Around the Razumov-Stroganov correspondence

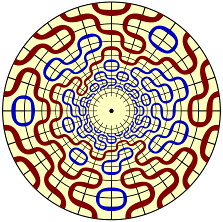
Università degli Studi di Milano

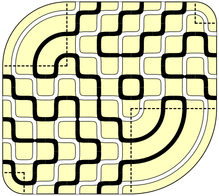
Andrea Sportiello

Séminaire de Combinatoire Enumérative et Analytique
Institut Henri Poincaré – Paris
Jeudi 2 décembre 2010

travail en collaboration avec Luigi Cantini

Université Paris 13 Nord





The Razumov-Stroganov correspondence:
in a few words

Digression on contextual combinatorial objects

Integer and Plane Partitions

Lindström–Gessel-Viennot NILP

The ASM-TSSCPP Theorem

6-Vertex Model and the many faces of ASM

The Laurent Phenomenon

The Razumov-Stroganov correspondence:
a proof

An application: FPL on the three-bundle domain

The Razumov-Stroganov correspondence:
in a few words

Digression on contextual combinatorial objects

Integer and Plane Partitions

Lindström–Gessel-Viennot NILP

The ASM-TSSCPP Theorem

6-Vertex Model and the many faces of ASM

The Laurent Phenomenon

The Razumov-Stroganov correspondence:
a proof

An application: FPL on the three-bundle domain

Three Random Tiling Problems

O(1) Dense Loop Model
XXZ Quantum Spin Chain at $\Delta = -\frac{1}{2}$
Potts Model at edge-percolation
–

Fully-Packed Loops (FPL) in a square
Alternating Sign Matrices (ASM)
Six-Vertex Model at $\Delta = +\frac{1}{2}$ (Ice Model)
"Gog" triangles
–

TSSCPP (Plane Partitions)
Dimer coverings / Lozenge tilings
NILP (Non-intersecting Lattice Paths)
"Magog" triangles

Three Random Tiling Problems

➨ **O(1) Dense Loop Model**

XXZ Quantum Spin Chain at $\Delta = -\frac{1}{2}$

Potts Model at edge-percolation

–

Fully-Packed Loops (FPL) in a square

Alternating Sign Matrices (ASM)

Six-Vertex Model at $\Delta = +\frac{1}{2}$ (Ice Model)

"Gog" triangles

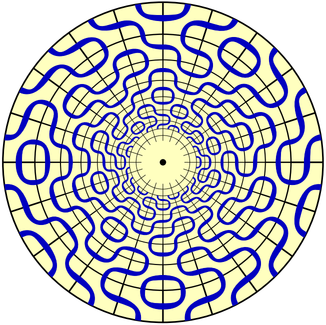
–

TSSCPP (Plane Partitions)

Dimer coverings / Lozenge tilings

NILP (Non-intersecting Lattice Paths)

"Magog" triangles



Three Random Tiling Problems

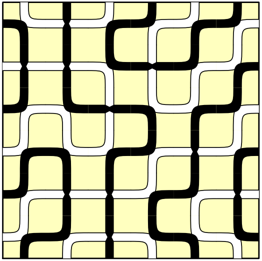
O(1) Dense Loop Model
XXZ Quantum Spin Chain at $\Delta = -\frac{1}{2}$
Potts Model at edge-percolation

–

➡ **Fully-Packed Loops** (FPL) in a square
Alternating Sign Matrices (ASM)
Six-Vertex Model at $\Delta = +\frac{1}{2}$ (Ice Model)
"Gog" triangles

–

TSSCPP (Plane Partitions)
Dimer coverings / Lozenge tilings
NILP (Non-intersecting Lattice Paths)
"Magog" triangles



Three Random Tiling Problems

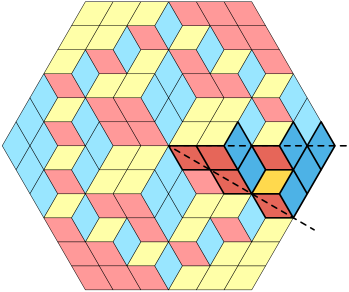
O(1) Dense Loop Model
XXZ Quantum Spin Chain at $\Delta = -\frac{1}{2}$
Potts Model at edge-percolation

–

Fully-Packed Loops (FPL) in a square
Alternating Sign Matrices (ASM)
Six-Vertex Model at $\Delta = +\frac{1}{2}$ (Ice Model)
"Gog" triangles

–

➨ **TSSCPP** (Plane Partitions)
Dimer coverings / Lozenge tilings
NILP (Non-intersecting Lattice Paths)
"Magog" triangles



Three Random Tiling Problems

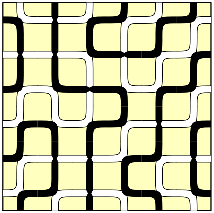
O(1) Dense Loop Model
XXZ Quantum Spin Chain at $\Delta = -\frac{1}{2}$
Potts Model at edge-percolation

–

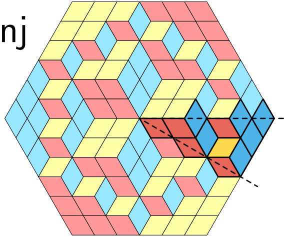
Fully-Packed Loops (FPL) in a square
Alternating Sign Matrices (ASM)
Six-Vertex Model at $\Delta = +\frac{1}{2}$ (Ice Model)
"Gog" triangles

–

TSSCPP (Plane Partitions)
Dimer coverings / Lozenge tilings
NILP (Non-intersecting Lattice Paths)
"Magog" triangles



ASM-conj

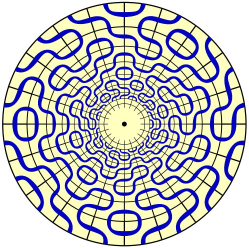


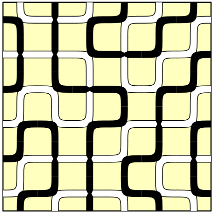
Three Random Tiling Problems

O(1) Dense Loop Model
XXZ Quantum Spin Chain at $\Delta = -\frac{1}{2}$
Potts Model at edge-percolation
–

Fully-Packed Loops (FPL) in a square
Alternating Sign Matrices (ASM)
Six-Vertex Model at $\Delta = +\frac{1}{2}$ (Ice Model)
“Gog” triangles
–

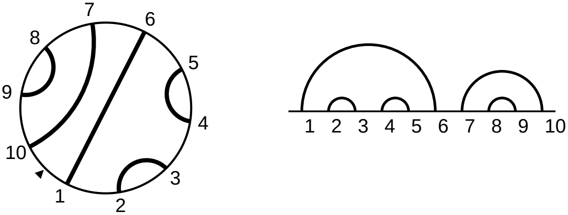
TSSCPP (Plane Partitions)
Dimer coverings / Lozenge tilings
NILP (Non-intersecting Lattice Paths)
“Magog” triangles





Link patterns

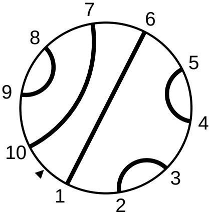
A link pattern $\pi \in \mathcal{LP}(n)$ is a pairing of $\{1, 2, \ldots, 2n\}$ having no pairs (a, c), (b, d) such that $a < b < c < d$ (i.e., the drawing consists of n non-crossing arcs).

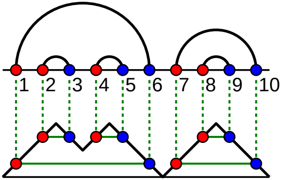


They are $C_n = \frac{1}{n+1}\binom{2n}{n}$ (the n-th *Catalan number*),

Link patterns

A link pattern $\pi \in \mathcal{LP}(n)$ is a pairing of $\{1, 2, \ldots, 2n\}$ having no pairs (a, c), (b, d) such that $a < b < c < d$ (i.e., the drawing consists of n non-crossing arcs).

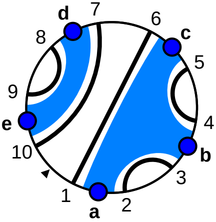


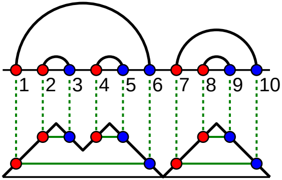


They are $C_n = \frac{1}{n+1}\binom{2n}{n}$ (the n-th *Catalan number*), are in easy bijection with Dyck Paths of length $2n$

Link patterns

A link pattern $\pi \in \mathcal{LP}(n)$ is a pairing of $\{1, 2, \dots, 2n\}$ having no pairs (a, c), (b, d) such that $a < b < c < d$ (i.e., the drawing consists of n non-crossing arcs).



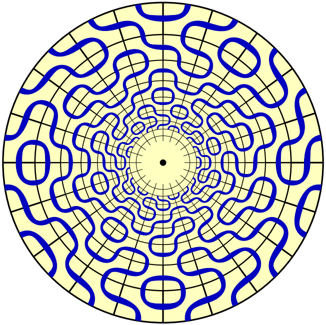


They are $C_n = \frac{1}{n+1}\binom{2n}{n}$ (the n-th *Catalan number*), are in easy bijection with Dyck Paths of length $2n$ and with non-crossing partitions of n elements.

...and many other things...

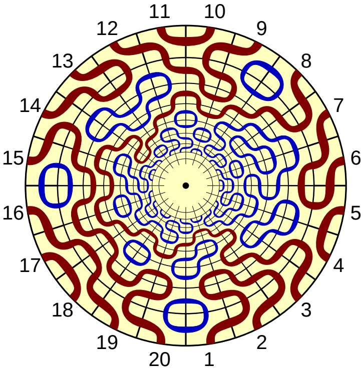
Link patterns in the Dense Loop Model

To a dense-loop configuration on a semi-infinite cylinder,
a link pattern π is naturally associated,
as the connectivity pattern for the points on the boundary.



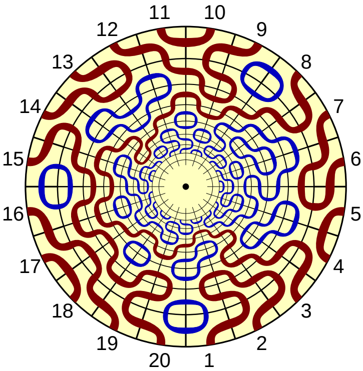
Link patterns in the Dense Loop Model

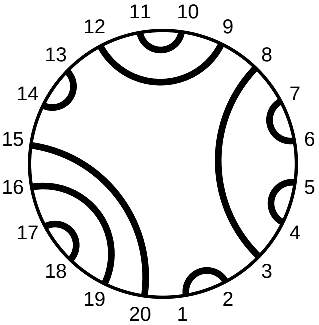
To a dense-loop configuration on a semi-infinite cylinder,
a link pattern π is naturally associated,
as the connectivity pattern for the points on the boundary.



Link patterns in the Dense Loop Model

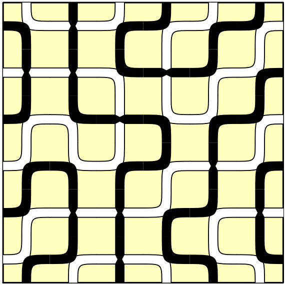
To a dense-loop configuration on a semi-infinite cylinder, a link pattern π is naturally associated, as the connectivity pattern for the points on the boundary.





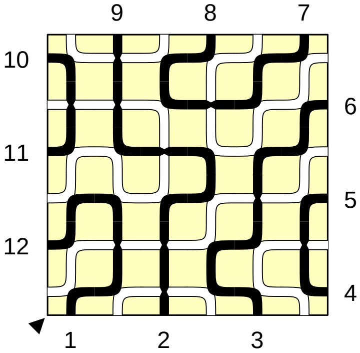
Link patterns in Fully-Packed Loops

To a Fully-Packed Loop configuration,
a link pattern π is naturally associated,
from connectivities among the black terminations on the boundary.



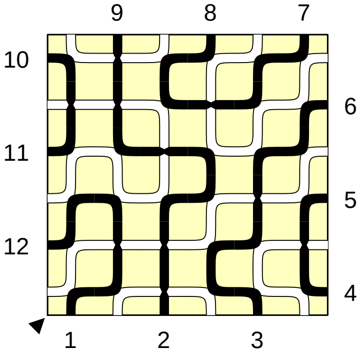
Link patterns in Fully-Packed Loops

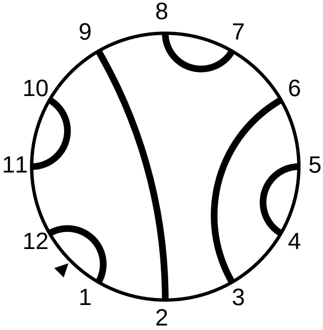
To a Fully-Packed Loop configuration,
a link pattern π is naturally associated,
from connectivities among the black terminations on the boundary.



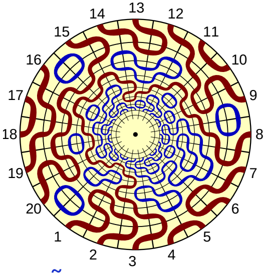
Link patterns in Fully-Packed Loops

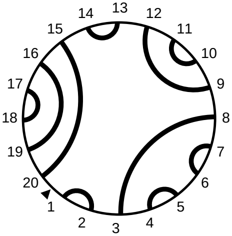
To a Fully-Packed Loop configuration,
a link pattern π is naturally associated,
from connectivities among the black terminations on the boundary.

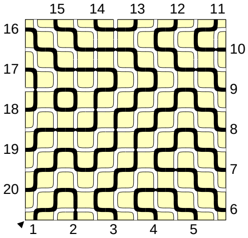




The Razumov-Stroganov correspondence



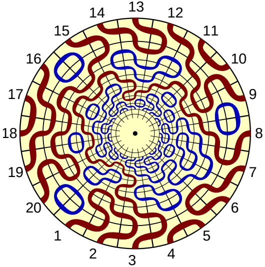


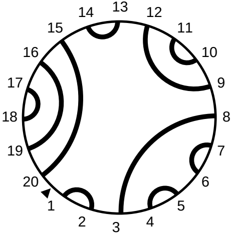


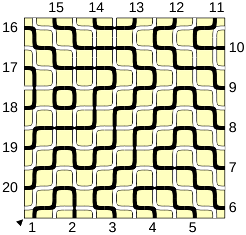
$\tilde{\Psi}_n(\pi)$: probability of π
in the $O(1)$ Dense Loop Model
in the $\{1, \dots, 2n\} \times \mathbb{N}$ cylinder

$\Psi_n(\pi)$: probability of π
for FPL with uniform measure
in the $n \times n$ square

The Razumov-Stroganov correspondence







$\tilde{\Psi}_n(\pi)$: probability of π
in the $O(1)$ Dense Loop Model
in the $\{1, \dots, 2n\} \times \mathbb{N}$ cylinder

$\Psi_n(\pi)$: probability of π
for FPL with uniform measure
in the $n \times n$ square

Razumov-Stroganov correspondence
(conjecture: Razumov Stroganov, 2001; proof: AS Cantini, 2010)

$$\tilde{\Psi}_n(\pi) = \Psi_n(\pi)$$

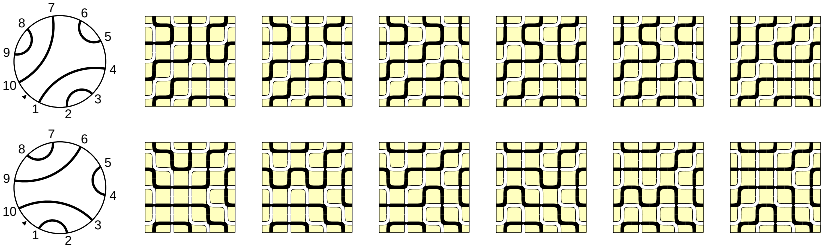
Dihedral symmetry of FPL

A corollary of the Razumov-Stroganov correspondence...
(...that was known *before* the Razumov-Stroganov conjecture)

call R the operator that rotates a link pattern by one position

Dihedral symmetry of FPL (proof: Wieland, 2000)

$$\Psi_n(\pi) = \Psi_n(R\pi)$$



A scheme of the proof

(O(1) DLM) $\tilde{\Psi} \overset{?}{=} \Psi$ *(FPL)*

Temperley-Lieb Algebra in the $O(1)$ Dense Loop Model.
Use of **Yang-Baxter** $\implies$ $H\Psi = \sum_j (e_j - 1)\Psi = \vec{0}$.

Produce a **generalized Wieland gyration** for refined domains

$$H\Psi = (1 + R + R^2 + \cdots + R^{2n-1})e_j\Psi.$$

Break $\Psi = \Psi^{(a,j)} + \Psi^{(c,j)}$.

A **recursion** using gyration gives

$$e_j\Psi^{(a,j)} = Re_{j-1}\Psi^{(a,j-1)} + (\cdots)\Psi^{(c,j)}.$$

What remains is $\sum_j (1 + R + R^2 + \cdots + R^{2n-1})(e_j - 1)\Psi^{(c,j)}$.
Summands are separately zero, from a **lemma on gyration orbits**.

□

The Razumov-Stroganov correspondence: in a few words

Digression on contextual combinatorial objects

Integer and Plane Partitions

Lindström–Gessel-Viennot NILP

The ASM-TSSCPP Theorem

6-Vertex Model and the many faces of ASM

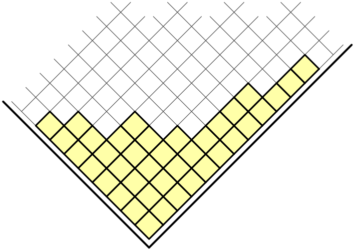
The Laurent Phenomenon

The Razumov-Stroganov correspondence: a proof

An application: FPL on the three-bundle domain

Integer Partitions and Plane Partitions

Take a 2D quadrant $\mathbb{N}^2$,
Pile squares (subject to “gravity” along the $(1,1)$ axis).
That is, produce subsets $\pi \subset \mathbb{N}^2$ such that, if $(x,y) \in \pi$, then $\{(x',y')\}_{\substack{1 \leq x' \leq x \\ 1 \leq y' \leq y}} \subseteq \pi$
Call $|\pi|$ the number of squares in π

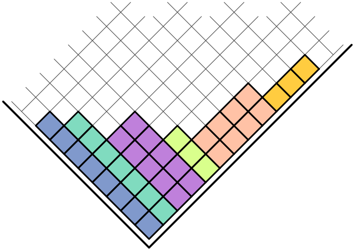


Related to partitions of an integer:
$|\pi| = a_1 + a_2 + \ldots + a_k$
with $a_1 \geq a_2 \geq \ldots \geq a_k$,
and thus with a long history
(Euler, Sylvester, Frobenius, Hardy-Ramanujan,...)

Generating function: $$\sum_{\pi} q^{|\pi|} = \prod_{j \geq 1} \frac{1}{1-q^j}$$

Integer Partitions and Plane Partitions

Take a 2D quadrant $\mathbb{N}^2$,
Pile squares (subject to “gravity” along the $(1,1)$ axis).
That is, produce subsets $\pi \subset \mathbb{N}^2$ such that, if $(x,y) \in \pi$, then $\{(x',y')\}_{\substack{1 \le x' \le x \\ 1 \le y' \le y}} \subseteq \pi$
Call $|\pi|$ the number of squares in π



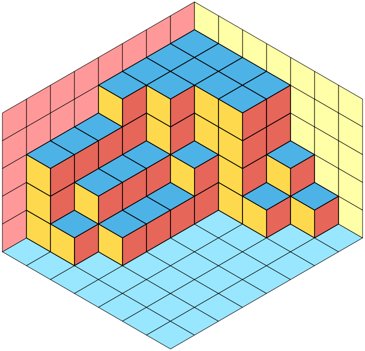
Related to partitions of an integer:
$|\pi| = a_1 + a_2 + \ldots + a_k$
with $a_1 \ge a_2 \ge \ldots \ge a_k$,
and thus with a long history
(Euler, Sylvester, Frobenius, Hardy-Ramanujan,...)

