Shuffle algebras and lattice paths

IPAM workshop "Vertex Models: Algebraic and Probabilistic Aspects of Universality"

Alexandr Garbali, University of Melbourne, May 2024

Based on works with Paul Zinn-Justin and Ajeeth Gunna

Overview

- Lattice paths
- Partition functions
- An algebraic tool
- Computation of partition functions
- Application to skew Macdonald polynomials

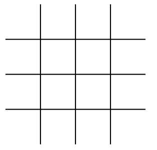
Lattice paths

Lattice paths

Consider lattice paths on square lattice

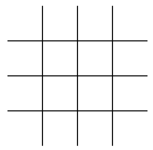
Lattice paths

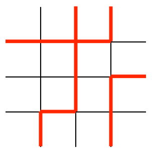
Consider lattice paths on square lattice



Lattice paths

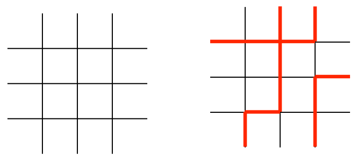
Consider lattice paths on square lattice





Lattice paths

Consider lattice paths on square lattice

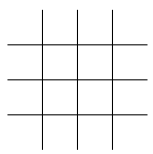


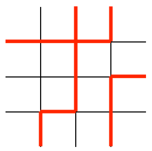
Rules:

- Path starts at a bottom or left boundary edge
- Path ends at a top or right edge
- Path makes unit steps at vertex:
  - straight
  - turning north
  - turning east
- Paths do not share edges

Lattice paths

Consider lattice paths on square lattice





Rules:

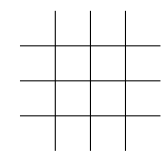
- Path starts at a bottom or left boundary edge
- Path ends at a top or right edge
- Path makes unit steps at vertex:
  straight
  turning north
  turning east
- Paths do not share edges

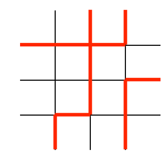
Consider all lattice paths with fixed starting and ending points.

Use 0/1 to denote unoccupied/occupied boundary edges

Lattice paths

Consider lattice paths on square lattice



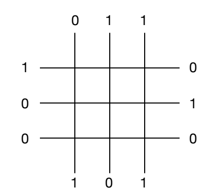


Rules:

- Path starts at a bottom or left boundary edge
- Path ends at a top or right edge
- Path makes unit steps at vertex:
  - straight
  - turning north
  - turning east
- Paths do not share edges

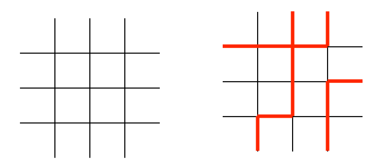
Consider all lattice paths with fixed starting and ending points.

Use 0/1 to denote unoccupied/occupied boundary edges



Lattice paths

Consider lattice paths on square lattice

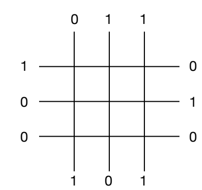


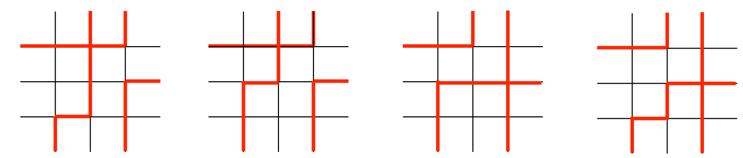
Rules:

- Path starts at a bottom or left boundary edge
- Path ends at a top or right edge
- Path makes unit steps at vertex:
  straight
  turning north
  turning east
- Paths do not share edges

Consider all lattice paths with fixed starting and ending points.

Use 0/1 to denote unoccupied/occupied boundary edges





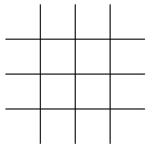
Coloured lattice paths

Coloured lattice paths

Consider coloured lattice paths on square lattice

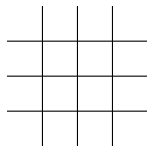
Coloured lattice paths

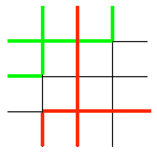
Consider coloured lattice paths on square lattice



Coloured lattice paths

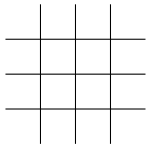
Consider coloured lattice paths on square lattice

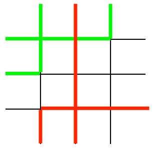




Coloured lattice paths

Consider coloured lattice paths on square lattice





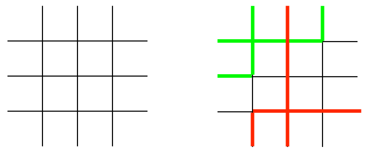
Rules:

- Path starts at a bottom or left boundary edge
- Path ends at a top or right edge
- Path makes unit steps:

  straight
  turning north
  turning east
- Paths do not share edges

Coloured lattice paths

Consider coloured lattice paths on square lattice



Rules:

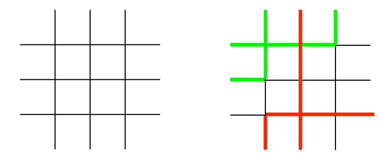
- Path starts at a bottom or left boundary edge
- Path ends at a top or right edge
- Path makes unit steps:
  straight
  turning north
  turning east
- Paths do not share edges

Consider all lattice paths with fixed starting and ending points.

Use 0/1/2 to denote none/red/green path on boundary edges

Coloured lattice paths

Consider coloured lattice paths on square lattice

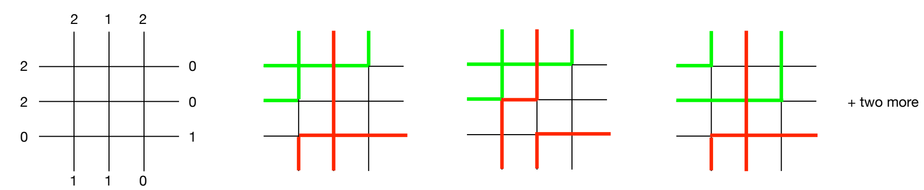


Rules:

- Path starts at a bottom or left boundary edge
- Path ends at a top or right edge
- Path makes unit steps:
  straight
  turning north
  turning east
- Paths do not share edges

Consider all lattice paths with fixed starting and ending points.

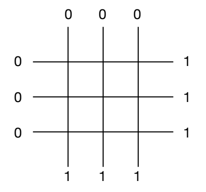
Use 0/1/2 to denote none/red/green path on boundary edges



Lattice paths: special boundary conditions

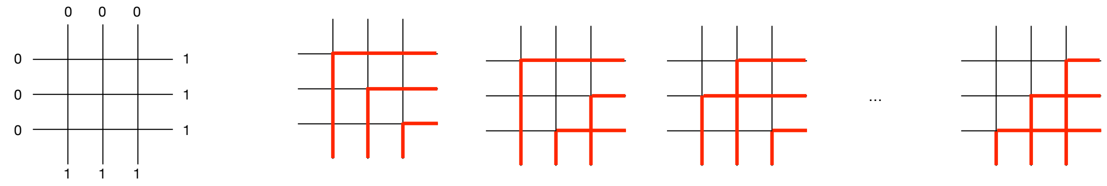
Lattice paths: special boundary conditions

Domain wall boundary conditions:



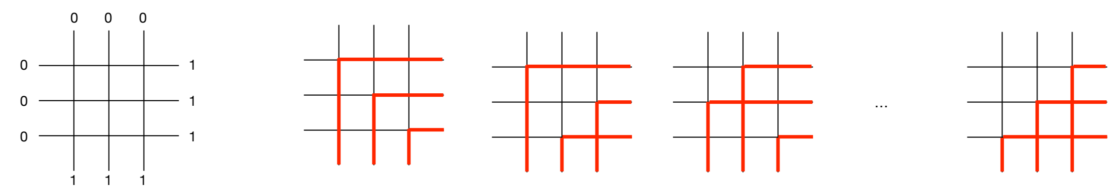
Lattice paths: special boundary conditions

Domain wall boundary conditions:

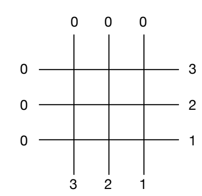


Lattice paths: special boundary conditions

Domain wall boundary conditions:

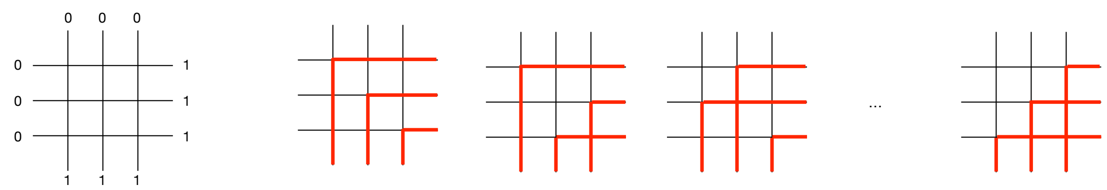


Rainbow domain wall boundary conditions:

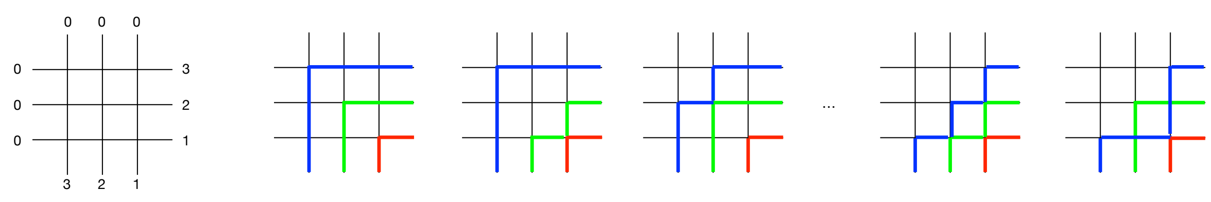


Lattice paths: special boundary conditions

Domain wall boundary conditions:



Rainbow domain wall boundary conditions:

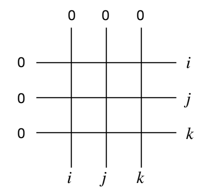


Lattice paths: special boundary conditions

Lattice paths: special boundary conditions

Trace boundary conditions:

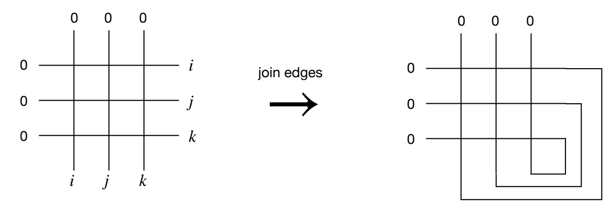
$$\sum_{i,j,k} z_i\ z_j\ z_k$$



Lattice paths: special boundary conditions

Trace boundary conditions:

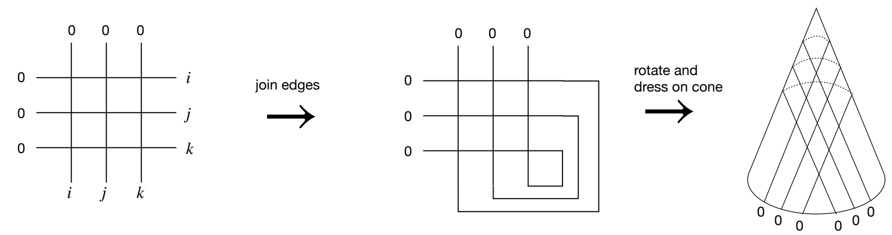
$$\sum_{i,j,k} z_i \, z_j \, z_k$$



Lattice paths: special boundary conditions

Trace boundary conditions:

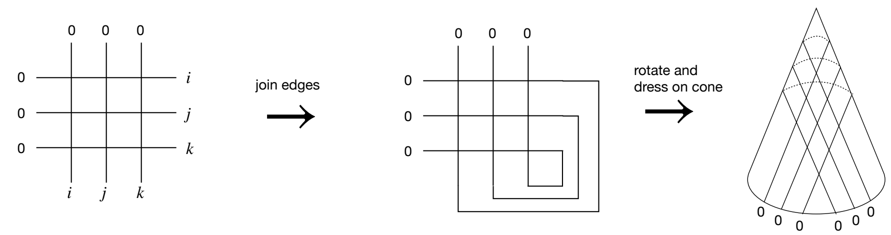
$$\sum_{i,j,k} z_i \, z_j \, z_k$$



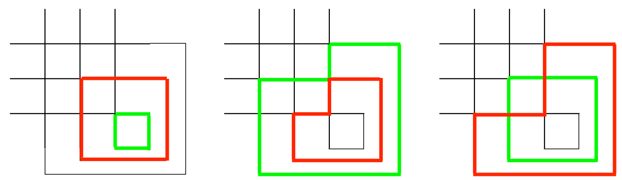
Lattice paths: special boundary conditions

Trace boundary conditions:

$$\sum_{i,j,k} z_i\, z_j\, z_k$$



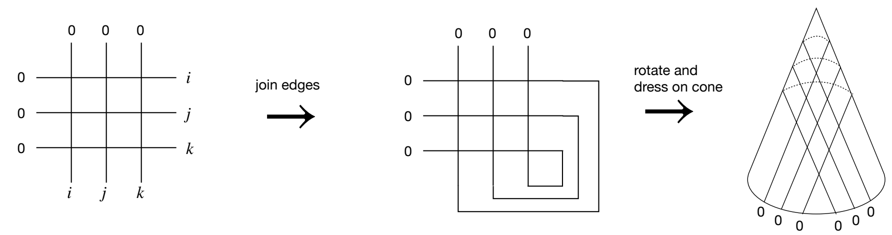
Picking coefficient of $z_1 z_2$ gives 32 configurations among which:



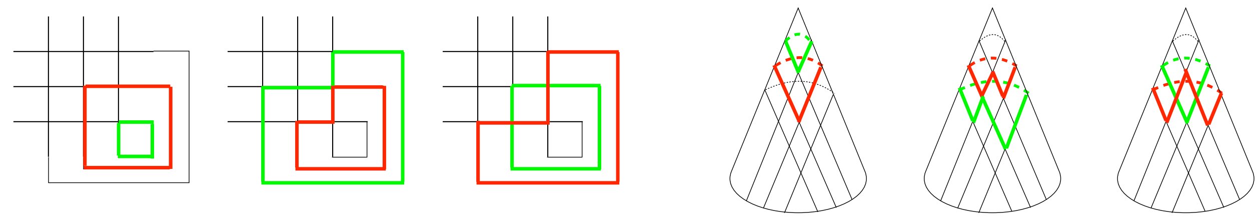
Lattice paths: special boundary conditions

Trace boundary conditions:

$$\sum_{i,j,k} z_i \, z_j \, z_k$$



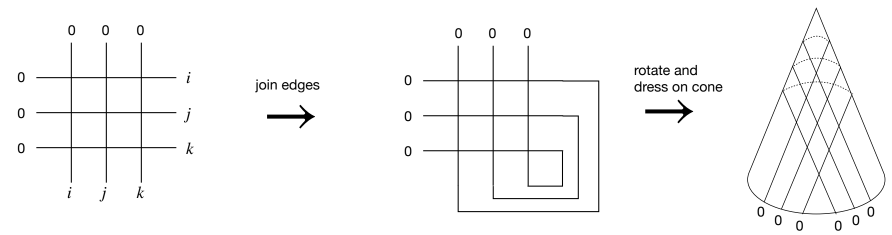
Picking coefficient of $z_1 z_2$ gives 32 configurations among which:



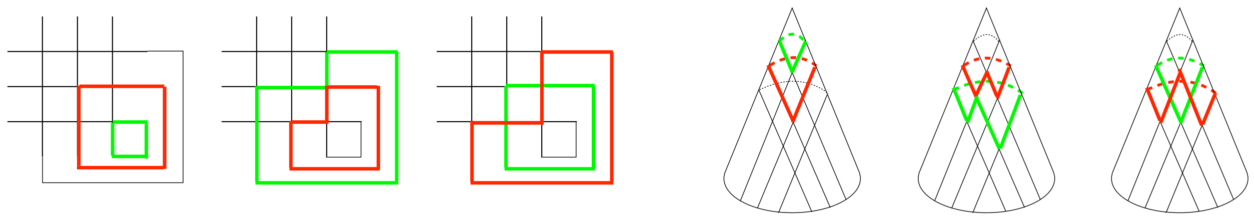
Lattice paths: special boundary conditions

Trace boundary conditions:

$$\sum_{i,j,k} z_i \, z_j \, z_k$$



Picking coefficient of $z_1 z_2$ gives 32 configurations among which:



The coefficients of z_i^3 produce domain wall boundaries for paths of colour i.

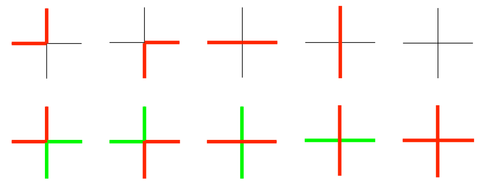
Lattice paths and partition functions

Lattice paths and partition functions

The rules for drawing lattice paths imply the following local configurations where *red* and *green* are <u>any</u> two colours i and j (assume that $i < j$).

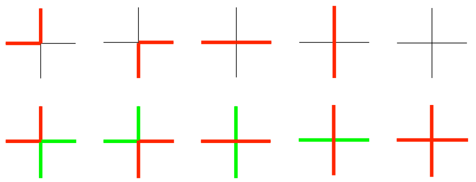
Lattice paths and partition functions

The rules for drawing lattice paths imply the following local configurations where *red* and *green* are <u>any</u> two colours i and j (assume that $i < j$).



