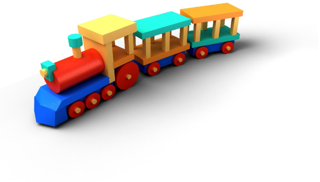
Heegaard Floer Homology
Lecture 2: Bordered HF homology, a toy model

Dylan Thurston

Joint with Robert Lipshitz, Peter Ozsváth

arXiv:0810.0695

http://www.math.columbia.edu/~dpt/speaking



July 21, 2010, XIX OMGTP, Faro

Outline

- **TQFT setup**
- Splitting diagrams
- The algebra
- Engineering $\mathcal{A}$

Extending down a dimension

Want: Extend *HF* as a TQFT down a dimension

Geometry	Algebra
Closed 4-manifold W^4	Invariant $HF(W, \mathfrak{s})$
3-manifold Y^3	Homology $HF(Y, \mathfrak{s})$
4-manifold w/ $\partial W^4 = Y$	$HF(W) \in HF(Y)$
Surface F	Algebra A(F)
3-manifold w/ $\partial Y = F$	Module $CF(Y)$ over $\mathcal{A}(F)$
$Y = Y_1 \cup_F Y_2$	$CF(Y) \simeq CF(Y_1) \otimes_{\mathcal{A}(F)} CF(Y_2)$

Benefits:

- Computability (theoretical)
- Computability (practical)
- Axioms

Extending down a dimension

Have: Extend HF as a TQFT down a dimension

Geometry	Algebra
Closed 4-manifold W^4	Invariant $HF(W, \mathfrak{s})$
3-manifold Y^3	Homology $HF(Y, \mathfrak{s})$
4-manifold w/ $\partial W^4 = Y$	$HF(W) \in HF(Y)$
Surface F	Differential algebra $\mathcal{A}(F)$
3-manifold w/ $\partial Y = F$	Differential, $\mathcal{A}_\infty$ module $\widehat{CFA}(Y)$
	Differential projective module $\widehat{CFD}(Y)$
$Y = Y_1 \cup_F Y_2$	$\widehat{CF}(Y) \simeq \widehat{CFA}(Y_1) \widetilde{\otimes}_{\mathcal{A}(F)} \widehat{CFD}(Y_2)$

Notes:

- Two versions of modules, left and right actions
- Disconnected surfaces (etc.) behave differently
- Only $\widehat{HF}$ so far
- Everything graded

Decategorifying

What happens when we decategorify?

Invariant $HF(W^4)$	$\rightsquigarrow$	$\varnothing$
Homology $HF(Y^3)$	$\overset{\text{Euler char.}}{\rightsquigarrow}$	Invariant
Algebra $\mathcal{A}(F^2)$	$\overset{\text{Grothendieck gp.}}{\rightsquigarrow}$	Abelian group
Module $CF(Y^3)$	$\rightsquigarrow$	Elt. of Grothendieck group
$CF(Y_1) \otimes_{\mathcal{A}(F)} CF(Y_2)$	$\rightsquigarrow$	Pairing of Grothendieck groups

Right hand side looks like a 3D TQFT. (Presumably this is like a Reshetikhin-Turaev TQFT, but this has not been worked out.)

Slogan

Categorification is going up a dimension.

Comparison: Reshetikhin-Turaev categorification

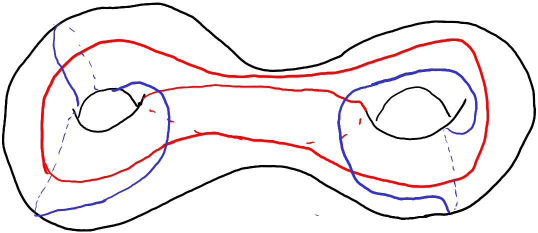
- Higher-dimensional invariants are better understood.
- Still looking for convincing algebraic/geometric background. Don't have anything like Grassmannians.
- We need differentials in the algebra. Homological direction is built in.
- Best developed for surfaces/3-manifolds (rather than tangles).
- Related to $GL(1|1)$ supergroup. This may cause some of the difficulties above.
- Tensor product of representations would be another level of structure: connect sum of surfaces

Outline

- TQFT setup
- **Splitting diagrams**
- The algebra
- Engineering $\mathcal{A}$

Splitting Heegaard diagrams

Recall that *HF* homology is defined based on Heegaard diagrams.

