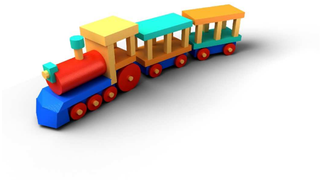
Bordered Heegaard Floer homology: A toy model

Dylan Thurston

Joint with Peter Ozsváth, Robert Lipshitz

arXiv:0810.0695



November 27, 2008, Nordic Topology Conference

Outline

- **Introduction**

  Grid diagrams: Computing HFK^-

  Toy model: Planar diagrams

  Building the algebra

  Summary statement

What are knot homologies?

Many knot invariants are one- or two-variable Laurent polynomials, associated to quantum groups.

Can often find a doubly- or triply-graded homology theory whose Euler characteristic is the polynomial invariant.

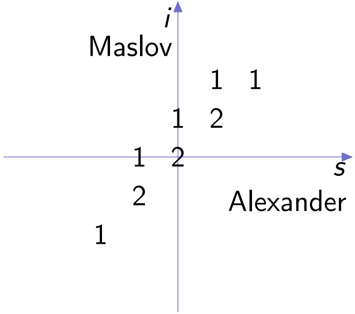
Group	Knot poly	Knot homology
$SL(2)$	Jones $J(t)$	Khovanov (1999)
$SL(n)$	HOMFLY $H(a, z)$	Kh-Roz (2004) ($n \in \mathbb{Z}$) Kh-Roz (2005) (n variable)
$GL(1 \mid 1)$	Alexander $\Delta(t)$	Heegaard Floer Seiberg–Witten Floer
$OSp(n)$	Kauffman $F(a, z)$	Kh-Roz (2007) (conjectural)

Passage polynomial $\Rightarrow$ homology called <u>categorification</u>.

Similar picture for 3-manifolds for Heegaard Floer homology.

What is Heegaard Floer homology?

$\dim(\widehat{HFK}_i(K; s))$:
$(K = 10_{132})$



Characteristics of $\widehat{HFK}$:

- Bigraded;
- Euler characteristic is Conway-Alexander polynomial;
- Max grading is knot genus (detects unknot) (Ozsváth-Szabó 2001);
- Determines knot fibration (Ghiggini, Ni 2006);
- Defined with pseudo-holomorphic curves.

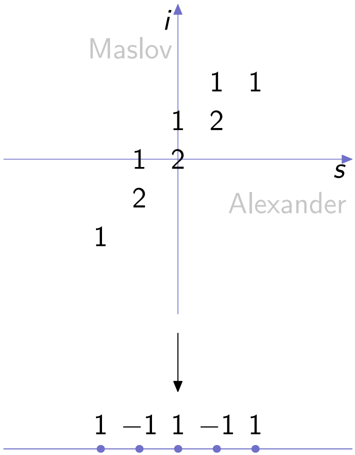
We will give a simple algorithm for computing HFK . . .

. . . and so the world's simplest algorithm for knot genus!†

†But not fastest

What is Heegaard Floer homology?

$\dim(\widehat{HFK}_i(K; s))$:
$(K = 10_{132})$



Characteristics of $\widehat{HFK}$:

- Bigraded;
- Euler characteristic is Conway-Alexander polynomial;
- Max grading is knot genus (detects unknot) (Ozsváth-Szabó 2001);
- Determines knot fibration (Ghiggini, Ni 2006);
- Defined with pseudo-holomorphic curves.

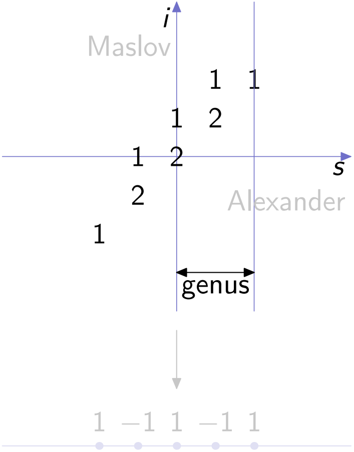
We will give a simple algorithm for computing HFK…

…and so the world's simplest algorithm for knot genus![†]

[†] But not fastest

What is Heegaard Floer homology?

