Fun with Diagram Algebras

Madeline Nurcombe

The University of Queensland

September 17, 2021

Context

The Temperley-Lieb algebra:

- Introduced by Temperley & Lieb (1971) where it arose in lattice models in statistical mechanics

Context

The Temperley-Lieb algebra:

- Introduced by Temperley & Lieb (1971) where it arose in lattice models in statistical mechanics
- Rediscovered and popularised by Jones (1983), who applied it to knot theory

Context

The Temperley-Lieb algebra:

- Introduced by Temperley & Lieb (1971) where it arose in lattice models in statistical mechanics
- Rediscovered and popularised by Jones (1983), who applied it to knot theory
- Representation theory well-understood

Context

The Temperley-Lieb algebra:

- Introduced by Temperley & Lieb (1971) where it arose in lattice models in statistical mechanics
- Rediscovered and popularised by Jones (1983), who applied it to knot theory
- Representation theory well-understood
- Original work on the representation theory: Martin (1991), Westbury (1995)

Context

The Temperley-Lieb algebra:

- Introduced by Temperley & Lieb (1971) where it arose in lattice models in statistical mechanics
- Rediscovered and popularised by Jones (1983), who applied it to knot theory
- Representation theory well-understood
- Original work on the representation theory: Martin (1991), Westbury (1995)
- Review paper by Ridout & Saint-Aubin (2014) discusses the representation theory very thoroughly

Context

One-boundary Temperley-Lieb algebra:

- Now accounts for a boundary on one side

Context

One-boundary Temperley-Lieb algebra:

- Now accounts for a boundary on one side
- I'll use the definition from Pearce, Rasmussen & Tipunin (2014), which depends on three complex parameters β, β_1, β_2

Context

One-boundary Temperley-Lieb algebra:

- Now accounts for a boundary on one side
- I'll use the definition from Pearce, Rasmussen & Tipunin (2014), which depends on three complex parameters β, β_1, β_2
- Related *blob algebra* has been studied by Martin & Saleur (1994), Martin & Woodcock (2000), but only has two parameters

Context

One-boundary Temperley-Lieb algebra:

- Now accounts for a boundary on one side
- I'll use the definition from Pearce, Rasmussen & Tipunin (2014), which depends on three complex parameters β, β_1, β_2
- Related *blob algebra* has been studied by Martin & Saleur (1994), Martin & Woodcock (2000), but only has two parameters

My task:

- Adapt the Ridout & Saint-Aubin paper to study the three-parameter one-boundary Temperley-Lieb algebra $1\mathrm{BTL}_n(\beta; \beta_1, \beta_2)$

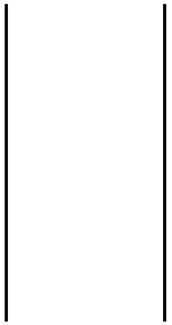
Diagrams

We now introduce *n*-diagrams.

Diagrams

We now introduce n-diagrams.

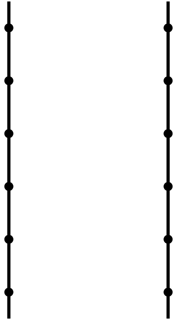
- Two vertical lines



Diagrams

We now introduce n-diagrams.

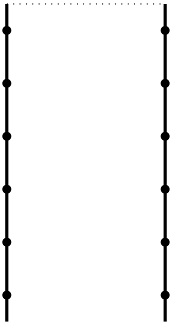
- Two vertical lines
- n nodes on each line (here $n = 6$)



Diagrams

We now introduce n-diagrams.

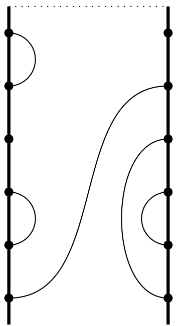
- Two vertical lines
- n nodes on each line (here $n = 6$)
- Dotted boundary line at top



Diagrams

We now introduce n-diagrams.

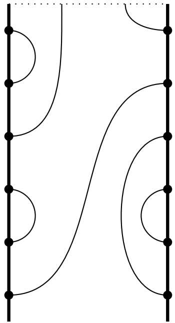
- Two vertical lines
- n nodes on each line (here $n = 6$)
- Dotted boundary line at top
- Strings may connect pairs of nodes



Diagrams

We now introduce n-diagrams.

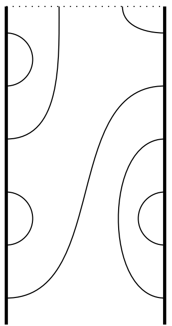
- Two vertical lines
- n nodes on each line (here $n = 6$)
- Dotted boundary line at top
- Strings may connect pairs of nodes
- Or nodes to boundary



Diagrams

We now introduce n-diagrams.

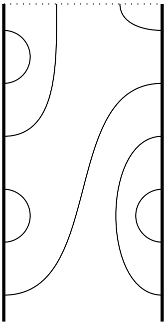
- Two vertical lines
- n nodes on each line (here $n = 6$)
- Dotted boundary line at top
- Strings may connect pairs of nodes
- Or nodes to boundary
- One string connected to each node



Diagrams

We now introduce n-diagrams.