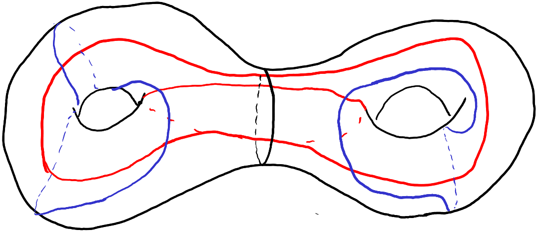


Split the diagram into two by stretching along a neck intersecting only α circles.

Need to keep track of differentials that cross the boundary.

Splitting Heegaard diagrams

Recall that *HF* homology is defined based on Heegaard diagrams.

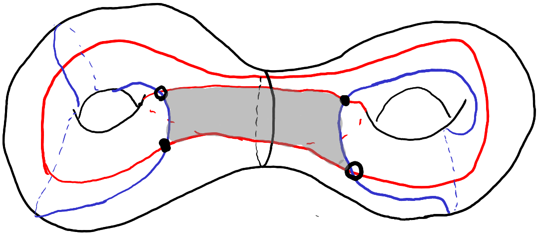


Split the diagram into two by stretching along a neck intersecting only α circles.

Need to keep track of differentials that cross the boundary.

Splitting Heegaard diagrams

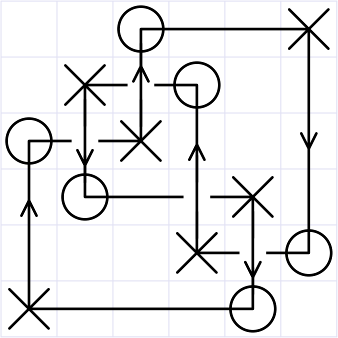
Recall that *HF* homology is defined based on Heegaard diagrams.



Split the diagram into two by stretching along a neck intersecting only α circles.

Need to keep track of differentials that cross the boundary.

Toy model: Planar diagrams



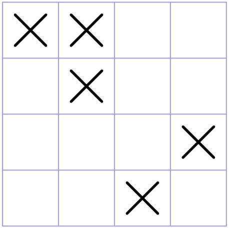
A *grid diagram* represents a knot.

A *planar diagram P* is a square grid with blocks. Do not identify sides.

Chain complex $CF(P)$:

- Generators given by permutations
- Differential counts empty rectangles (*not* wrapping)
- Not an invariant of anything

Toy model: Planar diagrams

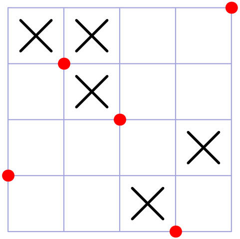


A *grid diagram* represents a knot.
A *planar diagram* P is a square grid with blocks. Do not identify sides.

Chain complex $CF(P)$:

- Generators given by permutations
- Differential counts empty rectangles (*not* wrapping)
- Not an invariant of anything

Toy model: Planar diagrams



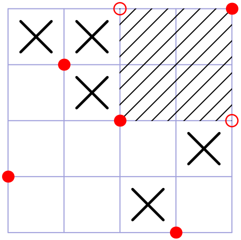
A *grid diagram* represents a knot.

A *planar diagram* P is a square grid with blocks. Do not identify sides.

Chain complex $CF(P)$:

- Generators given by permutations
- Differential counts empty rectangles (*not* wrapping)
- Not an invariant of anything

Toy model: Planar diagrams



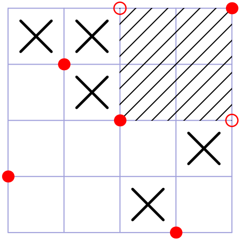
A *grid diagram* represents a knot.