$\dim(\widehat{HFK}_i(K;s))$:
$(K = 10_{132})$



Characteristics of $\widehat{HFK}$:

- Bigraded;
- Euler characteristic is Conway-Alexander polynomial;
- Max grading is knot genus (detects unknot) (Ozsváth-Szabó 2001);
- Determines knot fibration (Ghiggini, Ni 2006);
- Defined with pseudo-holomorphic curves.

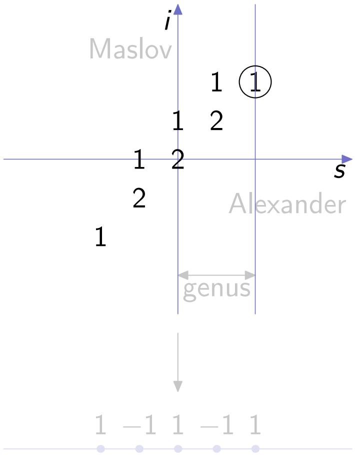
We will give a simple algorithm for computing HFK...

...and so the world's simplest algorithm for knot genus![†]

[†]But not fastest

What is Heegaard Floer homology?

$\dim(\widehat{HFK}_i(K;s))$:
$(K = 10_{132})$



Characteristics of $\widehat{HFK}$:

- Bigraded;
- Euler characteristic is Conway-Alexander polynomial;
- Max grading is knot genus (detects unknot) (Ozsváth-Szabó 2001);
- Determines knot fibration (Ghiggini, Ni 2006);
- Defined with pseudo-holomorphic curves.

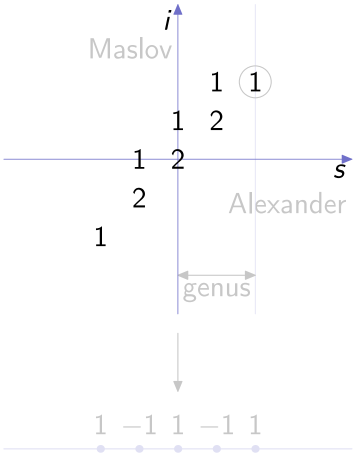
We will give a simple algorithm for computing HFK...

...and so the world's simplest algorithm for knot genus![†]

[†]But not fastest

What is Heegaard Floer homology?

$\dim(\widehat{HFK}_i(K;s))$:
$(K = 10_{132})$



Characteristics of $\widehat{HFK}$:

- Bigraded;
- Euler characteristic is Conway-Alexander polynomial;
- Max grading is knot genus (detects unknot) (Ozsváth-Szabó 2001);
- Determines knot fibration (Ghiggini, Ni 2006);
- Defined with pseudo-holomorphic curves.

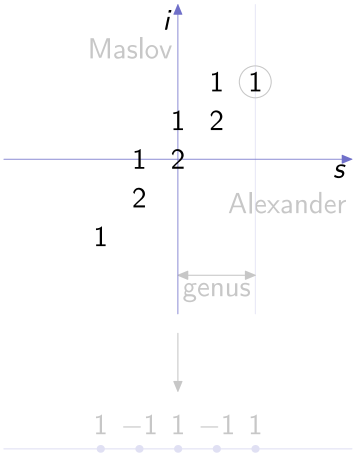
We will give a simple algorithm for computing HFK...

...and so the world's simplest algorithm for knot genus!†

†But not fastest

What is Heegaard Floer homology?

$\dim(\widehat{HFK}_i(K;s))$:
$(K = 10_{132})$



Characteristics of $\widehat{HFK}$:

- Bigraded;
- Euler characteristic is Conway-Alexander polynomial;
- Max grading is knot genus (detects unknot) (Ozsváth-Szabó 2001);
- Determines knot fibration (Ghiggini, Ni 2006);
- Defined with pseudo-holomorphic curves.