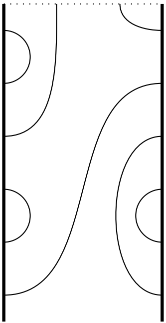
- Two vertical lines
- n nodes on each line (here $n = 6$)
- Dotted boundary line at top
- Strings may connect pairs of nodes
- Or nodes to boundary
- One string connected to each node
- Strings may not cross!



Diagrams

We now introduce n-diagrams.

- Two vertical lines
- n nodes on each line (here $n = 6$)
- Dotted boundary line at top
- Strings may connect pairs of nodes
- Or nodes to boundary
- One string connected to each node
- Strings may not cross!



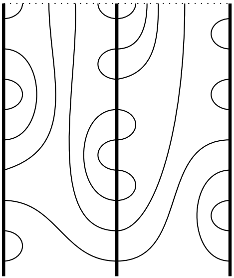
Introduce $1\mathrm{BTL}_n$ as the complex vector space with the set of all n-diagrams as basis.

Diagram operations: multiplication in $1\mathrm{BTL}_n(\beta; \beta_1, \beta_2)$

We now want to multiply n-diagrams.

Diagram operations: multiplication in $1\mathrm{BTL}_n(\beta; \beta_1, \beta_2)$

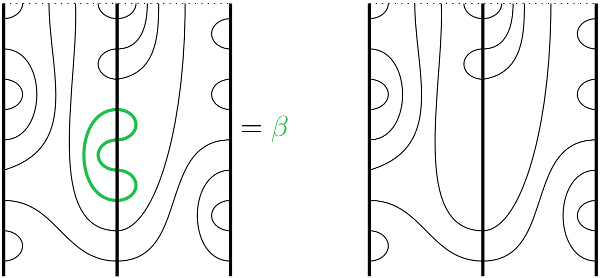
We now want to multiply n-diagrams.



- Draw the diagrams next to each other

Diagram operations: multiplication in $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$

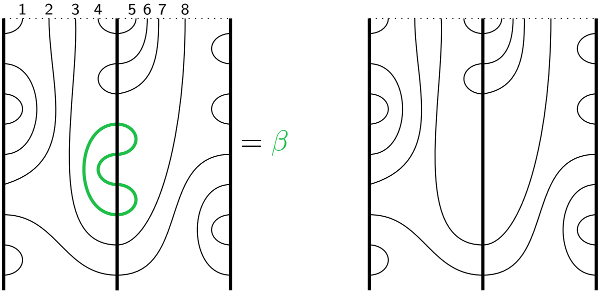
We now want to multiply *n*-diagrams.



- Draw the diagrams next to each other
- Replace each loop with a factor of β

Diagram operations: multiplication in $1\mathrm{BTL}_n(\beta; \beta_1, \beta_2)$

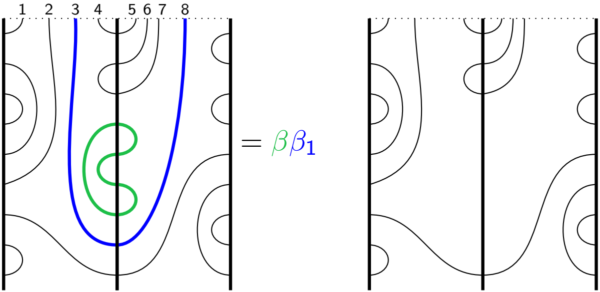
We now want to multiply n-diagrams.



- Draw the diagrams next to each other
- Replace each loop with a factor of β

Diagram operations: multiplication in $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$

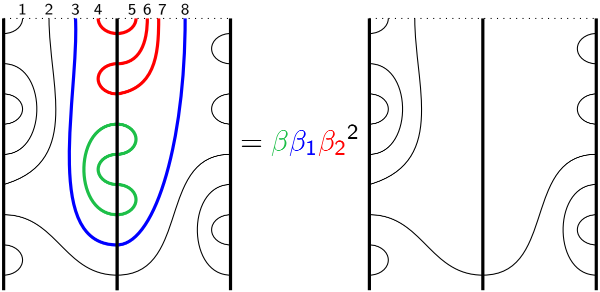
We now want to multiply *n*-diagrams.



- Draw the diagrams next to each other
- Replace each loop with a factor of β
- Replace each boundary arc with odd (even) left connection by a factor of β_1 (β_2)

Diagram operations: multiplication in $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$

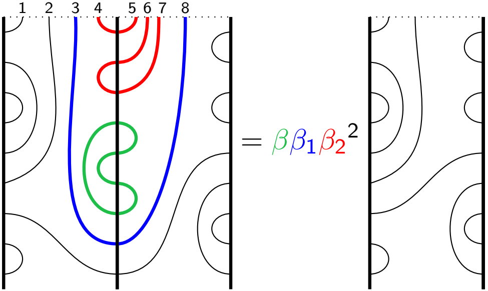
We now want to multiply *n*-diagrams.



