Edge stabilization in graph configuration spaces

Gabriel C. Drummond-Cole
(joint with Byunghee An and Ben Knudsen)

https://arxiv.org/abs/1708.02351 https://arxiv.org/abs/1806.05585

IBS-CGP

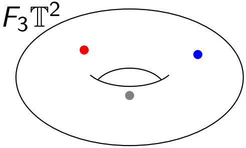
June 25, 2018

Configuration spaces

Definition (Configuration spaces)

Let X be a space. The *ordered configuration space* $F_k(X)$ of X is the space of k-tuples of distinct points in X:

$$F_k(X) := \{(x_1, \ldots, x_k) \in X^k | x_i \neq x_j \text{ for } i \neq j\}.$$



Configuration spaces

Definition (Configuration spaces)

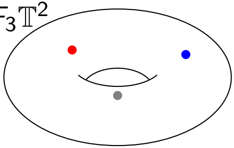
Let X be a space. The *ordered configuration space* $F_k(X)$ of X is the space of k-tuples of distinct points in X:

$$F_k(X) := \{(x_1, \ldots, x_k) \in X^k | x_i \neq x_j \text{ for } i \neq j\}.$$

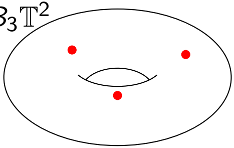
The *unordered configuration space* $B_k(X)$ is the quotient of $F_k(X)$ by the symmetric group action.

$$B_k(X) := F_k(X)/S_k$$

$F_3\mathbb{T}^2$

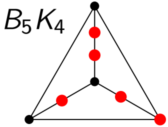


$B_3\mathbb{T}^2$



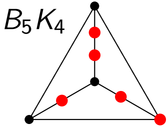
Configuration spaces of graphs

This talk is about the unordered configuration space of a graph Γ.



Configuration spaces of graphs

This talk is about the unordered configuration space of a graph Γ.

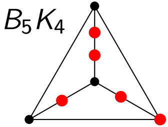


Theorem (Ghrist; Abrams)

Let Γ be a connected graph. The space $B_k(\Gamma)$ is a $K(\pi, 1)$ space.

Configuration spaces of graphs

This talk is about the unordered configuration space of a graph Γ.



Theorem (Ghrist; Abrams)

Let Γ be a connected graph. The space $B_k(\Gamma)$ is a $K(\pi, 1)$ space.

Motivating question

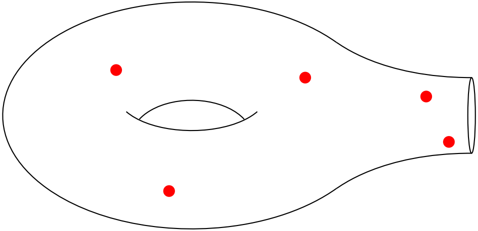
Let Γ be a graph. Calculate the bigraded groups $H_i(B_k(\Gamma))$.

(bigrading by *homological degree* and *cardinality*).

Stabilization phenomena

Stabilization at boundary components

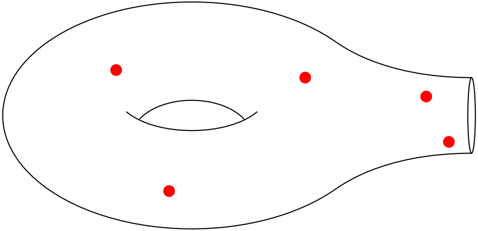
Let M be a manifold with a boundary component ∂.



Stabilization phenomena

Stabilization at boundary components

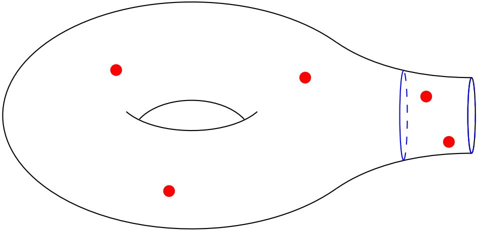
Let M be a manifold with a boundary component ∂.
There is a ∂-*stabilization* operation from $B_k(M)$ to $B_{k+1}(M)$.



Stabilization phenomena

Stabilization at boundary components

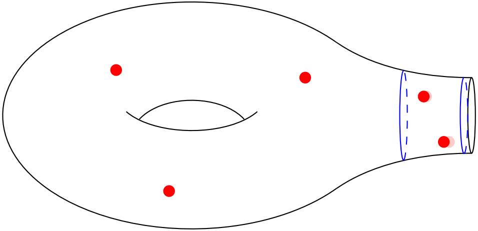
Let M be a manifold with a boundary component ∂.
There is a ∂-*stabilization* operation from $B_k(M)$ to $B_{k+1}(M)$.



Stabilization phenomena

Stabilization at boundary components

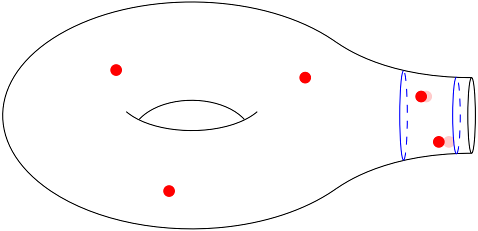
Let M be a manifold with a boundary component ∂.
There is a ∂-*stabilization* operation from $B_k(M)$ to $B_{k+1}(M)$.



Stabilization phenomena

Stabilization at boundary components

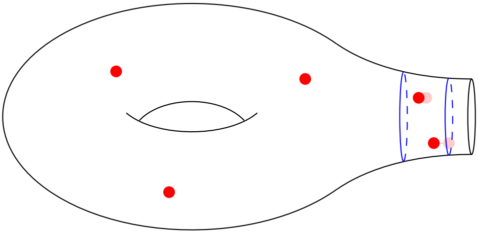
Let M be a manifold with a boundary component ∂.
There is a ∂-*stabilization* operation from $B_k(M)$ to $B_{k+1}(M)$.



Stabilization phenomena

Stabilization at boundary components

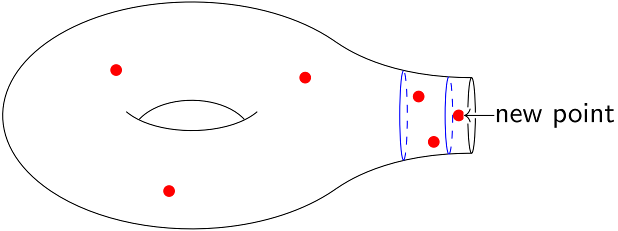
Let M be a manifold with a boundary component ∂.
There is a ∂-*stabilization* operation from $B_k(M)$ to $B_{k+1}(M)$.



Stabilization phenomena

Stabilization at boundary components

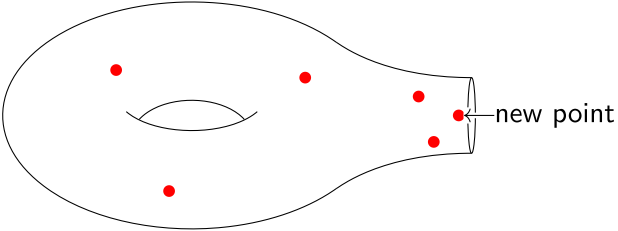
Let M be a manifold with a boundary component ∂.
There is a ∂-*stabilization* operation from $B_k(M)$ to $B_{k+1}(M)$.



Stabilization phenomena

Stabilization at boundary components

Let M be a manifold with a boundary component ∂.
There is a ∂*-stabilization* operation from $B_k(M)$ to $B_{k+1}(M)$.

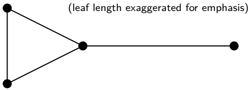


We stabilize at ∂ by deforming a cylindrical end near ∂ inwards from the boundary and adding a new point.

Stabilization phenomena

Stabilization at cylindrical ends

We can always stabilize at a cylindrical end in any space.
For example, we can stabilize at a leaf of a graph.

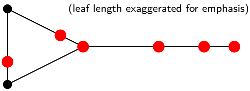


Γ

Stabilization phenomena

Stabilization at cylindrical ends

We can always stabilize at a cylindrical end in any space.
For example, we can stabilize at a leaf of a graph.

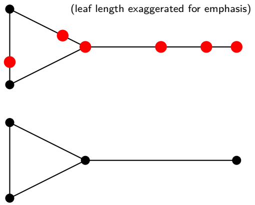


$B_6(\Gamma)$

Stabilization phenomena

Stabilization at cylindrical ends

We can always stabilize at a cylindrical end in any space.
For example, we can stabilize at a leaf of a graph.

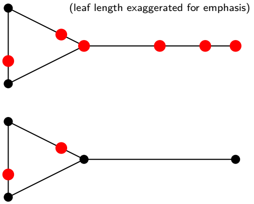


$B_6(\Gamma)$

Stabilization phenomena

Stabilization at cylindrical ends

We can always stabilize at a cylindrical end in any space.
For example, we can stabilize at a leaf of a graph.

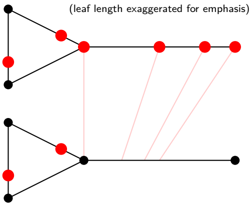


$B_6(\Gamma)$

Stabilization phenomena

Stabilization at cylindrical ends

We can always stabilize at a cylindrical end in any space.
For example, we can stabilize at a leaf of a graph.

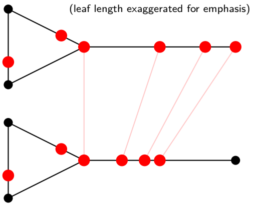


$B_6(\Gamma)$

Stabilization phenomena

Stabilization at cylindrical ends

We can always stabilize at a cylindrical end in any space.
For example, we can stabilize at a leaf of a graph.

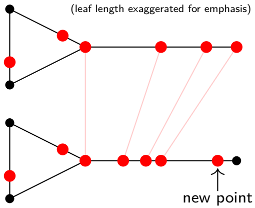


$B_6(\Gamma)$

Stabilization phenomena

Stabilization at cylindrical ends

We can always stabilize at a cylindrical end in any space.
For example, we can stabilize at a leaf of a graph.

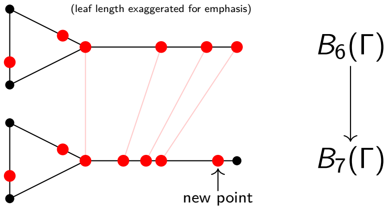


$$B_6(\Gamma) \longrightarrow B_7(\Gamma)$$

Stabilization phenomena

Stabilization at cylindrical ends

We can always stabilize at a cylindrical end in any space.
For example, we can stabilize at a leaf of a graph.



