Convex polyominoes, convex permutominoes and square permutations

ENRICA DUCHI
IRIF, Université Paris Diderot

Aléa 2018

Summary of the talk

Square permutations and a generation tree

Related structures: convex polyominoes
and permutominoes

Combinatorial interpretations:

bijections and encodings

Some extensions and refinements

Permutations

Permutation $=$ bijection from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$

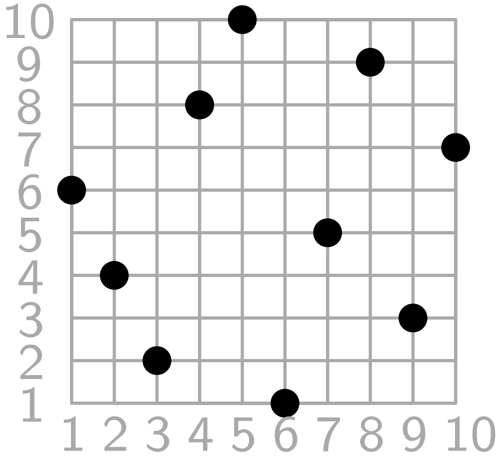
$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 6 & 4 & 2 & 8 & 10 & 1 & 5 & 9 & 3 & 7 \end{pmatrix}$$

Permutations

Permutation $=$ bijection from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 6 & 4 & 2 & 8 & 10 & 1 & 5 & 9 & 3 & 7 \end{pmatrix}$$

identified with its diagram:

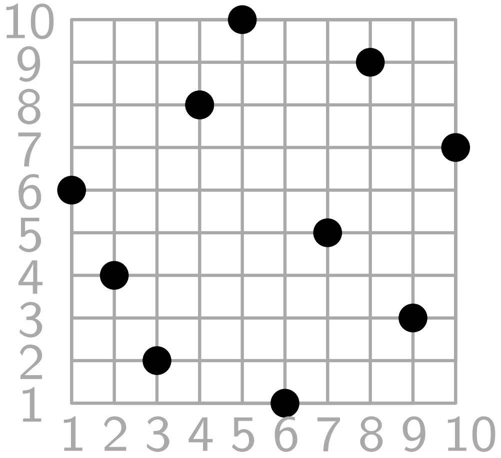


Permutations

Permutation $=$ bijection from $\{1,\ldots,n\}$ to $\{1,\ldots,n\}$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 6 & 4 & 2 & 8 & 10 & 1 & 5 & 9 & 3 & 7 \end{pmatrix}$$

identified with its diagram:



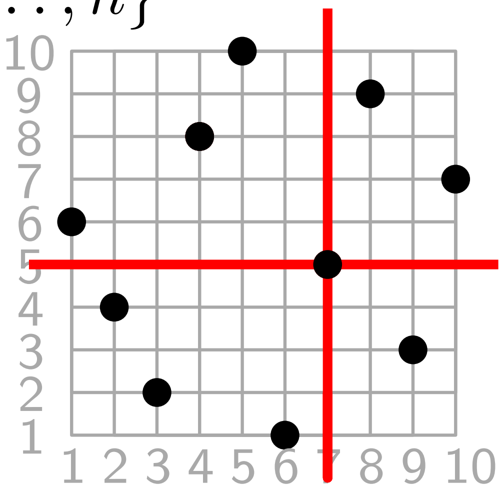
Size of a permutation $=$ its number of points

Permutations

Permutation = bijection from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 6 & 4 & 2 & 8 & 10 & 1 & 5 & 9 & 3 & 7 \end{pmatrix}$$

identified with its diagram:



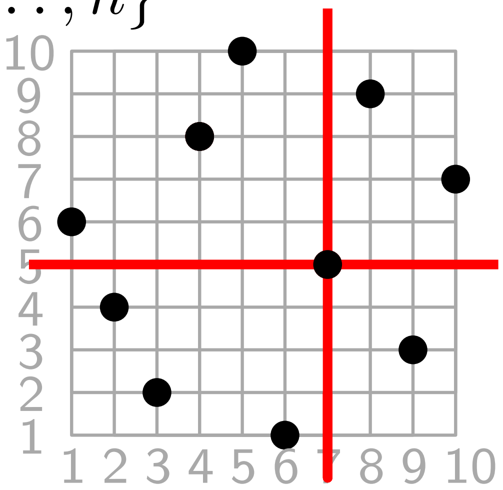
Size of a permutation = its number of points

Each point defines four quadrants.

Permutations

Permutation = bijection from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 6 & 4 & 2 & 8 & 10 & 1 & 5 & 9 & 3 & 7 \end{pmatrix}$$



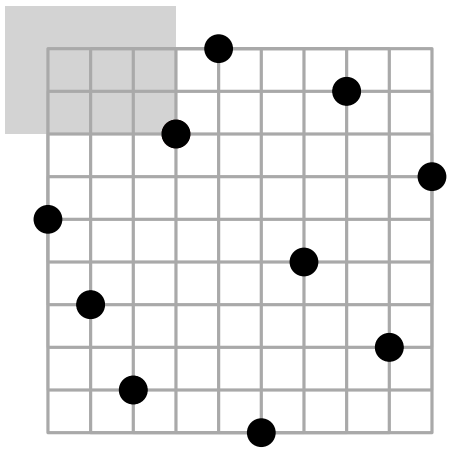
identified with its diagram:

Size of a permutation = its number of points

Each point defines four quadrants.

A point is:

- **upper left** if its upper left quadrant is empty

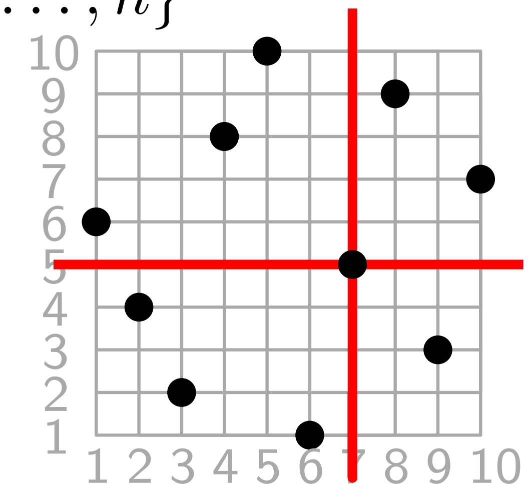


Permutations

Permutation $=$ bijection from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 6 & 4 & 2 & 8 & 10 & 1 & 5 & 9 & 3 & 7 \end{pmatrix}$$

identified with its diagram:

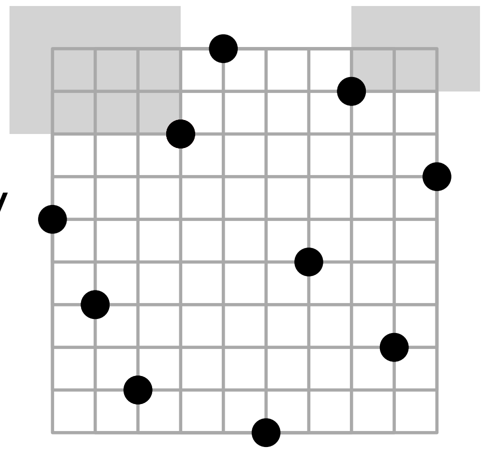


Size of a permutation $=$ its number of points

Each point defines four quadrants.

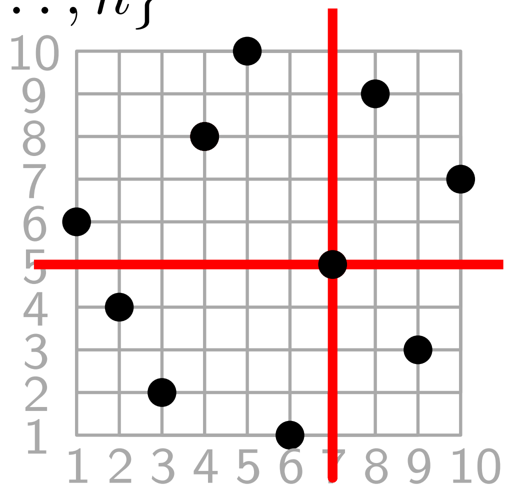
A point is:

- **upper left** if its upper left quadrant is empty
- **upper right** if its upper right quadrant is empty



Permutations

Permutation $=$ bijection from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$



$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 6 & 4 & 2 & 8 & 10 & 1 & 5 & 9 & 3 & 7 \end{pmatrix}$$

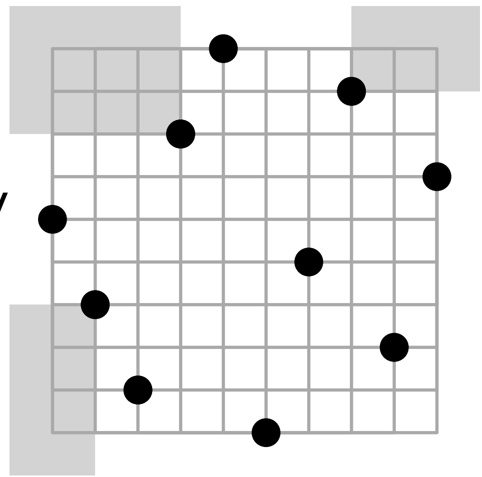
identified with its diagram:

Size of a permutation $=$ its number of points

Each point defines four quadrants.

A point is:

- **upper left** if its upper left quadrant is empty
- **upper right** if its upper right quadrant is empty
- **down left** if its down left quadrant is empty

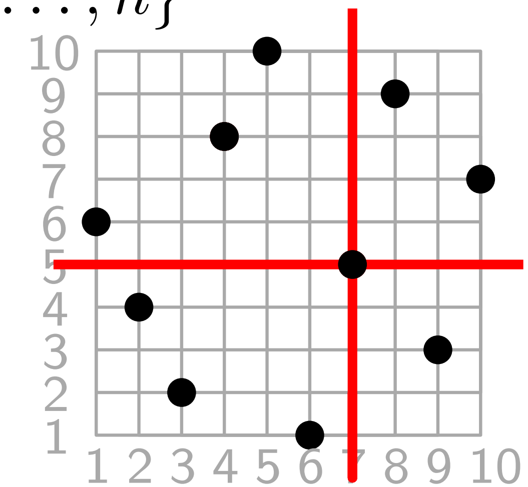


Permutations

Permutation = bijection from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 6 & 4 & 2 & 8 & 10 & 1 & 5 & 9 & 3 & 7 \end{pmatrix}$$

identified with its diagram:

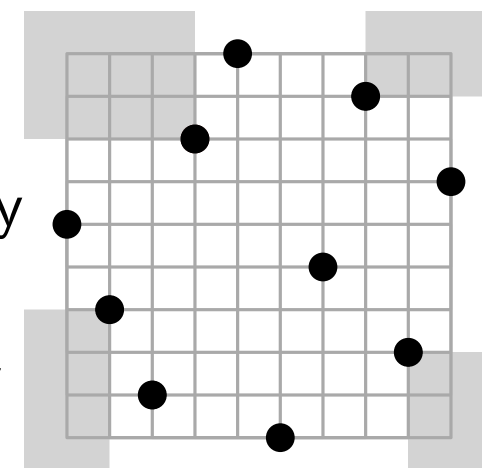


Size of a permutation = its number of points

Each point defines four quadrants.

A point is:

- **upper left** if its upper left quadrant is empty
- **upper right** if its upper right quadrant is empty
- **down left** if its down left quadrant is empty
- **down right** if its down right quadrant is empty

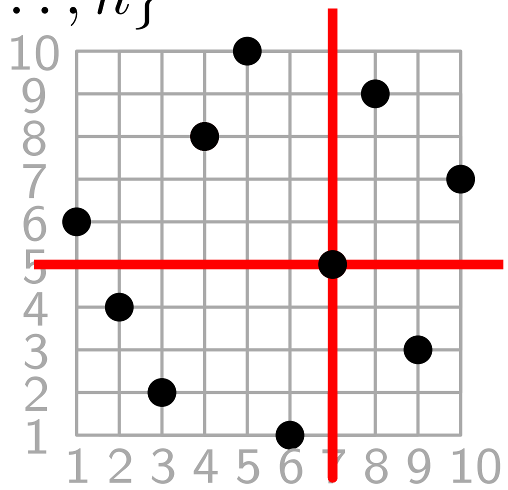


Permutations

Permutation = bijection from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 6 & 4 & 2 & 8 & 10 & 1 & 5 & 9 & 3 & 7 \end{pmatrix}$$

identified with its diagram:

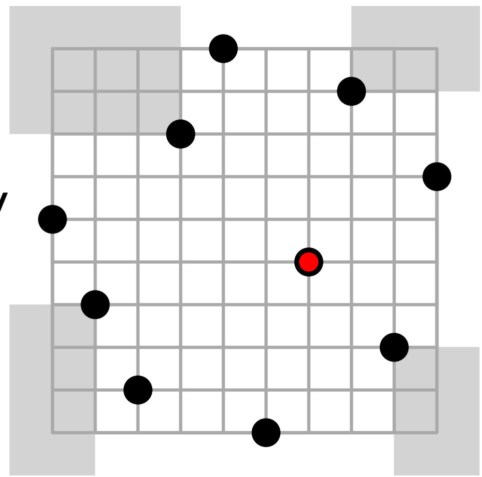


Size of a permutation = its number of points

Each point defines four quadrants.

A point is:

- **upper left** if its upper left quadrant is empty
- **upper right** if its upper right quadrant is empty
- **down left** if its down left quadrant is empty
- **down right** if its down right quadrant is empty
- internal if it is none of above



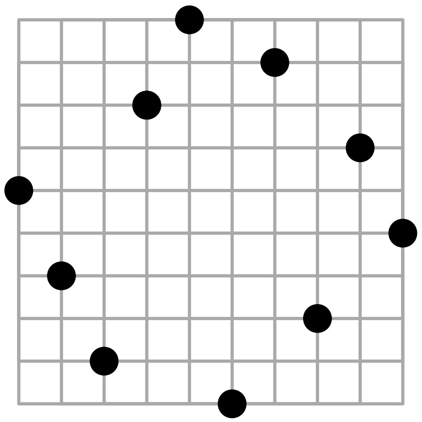
(Equivalent to notions of LR-max, RL-max, LR-min, RL-min)

Square permutations

Square permutation

= permutation without interieur point

= all points are upper left, upper right, down left or down right



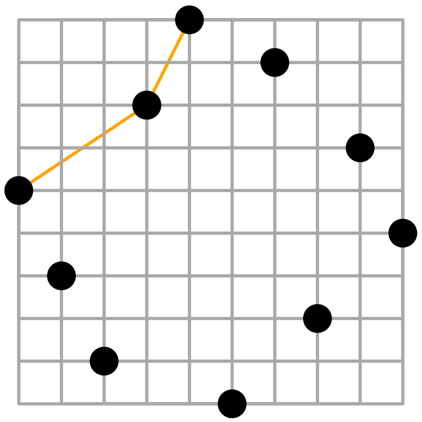
Square permutations

Square permutation

= permutation without interieur point

= all points are upper left, upper right, down left or down right

Upper left points form the upper left path

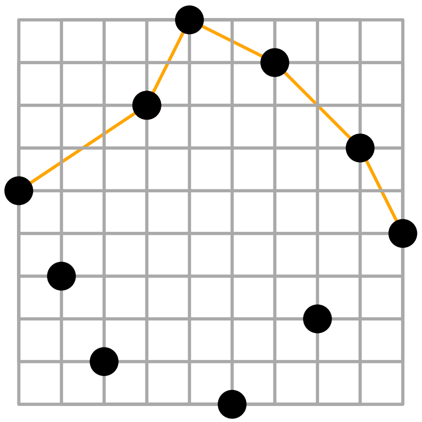


Square permutations

Square permutation

= permutation without interieur point

= all points are upper left, upper right, down left or down right



Upper left points form the upper left path

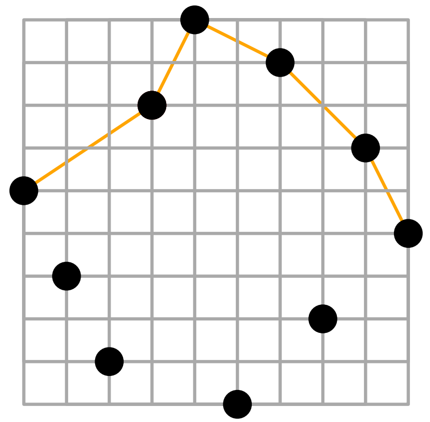
Upper right points form the upper right path

Square permutations

Square permutation

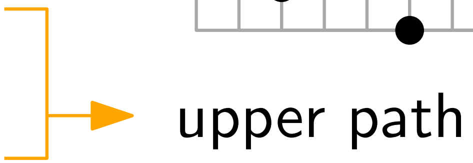
= permutation without interieur point

= all points are upper left, upper right,
down left or down right

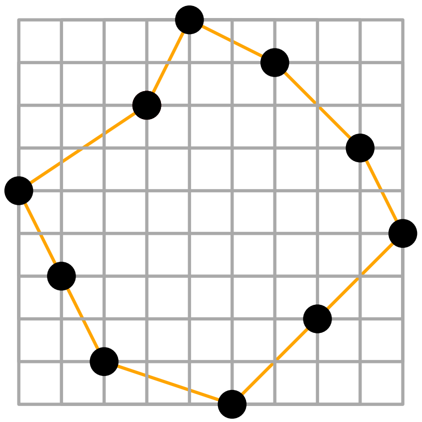


Upper left points form the upper left path

Upper right points form the upper right path



Square permutations



Square permutation

= permutation without interieur point

= all points are upper left, upper right,
down left or down right

Upper left points form the upper left path

Upper right points form the upper right path → upper path

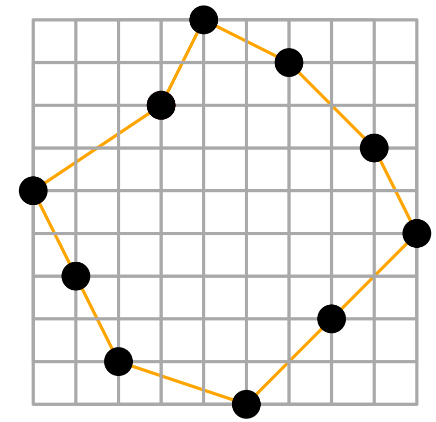
idem for down-left, down-right, and left, right and down path.

Square permutations

Square permutation

= permutation without interieur point

= all points are upper left, upper right, down left or down right

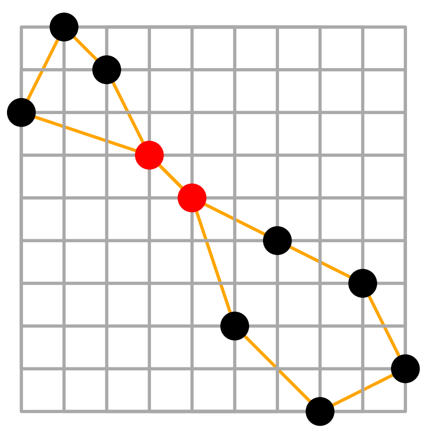


Upper left points form the upper left path
Upper right points form the upper right path

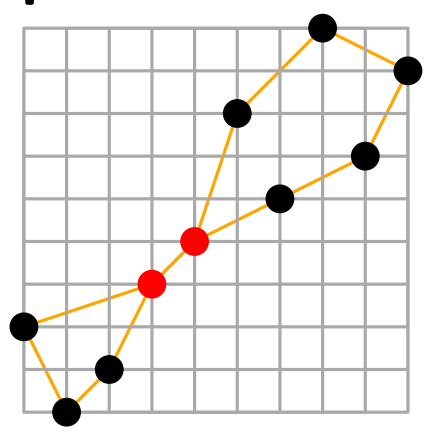
→ upper path

idem for down-left, down-right, and left, right and down path.

Square permutations can have **double points**:

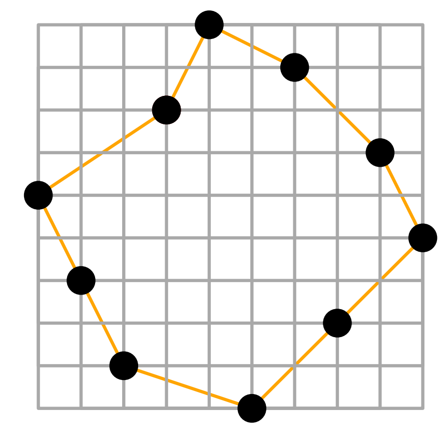


down-left and upper-right



upper-left and down-right

Square permutations



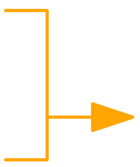
Square permutation

= permutation without interieur point

= all points are upper left, upper right, down left or down right

Upper left points form the upper left path

Upper right points form the upper right path



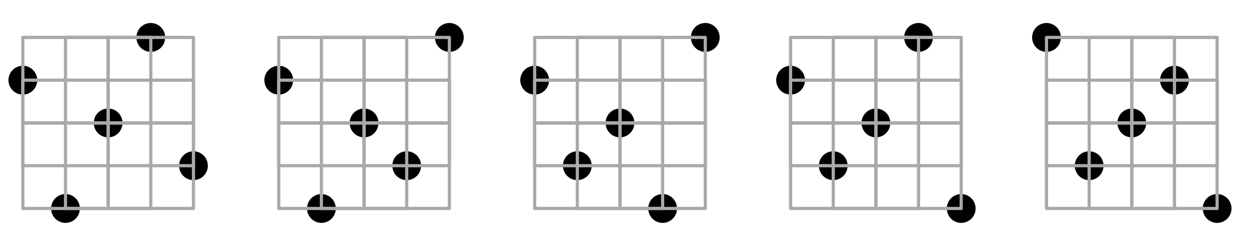
upper path

idem for down-left, down-right, and left, right and down path.

Our aim is to count these square permutations

First terms: 1, 2, 6, 24, 104=128-24, ...