A *planar diagram* P is a square grid with blocks. Do not identify sides.

Chain complex $CF(P)$:

- Generators given by permutations
- Differential counts empty rectangles (*not* wrapping)
- Not an invariant of anything

Toy model: Planar diagrams



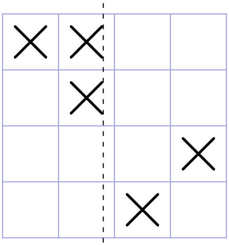
A *grid diagram* represents a knot.

A *planar diagram* P is a square grid with blocks. Do not identify sides.

Chain complex $CF(P)$:

- Generators given by permutations
- Differential counts empty rectangles (*not* wrapping)
- Not an invariant of anything

Splitting planar diagrams



Want to split planar diagrams in two: $P = P_1 \cup P_2$.

Associate modules $CPA(P_1)$, $CPD(P_2)$

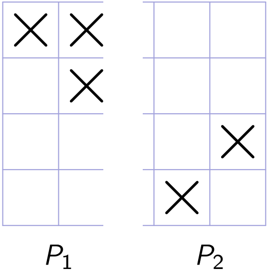
$$CP(P) = CPA(P_1) \otimes CPD(P_2)$$

$CPA(P_1)$ is a right differential module. Interactions on boundary encoded in algebra action.

$CPD(P_2)$ is a left, projective module. Interactions on boundary encoded in differential.

Tensor product is nice since $CPD(P_2)$ is projective.

Splitting planar diagrams



Want to split planar diagrams in two: $P = P_1 \cup P_2$.

Associate modules $CPA(P_1)$, $CPD(P_2)$

$$CP(P) = CPA(P_1) \otimes CPD(P_2)$$

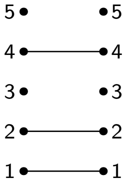
$CPA(P_1)$ is a right differential module. Interactions on boundary encoded in algebra action.

$CPD(P_2)$ is a left, projective module. Interactions on boundary encoded in differential.

Tensor product is nice since $CPD(P_2)$ is projective.

Outline

Naive algebra $\widetilde{\mathcal{A}}(n,k)$

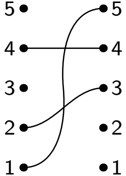


Idempotent
$\{1,2,4\} \subset \{1,\dots,5\}$

Defining a (naive) version of strands algebra $\widetilde{\mathcal{A}}(n,k)$ (really a category).

- Objects (idempotents): k-element subsets $S \subset \{1,\dots,n\}$
- Elements (morphisms): $\mathrm{Mor}(S,T)$ spanned by $\phi: S \to T$, $\phi(i) \geqslant i$
- Product: composition
- Differential: sum over smoothings of crossings

Naive algebra $\widetilde{\mathcal{A}}(n,k)$



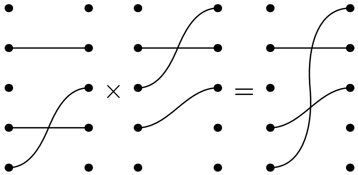
Element

$1 \mapsto 5,\ 2 \mapsto 3,\ 4 \mapsto 4$

Defining a (naive) version of strands algebra $\widetilde{\mathcal{A}}(n,k)$ (really a category).

- Objects (idempotents): k-element subsets $S \subset \{1,\ldots,n\}$
- Elements (morphisms): $\mathrm{Mor}(S,T)$ spanned by $\phi : S \to T$, $\phi(i) \geqslant i$
- Product: composition
- Differential: sum over smoothings of crossings

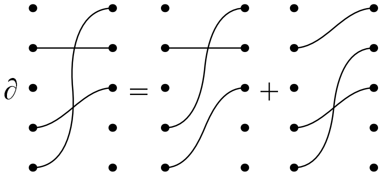
Naive algebra $\widetilde{\mathcal{A}}(n, k)$



Defining a (naive) version of strands algebra $\widetilde{\mathcal{A}}(n, k)$ (really a category).