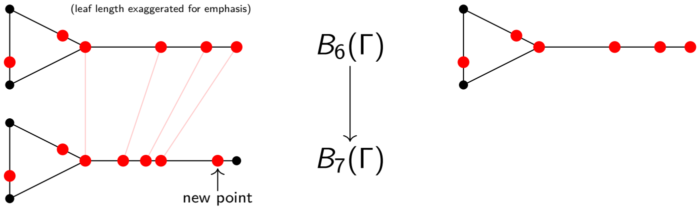
An alternative parameterization for graphs

Idea: use midpoints between configuration points in the leaf.

Stabilization phenomena

Stabilization at cylindrical ends

We can always stabilize at a cylindrical end in any space.
For example, we can stabilize at a leaf of a graph.



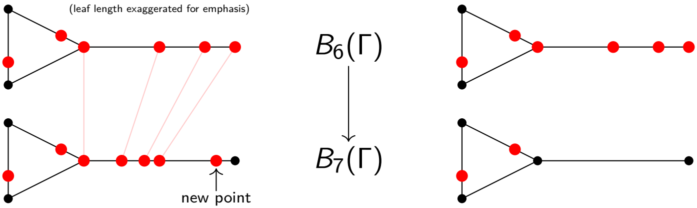
An alternative parameterization for graphs

Idea: use midpoints between configuration points in the leaf.

Stabilization phenomena

Stabilization at cylindrical ends

We can always stabilize at a cylindrical end in any space.
For example, we can stabilize at a leaf of a graph.



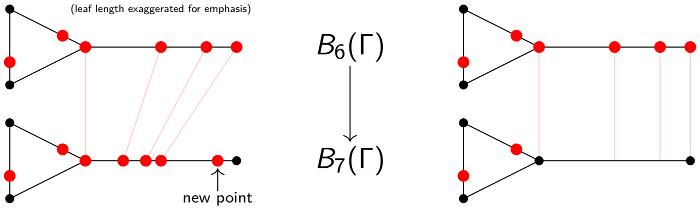
An alternative parameterization for graphs

Idea: use midpoints between configuration points in the leaf.

Stabilization phenomena

Stabilization at cylindrical ends

We can always stabilize at a cylindrical end in any space.
For example, we can stabilize at a leaf of a graph.



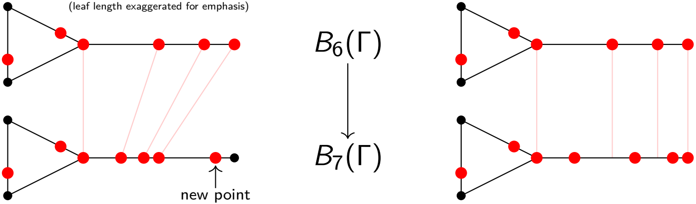
An alternative parameterization for graphs

Idea: use midpoints between configuration points in the leaf.

Stabilization phenomena

Stabilization at cylindrical ends

We can always stabilize at a cylindrical end in any space.
For example, we can stabilize at a leaf of a graph.



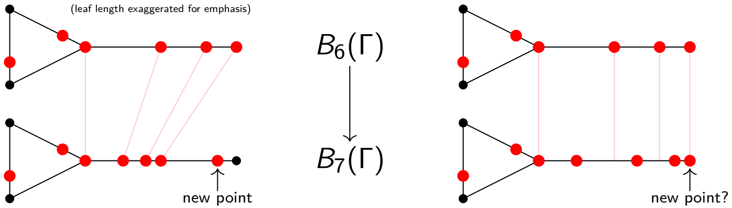
An alternative parameterization for graphs

Idea: use midpoints between configuration points in the leaf.

Stabilization phenomena

Stabilization at cylindrical ends

We can always stabilize at a cylindrical end in any space.
For example, we can stabilize at a leaf of a graph.



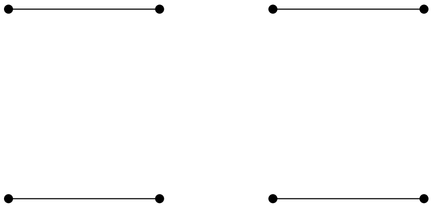
An alternative parameterization for graphs

Idea: use midpoints between configuration points in the leaf.

Midpoint stabilization

Key observation

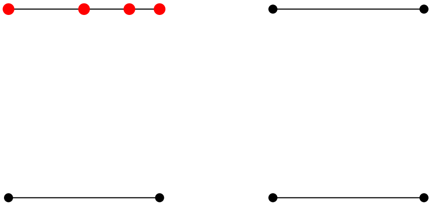
The midpoint parameterization is (basically) insensitive to the presence of configuration points at the vertices.



Midpoint stabilization

Key observation

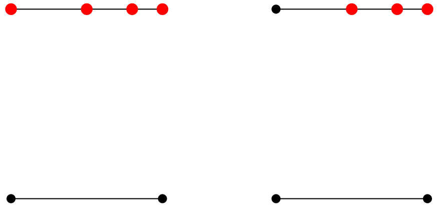
The midpoint parameterization is (basically) insensitive to the presence of configuration points at the vertices.



Midpoint stabilization

Key observation

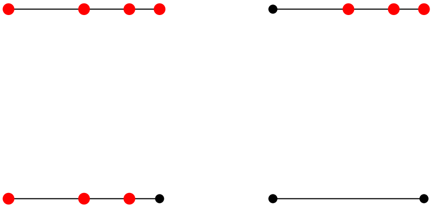
The midpoint parameterization is (basically) insensitive to the presence of configuration points at the vertices.



Midpoint stabilization

Key observation

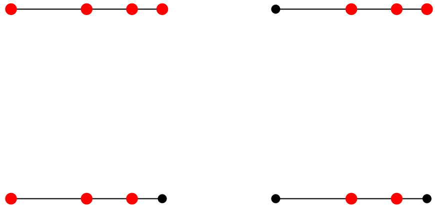
The midpoint parameterization is (basically) insensitive to the presence of configuration points at the vertices.



Midpoint stabilization

Key observation

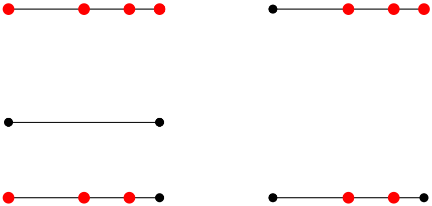
The midpoint parameterization is (basically) insensitive to the presence of configuration points at the vertices.



Midpoint stabilization

Key observation

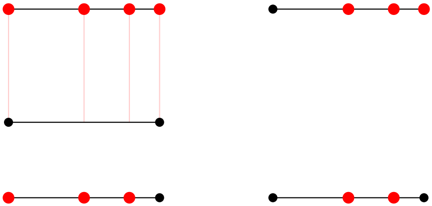
The midpoint parameterization is (basically) insensitive to the presence of configuration points at the vertices.



Midpoint stabilization

Key observation

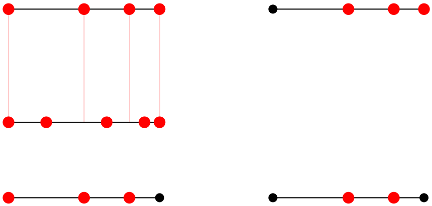
The midpoint parameterization is (basically) insensitive to the presence of configuration points at the vertices.



Midpoint stabilization

Key observation

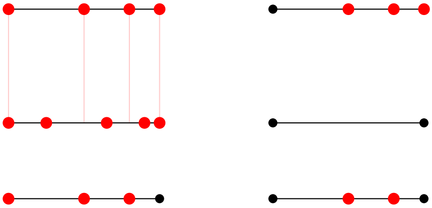
The midpoint parameterization is (basically) insensitive to the presence of configuration points at the vertices.



Midpoint stabilization

Key observation

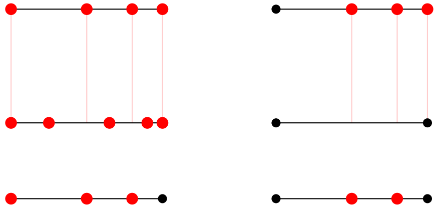
The midpoint parameterization is (basically) insensitive to the presence of configuration points at the vertices.



Midpoint stabilization

Key observation

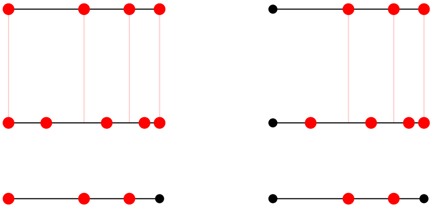
The midpoint parameterization is (basically) insensitive to the presence of configuration points at the vertices.



Midpoint stabilization

Key observation

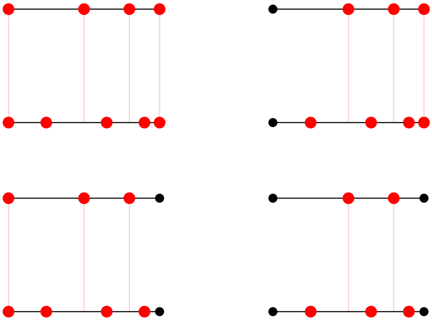
The midpoint parameterization is (basically) insensitive to the presence of configuration points at the vertices.



Midpoint stabilization

Key observation

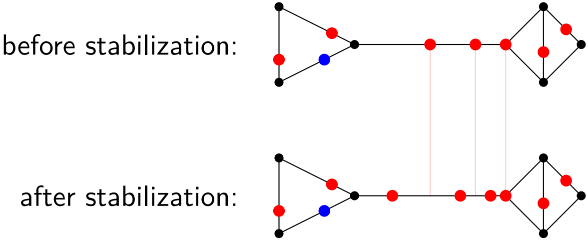
The midpoint parameterization is (basically) insensitive to the presence of configuration points at the vertices.



Edge stabilization

Consequence

We can stabilize at any edge, not just at leaves.