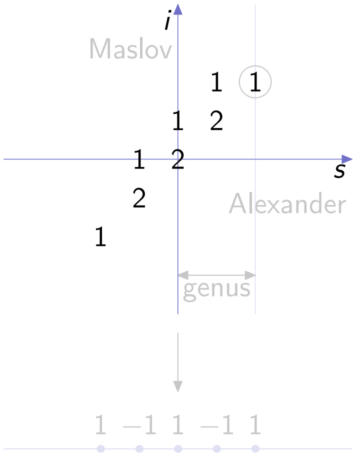
We will give a simple algorithm for computing HFK...

...and so the world's simplest algorithm for knot genus![†]

[†]But not fastest

What is Heegaard Floer homology?

$\dim(\widehat{HFK}_i(K;s))$:
$(K = 10_{132})$



Characteristics of $\widehat{HFK}$:

- Bigraded;
- Euler characteristic is Conway-Alexander polynomial;
- Max grading is knot genus (detects unknot) (Ozsváth-Szabó 2001);
- Determines knot fibration (Ghiggini, Ni 2006);
- Defined with pseudo-holomorphic curves.

We will give a simple algorithm for computing HFK...

...and so the world's simplest algorithm for knot genus![†]

[†] But not fastest

Bordered Heegaard Floer homology

Goal: **Theory for bordered 3-manifolds**

Heegaard Floer theory like a TQFT for 4-manifolds:

$$\begin{aligned}
\text{Suitable 4-manifold } W^4 &\rightsquigarrow \text{Numerical invts} \\
\text{3-manifold } Y^3 &\rightsquigarrow \text{Chain complex } CF^-(Y) \\
\text{4-manifold } W^4 \text{ with } \partial W = Y^3 &\rightsquigarrow \text{Element of } HF^-(Y) \\
\text{Invariant of } W_1 \cup_Y W_2 &= \text{Pairing of invts. of } W_1 \text{ and } W_2
\end{aligned}$$

Want to extend this down:

$$\begin{aligned}
\text{2-manifold } F^2 &\rightsquigarrow \text{Algebra } \mathcal{A}(F) \\
\text{3-manifold } Y^3 \text{ with } \partial Y = F^2 &\rightsquigarrow \text{Diff. module } CF^-(Y) \text{ over } \mathcal{A}(F) \\
CF^-(Y_1 \cup_F Y_2) &= CF^-(Y_1) \otimes_{\mathcal{A}(F)} CF^-(Y_2)
\end{aligned}$$

Complications

Complications in bordered Heegaard Floer theory:

Duality Relative invts of Y^3 with $\partial Y = F^2$ have two flavors: $CFA^-(Y)$ (over $\mathcal{A}(F)$) and $CFD^-(Y)$ (over $\mathcal{A}(-F)$).
$CF^-(Y_1 \cup_F Y_2) = CFA^-(Y_1) \otimes_{\mathcal{A}} CFD^-(Y_2)$

Analytic Defined with pseudo-holomorphic curves.
Additional complications with boundary.

Algebraic $\mathcal{A}(F)$ is a differential algebra.
Get $\mathcal{A}_\infty$ modules up to $\mathcal{A}_\infty$ homotopy equivalence.
Take derived tensor product.

We will look at toy setting where most difficulties are avoided, but leaving core of algebra.

Complications

Complications in bordered Heegaard Floer theory:

Duality Relative invts of Y^3 with $\partial Y = F^2$ have two flavors:
$CFA^-(Y)$ (over $\mathcal{A}(F)$) and $CFD^-(Y)$ (over $\mathcal{A}(-F)$).
$CF^-(Y_1 \cup_F Y_2) = CFA^-(Y_1) \otimes_{\mathcal{A}} CFD^-(Y_2)$

Analytic ~~Defined with pseudo-holomorphic curves.~~
~~Additional complications with boundary.~~

Algebraic $\mathcal{A}(F)$ is a differential algebra.
~~Get $\mathcal{A}_\infty$ modules up to $\mathcal{A}_\infty$ homotopy equivalence.~~
~~Take derived tensor product.~~

We will look at toy setting where most difficulties are avoided, but leaving core of algebra.