Lattice paths and partition functions

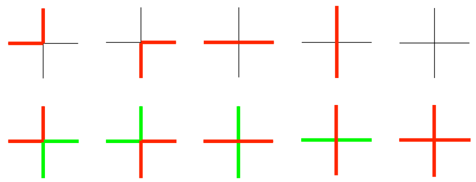
The rules for drawing lattice paths imply the following local configurations where *red* and *green* are <u>any</u> two colours i and j (assume that $i < j$).



Introduce two parameters $u, t \in \mathbb{C}$.
To each local vertex assign Boltzmann weights:

Lattice paths and partition functions

The rules for drawing lattice paths imply the following local configurations where *red* and *green* are any two colours i and j (assume that $i < j$).

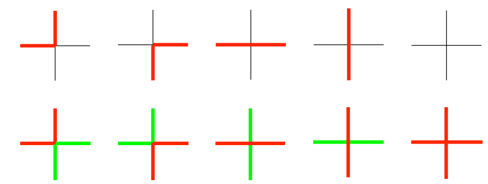


Introduce two parameters $u, t \in \mathbb{C}$.
To each local vertex assign Boltzmann weights:

$\frac{1-t}{1-tu}$	$\frac{(1-t)u}{1-tu}$	$\frac{t(1-u)}{1-tu}$	$\frac{1-u}{1-tu}$	1

Lattice paths and partition functions

The rules for drawing lattice paths imply the following local configurations where *red* and *green* are any two colours i and j (assume that $i < j$).



Introduce two parameters $u, t \in \mathbb{C}$.
To each local vertex assign Boltzmann weights:

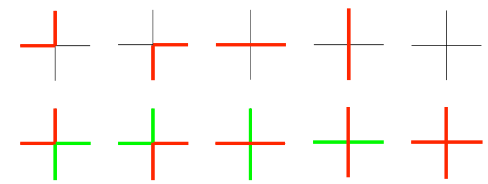
$$\frac{1-t}{1-tu} \qquad \frac{(1-t)u}{1-tu} \qquad \frac{t(1-u)}{1-tu} \qquad \frac{1-u}{1-tu} \qquad 1$$

Fix boundary conditions on edges of $N \times N$ lattice.
Let $\mathscr{C}$ be the set of all configurations.

The *partition function* Z is the sum over $\mathscr{C}$ weighted by the product of local Boltzmann weights.

Lattice paths and partition functions

The rules for drawing lattice paths imply the following local configurations where *red* and *green* are <u>any</u> two colours i and j (assume that $i < j$).



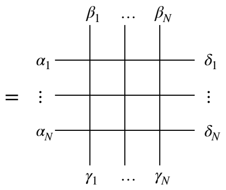
Introduce two parameters $u, t \in \mathbb{C}$.
To each local vertex assign Boltzmann weights:

$$\frac{1-t}{1-tu} \qquad \frac{(1-t)u}{1-tu} \qquad \frac{t(1-u)}{1-tu} \qquad \frac{1-u}{1-tu} \qquad 1$$

Fix boundary conditions on edges of $N \times N$ lattice.
Let $\mathscr{C}$ be the set of all configurations.

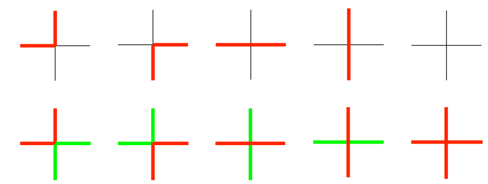
The *partition function* Z is the sum over $\mathscr{C}$ weighted by the product of local Boltzmann weights.

$Z^{\beta,\delta}_{\alpha,\gamma} =$



Lattice paths and partition functions

The rules for drawing lattice paths imply the following local configurations where *red* and *green* are <u>any</u> two colours i and j (assume that $i < j$).



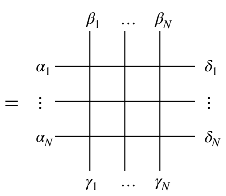
Introduce two parameters $u, t \in \mathbb{C}$.
To each local vertex assign Boltzmann weights:

$$\frac{1-t}{1-tu} \qquad \frac{(1-t)u}{1-tu} \qquad \frac{t(1-u)}{1-tu} \qquad \frac{1-u}{1-tu} \qquad 1$$

Fix boundary conditions on edges of $N \times N$ lattice.
Let $\mathscr{C}$ be the set of all configurations.

The *partition function* Z is the sum over $\mathscr{C}$ weighted by the product of local Boltzmann weights.

$$Z^{\beta,\delta}_{\alpha,\gamma} =$$



$$= \sum_{c \in \mathscr{C}} \prod_{i,j=1}^{N} w_{i,j}(c)$$

Example

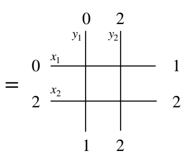
Example

We consider inhomogeneous partition functions: the u parameter is replaced with $u_{i,j} = x_i/y_j$, with (i, j) being the position of the vertex.

Example

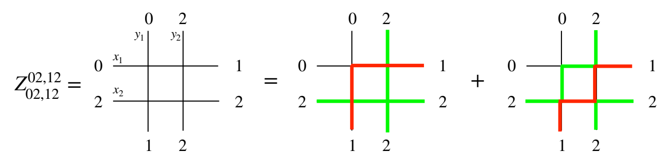
We consider inhomogeneous partition functions: the u parameter is replaced with $u_{i,j} = x_i/y_j$, with (i, j) being the position of the vertex.

$Z^{02,12}_{02,12} =$



Example

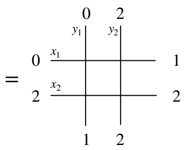
We consider inhomogeneous partition functions: the u parameter is replaced with $u_{i,j} = x_i/y_j$, with (i, j) being the position of the vertex.



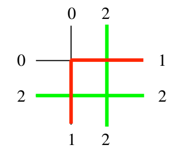
Example

We consider inhomogeneous partition functions: the u parameter is replaced with $u_{i,j} = x_i/y_j$, with (i, j) being the position of the vertex.

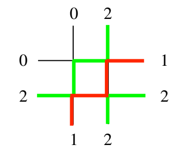
$Z^{02,12}_{02,12} =$



$=$



$+$



$$= \frac{(1-t)x_1/y_1\left(1-x_2/y_1\right)t\left(1-x_1/y_2\right)\left(x_2/y_2-t\right)}{\left(1-tx_1/y_1\right)\left(1-tx_2/y_1\right)\left(1-tx_1/y_2\right)\left(1-tx_2/y_2\right)} + \frac{(1-t)^4 x_1/y_1\, x_2/y_1\, x_1/y_2}{\left(1-tx_1/y_1\right)\left(1-tx_2/y_1\right)\left(1-tx_1/y_2\right)\left(1-tx_2/y_2\right)}$$

R-matrix

R-matrix

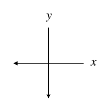
An efficient way of computing these partition functions is with matrix multiplication.

R-matrix

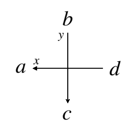
An efficient way of computing these partition functions is with matrix multiplication.

Define the R-matrix:

$$\check{R}(x/y) =$$



$$= \sum_{a,b,c,d=0}^{n}$$

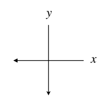


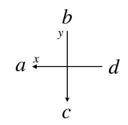
$$|a, c\rangle\langle b, d|$$

R-matrix

An efficient way of computing these partition functions is with matrix multiplication.

Define the R-matrix:

$\check{R}(x/y) =$ 

$= \sum_{a,b,c,d=0}^{n}$ 

$|a,c\rangle\langle b,d|$

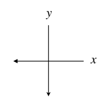
For $n = 1$:

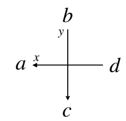
$$\check{R}(x/y) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{(1-t)x}{y-tx} & \frac{y-x}{y-tx} & 0 \\ 0 & \frac{t(y-x)}{y-tx} & \frac{(1-t)y}{y-tx} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

R-matrix

An efficient way of computing these partition functions is with matrix multiplication.

Define the R-matrix:

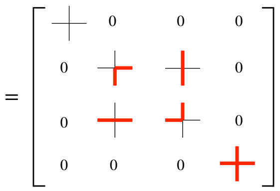
$\check{R}(x/y) =$ 

$= \sum_{a,b,c,d=0}^{n}$ 

$|a,c\rangle\langle b,d|$

For $n = 1$:

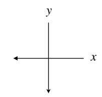
$$\check{R}(x/y) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{(1-t)x}{y-tx} & \frac{y-x}{y-tx} & 0 \\ 0 & \frac{t(y-x)}{y-tx} & \frac{(1-t)y}{y-tx} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

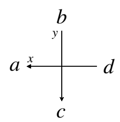


R-matrix

An efficient way of computing these partition functions is with matrix multiplication.

