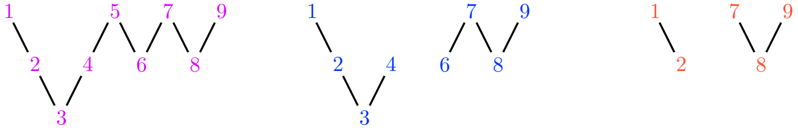
Specifically, if e_γ is the element of $P_{\alpha'}$ labeled $\hat{y}_\gamma$, then

- $F(P_{\text{blue}})$ is the weighted sum of downsets not containing e_γ.
- $\hat{y}_{\text{orange}} \cdot F(P_{\text{red}})$ is weighted sum of downsets containing e_γ.



Arc 5 is γ

$\hat{y}_{\text{orange}} = \hat{y}_3 \hat{y}_4 \hat{y}_5 \hat{y}_6$



P_α P_{blue} P_{red}

Comments

For the experts: Everything I have explained here works for the coefficient-free case (after you set all the y_i to 1). To do principal coefficients, we use laminated lambda lengths (FT 2012). Basically the same proof, once one digests FT's tropical hyperbolic geometry.

Relationship to other work: Insights from the MSW work are very important. Downsets in posets, in puncture-free case, are already in Musiker-Schiffler-Williams (2011) and Çanakçı-Schroll (2021). The non-degenerate case is in Oğuz–Yıldırım (2022). Similar posets are in work of Wilson (2020) and Weng (2023). Exchange (skein) relations are in MSW and Çanakçı–Schiffler.

What's new here:
- Make FTFDL the crucial idea.
- Treat all cases (all tagged arcs, no restrictions on **S** or **M**).
- Simple proof (poset combinatorics + FT's hyperbolic geometry).

References

Vincent Pilaud, NR, and Sibylle Schroll, (arXiv:2311.06033),
Posets for F-polynomials in cluster algebras from surfaces.

Çanakçı–Schiffler, *Snake Graph Calculus and ... surfaces. I–III.*

Çanakçı–Schroll, *Lattice bijections for string modules...*

Fomin–Shapiro–Thurston, *Cluster alg.s and triangulated surfaces. I.*

Fomin–Thurston, *Cluster algebras and triangulated surfaces. II.*

Fomin–Zelevinsky, *Cluster algebras. IV.*

Musiker–Schiffler, *Cluster expansion formulas...*

Musiker–Schiffler–Williams, *Positivity for cluster algebras...*

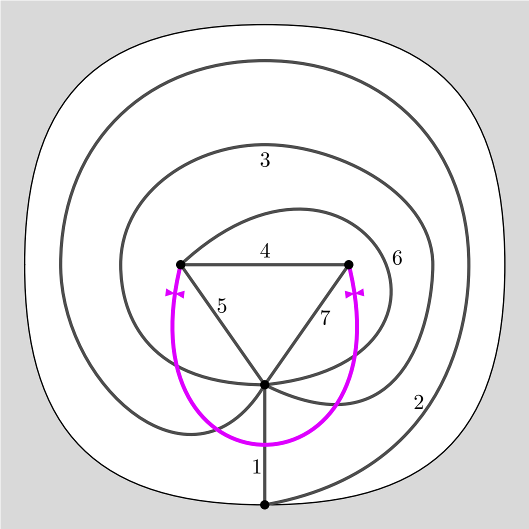
Musiker–Schiffler–Williams, *Bases for cluster algebras...*

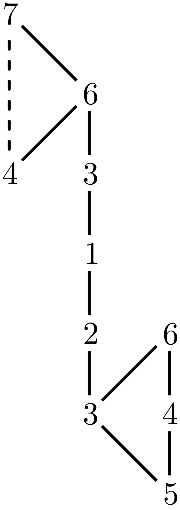
Oğuz–Yıldırım, *Cluster algebras and oriented posets*

Weng, *F-polynomials of Donaldson-Thomas transformations.*

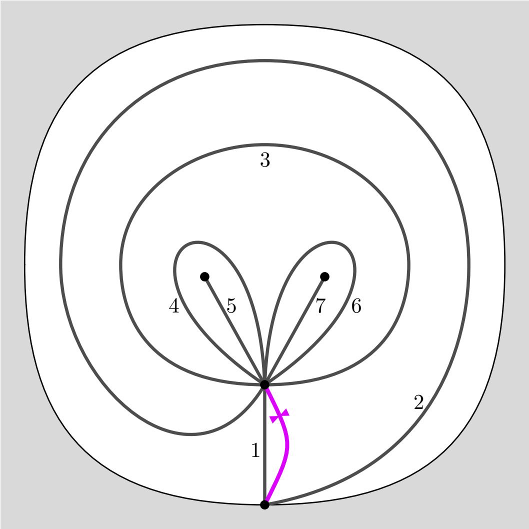
Wilson, *Surface cluster algebra expansion formulae via loop graphs.*