- Objects (idempotents): k-element subsets $S \subset \{1, \ldots, n\}$
- Elements (morphisms): $\mathrm{Mor}(S, T)$ spanned by $\phi : S \to T$, $\phi(i) \geqslant i$
- Product: composition
- Differential: sum over smoothings of crossings

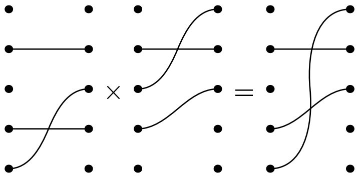
Naive algebra $\widetilde{\mathcal{A}}(n,k)$



Defining a (naive) version of strands algebra $\widetilde{\mathcal{A}}(n,k)$ (really a category).

- Objects (idempotents): k-element subsets $S \subset \{1, \dots, n\}$
- Elements (morphisms): $\mathrm{Mor}(S, T)$ spanned by $\phi : S \to T$, $\phi(i) \geqslant i$
- Product: composition
- Differential: sum over smoothings of crossings

Strands algebra $\mathcal{A}(n, k)$



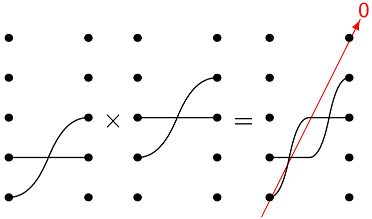
$\widetilde{\mathcal{A}}(n, k)$ has a filtration by number of crossings.

$\mathcal{A}(n, k)$ is the associated graded algebra.

- Product is 0 if it introduces a double-crossing.
- Differential: likewise.

Related to switching to nilHecke algebra (but with a differential).

Strands algebra $\mathcal{A}(n, k)$

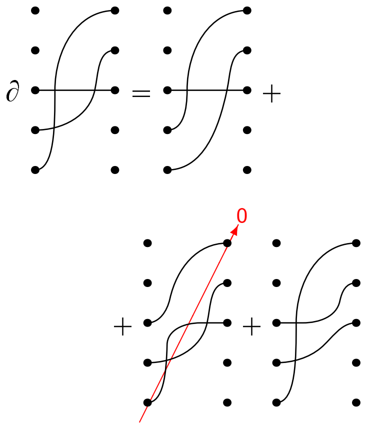


$\widetilde{\mathcal{A}}(n, k)$ has a filtration by number of crossings.
$\mathcal{A}(n, k)$ is the associated graded algebra.

- Product is 0 if it introduces a double-crossing.
- Differential: likewise.

Related to switching to nilHecke algebra (but with a differential).

Strands algebra $\mathcal{A}(n, k)$



$\widetilde{\mathcal{A}}(n, k)$ has a filtration by number of crossings.
$\mathcal{A}(n, k)$ is the associated graded algebra.

- Product is 0 if it introduces a double-crossing.
- Differential: likewise.

Related to switching to nilHecke algebra (but with a differential).

Properties of $\mathcal{A}$

$\mathcal{A}(n, k)$ is . . .

- finite-dimensional;
- Koszul dual to $\mathcal{A}(n, n-k)$ (in suitable sense); and
- very simple when $k = 1$.

The actual algebra $\mathcal{A}(F)$ (for non-toy model) is . . .

- an idempotent restriction of $\mathcal{A}(n, k)$,
- incorporates structure of the surface,
- a quiver algebra when $k = 1$, and
- not formal in general (not equivalent to its homology).

Outline

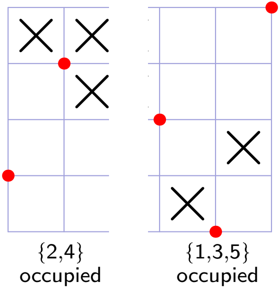
TQFT setup

Splitting diagrams

The algebra

- **Engineering $\mathcal{A}$**

Idempotents



A *generator* of a partial planar diagram is a subset **x** of the intersections with

- one element per vertical strand;
- at most one element per horizontal strand.

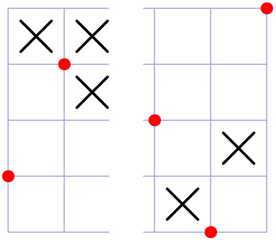
To form a complete generator, the set of occupied horizontal strands must be complementary on left and right.