Outline

Introduction

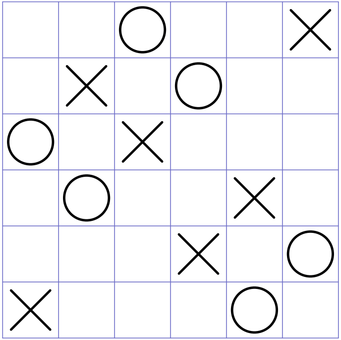
- **Grid diagrams: Computing HFK^-**

Toy model: Planar diagrams

Building the algebra

Summary statement

Setting: Grid diagrams



Grid diagram: square diagram with one X and one O per row and column.

Turn it into a knot: connect

X to O in each column;

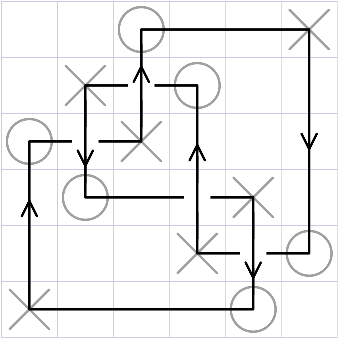
O to X in each row.

Cross vertical strands over horizontal.

Grid diagrams exist: take any diagram, rotate crossings so vertical crosses over horizontal.

The knot is unchanged under cyclic rotations:
Move top segment to bottom.
So effectively on torus.

Setting: Grid diagrams



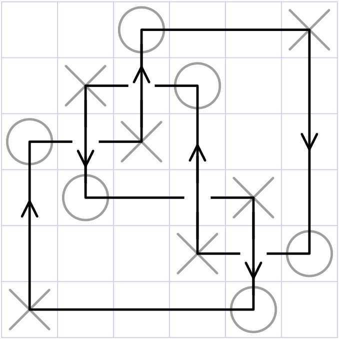
Grid diagram: square diagram with one X and one O per row and column.

Turn it into a knot: connect
X to O in each column;
O to X in each row.
Cross vertical strands over horizontal.

Grid diagrams exist: take any diagram, rotate crossings so vertical crosses over horizontal.

The knot is unchanged under
cyclic rotations:
Move top segment to bottom.
So effectively on torus.

Setting: Grid diagrams



Grid diagram: square diagram with one X and one O per row and column.

Turn it into a knot: connect

X to O in each column;

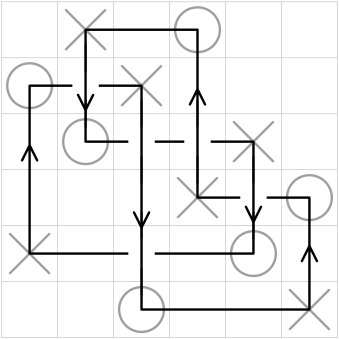
O to X in each row.

Cross vertical strands over horizontal.

Grid diagrams exist: take any diagram, rotate crossings so vertical crosses over horizontal.

The knot is unchanged under
<u>cyclic rotations</u>:
Move top segment to bottom.
So effectively on torus.

Setting: Grid diagrams



Grid diagram: square diagram with one X and one O per row and column.

Turn it into a knot: connect

X to O in each column;

O to X in each row.

Cross vertical strands over horizontal.

Grid diagrams exist: take any diagram, rotate crossings so vertical crosses over horizontal.

The knot is unchanged under <u>cyclic rotations</u>:
Move top segment to bottom.
So effectively on torus.