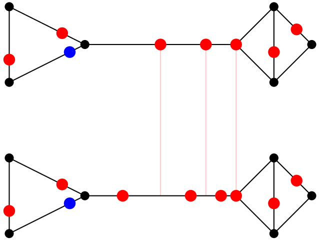
Edge stabilization

Consequence

We can stabilize at any edge, not just at leaves.

before stabilization:

after stabilization:



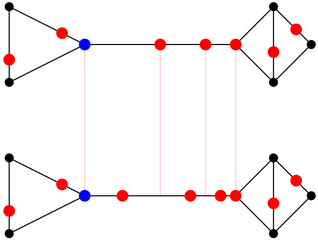
Edge stabilization

Consequence

We can stabilize at any edge, not just at leaves.

before stabilization:

after stabilization:



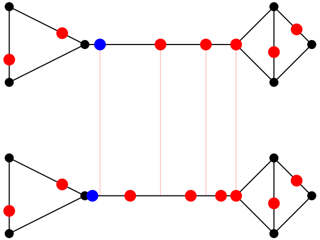
Edge stabilization

Consequence

We can stabilize at any edge, not just at leaves.

before stabilization:

after stabilization:



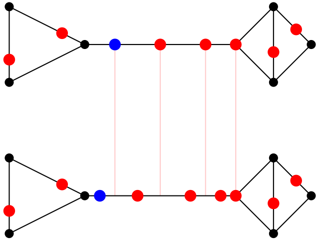
Edge stabilization

Consequence

We can stabilize at any edge, not just at leaves.

before stabilization:

after stabilization:



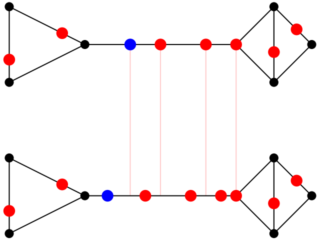
Edge stabilization

Consequence

We can stabilize at any edge, not just at leaves.

before stabilization:

after stabilization:



Edge stabilization: framework

Fix a graph Γ with edges E.

Proposition (An–D.–Knudsen)

The singular chains $C_*(B_*(\Gamma))$ are a differential bigraded $\mathbb{Z}[E]$-module.

Edge stabilization: framework

Fix a graph Γ with edges E.

Proposition (An–D.–Knudsen)

The singular chains $C_*(B_*(\Gamma))$ are a differential bigraded $\mathbb{Z}[E]$-module. There is an equivalent finitely generated $\mathbb{Z}[E]$-linear chain model.

Edge stabilization: framework

Fix a graph Γ with edges E.

Proposition (An–D.–Knudsen)

The singular chains $C_*(B_*(\Gamma))$ are a differential bigraded $\mathbb{Z}[E]$-module. There is an equivalent finitely generated $\mathbb{Z}[E]$-linear chain model.

This makes contact with earlier work at the level of homology.

Proposition (Ramos; An–D.–Knudsen)

The homology groups $H_*(B_*(\Gamma))$ are (naturally) a bigraded $\mathbb{Z}[E]$-module.

Edge stabilization: framework

Fix a graph Γ with edges E.

Proposition (An–D.–Knudsen)

The singular chains $C_*(B_*(\Gamma))$ are a differential bigraded $\mathbb{Z}[E]$-module. There is an equivalent finitely generated $\mathbb{Z}[E]$-linear chain model.

This makes contact with earlier work at the level of homology.

Proposition (Ramos; An–D.–Knudsen)

The homology groups $H_*(B_*(\Gamma))$ are (naturally) a bigraded $\mathbb{Z}[E]$-module.

Finite generation has the following consequence.

Corollary

Over a field, for any i, $\dim H_i(B_k(\Gamma))$ is eventually polynomial in k.

Growth rate of the homology

Corollary

Over a field, for any i, $\dim H_i(B_k(\Gamma))$ is eventually polynomial in k.

We can calculate the growth rate more explicitly.

Growth rate of the homology

Corollary

Over a field, for any i, $\dim H_i(B_k(\Gamma))$ is eventually polynomial in k.

We can calculate the growth rate more explicitly.

Theorem (Ramos; An–D.–Knudsen)

The polynomial degree is equal to a certain connectivity invariant of Γ.

The invariant is roughly the maximum number of connected components of the complement of i vertices in Γ.

Formality over the stabilization ring

The chain module is richer than the homology module.

Theorem (An–D.–Knudsen)

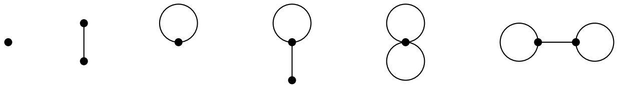
The chains $C_*(B_*(\Gamma))$ are formal as a $\mathbb{Z}[E]$-module if and only if Γ is a disjoint union of connected small graphs.

Formality over the stabilization ring

The chain module is richer than the homology module.

Theorem (An–D.–Knudsen)

The chains $C_*(B_*(\Gamma))$ are formal as a $\mathbb{Z}[E]$-module if and only if Γ is a disjoint union of connected small graphs.



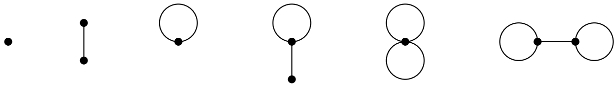
(homeomorphism classes of) connected small graphs

Formality over the stabilization ring

The chain module is richer than the homology module.

Theorem (An–D.–Knudsen)

The chains $C_*(B_*(\Gamma))$ are formal as a $\mathbb{Z}[E]$-module if and only if Γ is a disjoint union of connected small graphs.



(homeomorphism classes of) connected small graphs

Non-formality means the homotopy type of the homology module is not the same as the homotopy type of the chain module.

A local chain model

An important tool is a chain model for configurations near a vertex v.

A local chain model

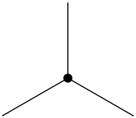
An important tool is a chain model for configurations near a vertex v.

Construction-Definition: $S(v)$

Consider the free bigraded $\mathbb{Z}[e_1, \ldots, e_n]$-module $S(v)$ on the generators:

A local chain model

An important tool is a chain model for configurations near a vertex v.



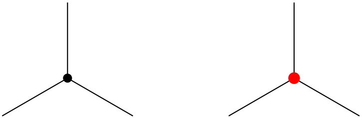
Construction-Definition: $S(v)$

Consider the free bigraded $\mathbb{Z}[e_1, \ldots, e_n]$-module $S(v)$ on the generators:

- the symbol $\emptyset$ in bidegree $(0, 0)$,

A local chain model

An important tool is a chain model for configurations near a vertex v.



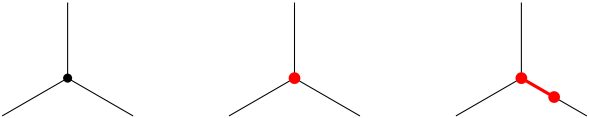
Construction-Definition: $S(v)$

Consider the free bigraded $\mathbb{Z}[e_1, \ldots, e_n]$-module $S(v)$ on the generators:

- the symbol $\emptyset$ in bidegree $(0, 0)$,
- the symbol v in bidegree $(0, 1)$, and

A local chain model

An important tool is a chain model for configurations near a vertex v.



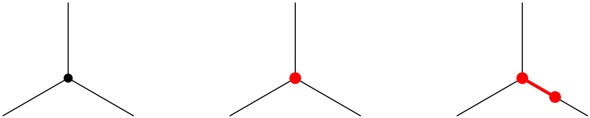
Construction-Definition: $S(v)$

Consider the free bigraded $\mathbb{Z}[e_1, \ldots, e_n]$-module $S(v)$ on the generators:

- the symbol $\emptyset$ in bidegree $(0, 0)$,
- the symbol v in bidegree $(0, 1)$, and
- the symbols $h_1, \ldots, h_n$ in bidegree $(1, 1)$

A local chain model

An important tool is a chain model for configurations near a vertex v.



Construction-Definition: $S(v)$

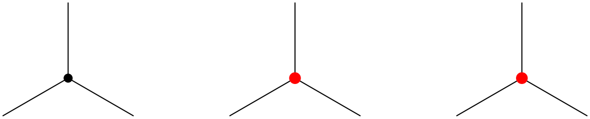
Consider the free bigraded $\mathbb{Z}[e_1, \ldots, e_n]$-module $S(v)$ on the generators:

- the symbol $\emptyset$ in bidegree $(0, 0)$,
- the symbol v in bidegree $(0, 1)$, and
- the symbols $h_1, \ldots, h_n$ in bidegree $(1, 1)$

Equip this module with the differential $d(h_j) = e_j - v$.

A local chain model

An important tool is a chain model for configurations near a vertex v.



Construction-Definition: $S(v)$

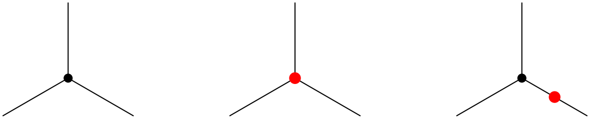
Consider the free bigraded $\mathbb{Z}[e_1, \ldots, e_n]$-module $S(v)$ on the generators:

- the symbol $\emptyset$ in bidegree $(0, 0)$,
- the symbol v in bidegree $(0, 1)$, and
- the symbols $h_1, \ldots, h_n$ in bidegree $(1, 1)$

Equip this module with the differential $d(h_j) = e_j - v$.

A local chain model

An important tool is a chain model for configurations near a vertex v.



Construction-Definition: $S(v)$

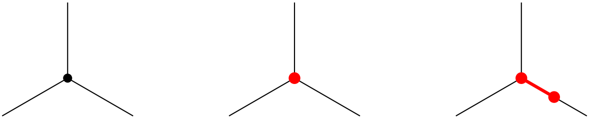
Consider the free bigraded $\mathbb{Z}[e_1, \ldots, e_n]$-module $S(v)$ on the generators:

- the symbol $\emptyset$ in bidegree $(0, 0)$,
- the symbol v in bidegree $(0, 1)$, and
- the symbols $h_1, \ldots, h_n$ in bidegree $(1, 1)$

Equip this module with the differential $d(h_j) = e_j - v$.

A local chain model

An important tool is a chain model for configurations near a vertex v.



Construction-Definition: $S(v)$

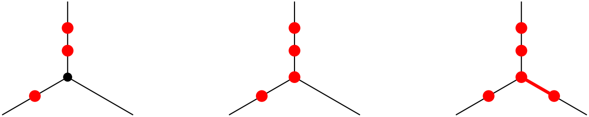
Consider the free bigraded $\mathbb{Z}[e_1, \ldots, e_n]$-module $S(v)$ on the generators:

- the symbol $\emptyset$ in bidegree $(0, 0)$,
- the symbol v in bidegree $(0, 1)$, and
- the symbols $h_1, \ldots, h_n$ in bidegree $(1, 1)$

Equip this module with the differential $d(h_j) = e_j - v$.

A local chain model

An important tool is a chain model for configurations near a vertex v.



Construction-Definition: $S(v)$

Consider the free bigraded $\mathbb{Z}[e_1, \ldots, e_n]$-module $S(v)$ on the generators:

- the symbol $\emptyset$ in bidegree $(0, 0)$,
- the symbol v in bidegree $(0, 1)$, and
- the symbols $h_1, \ldots, h_n$ in bidegree $(1, 1)$

Equip this module with the differential $d(h_j) = e_j - v$.

A finitely generated chain model

Theorem (Świątkowski; Lütgehetmann; Chettih–Lütgehetmann; An–D.–Knudsen)

There is a finitely generated bigraded differential $\mathbb{Z}[E]$*-module* $S_{*,*}(\Gamma)$ *weakly equivalent to* $C_*(B_*(\Gamma))$.