For **x** a generator, define

- $I_A(\mathbf{x})$ = set of horizontal strands occupied in **x**
- $I_D(\mathbf{x})$ = complement of $I_A(\mathbf{x})$

So idempotents in $\mathcal{A}$ correspond to subsets of $\{1, \dots, n\}$, and I_A and I_D are the idempotents of generators in *CPA* and *CPD*.

From rectangles to chords



Must arrange that $CPA(P_1) \otimes CPD(P_2)$ counts all empty rectangles in P.

For rects on left or right, include a term in differential on appropriate side.

For rects crossing boundary, need the algebra.

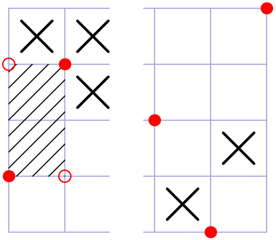
We have an algebra element associated to possible intersections of rectangles with the boundary.

$CPD(P_2)$ is a projective module generated over $\mathcal{A}$ by generators. Differential counts half-rectangles.

$CPA(P_1)$ is a module generated over $\mathbb{F}_2$ by generators. Algebra action counts half-rectangles.

Combination counts rectangles across boundary in tensor product.

From rectangles to chords



Must arrange that $CPA(P_1) \otimes CPD(P_2)$ counts all empty rectangles in P.

For rects on left or right, include a term in differential on appropriate side.

For rects crossing boundary, need the algebra.

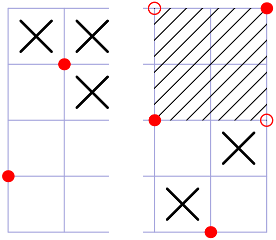
We have an algebra element associated to possible intersections of rectangles with the boundary.

$CPD(P_2)$ is a projective module generated over $\mathcal{A}$ by generators. Differential counts half-rectangles.

$CPA(P_1)$ is a module generated over $\mathbb{F}_2$ by generators. Algebra action counts half-rectangles.

Combination counts rectangles across boundary in tensor product.

From rectangles to chords



Must arrange that $CPA(P_1) \otimes CPD(P_2)$ counts all empty rectangles in P.

For rects on left or right, include a term in differential on appropriate side.

For rects crossing boundary, need the algebra.

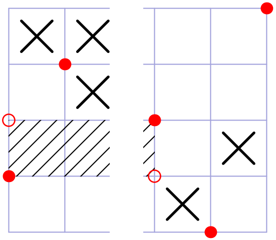
We have an algebra element associated to possible intersections of rectangles with the boundary.

$CPD(P_2)$ is a projective module generated over $\mathcal{A}$ by generators. Differential counts half-rectangles.

$CPA(P_1)$ is a module generated over $\mathbb{F}_2$ by generators. Algebra action counts half-rectangles.

Combination counts rectangles across boundary in tensor product.

From rectangles to chords



Must arrange that $CPA(P_1) \otimes CPD(P_2)$ counts all empty rectangles in P.

For rects on left or right, include a term in differential on appropriate side.

For rects crossing boundary, need the algebra.

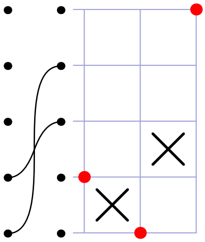
We have an algebra element associated to possible intersections of rectangles with the boundary.

$CPD(P_2)$ is a projective module generated over $\mathcal{A}$ by generators. Differential counts half-rectangles.

$CPA(P_1)$ is a module generated over $\mathbb{F}_2$ by generators. Algebra action counts half-rectangles.

Combination counts rectangles across boundary in tensor product.

From rectangles to chords



Must arrange that $CPA(P_1) \otimes CPD(P_2)$ counts all empty rectangles in P.

For rects on left or right, include a term in differential on appropriate side.

For rects crossing boundary, need the algebra.

We have an algebra element associated to possible intersections of rectangles with the boundary.