Generating function: $$\sum_\pi q^{|\pi|} = \prod_{j \ge 1} \frac{1}{1-q^j}$$

Unrestricted Plane Partitions

Take the 3D octant $\mathbb{N}^3$.
Pile cubes (subject to "gravity" along the $(1,1,1)$ axis).
That is, produce subsets $\pi \subset \mathbb{N}^3$ such that, if $(x,y,z) \in \pi$, then $\{(x',y',z')\}_{\substack{1\leq x'\leq x\\ 1\leq y'\leq y\\ 1\leq z'\leq z}} \subseteq \pi$

Call $|\pi|$ the number of cubes in π



Generating fn.: (MacMahon, 1912)

$$\sum_{\pi} q^{|\pi|} = \prod_{j\geq 1} \frac{1}{(1-q^j)^j}$$

Meaningful for $q \in \mathbb{C}$, $|q| < 1$

Unrestricted Plane Partitions

Take the 3D octant $\mathbb{N}^3$.

Pile cubes (subject to “gravity” along the $(1, 1, 1)$ axis).

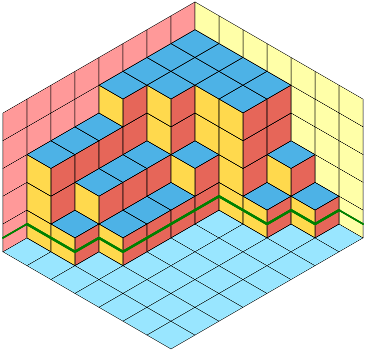
That is, produce subsets $\pi \subset \mathbb{N}^3$ such that, if $(x, y, z) \in \pi$, then $\{(x', y', z')\}_{\substack{1 \leq x' \leq x \\ 1 \leq y' \leq y \\ 1 \leq z' \leq z}} \subseteq \pi$

Call $|\pi|$ the number of cubes in π

Generating fn.: (MacMahon, 1912)

$$\sum_{\pi} q^{|\pi|} = \prod_{j \geq 1} \frac{1}{(1 - q^j)^j}$$

Meaningful for $q \in \mathbb{C}$, $|q| < 1$



Can be sliced into a string of integer partitions, ordered w.r.t. inclusion

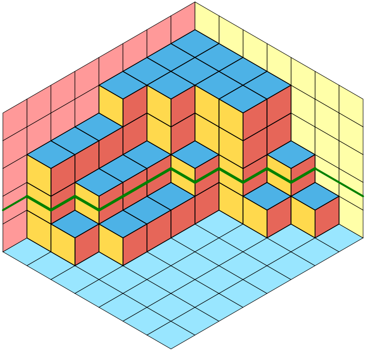
Unrestricted Plane Partitions

Take the 3D octant $\mathbb{N}^3$.

Pile cubes (subject to "gravity" along the $(1, 1, 1)$ axis).

That is, produce subsets $\pi \subset \mathbb{N}^3$ such that, if $(x, y, z) \in \pi$, then $\{(x', y', z')\}_{\substack{1 \leq x' \leq x \\ 1 \leq y' \leq y \\ 1 \leq z' \leq z}} \subseteq \pi$

Call $|\pi|$ the number of cubes in π



Generating fn.: (MacMahon, 1912)

$$\sum_{\pi} q^{|\pi|} = \prod_{j \geq 1} \frac{1}{(1 - q^j)^j}$$

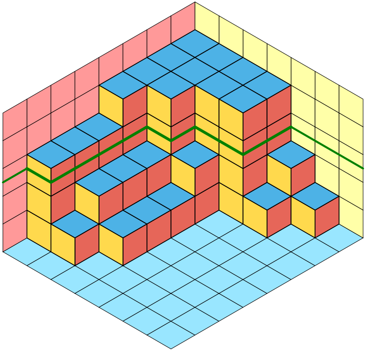
Meaningful for $q \in \mathbb{C}$, $|q| < 1$

Can be sliced into a string of integer partitions, ordered w.r.t. inclusion

Unrestricted Plane Partitions

Take the 3D octant $\mathbb{N}^3$.
Pile cubes (subject to "gravity" along the $(1,1,1)$ axis).
That is, produce subsets $\pi \subset \mathbb{N}^3$ such that, if $(x,y,z) \in \pi$, then $\{(x',y',z')\}_{\substack{1\leq x'\leq x\\ 1\leq y'\leq y\\ 1\leq z'\leq z}} \subseteq \pi$

Call $|\pi|$ the number of cubes in π



Generating fn.: (MacMahon, 1912)

$$\sum_{\pi} q^{|\pi|} = \prod_{j\geq 1} \frac{1}{(1-q^j)^j}$$

Meaningful for $q \in \mathbb{C}$, $|q| < 1$

Can be sliced into a string of integer partitions, ordered w.r.t. inclusion

Unrestricted Plane Partitions

Take the 3D octant $\mathbb{N}^3$.

Pile cubes (subject to "gravity" along the $(1,1,1)$ axis).

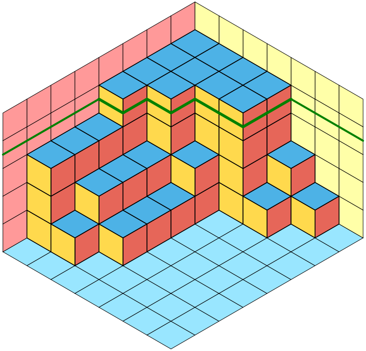
That is, produce subsets $\pi \subset \mathbb{N}^3$ such that, if $(x,y,z) \in \pi$, then $\{(x',y',z')\}_{\substack{1 \leq x' \leq x \\ 1 \leq y' \leq y \\ 1 \leq z' \leq z}} \subseteq \pi$

Call $|\pi|$ the number of cubes in π

Generating fn.: (MacMahon, 1912)

$$\sum_{\pi} q^{|\pi|} = \prod_{j \geq 1} \frac{1}{(1-q^j)^j}$$

Meaningful for $q \in \mathbb{C}$, $|q| < 1$



Can be sliced into a string of integer partitions, ordered w.r.t. inclusion

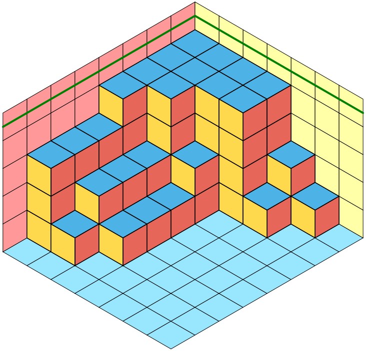
Unrestricted Plane Partitions

Take the 3D octant $\mathbb{N}^3$.

Pile cubes (subject to "gravity" along the $(1, 1, 1)$ axis).

That is, produce subsets $\pi \subset \mathbb{N}^3$ such that, if $(x, y, z) \in \pi$, then $\{(x', y', z')\}_{\substack{1 \leq x' \leq x \\ 1 \leq y' \leq y \\ 1 \leq z' \leq z}} \subseteq \pi$

Call $|\pi|$ the number of cubes in π



Generating fn.: (MacMahon, 1912)

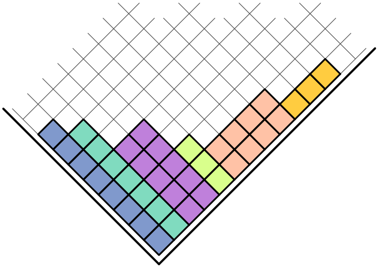
$$\sum_{\pi} q^{|\pi|} = \prod_{j \geq 1} \frac{1}{(1 - q^j)^j}$$

Meaningful for $q \in \mathbb{C}$, $|q| < 1$

Can be sliced into a string of integer partitions, ordered w.r.t. inclusion

On factorization of Unrestricted Plane Partitions

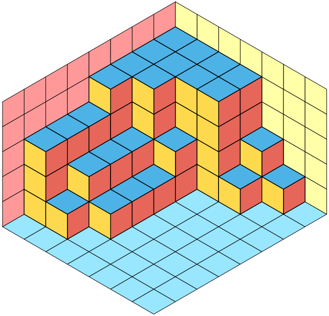
Unrestricted Integer Partitions



Generating function:

$$\sum_{\pi} q^{|\pi|} = \prod_{j \geq 1} \frac{1}{1 - q^j}$$

Unrestricted Plane Partitions



Generating function:

$$\sum_{\pi} q^{|\pi|} = \prod_{j \geq 1} \frac{1}{(1 - q^j)^j}$$

On factorization of Unrestricted Plane Partitions

$$\text{General form: } \sum_{\pi} q^{|\pi|} = \prod_{j\geq 1} \frac{1}{(1-q^j)^{b(j)}}$$

Is there a hidden structure of
graded Unique Factorization Domain? (combinatorial prefab)

I.e., do we have "prime" objects $\{p_{j,\alpha}\}_{j\geq 1\,;\,1\leq\alpha\leq b(j)}$ and

$$a = \prod_{\substack{j\geq 1\\ 1\leq\alpha\leq b(j)}} p_{j,\alpha}^{\nu(j,\alpha)} \qquad \text{(w.r.t. some notion of "product"?)}$$

- ✍ Bender and Knuth, *Enumeration of Plane Partitions*, 1972
- ✍ I. Pak, *Hook length formula and geometric combinatorics*, 2001

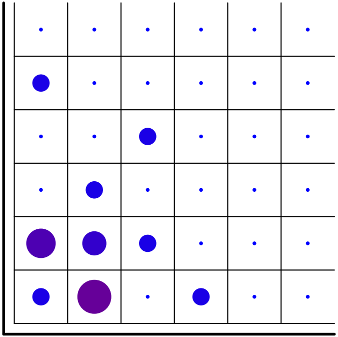
The Pak Algorithm

	N	
W	X	E
	S	

operation **A**: $X \rightarrow X + \max(N, E)$;

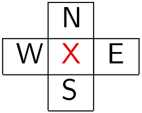
operation **B**: $X \rightarrow -X + \max(N, E) + \min(S, W)$;

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



1. the input is your $\boldsymbol{\nu} = \{\nu(x, y)\}$.

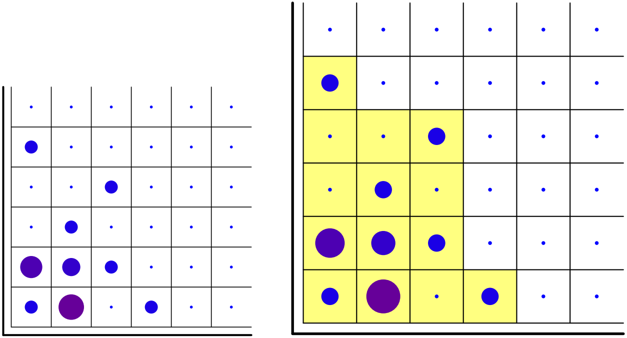
The Pak Algorithm



operation **A**: $X \to X + \max(N, E)$;

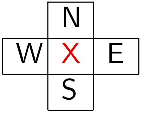
operation **B**: $X \to -X + \max(N, E) + \min(S, W)$;

$\mathbf{C}(x, y)$: apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



2. take $S \subset \mathbb{N}^2$, convex and containing all positive ν's.

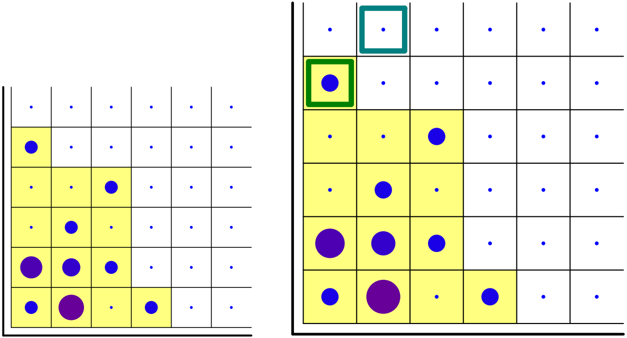
The Pak Algorithm



operation **A**: $X \to X + \max(N, E)$;

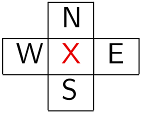
operation **B**: $X \to -X + \max(N, E) + \min(S, W)$;

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



3. apply **C** to all $(x, y) \in S$, larger first, w.r.t. partial ordering.

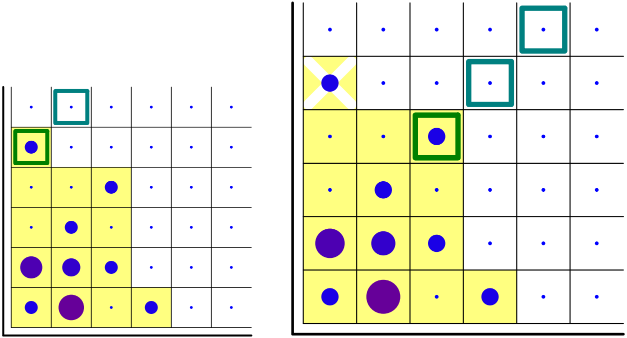
The Pak Algorithm



operation **A**: $X \to X + \max(N, E)$;

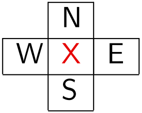
operation **B**: $X \to -X + \max(N, E) + \min(S, W)$;

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



3. apply **C** to all $(x, y) \in S$, larger first, w.r.t. partial ordering.

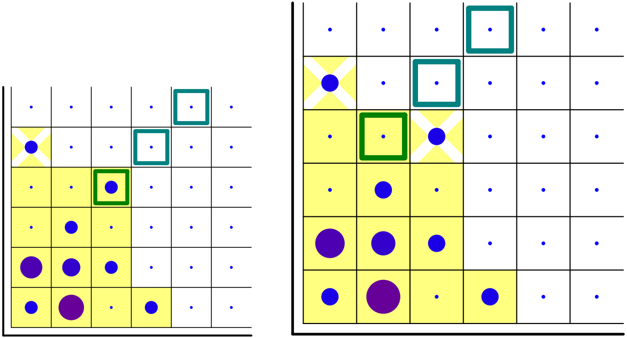
The Pak Algorithm



operation **A**: $\quad X \to X + \max(N, E);$

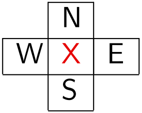
operation **B**: $\quad X \to -X + \max(N, E) + \min(S, W);$

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



3. apply **C** to all $(x, y) \in S$, larger first, w.r.t. partial ordering.

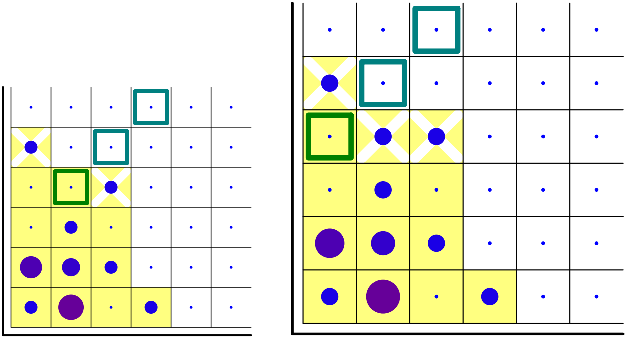
The Pak Algorithm



operation **A**: $X \to X + \max(N, E)$;

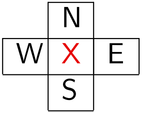
operation **B**: $X \to -X + \max(N, E) + \min(S, W)$;

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



3. apply **C** to all $(x, y) \in S$, larger first, w.r.t. partial ordering.

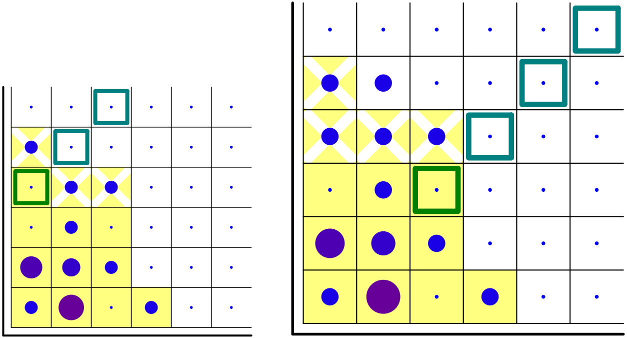
The Pak Algorithm



operation **A**: $X \to X + \max(N, E)$;

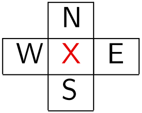
operation **B**: $X \to -X + \max(N, E) + \min(S, W)$;

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



3. apply **C** to all $(x, y) \in S$, larger first, w.r.t. partial ordering.

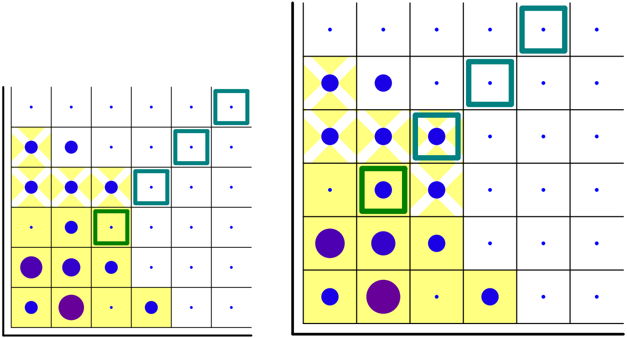
The Pak Algorithm



operation **A**: $X \to X + \max(N, E)$;

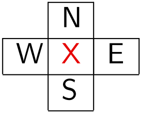
operation **B**: $X \to -X + \max(N, E) + \min(S, W)$;

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



3. apply **C** to all $(x, y) \in S$, larger first, w.r.t. partial ordering.

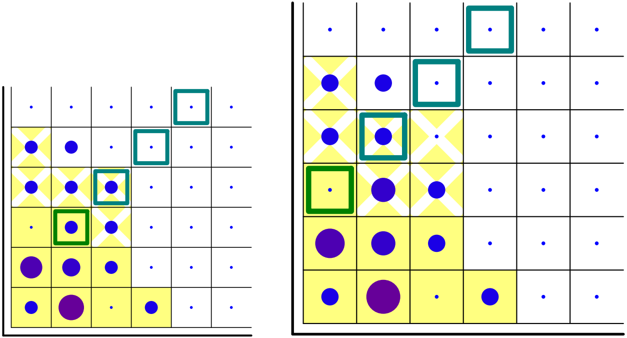
The Pak Algorithm



operation **A**: $X \to X + \max(N, E)$;

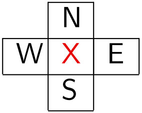
operation **B**: $X \to -X + \max(N, E) + \min(S, W)$;

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



3. apply **C** to all $(x, y) \in S$, larger first, w.r.t. partial ordering.

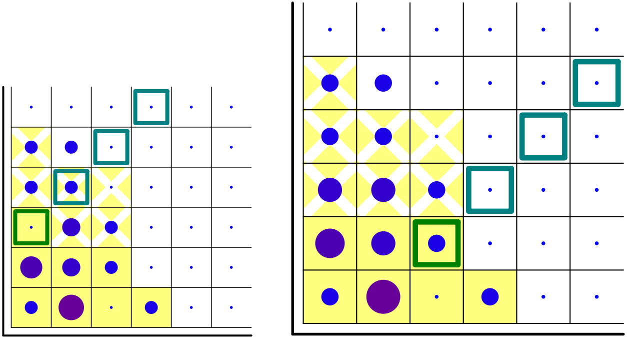
The Pak Algorithm



operation **A**: $X \rightarrow X + \max(N, E)$;

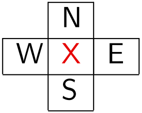
operation **B**: $X \rightarrow -X + \max(N, E) + \min(S, W)$;

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



3. apply **C** to all $(x, y) \in S$, larger first, w.r.t. partial ordering.