Computing the Alexander polynomial

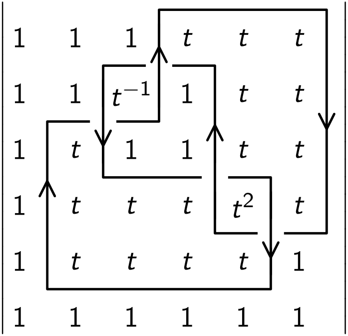
We categorify the following formula:

$$\begin{vmatrix} 1 & 1 & 1 & t & t & t \\ 1 & 1 & t^{-1} & 1 & t & t \\ 1 & t & 1 & 1 & t & t \\ 1 & t & t & t & t^2 & t \\ 1 & t & t & t & t & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{vmatrix} = \pm t^{*}(1-t)^{n-1}\Delta(K;t)$$

- Make matrix of $t^{-\text{winding }\#}$ (with extra row/column of 1's);
- det determines the Conway-Alexander polynomial Δ ($n =$ size of diagram; here 6)

Computing the Alexander polynomial

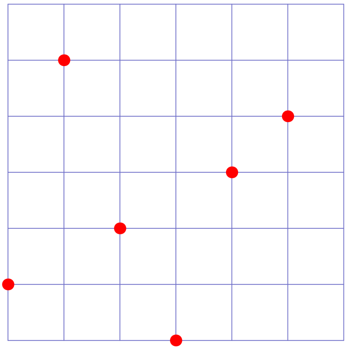
We categorify the following formula:



$$\begin{vmatrix} 1 & 1 & 1 & t & t & t \\ 1 & 1 & t^{-1} & 1 & t & t \\ 1 & t & 1 & 1 & t & t \\ 1 & t & t & t & t^2 & t \\ 1 & t & t & t & t & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{vmatrix} = \pm t^*(1-t)^{n-1}\Delta(K;t)$$

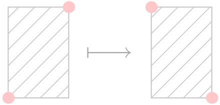
- Make matrix of $t^{-\text{winding }\#}$ (with extra row/column of 1's);
- det determines the Conway-Alexander polynomial Δ (n = size of diagram; here 6)

Computing *HFK*: Chain complex CK^-



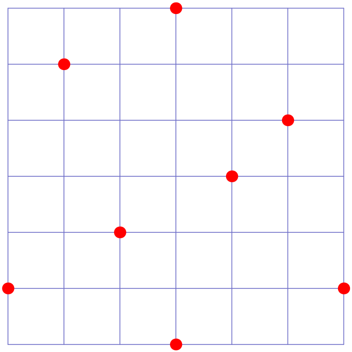
Define a chain complex CK^- over $\mathbb{F}_2$.

- $n!$ generators: matchings between horizontal and vertical gridcircles (as counted in det for Alexander).
- Boundary ∂ switches corners on empty rectangles:



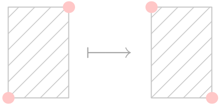
  Sum over all ways to switch SW-NE corners of an empty rectangle to NW-SE corners. (Empty means: no X's, O's, or other points in generator.)
- Upgrade: Allow O's, contribute factor U_i, over $\mathbb{F}_2[U_i]$.

Computing *HFK*: Chain complex CK^-



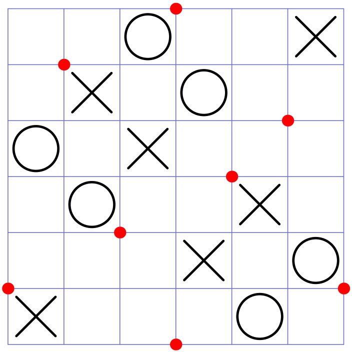
Define a chain complex CK^- over $\mathbb{F}_2$.

- $n!$ generators: matchings between horizontal and vertical gridcircles (as counted in det for Alexander).
- Boundary ∂ switches corners on empty rectangles:



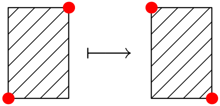
  Sum over all ways to switch SW-NE corners of an empty rectangle to NW-SE corners. (Empty means: no X's, O's, or other points in generator.)
- Upgrade: Allow O's, contribute factor U_i, over $\mathbb{F}_2[U_i]$.

Computing *HFK*: Chain complex CK^-



Define a chain complex CK^- over $\mathbb{F}_2$.