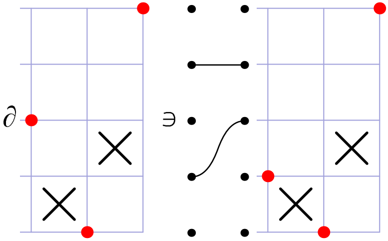
$CPD(P_2)$ is a projective module generated over $\mathcal{A}$ by generators.

Differential counts half-rectangles.

$CPA(P_1)$ is a module generated over $\mathbb{F}_2$ by generators. Algebra action counts half-rectangles.

Combination counts rectangles across boundary in tensor product.

From rectangles to chords



Must arrange that $CPA(P_1) \otimes CPD(P_2)$ counts all empty rectangles in P.

For rects on left or right, include a term in differential on appropriate side.

For rects crossing boundary, need the algebra.

We have an algebra element associated to possible intersections of rectangles with the boundary.

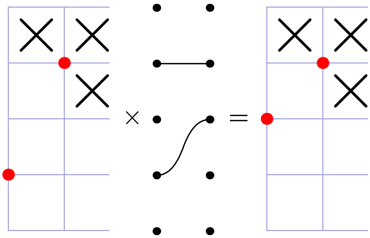
$CPD(P_2)$ is a projective module generated over $\mathcal{A}$ by generators.

Differential counts half-rectangles.

$CPA(P_1)$ is a module generated over $\mathbb{F}_2$ by generators. Algebra action counts half-rectangles.

Combination counts rectangles across boundary in tensor product.

From rectangles to chords



Must arrange that $CPA(P_1) \otimes CPD(P_2)$ counts all empty rectangles in P.

For rects on left or right, include a term in differential on appropriate side.

For rects crossing boundary, need the algebra.

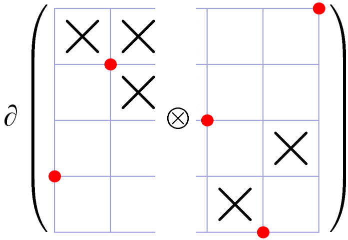
We have an algebra element associated to possible intersections of rectangles with the boundary.

$CPD(P_2)$ is a projective module generated over $\mathcal{A}$ by generators. Differential counts half-rectangles.

$CPA(P_1)$ is a module generated over $\mathbb{F}_2$ by generators. Algebra action counts half-rectangles.

Combination counts rectangles across boundary in tensor product.

From rectangles to chords



Must arrange that $CPA(P_1) \otimes CPD(P_2)$ counts all empty rectangles in P.

For rects on left or right, include a term in differential on appropriate side.

For rects crossing boundary, need the algebra.

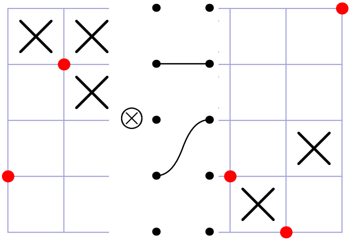
We have an algebra element associated to possible intersections of rectangles with the boundary.

$CPD(P_2)$ is a projective module generated over $\mathcal{A}$ by generators. Differential counts half-rectangles.

$CPA(P_1)$ is a module generated over $\mathbb{F}_2$ by generators. Algebra action counts half-rectangles.

Combination counts rectangles across boundary in tensor product.

From rectangles to chords



Must arrange that $CPA(P_1) \otimes CPD(P_2)$ counts all empty rectangles in P.

For rects on left or right, include a term in differential on appropriate side.

For rects crossing boundary, need the algebra.

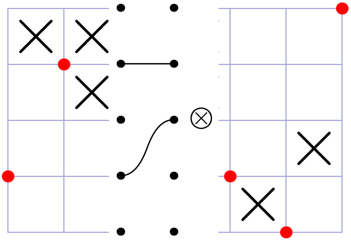
We have an algebra element associated to possible intersections of rectangles with the boundary.

$CPD(P_2)$ is a projective module generated over $\mathcal{A}$ by generators. Differential counts half-rectangles.