A finitely generated chain model

Theorem (Świątkowski; Lütgehetmann; Chettih–Lütgehetmann; An–D.–Knudsen)

There is a finitely generated bigraded differential $\mathbb{Z}[E]$*-module* $S_{*,*}(\Gamma)$ *weakly equivalent to* $C_*(B_*(\Gamma))$*.*

This model can can be constructed by tensoring together our local vertex models over the stabilization ring.

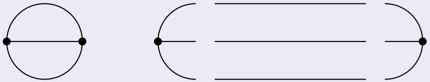
A finitely generated chain model

Theorem (Świątkowski; Lütgehetmann; Chettih–Lütgehetmann; An–D.–Knudsen)

There is a finitely generated bigraded differential $\mathbb{Z}[E]$*-module* $S_{*,*}(\Gamma)$ *weakly equivalent to* $C_*(B_*(\Gamma))$.

This model can can be constructed by tensoring together our local vertex models over the stabilization ring.

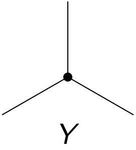
Example



$$S_{*,*}(\Gamma) \cong \quad S(v) \quad \otimes_{\mathbb{Z}[x_1,x_2,x_3]} \quad S(w)$$

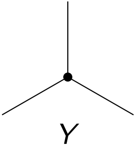
Star classes

Many things can be built from a kind of local class we call a *star class*.



Star classes

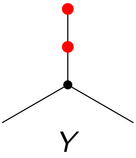
Many things can be built from a kind of local class we call a *star class*.



$B_2(Y)$

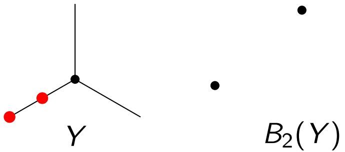
Star classes

Many things can be built from a kind of local class we call a *star class*.



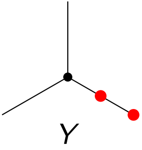
Star classes

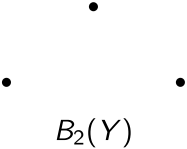
Many things can be built from a kind of local class we call a *star class*.



Star classes

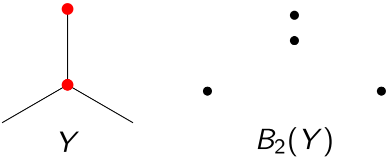
Many things can be built from a kind of local class we call a *star class*.





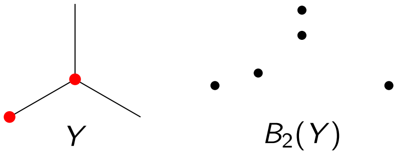
Star classes

Many things can be built from a kind of local class we call a *star class*.



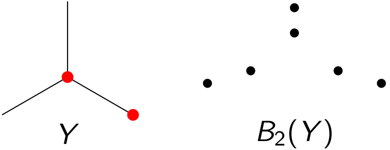
Star classes

Many things can be built from a kind of local class we call a *star class*.



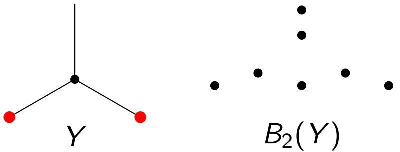
Star classes

Many things can be built from a kind of local class we call a *star class*.



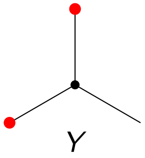
Star classes

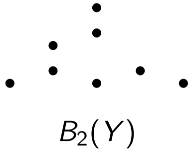
Many things can be built from a kind of local class we call a *star class*.



Star classes

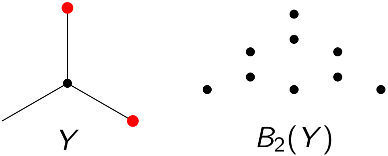
Many things can be built from a kind of local class we call a *star class*.





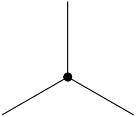
Star classes

Many things can be built from a kind of local class we call a *star class*.

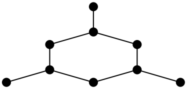


Star classes

Many things can be built from a kind of local class we call a *star class*.



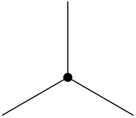
Y

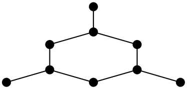


$B_2(Y)$

Star classes

Many things can be built from a kind of local class we call a *star class*.



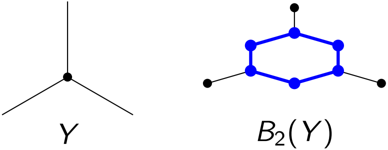


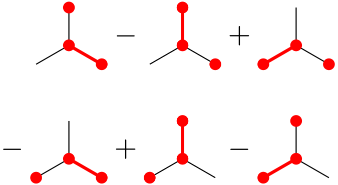
Definition

A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

Many things can be built from a kind of local class we call a *star class*.



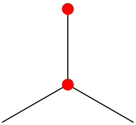


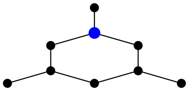
Definition

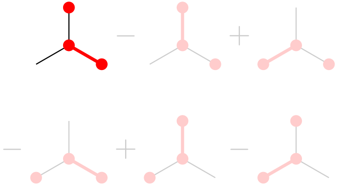
A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

Many things can be built from a kind of local class we call a *star class*.





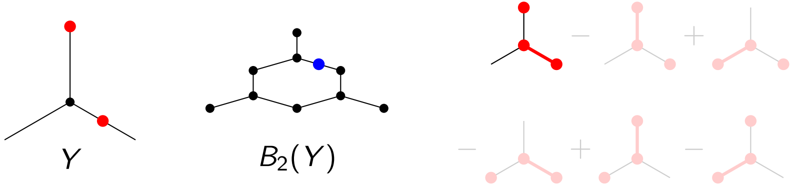


Definition

A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

Many things can be built from a kind of local class we call a *star class*.

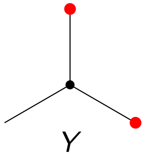


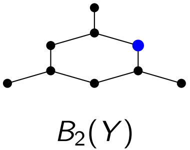
Definition

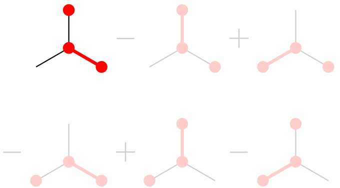
A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

Many things can be built from a kind of local class we call a *star class*.





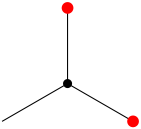


Definition

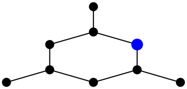
A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

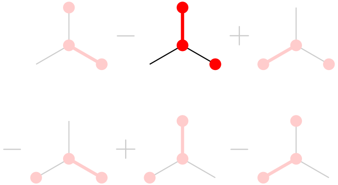
Many things can be built from a kind of local class we call a *star class*.



Y



$B_2(Y)$

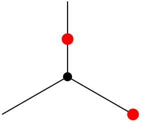


Definition

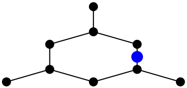
A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

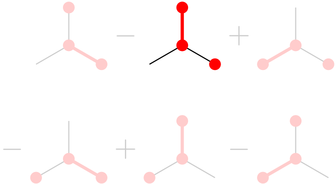
Many things can be built from a kind of local class we call a *star class*.



Y



$B_2(Y)$

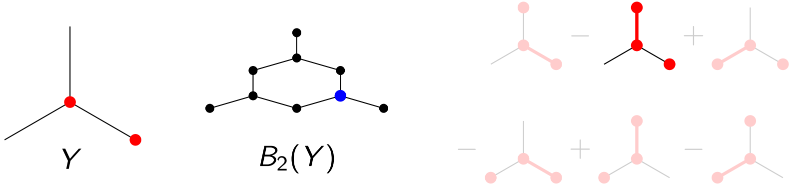


Definition

A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

Many things can be built from a kind of local class we call a *star class*.

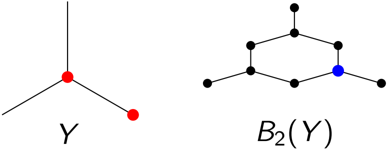


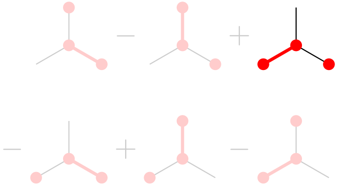
Definition

A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

Many things can be built from a kind of local class we call a *star class*.



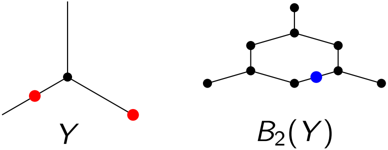


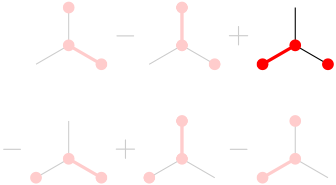
Definition

A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

Many things can be built from a kind of local class we call a *star class*.



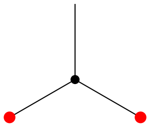


Definition

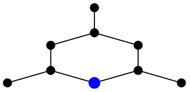
A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

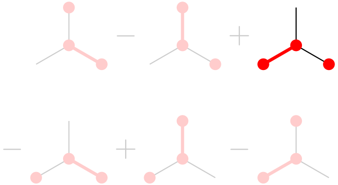
Many things can be built from a kind of local class we call a *star class*.



Y



$B_2(Y)$

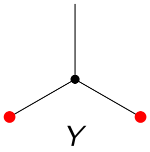


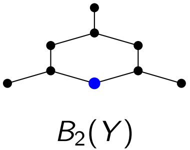
Definition

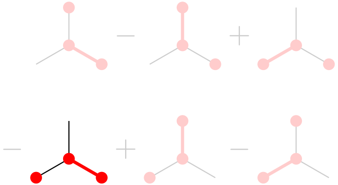
A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

Many things can be built from a kind of local class we call a *star class*.





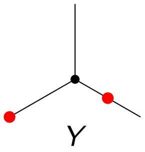


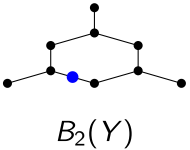
Definition

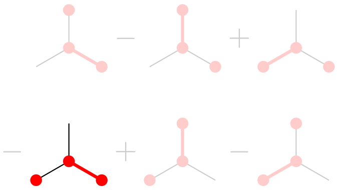
A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

Many things can be built from a kind of local class we call a *star class*.