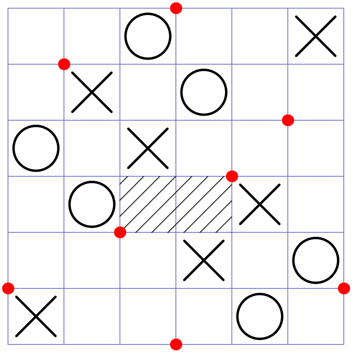
- $n!$ generators: matchings between horizontal and vertical gridcircles (as counted in det for Alexander).
- Boundary ∂ switches corners on empty rectangles:



Sum over all ways to switch SW-NE corners of an empty rectangle to NW-SE corners. (Empty means: no X's, O's, or other points in generator.)

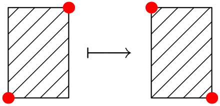
- Upgrade: Allow O's, contribute factor U_i, over $\mathbb{F}_2[U_i]$.

Computing *HFK*: Chain complex CK^-



Define a chain complex CK^- over $\mathbb{F}_2$.

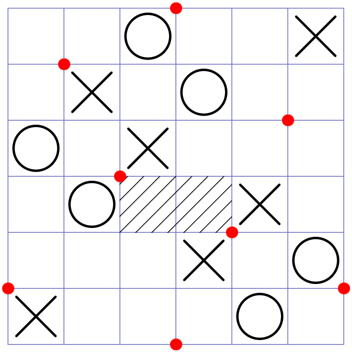
- $n!$ generators: matchings between horizontal and vertical gridcircles (as counted in det for Alexander).
- Boundary ∂ switches corners on empty rectangles:



Sum over all ways to switch SW-NE corners of an empty rectangle to NW-SE corners. (Empty means: no X's, O's, or other points in generator.)

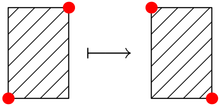
- Upgrade: Allow O's, contribute factor U_i, over $\mathbb{F}_2[U_i]$.

Computing *HFK*: Chain complex CK^-



Define a chain complex CK^- over $\mathbb{F}_2$.

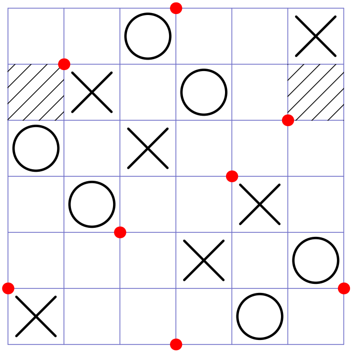
- $n!$ generators: matchings between horizontal and vertical gridcircles (as counted in det for Alexander).
- Boundary ∂ switches corners on empty rectangles:



Sum over all ways to switch SW-NE corners of an empty rectangle to NW-SE corners. (Empty means: no X's, O's, or other points in generator.)

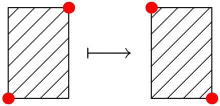
- Upgrade: Allow O's, contribute factor U_i, over $\mathbb{F}_2[U_i]$.

Computing *HFK*: Chain complex CK^-



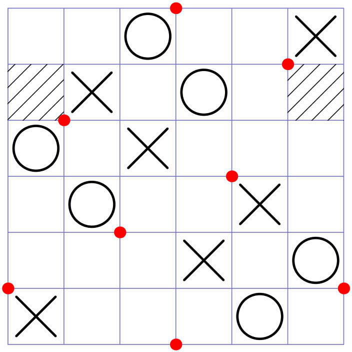
Define a chain complex CK^- over $\mathbb{F}_2$.

- $n!$ generators: matchings between horizontal and vertical gridcircles (as counted in det for Alexander).
- Boundary ∂ switches corners on empty rectangles:



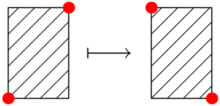
  Sum over all ways to switch SW-NE corners of an empty rectangle to NW-SE corners. (Empty means: no X's, O's, or other points in generator.)
- Upgrade: Allow O's, contribute factor U_i, over $\mathbb{F}_2[U_i]$.