Indeed the smallest non-square permutations have size 5:



$\times 2$ $\times 8$ $\times 4$ $\times 8$ $\times 2$ $= 24$

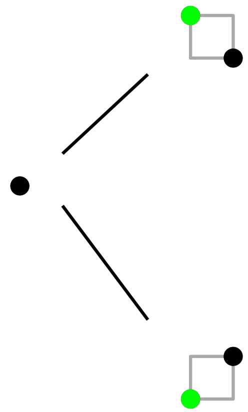
Enumeration via generating trees

How to grow a permutation?

A standard way to generate permutations is by inserting a front point in all possible ways:

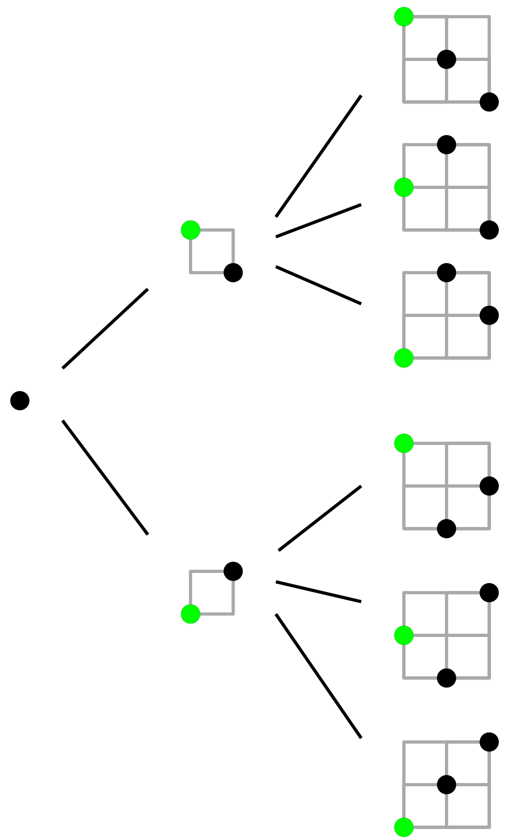
How to grow a permutation?

A standard way to generate permutations is by inserting a front point in all possible ways:



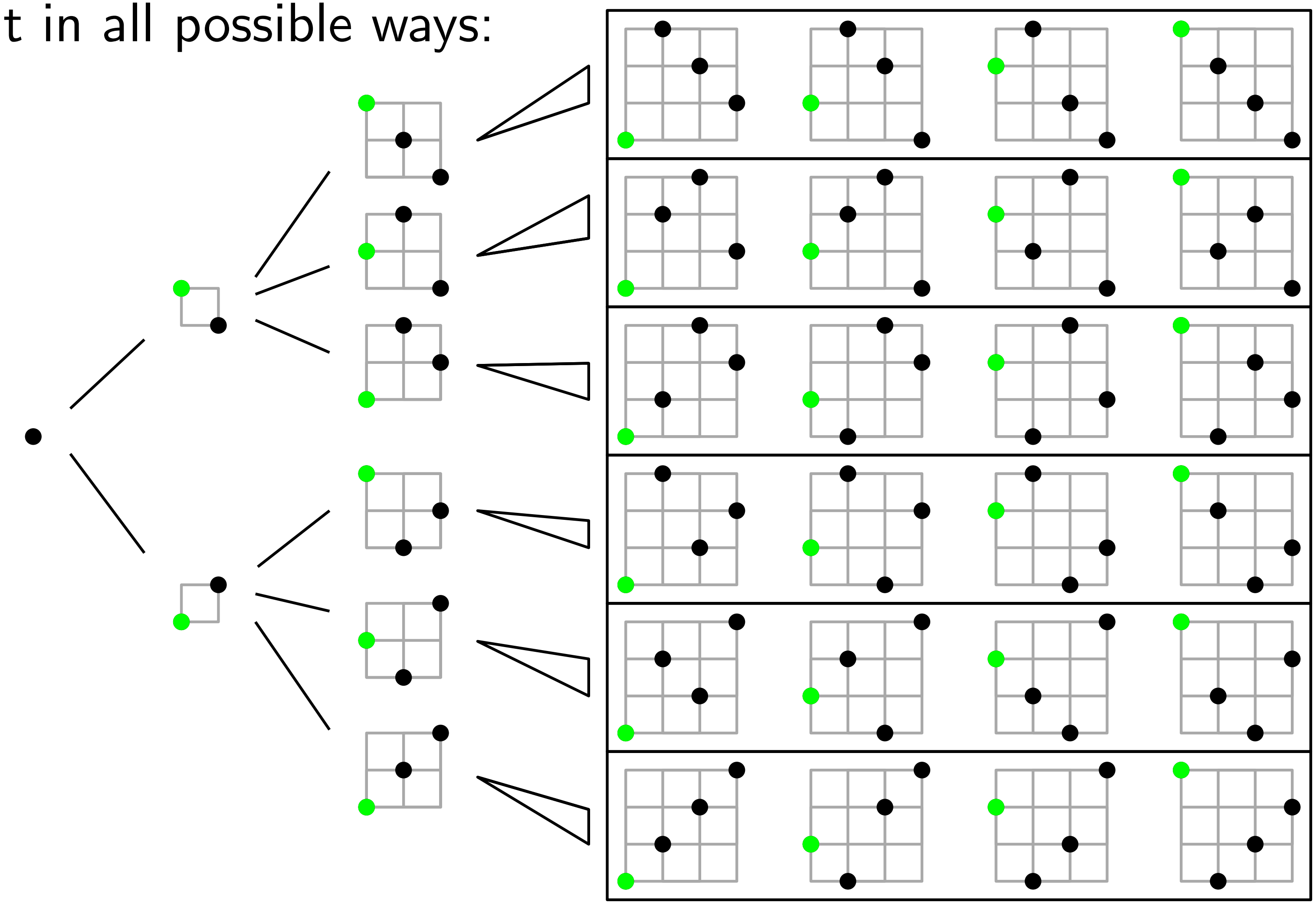
How to grow a permutation?

A standard way to generate permutations is by inserting a front point in all possible ways:



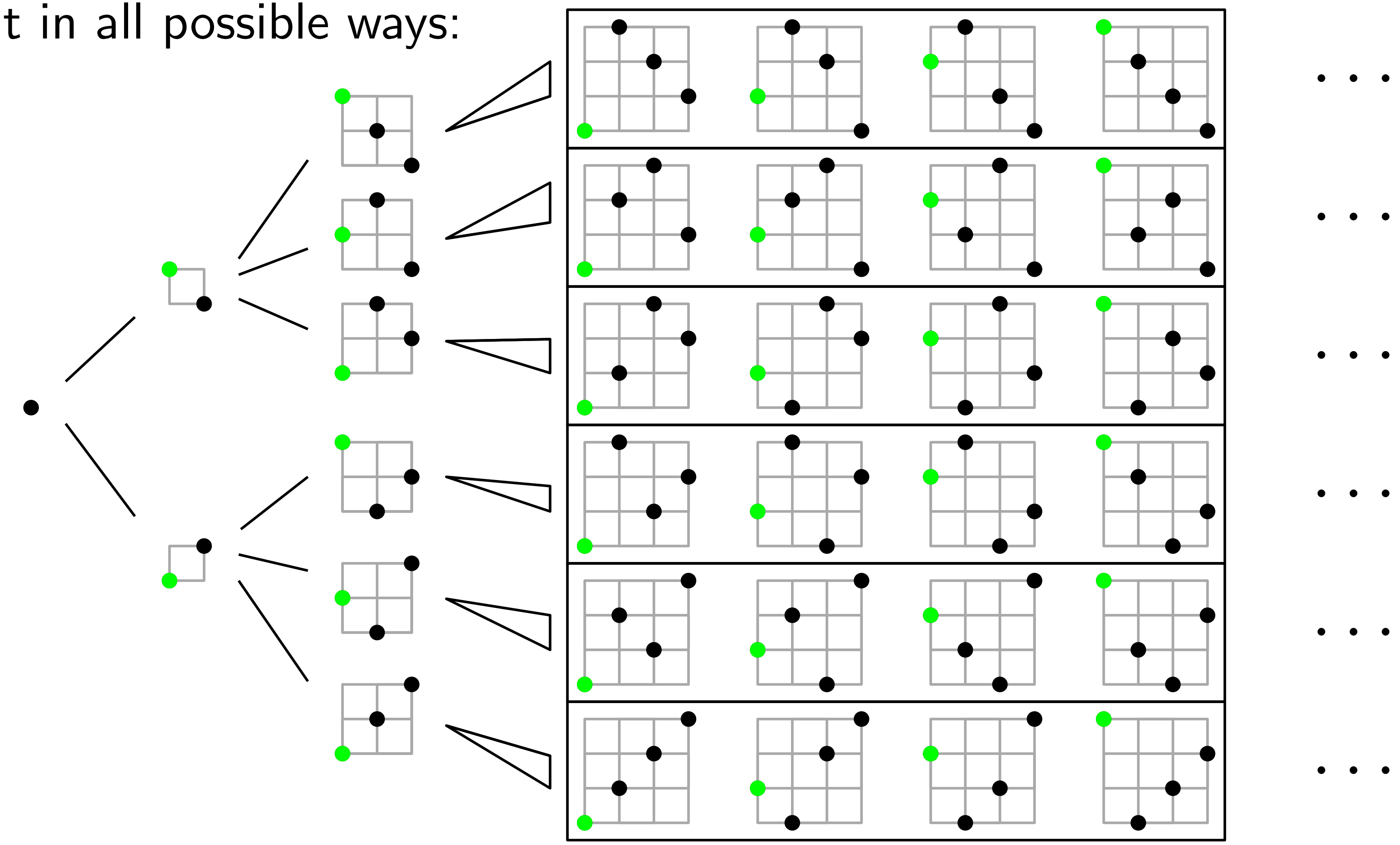
How to grow a permutation?

A standard way to generate permutations is by inserting a front point in all possible ways:



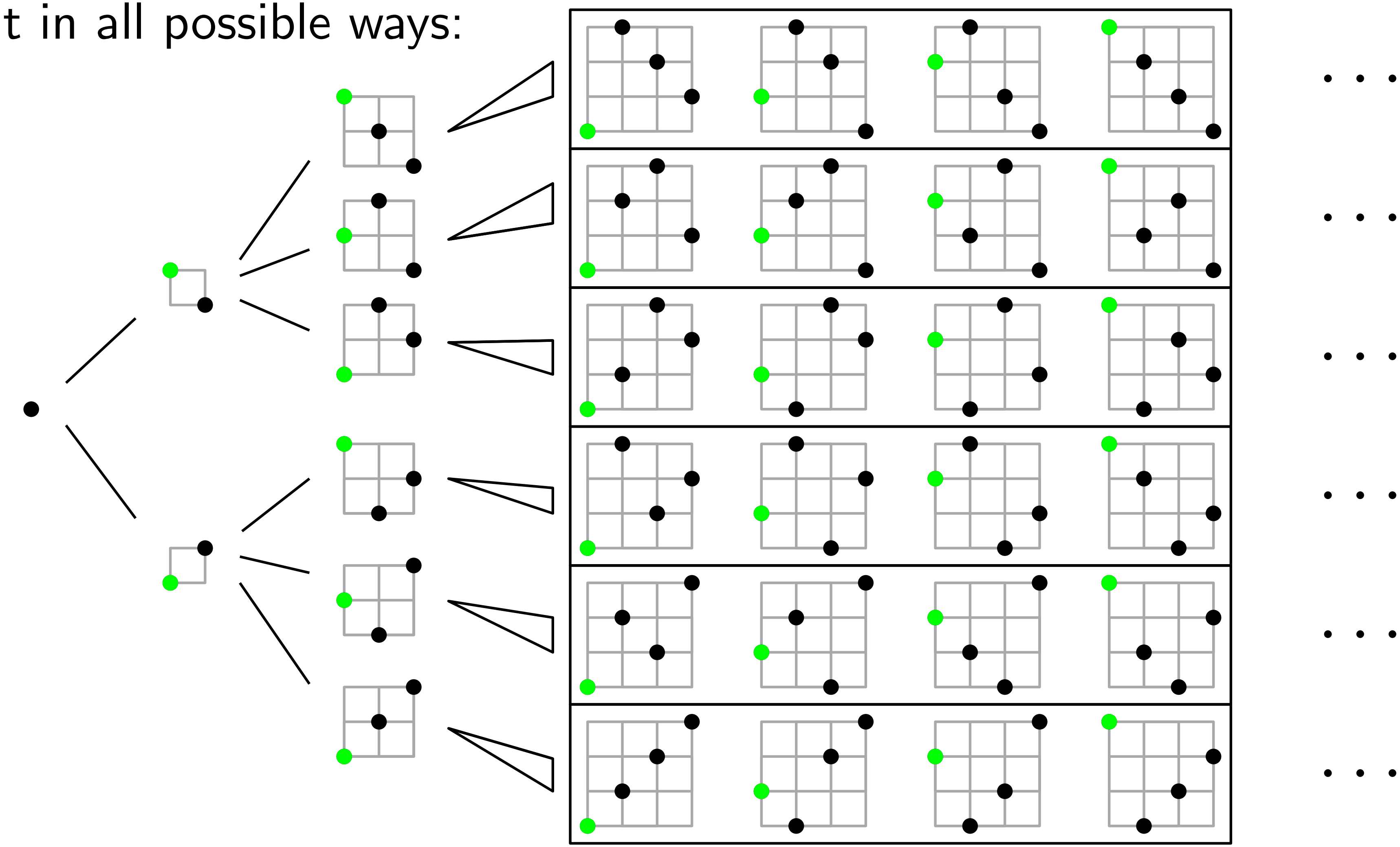
How to grow a permutation?

A standard way to generate permutations is by inserting a front point in all possible ways:



How to grow a permutation?

A standard way to generate permutations is by inserting a front point in all possible ways:



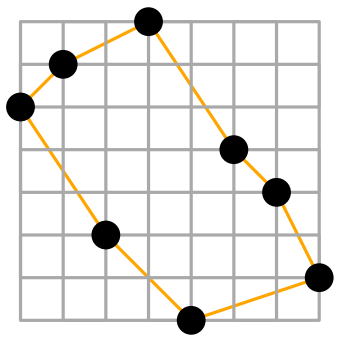
At the nth level of the generating tree, each vertex has $n+1$ children

$\Rightarrow$ $n!$ nodes at level n

$\Rightarrow$ $n!$ permutations of size n

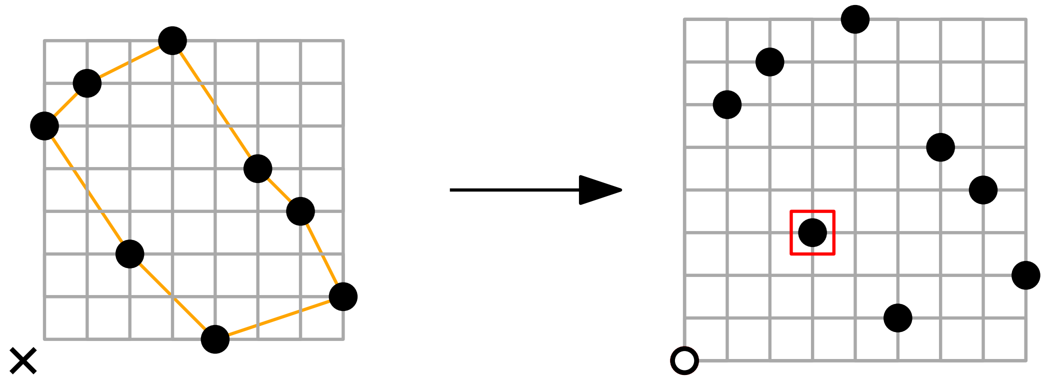
How to grow a **square** permutation?

Insert new points without creating internal points !



How to grow a **square** permutation?

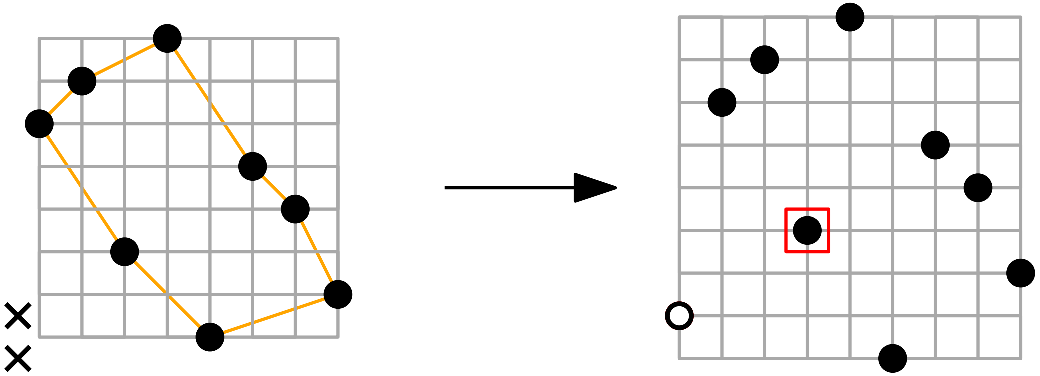
Insert new points without creating internal points !



No!

How to grow a **square** permutation?

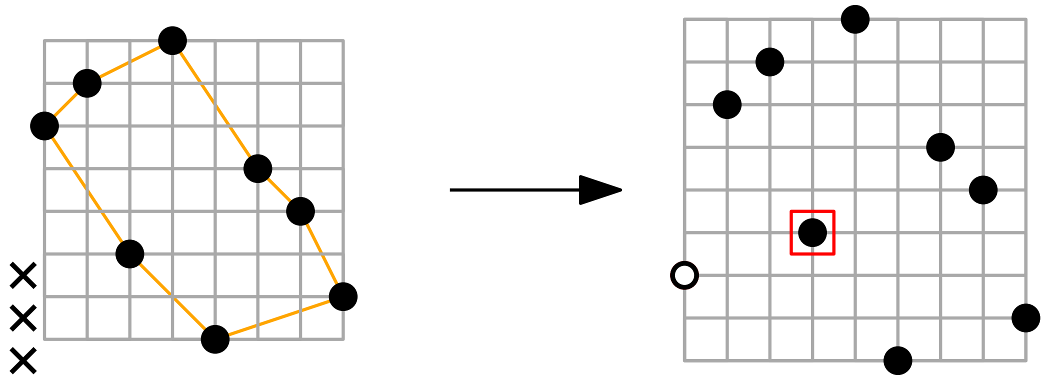
Insert new points without creating internal points !



No!

How to grow a **square** permutation?

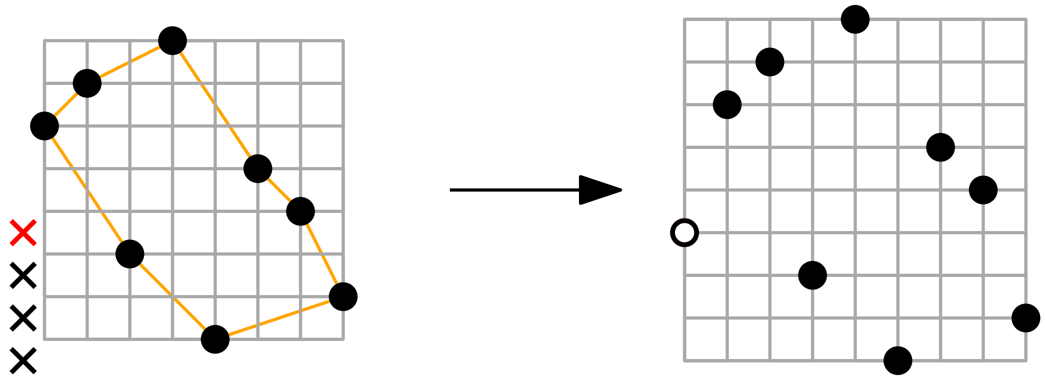
Insert new points without creating internal points !



No!

How to grow a **square** permutation?

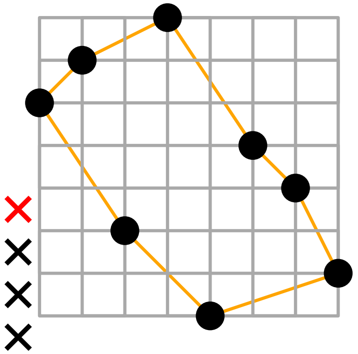
Insert new points without creating internal points !

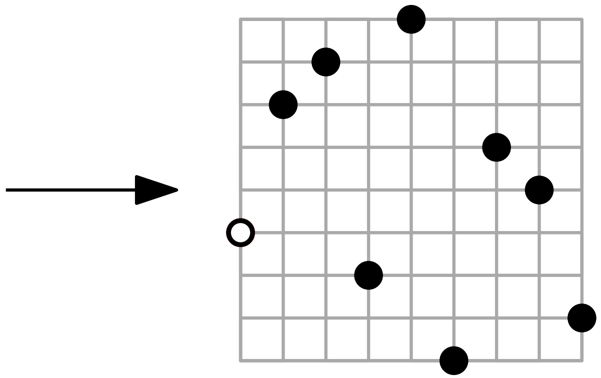


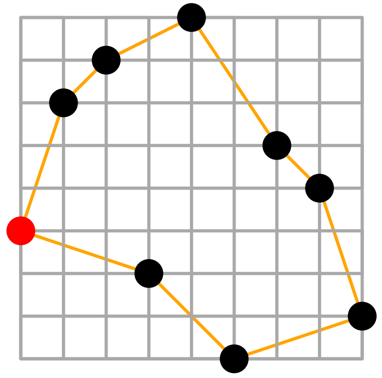
Yes!

How to grow a **square** permutation?

Insert new points without creating internal points !