- Draw the diagrams next to each other
- Replace each loop with a factor of β
- Replace each boundary arc with odd (even) left connection by a factor of β_1 (β_2)

Diagram operations: multiplication in $1\mathrm{BTL}_n(\beta; \beta_1, \beta_2)$

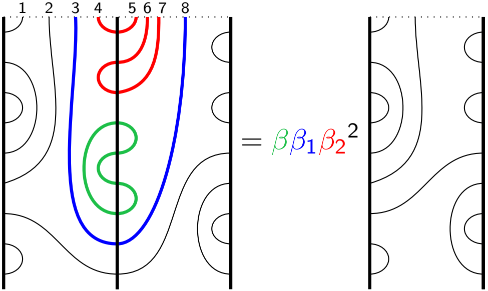
We now want to multiply *n*-diagrams.



- Draw the diagrams next to each other
- Replace each loop with a factor of β
- Replace each boundary arc with odd (even) left connection by a factor of β_1 (β_2)
- Remove the middle line and tighten the strings

Diagram operations: multiplication in $1\mathrm{BTL}_n(\beta; \beta_1, \beta_2)$

We now want to multiply *n*-diagrams.



- Draw the diagrams next to each other
- Replace each loop with a factor of β
- Replace each boundary arc with odd (even) left connection by a factor of β_1 (β_2)
- Remove the middle line and tighten the strings
- Extend this bilinearly to all of $1\mathrm{BTL}_n(\beta; \beta_1, \beta_2)$

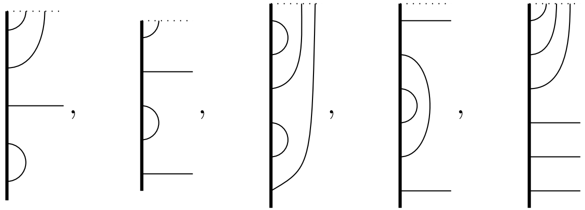
Introducing standard modules $\mathcal{V}_{n,d}$

We now want to construct a representation of $1\mathrm{BTL}_n(\beta; \beta_1, \beta_2)$.

Introducing standard modules $\mathcal{V}_{n,d}$

We now want to construct a representation of $1\mathrm{BTL}_n(\beta; \beta_1, \beta_2)$.

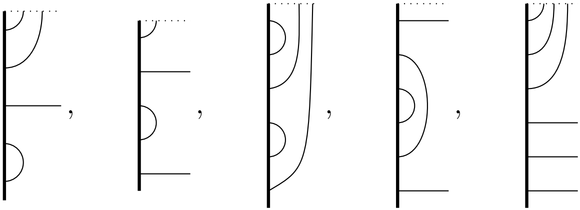
- Each n-diagram can be cut in half vertically to make *half-diagrams*, e.g.:



Introducing standard modules $\mathcal{V}_{n,d}$

We now want to construct a representation of $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$.

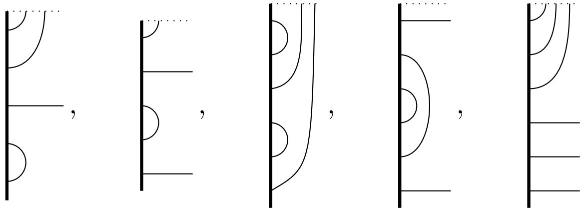
- Each n-diagram can be cut in half vertically to make *half-diagrams*, e.g.:



- Each node in a half-diagram can have
  - a *link*, connecting it to another node,
  - a *boundary link*, connected to the boundary, or
  - a *defect*, sticking straight out to the right.

We now want to construct a representation of $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$.

- Each n-diagram can be cut in half vertically to make *half-diagrams*, e.g.:



- Each node in a half-diagram can have
    - a *link*, connecting it to another node,
    - a *boundary link*, connected to the boundary, or
    - a *defect*, sticking straight out to the right.
- No crossing of strings! Also, links and boundary links cannot pass over defects.

Introducing standard modules $\mathcal{V}_{n,d}$

Let $\mathcal{V}_{n,d}$ be the complex vector space with the set of all n-half-diagrams with precisely d defects as its basis.

Introducing standard modules $\mathcal{V}_{n,d}$

Let $\mathcal{V}_{n,d}$ be the complex vector space with the set of all n-half-diagrams with precisely d defects as its basis.

We now want to define the action of $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$ on $\mathcal{V}_{n,d}$ to make it a *(left)* $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$*-module*.

Introducing standard modules $\mathcal{V}_{n,d}$

Let $\mathcal{V}_{n,d}$ be the complex vector space with the set of all n-half-diagrams with precisely d defects as its basis.

We now want to define the action of $1\mathrm{BTL}_n(\beta; \beta_1, \beta_2)$ on $\mathcal{V}_{n,d}$ to make it a *(left)* $1\mathrm{BTL}_n(\beta; \beta_1, \beta_2)$*-module*.