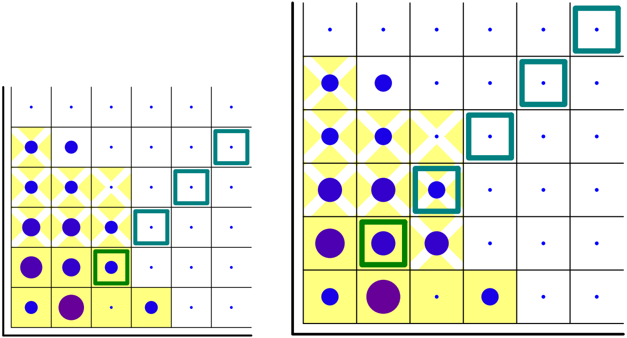
The Pak Algorithm



operation **A**: $X \rightarrow X + \max(N, E);$

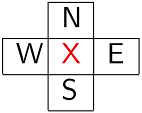
operation **B**: $X \rightarrow -X + \max(N, E) + \min(S, W);$

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



3. apply **C** to all $(x, y) \in S$, larger first, w.r.t. partial ordering.

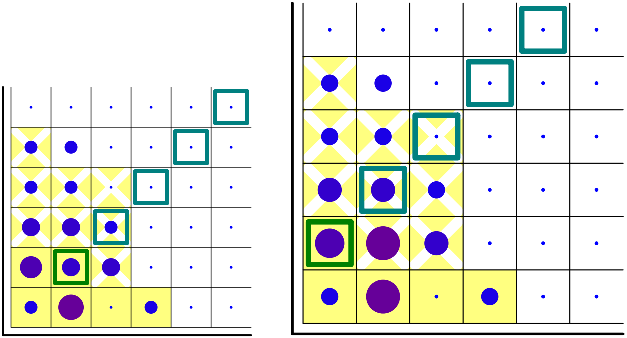
The Pak Algorithm



operation **A**: $X \to X + \max(N, E)$;

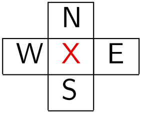
operation **B**: $X \to -X + \max(N, E) + \min(S, W)$;

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



3. apply **C** to all $(x, y) \in S$, larger first, w.r.t. partial ordering.

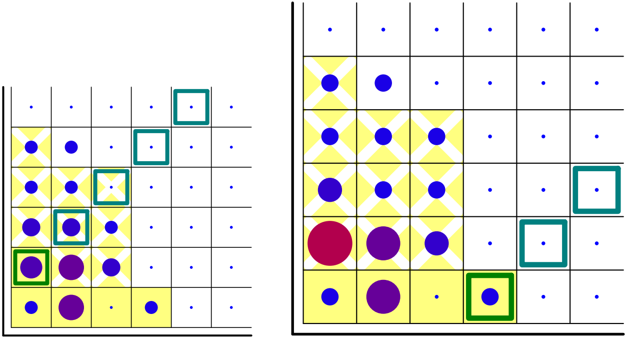
The Pak Algorithm



operation **A**: $X \to X + \max(N, E)$;

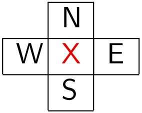
operation **B**: $X \to -X + \max(N, E) + \min(S, W)$;

$\mathbf{C}(x, y)$: apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



3. apply **C** to all $(x, y) \in S$, larger first, w.r.t. partial ordering.

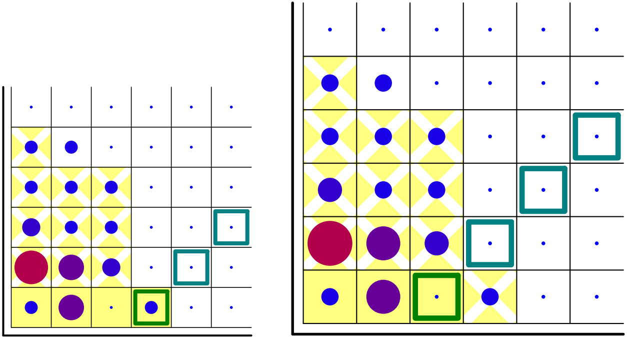
The Pak Algorithm



operation **A**: $X \rightarrow X + \max(N, E)$;

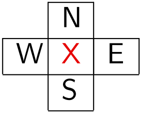
operation **B**: $X \rightarrow -X + \max(N, E) + \min(S, W)$;

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



3. apply **C** to all $(x, y) \in S$, larger first, w.r.t. partial ordering.

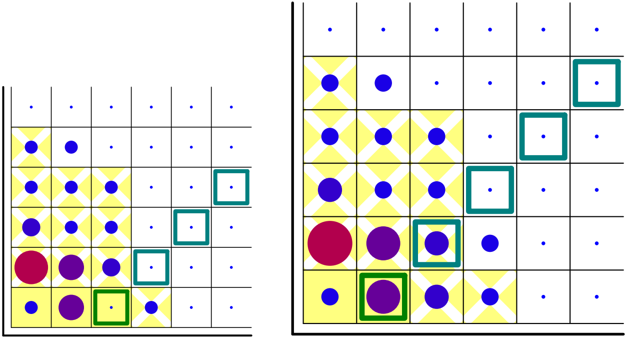
The Pak Algorithm



operation **A**: $X \to X + \max(N, E)$;

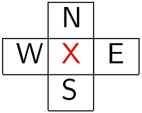
operation **B**: $X \to -X + \max(N, E) + \min(S, W)$;

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



3. apply **C** to all $(x, y) \in S$, larger first, w.r.t. partial ordering.

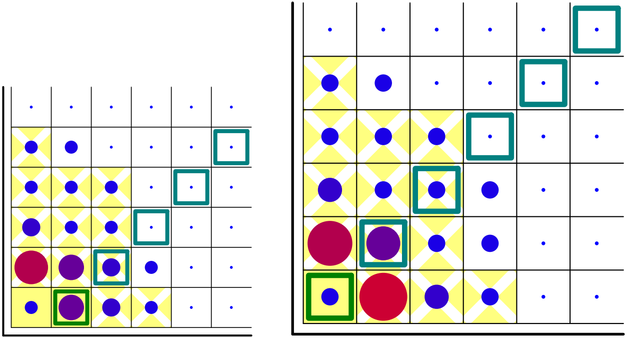
The Pak Algorithm



operation **A**: $X \to X + \max(N, E)$;

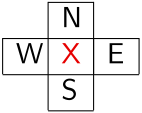
operation **B**: $X \to -X + \max(N, E) + \min(S, W)$;

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



3. apply **C** to all $(x, y) \in S$, larger first, w.r.t. partial ordering.

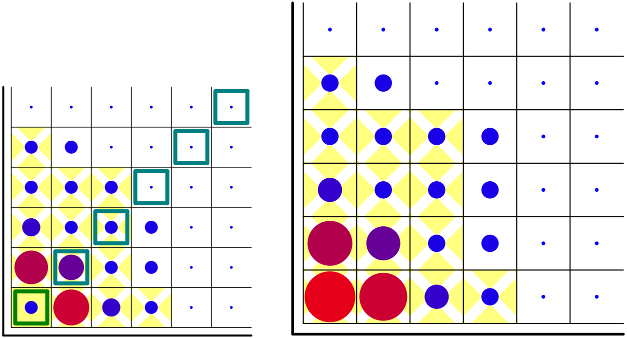
The Pak Algorithm



operation **A**: $X \to X + \max(N, E)$;

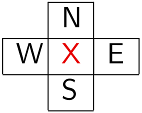
operation **B**: $X \to -X + \max(N, E) + \min(S, W)$;

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



3. apply **C** to all $(x, y) \in S$, larger first, w.r.t. partial ordering.

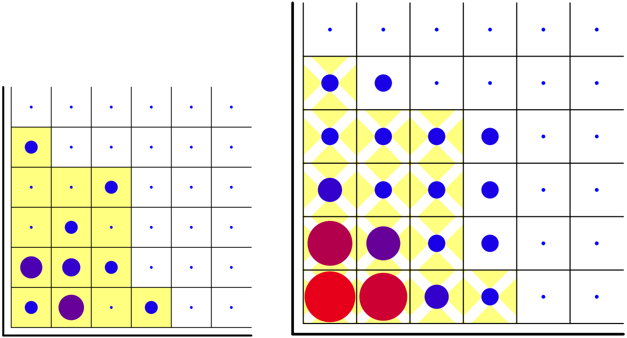
The Pak Algorithm



operation **A**: $X \to X + \max(N, E)$;

operation **B**: $X \to -X + \max(N, E) + \min(S, W)$;

C(x, y): apply **A** at (x, y), and **B** at $(x + z, y + z)_{z \geq 1}$



4. the result is your $\mathbf{h} = \{h(x, y)\}$.

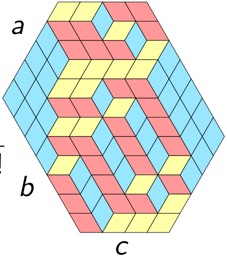
Plane Partitions in a box

In a compact box, can push q to the "combinatorial point" $q = 1$

No symmetry:

P.A. MacMahon (1915)

$$M_{a,b,c} = \prod_{\substack{0\leq i<a \\ 0\leq j<b \\ 0\leq k<c}} \frac{i+j+k+2}{i+j+k+1} = \prod_{0\leq j<c} \frac{j!(j+a+b)!}{(j+a)!(j+b)!}$$

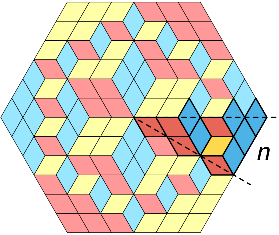


...various symmetry classes...

Maximally symmetric (TSSCPP):

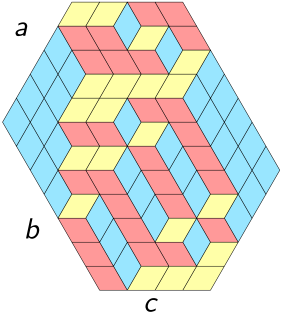
G. Andrews (1994)

$$A_n = \prod_{0\leq j<n} \frac{(3j+1)!}{(n+j)!} = \prod_{0\leq j<n} \frac{j!\,(3j+1)!}{(2j)!(2j+1)!}$$



Boxed Plane Partitions as Non-Intersecting Lattice Paths

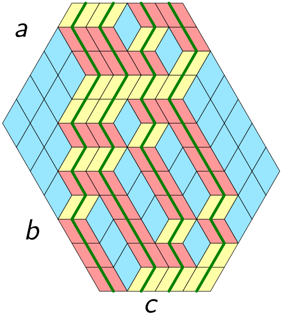
$a \times b \times c$ boxed Plane Partition are counted by a determinant:



I. Gessel and G. Viennot, *Binomial determinants, paths, and hook length formulae*, 1985

Boxed Plane Partitions as Non-Intersecting Lattice Paths

$a \times b \times c$ boxed Plane Partition are counted by a determinant:

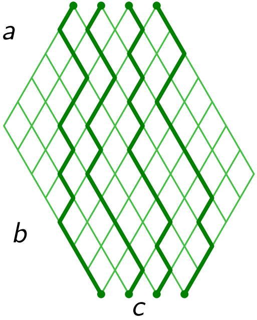


We have c directed paths on the square lattice, connecting top and bottom sides, which do not intersect (NILP)

I. Gessel and G. Viennot, *Binomial determinants, paths, and hook length formulae*, 1985

Boxed Plane Partitions as Non-Intersecting Lattice Paths

$a \times b \times c$ boxed Plane Partition are counted by a determinant:

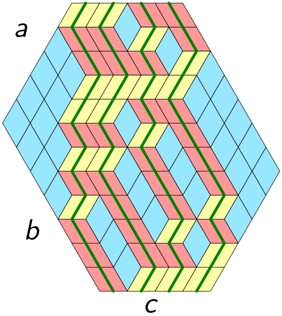


We have c directed paths on the square lattice, connecting top and bottom sides, which do not intersect (NILP)

I. Gessel and G. Viennot, *Binomial determinants, paths, and hook length formulae,* 1985

Boxed Plane Partitions as Non-Intersecting Lattice Paths

$a \times b \times c$ boxed Plane Partition are counted by a determinant:

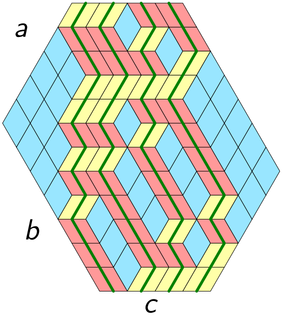


If it weren't for the non-intersecting constraint, the number of path configs would just be $\binom{a+b}{a}^c$, that is....

I. Gessel and G. Viennot, *Binomial determinants, paths, and hook length formulae,* 1985

Boxed Plane Partitions as Non-Intersecting Lattice Paths

$a \times b \times c$ boxed Plane Partition are counted by a determinant:



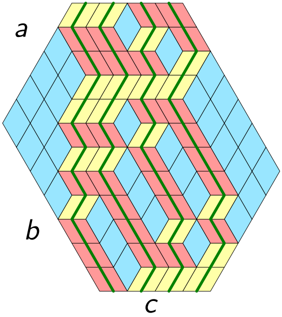
If it weren't for the non-intersecting constraint, the number of path configs would just be $\binom{a+b}{a}^c$, that is....

$$\det \begin{pmatrix} \binom{a+b}{a} & 0 & 0 & 0 \\ 0 & \binom{a+b}{a} & 0 & 0 \\ 0 & 0 & \binom{a+b}{a} & 0 \\ 0 & 0 & 0 & \binom{a+b}{a} \end{pmatrix}$$

I. Gessel and G. Viennot, *Binomial determinants, paths, and hook length formulae,* 1985

Boxed Plane Partitions as Non-Intersecting Lattice Paths

$a \times b \times c$ boxed Plane Partition are counted by a determinant:

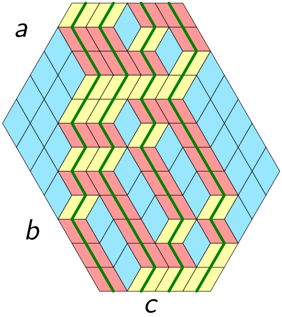


The non-intersecting constraint, through a magic cancellation coming from configs with “the wrong pairing”, leads to the formula...

I. Gessel and G. Viennot, *Binomial determinants, paths, and hook length formulae*, 1985

Boxed Plane Partitions as Non-Intersecting Lattice Paths

$a \times b \times c$ boxed Plane Partition are counted by a determinant:

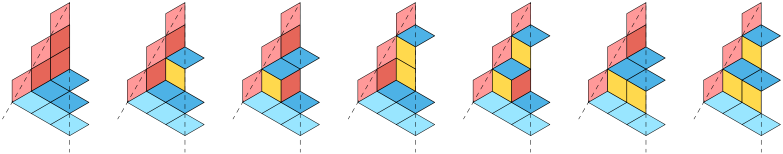


The non-intersecting constraint, through a magic cancellation coming from configs with "the wrong pairing", leads to the formula...

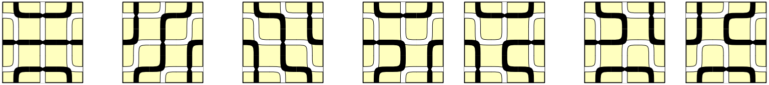
$$\det \begin{pmatrix} \binom{a+b}{a} & \binom{a+b}{a+1} & \binom{a+b}{a+2} & \binom{a+b}{a+3} \\ \binom{a+b}{a-1} & \binom{a+b}{a} & \binom{a+b}{a+1} & \binom{a+b}{a+2} \\ \binom{a+b}{a-2} & \binom{a+b}{a-1} & \binom{a+b}{a} & \binom{a+b}{a+1} \\ \binom{a+b}{a-3} & \binom{a+b}{a-2} & \binom{a+b}{a-1} & \binom{a+b}{a} \end{pmatrix}$$

I. Gessel and G. Viennot, *Binomial determinants, paths, and hook length formulae,* 1985

Plane Partitions and Fully-Packed Loops

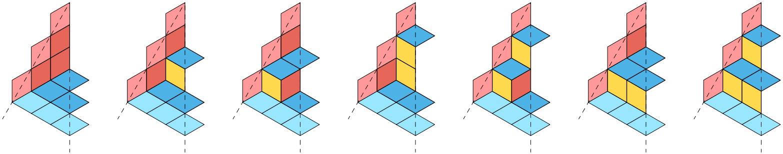


# TSSCPP in a hexagon of side $2n$ = # FPL in a square of side n

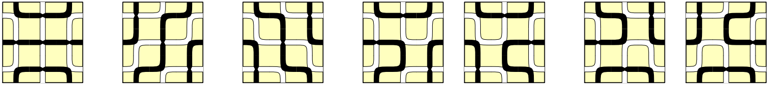


(Proof: Zeilberger 1996, with generating functions and much more; Kuperberg 1996, specializing results from the Six-vertex model)

Plane Partitions and Fully-Packed Loops



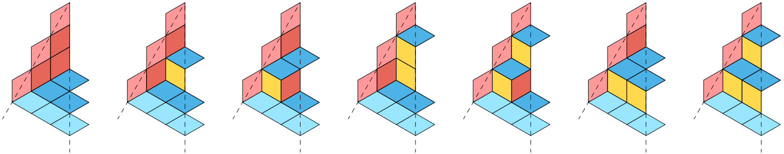
# TSSCPP in a hexagon of side $2n$ = # FPL in a square of side n



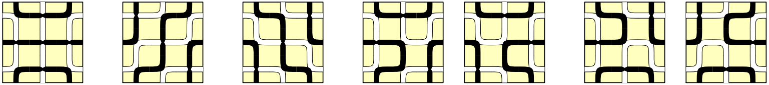
(Proof: Zeilberger 1996, with generating functions and much more; Kuperberg 1996, specializing results from the Six-vertex model)

We have no bijectional clue of why this is true

Plane Partitions and Fully-Packed Loops



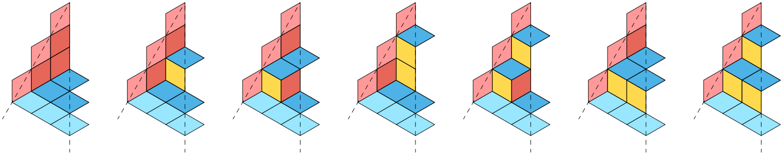
# TSSCPP in a hexagon of side $2n$ $=$ # FPL in a square of side n



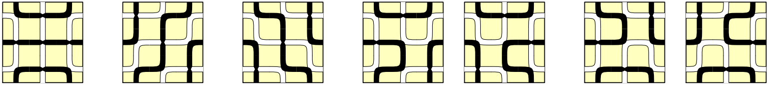
(Proof: Zeilberger 1996, with generating functions and much more; Kuperberg 1996, specializing results from the Six-vertex model)

We have no bijectional clue of why this is true
We have no TSSCPP candidate for link pattern classes

Plane Partitions and Fully-Packed Loops



$\#$ TSSCPP in a hexagon of side $2n = \#$ FPL in a square of side n



(Proof: Zeilberger 1996, with generating functions and much more; Kuperberg 1996, specializing results from the Six-vertex model)

We have no bijectional clue of why this is true
We have no TSSCPP candidate for link pattern classes
But a natural τ-enumeration for TSSCPP
is also natural for the $O(1)$ Dense Loop Model

Arrows on lines : The 6-Vertex Model

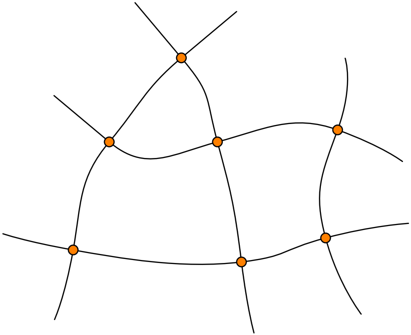
- you have a degree-4 graph $\mathcal{G}$,
- variables are edge-orientations,
- weights are on the vertices,

it is Yang-Baxter–integrable
if weights depend on positions
through spectral parameters
attached to the lines,
and a global parameter q

Arrows on lines : The 6-Vertex Model

- you have a degree-4 graph $\mathcal{G}$,
- variables are edge-orientations,
- weights are on the vertices,