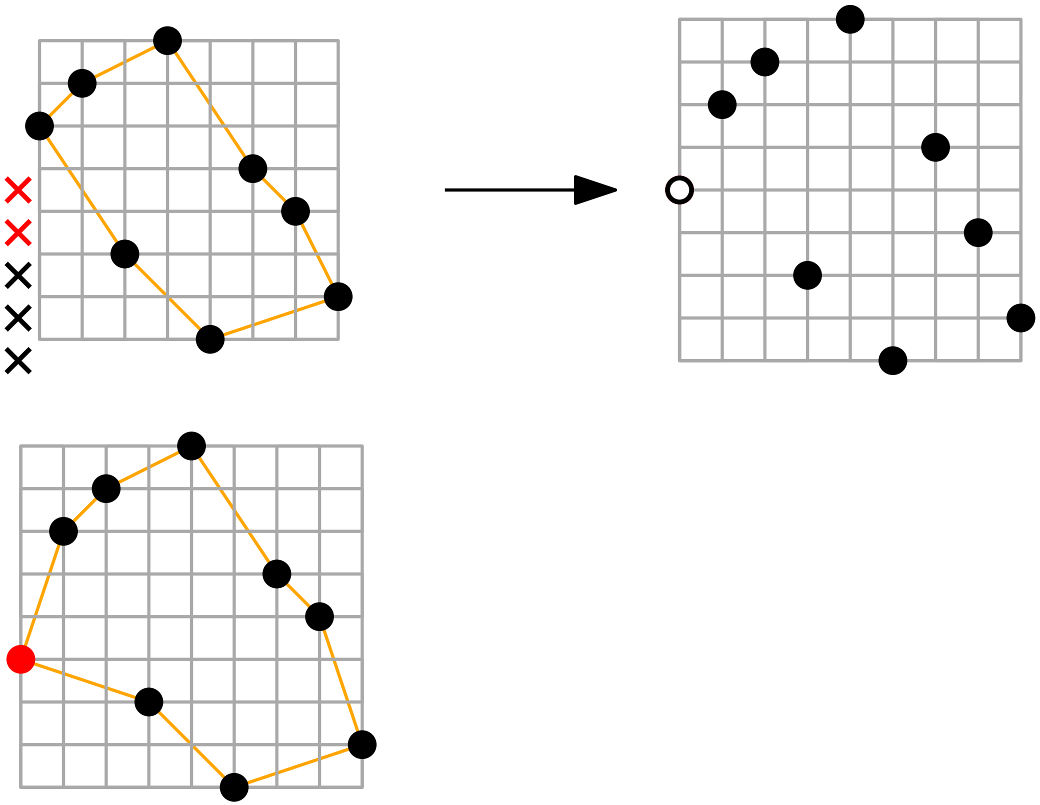






How to grow a **square** permutation?

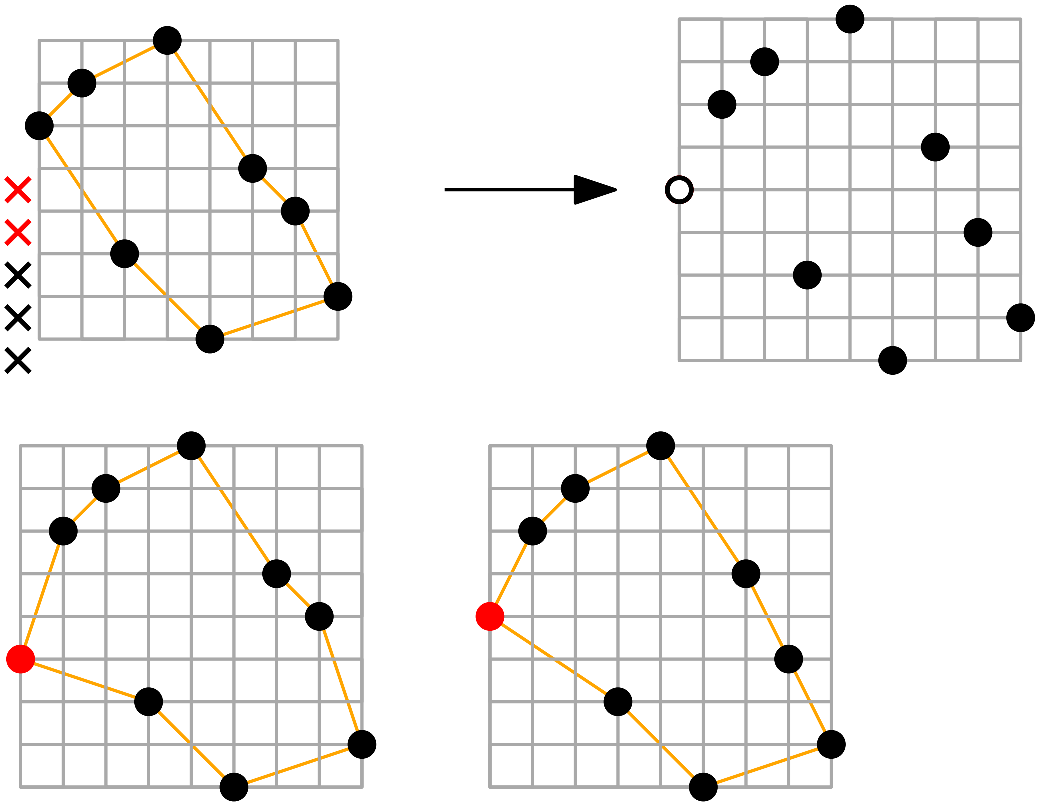
Insert new points without creating internal points !



Yes!

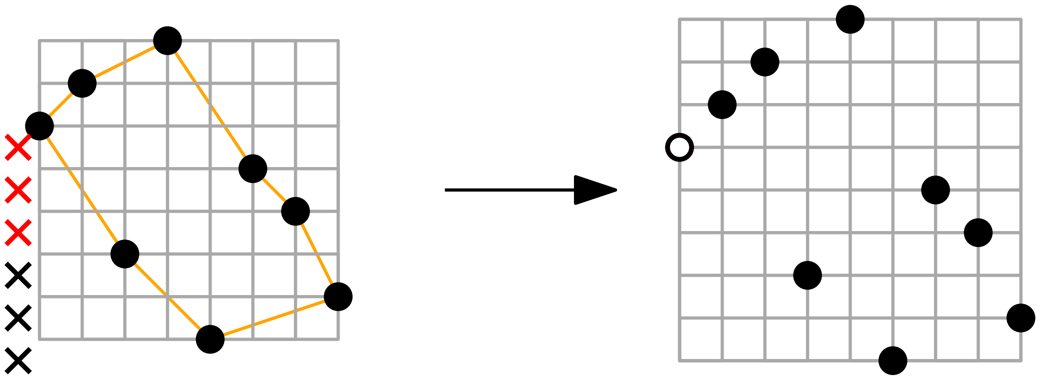
How to grow a **square** permutation?

Insert new points without creating internal points !

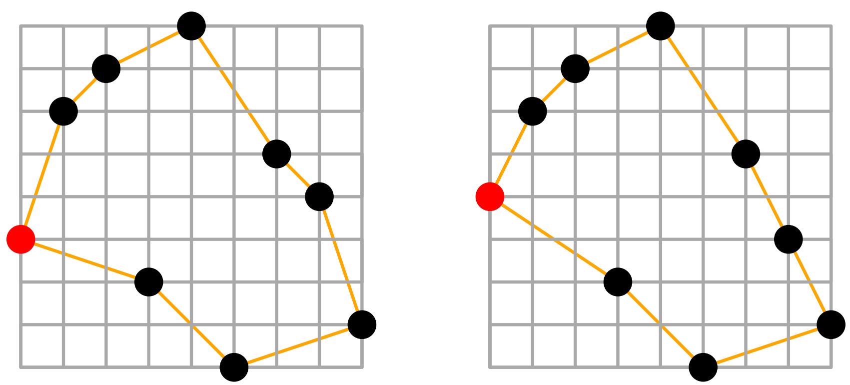


How to grow a **square** permutation?

Insert new points without creating internal points !

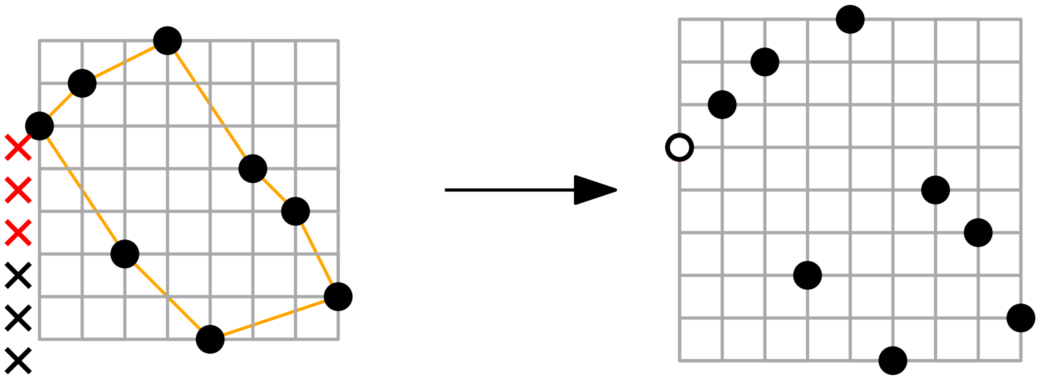


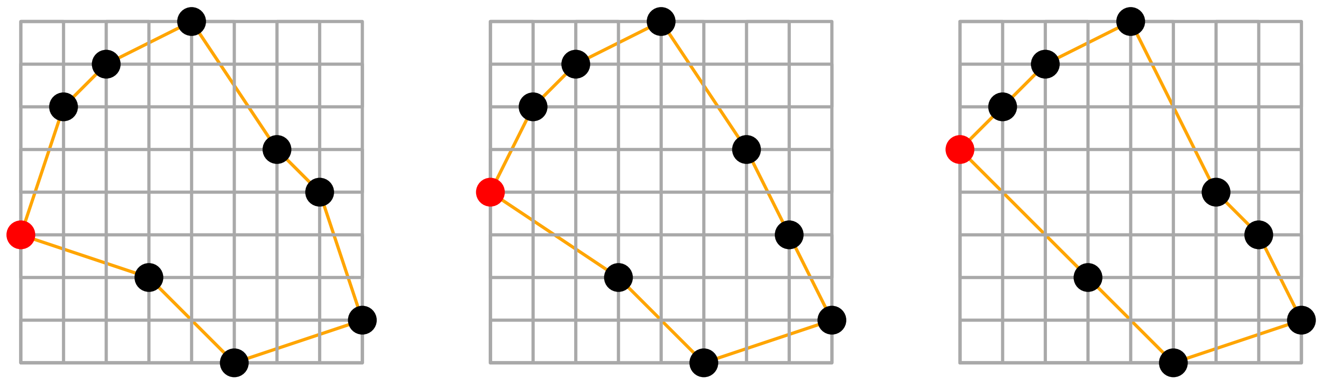
Yes!



How to grow a **square** permutation?

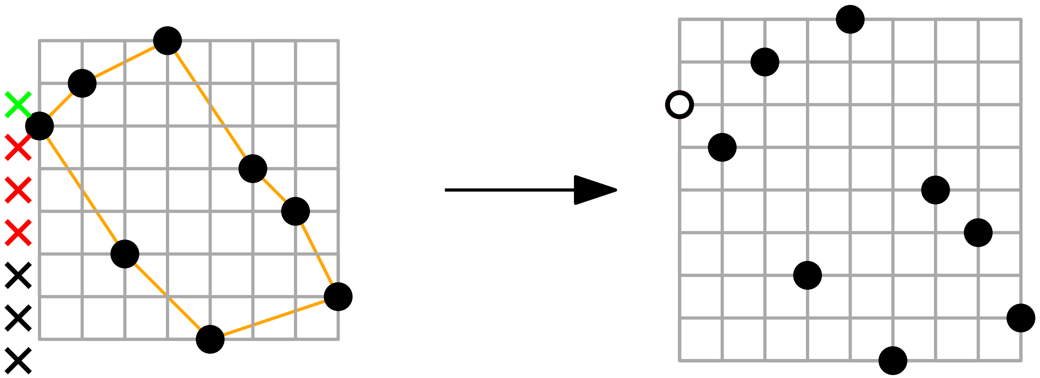
Insert new points without creating internal points !



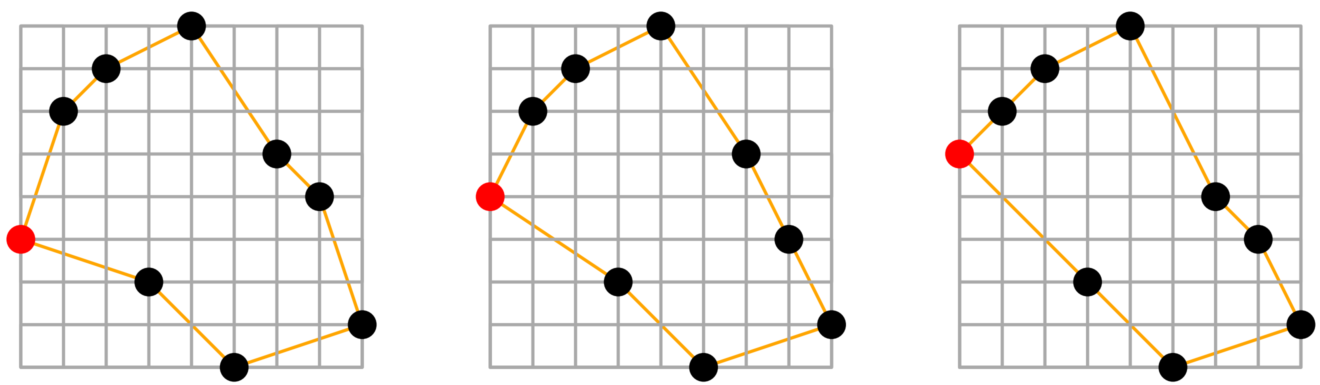


How to grow a **square** permutation?

Insert new points without creating internal points !

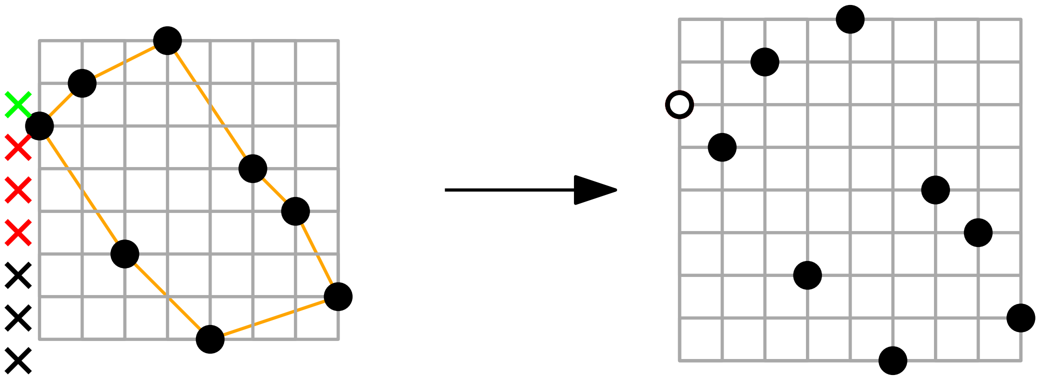


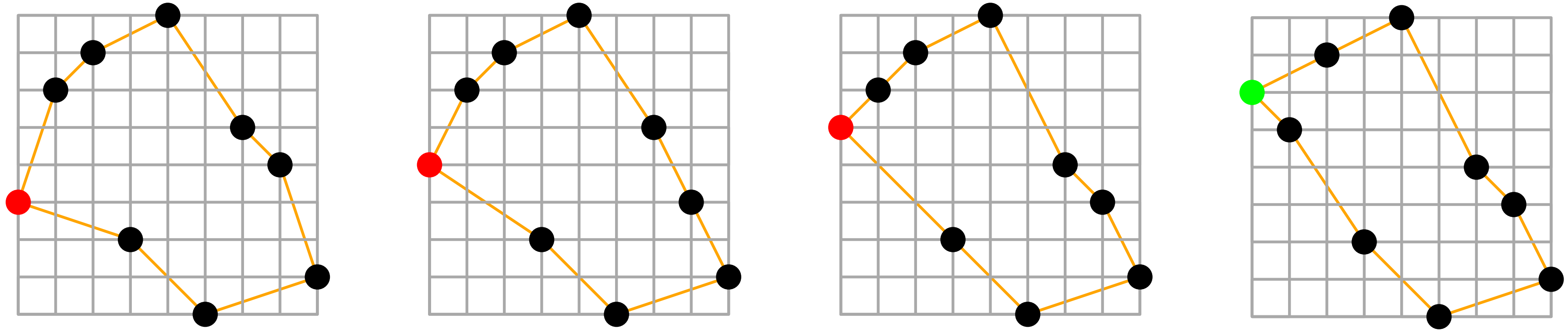
Yes!



How to grow a **square** permutation?

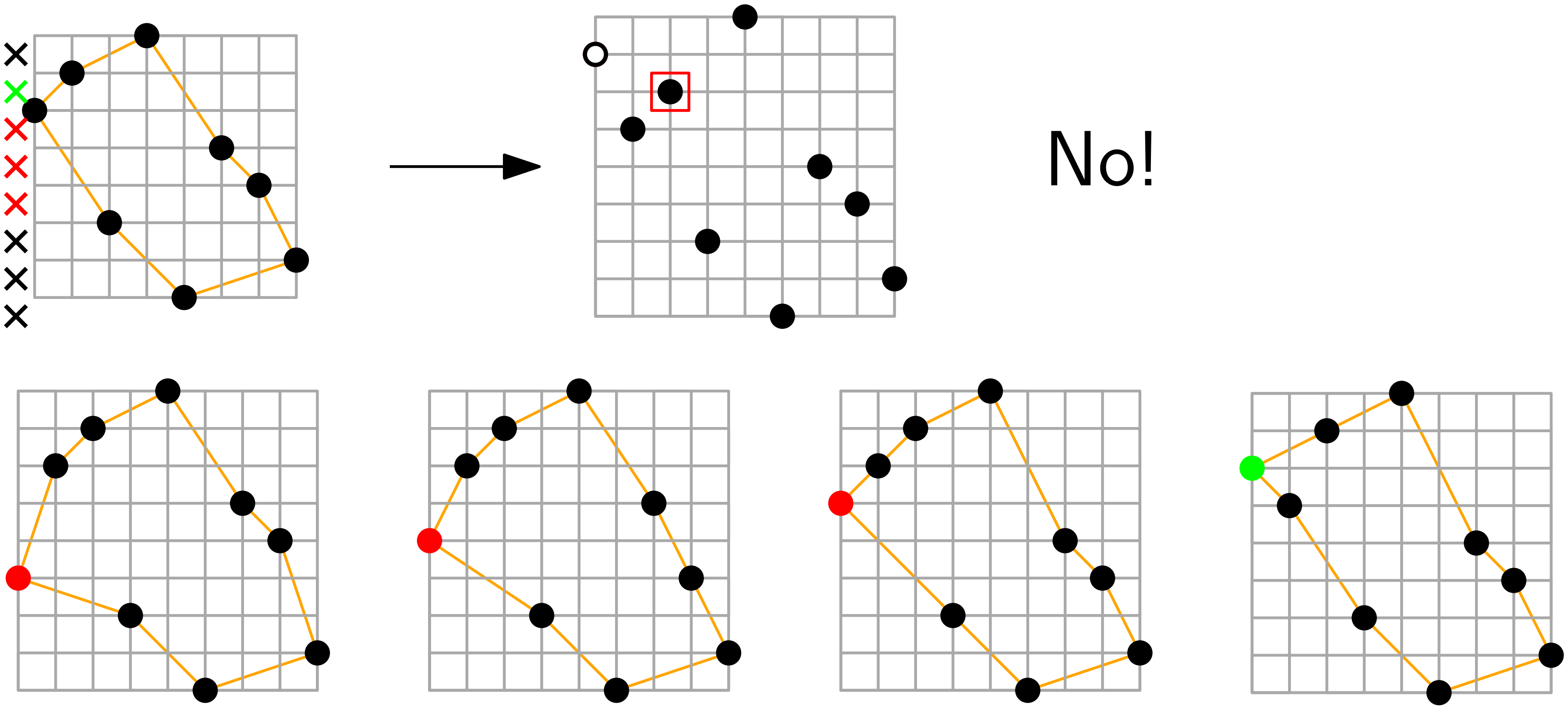
Insert new points without creating internal points !





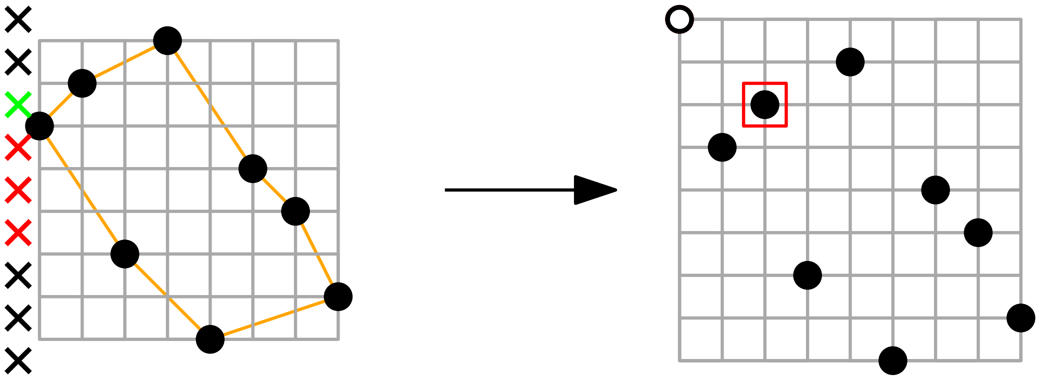
How to grow a **square** permutation?

Insert new points without creating internal points !