Computing *HFK*: Chain complex CK^-



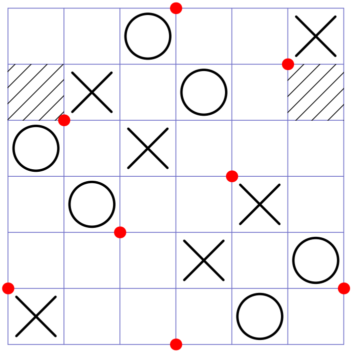
Define a chain complex CK^- over $\mathbb{F}_2$.

- $n!$ generators: matchings between horizontal and vertical gridcircles (as counted in det for Alexander).
- Boundary ∂ switches corners on empty rectangles:



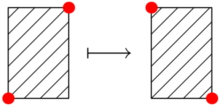
  Sum over all ways to switch SW-NE corners of an empty rectangle to NW-SE corners. (Empty means: no X's, O's, or other points in generator.)
- Upgrade: Allow O's, contribute factor U_i, over $\mathbb{F}_2[U_i]$.

Computing *HFK*: Chain complex CK^-



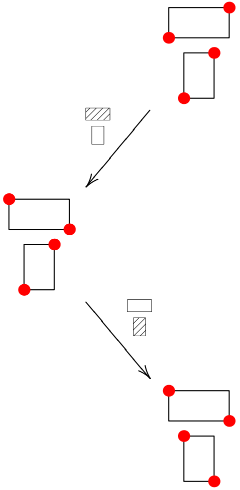
Define a chain complex CK^- over $\mathbb{F}_2$.

- $n!$ generators: matchings between horizontal and vertical gridcircles (as counted in det for Alexander).
- Boundary ∂ switches corners on empty rectangles:



  Sum over all ways to switch SW-NE corners of an empty rectangle to NW-SE corners. (Empty means: no X's, O's, or other points in generator.)
- Upgrade: Allow O's, contribute factor U_i, over $\mathbb{F}_2[U_i]$.

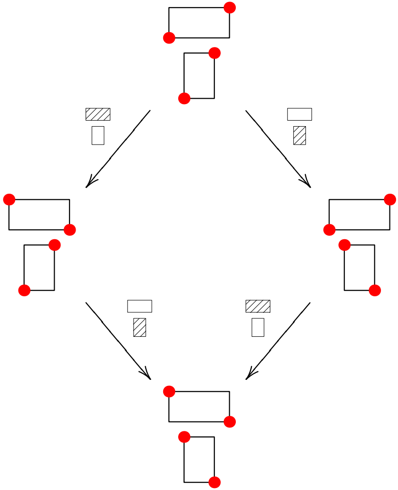
Computing *HFK*: $\partial^2 = 0$



Each term in ∂^2 must have a mate:

- If rectangles are disjoint, take rectangles in either order.
- If rectangles share a corner, decompose the union in another way.

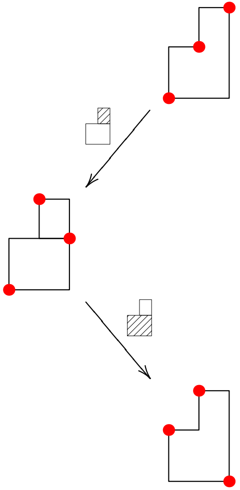
Computing *HFK*: $\partial^2 = 0$



Each term in ∂^2 must have a mate:

- If rectangles are disjoint, take rectangles in either order.
- If rectangles share a corner, decompose the union in another way.

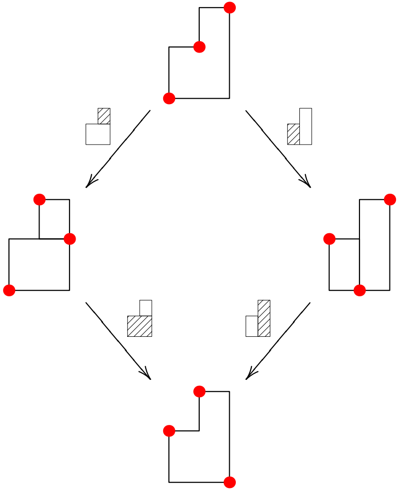
Computing *HFK*: $\partial^2 = 0$



Each term in ∂^2 must have a mate:

- If rectangles are disjoint, take rectangles in either order.
- If rectangles share a corner, decompose the union in another way.