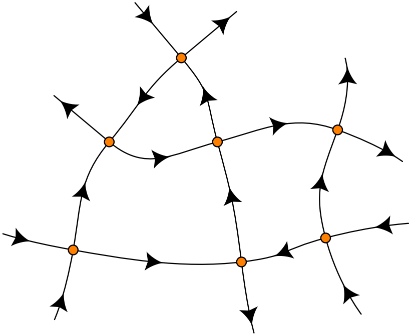
it is Yang-Baxter–integrable
if weights depend on positions
through spectral parameters
attached to the lines,
and a global parameter q



Arrows on lines : The 6-Vertex Model

- you have a degree-4 graph $\mathcal{G}$,
- variables are edge-orientations,
- weights are on the vertices,

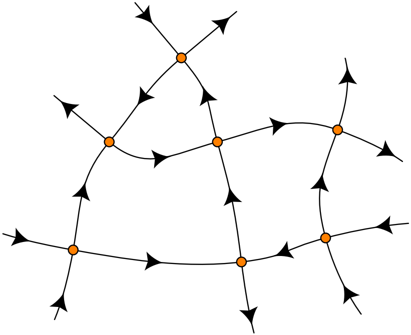
it is Yang-Baxter–integrable
if weights depend on positions
through spectral parameters
attached to the lines,
and a global parameter q



Arrows on lines : The 6-Vertex Model

- you have a degree-4 graph $\mathcal{G}$,
- variables are edge-orientations,
- weights are on the vertices,

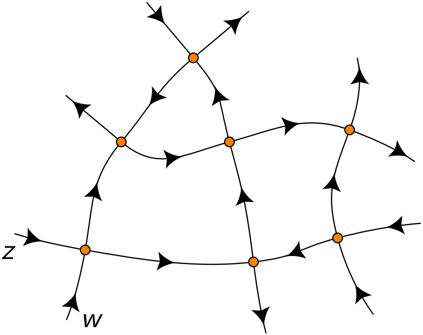
it is Yang-Baxter–integrable
if weights depend on positions
through spectral parameters
attached to the lines,
and a global parameter q



Arrows on lines : The 6-Vertex Model

- you have a degree-4 graph $\mathcal{G}$,
- variables are edge-orientations,
- weights are on the vertices,

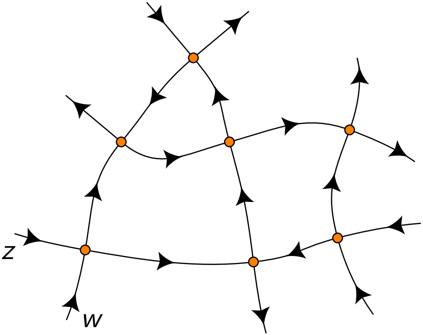
it is Yang-Baxter–integrable
if weights depend on positions
through spectral parameters
attached to the lines,
and a global parameter q

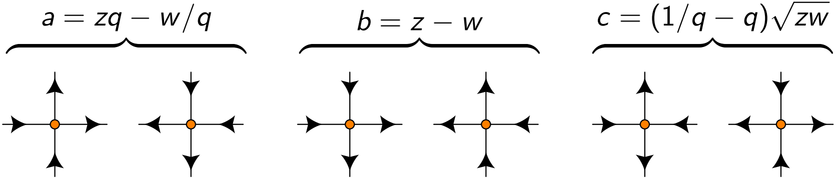


Arrows on lines : The 6-Vertex Model

- you have a degree-4 graph $\mathcal{G}$,
- variables are edge-orientations,
- weights are on the vertices,

it is Yang-Baxter–integrable
if weights depend on positions
through spectral parameters
attached to the lines,
and a global parameter q



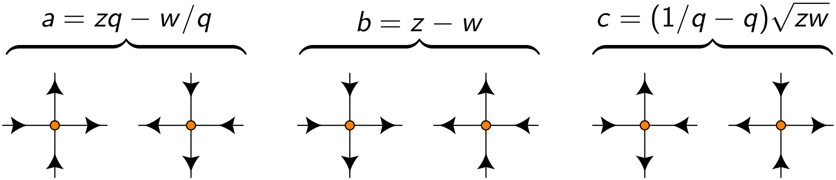


Arrows on lines : The 6-Vertex Model

- you have a degree-4 graph $\mathcal{G}$,
- variables are edge-orientations,
- weights are on the vertices,

it is Yang-Baxter–integrable
if weights depend on positions
through spectral parameters
attached to the lines,
and a global parameter q

$$\Delta = \frac{a^2 + b^2 - c^2}{2ab} = \frac{1}{2}\left(q + \frac{1}{q}\right)$$

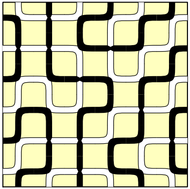


$a = zq - w/q$

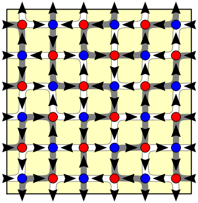
$b = z - w$

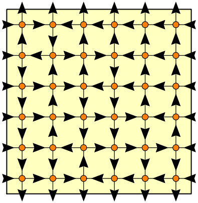
$c = (1/q - q)\sqrt{zw}$

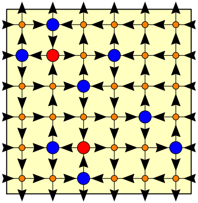
Fully-Packed Loops ➽ 6VM ➽ Alternating Sign Matrices



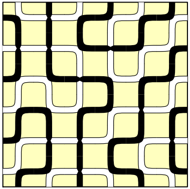
FPL
config



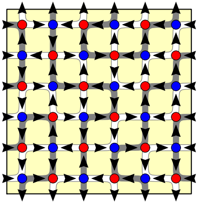




Fully-Packed Loops ➽ 6VM ➽ Alternating Sign Matrices



FPL config

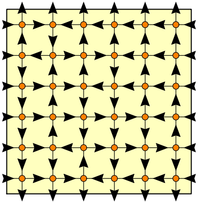


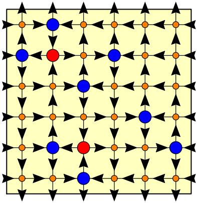
● or ● according to parity;

●━━● → ●─▶─●

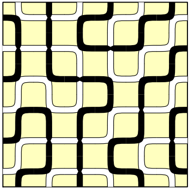
●▭● → ●─◀─●

Forget parity;

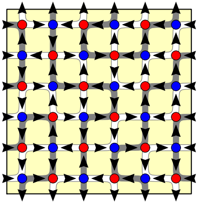




Fully-Packed Loops ➽ 6VM ➽ Alternating Sign Matrices



FPL config

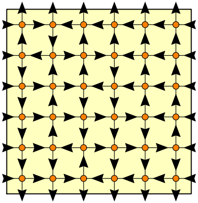


● or ● according to parity;

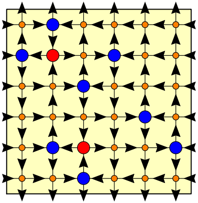
●━━● → ●─▶─●

●▭● → ●─◀─●

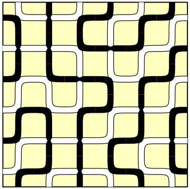
Forget parity;



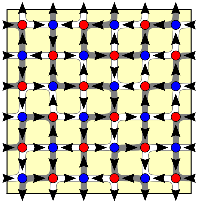
6-vertex config (DWBC)



Fully-Packed Loops ➽ 6VM ➽ Alternating Sign Matrices



FPL config

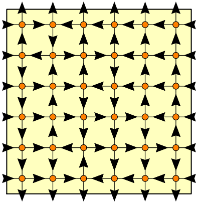


● or ● according to parity;

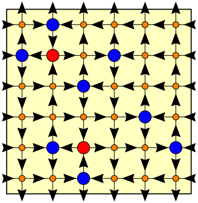
●▬● → ●▶●

●▭● → ●◀●

Forget parity;

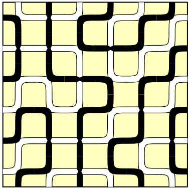


6-vertex config (DWBC)

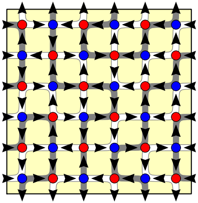


Arrow directions along rows/cols get flipped at ●, ●

Fully-Packed Loops ➭ 6VM ➭ Alternating Sign Matrices



FPL config

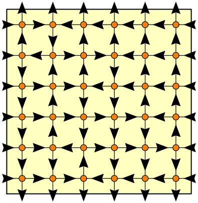


● or ● according to parity;

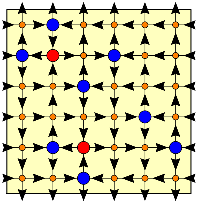
●━━● → ●─▶─●

●▭● → ●─◀─●

Forget parity;



6-vertex config (DWBC)

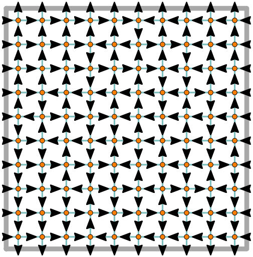


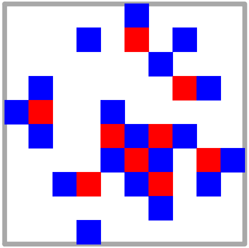
Arrow directions along rows/cols get flipped at ●, ●

ASM config

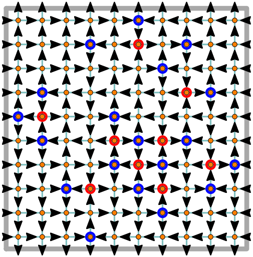
$$\begin{array}{|cccccc|}
\hline
0 & +1 & 0 & 0 & 0 & 0 \\
+1 & -1 & 0 & +1 & 0 & 0 \\
0 & 0 & +1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & +1 & 0 \\
0 & +1 & -1 & 0 & 0 & +1 \\
0 & 0 & +1 & 0 & 0 & 0 \\
\hline
\end{array}$$

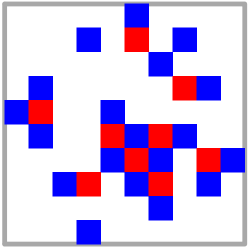
6VM ➽ permutation, height function, monotone triangle



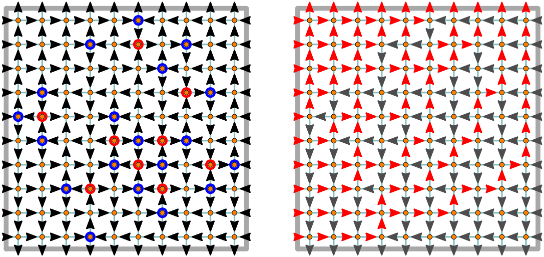


6VM ➽ permutation, height function, monotone triangle

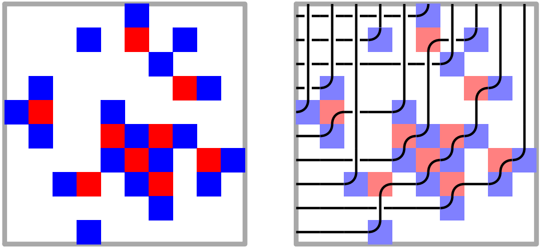




6VM ➽ permutation, height function, monotone triangle

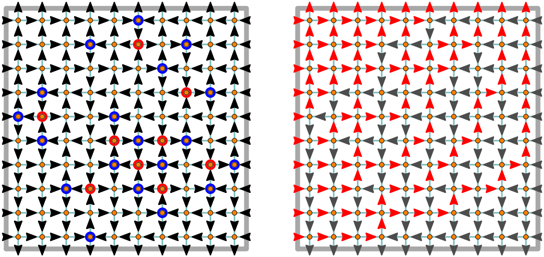


mark east- and north-bound arrows...

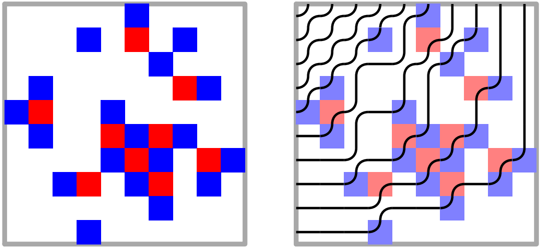


...you see a permutation of row/column-indices (crossings count the inversion number)

6VM ➠ permutation, height function, monotone triangle

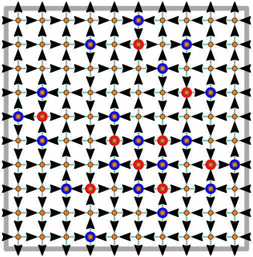


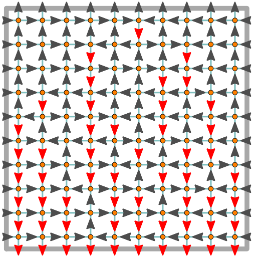
mark east- and north-bound arrows…



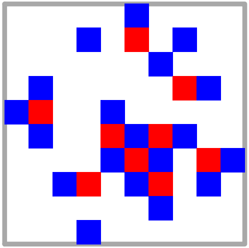
…or directed non-crossing paths, which are not of Gessel-Viennot type…

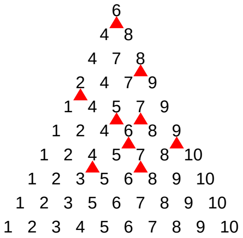
6VM ➠ permutation, height function, monotone triangle





mark south-bound arrows, and read column positions...





6

4 8

4 7 8

2 4 7 9

1 4 5 7 9

1 2 4 6 8 9

1 2 4 5 7 8 10

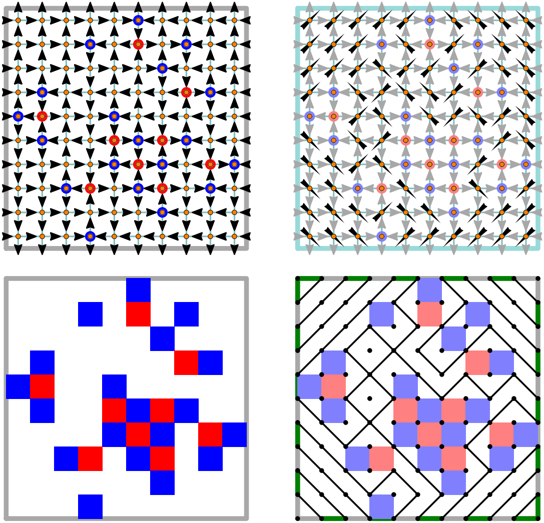
1 2 3 5 6 8 9 10

1 2 3 5 6 7 8 9 10

1 2 3 4 5 6 7 8 9 10

...you get a monotone triangle, base $= (1, 2, \dots, n)$, strict horizontally and weak elsewhere

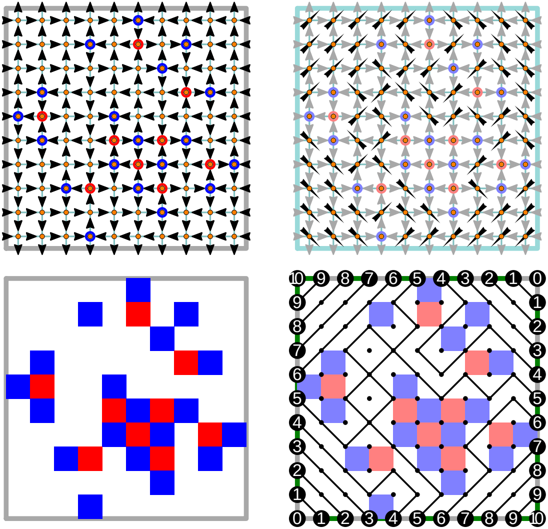
6VM ➽ permutation, height function, monotone triangle



draw a line for a coherent flow...

...you get an Eulerian graph, regions can be 2-coloured resp. boundaries

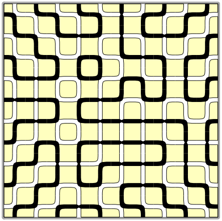
6VM ➠ permutation, height function, monotone triangle



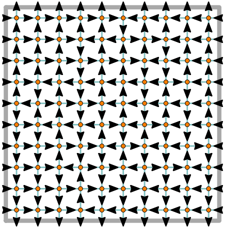
draw a line for a coherent flow...

...they're also level lines of a height function, with ±1-slope b.c.

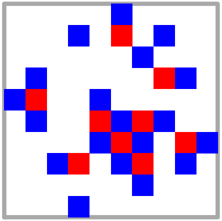
...in summary...



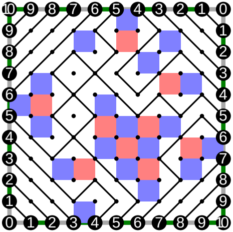
FPL



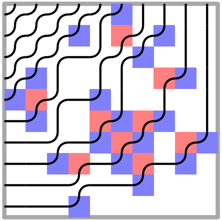
6-vertex



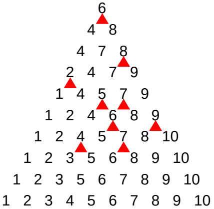
ASM



height function



quasi-NILP



6

4 8

4 7 8

2 4 7 9

1 4 5 7 9

1 2 4 6 8 9

1 2 4 5 7 8 10

1 2 3 5 6 8 9 10

1 2 3 5 6 7 8 9 10

1 2 3 4 5 6 7 8 9 10

monotone triangle

Alternating Sign Matrices: some history

Alternating Sign Matrices arose in combinatorics through the work of Mills, Robbins and Rumsey ('80s)... they took the old Dodgson Condensation Algorithm (1866)

$$\det M = \frac{\det M_{1,1} \det M_{n,n} - \det M_{1,n} \det M_{n,1}}{\det M_{1n,1n}}$$

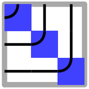
and defined a λ-determinant algorithmically, as

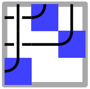
$$\det_\lambda M = \frac{\det_\lambda M_{1,1} \det_\lambda M_{n,n} - \boldsymbol{\lambda} \det_\lambda M_{1,n} \det_\lambda M_{n,1}}{\det_\lambda M_{1n,1n}}$$

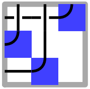
The result is (surprisingly) a Laurent polynomial in entries m_{ij}: "old" permutations take a λ^k factor, "new" terms are the non-trivial ASM, and have also $(1-\lambda)^h$ factors...

...a 3 × 3 example:

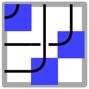
$$\det\ M = m_{11}m_{22}m_{33} + m_{12}m_{23}m_{31} + m_{13}m_{21}m_{32}$$

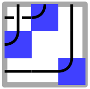


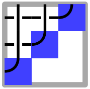




$$- m_{11}m_{23}m_{32} - m_{12}m_{21}m_{33} - m_{13}m_{22}m_{31}$$



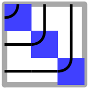


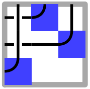


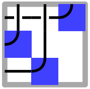
J. Propp: *Lambda-determinants and Domino Tilings*, 2005

...a 3 × 3 example:

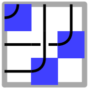
$$\det_\lambda M = m_{11}m_{22}m_{33} + \lambda^2 m_{12}m_{23}m_{31} + \lambda^2 m_{13}m_{21}m_{32}$$

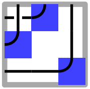


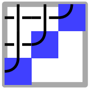




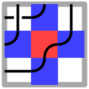
$$-\lambda m_{11}m_{23}m_{32} - \lambda m_{12}m_{21}m_{33} - \lambda^3 m_{13}m_{22}m_{31}$$







$$-\lambda(1-\lambda)\,\frac{m_{12}m_{21}m_{23}m_{32}}{m_{22}}$$



✍ J. Propp: *Lambda-determinants and Domino Tilings*, 2005

λ-determinants, years later...