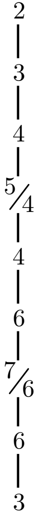
Non-degenerate case (α tagged notched at both ends)





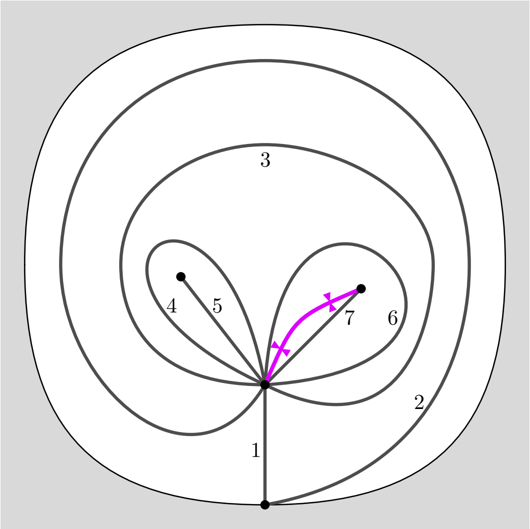
Degenerate case (α notched at one end)

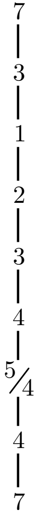




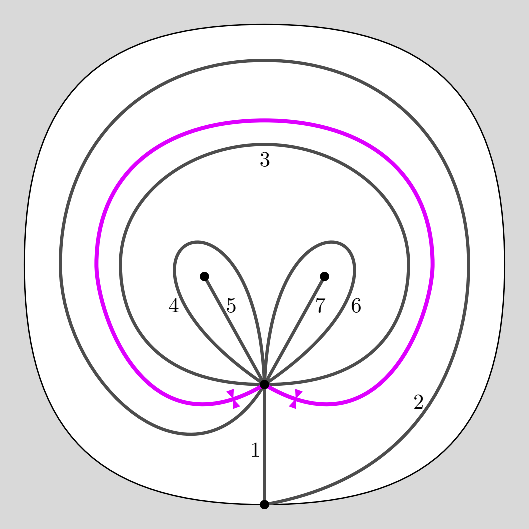
Degenerate means α coincides, up to tagging, with an arc in T.

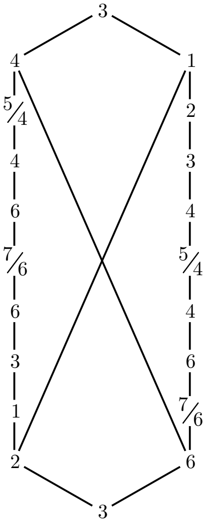
Degenerate case: α inside a self-folded triangle



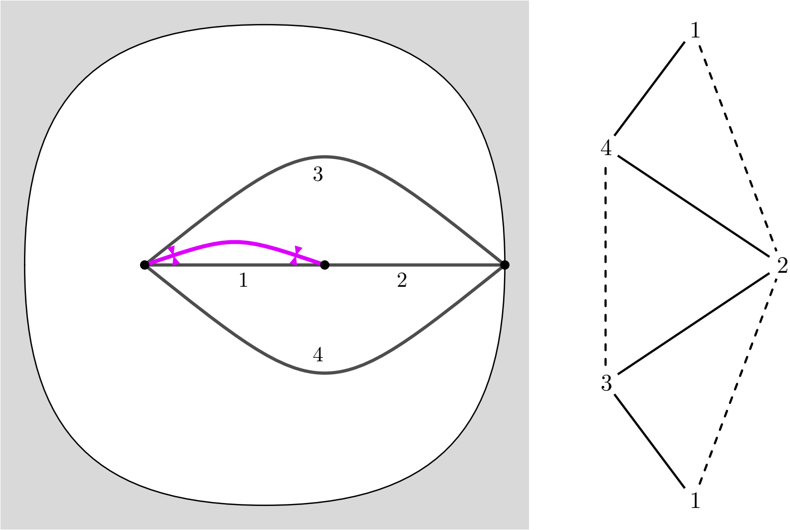


Degenerate case (α notched at both ends, 2nd example)





Degenerate case (α notched at both ends, 3rd example)