- Define for diagrams and half-diagrams and extend bilinearly

Introducing standard modules $\mathcal{V}_{n,d}$

Let $\mathcal{V}_{n,d}$ be the complex vector space with the set of all n-half-diagrams with precisely d defects as its basis.

We now want to define the action of $1\mathrm{BTL}_n(\beta; \beta_1, \beta_2)$ on $\mathcal{V}_{n,d}$ to make it a *(left)* $1\mathrm{BTL}_n(\beta; \beta_1, \beta_2)$*-module*.

- Define for diagrams and half-diagrams and extend bilinearly
- Draw side-by-side, replace loops and boundary arcs with factors of β, β_1, β_2 as before, then tighten the strings

Introducing standard modules $\mathcal{V}_{n,d}$

Let $\mathcal{V}_{n,d}$ be the complex vector space with the set of all n-half-diagrams with precisely d defects as its basis.

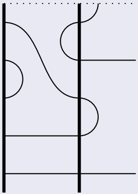
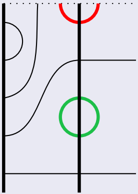
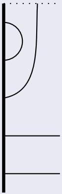
We now want to define the action of $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$ on $\mathcal{V}_{n,d}$ to make it a *(left)* $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$*-module*.

- Define for diagrams and half-diagrams and extend bilinearly
- Draw side-by-side, replace loops and boundary arcs with factors of β, β_1, β_2 as before, then tighten the strings
- Must preserve number of defects d; if not, set the result to 0

Diagram operations: action on standard modules $\mathcal{V}_{n,d}$

Examples

In $\mathcal{V}_{5,2}$, we have

 $= 0,$  $= \beta\beta_2$  ,

while in $\mathcal{V}_{5,1}$ we have

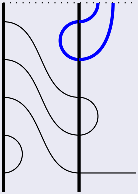
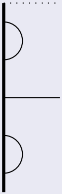
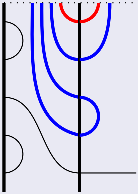
 $= \beta_1$  ,  $= {\beta_1}^2\beta_2$  .

Diagram operations: outer product

From a pair of n-half-diagrams with the same number of defects d, we can construct an n-diagram.

From a pair of n-half-diagrams with the same number of defects d, we can construct an n-diagram.

Example

In $\mathcal{V}_{6,2}$, we can take

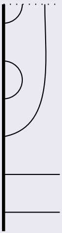
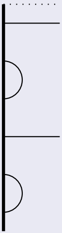
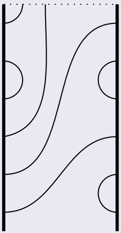
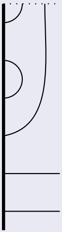
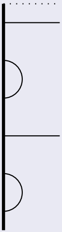
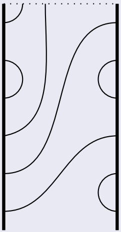
$x =$  , $y =$  , $|x \;\; y| :=$  $\in 1\mathrm{BTL}_6$.

Diagram operations: outer product

From a pair of *n*-half-diagrams with the same number of defects d, we can construct an *n*-diagram.

Example

In $\mathcal{V}_{6,2}$, we can take

$x =$  , $y =$  , $|x \;\; y| :=$  $\in \mathrm{1BTL}_6$.

This can be extended bilinearly to a map $|\cdot \;\; \cdot| : \mathcal{V}_{n,d} \times \mathcal{V}_{n,d} \to \mathrm{1BTL}_n$.

Diagram operations: bilinear form on $\mathcal{V}_{n,d}$

What if we put two (n, d)-half-diagrams back-to-back instead?

Diagram operations: bilinear form on $\mathcal{V}_{n,d}$

What if we put two (n, d)-half-diagrams back-to-back instead?

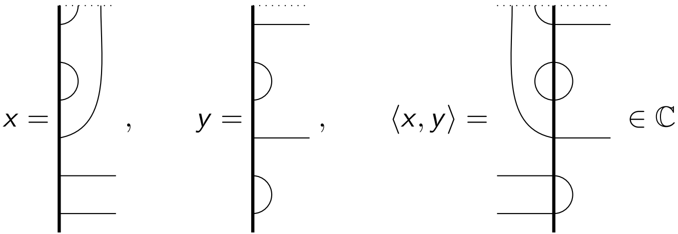
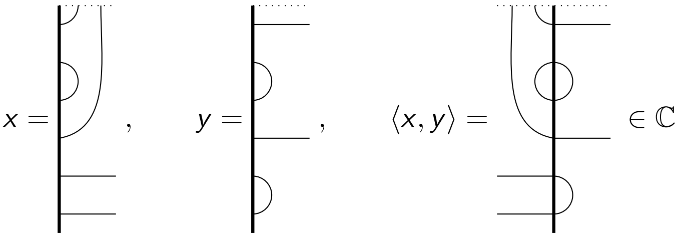


Diagram operations: bilinear form on $\mathcal{V}_{n,d}$

What if we put two (n, d)-half-diagrams back-to-back instead?