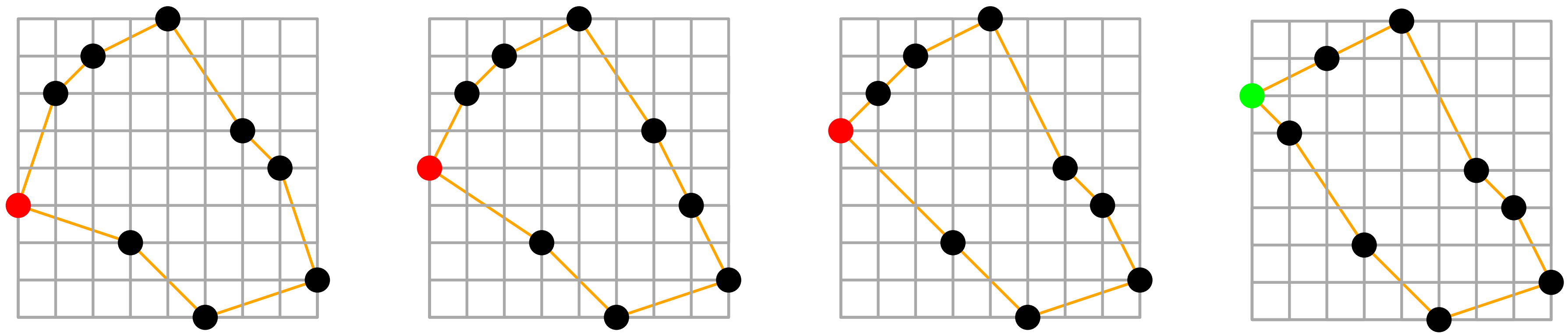


How to grow a **square** permutation?

Insert new points without creating internal points !

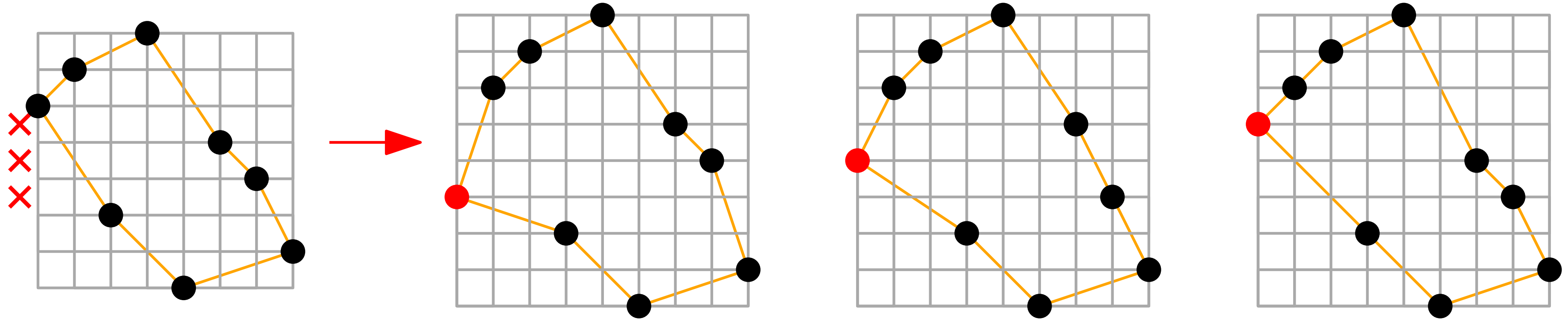


No!



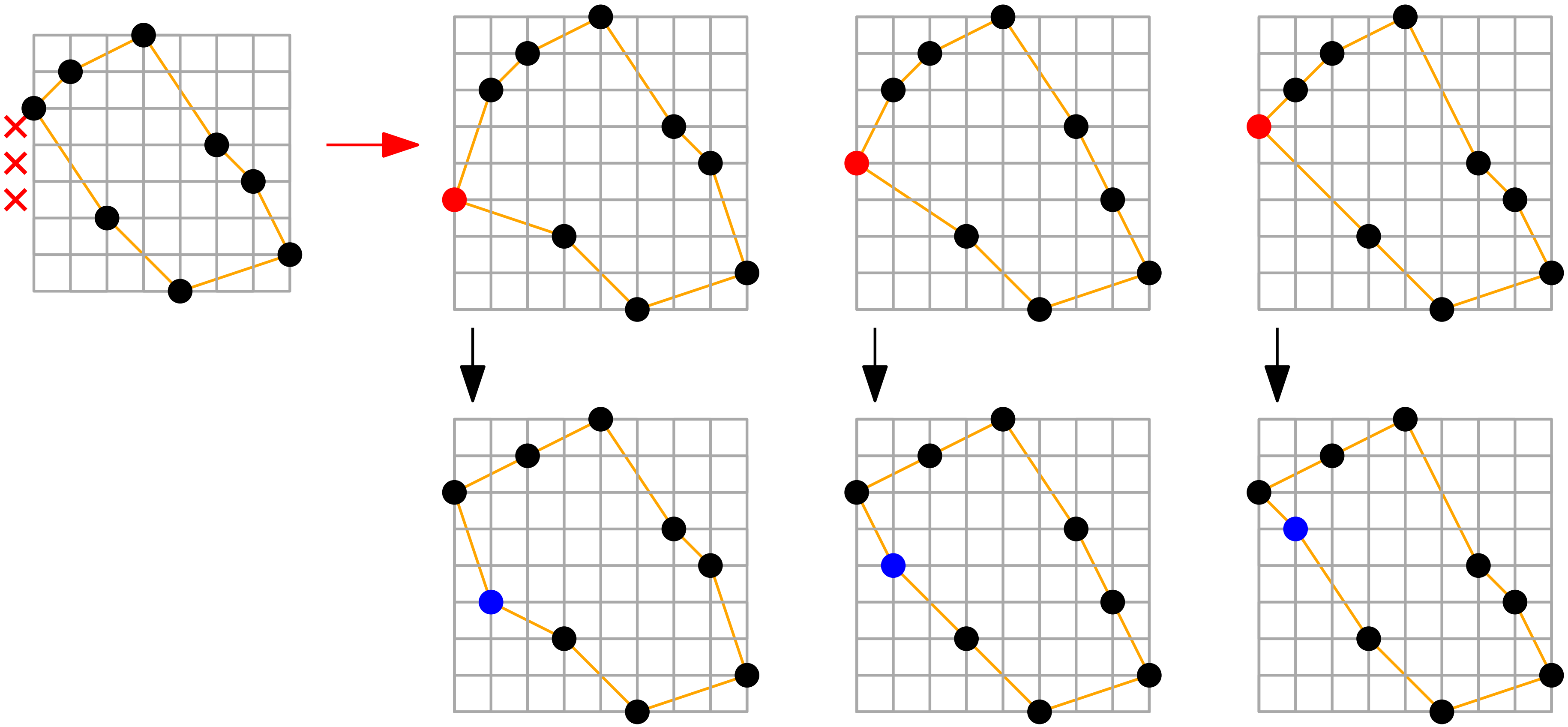
How to grow a square permutation?

The red **active sites**, $\times$, generate permutations σ' with $\sigma'(1) < \sigma'(2)$



How to grow a square permutation?

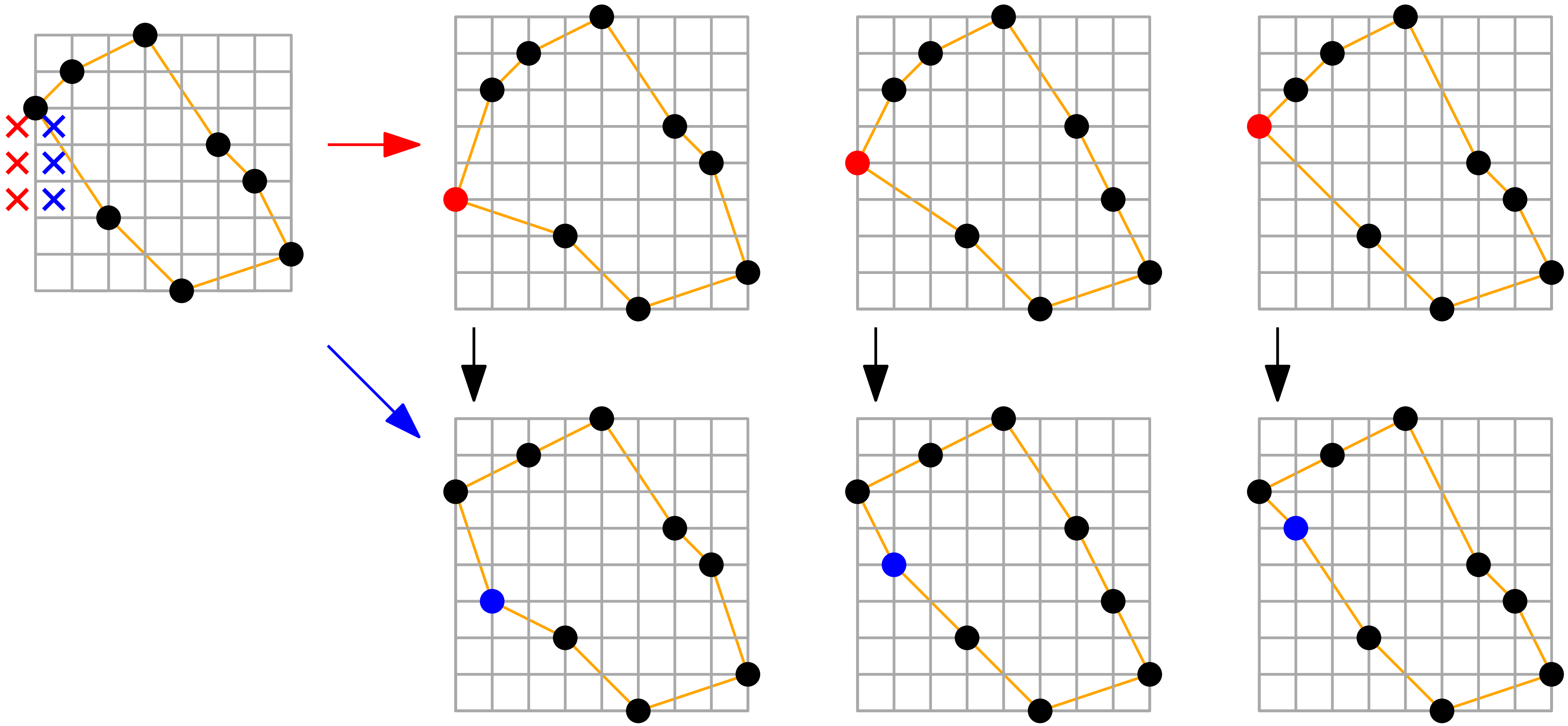
The red **active sites**, $\times$, generate permutations σ' with $\sigma'(1) < \sigma'(2)$



To generate σ' with $\sigma'(1) > \sigma'(2)$ we switch the first two column

How to grow a square permutation?

The red **active sites**, $\times$, generate permutations σ' with $\sigma'(1) < \sigma'(2)$



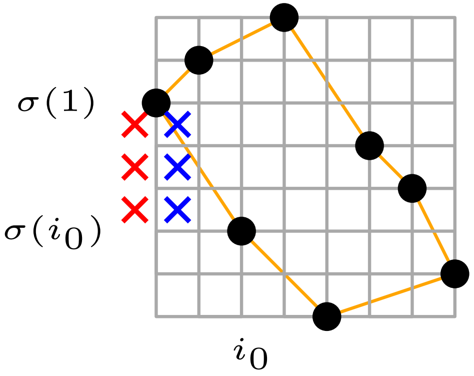
To generate σ' with $\sigma'(1) > \sigma'(2)$ we switch the first two column

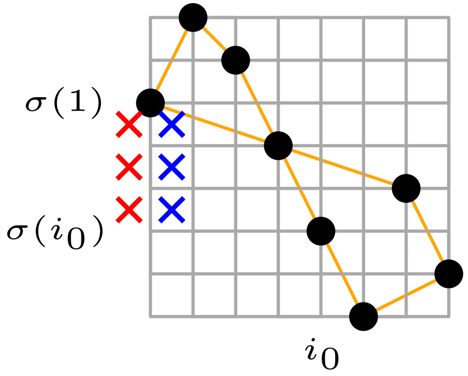
Equivalently we generate them by using blue active sites, $\times$

How to grow a square permutation?

Insertions are possible between $\sigma(1)$ and $\sigma(i_0)$ where

$(i_0, \sigma(i_0))$ is the leftmost lower point which is not a double point.



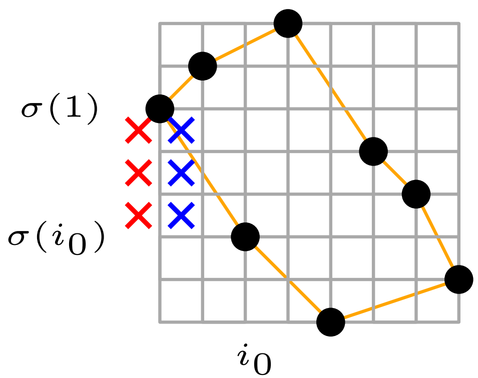


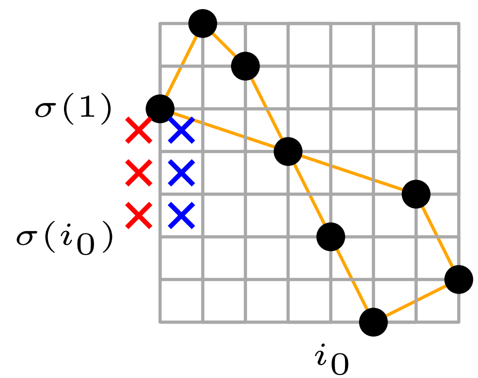
Indeed observe that double points do not block insertion.

How to grow a square permutation?

Insertions are possible between $\sigma(1)$ and $\sigma(i_0)$ where

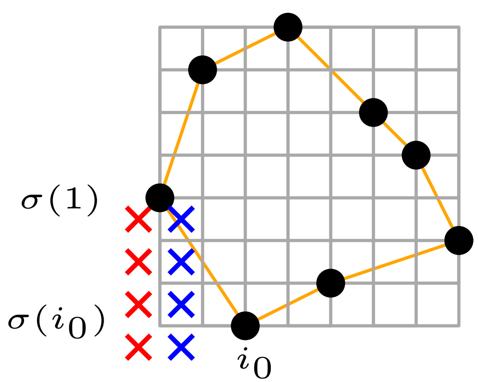
$(i_0, \sigma(i_0))$ is the leftmost lower point which is not a double point.

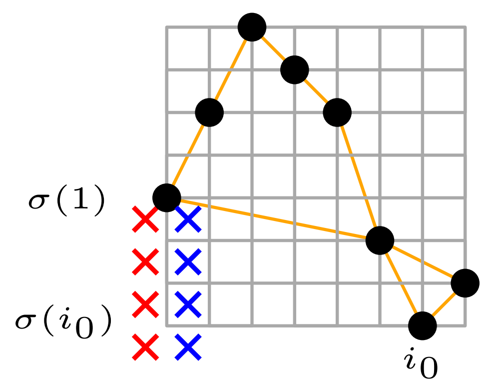




Indeed observe that double points do not block insertion.

When $\sigma(i_0) = 1$, insertion is also possible at the bottom row:





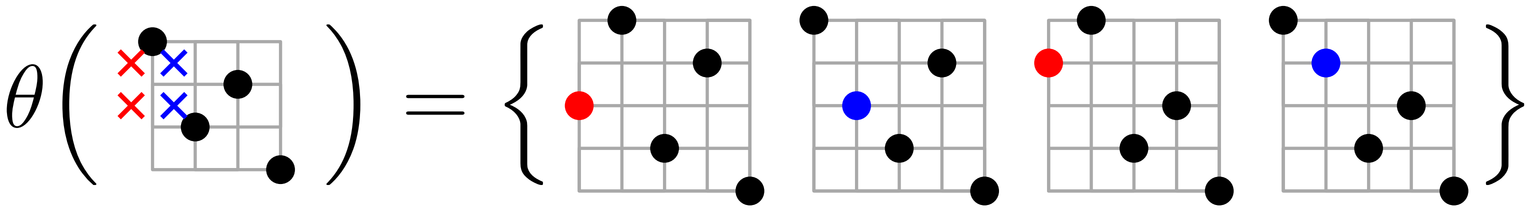
How to grow a square permutation?

Let the label $k(\sigma)$ of a permutation σ be its number of red active sites.

How to grow a square permutation?

Let the label $k(\sigma)$ of a permutation σ be its number of red active sites.

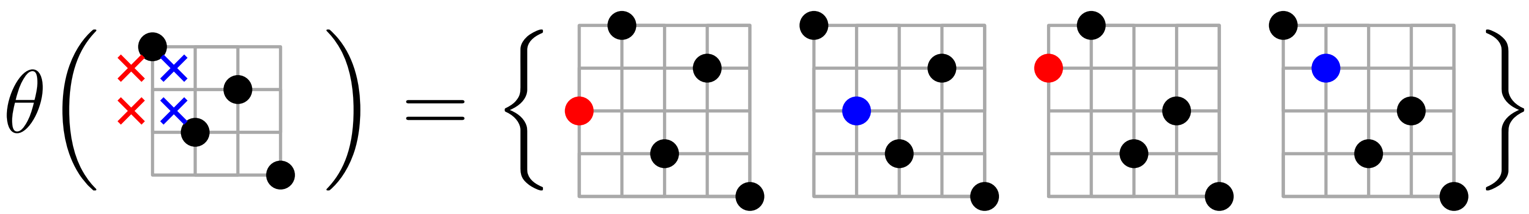
Moreover let $\theta(\sigma)$ denote the set of $2k(\sigma)$ permutations obtained from σ by insertion at an active site.



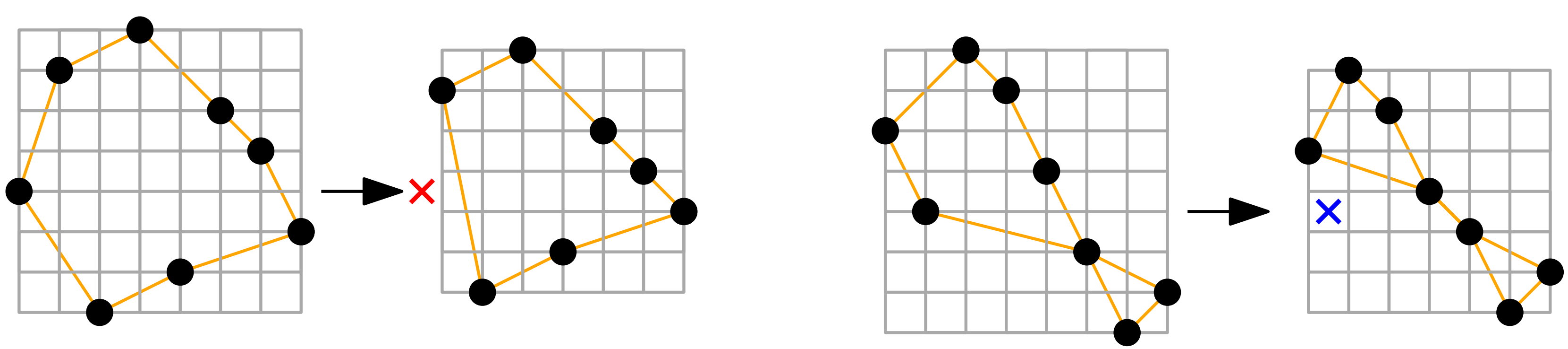
How to grow a square permutation?

Let the label $k(\sigma)$ of a permutation σ be its number of red active sites.

Moreover let $\theta(\sigma)$ denote the set of $2k(\sigma)$ permutations obtained from σ by insertion at an active site.



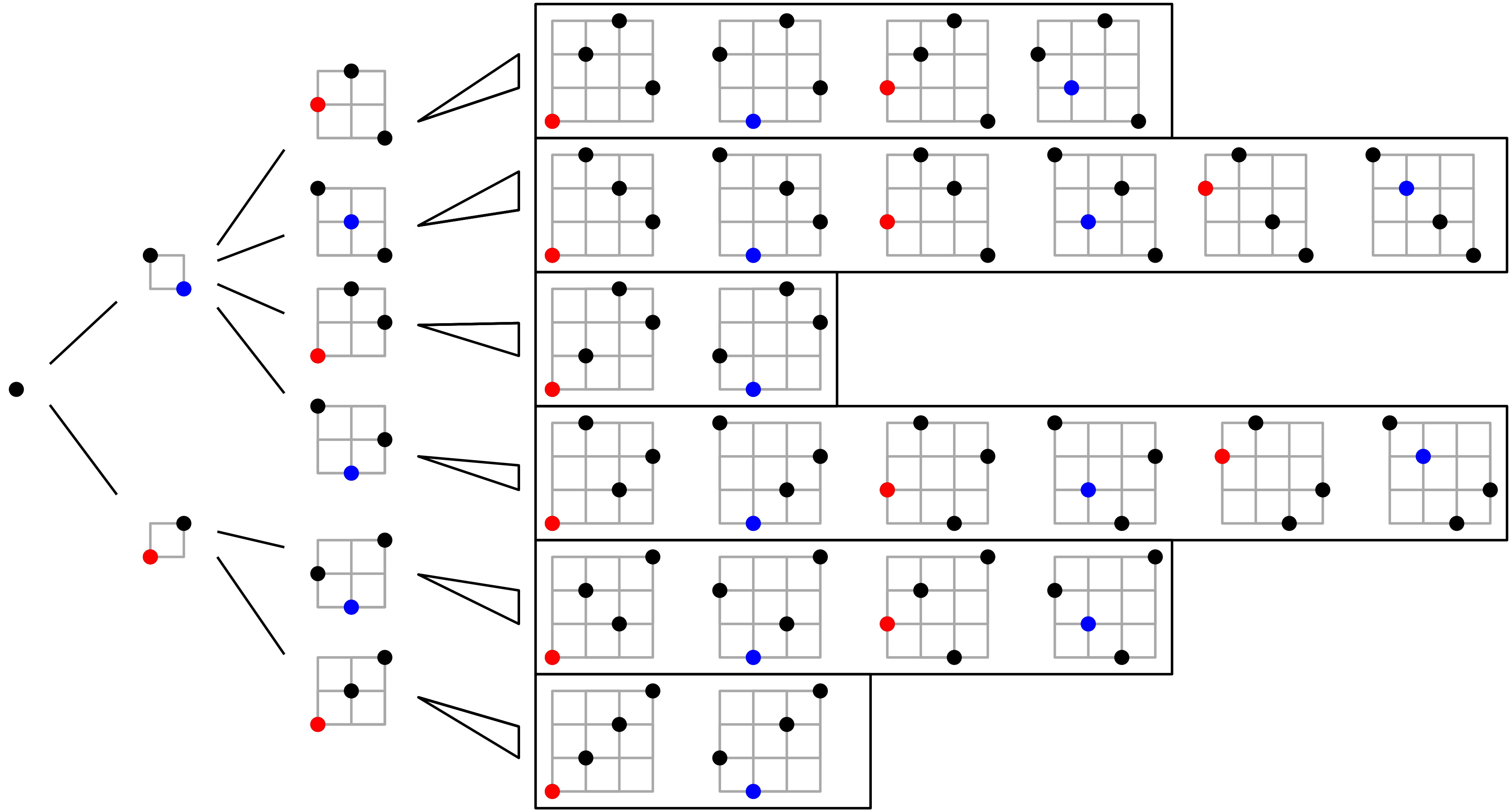
Proposition. For any square permutation σ' of size $n \geq 2$, there is a unique square permutation σ such that $\sigma' \in \theta(\sigma)$.