…Now this Laurent phenomenon, i.e. the λ-determinant being a Laurent polynomial in matrix entries, is well understood in the wider frame of Fomin-Zelevinsky Cluster Algebras

S. Fomin, A. Zelevinsky: *The Laurent Phenomenon*, 2002

Ph. Di Francesco, R. Kedem: *Q-system, Cluster Algebras, Paths and Total Positivity*, 2010

...and the λ-determinant is a DWBC 6-Vertex partition function (with "electric fields"), integrable, at a fermionic point

$$a = -\lambda \qquad a' = 1 \qquad b = 1 \qquad b' = 1 \qquad c = m_{ij} \qquad c' = \frac{1-\lambda}{m_{ij}}$$

$$a'/a = -\lambda; \quad b'/b = 1; \quad \Delta = \frac{aa' + bb' - cc'}{2\sqrt{aa'bb'}} = 0; \quad t = \sqrt{\frac{bb'}{aa'}} = \sqrt{-\lambda}.$$

λ-determinants, years later...

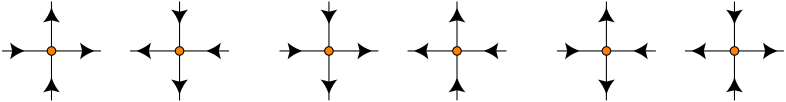
. . . Now this Laurent phenomenon, i.e. the λ-determinant being a Laurent polynomial in matrix entries, is well understood in the wider frame of Fomin-Zelevinsky Cluster Algebras

✍ S. Fomin, A. Zelevinsky: *The Laurent Phenomenon*, 2002

✍ Ph. Di Francesco, R. Kedem: *Q-system, Cluster Algebras, Paths and Total Positivity*, 2010

...and the λ-determinant is a DWBC 6-Vertex partition function (with "electric fields"), integrable, at a fermionic point

$a=-\lambda \quad a'=1 \quad b=1 \quad b'=1 \quad c=m_{ij} \quad c'=\dfrac{1-\lambda}{m_{ij}}$



$$a'/a=-\lambda; \quad b'/b=1; \quad \Delta=\frac{aa'+bb'-cc'}{2\sqrt{aa'bb'}}=0; \quad t=\sqrt{\frac{bb'}{aa'}}=\sqrt{-\lambda}.$$

The Razumov-Stroganov correspondence: in a few words

Digression on contextual combinatorial objects

Integer and Plane Partitions

Lindström–Gessel-Viennot NILP

The ASM-TSSCPP Theorem

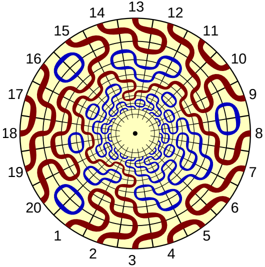
6-Vertex Model and the many faces of ASM

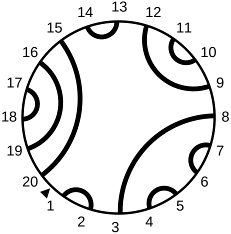
The Laurent Phenomenon

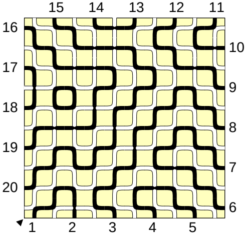
The Razumov-Stroganov correspondence: a proof

An application: FPL on the three-bundle domain

The Razumov-Stroganov correspondence... a reminder







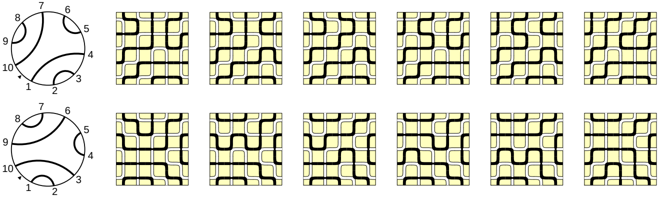
$\tilde{\Psi}_n(\pi)$: probability of π
in the $O(1)$ Dense Loop Model
in the $\{1, \ldots, 2n\} \times \mathbb{N}$ cylinder

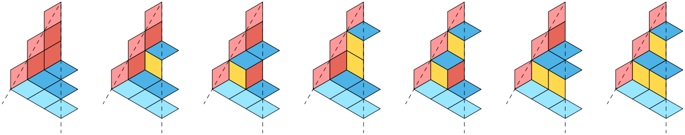
$\Psi_n(\pi)$: probability of π
for FPL with uniform measure
in the $n \times n$ square

Razumov-Stroganov correspondence
(conjecture: Razumov Stroganov, 2001; proof: AS Cantini, 2010)

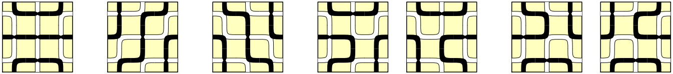
$$\tilde{\Psi}_n(\pi) = \Psi_n(\pi)$$

Dihedral symmetry of FPL: $\Psi_n(\pi) = \Psi_n(R\pi)$





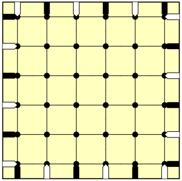
$\#$ TSSCPP in a hexagon of side $2n$ = $\#$ FPL in a square of side n



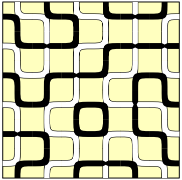
FPL in fancy domains...

We considered so far FPL in the $n \times n$ square domain, with alternating boundary conditions,

i.e. consistent fillings of this:



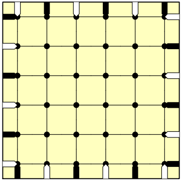
into things like this:



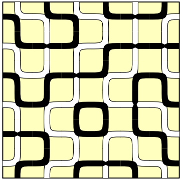
FPL in fancy domains...

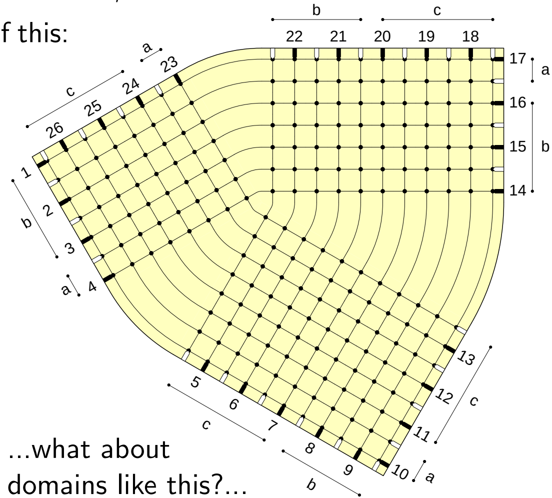
We considered so far FPL in the $n \times n$ square domain, with alternating boundary conditions,

i.e. consistent fillings of this:



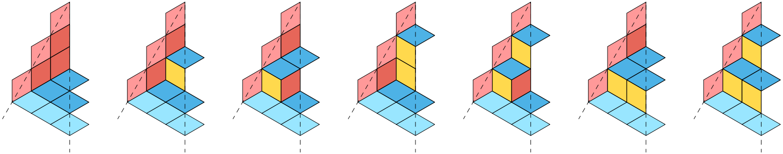
into things like this:



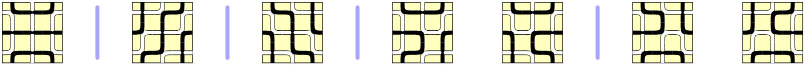


...what about domains like this?...

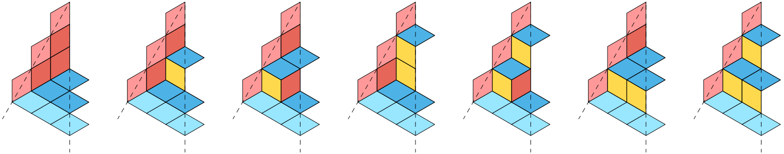
Plane Partitions and Fully-Packed Loops



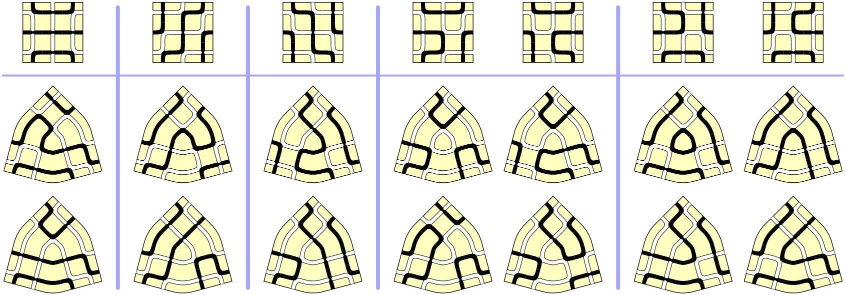
# TSSCPP in a hexagon of side $2n$ $=$ # FPL in a square of side n



Plane Partitions and Fully-Packed Loops



# TSSCPP in a hexagon of side $2n$ = # FPL in a square of side n



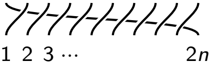
...maybe generalize Razumov-Stroganov before proving it?...

The Temperley-Lieb(1) monoid

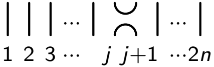
Consider the graphical action over link patterns $\pi \in \mathcal{LP}(n)$

(throw away detached cycles)

R :



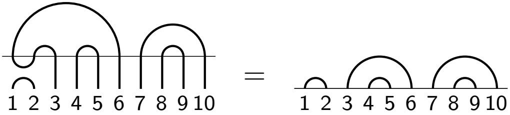
e_j :



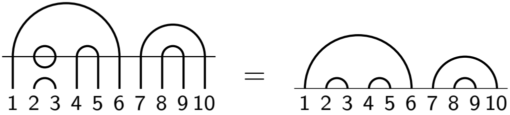
The maps $\{e_j\}_{1 \leq j \leq 2n}$ and $R^{\pm 1}$ generate a semigroup

Example:

$e_1(\pi)$:



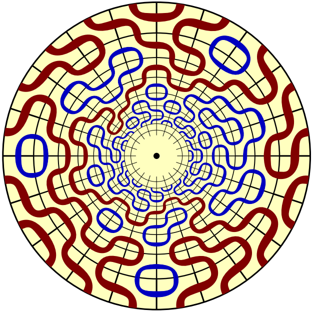
$e_2(\pi)$:



Consider the linear space $\mathbb{C}^{\mathcal{LP}(n)}$, linear span of basis vectors $|\pi\rangle$.

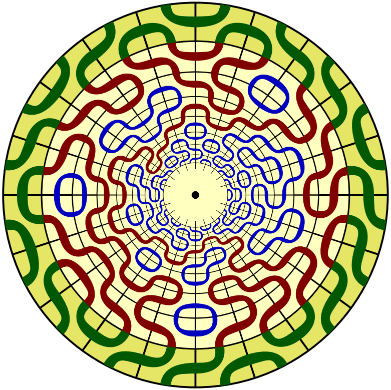
Operators e_j and $R^{\pm 1}$ are linear operators over $\mathbb{C}^{\mathcal{LP}(n)}$

$O(1)$ dense loop model: the Markov Chain over $\mathcal{LP}(n)$



A config with $t - 1$ layers.

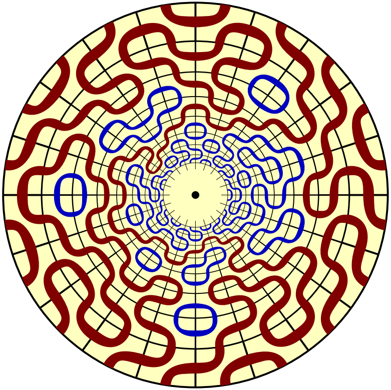
$O(1)$ dense loop model: the Markov Chain over $\mathcal{LP}(n)$



A config with $t-1$ layers.

Add a new layer, of i.i.d. tiles, with prob. $p = 1/2$...

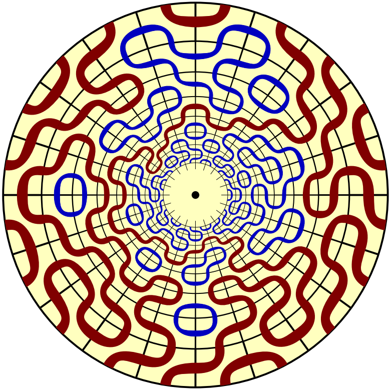
$O(1)$ dense loop model: the Markov Chain over $\mathcal{LP}(n)$



A config with $t-1$ layers.

Add a new layer, of i.i.d. tiles, with prob. $p = 1/2$...

$O(1)$ dense loop model: the Markov Chain over $\mathcal{LP}(n)$



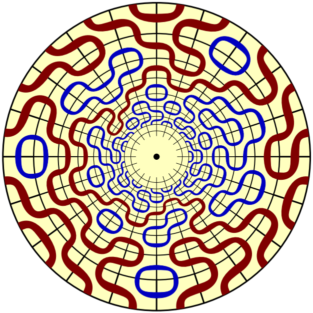
A config with $t-1$ layers.

Add a new layer, of i.i.d. tiles, with prob. $p = 1/2$...

Some loops get detached from the boundary. You have a config with t layers, and a new link pattern.

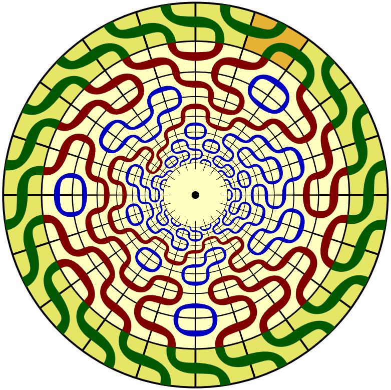
Rates $T_{p=1/2}(\pi, \pi')$

$O(1)$ dense loop model: an example at work



Now repeat the game...

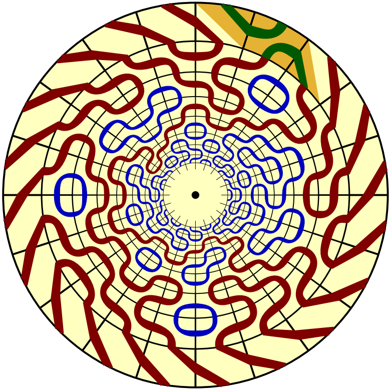
$O(1)$ dense loop model: an example at work



Now repeat the game...

...but add i.i.d. tiles, with prob. $p \longrightarrow 0$ ...

$O(1)$ dense loop model: an example at work



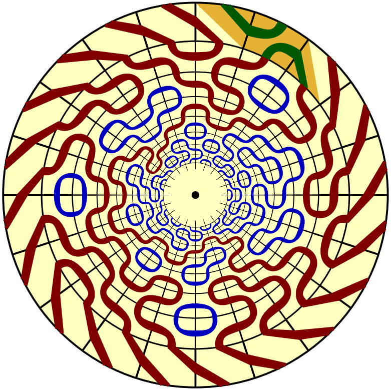
Now repeat the game...

...but add i.i.d. tiles, with prob. $p \longrightarrow 0$ ...

For most of the layers you just rotate. From time to time, you have a single non-trivial tile.

Rates $T_{p\to 0}(\pi, \pi')$

$O(1)$ dense loop model: an example at work



Now repeat the game...

...but add i.i.d. tiles, with prob. $p \longrightarrow 0$ ...

For most of the layers you just rotate. From time to time, you have a single non-trivial tile.

Rates $T_{p \to 0}(\pi, \pi')$

Non-trivial layers look like operators $R\, e_j$

Integrability: commutation of Transfer Matrices

Call $T_p(\pi, \pi')$ the matrix of transition rates (on the space of link patterns $\mathbb{C}^{\mathcal{LP}(n)}$) for tiling one layer using probability p.

Trivial: $\tilde{\Psi}_p(\pi)$, the steady state, is the unique eigenstate of $T_p(\pi, \pi')$ with all positive entries

A magic application of Yang-Baxter: $[T_p, T_{p'}] = 0$

Consequence: $\tilde{\Psi}_p(\pi) \equiv \tilde{\Psi}_{p'}(\pi)$ and we can get $\tilde{\Psi}(\pi) := \tilde{\Psi}_{1/2}(\pi)$ from the study of the easier $T_{p\to 0}(\pi, \pi')$

Call $H_n = \sum_{i=1}^{2n}(e_i - 1)$ and $|\tilde{s}_n\rangle = \sum_\pi \tilde{\Psi}(\pi)|\pi\rangle$.

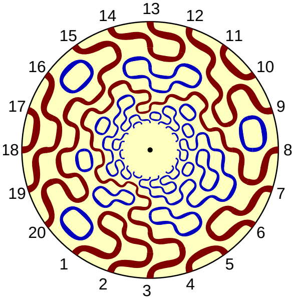
Realize $R^{-1}T_p = I + pH + \mathcal{O}(p^2)$. We thus have

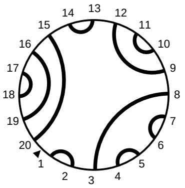
$$H_n|\tilde{s}_n\rangle = 0$$

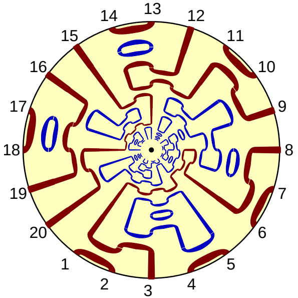
linear-algebra characterization of $\tilde{\Psi}(\pi)$

Integrability: commutation of Transfer Matrices

...said with a picture...







$$|\tilde{s}_n\rangle := \sum_{\pi\in\mathcal{LP}(n)} \tilde{\Psi}_n(\pi)|\pi\rangle$$

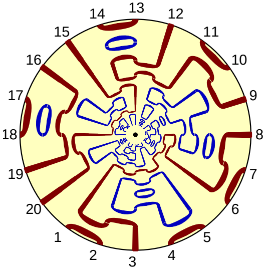
$$(T_n - 1)|\tilde{s}_n\rangle = 0$$

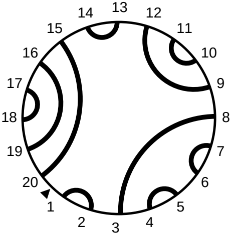
$$|\tilde{s}_n\rangle := \sum_{\pi\in\mathcal{LP}(n)} \tilde{\Psi}_n(\pi)|\pi\rangle$$

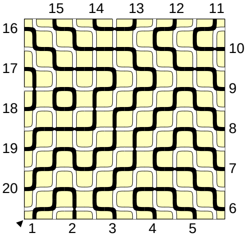
$$H_n|\tilde{s}_n\rangle = 0$$

the two linear equations for $|\tilde{s}_n\rangle$ are equivalent!