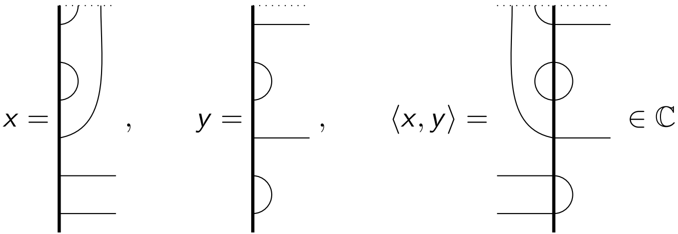


Extending this bilinearly to $\mathcal{V}_{n,d} \times \mathcal{V}_{n,d}$ would give a *bilinear form*:

- Like an inner product, except without some restrictions like $\langle x, x \rangle \geq 0$ and $\langle x, x \rangle = 0 \Rightarrow x = 0$

Diagram operations: bilinear form on $\mathcal{V}_{n,d}$

What if we put two (n, d)-half-diagrams back-to-back instead?



Extending this bilinearly to $\mathcal{V}_{n,d} \times \mathcal{V}_{n,d}$ would give a *bilinear form*:

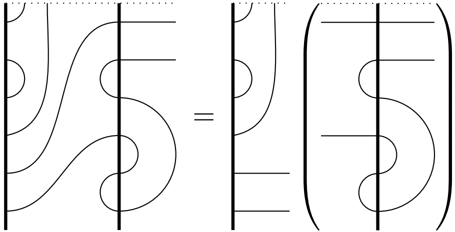
- Like an inner product, except without some restrictions like $\langle x, x\rangle \geq 0$ and $\langle x, x\rangle = 0 \Rightarrow x = 0$
- Bilinear NOT sesquilinear

Diagram operations: bilinear form on $\mathcal{V}_{n,d}$

For the bilinear form to be useful, we would like to have

$$|x \quad y|\, z \qquad = \qquad x\,\langle y, z\rangle$$

for all $x, y, z \in \mathcal{V}_{n,d}$, e.g.:



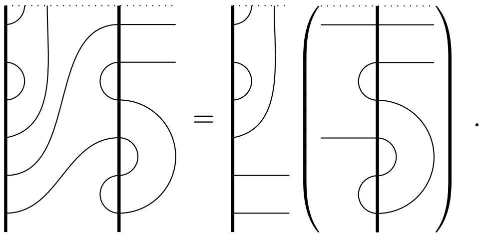
.

Diagram operations: bilinear form on $\mathcal{V}_{n,d}$

For the bilinear form to be useful, we would like to have

$$|x \quad y|\, z \qquad = \qquad x\,\langle y, z\rangle$$

for all $x, y, z \in \mathcal{V}_{n,d}$, e.g.:



Observing that $|x \quad y|\, z \propto x$ for all $x, y, z \in \mathcal{V}_{n,d}$, this is possible, and uniquely defines a bilinear form $\langle \cdot, \cdot \rangle : \mathcal{V}_{n,d} \times \mathcal{V}_{n,d} \to \mathbb{C}$.

Diagram operations: bilinear form on $\mathcal{V}_{n,d}$

Examples

In $\mathcal{V}_{6,2}$, we have

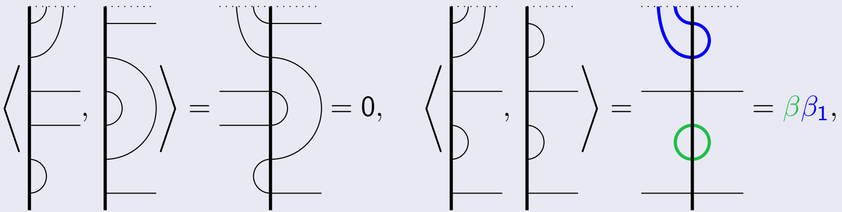
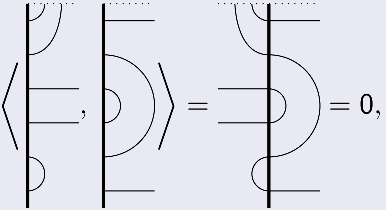
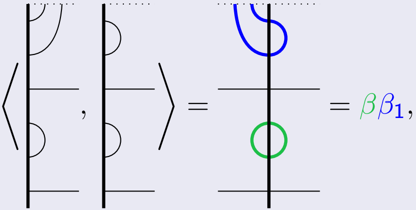


Diagram operations: bilinear form on $\mathcal{V}_{n,d}$

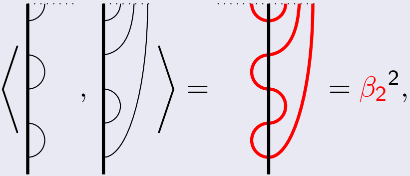
Examples

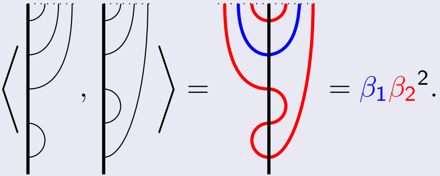
In $\mathcal{V}_{6,2}$, we have





while in $\mathcal{V}_{5,0}$ we have





Bilinear form: Gram Matrices

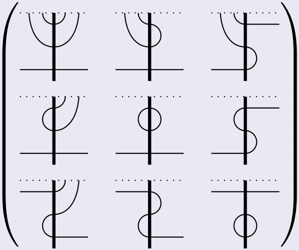
Each of these bilinear forms has an associated *Gram matrix*.