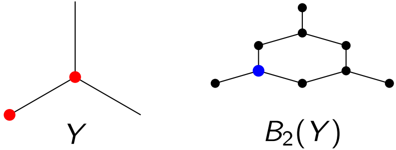


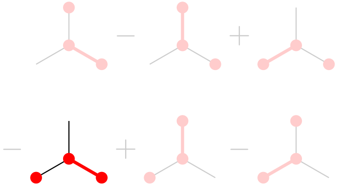
Definition

A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

Many things can be built from a kind of local class we call a *star class*.



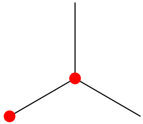


Definition

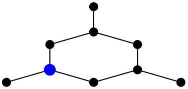
A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

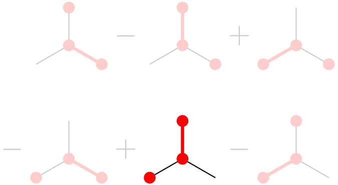
Many things can be built from a kind of local class we call a *star class*.



Y



$B_2(Y)$

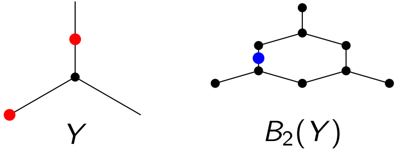


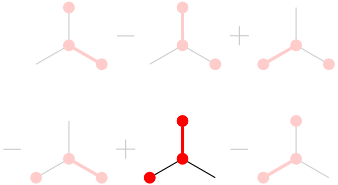
Definition

A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

Many things can be built from a kind of local class we call a *star class*.



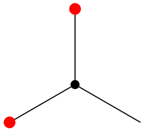


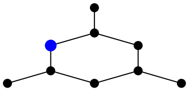
Definition

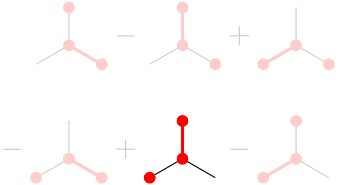
A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

Many things can be built from a kind of local class we call a *star class*.





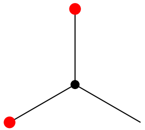


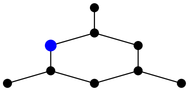
Definition

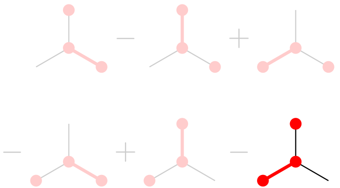
A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

Many things can be built from a kind of local class we call a *star class*.





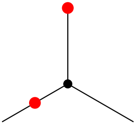


Definition

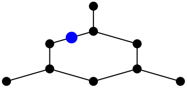
A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

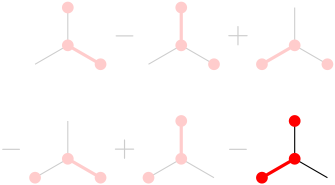
Many things can be built from a kind of local class we call a *star class*.



Y



$B_2(Y)$

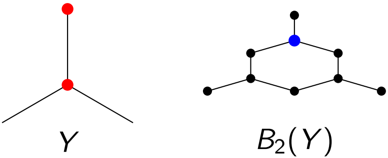


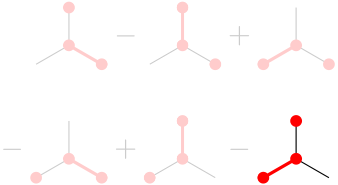
Definition

A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

Many things can be built from a kind of local class we call a *star class*.



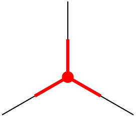


Definition

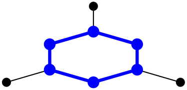
A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

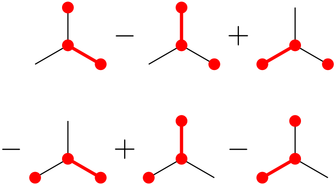
Many things can be built from a kind of local class we call a *star class*.



Y



$B_2(Y)$

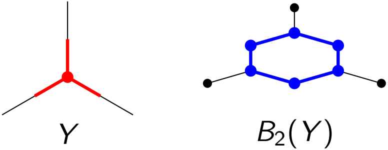


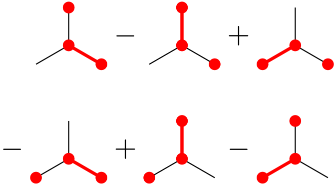
Definition

A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Star classes

Many things can be built from a kind of local class we call a *star class*.





Definition

A class pushed forward from the cycle in $H_1(B_2(Y))$ is a *star class*.

Lemma

Star classes are nontrivial (they may be 2-torsion).

Ramos' connectivity invariant

Recall that over a field, $\dim H_i(B_k(\Gamma)))$ is eventually polynomial in k.

Ramos' connectivity invariant

Recall that over a field, $\dim H_i(B_k(\Gamma)))$ is eventually polynomial in k. Write N_Γ^i for the degree of this polynomial.

Ramos' connectivity invariant

Recall that over a field, $\dim H_i(B_k(\Gamma))$ is eventually polynomial in k. Write N^i_Γ for the degree of this polynomial.

Definition (Ramos' invariant [all vertices valence ≥ 3 case])

$$\Delta^i_\Gamma = \max_{\substack{W \subset V(\Gamma) \\ |W| = i}} |\pi_0(\Gamma \setminus W)|$$

Ramos' connectivity invariant

Recall that over a field, $\dim H_i(B_k(\Gamma))$ is eventually polynomial in k. Write N^i_Γ for the degree of this polynomial.

Definition (Ramos' invariant [all vertices valence ≥ 3 case]**)**

$$\Delta^i_\Gamma = \max_{\substack{W \subset V(\Gamma) \\ |W| = i}} |\pi_0(\Gamma \setminus W)|$$

Theorem (Ramos)

For Γ a tree, $N^i_\Gamma = \Delta^i_\Gamma - 1$.

Ramos' connectivity invariant

Recall that over a field, $\dim H_i(B_k(\Gamma)))$ is eventually polynomial in k. Write N_Γ^i for the degree of this polynomial.

Definition (Ramos' invariant [all vertices valence ≥ 3 case])

$$\Delta_\Gamma^i = \max_{\substack{W \subset V(\Gamma) \\ |W|=i}} |\pi_0(\Gamma \setminus W)|$$

Theorem (Ramos)

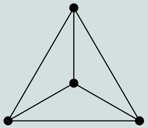
For Γ a tree, $N_\Gamma^i = \Delta_\Gamma^i - 1$.

Theorem (An–D.–Knudsen)

For Γ a graph, $N_\Gamma^i = \Delta_\Gamma^i - 1$

Examples of Ramos' invariant

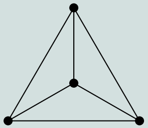
Example



i	Δ_i^Γ	N_i^Γ	$\dim H_i B_k(\Gamma)$	valid for

Examples of Ramos' invariant

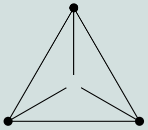
Example



i	Δ_i^Γ	N_i^Γ	$\dim H_i B_k(\Gamma)$	valid for
0	1	0	1	all k

Examples of Ramos' invariant

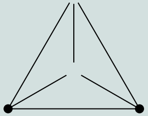
Example



i	Δ_i^Γ	N_i^Γ	$\dim H_i B_k(\Gamma)$	valid for
0	1	0	1	all k
1	1	0	4	$k \geq 2$

Examples of Ramos' invariant

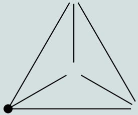
Example



i	Δ_i^Γ	N_i^Γ	$\dim H_i B_k(\Gamma)$	valid for
0	1	0	1	all k
1	1	0	4	$k \geq 2$
2	2	1	$6k - 15$	$k \geq 3$

Examples of Ramos' invariant

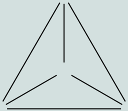
Example



i	Δ_i^Γ	N_i^Γ	$\dim H_i B_k(\Gamma)$	valid for
0	1	0	1	all k
1	1	0	4	$k \geq 2$
2	2	1	$6k - 15$	$k \geq 3$
3	4	3	$4\binom{k-3}{3}$	all k

Examples of Ramos' invariant

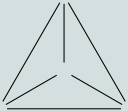
Example



i	Δ_i^Γ	N_i^Γ	$\dim H_i B_k(\Gamma)$	valid for
0	1	0	1	all k
1	1	0	4	$k \geq 2$
2	2	1	$6k - 15$	$k \geq 3$
3	4	3	$4\binom{k-3}{3}$	all k
4	6	5	$\binom{k-3}{5}$	all k

Examples of Ramos' invariant

Example



i	Δ_i^Γ	N_i^Γ	$\dim H_i B_k(\Gamma)$	valid for
0	1	0	1	all k
1	1	0	4	$k \geq 2$
2	2	1	$6k - 15$	$k \geq 3$
3	4	3	$4\binom{k-3}{3}$	all k
4	6	5	$\binom{k-3}{5}$	all k
≥ 5	$-\infty$	$-\infty$	0	all k

Lower bound: $N_\Gamma^i \geq \Delta_\Gamma^i - 1$

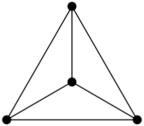
Idea to prove lower bound

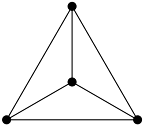
Find a torus α of star classes at $W \subset V$ whose stabilizations grow quickly.

Lower bound: $N_\Gamma^i \geq \Delta_\Gamma^i - 1$

Idea to prove lower bound

Find a torus α of star classes at $W \subset V$ whose stabilizations grow quickly.

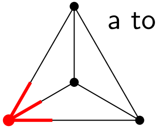




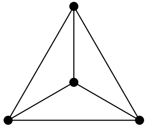
Lower bound: $N_\Gamma^i \geq \Delta_\Gamma^i - 1$

Idea to prove lower bound

Find a torus α of star classes at $W \subset V$ whose stabilizations grow quickly.



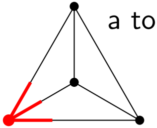
a torus α with $|W| = 1$



Lower bound: $N_\Gamma^i \geq \Delta_\Gamma^i - 1$

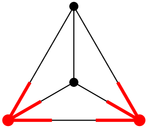
Idea to prove lower bound

Find a torus α of star classes at $W \subset V$ whose stabilizations grow quickly.



a torus α with $|W| = 1$

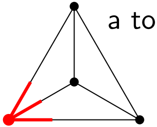
a torus α with $|W| = 2$



Lower bound: $N_\Gamma^i \geq \Delta_\Gamma^i - 1$

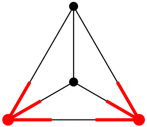
Idea to prove lower bound

Find a torus α of star classes at $W \subset V$ whose stabilizations grow quickly.



a torus α with $|W| = 1$

a torus α with $|W| = 2$



Lemma

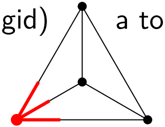
If each star touches two or more components of $\Gamma \setminus W$, then α is rigid.