Define the R-matrix:

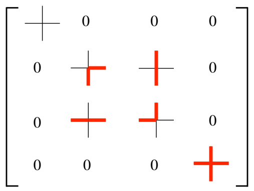
$\check{R}(x/y) =$ 

$= \sum_{a,b,c,d=0}^{n}$ 

$|a,c\rangle\langle b,d|$

For $n = 1$:

$$\check{R}(x/y) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{(1-t)x}{y-tx} & \frac{y-x}{y-tx} & 0 \\ 0 & \frac{t(y-x)}{y-tx} & \frac{(1-t)y}{y-tx} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

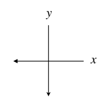


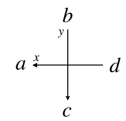
Role of arrows: order of matrix multiplication follows the flow of arrows.

R-matrix

An efficient way of computing these partition functions is with matrix multiplication.

Define the R-matrix:

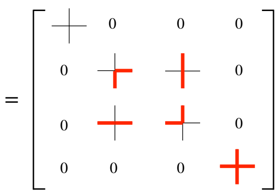
$\check{R}(x/y) =$ 

$= \sum_{a,b,c,d=0}^{n}$ 

$|a,c\rangle\langle b,d|$

For $n = 1$:

$$\check{R}(x/y) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{(1-t)x}{y-tx} & \frac{y-x}{y-tx} & 0 \\ 0 & \frac{t(y-x)}{y-tx} & \frac{(1-t)y}{y-tx} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$



Role of arrows: order of matrix multiplication follows the flow of arrows.

The R-matrix satisfies the *Yang—Baxter equation*:

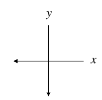
$$\check{R}_1(z/y)\check{R}_2(z/x)\check{R}_1(y/x) = \check{R}_2(y/x)\check{R}_1(z/x)\check{R}_2(z/y)$$

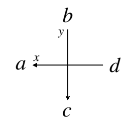
where $\check{R}_1(u) = \check{R}(u) \otimes I$ and $\check{R}_2(u) = I \otimes \check{R}(u)$

R-matrix

An efficient way of computing these partition functions is with matrix multiplication.

Define the R-matrix:

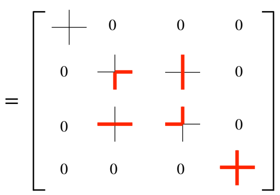
$\check{R}(x/y) =$ 

$= \sum_{a,b,c,d=0}^{n}$ 

$|a,c\rangle\langle b,d|$

For $n = 1$:

$$\check{R}(x/y) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{(1-t)x}{y-tx} & \frac{y-x}{y-tx} & 0 \\ 0 & \frac{t(y-x)}{y-tx} & \frac{(1-t)y}{y-tx} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

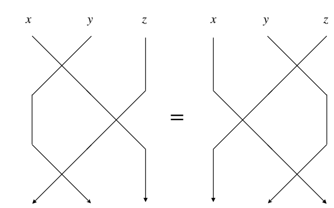


Role of arrows: order of matrix multiplication follows the flow of arrows.

The R-matrix satisfies the *Yang—Baxter equation*:

$$\check{R}_1(z/y)\check{R}_2(z/x)\check{R}_1(y/x) = \check{R}_2(y/x)\check{R}_1(z/x)\check{R}_2(z/y)$$

where $\check{R}_1(u) = \check{R}(u) \otimes I$ and $\check{R}_2(u) = I \otimes \check{R}(u)$



Partition functions

Partition functions

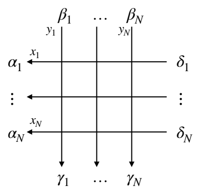
We compute any partition function by multiplying R-matrices and taking matrix elements.

$$Z_N(x;y) := \check{R}_N(x_N/y_1)\check{R}_{N-1}(x_{N-1}/y_1)\check{R}_{N+1}(x_N/y_2)\cdots\check{R}_N(x_1/y_N)$$

Partition functions

We compute any partition function by multiplying R-matrices and taking matrix elements.

$$Z_N(x;y) := \check{R}_N(x_N/y_1)\check{R}_{N-1}(x_{N-1}/y_1)\check{R}_{N+1}(x_N/y_2)\cdots\check{R}_N(x_1/y_N)$$

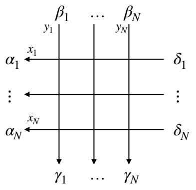


$$= \langle \alpha, \gamma \,|\, Z_N(x,y) \,|\, \beta, \delta \rangle$$

Partition functions

We compute any partition function by multiplying R-matrices and taking matrix elements.

$$Z_N(x;y) := \check{R}_N(x_N/y_1)\check{R}_{N-1}(x_{N-1}/y_1)\check{R}_{N+1}(x_N/y_2)\cdots\check{R}_N(x_1/y_N)$$



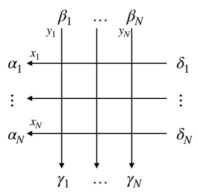
$$= \langle \alpha, \gamma \,|\, Z_N(x,y) \,|\, \beta, \delta \rangle$$

Let $q \in \mathbb{C}$. In the following we specialize: $y_i = qx_i$.

Partition functions

We compute any partition function by multiplying R-matrices and taking matrix elements.

$$Z_N(x; y) := \check{R}_N(x_N/y_1)\check{R}_{N-1}(x_{N-1}/y_1)\check{R}_{N+1}(x_N/y_2)\cdots\check{R}_N(x_1/y_N)$$



$$= \langle \alpha, \gamma \,|\, Z_N(x, y) \,|\, \beta, \delta \rangle$$

Let $q \in \mathbb{C}$. In the following we specialize: $y_i = qx_i$.

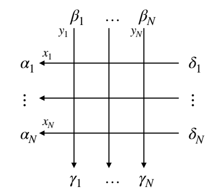
The conic partition function is a partial trace of Z_N:

$$T_N(x; z_0 \ldots z_n) := \sum_{\alpha \in \{0\ldots n\}^N} z_{\alpha_1} \cdots z_{\alpha_N} \langle 0^N, \alpha \,|\, Z_N(x; qx) \,|\, 0^N, \alpha \rangle$$

Partition functions

We compute any partition function by multiplying R-matrices and taking matrix elements.

$$Z_N(x;y) := \check{R}_N(x_N/y_1)\check{R}_{N-1}(x_{N-1}/y_1)\check{R}_{N+1}(x_N/y_2)\cdots\check{R}_N(x_1/y_N)$$

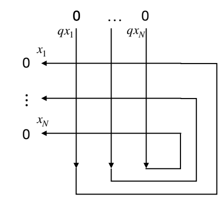


$$= \langle \alpha, \gamma \,|\, Z_N(x,y) \,|\, \beta, \delta \rangle$$

Let $q \in \mathbb{C}$. In the following we specialize: $y_i = qx_i$.

The conic partition function is a partial trace of Z_N:

$$T_N(x; z_0 \ldots z_n) := \sum_{\alpha \in \{0\ldots n\}^N} z_{\alpha_1} \ \cdots \ z_{\alpha_N} \langle 0^N, \alpha \,|\, Z_N(x; qx) \,|\, 0^N, \alpha \rangle$$



Conic partition function

Conic partition function

Goal. Compute the grand canonical partition function on the cone:

$$T(v \,|\, x; z) = \sum_{N=0}^{\infty} v^N T_N(x; z)$$

Conic partition function

Goal. Compute the grand canonical partition function on the cone:

$$T(v\,|\,x;z) = \sum_{N=0}^{\infty} v^N T_N(x;z)$$

In non-interacting many body systems grand canonical partition functions can be resumed and written in exponential form:

$$\Xi = \sum_{N=0}^{\infty} \sum_{\text{states}} e^{-\beta(E_N - \mu N)} = e^{\Phi(T,\mu)}$$

Conic partition function

Goal. Compute the grand canonical partition function on the cone:

$$T(v\,|\,x;z) = \sum_{N=0}^{\infty} v^N T_N(x;z)$$

In non-interacting many body systems grand canonical partition functions can be resumed and written in exponential form:

$$\Xi = \sum_{N=0}^{\infty} \sum_{\text{states}} e^{-\beta(E_N-\mu N)} = e^{\Phi(T,\mu)}$$

Our problem involves an interacting system but Yang—Baxter integrable. We find a *shuffle-exponential* expression for $T(v\,|\,x;z)$.

Conic partition function

Goal. Compute the grand canonical partition function on the cone:

$$T(v\,|\,x;z) = \sum_{N=0}^{\infty} v^N T_N(x;z)$$

In non-interacting many body systems grand canonical partition functions can be resumed and written in exponential form:

$$\Xi = \sum_{N=0}^{\infty} \sum_{\text{states}} e^{-\beta(E_N-\mu N)} = e^{\Phi(T,\mu)}$$

Our problem involves an interacting system but Yang—Baxter integrable. We find a *shuffle-exponential* expression for $T(v\,|\,x;z)$.