Bilinear form: Gram Matrices

Each of these bilinear forms has an associated *Gram matrix*.

Example

$G_{3,1} =$



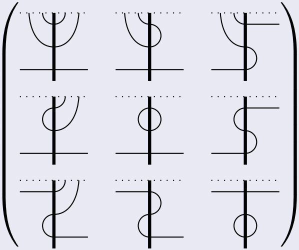
$$= \begin{pmatrix} \beta_1\beta_2 & \beta_1 & 0 \\ \beta_1 & \beta & 1 \\ 0 & 1 & \beta \end{pmatrix}$$

Bilinear form: Gram Matrices

Each of these bilinear forms has an associated *Gram matrix*.

Example

$G_{3,1} =$



$$= \begin{pmatrix} \beta_1\beta_2 & \beta_1 & 0 \\ \beta_1 & \beta & 1 \\ 0 & 1 & \beta \end{pmatrix}$$

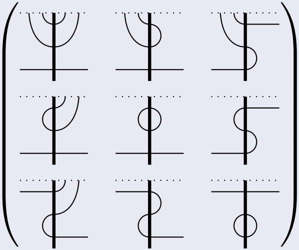
Note that $\det(G_{n,d})$ must be a polynomial in β, β_1, β_2.

Bilinear form: Gram Matrices

Each of these bilinear forms has an associated *Gram matrix*.

Example

$G_{3,1} =$



$$= \begin{pmatrix} \beta_1\beta_2 & \beta_1 & 0 \\ \beta_1 & \beta & 1 \\ 0 & 1 & \beta \end{pmatrix}$$

Note that $\det(G_{n,d})$ must be a polynomial in β, β_1, β_2.

Keep this in mind! Very important later.

Structure of $\mathcal{V}_{n,d}$

We now have the tools to determine some structural properties of $\mathcal{V}_{n,d}$.

Structure of $\mathcal{V}_{n,d}$

We now have the tools to determine some structural properties of $\mathcal{V}_{n,d}$.

Key questions

Structure of $\mathcal{V}_{n,d}$

We now have the tools to determine some structural properties of $\mathcal{V}_{n,d}$.

Key questions

- If you start with a vector $v \in \mathcal{V}_{n,d}$, which other elements of $\mathcal{V}_{n,d}$ can you get to by acting on v with elements of the algebra $\mathrm{1BTL}_n(\beta; \beta_1, \beta_2)$?

Structure of $\mathcal{V}_{n,d}$

We now have the tools to determine some structural properties of $\mathcal{V}_{n,d}$.

Key questions

- If you start with a vector $v \in \mathcal{V}_{n,d}$, which other elements of $\mathcal{V}_{n,d}$ can you get to by acting on v with elements of the algebra $\mathrm{1BTL}_n(\beta; \beta_1, \beta_2)$?
- **Are there any subspaces you can get stuck in?**

Structure of $\mathcal{V}_{n,d}$

Definition

A *submodule* of an A-module V, for an algebra A, is a subspace W of V such that

$$\forall a \in A,\ \forall w \in W, \qquad aw \in W.$$

Structure of $\mathcal{V}_{n,d}$

Definition

A *submodule* of an A-module V, for an algebra A, is a subspace W of V such that

$$\forall a \in A,\ \forall w \in W, \qquad aw \in W.$$

Informally, a submodule is "a subspace you can get stuck in".

Structure of $\mathcal{V}_{n,d}$

Definition

A *submodule* of an A-module V, for an algebra A, is a subspace W of V such that

$$\forall a \in A,\ \forall w \in W, \qquad aw \in W.$$

Informally, a submodule is "a subspace you can get stuck in".

Definition

An A-module V is called *irreducible* if it has no submodules other than $\{0\}$ and V itself.

Structure of $\mathcal{V}_{n,d}$

Question: Is $\mathcal{V}_{n,d}$ irreducible?

Structure of $\mathcal{V}_{n,d}$

Question: Is $\mathcal{V}_{n,d}$ irreducible?

- If $G_{n,d} \neq 0$, then the property $|x \quad y|\, z = x \langle y, z \rangle$ actually implies

$$\mathcal{V}_{n,d} \text{ is irreducible} \quad \Leftrightarrow \quad \det(G_{n,d}) \neq 0.$$

Structure of $\mathcal{V}_{n,d}$

Question: Is $\mathcal{V}_{n,d}$ irreducible?