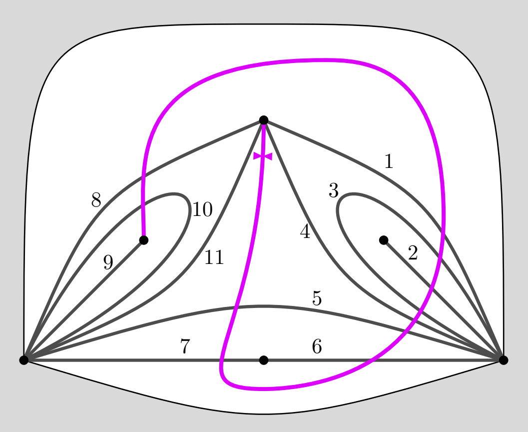


g-Vectors

Proposition. The **g**-vector of x_α is the negative of the shear coordinate vector of $\kappa(\alpha)$.

(Labardini-Fragoso, 2010, Musiker–Schiffler–Williams 2011, R. 2014, Felikson–Tumarkin 2017 + orbifolds, Pilaud–R.–Schroll 2023.)

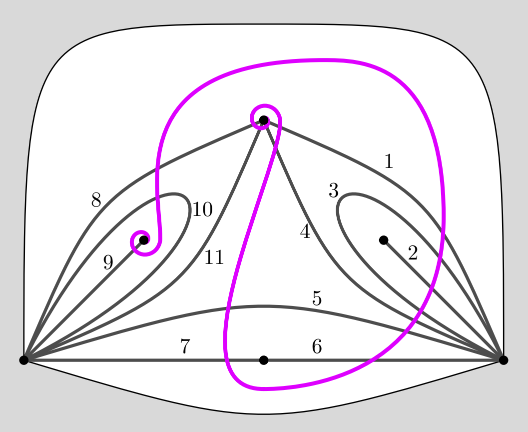
α



Proposition. The **g**-vector of x_α is the negative of the shear coordinate vector of $\kappa(\alpha)$.

(Labardini-Fragoso, 2010, Musiker–Schiffler–Williams 2011, R. 2014, Felikson–Tumarkin 2017 + orbifolds, Pilaud–R.–Schroll 2023.)

$\kappa(\alpha)$

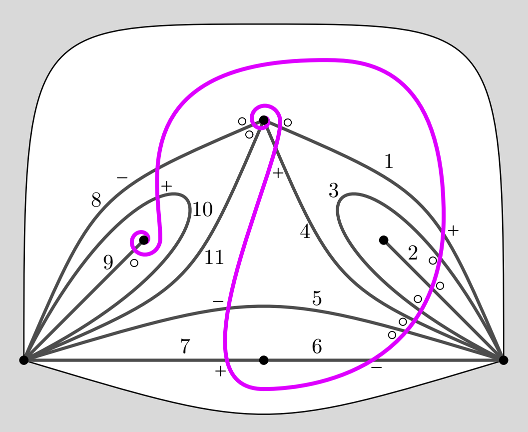


g-Vectors

Proposition. The **g**-vector of x_α is the negative of the shear coordinate vector of $\kappa(\alpha)$.

(Labardini-Fragoso, 2010, Musiker–Schiffler–Williams 2011, R. 2014, Felikson–Tumarkin 2017 + orbifolds, Pilaud–R.–Schroll 2023.)

$\kappa(\alpha)$

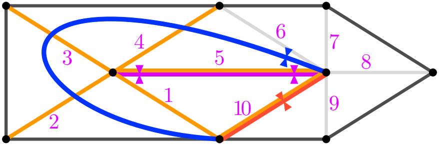


$$x^{\mathbf{g}} = \frac{x_5 x_6 x_8}{x_1 x_4 x_7 x_{10}}$$

Another exchange relation example

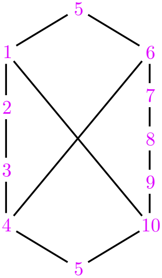
$$F(P_{\alpha'}) = F(P_{\text{blue}}) + \hat{y}_{\text{orange}} \cdot F(P_{\text{red}}).$$

- $F(P_{\text{blue}})$ is the weighted sum of downsets not containing e_γ.
- $\hat{y}_{\text{orange}} \cdot F(P_{\text{red}})$ is weighted sum of downsets containing e_γ.

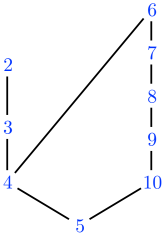


Arc 1 is γ

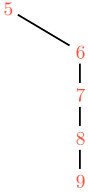
$\hat{y}_{\text{orange}} = \hat{y}_1 \hat{y}_2 \hat{y}_3 \hat{y}_4 \hat{y}_5 \hat{y}_{10}$



P_α



P_{blue}



P_{red}