The Razumov-Stroganov correspondence: reloaded







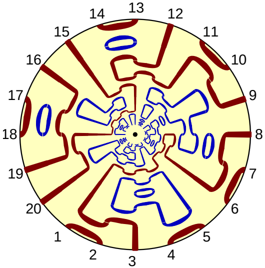
$$|\tilde{s}_n\rangle := \sum_{\pi\in\mathcal{LP}(n)} \tilde{\Psi}_n(\pi)|\pi\rangle$$

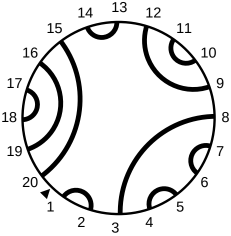
$$H_n|\tilde{s}_n\rangle = 0$$

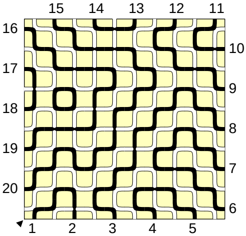
$$|s_n\rangle = \sum_{\phi\in\mathcal{F}pl(n)} |\pi(\phi)\rangle$$

$$\mathcal{F}pl(n) = \{\,\text{FPL in } n\times n \text{ square}\,\}$$

The Razumov-Stroganov correspondence: reloaded







$$|\tilde{s}_n\rangle := \sum_{\pi \in \mathcal{LP}(n)} \tilde{\Psi}_n(\pi) |\pi\rangle \qquad\qquad |s_n\rangle = \sum_{\phi \in \mathcal{Fpl}(n)} |\pi(\phi)\rangle$$

$$H_n |\tilde{s}_n\rangle = 0 \qquad\qquad \mathcal{Fpl}(n) = \{ \text{FPL in } n \times n \text{ square} \}$$

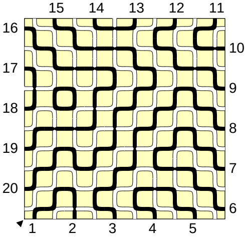
Razumov-Stroganov correspondence

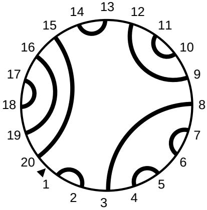
(conjecture: Razumov Stroganov, 2001; proof: AS Cantini, 2010)

$$H_n |s_n\rangle = 0$$

Wieland gyration: how it works

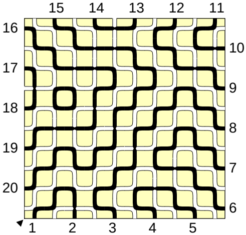
FPL config



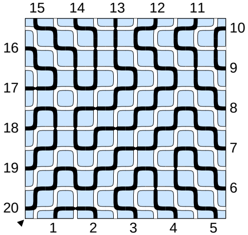


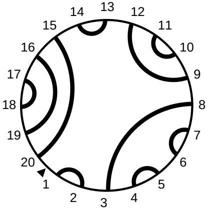
Wieland gyration: how it works

FPL config



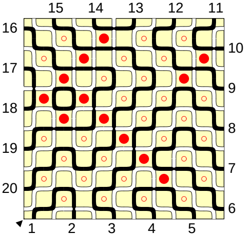
…and its conjugate,
exchanging black and white

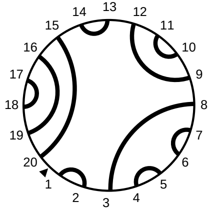




Wieland gyration: how it works

Mark faces ▣ and ▣,
of given parity

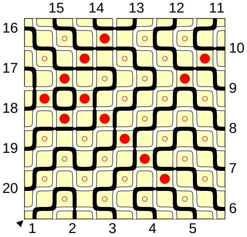


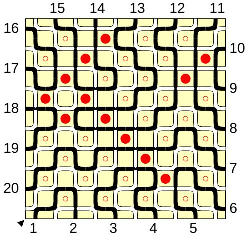


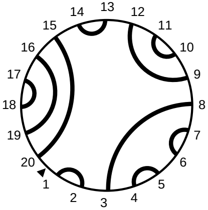
Wieland gyration: how it works

Mark faces ◫ and ⊟, of given parity

Exchange ◫⇔⊟

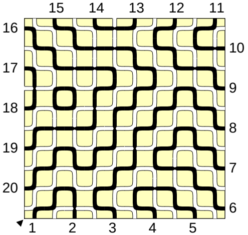


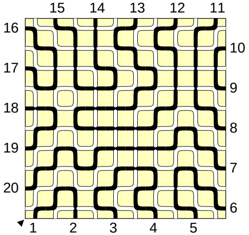


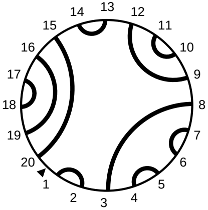


Wieland gyration: how it works

Mark faces ▣ and ▣,
of other parity

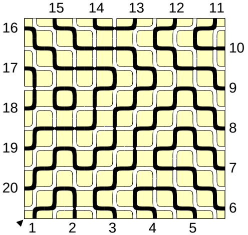


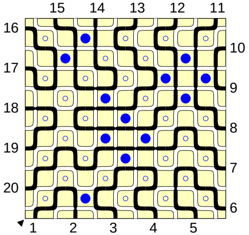


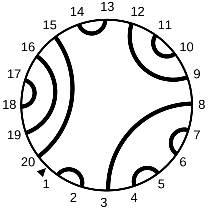


Wieland gyration: how it works

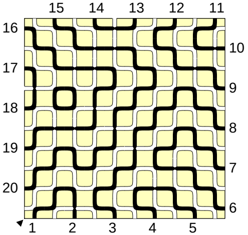
Mark faces ▣ and ▣, of other parity



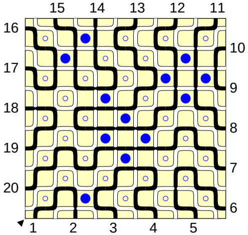




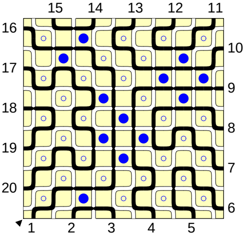
Wieland gyration: how it works

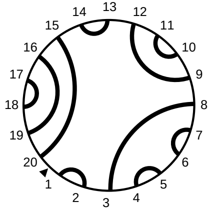


Mark faces ▣ and ▢, of other parity

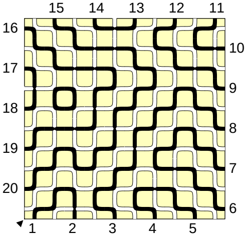


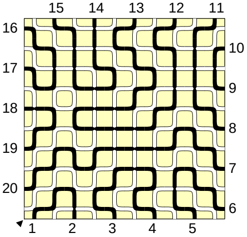
Exchange ▣ ⇔ ▢

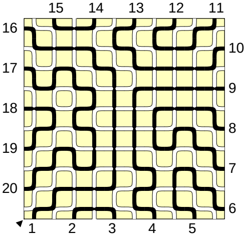


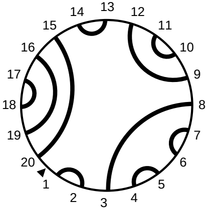


Wieland gyration: how it works





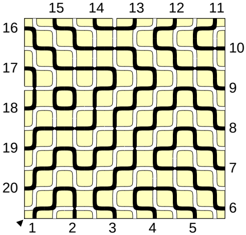


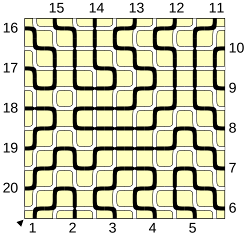


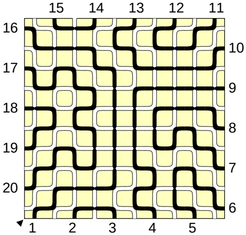
Wieland gyration: how it works

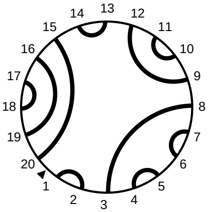
Link pattern π...

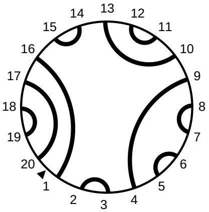
...and $R\,\pi$...





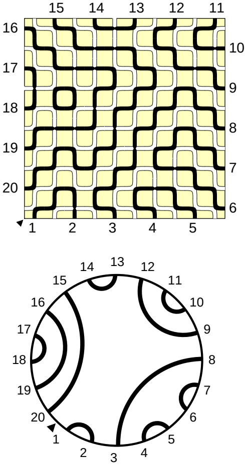




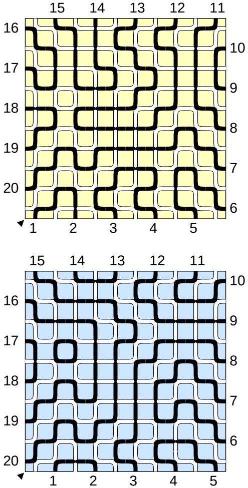


Wieland gyration: how it works

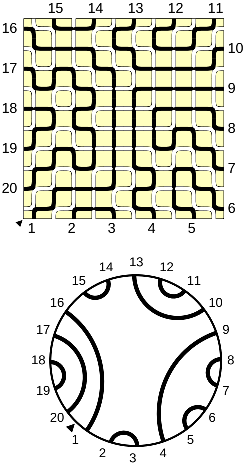
Link pattern π...



...and, on the conjugate of the intermediate step...

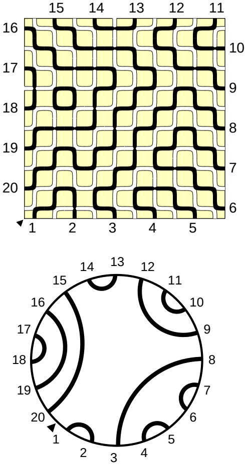


...and $R\,\pi$...

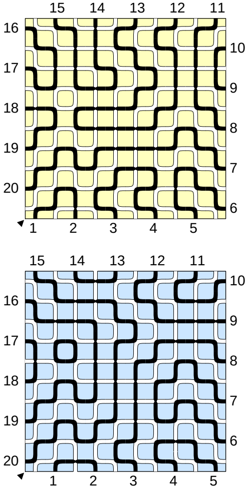


Wieland gyration: how it works

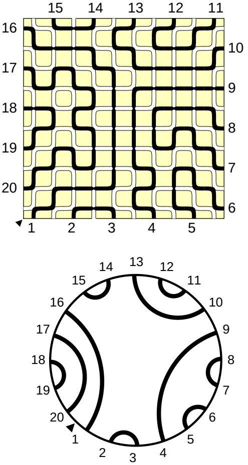
Link pattern π...



...$R^{\frac{1}{2}}\,\pi$...



...and $R\,\pi$...



An unnoticed lemma on gyration orbits

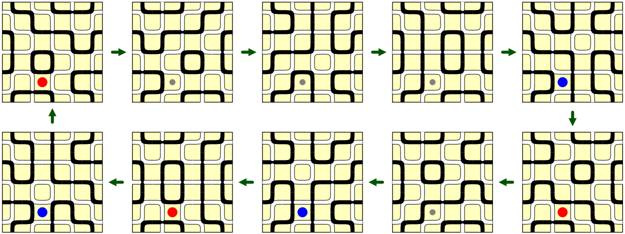
Call $\mathcal{O}(\phi)$ the orbit of ϕ under Wieland gyration.

For a face α, say

$$\mathcal{N}_\alpha(\phi) = \begin{cases} +1 & \text{if you have } ⌶ \\ -1 & \text{if you have } Ⲏ \\ 0 & \text{otherwise} \end{cases}$$

A lemma on $\mathcal{N}_\alpha$

$$\forall \text{ FPL } \phi, \text{ face } \alpha \qquad \sum_{\phi' \in \mathcal{O}(\phi)} \mathcal{N}_\alpha(\phi') = 0$$

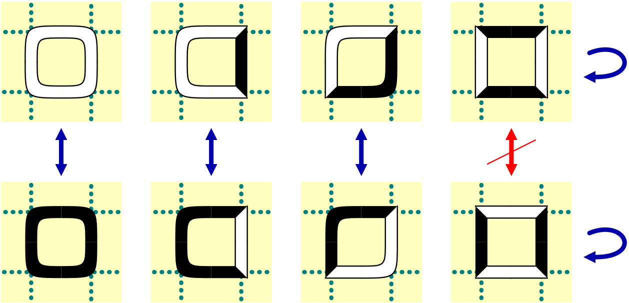


Wieland gyration: why it works

Easier to visualize the ▥⇔▤ exchange on the few ▥, ▤ faces...

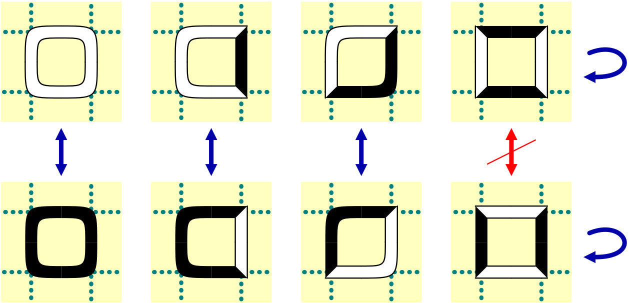
...but better use the conjugate config at intermediate step,

and think that ▥, ▤ are the only faces fixed in the transformation



Wieland gyration: why it works

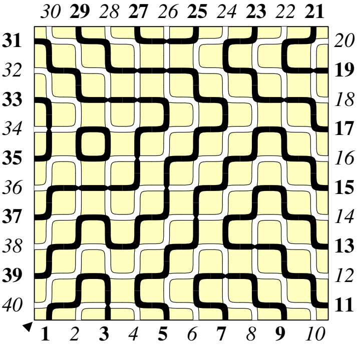
Easier to visualize the ▥⇔▤ exchange on the few ▥, ▤ faces...
...but better use the conjugate config at intermediate step,
and think that ▥, ▤ are the only faces fixed in the transformation



This inverts $\deg_{\mathrm{black}}(v) \leftrightarrow \deg_{\mathrm{white}}(v)$,
and preserves connectivity of open-path endpoints

Wieland gyration: where it works

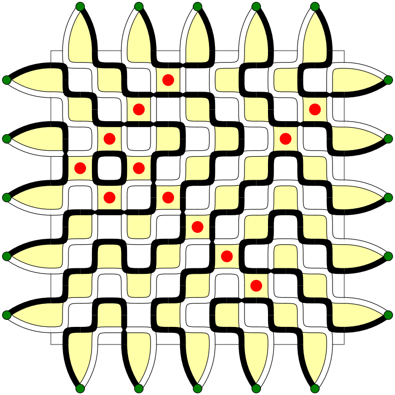
...in the original square domain for FPL we have “external legs” (i.e., vertices of degree 1)... if we pair them, to produce triangles, we solve this annoyance...



A configuration on (Λ, τ_+) (i.e., first leg is black)

Wieland gyration: where it works

...in the original square domain for FPL we have "external legs" (i.e., vertices of degree 1)... if we pair them, to produce triangles, we solve this annoyance...

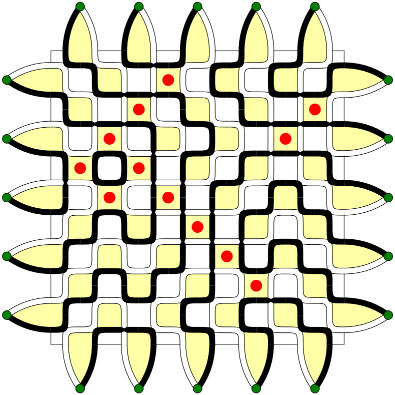


The construction of $\mathcal{G}_+$, pairing $(2j-1,2j)$ legs (plaquettes are in yellow)

mark in red and

Wieland gyration: where it works

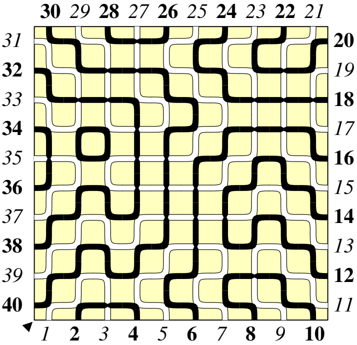
...in the original square domain for FPL we have "external legs" (i.e., vertices of degree 1)... if we pair them, to produce triangles, we solve this annoyance...



The result of map H_+

Wieland gyration: where it works

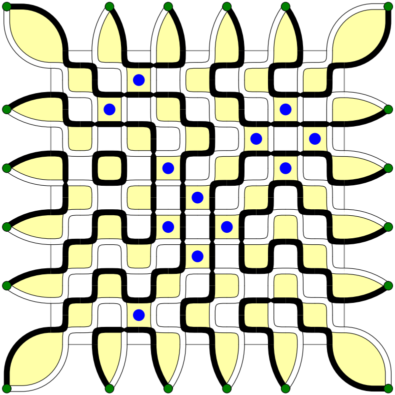
...in the original square domain for FPL we have "external legs" (i.e., vertices of degree 1)... if we pair them, to produce triangles, we solve this annoyance...



Split auxiliary vertices to recover the (Λ, τ_-) geometry (i.e., first leg is white)

Wieland gyration: where it works

...in the original square domain for FPL we have "external legs" (i.e., vertices of degree 1)... if we pair them, to produce triangles, we solve this annoyance...

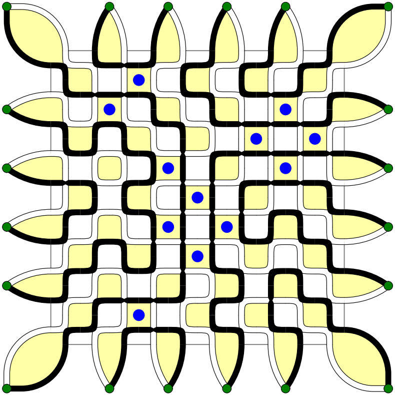


The construction of $\mathcal{G}_-$, pairing $(2j, 2j+1)$ legs

mark in blue and

Wieland gyration: where it works

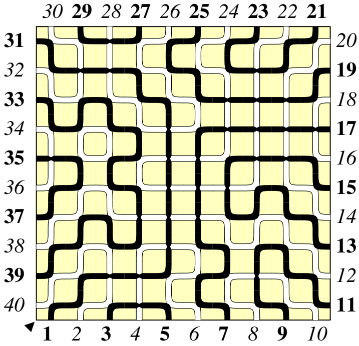
...in the original square domain for FPL we have “external legs” (i.e., vertices of degree 1)... if we pair them, to produce triangles, we solve this annoyance...



The result of map H_{-}

Wieland gyration: where it works

...in the original square domain for FPL we have "external legs" (i.e., vertices of degree 1)... if we pair them, to produce triangles, we solve this annoyance...



Split auxiliary vertices to recover the (Λ, τ_+) original geometry (with a rotated link pattern)...

Wieland gyration: where it works

So, the trick is:

- invert $\deg_{\rm black}(v) \leftrightarrow \deg_{\rm white}(v)$
- preserve connectivity of open paths

- Works with the Wieland recipe, on faces $\ell = 4$
- Works with just complementation, on faces $\ell = 1, 2, 3$
- Can't work at all on faces $\ell \geq 5$
- At boundaries, pair external legs to produce triangles

A single move exists on plenty of graphs...

then, rotation comes from two moves

...many more domains than just $n \times n$ squares have this property!

Wieland gyration: where it works

So, the trick is:
- invert $\deg_{\text{black}}(v) \leftrightarrow \deg_{\text{white}}(v)$
- preserve connectivity of open paths

- Works with the Wieland recipe, on faces $\ell = 4$
- Works with just complementation, on faces $\ell = 1, 2, 3$
- Can't work at all on faces $\ell \geq 5$
- At boundaries, pair external legs to produce triangles

A single move exists on plenty of graphs...

then, rotation comes from two moves

...many more domains than just $n \times n$ squares have this property!

Wieland gyration: where it works

So, the trick is:
- invert $\deg_{\mathrm{black}}(v) \leftrightarrow \deg_{\mathrm{white}}(v)$
- preserve connectivity of open paths

- Works with the Wieland recipe, on faces $\ell = 4$
- Works with just complementation, on faces $\ell = 1, 2, 3$
- Can't work at all on faces $\ell \geq 5$
- At boundaries, pair external legs to produce triangles

A single move exists on plenty of graphs...

then, rotation comes from two moves

...many more domains than just $n \times n$ squares have this property!

Wieland gyration: where it works

So, the trick is:

- invert $\deg_{\mathrm{black}}(v) \leftrightarrow \deg_{\mathrm{white}}(v)$
- preserve connectivity of open paths

- Works with the Wieland recipe, on faces $\ell = 4$
- Works with just complementation, on faces $\ell = 1, 2, 3$
- Can't work at all on faces $\ell \geq 5$
- At boundaries, pair external legs to produce triangles

A single move exists on plenty of graphs...

then, rotation comes from two moves

...many more domains than just $n \times n$ squares have this property!

Wieland gyration: where it works

So, the trick is:

- invert $\deg_{\mathrm{black}}(v) \leftrightarrow \deg_{\mathrm{white}}(v)$
- preserve connectivity of open paths

- Works with the Wieland recipe, on faces $\ell = 4$
- Works with just complementation, on faces $\ell = 1, 2, 3$
- Can't work at all on faces $\ell \geq 5$
- At boundaries, pair external legs to produce triangles

A single move exists on plenty of graphs...

then, rotation comes from two moves

...many more domains than just $n \times n$ squares have this property!