- If $G_{n,d} \neq 0$, then the property $|x \quad y|\, z = x \langle y, z \rangle$ actually implies

$$\mathcal{V}_{n,d} \text{ is irreducible} \quad \Leftrightarrow \quad \det(G_{n,d}) \neq 0.$$

Hence the technical goals of my thesis were:

- Find $\det(G_{n,d})$,
- Find when $\det(G_{n,d}) = 0$.

Determinant of the Gram matrix

Theorem

For any $\beta, \beta_1, \beta_2 \in \mathbb{C}$, the determinant of the Gram matrix $G_{n,d}$ is given by

$$\det(G_{n,d}) = \begin{cases} (-\beta_1)^{\left(\frac{n}{\frac{n-d}{2}}-1\right)} \prod_{j=1}^{\frac{n-d}{2}} \left(\beta_1 U_{d+j-1}\left(\frac{\beta}{2}\right) - \beta_2 U_{d+j}\left(\frac{\beta}{2}\right)\right)^{\left(\frac{n}{\frac{n-d}{2}-j}\right)} \\ \quad \times \prod_{k=1}^{\frac{n-d}{2}-1} \left(\beta_2 U_{k-1}\left(\frac{\beta}{2}\right) - \beta_1 U_k\left(\frac{\beta}{2}\right)\right)^{\left(\frac{n}{\frac{n-d}{2}-k-1}\right)}, & d \equiv n \mod 2, \\ \beta_2^{\left(\frac{n}{\frac{n-d-1}{2}}\right)} \prod_{j=1}^{\frac{n-d-1}{2}} \left(\beta_2 U_{d+j-1}\left(\frac{\beta}{2}\right) - \beta_1 U_{d+j}\left(\frac{\beta}{2}\right)\right)^{\left(\frac{n}{\frac{n-d-1}{2}-j}\right)} \\ \quad \times \prod_{k=1}^{\frac{n-d-1}{2}} \left(\beta_1 U_{k-1}\left(\frac{\beta}{2}\right) - \beta_2 U_k\left(\frac{\beta}{2}\right)\right)^{\left(\frac{n}{\frac{n-d-1}{2}-k}\right)}, & d \not\equiv n \mod 2, \end{cases}$$

where U_m is the mth Chebyshev polynomial of the second kind.

When $\det(G_{n,d}) = 0$

Theorem

We have $\det(G_{n,d}) = 0$ *if and only if* $d < n$, *and*

- $\beta_{n,d} = 0$; *or*
- $\beta_{n,d} \neq 0$, $q \neq \pm 1$, *and* $\beta'_{n,d} \notin \{q\beta_{n,d}, q^{-1}\beta_{n,d}\}$, *and*
  - $\xi_{n,d} - (d+j+1)\lambda \in \pi\mathbb{Z}$ *for some* $j \in \mathbb{Z}$ *with* $1 \leq j \leq \left\lfloor \frac{n-d}{2} \right\rfloor$, *or*
  - $\xi_{n,d} + k\lambda \in \pi\mathbb{Z}$ *for some* $k \in \mathbb{Z}$ *with* $1 \leq k \leq \left\lfloor \frac{n-d-1}{2} \right\rfloor$; *or*
- $\beta_{n,d} \neq 0$, $q = \pm 1$, *and*
  - $\beta'_{n,d} = \frac{d+j}{d+j+1} q^{-1} \beta_{n,d}$ *for some* $j \in \mathbb{Z}$ *with* $1 \leq j \leq \left\lfloor \frac{n-d}{2} \right\rfloor$, *or*
  - $\beta'_{n,d} = \frac{k+1}{k} q \beta_{n,d}$ *for some* $k \in \mathbb{Z}$ *with* $1 \leq k \leq \left\lfloor \frac{n-d-1}{2} \right\rfloor$,

where $\beta = q + q^{-1}$, $q = e^{i\lambda}$, *and* $\xi_{n,d}$ *comes from the parametrisation*

$$\beta'_{n,d} = \frac{q - q^{-1} e^{2i\xi_{n,d}}}{1 - e^{2i\xi_{n,d}}} \beta_{n,d},$$

where $\beta_{n,d} = \beta_1$ *and* $\beta'_{n,d} = \beta_2$ *if* $d \equiv n \mod 2$, *or vice versa if* $d \not\equiv n \mod 2$.

Results

Results from this:

Results

Results from this:

- For generic β, β_1 and β_2, $\mathcal{V}_{n,d}$ is irreducible.