σ is obtained from σ' by removing the lowest among $\sigma(1)$ and $\sigma(2)$.

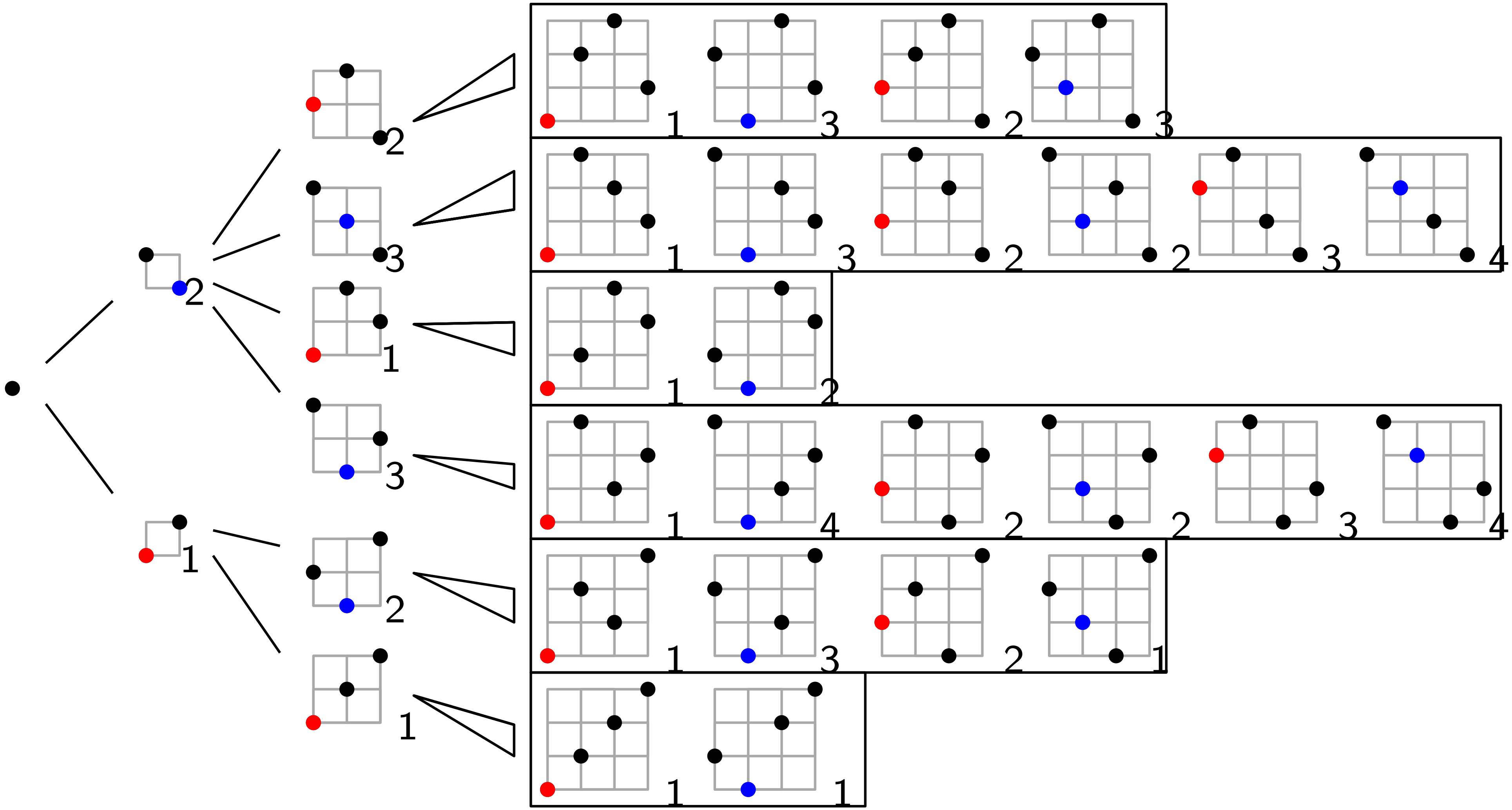
A generating tree for square permutations

Corollary. The mapping θ produces a generating tree for square permutations.



A generating tree for square permutations

Corollary. The mapping θ produces a generating tree for square permutations.



Each node with label k has $2k$ children.

But to count we need a more precise description of the shape of the tree

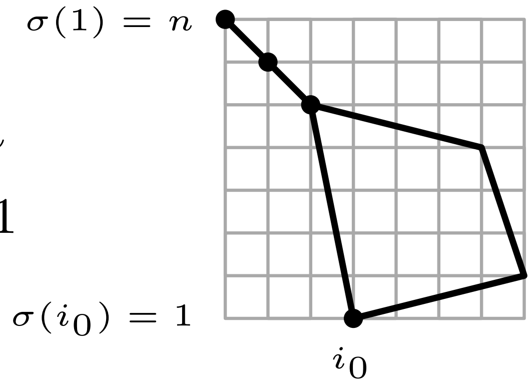
A growth rule

We classify permutations according to the value $\sigma(1)$ and $\sigma(i_0)$.

Type A:

$\sigma(1) = n$

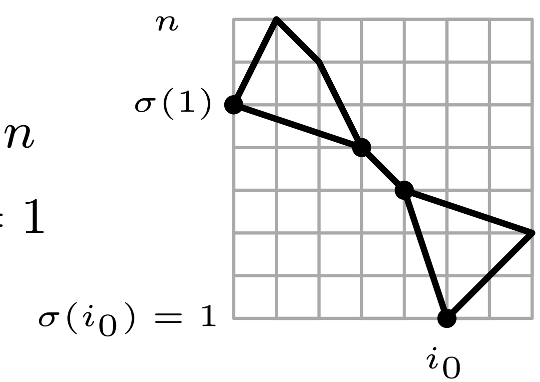
$\sigma(i_0) = 1$



Type B:

$\sigma(1) \neq n$

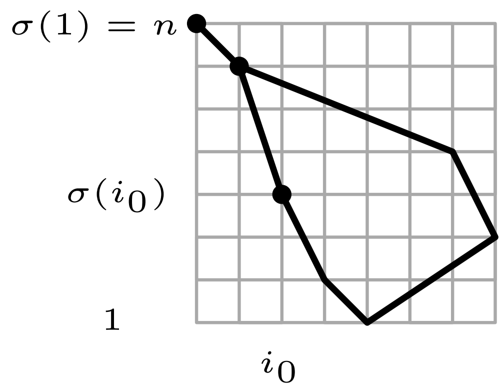
$\sigma(i_0) = 1$



Type C:

$\sigma(1) = n$

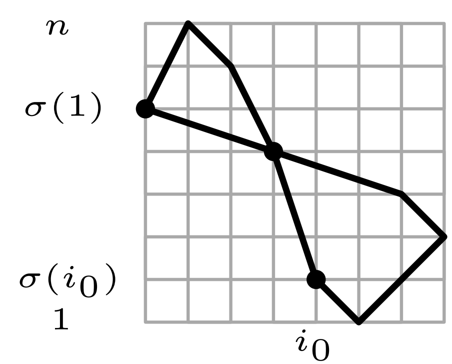
$\sigma(i_0) \neq 1$



Type D:

$\sigma(1) \neq n$

$\sigma(i_0) \neq 1$



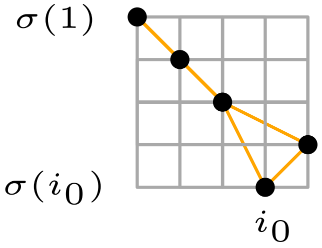
A growth rule

Describe the type of the children of a permutation with label k:

Type A:

$\sigma(1) = n$

$\sigma(i_0) = 1$



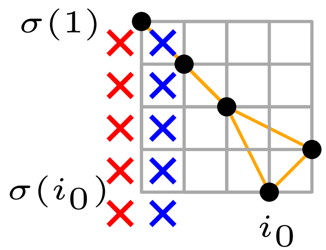
A growth rule

Describe the type of the children of a permutation with label k:

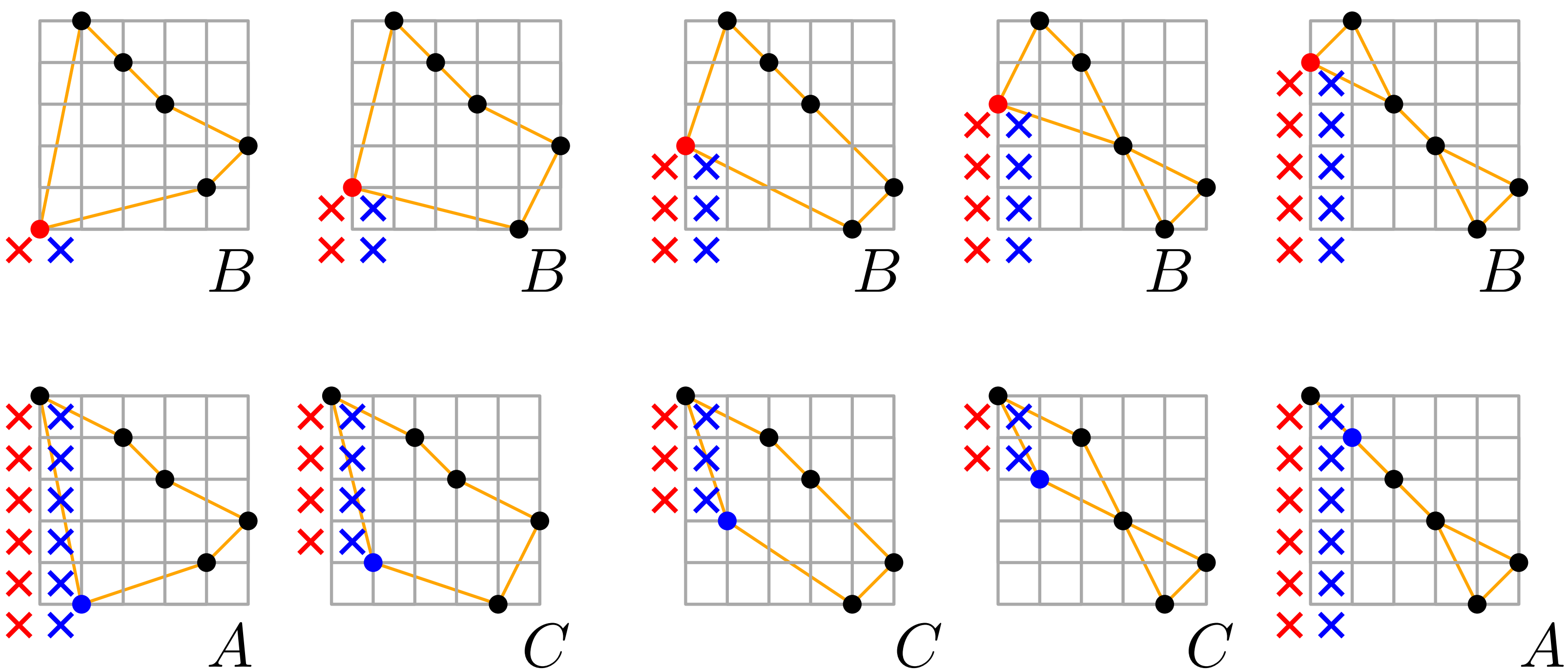
Type A:

$\sigma(1) = n$

$\sigma(i_0) = 1$



$$(k)_A \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\ (k+1)_A(k-1)_C\dots(2)_C(k+1)_A\end{array}$$



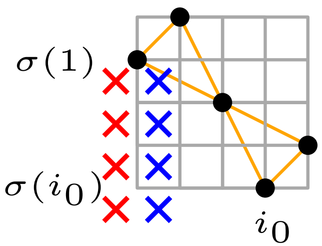
A growth rule

Describe the type of the children of a permutation with label k:

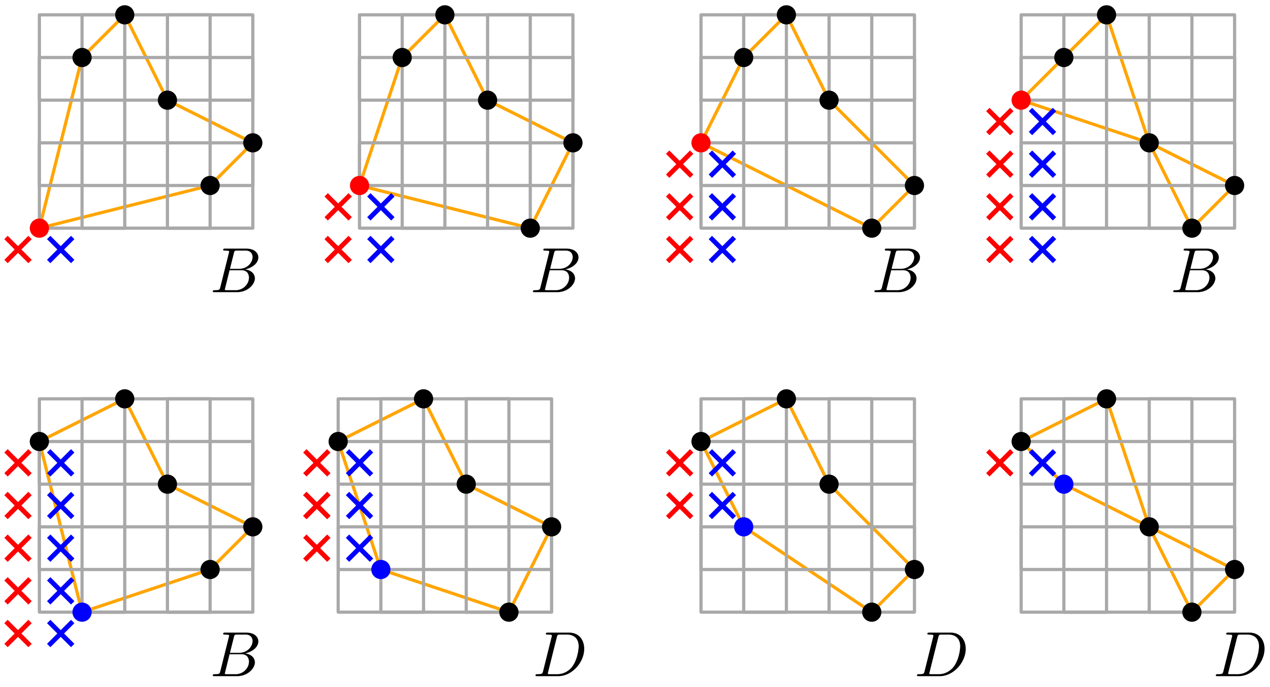
Type B:

$\sigma(1) \neq n$

$\sigma(i_0) = 1$



$$(k)_B \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B \dots (k-1)_B(k)_B \\ (k+1)_B(k-1)_D \dots (2)_D(1)_D\end{array}$$



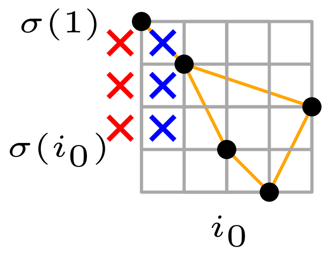
A growth rule

Describe the type of the children of a permutation with label k:

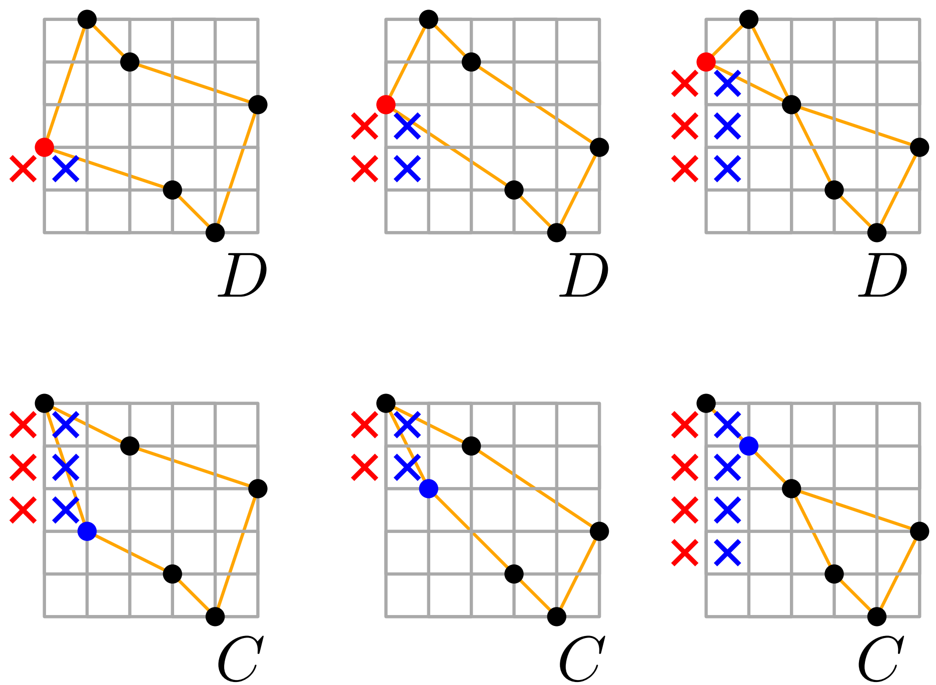
Type C:

$$\sigma(1) = n$$

$$\sigma(i_0) \neq 1$$



$$(k)_C \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D \dots (k-1)_D(k)_D \\ (k)_C \dots (3)_C(2)_C(k+1)_C\end{array}$$



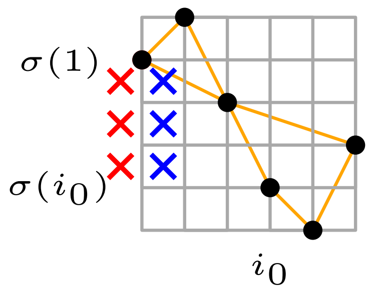
A growth rule

Describe the type of the children of a permutation with label k:

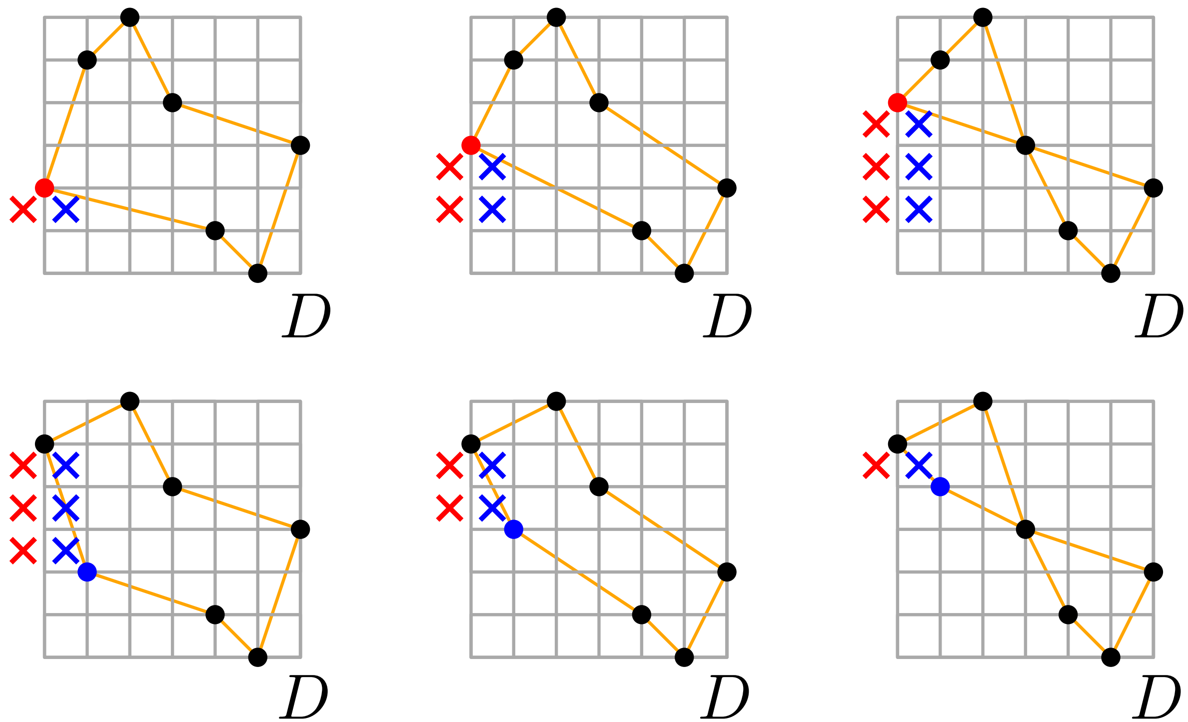
Type D:

$\sigma(1) \neq n$

$\sigma(i_0) \neq 1$



$$(k)_D \xrightarrow{\theta} \begin{array}{l}(1)_D(2)_D\ldots(k-1)_D(k)_D\\(k)_D(k-1)_D\ldots(2)_D(1)_D\end{array}$$



Equations for generating functions

- The permutation • has two children with respective label $(2)_A$ and $(1)_B$

$$(k)_A \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\(k+1)_A(k-1)_C\dots(2)_C(k+1)_A\end{array} \quad (k)_C \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\(k)_C\dots(3)_C(2)_C(k+1)_C\end{array}$$

$$(k)_B \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\(k+1)_B(k-1)_D\dots(2)_D(1)_D\end{array} \quad (k)_D \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\(k)_D(k-1)_D\dots(2)_D(1)_D\end{array}$$

Equations for generating functions

- The permutation $\bullet$ has two children with respective label $(2)_A$ and $(1)_B$

$$(k)_A \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\(k+1)_A(k-1)_C\dots(2)_C(k+1)_A\end{array} \qquad (k)_C \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\(k)_C\dots(3)_C(2)_C(k+1)_C\end{array}$$

$$(k)_B \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\(k+1)_B(k-1)_D\dots(2)_D(1)_D\end{array} \qquad (k)_D \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\(k)_D(k-1)_D\dots(2)_D(1)_D\end{array}$$

These growth rules induce equations for generating functions:

$$F_A(u) \equiv F_A(u;t) = \sum_{\sigma\in A} t^{|\sigma|}u^{k(\sigma)} = t^2u^2 + \sum_{\pi\in S\setminus\{\bullet\}} \sum_{\sigma\in\theta(\pi)\cap A} t^{|\sigma|}u^{k(\sigma)}$$

Equations for generating functions

- The permutation • has two children with respective label $(2)_A$ and $(1)_B$

$$(k)_A \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\(k+1)_A(k-1)_C\dots(2)_C(k+1)_A\end{array} \quad (k)_C \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\(k)_C\dots(3)_C(2)_C(k+1)_C\end{array}$$

$$(k)_B \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\(k+1)_B(k-1)_D\dots(2)_D(1)_D\end{array} \quad (k)_D \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\(k)_D(k-1)_D\dots(2)_D(1)_D\end{array}$$

These growth rules induce equations for generating functions:

$$F_A(u) \equiv F_A(u;t) = \sum_{\sigma\in A} t^{|\sigma|}u^{k(\sigma)} = \underline{t^2u^2} + \sum_{\pi\in S\setminus\{\bullet\}}\ \sum_{\sigma\in\theta(\pi)\cap A} t^{|\sigma|}u^{k(\sigma)}$$

Equations for generating functions

- The permutation • has two children with respective label $(2)_A$ and $(1)_B$

$$(k)_A \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\ \boxed{(k+1)_A}(k-1)_C\dots(2)_C\boxed{(k+1)_A}\end{array} \qquad (k)_C \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\ (k)_C\dots(3)_C(2)_C(k+1)_C\end{array}$$

$$(k)_B \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\ (k+1)_B(k-1)_D\dots(2)_D(1)_D\end{array} \qquad (k)_D \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\ (k)_D(k-1)_D\dots(2)_D(1)_D\end{array}$$

These growth rules induce equations for generating functions:

$$F_A(u) \equiv F_A(u;t) = \sum_{\sigma\in A} t^{|\sigma|}u^{k(\sigma)} = t^2u^2 + \sum_{\pi\in S\setminus\{\bullet\}}\ \sum_{\sigma\in\theta(\pi)\cap A} t^{|\sigma|}u^{k(\sigma)}$$

$$= t^2u^2 + \sum_{\pi\in A} t^{|\pi|+1}\left(\underline{u^{k(\pi)+1}} + \underline{u^{k(\pi)+1}}\right)$$

Equations for generating functions

- The permutation $\bullet$ has two children with respective label $(2)_A$ and $(1)_B$

$$(k)_A \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\ \boxed{(k+1)_A}(k-1)_C\dots(2)_C\boxed{(k+1)_A}\end{array} \qquad (k)_C \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\ (k)_C\dots(3)_C(2)_C(k+1)_C\end{array}$$

$$(k)_B \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\ (k+1)_B(k-1)_D\dots(2)_D(1)_D\end{array} \qquad (k)_D \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\ (k)_D(k-1)_D\dots(2)_D(1)_D\end{array}$$

These growth rules induce equations for generating functions:

$$F_A(u) \equiv F_A(u;t) = \sum_{\sigma\in A} t^{|\sigma|}u^{k(\sigma)} = t^2u^2 + \sum_{\pi\in S\setminus\{\bullet\}} \sum_{\sigma\in\theta(\pi)\cap A} t^{|\sigma|}u^{k(\sigma)}$$

$$= t^2u^2 + \sum_{\pi\in A} t^{|\pi|+1}\left(\underline{u^{k(\pi)+1}} + \underline{u^{k(\pi)+1}}\right)$$

$$= t^2u^2 + 2tuF_A(u)$$

Equations for generating functions

- The permutation • has two children with respective label $(2)_A$ and $(1)_B$

$$(k)_A \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\ldots(k-1)_B(k)_B\\(k+1)_A(k-1)_C\ldots(2)_C(k+1)_A\end{array} \qquad (k)_C \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\ldots(k-1)_D(k)_D\\(k)_C\ldots(3)_C(2)_C(k+1)_C\end{array}$$

$$(k)_B \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\ldots(k-1)_B(k)_B\\(k+1)_B(k-1)_D\ldots(2)_D(1)_D\end{array} \qquad (k)_D \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\ldots(k-1)_D(k)_D\\(k)_D(k-1)_D\ldots(2)_D(1)_D\end{array}$$

$$F_A(u) = \sum_{\sigma\in A} t^{|\sigma|}u^{k(\sigma)} = t^2u^2 + 2tuF_A(u)$$

Equations for generating functions

- The permutation • has two children with respective label $(2)_A$ and $(1)_B$

$$(k)_A \xrightarrow{\theta} \begin{matrix}(1)_B(2)_B\dots(k-1)_B(k)_B\\(k+1)_A(k-1)_C\dots(2)_C(k+1)_A\end{matrix} \qquad (k)_C \xrightarrow{\theta} \begin{matrix}(1)_D(2)_D\dots(k-1)_D(k)_D\\(k)_C\dots(3)_C(2)_C(k+1)_C\end{matrix}$$

$$(k)_B \xrightarrow{\theta} \begin{matrix}(1)_B(2)_B\dots(k-1)_B(k)_B\\(k+1)_B(k-1)_D\dots(2)_D(1)_D\end{matrix} \qquad (k)_D \xrightarrow{\theta} \begin{matrix}(1)_D(2)_D\dots(k-1)_D(k)_D\\(k)_D(k-1)_D\dots(2)_D(1)_D\end{matrix}$$

$$F_A(u) = \sum_{\sigma\in A} t^{|\sigma|}u^{k(\sigma)} = t^2u^2 + 2tuF_A(u)$$

$$F_B(u) = \sum_{\sigma\in B} t^{|\sigma|}u^{k(\sigma)} = t^2u + \sum_{\pi\in S\setminus\{\bullet\}}\ \sum_{\sigma\in\theta(\pi)\cap B} t^{|\sigma|}u^{k(\sigma)}$$

$$= t^2u + \sum_{\pi\in A} t^{|\pi|+1}\left(u+u^2+\dots+u^{k(\pi)}\right) + \sum_{\pi\in B} t^{|\pi|+1}\left(u+u^2+\dots+u^{k(\pi)}+u^{k(\pi)+1}\right)$$

$$= t^2u + t\sum_{\pi\in A} t^{|\pi|}\,\frac{u-u^{k(\pi)+1}}{1-u} + tu\sum_{\pi\in B} t^{|\pi|}\,\frac{u-u^{k(\pi)+2}}{1-u}$$

$$= t^2u + t\,\frac{uF_A(1)-uF_A(u)}{1-u} + t\,\frac{uF_B(1)-u^2F_B(u)}{1-u}$$

Equations for generating functions

- The permutation • has two children with respective label $(2)_A$ and $(1)_B$

$$(k)_A \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\(k+1)_A(k-1)_C\dots(2)_C(k+1)_A\end{array} \qquad (k)_C \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\(k)_C\dots(3)_C(2)_C(k+1)_C\end{array}$$

$$(k)_B \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\(k+1)_B(k-1)_D\dots(2)_D(1)_D\end{array} \qquad (k)_D \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\(k)_D(k-1)_D\dots(2)_D(1)_D\end{array}$$

$$F_A(u) = \textstyle\sum_{\sigma\in A} t^{|\sigma|}u^{k(\sigma)} = t^2u^2 + 2tuF_A(u)$$

$$F_B(u) = t^2u + t\frac{uF_A(1) - uF_A(u)}{1-u} + t\frac{uF_B(1) - u^2F_B(u)}{1-u}$$

Equations for generating functions

- The permutation • has two children with respective label $(2)_A$ and $(1)_B$

$$(k)_A \xrightarrow{\theta} \begin{matrix} (1)_B(2)_B \dots (k-1)_B(k)_B \\ (k+1)_A \boxed{(k-1)_C \dots (2)_C} (k+1)_A \end{matrix} \qquad (k)_C \xrightarrow{\theta} \begin{matrix} (1)_D(2)_D \dots (k-1)_D(k)_D \\ \boxed{(k)_C \dots (3)_C(2)_C(k+1)_C} \end{matrix}$$

$$(k)_B \xrightarrow{\theta} \begin{matrix} (1)_B(2)_B \dots (k-1)_B(k)_B \\ (k+1)_B(k-1)_D \dots (2)_D(1)_D \end{matrix} \qquad (k)_D \xrightarrow{\theta} \begin{matrix} (1)_D(2)_D \dots (k-1)_D(k)_D \\ (k)_D(k-1)_D \dots (2)_D(1)_D \end{matrix}$$

$$F_A(u) = \textstyle\sum_{\sigma \in A} t^{|\sigma|} u^{k(\sigma)} = t^2u^2 + 2tuF_A(u)$$

$$F_B(u) = t^2u + t\frac{uF_A(1) - uF_A(u)}{1-u} + t\frac{uF_B(1) - u^2F_B(u)}{1-u}$$

$$F_C(u) = \sum_{\sigma \in C} t^{|\sigma|} u^{k(\sigma)} = \sum_{\pi \in S \setminus \{\bullet\}} \sum_{\sigma \in \theta(\pi) \cap C} t^{|\sigma|} u^{k(\sigma)}$$

$$= t\frac{u^2F_A(1) - F_A(u)}{1-u} + t\frac{u^2F_C(1) - u^2F_C(u)}{1-u}$$

Equations for generating functions

- The permutation • has two children with respective label $(2)_A$ and $(1)_B$

$$(k)_A \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\(k+1)_A(k-1)_C\dots(2)_C(k+1)_A\end{array} \quad (k)_C \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\(k)_C\dots(3)_C(2)_C(k+1)_C\end{array}$$

$$(k)_B \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\(k+1)_B(k-1)_D\dots(2)_D(1)_D\end{array} \quad (k)_D \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\(k)_D(k-1)_D\dots(2)_D(1)_D\end{array}$$

$$F_A(u) = \textstyle\sum_{\sigma\in A} t^{|\sigma|}u^{k(\sigma)} = t^2u^2 + 2tuF_A(u)$$

$$F_B(u) = t^2u + t\frac{uF_A(1) - uF_A(u)}{1-u} + t\frac{uF_B(1) - u^2F_B(u)}{1-u}$$

$$F_C(u) = t\frac{u^2F_A(1) - F_A(u)}{1-u} + t\frac{u^2F_C(1) - u^2F_C(u)}{1-u}$$

Equations for generating functions

- The permutation • has two children with respective label $(2)_A$ and $(1)_B$

$$(k)_A \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\(k+1)_A(k-1)_C\dots(2)_C(k+1)_A\end{array} \qquad (k)_C \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\(k)_C\dots(3)_C(2)_C(k+1)_C\end{array}$$

$$(k)_B \xrightarrow{\theta} \begin{array}{c}(1)_B(2)_B\dots(k-1)_B(k)_B\\(k+1)_B(k-1)_D\dots(2)_D(1)_D\end{array} \qquad (k)_D \xrightarrow{\theta} \begin{array}{c}(1)_D(2)_D\dots(k-1)_D(k)_D\\(k)_D(k-1)_D\dots(2)_D(1)_D\end{array}$$

$$F_A(u) = \sum_{\sigma\in A} t^{|\sigma|}u^{k(\sigma)} = t^2u^2 + 2tuF_A(u)$$

$$F_B(u) = t^2u + t\frac{uF_A(1) - uF_A(u)}{1-u} + t\frac{uF_B(1) - u^2F_B(u)}{1-u}$$

$$F_C(u) = t\frac{u^2F_A(1) - F_A(u)}{1-u} + t\frac{u^2F_C(1) - u^2F_C(u)}{1-u}$$

$$F_D(u) = t\frac{uF_B(1) - F_B(u)}{1-u} + t\frac{uF_C(1) - uF_C(u)}{1-u} + 2t\frac{uF_D(1) - uF_D(u)}{1-u}$$

Resolution of the equations

The resulting system:

$$F_A(u) = t^2u^2 + 2tuF_A(u)$$

$$F_B(u) = t^2u + t\frac{uF_A(1) - uF_A(u)}{1-u} + t\frac{uF_B(1) - u^2F_B(u)}{1-u}$$

$$F_C(u) = t\frac{u^2F_A(1) - F_A(u)}{1-u} + t\frac{u^2F_C(1) - u^2F_C(u)}{1-u}$$

$$F_D(u) = t\frac{uF_B(1) - F_B(u)}{1-u} + t\frac{uF_C(1) - uF_C(u)}{1-u} + 2t\frac{uF_D(1) - uF_D(u)}{1-u}$$

Resolution of the equations

The resulting system:

$$F_A(u) = t^2u^2 + 2tuF_A(u)$$

$$F_B(u) = t^2u + t\frac{uF_A(1) - uF_A(u)}{1-u} + t\frac{uF_B(1) - u^2F_B(u)}{1-u}$$

$$F_C(u) = t\frac{u^2F_A(1) - F_A(u)}{1-u} + t\frac{u^2F_C(1) - u^2F_C(u)}{1-u}$$

$$F_D(u) = t\frac{uF_B(1) - F_B(u)}{1-u} + t\frac{uF_C(1) - uF_C(u)}{1-u} + 2t\frac{uF_D(1) - uF_D(u)}{1-u}$$

In view of the first equation, the series $F_A(u)$ is rational: $F_A(u) = \dfrac{t^2u^2}{1-2tu}$

Resolution of the equations

The resulting system:

$$F_A(u) = t^2u^2 + 2tuF_A(u)$$

$$F_B(u) = t^2u + t\frac{uF_A(1) - uF_A(u)}{1-u} + t\frac{uF_B(1) - u^2F_B(u)}{1-u}$$

$$F_C(u) = t\frac{u^2F_A(1) - F_A(u)}{1-u} + t\frac{u^2F_C(1) - u^2F_C(u)}{1-u}$$

$$F_D(u) = t\frac{uF_B(1) - F_B(u)}{1-u} + t\frac{uF_C(1) - uF_C(u)}{1-u} + 2t\frac{uF_D(1) - uF_D(u)}{1-u}$$

In view of the first equation, the series $F_A(u)$ is rational: $F_A(u) = \dfrac{t^2u^2}{1-2tu}$

Then we have a sequence of 3 simple linear equations with 1 catalytic variable.

$\Rightarrow$ standard resolution by applying 3 times the kernel method.

Resolution of the equations

Recall: $F_A(u) = \dfrac{t^2u^2}{1-2tu}$

Consider the second equation:

$$F_B(u) = t^2u + t\frac{uF_A(1) - uF_A(u)}{1-u} + t\frac{uF_B(1) - u^2F_B(u)}{1-u}$$

Resolution of the equations

Recall: $F_A(u) = \dfrac{t^2u^2}{1-2tu}$

Consider the second equation:

$$F_B(u) = t^2u + t\frac{uF_A(1) - uF_A(u)}{1-u} + t\frac{uF_B(1) - u^2F_B(u)}{1-u}$$

it rewrites as

$$(1-u+tu^2)F_B(u) = tu(t(1-u) + F_A(1) - F_A(u) + F_B(1))$$

Resolution of the equations

Recall: $F_A(u) = \dfrac{t^2u^2}{1-2tu}$

Consider the second equation:

$$F_B(u) = t^2u + t\frac{uF_A(1) - uF_A(u)}{1-u} + t\frac{uF_B(1) - u^2F_B(u)}{1-u}$$

it rewrites as

$$(1-u+tu^2)F_B(u) = tu(t(1-u) + F_A(1) - F_A(u) + F_B(1))$$

Kernel method: find a series that can be substituted for u on both sides and that cancels the kernel

$$K(u) = 1 - u + tu^2$$

Here we use the Catalan generating series $C \equiv C(t) = \dfrac{1-\sqrt{1-4t}}{2t}$ which indeed satisties $C = 1 + tC^2$, that is $K(C) = 0$.

Resolution of the equations

Recall: $F_A(u) = \dfrac{t^2u^2}{1-2tu}$

Consider the second equation:

$$F_B(u) = t^2u + t\frac{uF_A(1) - uF_A(u)}{1-u} + t\frac{uF_B(1) - u^2F_B(u)}{1-u}$$

it rewrites as

$$(1-u+tu^2)F_B(u) = tu(t(1-u) + F_A(1) - F_A(u) + F_B(1))$$

Kernel method: find a series that can be substituted for u on both sides and that cancels the kernel

$$K(u) = 1 - u + tu^2$$

Here we use the Catalan generating series $C \equiv C(t) = \dfrac{1-\sqrt{1-4t}}{2t}$ which indeed satisties $C = 1 + tC^2$, that is $K(C) = 0$.

This yields

$$F_B(1) = (C-1)t + F_A(C) - F_A(1) = \frac{t(t-1)}{1-2t} + \frac{t}{\sqrt{1-4t}}$$

Resolution of the equations

Solving similarly the third and fourth equations yields:

$$F_A(1) = \frac{t^2}{1-2t} \qquad F_B(1) = \frac{t(t-1)}{1-2t} + \frac{t}{\sqrt{1-4t}}$$

$$F_C(1) = -\frac{t^2}{1-2t} + \frac{t^2}{\sqrt{1-4t}} \quad F_D(1) = \frac{t(1-7t+14t^2-4t^3)}{(1-2t)(1-4t)^{3/2}} - \frac{t(1-3t)}{(1-4t)}$$

Resolution of the equations

Solving similarly the third and fourth equations yields:

$$F_A(1) = \frac{t^2}{1-2t} \qquad F_B(1) = \frac{t(t-1)}{1-2t} + \frac{t}{\sqrt{1-4t}}$$

$$F_C(1) = -\frac{t^2}{1-2t} + \frac{t^2}{\sqrt{1-4t}} \quad F_D(1) = \frac{t(1-7t+14t^2-4t^3)}{(1-2t)(1-4t)^{3/2}} - \frac{t(1-3t)}{(1-4t)}$$

and the generating series of square permutations of size at least 2 is

$$F_S = F_A(1) + F_B(1) + F_C(1) + F_D(1) = \frac{2t^2(1-3t)}{(1-4t)^2} - \frac{4t^3}{(1-4t)^{3/2}}$$

Resolution of the equations

Solving similarly the third and fourth equations yields:

$$F_A(1) = \frac{t^2}{1-2t} \qquad F_B(1) = \frac{t(t-1)}{1-2t} + \frac{t}{\sqrt{1-4t}}$$

$$F_C(1) = -\frac{t^2}{1-2t} + \frac{t^2}{\sqrt{1-4t}} \quad F_D(1) = \frac{t(1-7t+14t^2-4t^3)}{(1-2t)(1-4t)^{3/2}} - \frac{t(1-3t)}{(1-4t)}$$

and the generating series of square permutations of size at least 2 is

$$F_S = F_A(1) + F_B(1) + F_C(1) + F_D(1) = \frac{2t^2(1-3t)}{(1-4t)^2} - \frac{4t^3}{(1-4t)^{3/2}}$$

Extracting the coefficients with the binomial formula yields:

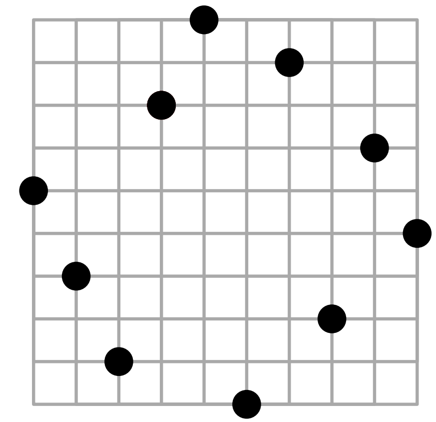
Theorem [Mansour/Severini 2007, this proof by D./Poulalhon 2008]
The number of square permutations of size $n \geq 2$ is

$$|\mathcal{S}_n| = (2n+4)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

Square permutations and convex polyominoes

Theorem [Mansour/Severini 2007]
For $n \geq 2$, the number of square permutations of size n is

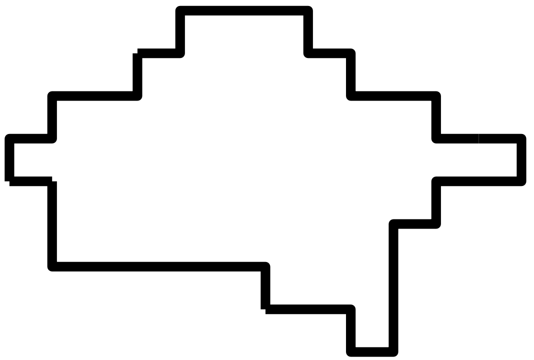
$$|\mathcal{S}_n| = (2n+4)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$



These numbers are reminiscent of a celebrated result of Delest and Viennot:

Theorem [Delest/Viennot 1984]
For $n \geq 3$, the number of convex polyominoes of semi perimeter $n+1$ is

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

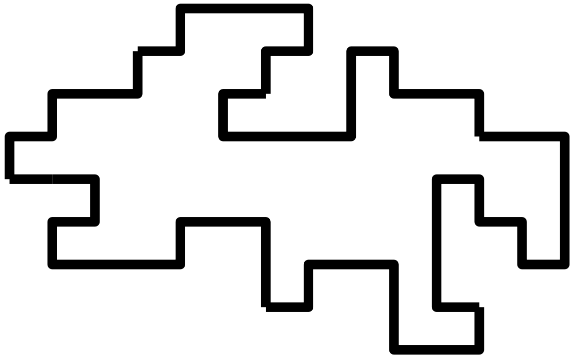


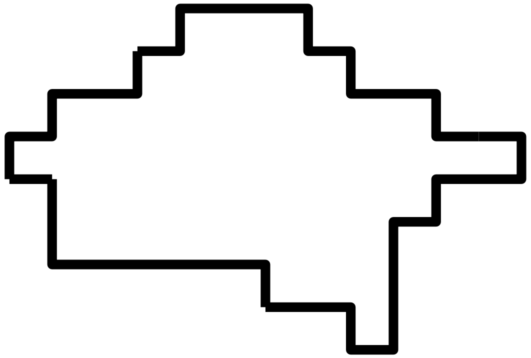
(definitions are coming !)