$CPA(P_1)$ is a module generated over $\mathbb{F}_2$ by generators. Algebra action counts half-rectangles.

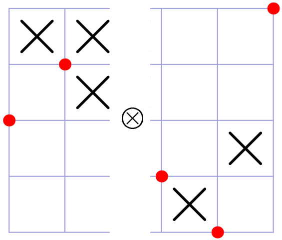
Combination counts rectangles across boundary in tensor product.

From rectangles to chords



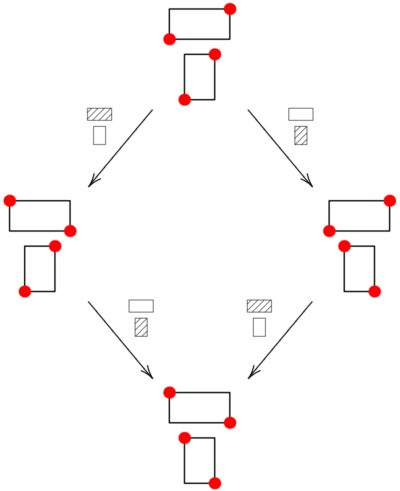
Must arrange that $CPA(P_1) \otimes CPD(P_2)$ counts all empty rectangles in P.
For rects on left or right, include a term in differential on appropriate side.
For rects crossing boundary, need the algebra.
We have an algebra element associated to possible intersections of rectangles with the boundary.
$CPD(P_2)$ is a projective module generated over $\mathcal{A}$ by generators. Differential counts half-rectangles.
$CPA(P_1)$ is a module generated over $\mathbb{F}_2$ by generators. Algebra action counts half-rectangles.
Combination counts rectangles across boundary in tensor product.

From rectangles to chords



Must arrange that $CPA(P_1) \otimes CPD(P_2)$ counts all empty rectangles in P.
For rects on left or right, include a term in differential on appropriate side.
For rects crossing boundary, need the algebra.
We have an algebra element associated to possible intersections of rectangles with the boundary.
$CPD(P_2)$ is a projective module generated over $\mathcal{A}$ by generators. Differential counts half-rectangles.
$CPA(P_1)$ is a module generated over $\mathbb{F}_2$ by generators. Algebra action counts half-rectangles.
Combination counts rectangles across boundary in tensor product.

Relations in $\mathcal{A}$



Recall that $\partial^2 = 0$ for grid/planar diagrams involves decomposing regions in two ways.

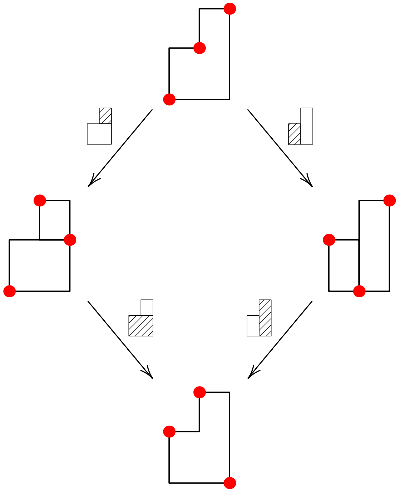
We need to account for this cancellation as it crosses the boundary.

Get relations that are ...

- always true for $CPA(P_1)$;
- necessary to get $\partial^2 = 0$ for $CPD(P_2)$.

We'll look at $CPD(P_2)$ side.

Relations in $\mathcal{A}$



Recall that $\partial^2 = 0$ for grid/planar diagrams involves decomposing regions in two ways.

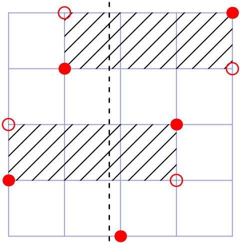
We need to account for this cancellation as it crosses the boundary.

Get relations that are . . .

- always true for $CPA(P_1)$;
- necessary to get $\partial^2 = 0$ for $CPD(P_2)$.

We'll look at $CPD(P_2)$ side.

