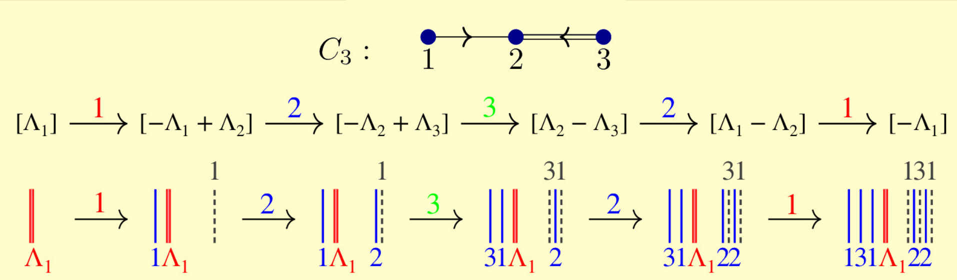
wKLRW algebras and crystals

Or: From path to strings

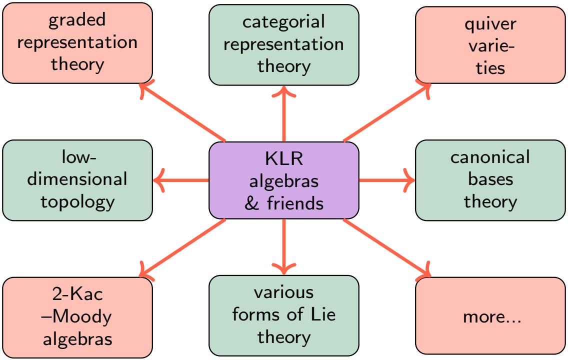
Daniel Tubbenhauer



Joint with Andrew Mathas

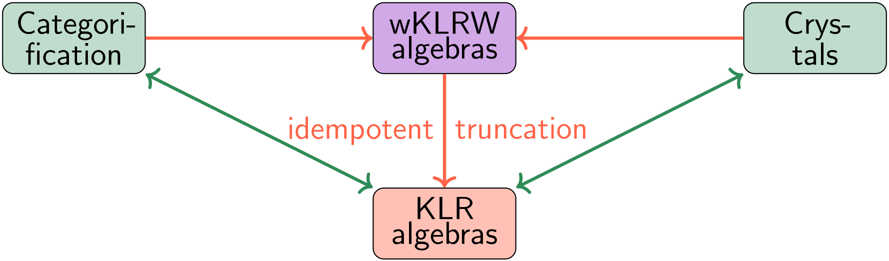
September 2022

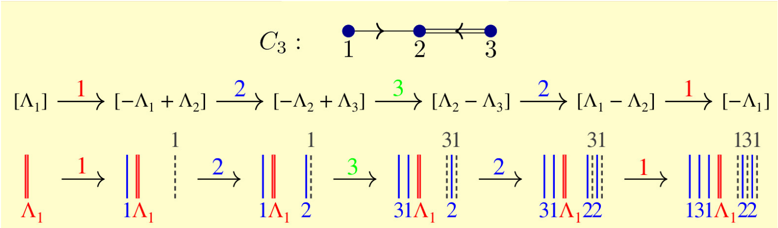
What? Why? How?



- **Khovanov–Lauda–Rouquier ∼2008 + many others** (including many people here) KLR algebras are at the heart of categorical representation theory
- Problem These are actually really complicated!
- Goal Try to find nice ("cellular") bases for them

What? Why? How?



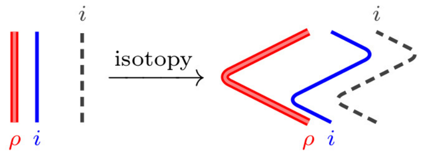


Use a richer combinatorics which is somewhat easier although more sophisticated

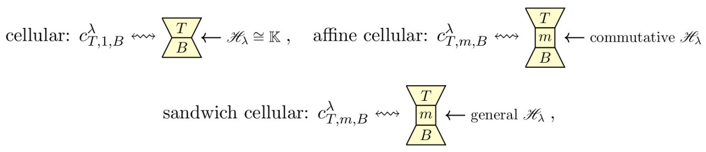
Categ… fica… …rys- …als

Today

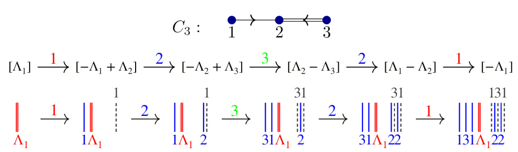
1) The diagram combinatorics



2) Sandwich cellularity



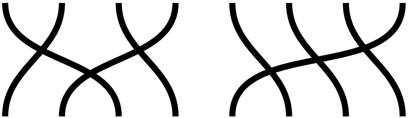
3) Bases and crystals



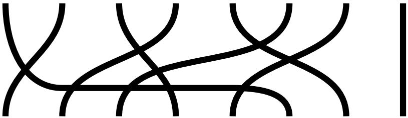
Use a ri… …isticated

String diagrams – the baby case

Connect eight points at the bottom with eight points at the top:

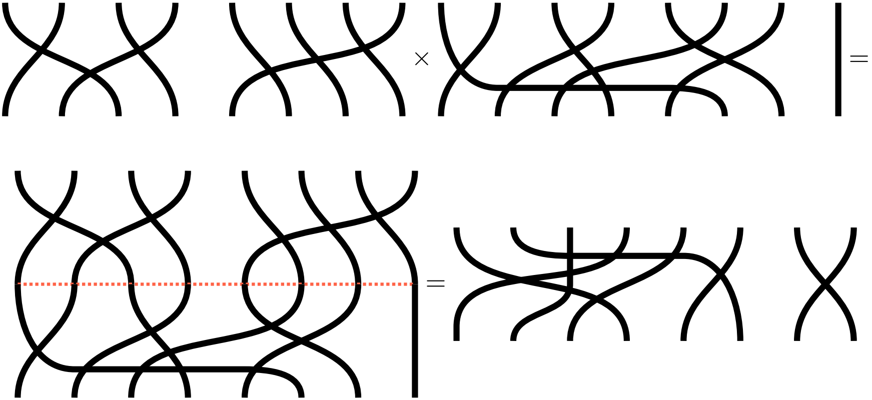
(1243)(5876) ⇜⇝ 

or

(12436)(57)(8) ⇜⇝ 

We just invented the symmetric group S_8 on $\{1, ..., 8\}$

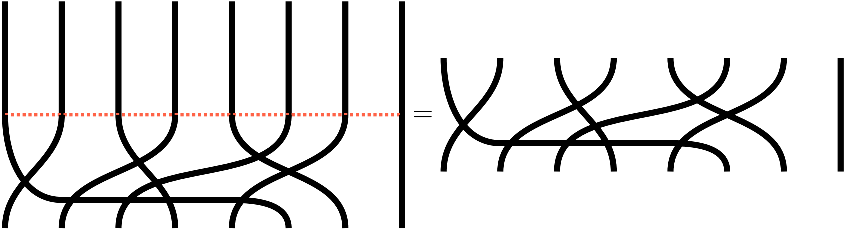
String diagrams – the baby case



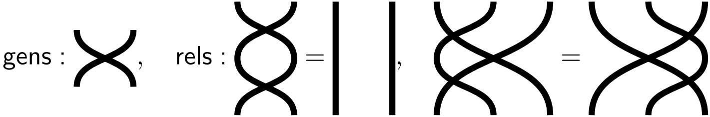
My multiplication rule for gh is “stack g on top of h”

String diagrams – the baby case

- We clearly have $g(hf) = (gh)f$
- There is a do nothing operation $1g = g = g1$



- Generators–relations (the Reidemeister moves)

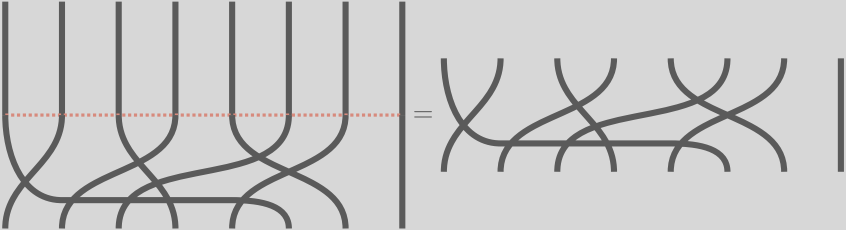


String diagrams –

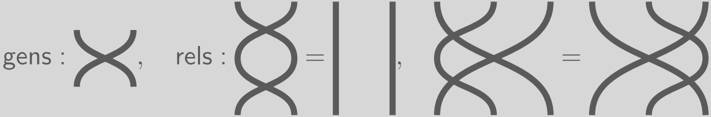
The bait

In diagram algebras relations, properties, etc. become visually clear

- We clearly have
- There is a do nothing operation $1g = g = g1$



- Generators–relations (the Reidemeister moves)



String diagrams –

The bait

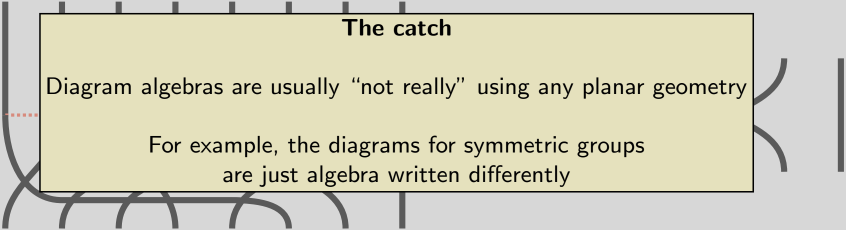
In diagram algebras relations, properties, etc. become visually clear

- We clearly have
- There is a do nothing operation $1g = g = g1$

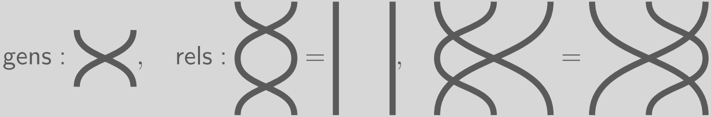
The catch

Diagram algebras are usually "not really" using any planar geometry

For example, the diagrams for symmetric groups are just algebra written differently



- Generators–relations (the Reidemeister moves)

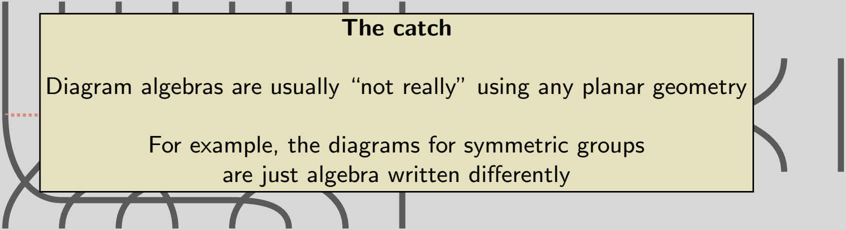


String diagrams –

The bait

In diagram algebras relations, properties, etc. become visually clear

- We clearly have
- There is a do nothing operation $1g = g = g1$



The catch

Diagram algebras are usually "not really" using any planar geometry

For example, the diagrams for symmetric groups are just algebra written differently

- Gene

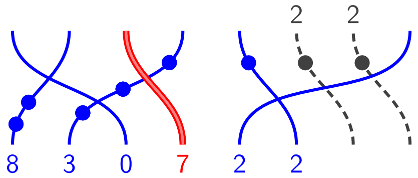
Idea (Webster ∼2012)

Define a diagram algebra that uses the distance in $\mathbb{R}^2$

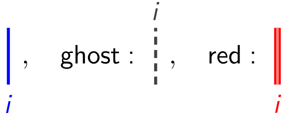
The result is called weighted KLRW (wKLRW) algebra

These are "planar-geometrically symmetric group diagram algebras"

Weighted string diagrams

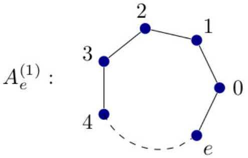


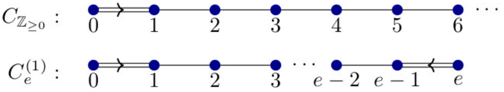
- Strings come in three types, solid, ghost and red



- Strings are labeled, and solid and ghost strings can carry dots
- Red strings anchor the diagram (red strings ⇝ level)
- Otherwise no difference to symmetric group diagrams

Weighted string diagrams



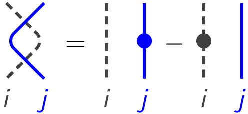


Examples of quivers Γ

An additional orientation fixes signs

- The strings are labeled by $i \in I$ from a fixed quiver $\Gamma = (I, E)$
- The relations (that I am not going to show you ;-)) depend on $e \in E$, e.g.:

"Reidemeister II with error term":

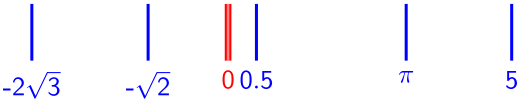


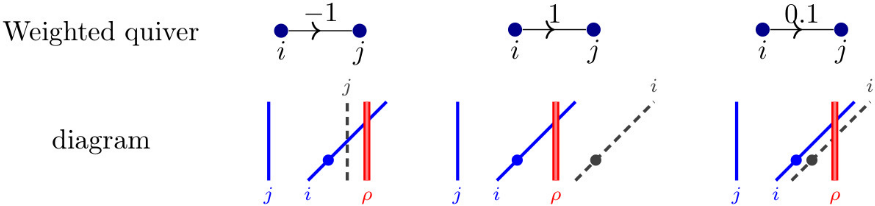
if $i \to j$

Weighted string diagrams

I usually never use the number π in a talk ;-)

$X = (-2\sqrt{3}, -\sqrt{2}, 0.5, \pi, 5) \leftrightsquigarrow$

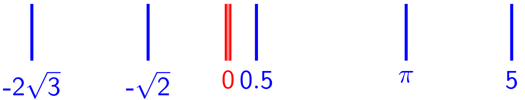




- Choose endpoints $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, $\rho \in \mathbb{R}^\ell$ for the solid and red strings
- Choose a weighting $\boldsymbol{\sigma}\colon E \to \mathbb{R}_{\neq 0}$ of the underlying graph $\Gamma = (I, E)$
- The wKLRW algebra crucially depends on these choices of endpoints! This is very different from "usual diagram algebras"

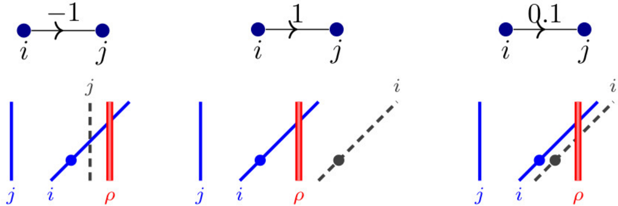
Weighted string diagrams

$X = (\text{-}2\sqrt{3}, -\sqrt{2}, 0.5, \pi, 5) \leftrightsquigarrow$



Weighted quiver

diagram



Weighting = ghost shifts

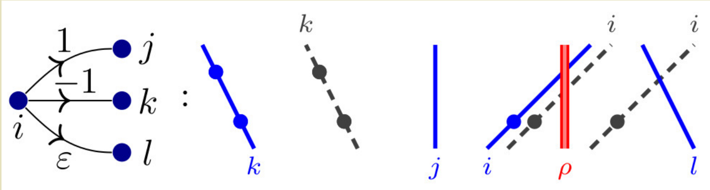
For $\epsilon\colon i \to j, \sigma_\epsilon > 0$, all solid i-strings get a ghost shifted $|\sigma_\epsilon|$ units and mimicking it
For $\epsilon\colon i \to j, \sigma_\epsilon < 0$, all solid j-strings get a ghost shifted $|\sigma_\epsilon|$ units and mimicking it

- The wKLR … points! This is very differ… usual diagram algebras

This "asymmetric" definition, always shifting rightwards makes life a bit more convenient but is not essential

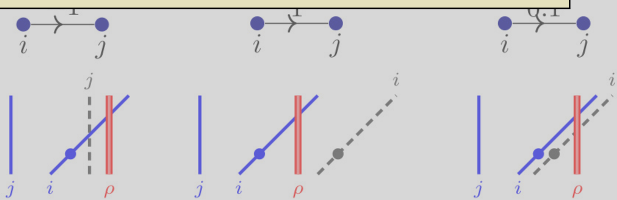
Weighted string diagrams

$X = ($ ... 5



Weighted quiver

diagram



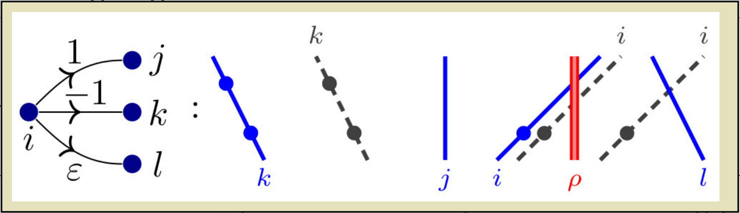
Weighting = ghost shifts

For $\epsilon\colon i \to j, \sigma_\epsilon > 0$, all solid i-strings get a ghost shifted $|\sigma_\epsilon|$ units and mimicking it
For $\epsilon\colon i \to j, \sigma_\epsilon < 0$, all solid j-strings get a ghost shifted $|\sigma_\epsilon|$ units and mimicking it

- The wKLRW algebra crucially depends on these choices of endpoints! This is very different from "usual diagram algebras"

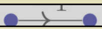
Weighted string diagrams

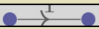
$X = ($

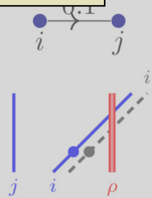


5

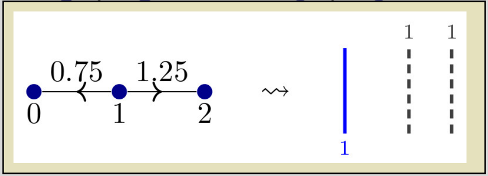
Weighted quiver







diagram



Weighting = ghost shifts

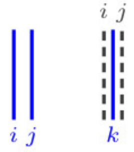
For $\epsilon\colon i \to j, \sigma_\epsilon > 0$, all solid i-strings get a ghost shifted $|\sigma_\epsilon|$ units and mimicking it
For $\epsilon\colon i \to j, \sigma_\epsilon < 0$, all solid j-strings get a ghost shifted $|\sigma_\epsilon|$ units and mimicking it

- The wKLRW algebra crucially depends on these choices of endpoints! This is very different from "usual diagram algebras"

Weighted string diagrams

$X = (-2\sqrt{3}, -\sqrt{2}, 0.5, \pi, 5) \rightsquigarrow$ 

The following i and j-strings are not close:

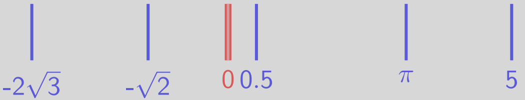


Slogan Ghosts prevent the diagrams from being scale-able as for "usual diagram algebras"

- Choose endpoints $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$, $\rho \in \mathbb{R}^\ell$ for the solid and red strings
- Choose a weighting $\boldsymbol{\sigma}\colon E \to \mathbb{R}_{\neq 0}$ of the underlying graph $\Gamma = (I, E)$
- The wKLRW algebra crucially depends on these choices of endpoints! This is very different from "usual diagram algebras"

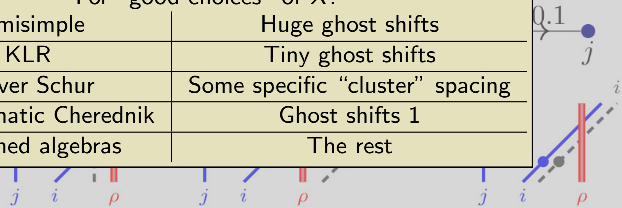
Weighted string diagrams

$X = (\text{-}2\sqrt{3}, -\sqrt{2}, 0.5, \pi, 5) \leftrightsquigarrow$



Weighted d

diagra



For “good choices” of X:

Semisimple	Huge ghost shifts
KLR	Tiny ghost shifts
Quiver Schur	Some specific “cluster” spacing
Diagrammatic Cherednik	Ghost shifts 1
Unnamed algebras	The rest

- Choose endpoints $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, $\rho \in \mathbb{R}^\ell$ for the solid and red strings
- Choose a weighting $\boldsymbol{\sigma}\colon E \to \mathbb{R}_{\neq 0}$ of the underlying graph $\Gamma = (I, E)$
- The wKLRW algebra crucially depends on these choices of endpoints! This is very different from “usual diagram algebras”

Sandwiches

Definition 2A.3. A ***sandwich cell datum*** for $\mathscr{A}$ is a quadruple $\big(\mathcal{P},(\mathcal{T},\mathcal{B}),(\mathscr{H}_\lambda,B_\lambda),C\big)$, where:

- $\mathcal{P}=(\mathcal{P},<_{\mathcal{P}})$ is a poset (the ***middle poset*** with ***sandwich order*** $<_{\mathcal{P}}$),
- $\mathcal{T}=\bigcup_{\lambda\in\mathcal{P}}\mathcal{T}(\lambda)$ and $\mathcal{B}=\bigcup_{\lambda\in\mathcal{P}}\mathcal{B}(\lambda)$ are collections of finite sets (the ***top/bottom sets***),
- For $\lambda\in\mathcal{P}$ we have algebras $\mathscr{H}_\lambda$ (the ***sandwiched algebras***) and bases B_λ of $\mathscr{H}_\lambda$,
- $C\colon \coprod_{\lambda\in\mathcal{P}}\mathcal{T}(\lambda)\times B_\lambda\times\mathcal{B}(\lambda)\to\mathscr{A};(T,m,B)\mapsto c^\lambda_{T,m,B}$ is an injective map,

such that:

(AC$_1$) The set $B_{\mathscr{A}}=\big\{c^\lambda_{T,m,B}\,\big|\,\lambda\in\mathcal{P},T\in\mathcal{T}(\lambda),B\in\mathcal{B}(\lambda),m\in B_\lambda\big\}$ is a basis of $\mathscr{A}$. (We call $B_{\mathscr{A}}$ a ***sandwich cellular basis***.)

(AC$_2$) For all $x\in\mathscr{A}$ there exist scalars $r^x_{TU}\in\mathbb{K}$ that do not depend on B or on m, such that

$$xc^\lambda_{T,m,B}\equiv\sum_{U\in\mathcal{T}(\lambda)n\in B_\lambda}r^x_{TU}c^\lambda_{U,n,B}\pmod{\mathscr{A}^{>_{\mathcal{P}}\lambda}}.\tag{2A.4}$$

Similarly for right multiplication by x.

(AC$_3$) There exists a free $\mathscr{A}$-$\mathscr{H}_\lambda$-bimodule $\Delta(\lambda)$, a free $\mathscr{H}_\lambda$-$\mathscr{A}$-bimodule $\nabla(\lambda)$, and an $\mathscr{A}$-bimodule isomorphism

$$\mathscr{A}_\lambda=\mathscr{A}^{\geq_{\mathcal{P}}\lambda}/\mathscr{A}^{>_{\mathcal{P}}\lambda}\cong\Delta(\lambda)\otimes_{\mathscr{H}_\lambda}\nabla(\lambda).\tag{2A.5}$$

We call $\mathscr{A}_\lambda$ the ***cell algebra***, and $\Delta(\lambda)$ and $\nabla(\lambda)$ left and right ***cell modules***.

The algebra $\mathscr{A}$ is a ***sandwich cellular algebra*** if it has a sandwich cell datum.

Sandwiches

Strategy (Green ∼1950, Brown ∼1953, König–Xi ∼1999, folklore)

ON THE STRUCTURE OF SEMIGROUPS

BY J. A. GREEN

(Received June 1, 1950)

THE SEMISIMPLICITY OF ω_f^{n}*

BY WILLIAM P. BROWN

(Received December 6, 1953)
(Revised November 15, 1954)

GENERALIZED MATRIX ALGEBRAS

W. P. BROWN

CELLULAR ALGEBRAS: INFLATIONS AND MORITA EQUIVALENCES

STEFFEN KÖNIG AND CHANGCHANG XI

Almost all of the theory of cellular algebras works verbatim with one difference:
All relevant λ give as many simples as $\mathscr{H}_\lambda$ has

Sandwiches

Definition 2. … $C)$, where:

- $\mathcal{P} = ($ …
- $\mathcal{T} = \bigcup$ … *m sets*),
- For λ … λ,
- $C\colon \coprod$ …

such that:

(AC$_1$) The se… e call $B_{\mathscr{A}}$ a *sandw*…

(AC$_2$) For all … h that

(2A.4)

Simila…

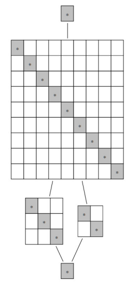
(AC$_3$) There … $\mathscr{A}$-bimodule isomor…