Polyominoes

Polyomino without hole (or self avoiding polygon)

$=$ (interior of a) closed simple curve on the grid $\mathbb{Z}^2$

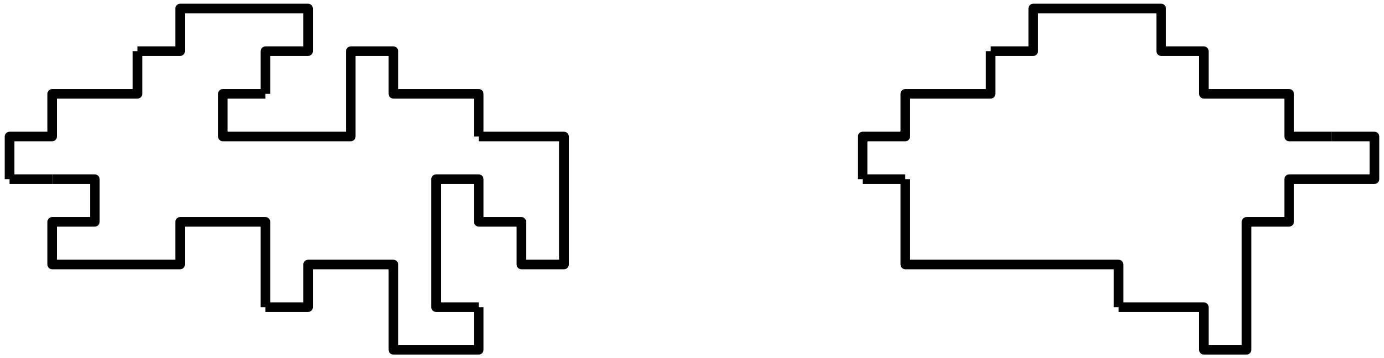




Polyominoes

Polyomino without hole (or self avoiding polygon)

$= $ (interior of a) closed simple curve on the grid $\mathbb{Z}^2$

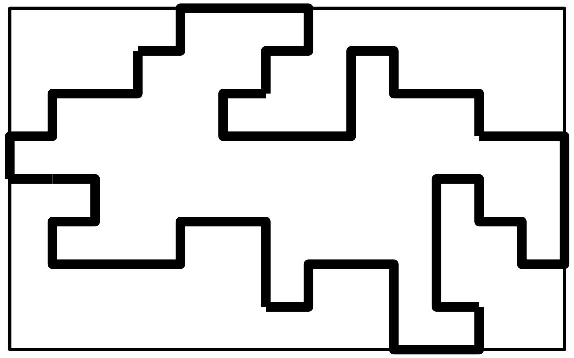


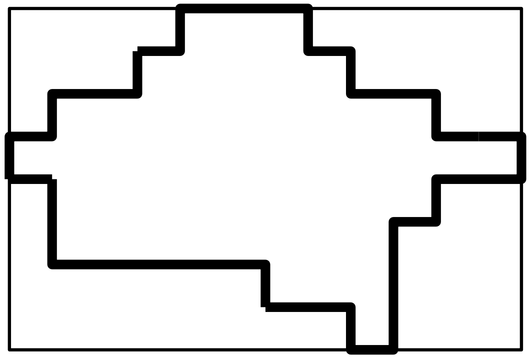
Size of a polyomino $=$ semi-perimeter

Polyominoes

Polyomino without hole (or self avoiding polygon)

$=$ (interior of a) closed simple curve on the grid $\mathbb{Z}^2$





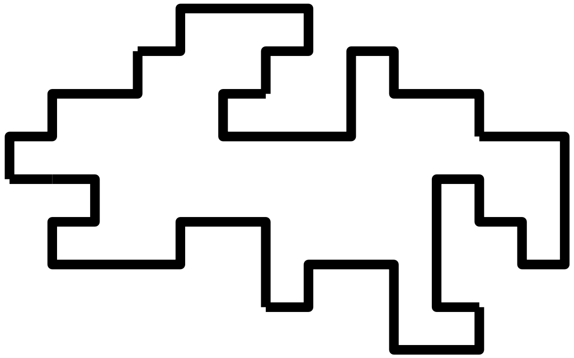
Size of a polyomino $=$ semi-perimeter

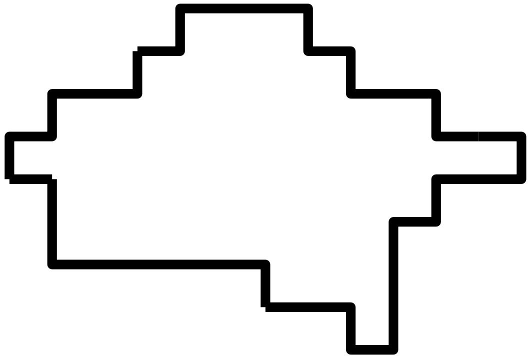
Bounding box $=$ smallest containing rectangle

Convex polyominoes

Convex polyomino $=$ a polyomino P is convex iff

- the intersection of its interior with any row or column is connected

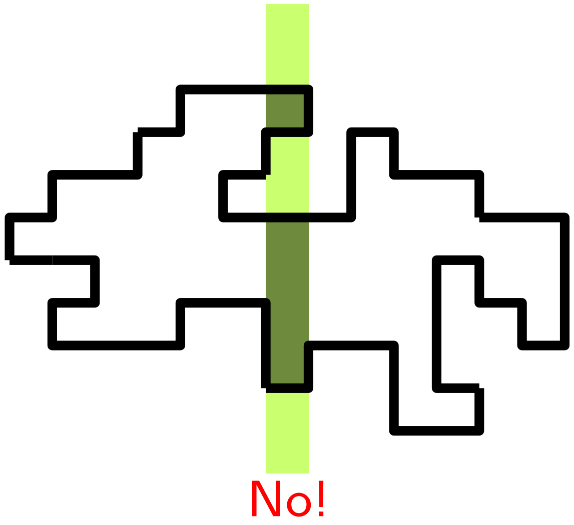


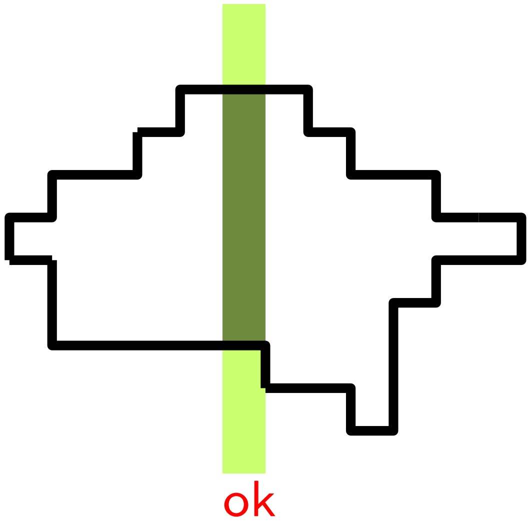


Convex polyominoes

Convex polyomino $=$ a polyomino P is convex iff

- the intersection of its interior with any row or column is connected

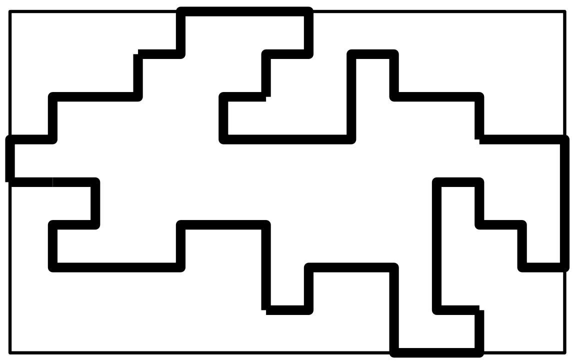


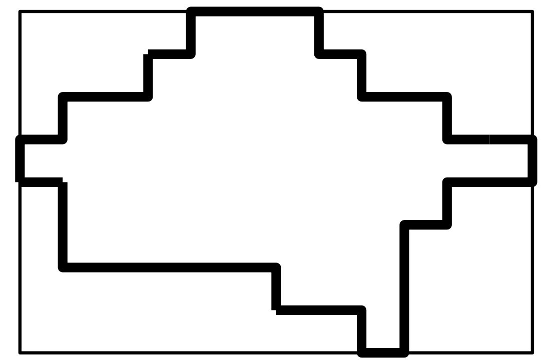


Convex polyominoes

Convex polyomino $=$ a polyomino P is convex iff

- the intersection of its interior with any row or column is connected





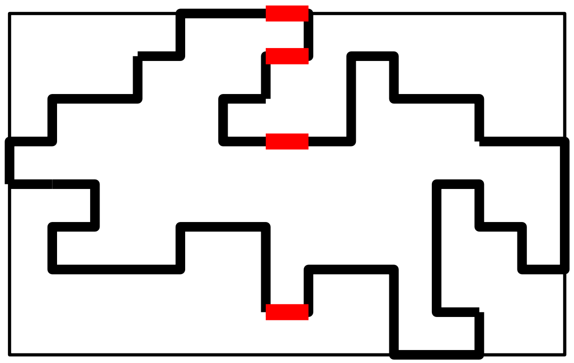
Equivalent conditions:

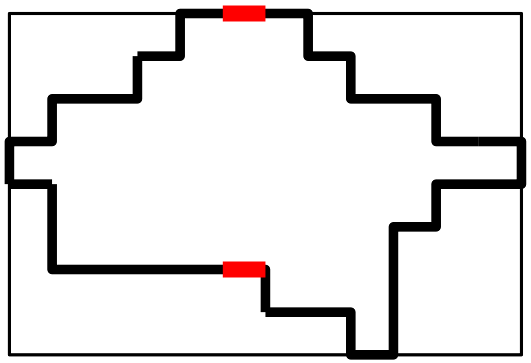
- its semi-perimeter is equal to that of its bounding box

Convex polyominoes

Convex polyomino $=$ a polyomino P is convex iff

- the intersection of its interior with any row or column is connected





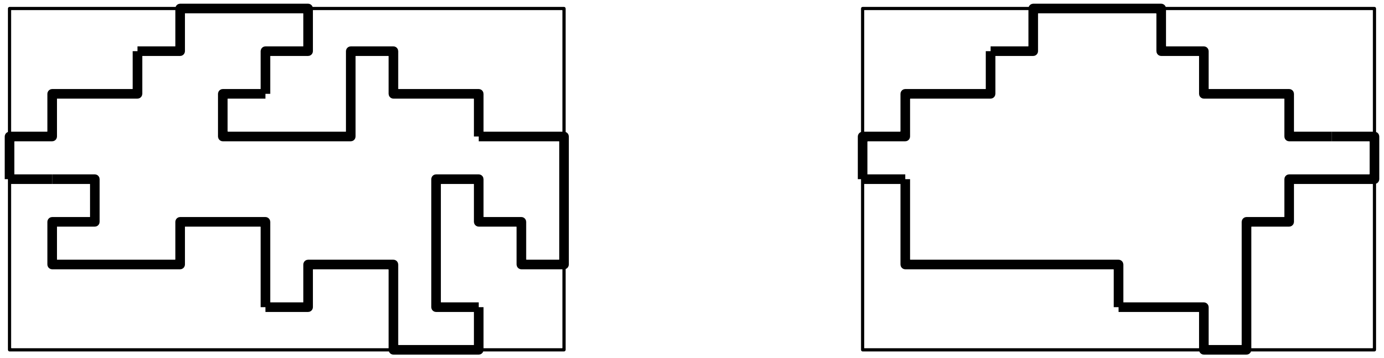
Equivalent conditions:

- its semi-perimeter is equal to that of its bounding box

Convex polyominoes

Convex polyomino $=$ a polyomino P is convex iff

- the intersection of its interior with any row or column is connected



Equivalent conditions:

- its semi-perimeter is equal to that of its bounding box

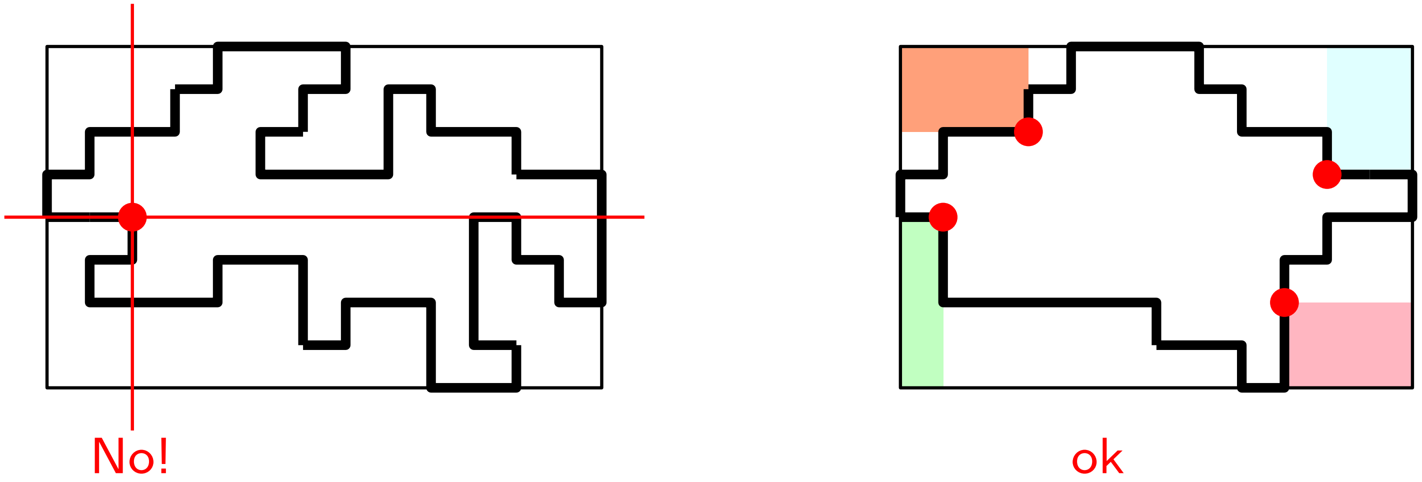
or

- any point of its boundary has at least one free quadrant

Convex polyominoes

Convex polyomino $=$ a polyomino P is convex iff

- the intersection of its interior with any row or column is connected



Equivalent conditions:

- its semi-perimeter is equal to that of its bounding box

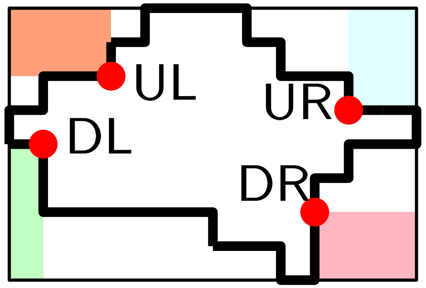
or

- any point of its boundary has at least one free quadrant

Subfamilies of convex polyominos

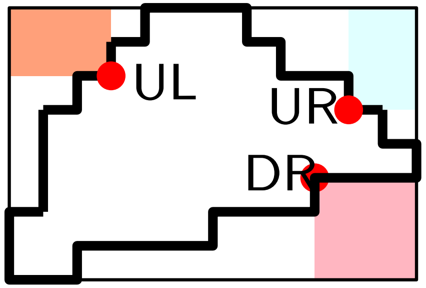
Convex polyomino

= all points of the boundary are upper-left (UL), upper-right (UR), down-left (DL) or down-right (DR)



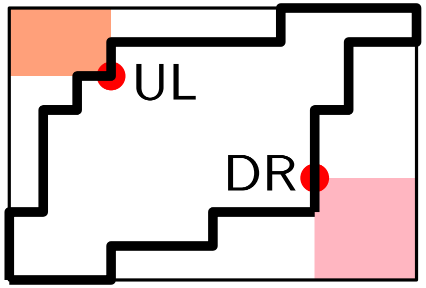
Directed convex polyomino

= all points of the boundary are UL, UR or DR



Parallelogram polyomino

= all points of the boundary are UL or DR

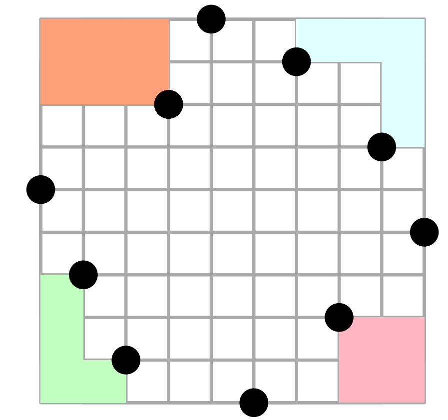


Subfamilies of square permutations

Square permutation

= permutation without internal point

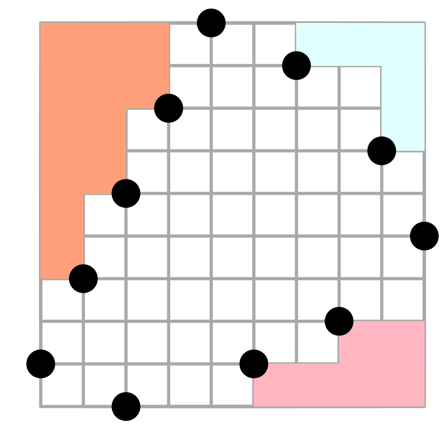
= all points are UL, UR, DL or DR.



Triangular permutation

= permutation without internal or DL points

= all points are UL, UR and DR

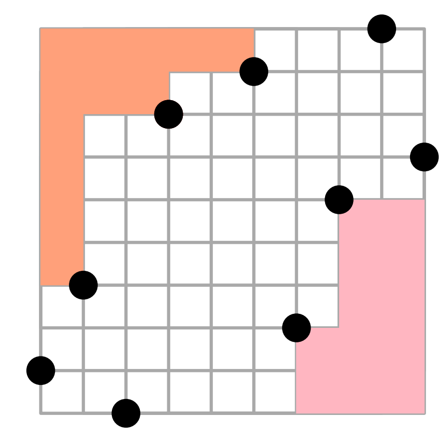


Parallel permutation

= permutation without internal, DL or UR points

= all points are UL or DR

= 321-avoiding permutations



Enumerative results

$\mathcal{C}_n = \{$ convex polyominoes of size $n+1\}$
$\mathcal{S}_n = \{$ square permutations of size $n\}$

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$
$$|\mathcal{S}_n| = (2n+4)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

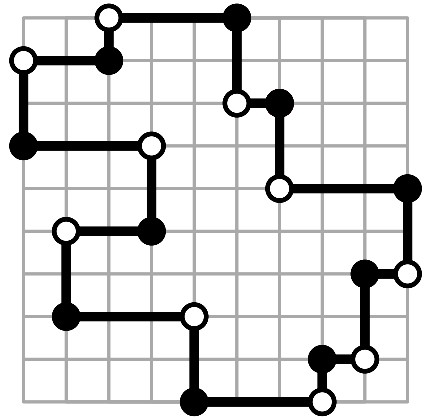
$\mathcal{D}_n = \{$ directed convex polyominos of size $n+1\}$
$\mathcal{T}_n^{\nearrow} = \{$ triangular permutations of size $n\}$

$$|\mathcal{D}_n| = \binom{2n-2}{n-1} \qquad |\mathcal{T}_n^{\nearrow}| = \binom{2n-2}{n-1}$$

$\mathcal{PP}_n = \{$ parallelogram polyominoes of size $n+1\}$
$\mathcal{P}_n = \{$ parallel permutations of size $n\}$

$$|\mathcal{PP}_n| = \frac{1}{n+1}\binom{2n}{n} \qquad |\mathcal{P}_n| = \frac{1}{n+1}\binom{2n}{n}$$

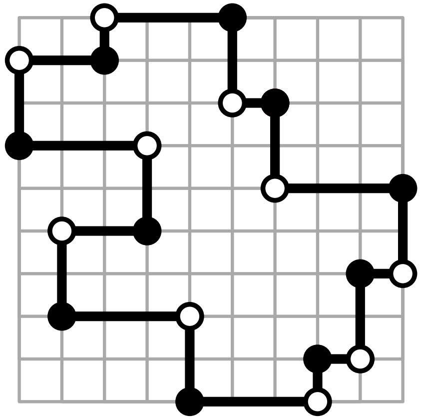
Permutominos: an intermediary structure?



Vertex = turnpoint of the boundary
Side = piece of boundary between two vertices

Permutomino[Incitti, 2006]
= polyomino whose sides use each vertical and horizontal line of its box exactly once

Permutominos: an intermediary structure?