Relations in $\mathcal{A}$

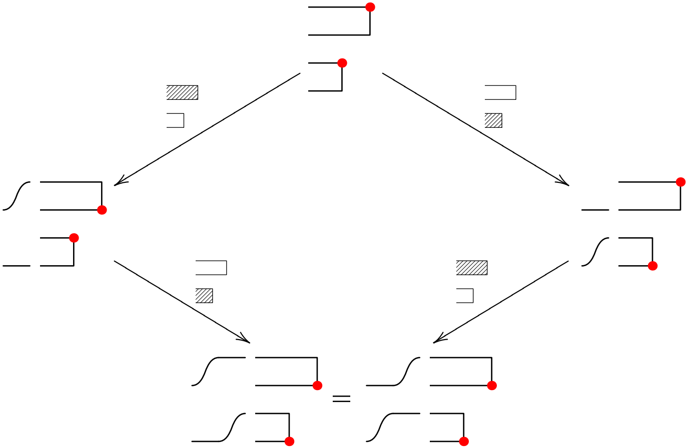


Recall that $\partial^2 = 0$ for grid/planar diagrams involves decomposing regions in two ways.
We need to account for this cancellation as it crosses the boundary.
Get relations that are ...

- always true for $CPA(P_1)$;
- necessary to get $\partial^2 = 0$ for $CPD(P_2)$.

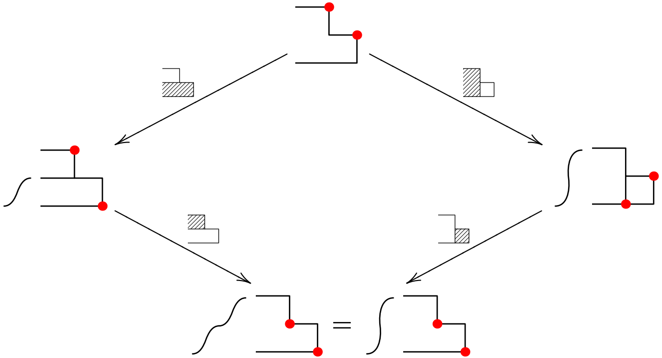
We'll look at $CPD(P_2)$ side.

Relation 1: commutativity



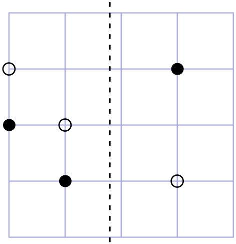
From disjoint rectangles, get commutativity of disjoint chords.

Relation 2: composition



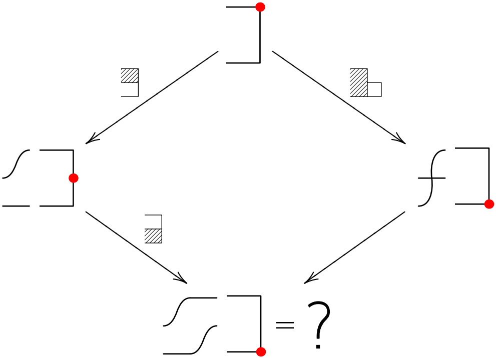
From L-shaped region, get composition of strands.

Differential



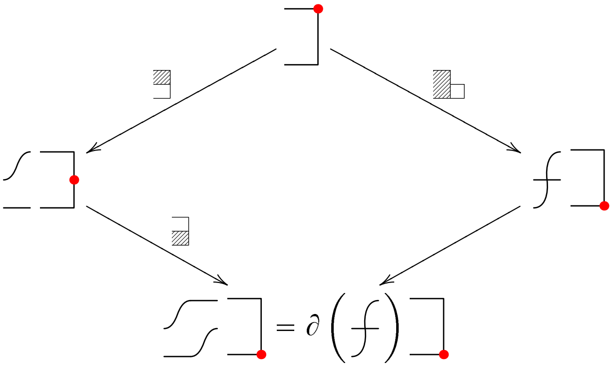
From L-shaped region facing other way, get a differential.

Differential



From L-shaped region facing other way, get a differential.

Differential



From L-shaped region facing other way, get a differential.