Lower bound: $N^i_\Gamma \geq \Delta^i_\Gamma - 1$

Idea to prove lower bound

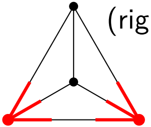
Find a torus α of star classes at $W \subset V$ whose stabilizations grow quickly.

(not rigid)



a torus α with $|W| = 1$

(rigid)



a torus α with $|W| = 2$

Lemma

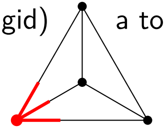
If each star touches two or more components of $\Gamma \setminus W$, then α is rigid.

Lower bound: $N_\Gamma^i \geq \Delta_\Gamma^i - 1$

Idea to prove lower bound

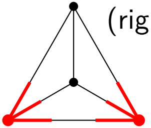
Find a torus α of star classes at $W \subset V$ whose stabilizations grow quickly.

(not rigid)



a torus α with $|W| = 1$

a torus α with $|W| = 2$



(rigid)

Lemma

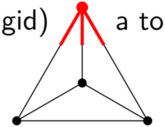
If each star touches two or more components of $\Gamma \setminus W$*, then* α *is rigid.*

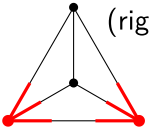
I.e., every representative of α contains generators blocking all of W.

Lower bound: $N_\Gamma^i \geq \Delta_\Gamma^i - 1$

Idea to prove lower bound

Find a torus α of star classes at $W \subset V$ whose stabilizations grow quickly.

(not rigid)  a torus α with $|W| = 1$

a torus α with $|W| = 2$  (rigid)

Lemma

If each star touches two or more components of $\Gamma \setminus W$, then α is rigid.

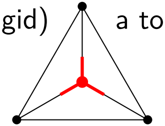
I.e., every representative of α contains generators blocking all of W.

Lower bound: $N_\Gamma^i \geq \Delta_\Gamma^i - 1$

Idea to prove lower bound

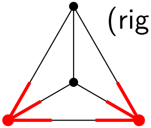
Find a torus α of star classes at $W \subset V$ whose stabilizations grow quickly.

(not rigid)



a torus α with $|W| = 1$

a torus α with $|W| = 2$



(rigid)

Lemma

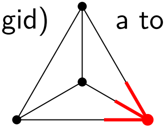
If each star touches two or more components of $\Gamma \setminus W$, then α is rigid.

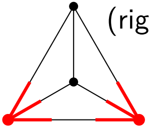
I.e., every representative of α contains generators blocking all of W.

Lower bound: $N_\Gamma^i \geq \Delta_\Gamma^i - 1$

Idea to prove lower bound

Find a torus α of star classes at $W \subset V$ whose stabilizations grow quickly.

(not rigid)  a torus α with $|W| = 1$

a torus α with $|W| = 2$  (rigid)

Lemma

If each star touches two or more components of $\Gamma \setminus W$, then α is rigid.

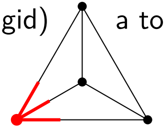
I.e., every representative of α contains generators blocking all of W.

Lower bound: $N_\Gamma^i \geq \Delta_\Gamma^i - 1$

Idea to prove lower bound

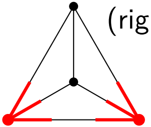
Find a torus α of star classes at $W \subset V$ whose stabilizations grow quickly.

(not rigid)



a torus α with $|W| = 1$

a torus α with $|W| = 2$



(rigid)

Lemma

If each star touches two or more components of $\Gamma \setminus W$, then α is rigid.

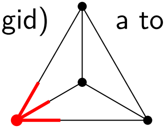
I.e., every representative of α contains generators blocking all of W.

Lower bound: $N_\Gamma^i \geq \Delta_\Gamma^i - 1$

Idea to prove lower bound

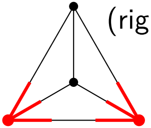
Find a torus α of star classes at $W \subset V$ whose stabilizations grow quickly.

(not rigid)



a torus α with $|W| = 1$

a torus α with $|W| = 2$



(rigid)

Lemma

If each star touches two or more components of $\Gamma \setminus W$, then α is rigid.

I.e., every representative of α contains generators blocking all of W.

Lemma

The stabilizations of rigid α look like polynomials in $\pi_0(\Gamma \setminus W)$.

Lower bound: $N_\Gamma^i \geq \Delta_\Gamma^i - 1$

Lemma

If each star touches two or more components of $\Gamma \setminus W$, then α is rigid.

Lemma

The stabilizations of rigid α look like polynomials in $\pi_0(\Gamma \setminus W)$.

Lower bound: $N_\Gamma^i \geq \Delta_\Gamma^i - 1$

Lemma

If each star touches two or more components of $\Gamma \setminus W$, then α is rigid.

Lemma

The stabilizations of rigid α look like polynomials in $\pi_0(\Gamma \setminus W)$.

$$\text{(reminder)} \qquad \Delta_\Gamma^i = \max_{\substack{W \subset V(\Gamma) \\ |W| = i}} |\pi_0(\Gamma \setminus W)|$$

Lower bound: $N_\Gamma^i \geq \Delta_\Gamma^i - 1$

Lemma

If each star touches two or more components of $\Gamma \setminus W$, then α is rigid.

Lemma

The stabilizations of rigid α look like polynomials in $\pi_0(\Gamma \setminus W)$.

(reminder) $$\Delta_\Gamma^i = \max_{\substack{W \subset V(\Gamma) \\ |W| = i}} |\pi_0(\Gamma \setminus W)|$$

Proof of lower bound.

For $|W| \neq 1$, if W realizes this maximum then every vertex of W touches two or more components of $\Gamma \setminus W$.

Lower bound: $N_\Gamma^i \geq \Delta_\Gamma^i - 1$

Lemma

If each star touches two or more components of $\Gamma \setminus W$, then α is rigid.

Lemma

The stabilizations of rigid α look like polynomials in $\pi_0(\Gamma \setminus W)$.

$$\text{(reminder)} \qquad \Delta_\Gamma^i = \max_{\substack{W \subset V(\Gamma) \\ |W| = i}} |\pi_0(\Gamma \setminus W)|$$

Proof of lower bound.

For $|W| \neq 1$, if W realizes this maximum then every vertex of W touches two or more components of $\Gamma \setminus W$.
Then we can build a rigid torus α. □

Upper bound: $N_\Gamma^i \leq \Delta_\Gamma^i - 1$

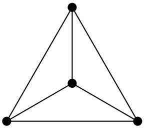
Idea to prove upper bound

Use rigidity at vertices to decompose $H_i(B(\Gamma))$.

Upper bound: $N_\Gamma^i \leq \Delta_\Gamma^i - 1$

Idea to prove upper bound

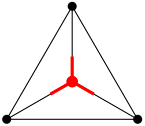
Use rigidity at vertices to decompose $H_i(B(\Gamma))$.



Upper bound: $N_\Gamma^i \leq \Delta_\Gamma^i - 1$

Idea to prove upper bound

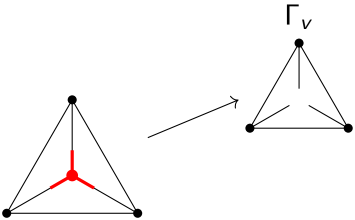
Use rigidity at vertices to decompose $H_i(B(\Gamma))$.



Upper bound: $N_\Gamma^i \leq \Delta_\Gamma^i - 1$

Idea to prove upper bound

Use rigidity at vertices to decompose $H_i(B(\Gamma))$.

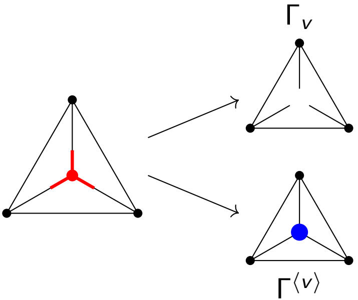


- The v-rigid part of $H_i(B_k(\Gamma))$ grows like $H_{i-1}(B_{k-1}(\Gamma_v))$.

Upper bound: $N_\Gamma^i \leq \Delta_\Gamma^i - 1$

Idea to prove upper bound

Use rigidity at vertices to decompose $H_i(B(\Gamma))$.

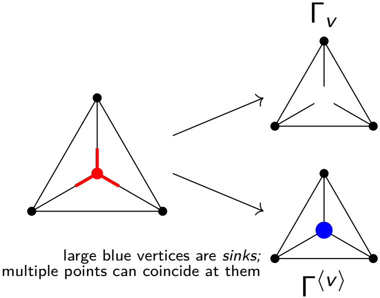


- The v-rigid part of $H_i(B_k(\Gamma))$ grows like $H_{i-1}(B_{k-1}(\Gamma_v))$.
- The non-v-rigid in $H_i(B_k(\Gamma))$ grows like $H_i(B_k(\Gamma^{\langle v \rangle}))$.

Upper bound: $N_\Gamma^i \leq \Delta_\Gamma^i - 1$

Idea to prove upper bound

Use rigidity at vertices to decompose $H_i(B(\Gamma))$.

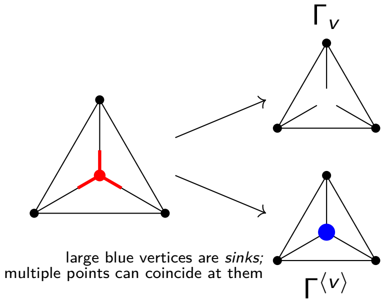


- The v-rigid part of $H_i(B_k(\Gamma))$ grows like $H_{i-1}(B_{k-1}(\Gamma_v))$.
- The non-v-rigid in $H_i(B_k(\Gamma))$ grows like $H_i(B_k(\Gamma^{\langle v \rangle}))$.

Upper bound: $N_\Gamma^i \leq \Delta_\Gamma^i - 1$

Idea to prove upper bound

Use rigidity recursively at vertices to decompose $H_i(B(\Gamma))$.

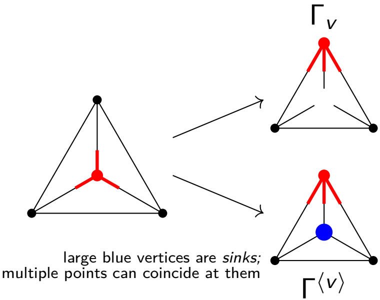


- The v-rigid part of $H_i(B_k(\Gamma))$ grows like $H_{i-1}(B_{k-1}(\Gamma_v))$.
- The non-v-rigid in $H_i(B_k(\Gamma))$ grows like $H_i(B_k(\Gamma^{\langle v \rangle}))$.