Theorem [AG, A Gunna '23]:

$$T(v\,|\,x;z) = \exp_*\left(-\sum_{k>0} \frac{v^k}{k}\left(z_1^k + \cdots + z_n^k + \frac{q^k - t^k}{1-t^k} z_0^k\right) L_k(x_1 \ldots x_k)\right)$$

where $\exp_* A = 1 + A + 1/2!A * A + 1/3!A * A * A + \cdots$ and $L_k(x_1 \ldots x_k)$ is another "small" conic partition function (with two types of paths) which has and explicit expression as a rational function in x_i.

The shuffle algebra

The shuffle algebra

Consider the vector space of symmetric rational functions: $V = \bigoplus_{k \geq 0} \mathbb{Q}(q,t)(x_1 \dots x_k)^{S_n}$

The shuffle algebra

Consider the vector space of symmetric rational functions: $V = \bigoplus_{k \geq 0} \mathbb{Q}(q, t)(x_1 \ldots x_k)^{\mathcal{S}_n}$

Multiplication of two elements $F, G \in V$:

$$F(x_1 \ldots x_k) * G(x_1 \ldots x_l) = \sum_{\sigma \in \mathcal{S}_{k+l}/\mathcal{S}_k \times \mathcal{S}_l} \sigma \left(F(x_1 \ldots x_k) G(x_{k+1} \ldots x_{k+l}) \prod_{\substack{i \in 1 \ldots k \\ j \in k+1 \ldots k+l}} \frac{(x_j - q x_i)(x_j - t^{-1} x_i)}{(x_j - x_i)(x_i - q t^{-1} x_i)} \right)$$

The shuffle algebra

Consider the vector space of symmetric rational functions: $V = \bigoplus_{k \geq 0} \mathbb{Q}(q,t)(x_1 \ldots x_k)^{\mathcal{S}_n}$

Multiplication of two elements $F, G \in V$:

$$F(x_1 \ldots x_k) * G(x_1 \ldots x_l) = \sum_{\sigma \in \mathcal{S}_{k+l}/\mathcal{S}_k \times \mathcal{S}_l} \sigma \left(F(x_1 \ldots x_k) G(x_{k+1} \ldots x_{k+l}) \prod_{\substack{i \in 1 \ldots k \\ j \in k+1 \ldots k+l}} \frac{(x_j - qx_i)(x_j - t^{-1}x_i)}{(x_j - x_i)(x_i - qt^{-1}x_i)} \right)$$

The shuffle algebra $\mathcal{A}$ is the subspace in V of elements of the form:

$$P(x_1 \ldots x_k) = \frac{p(x_1 \ldots x_k)}{\prod_{1 \leq i \neq j \leq k} (x_i - qt^{-1}x_j)} \qquad \text{s.t.:} \qquad p(\ldots x, qt^{-1}x, t^{-1}x \ldots) = p(\ldots x, qt^{-1}x, qx \ldots) = 0$$

The shuffle algebra

Consider the vector space of symmetric rational functions: $V = \bigoplus_{k \geq 0} \mathbb{Q}(q,t)(x_1 \ldots x_k)^{\mathcal{S}_n}$

Multiplication of two elements $F, G \in V$:

$$F(x_1 \ldots x_k) * G(x_1 \ldots x_l) = \sum_{\sigma \in \mathcal{S}_{k+l}/\mathcal{S}_k \times \mathcal{S}_l} \sigma \left(F(x_1 \ldots x_k) G(x_{k+1} \ldots x_{k+l}) \prod_{\substack{i \in 1 \ldots k \\ j \in k+1 \ldots k+l}} \frac{(x_j - qx_i)(x_j - t^{-1}x_i)}{(x_j - x_i)(x_i - qt^{-1}x_i)} \right)$$

The shuffle algebra $\mathcal{A}$ is the subspace in V of elements of the form:

$$P(x_1 \ldots x_k) = \frac{p(x_1 \ldots x_k)}{\prod_{1 \leq i \neq j \leq k} (x_i - qt^{-1}x_j)} \quad \text{s.t.:} \quad p(\ldots x, qt^{-1}x, t^{-1}x \ldots) = p(\ldots x, qt^{-1}x, qx \ldots) = 0$$

The commutative shuffle algebra is a subspace $\mathcal{A}^{\circ} \subset \mathcal{A}$ such that:

$$\lim_{\epsilon \to 0} P(\epsilon^{\pm 1} x_1, \ldots, \epsilon^{\pm 1} x_r, x_{r+1}, \ldots, x_n) = \kappa < \infty$$

Feigin—Odesskii, FHHSY

Commutative shuffle algebra

Commutative shuffle algebra

The simplest elements of $\mathscr{A}^\circ$ are the factorized elements:
[Feigin—Odesskii]

$$E_k(x;p) := \prod_{1\le i<j\le k} \frac{(x_i - px_j)(x_i - p^{-1}x_j)}{(x_i - qt^{-1}x_j)(x_i - tq^{-1}x_j)}, \qquad p = q, t^{-1}, tq^{-1}$$

Commutative shuffle algebra

The simplest elements of $\mathscr{A}^\circ$ are the factorized elements:
[Feigin—Odesskii]

$$E_k(x;p) := \prod_{1\le i<j\le k} \frac{(x_i - p x_j)(x_i - p^{-1}x_j)}{(x_i - qt^{-1}x_j)(x_i - tq^{-1}x_j)}, \qquad p = q, t^{-1}, tq^{-1}$$

Let (p, p', p'') be a permutation of (q, t^{-1}, tq^{-1}).
Another example of elements of $\mathscr{A}^\circ$ is given by determinants:
[Izergin]

$$H_k(x;p) := f(x) \det_{1\le i,j\le k} \frac{1}{(x_i - p' x_j)(x_j - p'' x_i)}$$

$f(x)$ is composed of simple factors which fix the poles s.t.: $H_k \in \mathscr{A}^\circ$

Commutative shuffle algebra

The simplest elements of $\mathscr{A}^\circ$ are the factorized elements:
[Feigin—Odesskii]

$$E_k(x;p) := \prod_{1\le i<j\le k} \frac{(x_i - p x_j)(x_i - p^{-1}x_j)}{(x_i - qt^{-1}x_j)(x_i - tq^{-1}x_j)}, \qquad p = q, t^{-1}, tq^{-1}$$

Let (p, p', p'') be a permutation of (q, t^{-1}, tq^{-1}).
Another example of elements of $\mathscr{A}^\circ$ is given by determinants:
[Izergin]

$$H_k(x;p) := f(x) \det_{1\le i,j\le k} \frac{1}{(x_i - p'x_j)(x_j - p''x_i)}$$

$f(x)$ is composed of simple factors which fix the poles s.t.: $H_k \in \mathscr{A}^\circ$

A third type of elements given by a symmetrization formula:
[Negut]

$$S_k(x) := c_k(q,t) \sum_{\sigma\in\mathcal{S}_k} \sigma\left(\frac{\sum_{j=0}^{k-1} (qt^{-1})^j x_{j+1}/x_1}{\prod_{j=1}^{k-1}\left(1 - qt^{-1}x_{j+1}/x_j\right)} \prod_{1\le i<j\le k} \frac{(x_j - qx_i)(x_j - t^{-1}x_i)}{(x_j - x_i)(x_i - qt^{-1}x_i)} \right)$$

Commutative shuffle algebra

The simplest elements of $\mathscr{A}^\circ$ are the factorized elements:
[Feigin—Odesskii]

$$E_k(x;p) := \prod_{1\le i<j\le k} \frac{(x_i - px_j)(x_i - p^{-1}x_j)}{(x_i - qt^{-1}x_j)(x_i - tq^{-1}x_j)}, \qquad p = q, t^{-1}, tq^{-1}$$

Let (p, p', p'') be a permutation of (q, t^{-1}, tq^{-1}).
Another example of elements of $\mathscr{A}^\circ$ is given by determinants:
[Izergin]

$$H_k(x;p) := f(x) \det_{1\le i,j\le k} \frac{1}{(x_i - p'x_j)(x_j - p''x_i)}$$

$f(x)$ is composed of simple factors which fix the poles s.t.: $H_k \in \mathscr{A}^\circ$

A third type of elements given by a symmetrization formula:
[Negut]

$$S_k(x) := c_k(q,t) \sum_{\sigma\in\mathfrak{S}_k} \sigma \left(\frac{\sum_{j=0}^{k-1} (qt^{-1})^j x_{j+1}/x_1}{\prod_{j=1}^{k-1}\left(1 - qt^{-1}x_{j+1}/x_j\right)} \prod_{1\le i<j\le k} \frac{(x_j - qx_i)(x_j - t^{-1}x_i)}{(x_j - x_i)(x_i - qt^{-1}x_i)} \right)$$

Lemma: The generating functions of $E_k(x;p), H_k(x;p)$ are equal to shuffle-exponentials:

$$E(v\,|\,p) = \sum_{k=0}^{\infty} v^k E_k(x;p) = \exp_*\left(\sum_{r>0} \frac{(-1)^{r+1}}{r} d_r v^r S_r(x) \right), \qquad H(v\,|\,p) = \sum_{k=0}^{\infty} v^k H_k(x;p) = \exp_*\left(\sum_{r>0} \frac{1}{r} d_r v^r S_r(x) \right)$$

where:
$$d_r = \frac{1-p^r}{1-q^r}\frac{(t-q)^r}{(1-p)^r}$$

Partition functions as elements of $\mathscr{A}^\circ$

Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x; z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

$$T_N(x; z) = \sum_\lambda z^\lambda T_\lambda(x)$$

Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x; z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

$$T_N(x; z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:

λ_1 loops of colour 1

$\dots$

λ_n loops of colour n

$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
…
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

Lemma: $T_\lambda(x) = T_\lambda(x_1 \ldots x_N) \in \mathscr{A}^\circ$

Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
$\ldots$
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

<u>Lemma</u>: $T_\lambda(x) = T_\lambda(x_1 \ldots x_N) \in \mathscr{A}^\circ$

Proof:

Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
…
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

Lemma: $T_\lambda(x) = T_\lambda(x_1 \ldots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \ldots x_N)$ is a symmetric function in x's.

Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

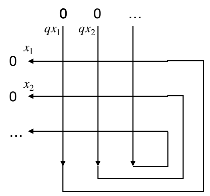
$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
$\ldots$
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

<u>Lemma</u>: $T_\lambda(x) = T_\lambda(x_1 \ldots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \ldots x_N)$ is a symmetric function in x's.



Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

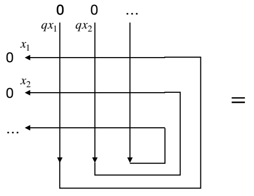
$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
$\dots$
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

Lemma: $T_\lambda(x) = T_\lambda(x_1 \dots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \dots x_N)$ is a symmetric function in x's.



Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

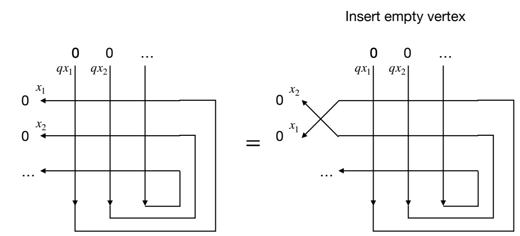
$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
$\dots$
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

Lemma: $T_\lambda(x) = T_\lambda(x_1 \dots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \dots x_N)$ is a symmetric function in x's.



Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

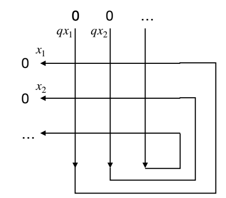
$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
…
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

<u>Lemma</u>: $T_\lambda(x) = T_\lambda(x_1 \ldots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \ldots x_N)$ is a symmetric function in x's.



$=$

Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:

λ_1 loops of colour 1

$\ldots$

λ_n loops of colour n

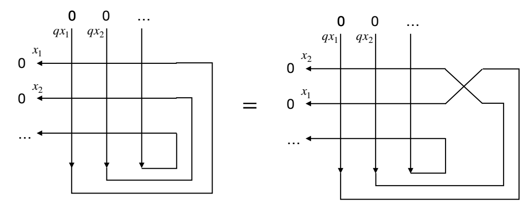
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ “empty” loops

<u>Lemma</u>: $T_\lambda(x) = T_\lambda(x_1 \ldots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \ldots x_N)$ is a symmetric function in x's.

Use YB equation



Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

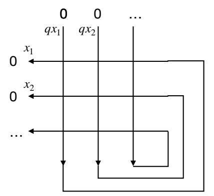
$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
$\dots$
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

Lemma: $T_\lambda(x) = T_\lambda(x_1 \dots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \dots x_N)$ is a symmetric function in x's.



$=$

Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

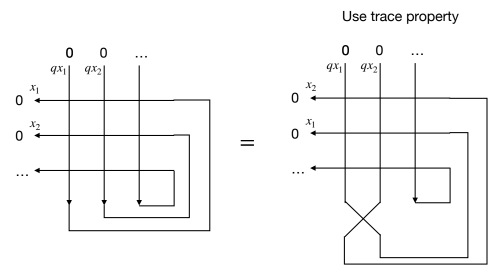
$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
$\ldots$
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

<u>Lemma</u>: $T_\lambda(x) = T_\lambda(x_1 \ldots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \ldots x_N)$ is a symmetric function in x's.



Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x; z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

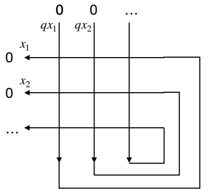
$$T_N(x; z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
$\dots$
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

Lemma: $T_\lambda(x) = T_\lambda(x_1 \dots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \dots x_N)$ is a symmetric function in x's.



$=$

Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:

λ_1 loops of colour 1

$\ldots$

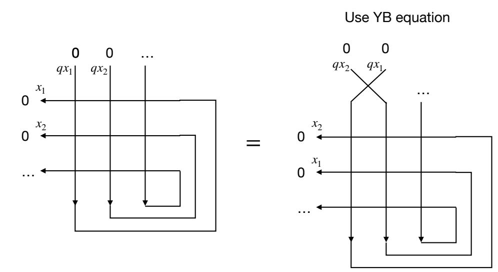
λ_n loops of colour n

$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

<u>Lemma</u>: $T_\lambda(x) = T_\lambda(x_1 \ldots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \ldots x_N)$ is a symmetric function in x's.



Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

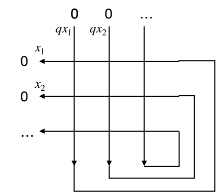
$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
$\dots$
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

Lemma: $T_\lambda(x) = T_\lambda(x_1 \dots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \dots x_N)$ is a symmetric function in x's.



$=$

Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

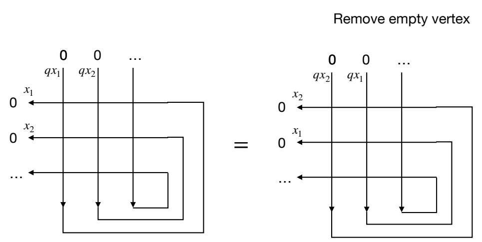
$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
$\dots$
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

Lemma: $T_\lambda(x) = T_\lambda(x_1 \dots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \dots x_N)$ is a symmetric function in x's.



Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

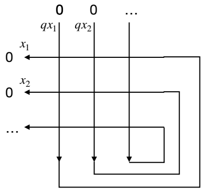
$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
…
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

Lemma: $T_\lambda(x) = T_\lambda(x_1 \ldots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \ldots x_N)$ is a symmetric function in x's.



$=$

Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

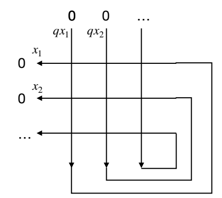
$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
…
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

<u>Lemma</u>: $T_\lambda(x) = T_\lambda(x_1 \ldots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \ldots x_N)$ is a symmetric function in x's.



Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
$\ldots$
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

Lemma: $T_\lambda(x) = T_\lambda(x_1 \ldots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \ldots x_N)$ is a symmetric function in x's.

Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
…
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

Lemma: $T_\lambda(x) = T_\lambda(x_1 \ldots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \ldots x_N)$ is a symmetric function in x's.

2) Wheel conditions. Set $(x_1, x_2, x_3) = (qt^{-1}x, x, t^{-1}x)$

Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

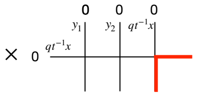
$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
$\dots$
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

Lemma: $T_\lambda(x) = T_\lambda(x_1 \dots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \dots x_N)$ is a symmetric function in x's.

2) Wheel conditions. Set $(x_1, x_2, x_3) = (qt^{-1}x, x, t^{-1}x)$

$\lim_{y_1 \to q^2 t^{-1} x} \lim_{y_2 \to qx} (y_1 - qx)(y_2 - qx) \times$  $= 0$

Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
…
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

Lemma: $T_\lambda(x) = T_\lambda(x_1 \ldots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \ldots x_N)$ is a symmetric function in x's.

2) Wheel conditions. Set $(x_1, x_2, x_3) = (qt^{-1}x, x, t^{-1}x)$

Partition functions as elements of $\mathscr{A}^\circ$

Expand the partition function $T_N(x;z)$ in monomials $z^\lambda := z_0^{\lambda_0} z_1^{\lambda_1} \cdots z_n^{\lambda_n}$:

$$T_N(x;z) = \sum_\lambda z^\lambda T_\lambda(x)$$

$T_\lambda(x)$ is the partition function with fixed loop content:
λ_1 loops of colour 1
$\dots$
λ_n loops of colour n
$\lambda_0 = N - (\lambda_1 + \cdots + \lambda_n)$ "empty" loops

Lemma: $T_\lambda(x) = T_\lambda(x_1 \dots x_N) \in \mathscr{A}^\circ$

Proof:

1) $T_\lambda(x_1 \dots x_N)$ is a symmetric function in x's.