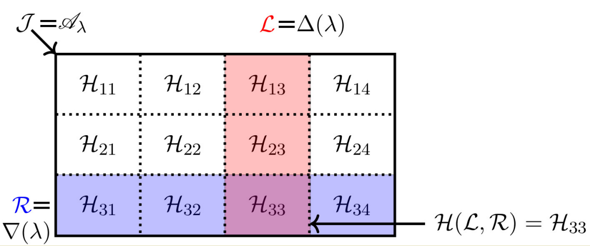
(2A.5)

We ca…

The algebra $\mathscr{A}$ …

Analogy





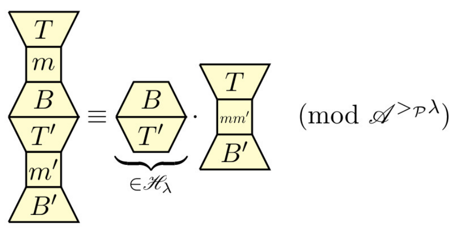
An ordered poset of matrices

Each matrix has values in the sandwiched algebras

Sandw

Def :

Approximate picture to keep in mind



such

(AC a

(AC

(2A

$U \in T(\lambda) n \in B_\lambda$

Similarly for right multiplication by x.

(AC$_3$) There exists a free $\mathscr{A}$-$\mathscr{H}_\lambda$-bimodule $\Delta(\lambda)$, a free $\mathscr{H}_\lambda$-$\mathscr{A}$-bimodule $\nabla(\lambda)$, and an $\mathscr{A}$-bimodule isomorphism

$$\mathscr{A}_\lambda = \mathscr{A}^{\geq_{\mathcal{P}} \lambda} / \mathscr{A}^{>_{\mathcal{P}} \lambda} \cong \Delta(\lambda) \otimes_{\mathscr{H}_\lambda} \nabla(\lambda). \tag{2A.5}$$

We call $\mathscr{A}_\lambda$ the ***cell algebra***, and $\Delta(\lambda)$ and $\nabla(\lambda)$ left and right ***cell modules***.

The algebra $\mathscr{A}$ is a ***sandwich cellular algebra*** if it has a sandwich cell datum.

Sandw

Def

such

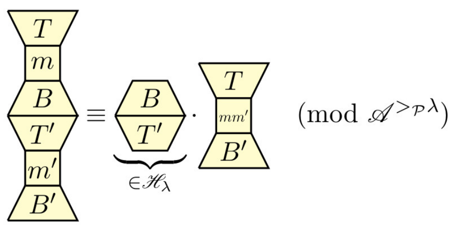
(AC

(AC

(2A

$U\in\mathcal{T}(\lambda) n\in B_\lambda$

Approximate picture to keep in mind



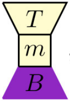
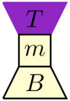
Similarly for right multiplication by x.

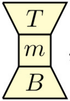
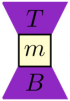
(A

(2A

The

As free $\mathbb{K}$ vector spaces:

$\mathbb{K}\mathcal{L}_\lambda \cong \mathbb{K}\mathcal{T}(\lambda) \otimes_{\mathbb{K}} \mathscr{H}_\lambda \leftrightsquigarrow$  , $\mathbb{K}\mathcal{R}_\lambda \cong \mathscr{H}_\lambda \otimes_{\mathbb{K}} \mathbb{K}\mathcal{B}(\lambda) \leftrightsquigarrow$  ,

$\mathcal{J}_\lambda \cong \mathbb{K}\mathcal{T}(\lambda) \otimes_{\mathbb{K}} \mathscr{H}_\lambda \otimes_{\mathbb{K}} \mathbb{K}\mathcal{B}(\lambda) \leftrightsquigarrow$  , $\mathbb{K}\mathcal{H}_\lambda \cong \mathscr{H}_\lambda \leftrightsquigarrow$  .

Sandw

De

suc
(A

(A
(2A

(A

(2A

Th

Example

All algebras are sandwich cellular with $\mathcal{P} = \{\bullet\}$ and $\mathscr{H}_\bullet = \mathscr{A}$

We get the fantastic tautology:

$$\left\{\begin{matrix}\text{simples of}\\ \mathscr{A}\end{matrix}\right\} = \left\{\begin{matrix}\text{simples asso-}\\ \text{ciated to } \bullet\end{matrix}\right\} \xleftrightarrow{\text{one-to-one}} \left\{\begin{matrix}\text{simples of}\\ \mathscr{H}_\bullet\end{matrix}\right\} = \left\{\begin{matrix}\text{simples of}\\ \mathscr{A}\end{matrix}\right\}$$

The point is to find a good sandwich datum!

Sandwiches

Example

Many monoid algebras with the monoid basis

Definition 2A.3. A $\mathscr{H}_\lambda, B_\lambda), C)$, where:

- $\mathcal{P} = (\mathcal{P}, <_{\mathcal{P}})$ is a poset (the ***middle poset*** with ***sandwich order*** $<_{\mathcal{P}}$),
- $\mathcal{T} = \bigcup_{\lambda \in \mathcal{P}} \mathcal{T}(\lambda)$ and $\mathcal{B} = \bigcup_{\lambda \in \mathcal{P}} \mathcal{B}(\lambda)$ are collections of finite sets (the ***top/bottom sets***),
- For $\lambda \in \mathcal{P}$ we have algebras $\mathscr{H}_\lambda$ (the ***sandwiched algebras***) and bases B_λ of $\mathscr{H}_\lambda$,
- $C\colon \coprod_{\lambda \in \mathcal{P}} \mathcal{T}(\lambda) \times B_\lambda \times \mathcal{B}(\lambda) \to \mathscr{A}; (T, m, B) \mapsto c^\lambda_{T,m,B}$ is an injective map,

such that:

(AC$_1$) The set $B_{\mathscr{A}} = \left\{ c^\lambda_{T,m,B} \,\middle|\, \lambda \in \mathcal{P}, T \in \mathcal{T}(\lambda), B \in \mathcal{B}(\lambda), m \in B_\lambda \right\}$ is a basis of $\mathscr{A}$. (We call $B_{\mathscr{A}}$ a ***sandwich cellular basis***.)