Upper bound: $N_\Gamma^i \leq \Delta_\Gamma^i - 1$

Idea to prove upper bound

Use rigidity recursively at vertices to decompose $H_i(B(\Gamma))$.

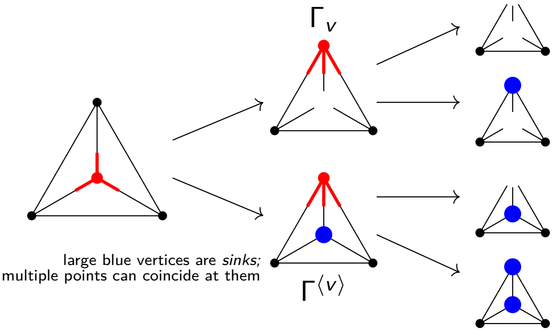


- The v-rigid part of $H_i(B_k(\Gamma))$ grows like $H_{i-1}(B_{k-1}(\Gamma_v))$.
- The non-v-rigid in $H_i(B_k(\Gamma))$ grows like $H_i(B_k(\Gamma^{\langle v \rangle}))$.

Upper bound: $N_\Gamma^i \leq \Delta_\Gamma^i - 1$

Idea to prove upper bound

Use rigidity recursively at vertices to decompose $H_i(B(\Gamma))$.

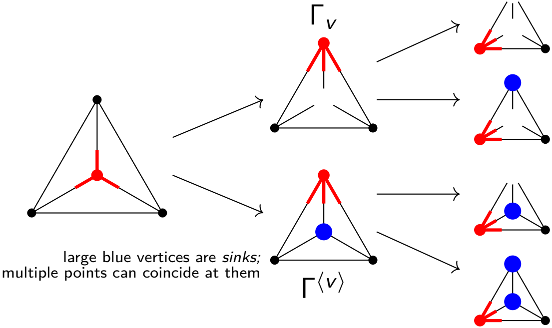


- The v-rigid part of $H_i(B_k(\Gamma))$ grows like $H_{i-1}(B_{k-1}(\Gamma_v))$.
- The non-v-rigid in $H_i(B_k(\Gamma))$ grows like $H_i(B_k(\Gamma^{\langle v \rangle}))$.

Upper bound: $N_\Gamma^i \leq \Delta_\Gamma^i - 1$

Idea to prove upper bound

Use rigidity recursively at vertices to decompose $H_i(B(\Gamma))$.

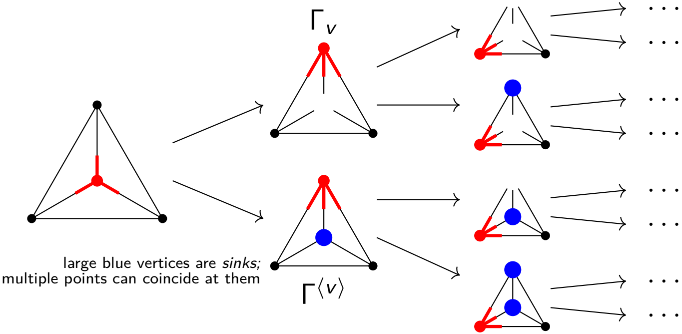


- The v-rigid part of $H_i(B_k(\Gamma))$ grows like $H_{i-1}(B_{k-1}(\Gamma_v))$.
- The non-v-rigid in $H_i(B_k(\Gamma))$ grows like $H_i(B_k(\Gamma^{\langle v \rangle}))$.

Upper bound: $N^i_\Gamma \leq \Delta^i_\Gamma - 1$

Idea to prove upper bound

Use rigidity recursively at vertices to decompose $H_i(B(\Gamma))$.



- The v-rigid part of $H_i(B_k(\Gamma))$ grows like $H_{i-1}(B_{k-1}(\Gamma_v))$.
- The non-v-rigid in $H_i(B_k(\Gamma))$ grows like $H_i(B_k(\Gamma^{\langle v \rangle}))$.

- The v-rigid part of $H_i(B_k(\Gamma))$ grows like $H_{i-1}(B_{k-1}(\Gamma_v))$.
- The non-v-rigid in $H_i(B_k(\Gamma))$ grows like $H_i(B_k(\Gamma^{\langle v \rangle}))$.

- The v-rigid part of $H_i(B_k(\Gamma))$ grows like $H_{i-1}(B_{k-1}(\Gamma_v))$.
- The non-v-rigid in $H_i(B_k(\Gamma))$ grows like $H_i(B_k(\Gamma^{\langle v \rangle}))$.

Proof outline.

Recursively we can reduce to

$$\bigoplus H_0(B_{k-i}(\Gamma^{\langle v_1 \rangle, \ldots \langle v_j \rangle}_{w_1, \ldots, w_i})).$$

- The v-rigid part of $H_i(B_k(\Gamma))$ grows like $H_{i-1}(B_{k-1}(\Gamma_v))$.
- The non-v-rigid in $H_i(B_k(\Gamma))$ grows like $H_i(B_k(\Gamma^{\langle v \rangle}))$.

Proof outline.

Recursively we can reduce to

$$\bigoplus H_0(B_{k-i}(\Gamma^{\langle v_1 \rangle, \ldots \langle v_j \rangle}_{w_1, \ldots, w_i})).$$

The Γ^V_W summand grows like

$$(k-i)^{|\pi_0(\Gamma \setminus W)|-1}.$$

□

- The v-rigid part of $H_i(B_k(\Gamma))$ grows like $H_{i-1}(B_{k-1}(\Gamma_v))$.
- The non-v-rigid in $H_i(B_k(\Gamma))$ grows like $H_i(B_k(\Gamma^{\langle v \rangle}))$.

Proof outline.

Recursively we can reduce to

$$\bigoplus H_0(B_{k-i}(\Gamma^{\langle v_1 \rangle, \ldots \langle v_j \rangle}_{w_1, \ldots, w_i})).$$

The Γ^V_W summand grows like

$$(k-i)^{|\pi_0(\Gamma \setminus W)|-1}.$$

□

One problem with this approach

- The v-rigid part of $H_i(B_k(\Gamma))$ grows like $H_{i-1}(B_{k-1}(\Gamma_v))$.
- The non-v-rigid in $H_i(B_k(\Gamma))$ grows like $H_i(B_k(\Gamma^{\langle v \rangle}))$.

Proof outline.

Recursively we can reduce to

$$\bigoplus H_0(B_{k-i}(\Gamma^{\langle v_1 \rangle, \ldots \langle v_j \rangle}_{w_1, \ldots, w_i})).$$

The Γ^V_W summand grows like

$$(k-i)^{|\pi_0(\Gamma \setminus W)|-1}.$$

□

One problem with this approach

I'm totally lying about the decomposition.

- The v-rigid part of $H_i(B_k(\Gamma))$ grows like $H_{i-1}(B_{k-1}(\Gamma_v))$.
- The non-v-rigid in $H_i(B_k(\Gamma))$ grows like $H_i(B_k(\Gamma^{\langle v \rangle}))$.

Proof outline.

Recursively we can reduce to

$$\bigoplus H_0(B_{k-i}(\Gamma^{\langle v_1 \rangle, \ldots \langle v_j \rangle}_{w_1, \ldots, w_i})).$$

The Γ^V_W summand grows like

$$(k-i)^{|\pi_0(\Gamma \setminus W)|-1}.$$

□

One problem with this approach

I'm totally lying about the decomposition.
The real groups involved are too hard to calculate explicitly.

- The v-rigid part of $H_i(B_k(\Gamma))$ grows like $H_{i-1}(B_{k-1}(\Gamma_v))$.
- The non-v-rigid in $H_i(B_k(\Gamma))$ grows like $H_i(B_k(\Gamma^{\langle v \rangle}))$.

Proof outline.

Recursively we can reduce to

$$\bigoplus H_0(B_{k-i}(\Gamma^{\langle v_1 \rangle, \ldots \langle v_j \rangle}_{w_1, \ldots, w_i})).$$

The Γ^V_W summand grows like

$$(k-i)^{|\pi_0(\Gamma \setminus W)|-1}.$$

□

One problem with this approach

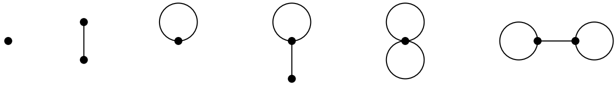
I'm totally lying about the decomposition.
The real groups involved are too hard to calculate explicitly.

But with the lies I told, you can do well enough to get an upper bound.

Formality

Theorem

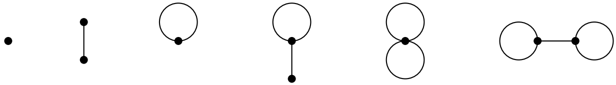
$C_*(B_*(\Gamma))$ is $\mathbb{Z}[E]$-formal if and only if Γ is small.



Formality

Theorem

$C_*(B_*(\Gamma))$ is $\mathbb{Z}[E]$-formal if and only if Γ is small.



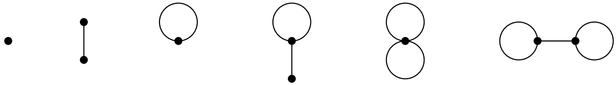
Question

- Why do we care?

Formality

Theorem

$C_*(B_*(\Gamma))$ is $\mathbb{Z}[E]$-formal if and only if Γ is small.



Question

- Why do we care?

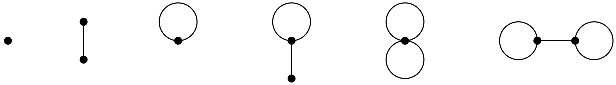
Answer

- Computation: assembling graphs

Formality

Theorem

$C_*(B_*(\Gamma))$ is $\mathbb{Z}[E]$-formal if and only if Γ is small.



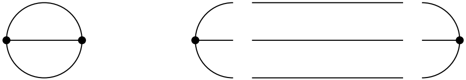
Question

- Why do we care?