Computing *HFK*: $\partial^2 = 0$



Each term in ∂^2 must have a mate:

- If rectangles are disjoint, take rectangles in either order.
- If rectangles share a corner, decompose the union in another way.

Computing *HFK*: The answer

Gradings can also be computed combinatorially.

Theorem (Manolescu-Ozsváth-Sarkar)

For G a grid diagram for K,

$$H_*(CK^-(G)) \simeq HFK^-(K).$$

Gillam and Baldwin used this to compute *HFK* for all knots with ≤ 11 crossings, including new values of knot genus.

Outline

Introduction

Grid diagrams: Computing HFK^-

▶ **Toy model: Planar diagrams**

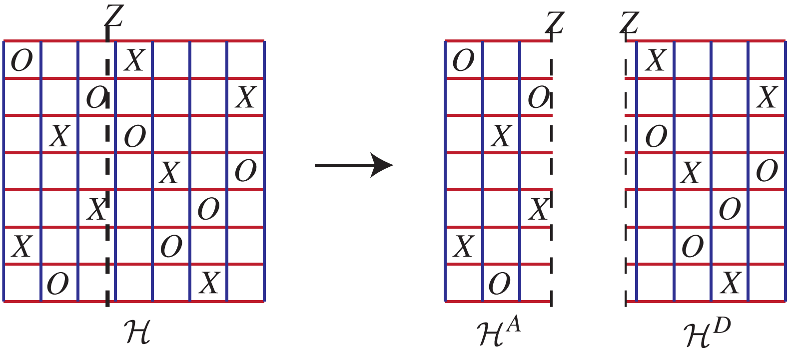
Building the algebra

Summary statement

Planar diagrams

A planar diagram $\mathcal{H}$ is just like a grid diagram, but put it on plane.

Define $CP^{-}(\mathcal{H})$ as before. Not a topological invariant of anything.



Slice along a vertical line. Parameters:

$$n = \#(\text{vertical lines}) = \#(\text{horizontal lines})$$
$$k = \#(\text{vertical lines to left of slice})$$

Outline

Introduction

Grid diagrams: Computing HFK^-

Toy model: Planar diagrams

- **Building the algebra**

Summary statement

Outline

Introduction

Grid diagrams: Computing HFK^-

Toy model: Planar diagrams

Building the algebra

- **Summary statement**

Appendix: Crossing number vs. Grid number

Knots are usually ordered by crossing number:
Minimum number of crossings in a planar diagram.

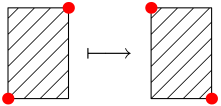
For grid diagrams, natural to consider grid number (or arc index):
Minimum size of a grid diagram.

Theorem (Bae–Park, Morton–Beltrami)

Grid number of an alternating knot is equal to crossing number $+2$.
For non-alternating knots, grid number strictly less.

Computing HFK: Gradings on $\widetilde{CK}$

In the plane,



removes one inversion.

For $A, B, C \subset \mathbb{R}^2$,

$$\mathcal{I}(A, B) := \#\{\, {}_a\square^b \mid a \in A, b \in B \,\}$$
$$\mathcal{I}(A - B, C) := \mathcal{I}(A, C) - \mathcal{I}(B, C)$$

For $\mathbf{x}$ a generator, $\mathbb{X}$ = set of X's, $\mathbb{O}$ = set of of O's, gradings are:

- **Maslov:** $M(\mathbf{x}) := \mathcal{I}(\mathbf{x} - \mathbb{O}, \mathbf{x} - \mathbb{O}) + 1$.
- **Alexander:** Sum of winding numbers around generator pts, or $A(\mathbf{x}) := \frac{1}{2}\big(\mathcal{I}(\mathbf{x} - \mathbb{O}, \mathbf{x} - \mathbb{O}) - \mathcal{I}(\mathbf{x} - \mathbb{X}, \mathbf{x} - \mathbb{X}) - (n-1)\big)$.