(AC$_2$) For all $x \in \mathscr{A}$ there exist scalars $r^x_{TU} \in \mathbb{K}$ that do not depend on B or on m, such that

$$x c^\lambda_{T,m,B} \equiv \sum_{U \in \mathcal{T}(\lambda) n \in B_\lambda} r^x_{TU} c^\lambda_{U,n,B} \pmod{\mathscr{A}^{>_{\mathcal{P}} \lambda}}. \tag{2A.4}$$

Similarly for right multiplication by x.

(AC$_3$) There exists a free $\mathscr{A}$-$\mathscr{H}_\lambda$-bimodule $\Delta(\lambda)$, a free $\mathscr{H}_\lambda$-$\mathscr{A}$-bimodule $\nabla(\lambda)$, and an $\mathscr{A}$-bimodule isomorphism

$$\mathscr{A}_\lambda = \mathscr{A}^{\geq_{\mathcal{P}} \lambda} / \mathscr{A}^{>_{\mathcal{P}} \lambda} \cong \Delta(\lambda) \otimes_{\mathscr{H}_\lambda} \nabla(\lambda). \tag{2A.5}$$

We call $\mathscr{A}_\lambda$ the ***cell algebra***, and $\Delta(\lambda)$ and $\nabla(\lambda)$ left and right ***cell modules***.

The algebra $\mathscr{A}$ is a ***sandwich cellular algebra*** if it has a sandwich cell datum.

Sandwiches

Example

Many monoid algebras with the monoid basis

Definition 2A.3. A ... $\mathscr{H}_\lambda, B_\lambda)), C)$, where:

- $\mathcal{P} = (\mathcal{P}, <_{\mathcal{P}})$ is a poset (the ***middle poset*** with ***sandwich order*** $<_{\mathcal{P}}$),
- $\mathcal{T} = \bigcup_{\lambda\in\mathcal{P}} \mathcal{T}(\lambda)$ and $\mathcal{B} = \coprod_{\lambda\in\mathcal{P}} \mathcal{B}(\lambda)$ are collections of finite sets (the ***top/bottom sets***),
- For $\lambda \in \mathcal{P}$ we ha... es B_λ of $\mathscr{H}_\lambda$,
- $C\colon \coprod_{\lambda\in\mathcal{P}} \mathcal{T}(\lambda) \times$ ... nap,

such that:

(AC$_1$) The set $B_{\mathscr{A}} = \{$... of $\mathscr{A}$. (We call $B_{\mathscr{A}}$ a ***sandwich cellu***...

(AC$_2$) For all $x \in \mathscr{A}$ th... on m, such that

(2A.4)

Similarly for righ...

(AC$_3$) There exists a fr...), and an $\mathscr{A}$-bimodule isomorphism

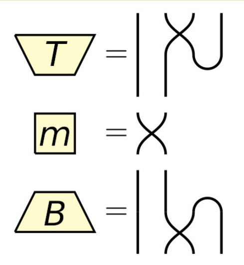
(2A.5)

We call $\mathscr{A}_\lambda$ the ... ***nodules***.

The algebra $\mathscr{A}$ is a ***san***... um.

Example

Diagram algebras with the diagram basis
e.g. the Brauer algebra



Sandwiches

Definition 2A.3. A

Example

Many monoid algebras with the monoid basis

$\mathscr{H}_\lambda, B_\lambda), C)$, where:

- $\mathcal{P} = (\mathcal{P}, <_{\mathcal{P}})$ is a poset (the ***middle poset*** with ***sandwich order*** $<_{\mathcal{P}}$),
- $\mathcal{T} = \bigcup_{\lambda\in\mathcal{P}} \mathcal{T}(\lambda)$ and $\mathcal{B} = \coprod$ ... are collections of finite sets (the ***top/bottom sets***),
- For $\lambda \in \mathcal{P}$ we ha ... es B_λ of $\mathscr{H}_\lambda$,
- $C\colon \coprod_{\lambda\in\mathcal{P}} \mathcal{T}(\lambda) \times$ ... nap,

such that:

(AC$_1$) The set $B_{\mathscr{A}} = \{c$ ... of $\mathscr{A}$. (We call $B_{\mathscr{A}}$ a ***sandwich cellu***

(AC$_2$) For all $x \in \mathscr{A}$ th ... on m, such that

(2A.4)

Similarly for righ

(AC$_3$) There exists a fre ...), and an $\mathscr{A}$-bimodule isomorphism

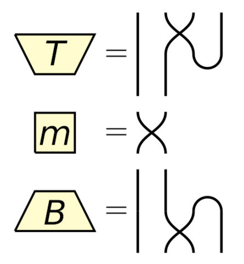
(2A.5)

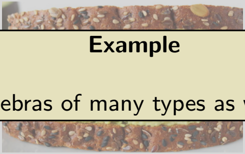
We call $\mathscr{A}_\lambda$ the ***c*** ... ***nodules***.

The algebra $\mathscr{A}$ is a ***san*** ... um.

Example

Diagram algebras with the diagram basis
e.g. the Brauer algebra



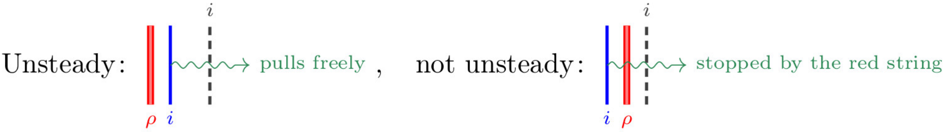


Example

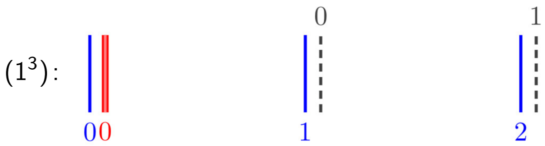
KLR algebras of many types as we will see

Let us sandwich wKLRW diagrams!

- Cyclotomic (fin dim) quotients ⇔ bounded regions:



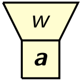
- Sandwich cellular bases ⇔ minimal regions (I will elaborate momentarily):



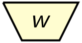
- More properties I won't explain today due to time restrictions...

Let us sandwich wKLRW diagrams!

- wKLRW algebras have standard bases, with the picture:



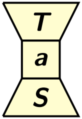
with



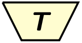
is a permutation diagram,

is an idempotent with dots.