Answer

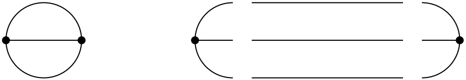
- Computation: assembling graphs
- Higher invariants

(Non)-formality and graph assembly



$$S(\Gamma_1 \sqcup_{E'} \Gamma_2) \cong S(\Gamma_1) \otimes_{\mathbb{Z}[E']} S(\Gamma_2)$$

(Non)-formality and graph assembly

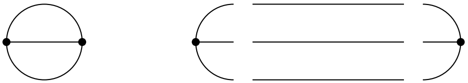


$$S(\Gamma_1 \sqcup_{E'} \Gamma_2) \;\cong\; S(\Gamma_1) \quad \otimes_{\mathbb{Z}[E']} \quad S(\Gamma_2)$$

If we had formality, we could conclude:

$$H_*B(\Gamma_1 \sqcup_{E'} \Gamma_2) \cong \mathrm{Tor}_*^{\mathbb{Z}[E']}(H_*B(\Gamma_1), H_*B(\Gamma_2)).$$

(Non)-formality and graph assembly



$$S(\Gamma_1 \sqcup_{E'} \Gamma_2) \cong S(\Gamma_1) \otimes_{\mathbb{Z}[E']} S(\Gamma_2)$$

If we had formality, we could conclude:

$$H_* B(\Gamma_1 \sqcup_{E'} \Gamma_2) \cong \mathrm{Tor}_*^{\mathbb{Z}[E']}(H_* B(\Gamma_1), H_* B(\Gamma_2)).$$

Instead this is just page two of a Künneth spectral sequence in general.

Local and global nonformality

Ideology

- Locally star classes are not in the image of the H_* stabilization maps.

Local and global nonformality

Ideology

- Locally star classes are not in the image of the H_* stabilization maps.
- But star *cycles* are always in the image of the S_* stabilization maps.

Local and global nonformality

Ideology

- Locally star classes are not in the image of the H_* stabilization maps.
- But star *cycles* are always in the image of the S_* stabilization maps.
- Therefore star classes obstruct formality locally.

Local and global nonformality

Ideology

- Locally star classes are not in the image of the H_* stabilization maps.
- But star *cycles* are always in the image of the S_* stabilization maps.
- Therefore star classes obstruct formality locally.
- Therefore *rigid* star classes obstruct formality globally.

Local and global nonformality

Ideology

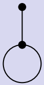
- Locally star classes are not in the image of the H_* stabilization maps.
- But star *cycles* are always in the image of the S_* stabilization maps.
- Therefore star classes obstruct formality locally.
- Therefore *rigid* star classes obstruct formality globally.
- Rigidify a star class at v by combining it with a “top class” in $S(\Gamma_v)$.

Local and global nonformality

Ideology

- Locally star classes are not in the image of the H_* stabilization maps.
- But star *cycles* are always in the image of the S_* stabilization maps.
- Therefore star classes obstruct formality locally.
- Therefore *rigid* star classes obstruct formality globally.
- Rigidify a star class at v by combining it with a “top class” in $S(\Gamma_v)$.

Counterexample

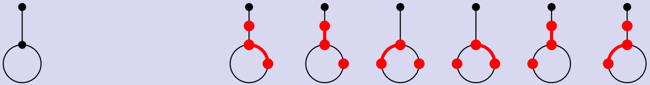


Local and global nonformality

Ideology

- Locally star classes are not in the image of the H_* stabilization maps.
- But star *cycles* are always in the image of the S_* stabilization maps.
- Therefore star classes obstruct formality locally.
- Therefore *rigid* star classes obstruct formality globally.
- Rigidify a star class at v by combining it with a “top class” in $S(\Gamma_v)$.

Counterexample

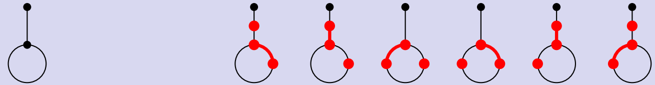


Local and global nonformality

Ideology

- Locally star classes are not in the image of the H_* stabilization maps.
- But star *cycles* are always in the image of the S_* stabilization maps.
- Therefore star classes obstruct formality locally.
- Therefore *rigid* star classes obstruct formality globally.
- Rigidify a star class at v by combining it with a "top class" in $S(\Gamma_v)$.

Counterexample



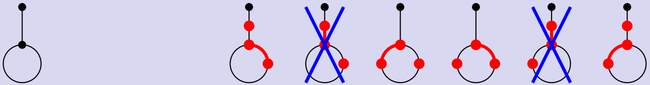
What goes wrong? The star class is a combination of stabilized classes.

Local and global nonformality

Ideology

- Locally star classes are not in the image of the H_* stabilization maps.
- But star *cycles* are always in the image of the S_* stabilization maps.
- Therefore star classes obstruct formality locally.
- Therefore *rigid* star classes obstruct formality globally.
- Rigidify a star class at v by combining it with a "top class" in $S(\Gamma_v)$.

Counterexample



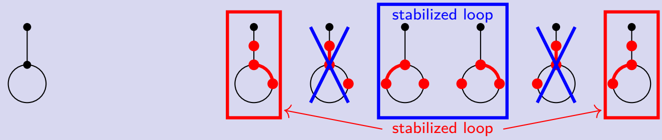
What goes wrong? The star class is a combination of stabilized classes.

Local and global nonformality

Ideology

- Locally star classes are not in the image of the H_* stabilization maps.
- But star *cycles* are always in the image of the S_* stabilization maps.
- Therefore star classes obstruct formality locally.
- Therefore *rigid* star classes obstruct formality globally.
- Rigidify a star class at v by combining it with a "top class" in $S(\Gamma_v)$.

Counterexample



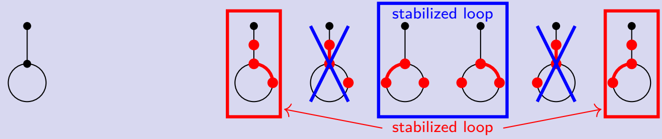
What goes wrong? The star class is a combination of stabilized classes.

Local and global nonformality

Ideology

- Locally star classes are not in the image of the H_* stabilization maps.
- But star *cycles* are always in the image of the S_* stabilization maps.
- Therefore star classes obstruct formality locally.
- Therefore *rigid* *indecomposable* star classes obstruct formality globally.
- Rigidify a star class at v by combining it with a "top class" in $S(\Gamma_v)$.

Counterexample



What goes wrong? The star class is a combination of stabilized classes.

Building rigid indecomposable star classes

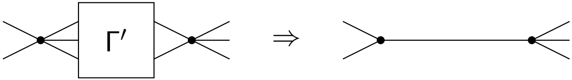
Definition

A *surgery* replaces a subgraph attached at two vertices with a single edge.

Building rigid indecomposable star classes

Definition

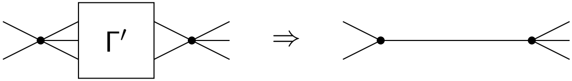
A *surgery* replaces a subgraph attached at two vertices with a single edge.



Building rigid indecomposable star classes

Definition

A *surgery* replaces a subgraph attached at two vertices with a single edge.



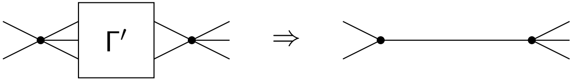
Lemma

Surgery reflects non-formality.

Building rigid indecomposable star classes

Definition

A *surgery* replaces a subgraph attached at two vertices with a single edge.



Lemma

Surgery reflects non-formality.

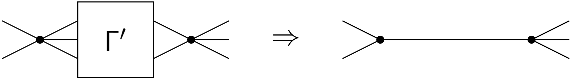
Lemma

Any large graph can be surgered to one of the following.

Building rigid indecomposable star classes

Definition

A *surgery* replaces a subgraph attached at two vertices with a single edge.

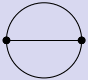


Lemma

Surgery reflects non-formality.

Lemma

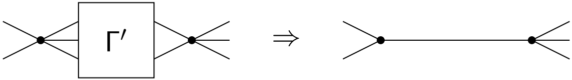
Any large graph can be surgered to one of the following.



Building rigid indecomposable star classes

Definition

A *surgery* replaces a subgraph attached at two vertices with a single edge.



Lemma

Surgery reflects non-formality.

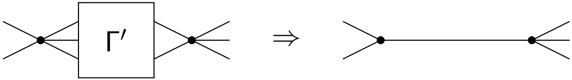
Lemma

Any large graph can be surgered to one of the following.

Building rigid indecomposable star classes

Definition

A *surgery* replaces a subgraph attached at two vertices with a single edge.

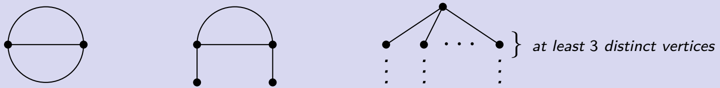


Lemma

Surgery reflects non-formality.

Lemma

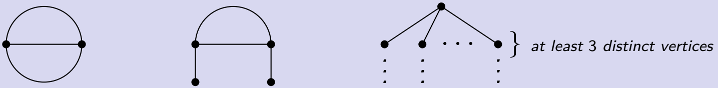
Any large graph can be surgered to one of the following.



Using surgery

Lemma

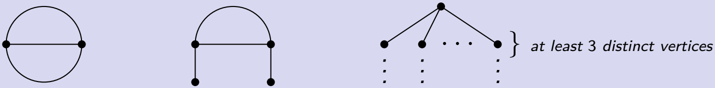
Any large graph can be surgered to one of the following.



Using surgery

Lemma

Any large graph can be surgered to one of the following.



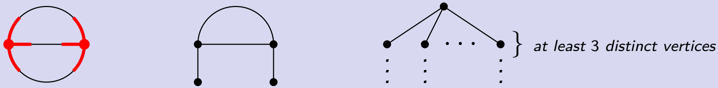
Lemma

In all three cases we can build a rigid indecomposable star class.

Using surgery

Lemma

Any large graph can be surgered to one of the following.



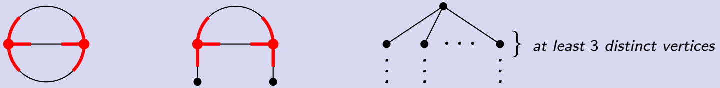
Lemma

In all three cases we can build a rigid indecomposable star class.

Using surgery

Lemma

Any large graph can be surgered to one of the following.



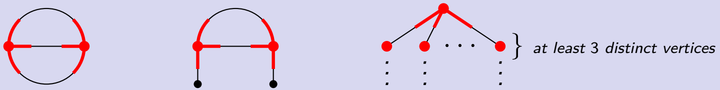
Lemma

In all three cases we can build a rigid indecomposable star class.

Using surgery

Lemma

Any large graph can be surgered to one of the following.



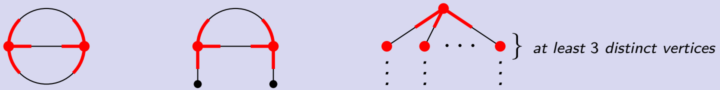
Lemma

In all three cases we can build a rigid indecomposable star class.

Using surgery

Lemma

Any large graph can be surgered to one of the following.



Lemma

In all three cases we can build a rigid indecomposable star class.

Corollary

Large graphs are non-formal.

Thank you for your attention.