Wieland gyration: where it works

So, the trick is:

- invert $\deg_{\text{black}}(v) \leftrightarrow \deg_{\text{white}}(v)$
- preserve connectivity of open paths

- Works with the Wieland recipe, on faces $\ell = 4$
- Works with just complementation, on faces $\ell = 1, 2, 3$
- Can't work at all on faces $\ell \geq 5$
- At boundaries, pair external legs to produce triangles

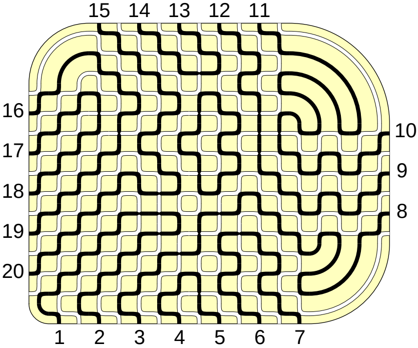
A single move exists on plenty of graphs...

then, rotation comes from two moves

...many more domains than just $n \times n$ squares have this property!

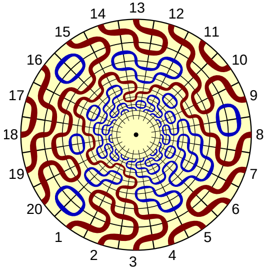
Wieland gyration: where it works

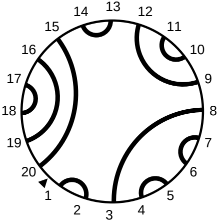
An example of our "convex planar quadrangulations, and up to 4 triangles" general domains...

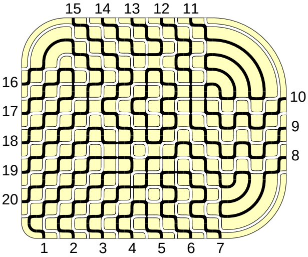


(bottom line: an elementary generalization of Wieland strategy gives rotational symmetry for FPL enumerations above)

The Razumov-Stroganov correspondence: generalised







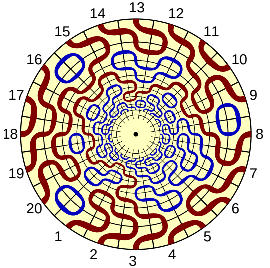
$$|\tilde{s}_n\rangle := \sum_{\pi\in\mathcal{LP}(n)} \tilde{\Psi}_n(\pi)|\pi\rangle$$

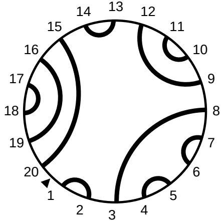
$$H_n|\tilde{s}_n\rangle = 0$$

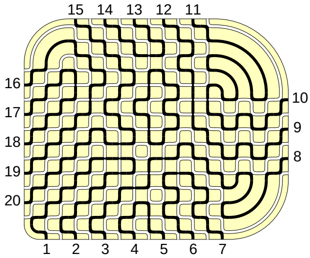
$$|s_\Lambda\rangle = \sum_{\phi\in\mathcal{F}pl(\Lambda)} |\pi(\phi)\rangle$$

$$\mathcal{F}pl(\Lambda) = \{\,\text{FPL in domain } \Lambda\}$$

The Razumov-Stroganov correspondence: generalised







$$|\tilde{s}_n\rangle := \sum_{\pi\in\mathcal{LP}(n)} \tilde{\Psi}_n(\pi)|\pi\rangle$$

$$H_n|\tilde{s}_n\rangle = 0$$

$$|s_\Lambda\rangle = \sum_{\phi\in\mathcal{Fpl}(\Lambda)} |\pi(\phi)\rangle$$

$$\mathcal{Fpl}(\Lambda) = \{\text{ FPL in domain } \Lambda\}$$

Razumov-Stroganov correspondence on Wieland domains

(proof: AS Cantini, 2010)

$$\tilde{\Psi}_n(\pi) = \Psi_\Lambda(\pi) \qquad \text{i.e.} \qquad H_n|s_\Lambda\rangle = 0$$

Yet one word on gyration... the boundary conditions

We have seen how to generalise the domain,
using black/white alternating boundary conditions

What does it happen if we generalise on boundary conditions?

Pairing consecutive legs with the same colour produces arcs, and "loses link-pattern information": gyration holds for linear combinations of $\Psi(\pi)$, instead of component-wise.

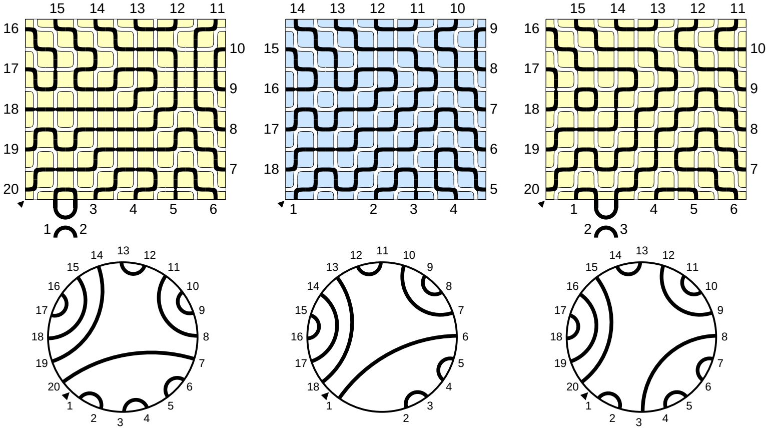
These linear combinations, induced by arcs, are well-described by Temperley-Lieb operators.

This fact suggested us that gyration on domains with a "defect" in the boundary conditions was related to Razumov-Stroganov (in its "linear-algebra formulation"...)

An example with generic boundary conditions

Example: the state $|s^c_j\rangle$ (that we define in the next slide) satisfies

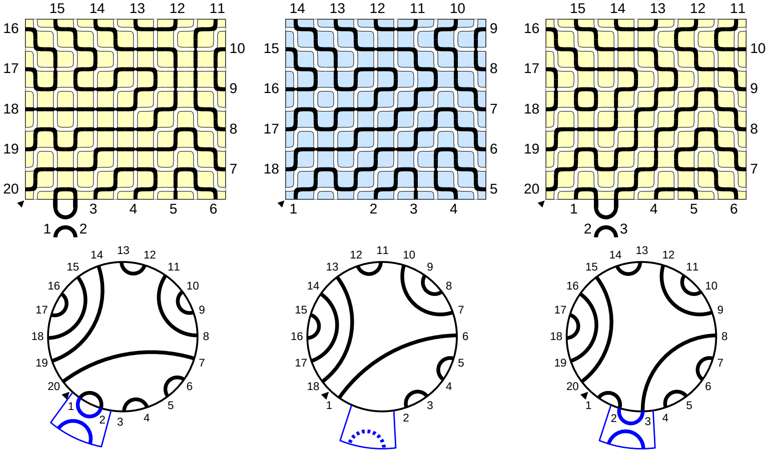
$$(R\, e_{j-1} - e_j)|s^c_j\rangle = 0$$



An example with generic boundary conditions

Example: the state $|s_j^c\rangle$ (that we define in the next slide) satisfies

$$(R\, e_{j-1} - e_j)|s_j^c\rangle = 0$$



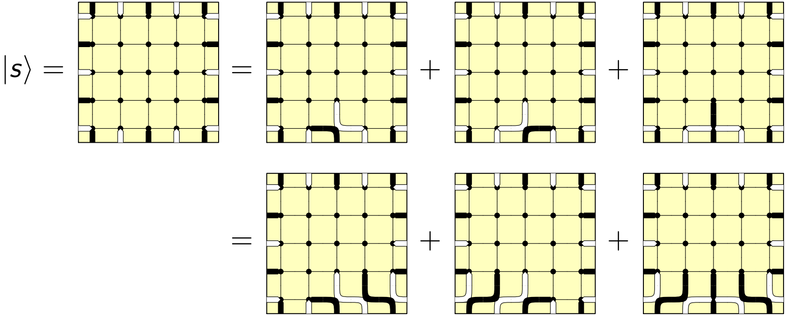
The structure of the proof

Rewrite the starting $H|s\rangle = 0$ as $\mathbf{S}(e_j - 1)|s\rangle = 0$

$$\mathbf{S} := 1 + R + \cdots + R^{2n-1}$$

Write "$|s\rangle = |s_j^a\rangle + |s_j^b\rangle + |s_j^c\rangle$",

i.e., marginalise w.r.t. a single matrix entry (on the boundary).



$$= \quad |s_j^a\rangle \quad + \quad |s_j^b\rangle \quad + \quad |s_j^c\rangle$$

The structure of the proof

Combining recursion relations with the new gyration relations gives

$$\mathbf{S}(e_j - 1)|s_j^a\rangle = \mathbf{S}(e_{j+1} - 1)(|s_{j+1}^a\rangle + |s_{j+1}^c\rangle)$$
$$\mathbf{S}(e_j - 1)|s_j^b\rangle = \mathbf{S}(e_{j-1} - 1)(|s_{j-1}^b\rangle + |s_{j-1}^c\rangle)$$

Recursion end up at the corners of the domain, and you get

$$H|s\rangle = \sum_j \mathbf{S}(e_j - 1)|s_j^c\rangle$$

Note: we have "$(e_{\boldsymbol{j}} - 1)|s_{\boldsymbol{j}}^c\rangle$" terms, not "$(e_{\boldsymbol{j}} - 1)|s_{\boldsymbol{k}}^c\rangle$" and a double sum, as in the naïve approach!

The summands are separately zero, as seen using the lemma on $\mathcal{N}_\alpha$

The Razumov-Stroganov correspondence:
in a few words

Digression on contextual combinatorial objects

Integer and Plane Partitions

Lindström–Gessel-Viennot NILP

The ASM-TSSCPP Theorem

6-Vertex Model and the many faces of ASM

The Laurent Phenomenon

The Razumov-Stroganov correspondence:
a proof

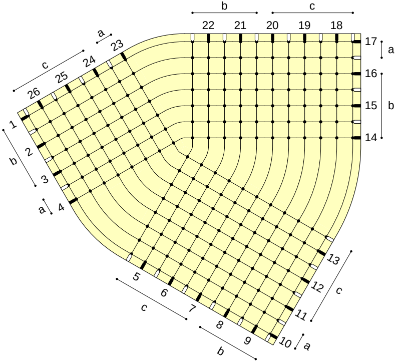
An application: FPL on the three-bundle domain

An example of our generalized ASM–TSSCPP Theorem

From Zeilberger / Kuperberg, we know that # TSSCPP of size $2n$ equals A_n, i.e. # FPL of size n.

From Razumov-Stroganov on a domain Λ (with $2n$ black legs), we know that

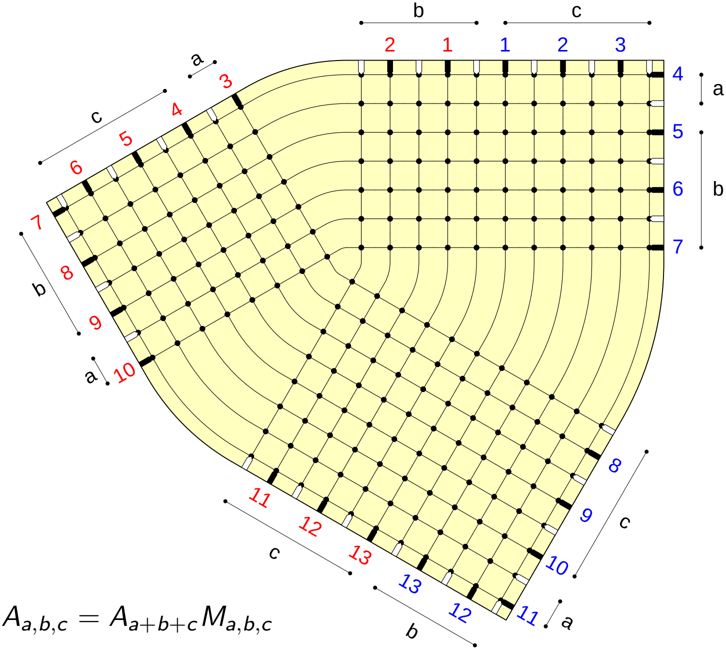
$$A_\Lambda = A_n K(\Lambda) \qquad K(\Lambda) \in \mathbb{N}$$



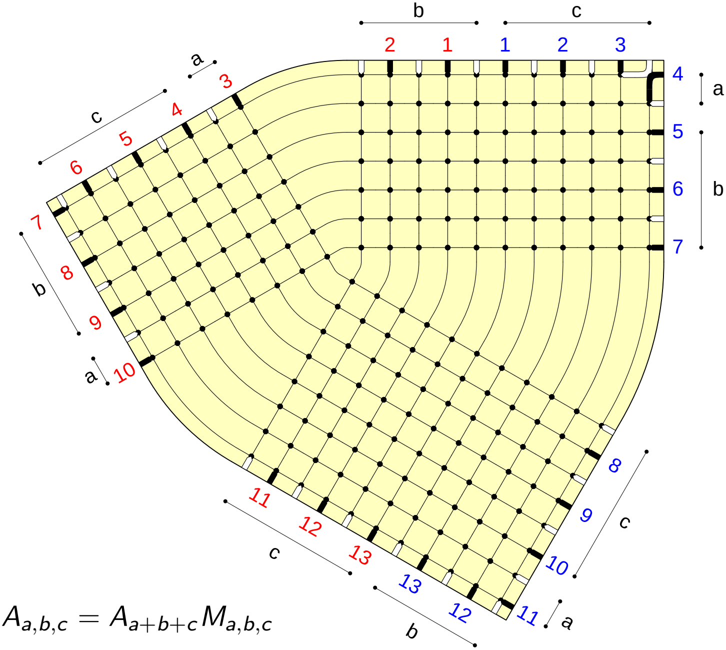
These numbers $K(\Lambda)$ are to be determined. We now do this for "three bundles", proving

$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$

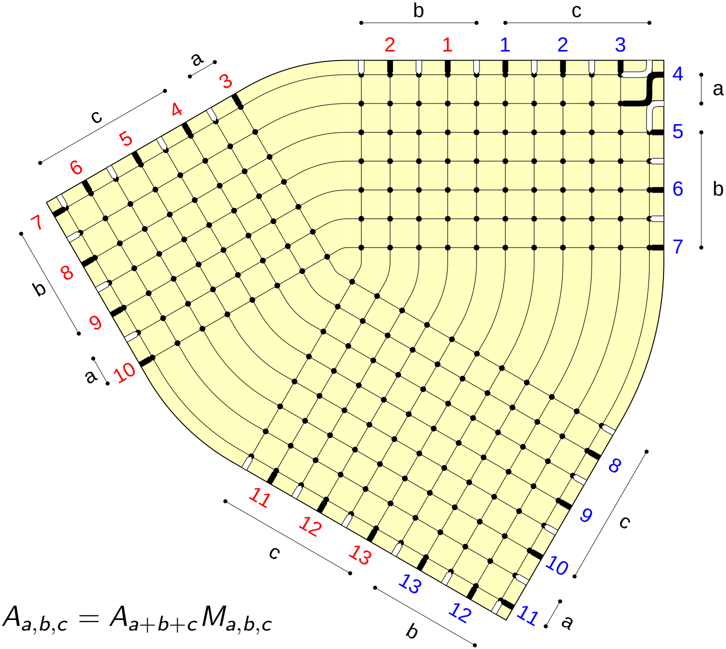
(where $M_{a,b,c}$ is the number of Plane Partitions in the $a \times b \times c$ box, MacMahon 1915 formula)



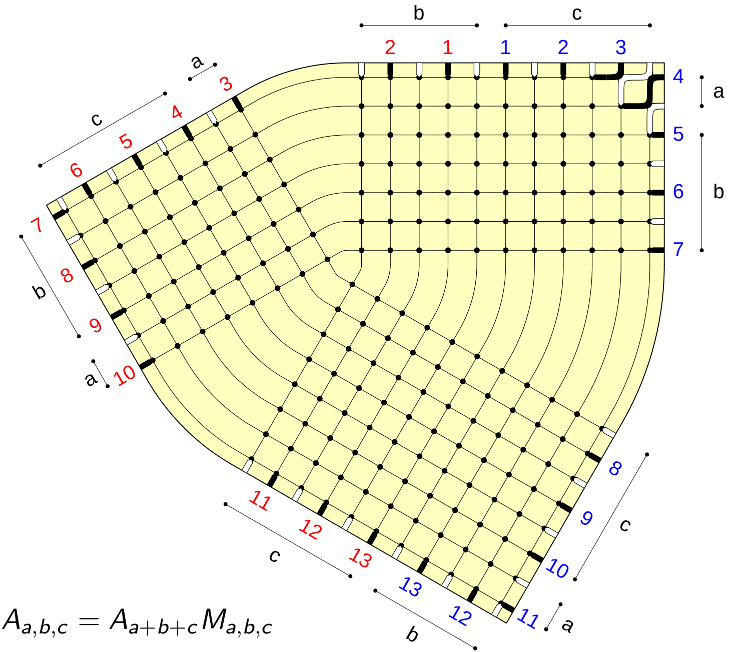
$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



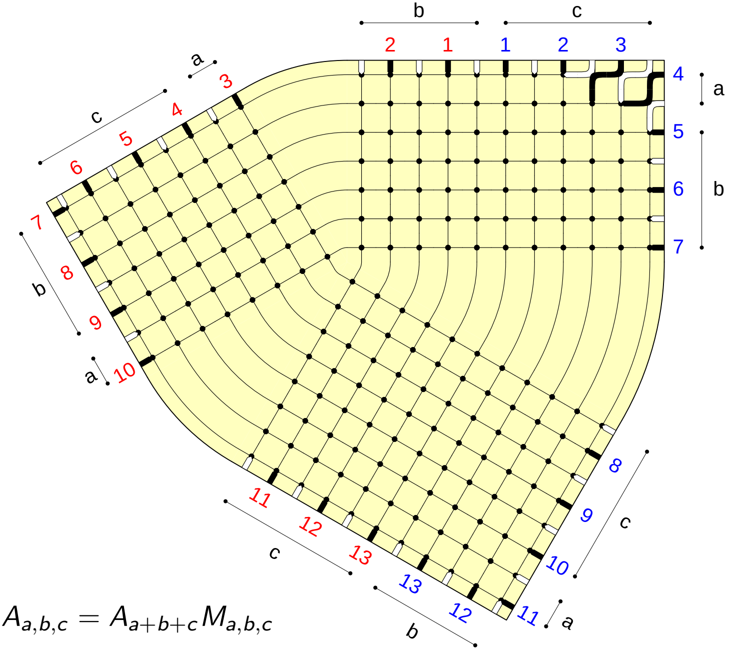
$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



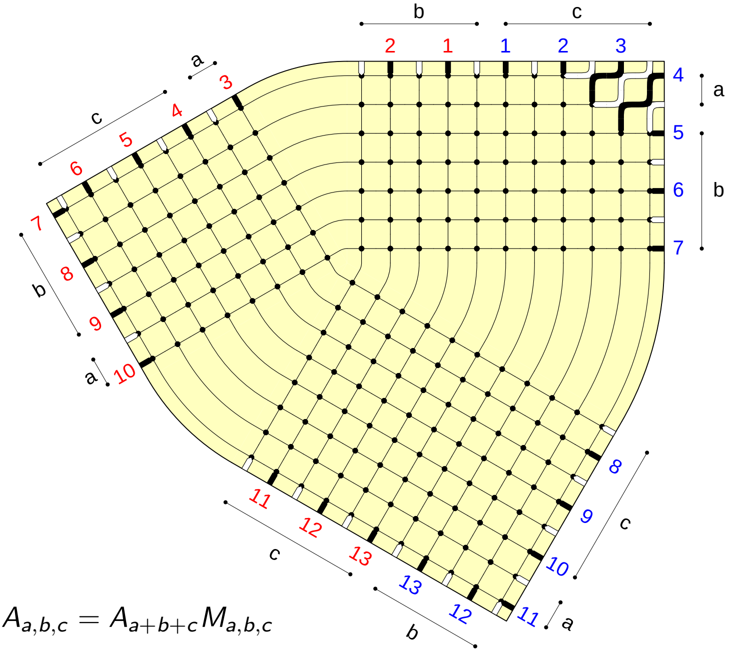
$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