- wKLRW algebras often have sandwich cellular bases, with the picture:

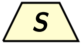


with

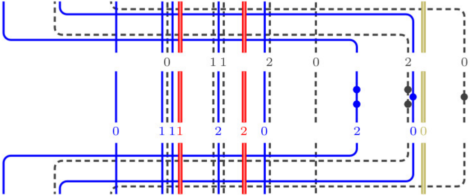


is a permutation diagram,

is an idempotent with dots,



is a permutation diagram.



Let us sandwich

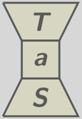
- wKLRW alge

am,

dots.

> ▶ Standard bases work regardless of the quiver but have no other property despite being a basis
>
> ▶ Sandwich cellular bases depend on the quiver and give a classification of simple modules
>
> ▶ What is sandwiched are (quotients of) polynomial algebras

- wKLRW algebras often have sandwich cellular bases, with the picture:

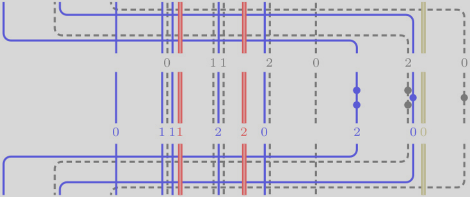


with

is a permutation diagram,

is an idempotent with dots,

is a permutation diagram.



Let us sandwich

- wKLRW alge…

am,

dots.

- Standard bases work regardless of the quiver but have no other property despite being a basis

- Sandwich cellular bases depend on the quiver and give a classification of simple modules

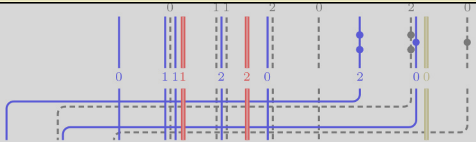
- What is sandwiched are (quotients of) polynomial algebras

- The overall strategy to construct such bases is the same for all types (but the details differ)

and for the infinite dimensional and the cyclotomic case the construction is also the same

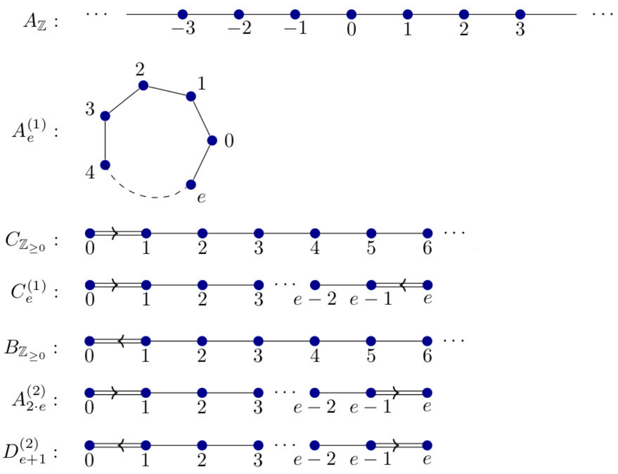
- We know that the cellular bases work in types $A_{\mathbb{Z}}$, $A_e^{(1)}$, $B_{\mathbb{N}}$, $C_e^{(1)}$, $A_{2e}^{(2)}$, $D_{e+1}^{(2)}$ other, in particular finite, types are work in progress

- The combinatorics is inspired by, but different from, constructions of **Bowman ∼2017, Ariki–Park ∼2012/2013, Ariki–Park–Speyer ∼2017**



Summary of the before

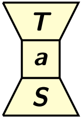
We know cellularity in these cases (for inf dim and cyclotomic quotients):



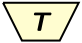
95% theorem This list can be extended to contain all finite types, $E_6^{(2)}$, $F_4^{(1)}$, $G_2^{(1)}$

Open Compare our inf dim case for finite types to **Kleshchev–Loubert(–Miemietz)** $\sim$**2013**

Let us sandwich wKLRW diagrams!

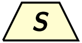


with



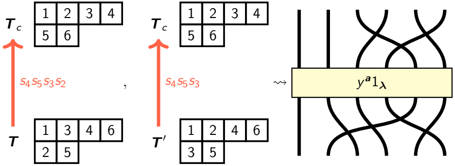
is a permutation diagram,

is an idempotent with dots,



is a permutation diagram.

- The definition of the permutation follows the usual strategy in this context:



- The main meat Let me focus on the middle $y^{a} 1_{\lambda}$

Let us sandwich wKLRW diagrams!

$C_3^{(1)}: (12, 6^3, 5) \rightsquigarrow$

0	1	2	3	2	1	0	1	2	3	2	1
1	0	1	2	3	2						
2	1	0	1	2	3						
3	2	1	0	1	2						
2	3	2	1	0							

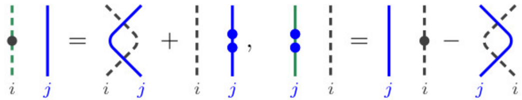
- Assume the tableaux combinatorics is given (a better statement later!)
- Place strings inductively as far to the right as possible (this is the order!)
- 1_λ is minimal with respect to placing the strings to the right
- 1_λ stays minimal when dots are put on certain strands $\rightsquigarrow$ get $y^{\mathbf{a}} 1_\lambda$
- Done!

Let us sandwich wKLRW diagrams!

$C_e^{(1)}$:

0	1	2	3	2	1	0	1	2	3	2	1

Lets ignore the dots for today – I bothered you with too much combinatorics anyway ;-)
But they come directly from the Reidemeister II relations, e.g.



for either of $\begin{cases} i=0, j=1, \\ i=e, j=e-1, \end{cases}$

This gives us the notion of the rightmost parking slot where stings are blocked

In other words: Stare at Reidemeister II !

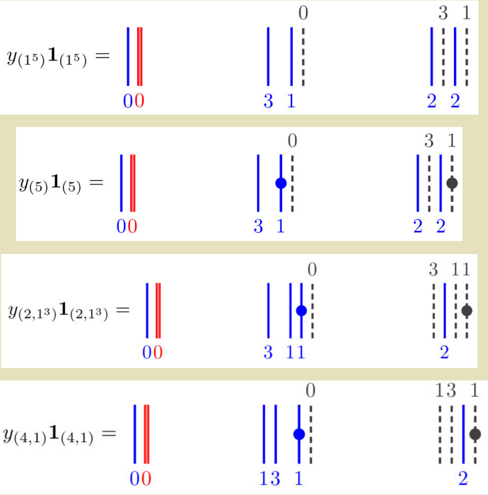
- Place strings inductively as far to the right as possible (this is the order!)
- 1_λ is minimal with respect to placing the strings to the right
- 1_λ stays minimal when dots are put on certain strands $\rightsquigarrow$ get $y^a 1_\lambda$
- Done!