2) Wheel conditions. Set $(x_1, x_2, x_3) = (qt^{-1}x, x, t^{-1}x)$

3) Proof of $\lim_{\epsilon \to 0} T_\lambda(\epsilon^{\pm 1} x_1, \dots, \epsilon^{\pm 1} x_r, x_{r+1}, \dots, x_N) = \kappa < \infty$ is similar in spirit.

Computing T_N

Computing T_N

Consider the case of paths of single colour (six vertex case $n = 1$)

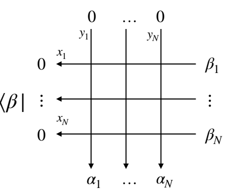
$$T_N(x; z_0, z_1) := \sum_{\alpha \in \{0\ldots1\}^N} z_{\alpha_1} \cdots z_{\alpha_N} \langle \alpha \,|\, W_N(x; qx) \,|\, \alpha \rangle$$

Computing T_N

Consider the case of paths of single colour (six vertex case $n = 1$)

$$T_N(x; z_0, z_1) := \sum_{\alpha \in \{0\ldots1\}^N} z_{\alpha_1} \cdots z_{\alpha_N} \langle \alpha \,|\, W_N(x; qx) \,|\, \alpha \rangle$$

$$W_N(x; y) := \sum_{\alpha,\beta} |\,\alpha\rangle\langle\beta\,|$$

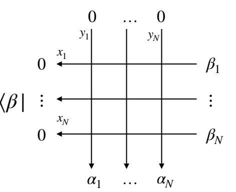


Computing T_N

Consider the case of paths of single colour (six vertex case $n = 1$)

$$T_N(x; z_0, z_1) := \sum_{\alpha \in \{0\ldots 1\}^N} z_{\alpha_1} \cdots z_{\alpha_N} \langle \alpha \,|\, W_N(x; qx) \,|\, \alpha \rangle$$

$$W_N(x; y) := \sum_{\alpha,\beta} |\alpha\rangle\langle\beta|$$



<u>Statement 1:</u> Let P_i be the permutation matrix. There exists a transformation F such that:

$$\widetilde{W}_N(x; y) := F_N(y) W_N(x; y) F_N^{-1}(x)$$

satisfies:

$$\widetilde{W}_N(x; y) P_i = \widetilde{W}_N(\ldots x_{i+1}, x_i \ldots; y)$$
$$P_i \widetilde{W}_N(x; y) = \widetilde{W}_N(x; \ldots y_{i+1}, y_i \ldots)$$

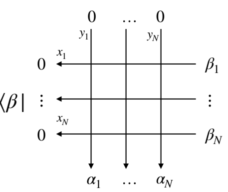
This implies that computing one matrix element of $\widetilde{W}_N$ is enough to recover all of them.

Computing T_N

Consider the case of paths of single colour (six vertex case $n = 1$)

$$T_N(x; z_0, z_1) := \sum_{\alpha \in \{0\dots1\}^N} z_{\alpha_1} \cdots z_{\alpha_N} \langle \alpha \,|\, W_N(x; qx) \,|\, \alpha \rangle$$

$$W_N(x; y) := \sum_{\alpha,\beta} |\,\alpha\rangle\langle\beta\,|$$



Statement 1: Let P_i be the permutation matrix. There exists a transformation F such that:

$$\widetilde{W}_N(x; y) := F_N(y) W_N(x; y) F_N^{-1}(x)$$

satisfies:

$$\widetilde{W}_N(x; y) P_i = \widetilde{W}_N(\dots x_{i+1}, x_i \dots; y)$$
$$P_i \widetilde{W}_N(x; y) = \widetilde{W}_N(x; \dots y_{i+1}, y_i \dots)$$

This implies that computing one matrix element of $\widetilde{W}_N$ is enough to recover all of them.

Statement 2 Denote by $D(x; y)$ the domain wall partition functions, then:

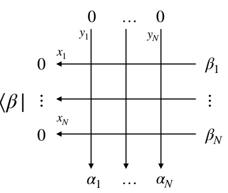
$$\widetilde{W}^{(1^k 0^{N-k})}_{(0^{N-k} 1^k)}(x; y) = \prod_{1 \le i < j \le N} \frac{x_i - t x_j}{x_i - x_j} \times W^{(1^k 0^{N-k})}_{(0^{N-k} 1^k)}(x; y)$$

Computing T_N

Consider the case of paths of single colour (six vertex case $n = 1$)

$$T_N(x; z_0, z_1) := \sum_{\alpha \in \{0 \dots 1\}^N} z_{\alpha_1} \cdots z_{\alpha_N} \langle \alpha \,|\, W_N(x; qx) \,|\, \alpha \rangle$$

$$W_N(x; y) := \sum_{\alpha, \beta} |\alpha\rangle\langle\beta|$$



Statement 1: Let P_i be the permutation matrix. There exists a transformation F such that:

$$\widetilde{W}_N(x; y) := F_N(y) W_N(x; y) F_N^{-1}(x)$$

satisfies:

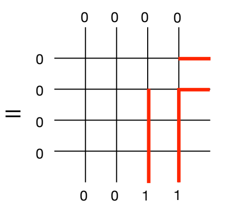
$$\widetilde{W}_N(x; y) P_i = \widetilde{W}_N(\dots x_{i+1}, x_i \dots; y)$$
$$P_i \widetilde{W}_N(x; y) = \widetilde{W}_N(x; \dots y_{i+1}, y_i \dots)$$

This implies that computing one matrix element of $\widetilde{W}_N$ is enough to recover all of them.

Statement 2 Denote by $D(x; y)$ the domain wall partition functions, then:

$$\widetilde{W}^{(1^k 0^{N-k})}_{(0^{N-k} 1^k)}(x; y) = \prod_{1 \le i < j \le N} \frac{x_i - t x_j}{x_i - x_j} \times W^{(1^k 0^{N-k})}_{(0^{N-k} 1^k)}(x; y)$$

$$W^{(1^k 0^{N-k})}_{(0^{N-k} 1^k)}(x; y) =$$

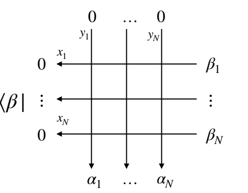


Computing T_N

Consider the case of paths of single colour (six vertex case $n = 1$)

$$T_N(x; z_0, z_1) := \sum_{\alpha \in \{0\ldots1\}^N} z_{\alpha_1} \cdots z_{\alpha_N} \langle \alpha \,|\, W_N(x; qx) \,|\, \alpha \rangle$$

$$W_N(x; y) := \sum_{\alpha,\beta} |\,\alpha\rangle\langle\beta\,|$$



Statement 1: Let P_i be the permutation matrix. There exists a transformation F such that:

$$\widetilde{W}_N(x; y) := F_N(y) W_N(x; y) F_N^{-1}(x)$$

satisfies:

$$\widetilde{W}_N(x; y) P_i = \widetilde{W}_N(\ldots x_{i+1}, x_i \ldots; y)$$
$$P_i \widetilde{W}_N(x; y) = \widetilde{W}_N(x; \ldots y_{i+1}, y_i \ldots)$$

This implies that computing one matrix element of $\widetilde{W}_N$ is enough to recover all of them.

Statement 2 Denote by $D(x; y)$ the domain wall partition functions, then:

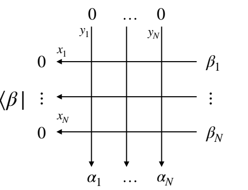
$$\widetilde{W}^{(1^k 0^{N-k})}_{(0^{N-k} 1^k)}(x; y) = \prod_{1 \le i < j \le N} \frac{x_i - t x_j}{x_i - x_j} \times W^{(1^k 0^{N-k})}_{(0^{N-k} 1^k)}(x; y)$$

Computing T_N

Consider the case of paths of single colour (six vertex case $n = 1$)

$$T_N(x; z_0, z_1) := \sum_{\alpha \in \{0\ldots1\}^N} z_{\alpha_1} \cdots z_{\alpha_N} \langle \alpha \,|\, W_N(x; qx) \,|\, \alpha \rangle$$

$$W_N(x; y) := \sum_{\alpha,\beta} |\,\alpha\rangle\langle\beta\,|$$



Statement 1: Let P_i be the permutation matrix. There exists a transformation F such that:

$$\widetilde{W}_N(x; y) := F_N(y) W_N(x; y) F_N^{-1}(x)$$

satisfies:

$$\widetilde{W}_N(x; y) P_i = \widetilde{W}_N(\ldots x_{i+1}, x_i \ldots; y)$$
$$P_i \widetilde{W}_N(x; y) = \widetilde{W}_N(x; \ldots y_{i+1}, y_i \ldots)$$

This implies that computing one matrix element of $\widetilde{W}_N$ is enough to recover all of them.