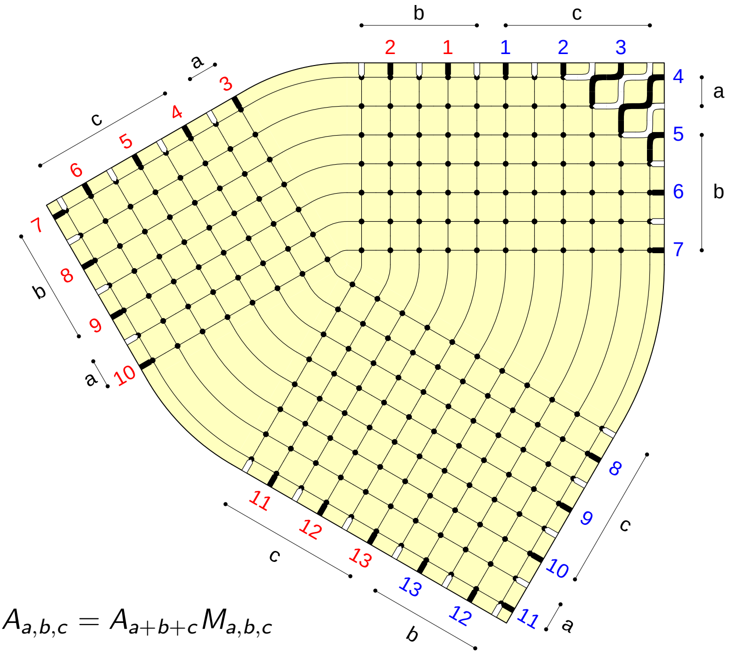
$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



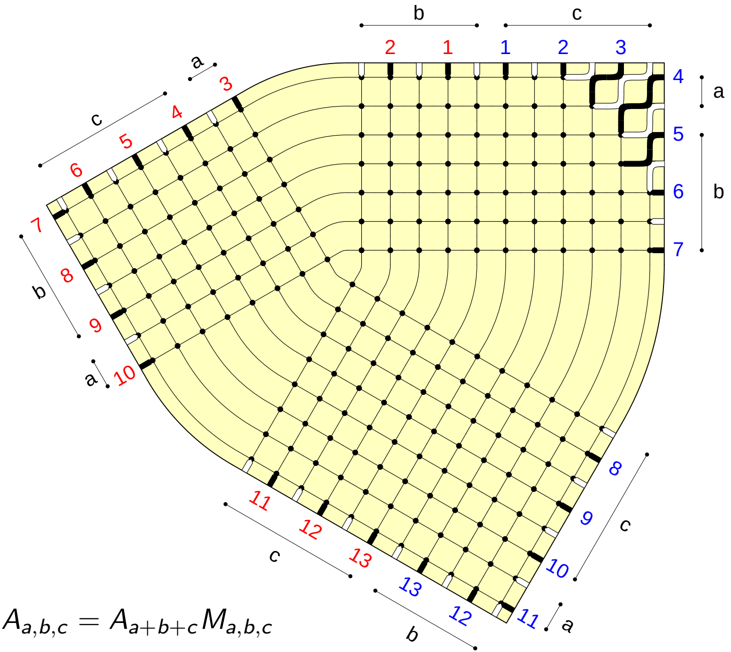
$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



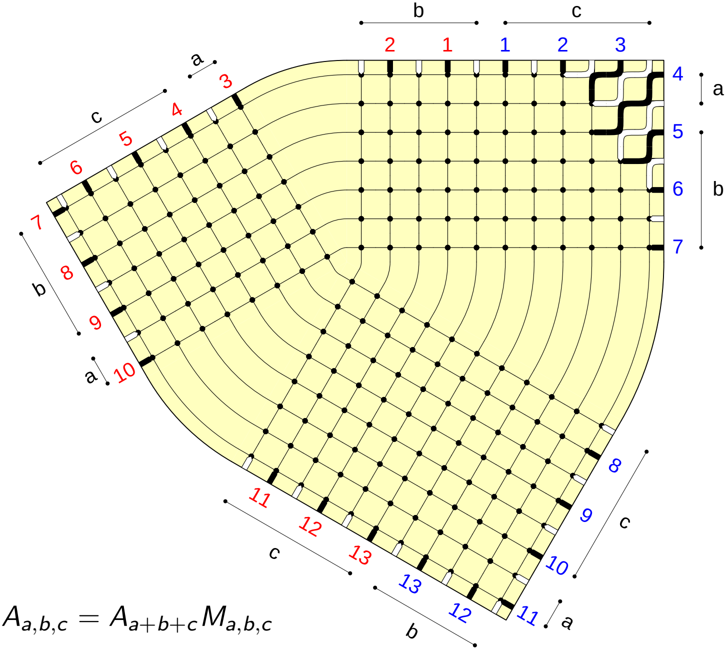
$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



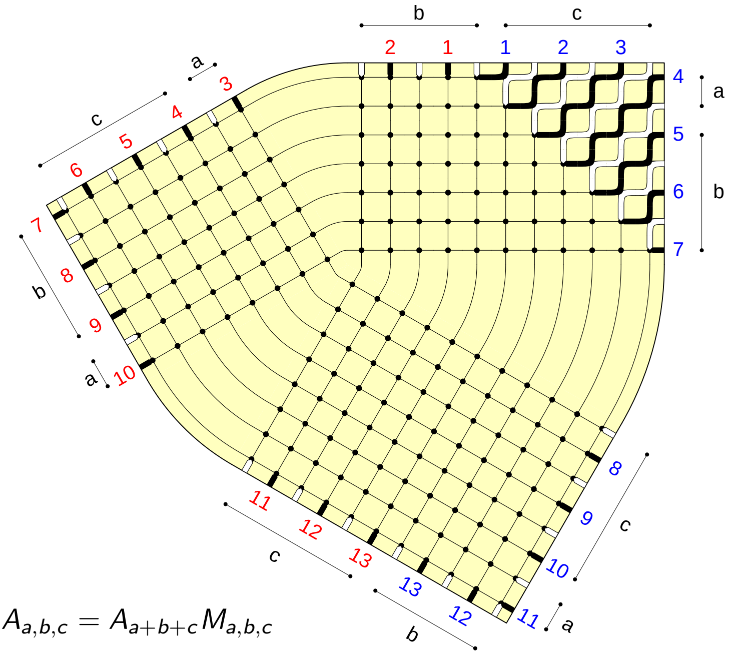
$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



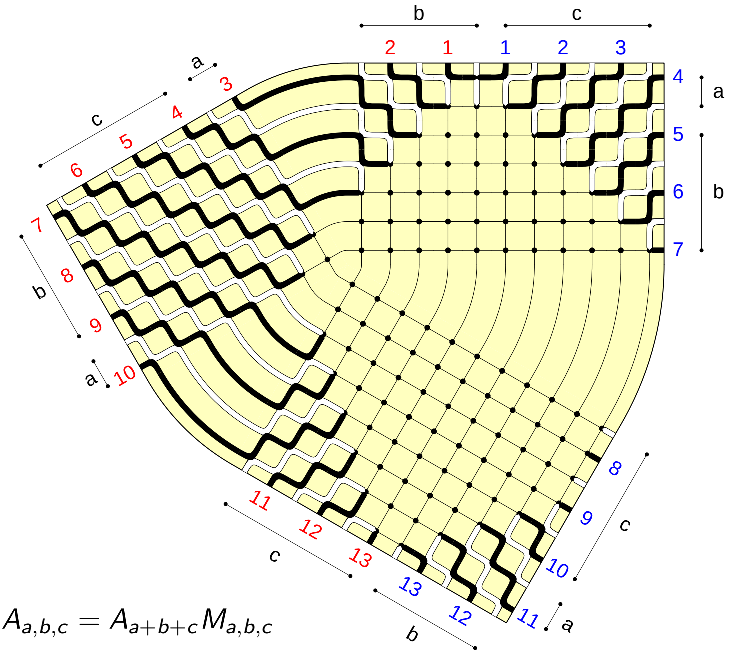
$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



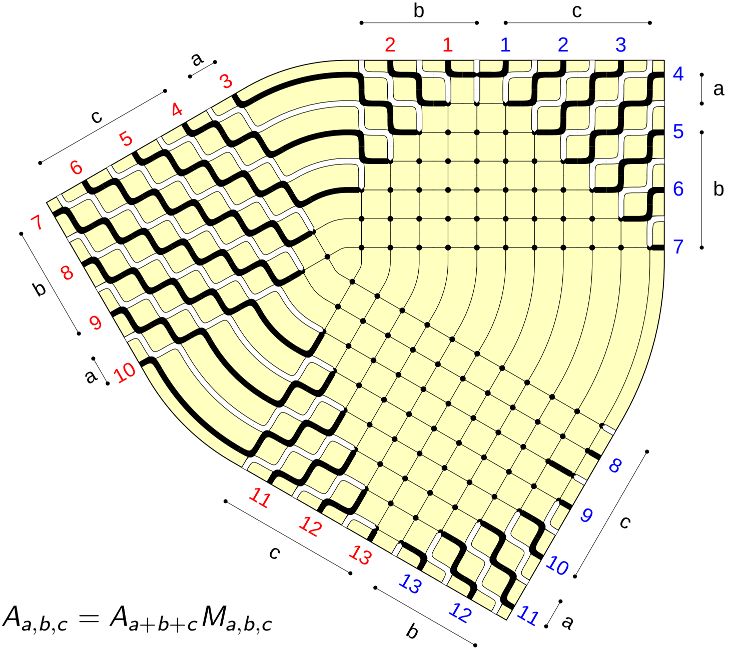
$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



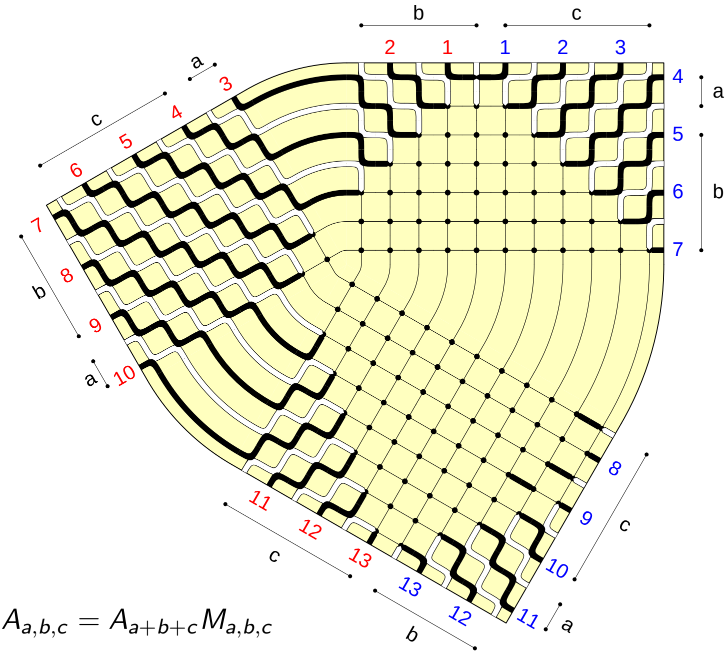
$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



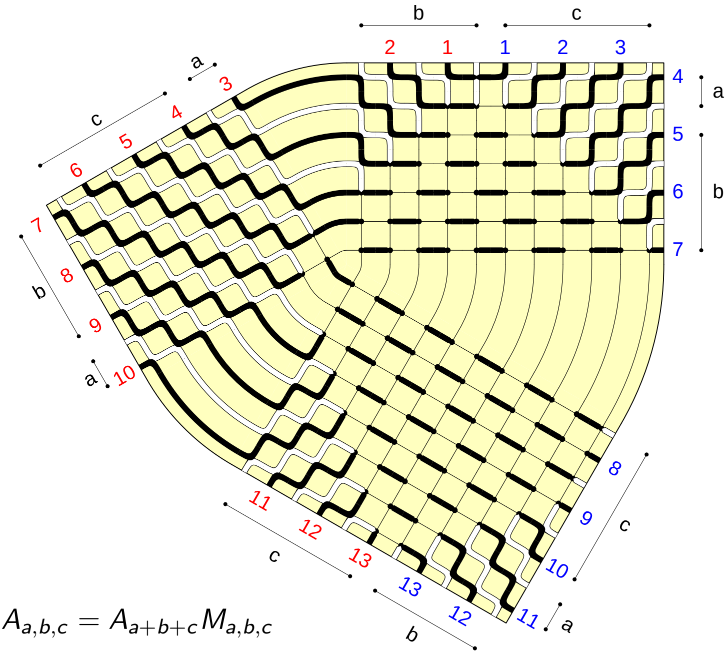
$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



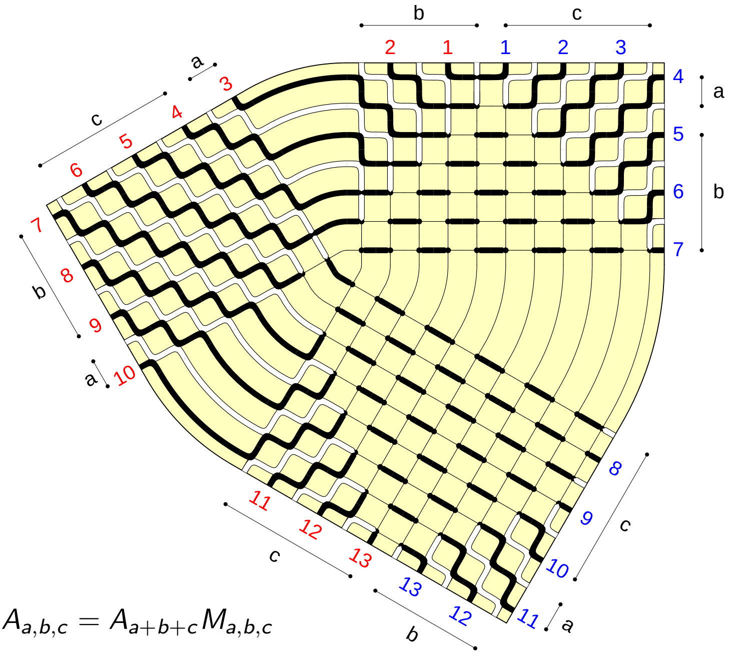
$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



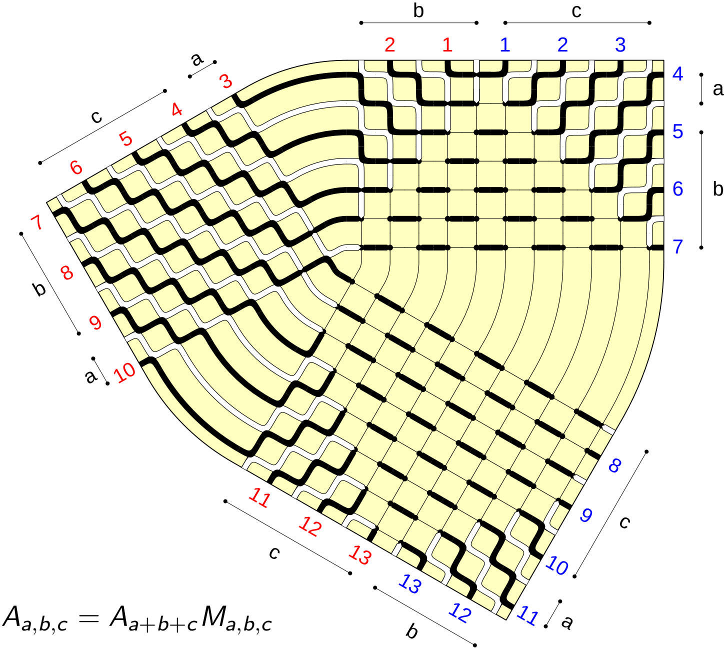
$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



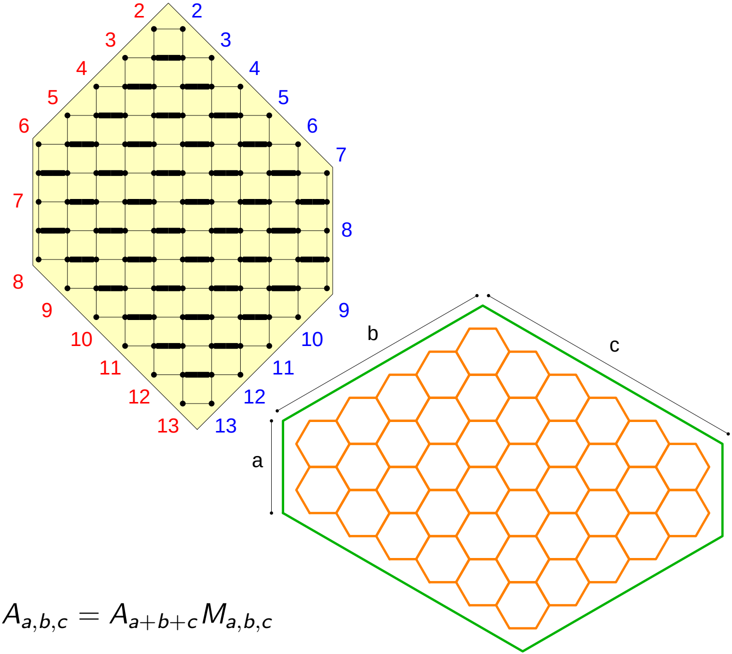
$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



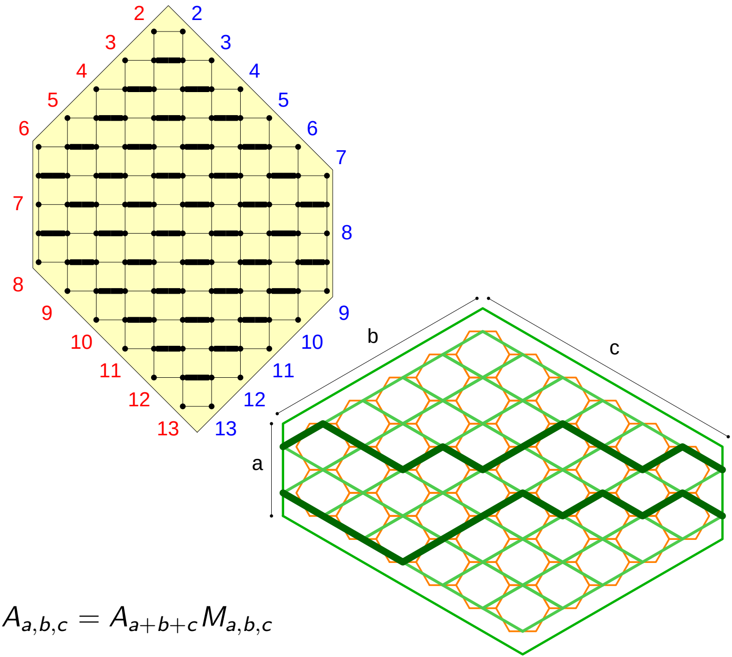
$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$



$$A_{a,b,c} = A_{a+b+c} M_{a,b,c}$$

Some bibliography

D.P. Robbins,
The story of 1, 2, 7, 42, 429, 7436,. . . Math. Intelligencer, 1991

D.M. Bressoud and J. Propp,
How the Alternating Sign Matrix Conjecture was solved Not. AMS, 1999

———

Lecture Notes of Les Houches Summer School, session 89, July 2008
Exact Methods in Low-dim. Statistical Physics and Quantum Computing

6	B. Nienhuis	*Loop models*
7	N. Reshetikhin	*Integrability of the 6-vertex model*
17	P. Zinn-Justin	*Integrability and combinatorics: selected topics*

———

L. Cantini, A. Sportiello, *Proof of the Razumov-Stroganov conjecture,*
`arXiv:1003.3376`, to appear on JCT-A