Results

Results from this:

- For generic β, β_1 and β_2, $\mathcal{V}_{n,d}$ is irreducible.
- Checked the $G_{n,d} = 0$ cases, so found precisely when each $\mathcal{V}_{n,d}$ is irreducible

Results

Results from this:

- For generic β, β_1 and β_2, $\mathcal{V}_{n,d}$ is irreducible.
- Checked the $G_{n,d} = 0$ cases, so found precisely when each $\mathcal{V}_{n,d}$ is irreducible
- Can deduce from this (and further arguments from Graham & Lehrer (1996)) the values of β, β_1, β_2 for which $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$ is *semisimple*, meaning any finite-dimensional $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$-module is isomorphic to a direct sum of irreducible modules

Results

Results from this:

- For generic β, β_1 and β_2, $\mathcal{V}_{n,d}$ is irreducible.
- Checked the $G_{n,d} = 0$ cases, so found precisely when each $\mathcal{V}_{n,d}$ is irreducible
- Can deduce from this (and further arguments from Graham & Lehrer (1996)) the values of β, β_1, β_2 for which $\mathrm{1BTL}_n(\beta; \beta_1, \beta_2)$ is *semisimple*, meaning any finite-dimensional $\mathrm{1BTL}_n(\beta; \beta_1, \beta_2)$-module is isomorphic to a direct sum of irreducible modules
  - Found that $\mathrm{1BTL}_n(\beta; \beta_1, \beta_2)$ is semisimple for generic parameter values, but not all parameter values

Results

Results from this:

- For generic β, β_1 and β_2, $\mathcal{V}_{n,d}$ is irreducible.
- Checked the $G_{n,d} = 0$ cases, so found precisely when each $\mathcal{V}_{n,d}$ is irreducible
- Can deduce from this (and further arguments from Graham & Lehrer (1996)) the values of β, β_1, β_2 for which $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$ is *semisimple*, meaning any finite-dimensional $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$-module is isomorphic to a direct sum of irreducible modules
  - Found that $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$ is semisimple for generic parameter values, but not all parameter values
  - $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$ is semisimple if and only if $\det(G_{n,d}) \neq 0$ for all $0 \leq d \leq n$.

Where to from here?

- Consider *indecomposable* $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$-modules, i.e. those which cannot be expressed as a direct sum of two nonzero submodules

Where to from here?

- Consider *indecomposable* $1\mathrm{BTL}_n(\beta;\beta_1,\beta_2)$-modules, i.e. those which cannot be expressed as a direct sum of two nonzero submodules
- Apply similar procedures to other algebras e.g. two-boundary and periodic Temperley-Lieb algebras, BMW algebra (crossing of strings allowed)

Where to from here?

- Consider *indecomposable* $1\mathrm{BTL}_n(\beta; \beta_1, \beta_2)$-modules, i.e. those which cannot be expressed as a direct sum of two nonzero submodules
- Apply similar procedures to other algebras e.g. two-boundary and periodic Temperley-Lieb algebras, BMW algebra (crossing of strings allowed)
- Connect back to physics – look at the Hamiltonians for the corresponding lattice models, and their energy eigenvalues

Determinant bloopers: Sierpinski triangle

1 1
-1 1 1
1 -1 1 1
1 1 -1 1 1
1 -1 1 -1 1 1
-1 1 1 1 -1 1 1
1 -1 1 -1 1 -1 1 1
1 1 1 1 1 1 -1 1 1
1 1 1 1 1 -1 1 -1 1 1
1 1 1 1 -1 1 1 1 -1 1 1
1 1 1 -1 1 -1 1 -1 1 -1 1 1
1 1 -1 1 1 1 1 1 1 1 -1 1 1
1 -1 1 -1 1 1 1 1 1 -1 1 -1 1 1
-1 1 1 1 -1 1 1 1 -1 1 1 1 -1 1 1
1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 -1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 -1 1 1 1 -1 1 1
1 1 1 1 1 1 1 1 1 1 1 -1 1 -1 1 -1 1 -1 1 1
1 1 1 1 1 1 1 1 1 1 -1 1 1 1 1 1 1 1 -1 1 1
1 1 1 1 1 1 1 1 1 -1 1 -1 1 1 1 1 1 -1 1 -1 1 1
1 1 1 1 1 1 1 1 -1 1 1 1 -1 1 1 1 -1 1 1 1 -1 1 1
1 1 1 1 1 1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 1
1 1 1 1 1 1 -1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 1
1 1 1 1 1 -1 1 -1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 -1 1 1
1 1 1 1 -1 1 1 1 -1 1 1 1 1 1 1 1 1 1 1 1 -1 1 1 1 -1 1 1
1 1 1 -1 1 -1 1 -1 1 -1 1 1 1 1 1 1 1 1 1 -1 1 -1 1 -1 1 -1 1 1
1 1 -1 1 1 1 1 1 1 1 -1 1 1 1 1 1 1 1 -1 1 1 1 1 1 1 1 -1 1 1
1 -1 1 -1 1 1 1 1 1 -1 1 -1 1 1 1 1 1 -1 1 -1 1 1 1 1 1 -1 1 -1 1 1
-1 1 1 1 -1 1 1 1 -1 1 1 1 -1 1 1 1 -1 1 1 1 -1 1 1 1 -1 1 1 1 -1 1 1
1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 -1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 1 1 -1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 -1 1 -1 1 -1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 1 1 1 1 1 1 -1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 -1 1 1 1 1 1 -1 1 -1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 1 1 -1 1 1 1 -1 1 1 1 -1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 1 1