Let us sandwich wKLRW diagrams!

$C_e^{(1)}$:

$C_3^{(1)}$: (12

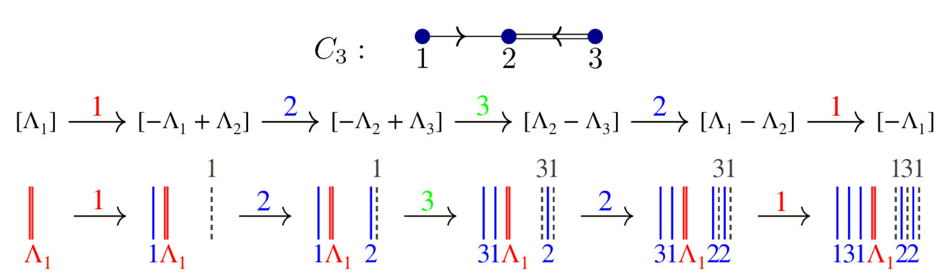
Example for the middles $y^a 1_\lambda$



- Assume the t … ment later!)
- Place strings … s is the order!)
- 1_λ is minima … ght
- 1_λ stays mi … get $y^a 1_\lambda$
- Done!

Let us sandwich wKLRW diagrams!

The (conjectural) picture for a lot of types



Checked for finite types (currently work in progress)

We are now looking for an abstract property on the crystal that ensures that everything works

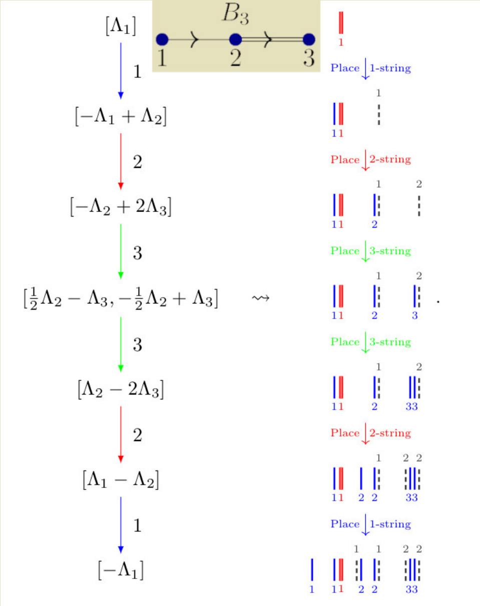
- 1_λ stays minimal when dots are put on certain strands $\rightsquigarrow$ get $y^a 1_\lambda$
- Done!

Let us sand

$C_e^{(1)}$

$C_3^{(1)}$

The (conjectural) picture for all types – second example



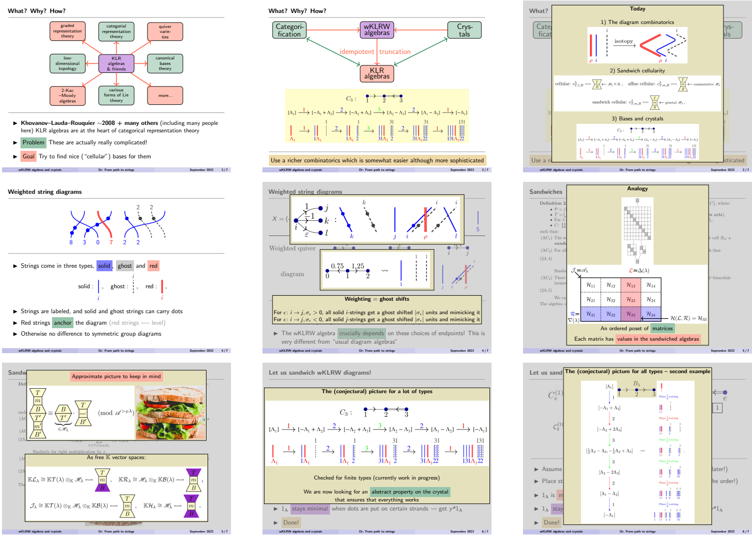
- Assume ... later!)
- Place st ... he order!)
- 1_λ is m
- 1_λ stay ... $^a 1_\lambda$
- Done!

Let us sandwich wKLRW diagrams!

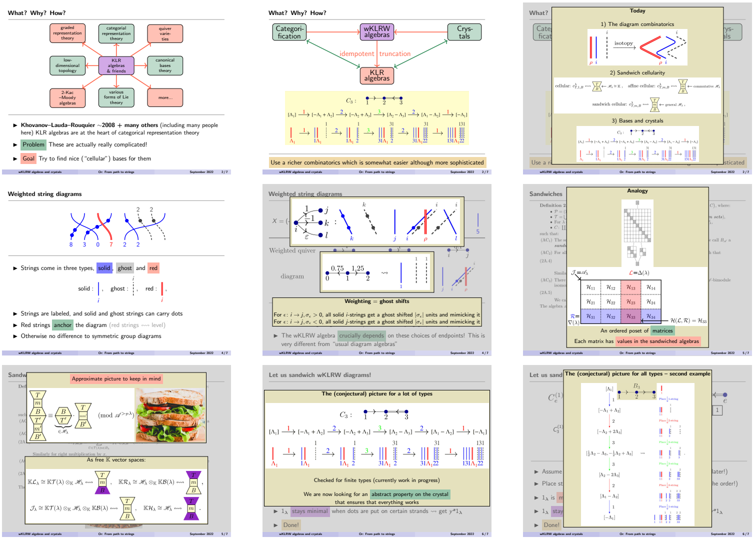
Wrap up

- wKLRW algebras generalize KLR algebras and friends
- They have a build in distance
- Most properties can be described using distance
- Most properties are type-independent
- Some properties are (in some form) type-independent
- Our dim calculations for the sandwich cellular basis match with the formulas of **Hu-Shi ∼2021** in the special cyclotomic KLR case

- 1_λ is minimal with respect to placing the strings to the right
- 1_λ stays minimal when dots are put on certain strands $\rightsquigarrow$ get $y^{a}1_\lambda$
- Done!



There is still much to do...



Thanks for your attention!