Statement 2 Denote by $D(x; y)$ the domain wall partition functions, then:

$$\widetilde{W}^{(1^k 0^{N-k})}_{(0^{N-k} 1^k)}(x; y) = \prod_{1 \le i < j \le N} \frac{x_i - t x_j}{x_i - x_j} \times W^{(1^k 0^{N-k})}_{(0^{N-k} 1^k)}(x; y)$$

$$W^{(1^k 0^{N-k})}_{(0^{N-k} 1^k)}(x; y) = \prod_{i,j > k} \frac{y_i - x_j}{y_i - t x_j} \times D(x_1 \ldots x_k; y_{N-k+1} \cdots y_N)$$

Computing T_N

Computing T_N

After simple algebra we get:

$$T_N(x; z_0, z_1) = \sum_{k=0}^{N} z_0^{N-k} z_1^k \frac{(1-t^{-1})^k}{(1-qt^{-1})^k} H_k(t^{-1}) * E_{N-k}(tq^{-1})$$

Computing T_N

After simple algebra we get:

$$T_N(x; z_0, z_1) = \sum_{k=0}^{N} z_0^{N-k} z_1^k \frac{(1-t^{-1})^k}{(1-qt^{-1})^k} H_k(t^{-1}) * E_{N-k}(tq^{-1})$$

Summing $T_N(x; z_0, z_1)$ with the generating parameter gives the shuffle exponential:

$$T(v \,|\, x; z) = \exp_* \left(-\sum_{k>0} \frac{v^k}{k} \left(z_1^k + \frac{q^k - t^k}{1-t^k} z_0^k \right) L_k(x_1 \ldots x_k) \right)$$

Computing T_N

After simple algebra we get:

$$T_N(x; z_0, z_1) = \sum_{k=0}^{N} z_0^{N-k} z_1^k \frac{(1-t^{-1})^k}{(1-qt^{-1})^k} H_k(t^{-1}) * E_{N-k}(tq^{-1})$$

Summing $T_N(x; z_0, z_1)$ with the generating parameter gives the shuffle exponential:

$$T(v\,|\,x; z) = \exp_* \left(-\sum_{k>0} \frac{v^k}{k} \left(z_1^k + \frac{q^k - t^k}{1-t^k} z_0^k \right) L_k(x_1 \ldots x_k) \right)$$

The coloured case follows the same logic.

Computing T_N

After simple algebra we get:

$$T_N(x; z_0, z_1) = \sum_{k=0}^{N} z_0^{N-k} z_1^k \frac{(1-t^{-1})^k}{(1-qt^{-1})^k} H_k(t^{-1}) * E_{N-k}(tq^{-1})$$

Summing $T_N(x; z_0, z_1)$ with the generating parameter gives the shuffle exponential:

$$T(v \,|\, x; z) = \exp_* \left(- \sum_{k>0} \frac{v^k}{k} \left(z_1^k + \frac{q^k - t^k}{1 - t^k} z_0^k \right) L_k(x_1 \ldots x_k) \right)$$

The coloured case follows the same logic.

Including supersymmetric Boltzmann weights leads to a model with mixtures of “bosonic” and “fermionic” paths. In this case:

$$T(v \,|\, x; z) = \exp_* \left(- \sum_{k>0} \frac{v^k}{k} \left(-w_1^k - \cdots - w_m^k + z_1^k + \cdots + z_n^k + \frac{q^k - t^k}{1 - t^k} z_0^k \right) L_k(x_1 \ldots x_k) \right)$$

where w_i count fermionic paths.

$T(x; w)$ as a mixed Macdonald Cauchy kernel

$T(x;w)$ as a mixed Macdonald Cauchy kernel

Consider the Boltzmann weights of $U_q(sl_{1|m})$ i.e. all m fermionic paths and set $z_0 = 0$, then:

$$T(x;w) = \exp_* \left(\sum_{k>0} \frac{1}{k} \left(w_1^k + \cdots + w_m^k \right) L_k(x_1 \ldots x_k) \right)$$

$T(x;w)$ as a mixed Macdonald Cauchy kernel

Consider the Boltzmann weights of $U_q(sl_{1|m})$ i.e. all m fermionic paths and set $z_0 = 0$, then:

$$T(x;w) = \exp_* \left(\sum_{k>0} \frac{1}{k} \left(w_1^k + \cdots + w_m^k \right) L_k(x_1 \ldots x_k) \right)$$

This is a mixed Cauchy kernel [Feigin et. al. '10]. Expand $T(x;w)$ in Macdonald polynomials $P_\lambda(w)$:

$$T(x;w) = \sum_\lambda P_\lambda(w) F_\lambda(x)$$

$T(x;w)$ as a mixed Macdonald Cauchy kernel

Consider the Boltzmann weights of $U_q(sl_{1|m})$ i.e. all m fermionic paths and set $z_0 = 0$, then:

$$T(x;w) = \exp_* \left(\sum_{k>0} \frac{1}{k} \left(w_1^k + \cdots + w_m^k \right) L_k(x_1 \ldots x_k) \right)$$

This is a mixed Cauchy kernel [Feigin et. al. '10]. Expand $T(x;w)$ in Macdonald polynomials $P_\lambda(w)$:

$$T(x;w) = \sum_\lambda P_\lambda(w) F_\lambda(x)$$

F_λ is the "Macdonald function" of $\mathscr{A}^\circ$. Using rep theory of $\mathscr{A}$ [Feigin—Tsymbaliuk '09, Schiffmann—Vasserot, '09] we can derive:

$$\mathrm{ev}_{\mu/\nu} \left(F_\lambda \right) \propto f^{\mu}_{\lambda,\nu}$$

where $\mathrm{ev}_{\mu/\nu}(x_i) =$ content of i-th box of diagram of μ/ν.

$T(x;w)$ as a mixed Macdonald Cauchy kernel

Consider the Boltzmann weights of $U_q(sl_{1|m})$ i.e. all m fermionic paths and set $z_0 = 0$, then:

$$T(x;w) = \exp_* \left(\sum_{k>0} \frac{1}{k} \left(w_1^k + \cdots + w_m^k \right) L_k(x_1 \ldots x_k) \right)$$

This is a mixed Cauchy kernel [Feigin et. al. '10]. Expand $T(x;w)$ in Macdonald polynomials $P_\lambda(w)$:

$$T(x;w) = \sum_\lambda P_\lambda(w) F_\lambda(x)$$

F_λ is the "Macdonald function" of $\mathscr{A}^\circ$. Using rep theory of $\mathscr{A}$ [Feigin—Tsymbaliuk '09, Schiffmann—Vasserot, '09] we can derive:

$$\mathrm{ev}_{\mu/\nu} \left(F_\lambda \right) \propto f^{\mu}_{\lambda,\nu}$$

where $\mathrm{ev}_{\mu/\nu}(x_i) =$ content of i-th box of diagram of μ/ν.

Proposition [AG, A Gunna '23]:

$$\mathrm{ev}_{\mu/\nu} \left(T(x;w) \right) = \mathrm{const}\ P_{\mu/\nu}(w)$$

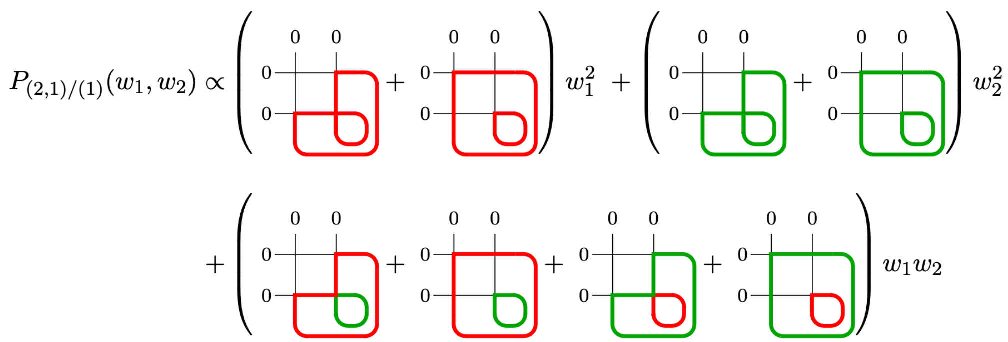
Example

Example

Compute skew Macdonald polynomial of diagram (21)/(1). We have $\mathrm{ev}_{(2,1)/(1)}(x_1, x_2) = (q, t^{-1})$ and

Example

Compute skew Macdonald polynomial of diagram (21)/(1). We have $\mathrm{ev}_{(2,1)/(1)}(x_1, x_2) = (q, t^{-1})$ and



$$\propto w_1^2 + \frac{(1-t)(2+q+t+2qt)}{1-qt^2} w_1 w_2 + w_2^2$$