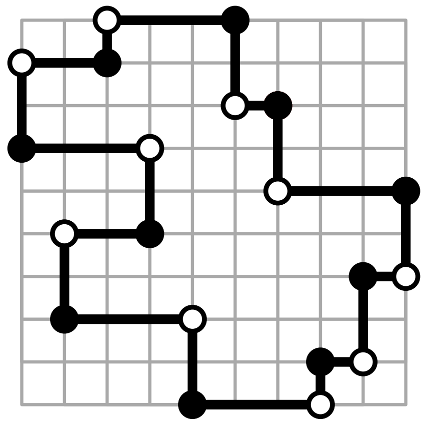


Vertex = turnpoint of the boundary
Side = piece of boundary between two vertices

Permutomino[Incitti, 2006]
= polyomino whose sides use each vertical and horizontal line of its box exactly once

Pair of permutations associated to a permutomino:
bicoloring vertices ⇒ a pair $(\sigma_\bullet, \sigma_\circ)$

Permutominos: an intermediary structure?



Vertex = turnpoint of the boundary
Side = piece of boundary between two vertices

Permutomino[Incitti, 2006]
= polyomino whose sides use each vertical and horizontal line of its box exactly once

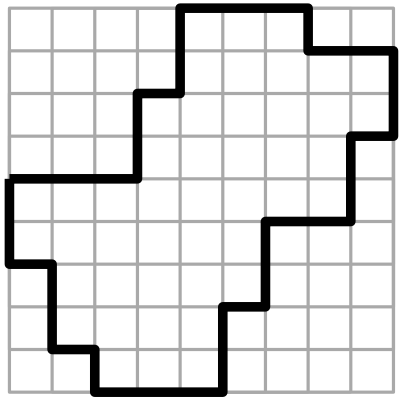
Pair of permutations associated to a permutomino:
bicoloring vertices ⇒ a pair $(\sigma_\bullet, \sigma_\circ)$

Size of a permutomino= size of $\sigma_\bullet$ (or $\sigma_\circ$)
The bounding box of a permutomino of size n is square with side of length $n-1$.

Convex permutominoes

Convex permutomino

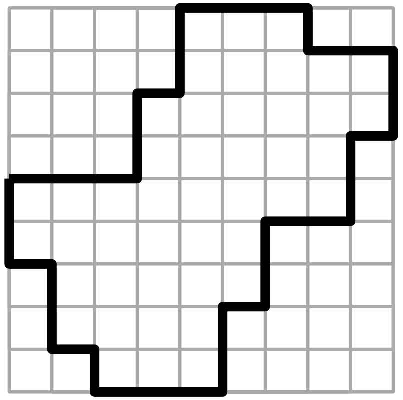
= permutomino whose underlying polyomino is convex



Convex permutominoes

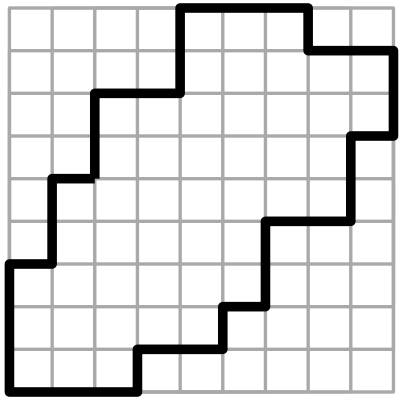
Convex permutomino

= permutomino whose underlying polyomino is convex



Directed convex permutomino

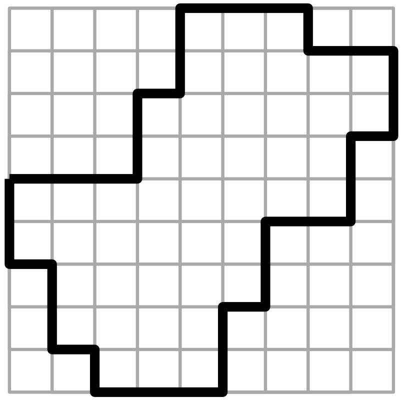
= permutomino whose underlying polyomino is directed convex



Convex permutominoes

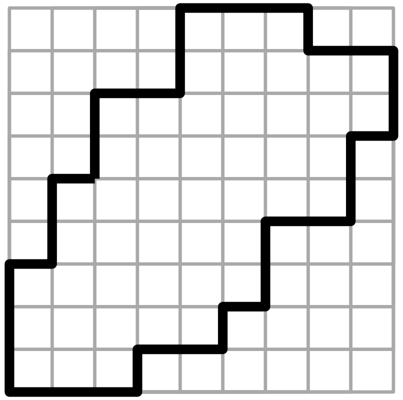
Convex permutomino

= permutomino whose underlying polyomino is convex



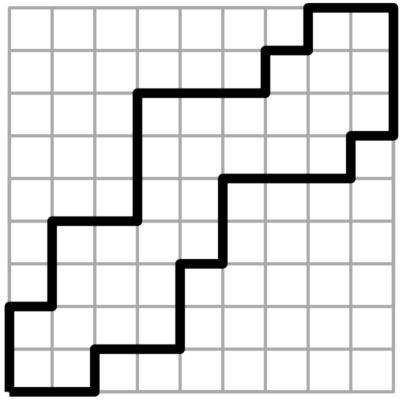
Directed convex permutomino

= permutomino whose underlying polyomino is directed convex



Parallelogram permutomino

= permutomino whose underlying polyomino is parallelogram



Enumerative results

$\mathcal{C}_n = \{$ convex polyominoes of size $n+1\}$
$\mathcal{S}_n = \{$ square permutations of size $n\}$

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$
$$|\mathcal{S}_n| = (2n+4)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

$\mathcal{D}_n = \{$ directed convex polyominos of size $n+1\}$
$\mathcal{T}_n = \{$ triangular permutations of size $n\}$

$$|\mathcal{D}_n| = \binom{2n-2}{n-1} \qquad |\mathcal{T}_n| = \binom{2n-2}{n-1}$$

$\mathcal{PP}_n = \{$ parallelogram polyominoes of size $n+1\}$
$\mathcal{P}_n = \{$ parallel permutations of size $n\}$

$$|\mathcal{PP}_n| = \frac{1}{n+1}\binom{2n}{n} \qquad |\mathcal{P}_n| = \frac{1}{n+1}\binom{2n}{n}$$

Enumerative results

$\mathcal{C}_n = \{$ convex polyominoes of size $n+1\}$
$\mathcal{S}_n = \{$ square permutations of size $n\}$

$\mathcal{CT}_n = \{$ convex permutominoes of size $n\}$

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$ [Delest Viennot,84]

$$|\mathcal{S}_n| = (2n+4)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$ [Mansour Severini, 2007]

$$|\mathcal{CT}_n| = (2n+4)\,4^{n-3} - (2n-3)\binom{2n-4}{n-2}$$ [Rinaldi et al., 2007]

$\mathcal{D}_n = \{$ directed convex polyominos of size $n+1\}$
$\mathcal{T}_n = \{$ triangular permutations of size $n\}$ [Mansour Severini, 2007]
$\mathcal{DT}_n = \{$ directed convex permutominos of size $n\}$ [Fanti et al., 2007]

$$|\mathcal{D}_n| = \binom{2n-2}{n-1} \qquad |\mathcal{T}_n| = \binom{2n-2}{n-1} \qquad |\mathcal{DT}_n| = \frac{1}{2}\binom{2n-2}{n-1}$$

$\mathcal{PP}_n = \{$ parallelogram polyominoes of size $n+1\}$
$\mathcal{P}_n = \{$ parallel permutations of size $n\}$
$\mathcal{PT}_n = \{$ parallelogram permutominoes of size $n\}$

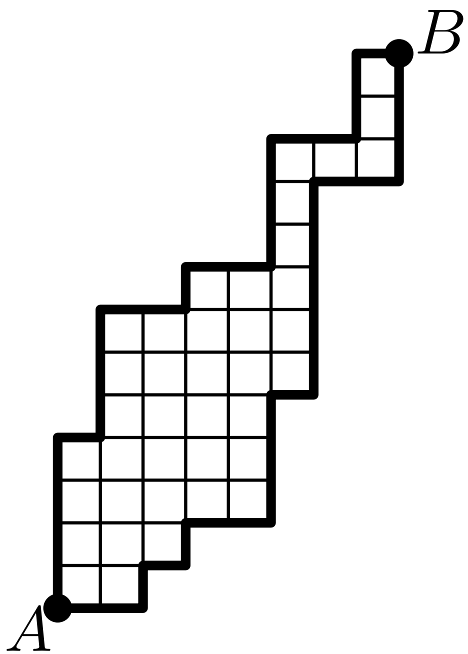
$$|\mathcal{PP}_n| = \frac{1}{n+1}\binom{2n}{n} \qquad |\mathcal{P}_n| = \frac{1}{n+1}\binom{2n}{n} \qquad |\mathcal{PT}_{n+1}| = \frac{1}{n+1}\binom{2n}{n}$$

Combinatorial interpretations

1) Some classical bijections for Catalan structures

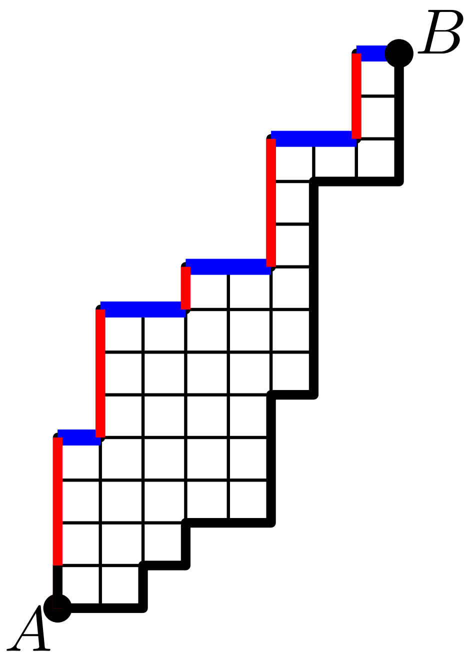
Catalan numbers and bijections

From parallelogram polyominoes of size $n + 1$
to square permutations of size n



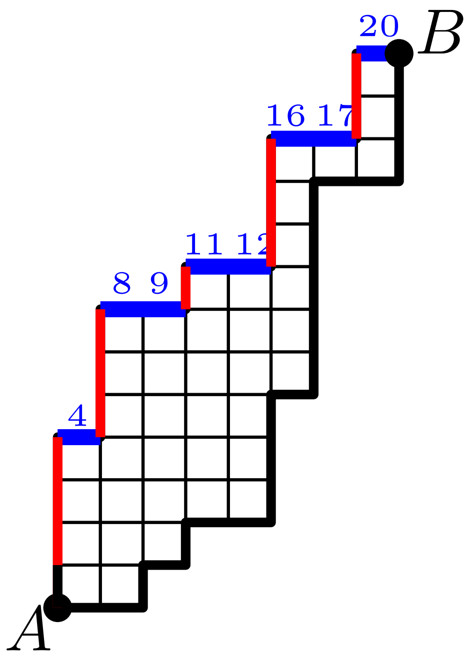
Catalan numbers and bijections

From parallelogram polyominoes of size $n+1$
to square permutations of size n



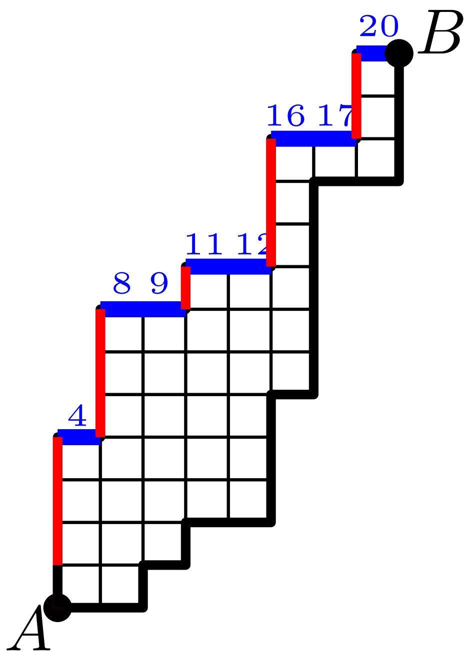
Catalan numbers and bijections

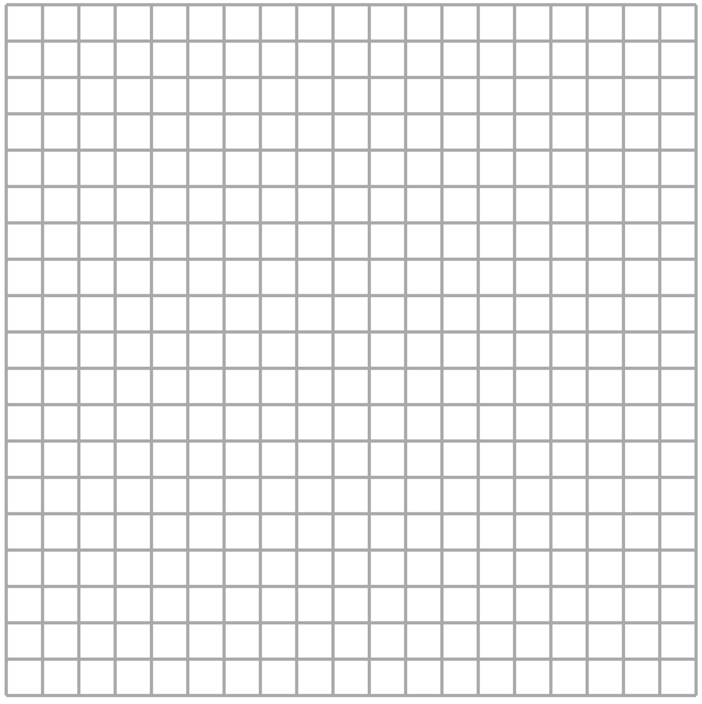
From parallelogram polyominoes of size $n+1$
to square permutations of size n



Catalan numbers and bijections

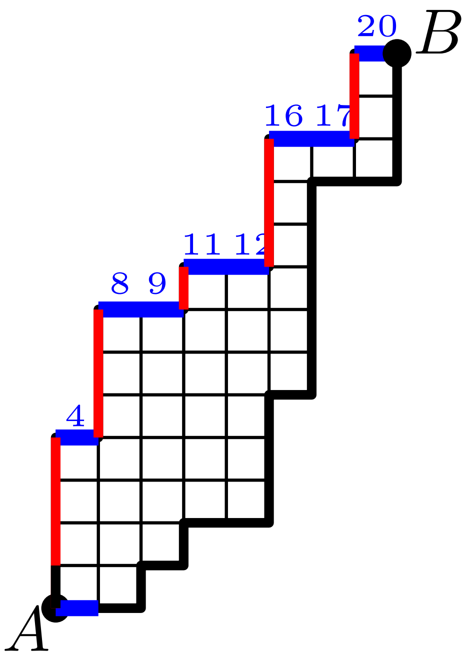
From parallelogram polyominoes of size $n+1$
to square permutations of size n

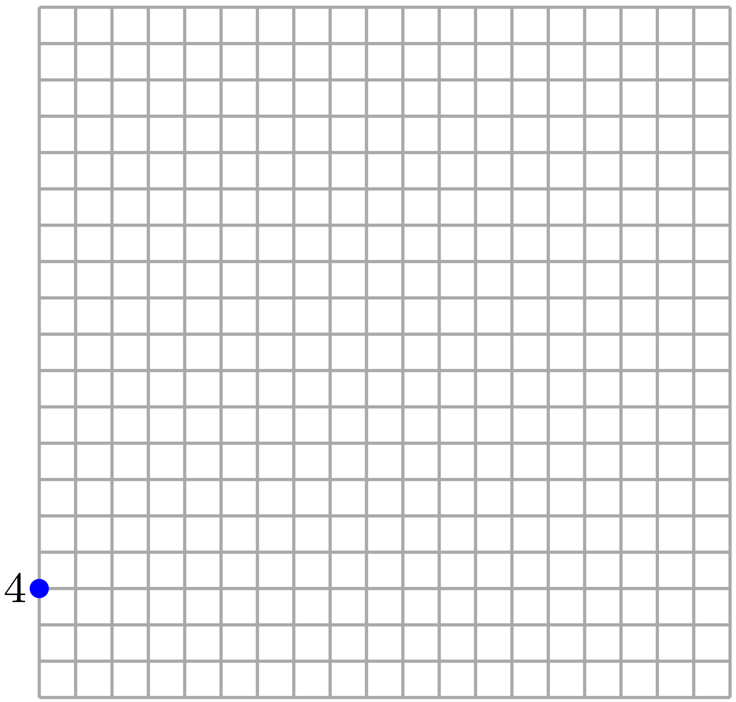




Catalan numbers and bijections

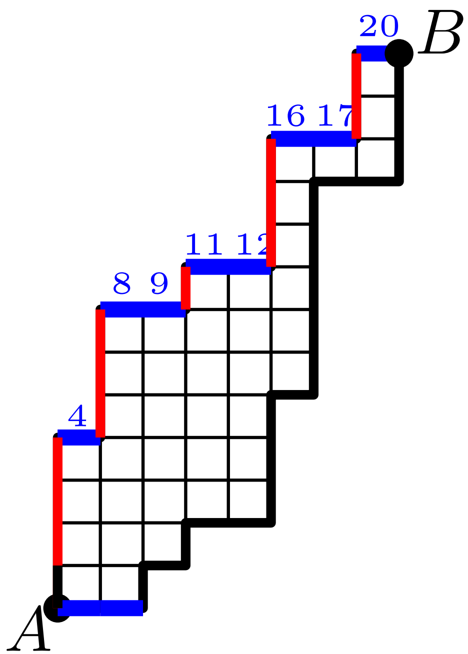
From parallelogram polyominoes of size $n+1$
to square permutations of size n

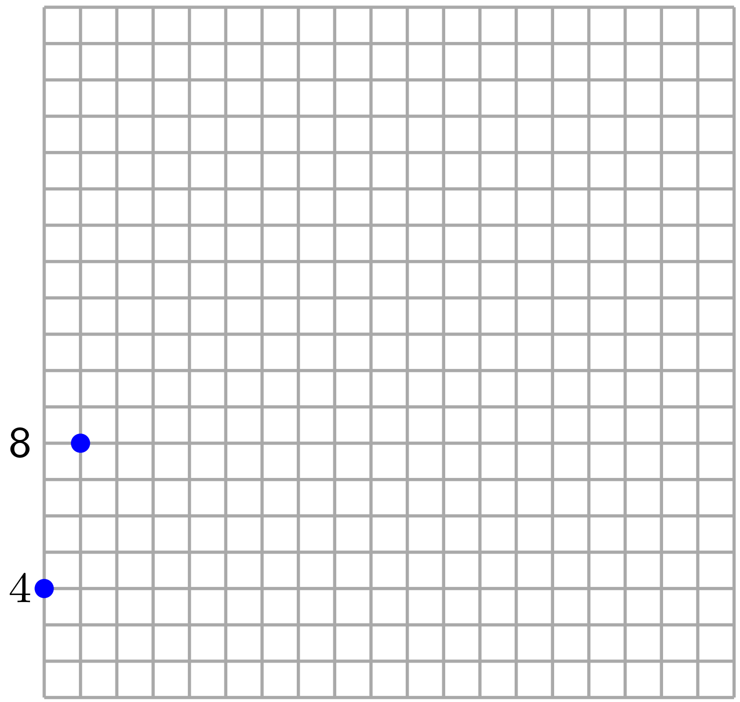




Catalan numbers and bijections

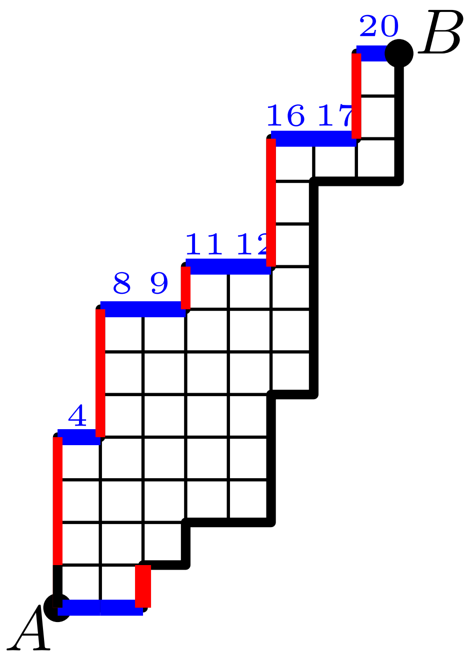
From parallelogram polyominoes of size $n+1$
to square permutations of size n

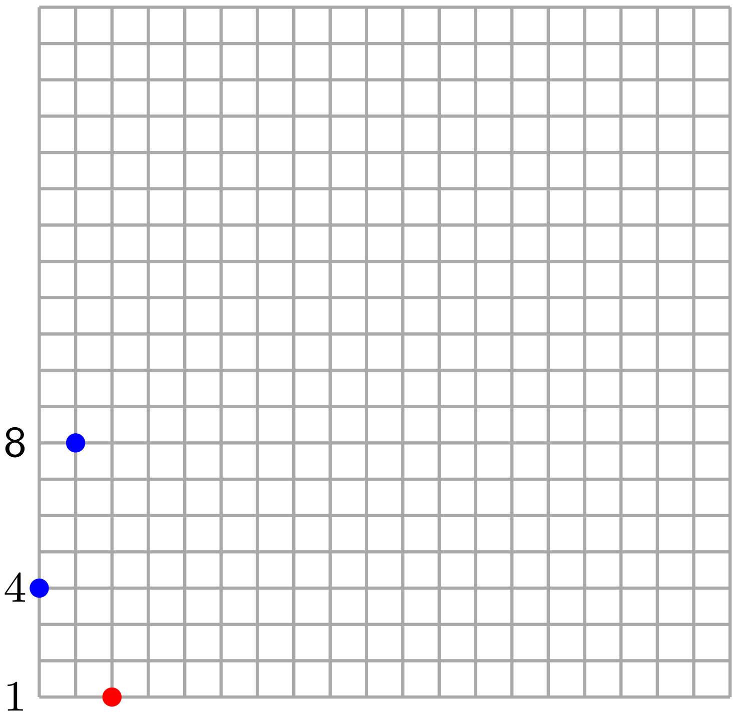




Catalan numbers and bijections

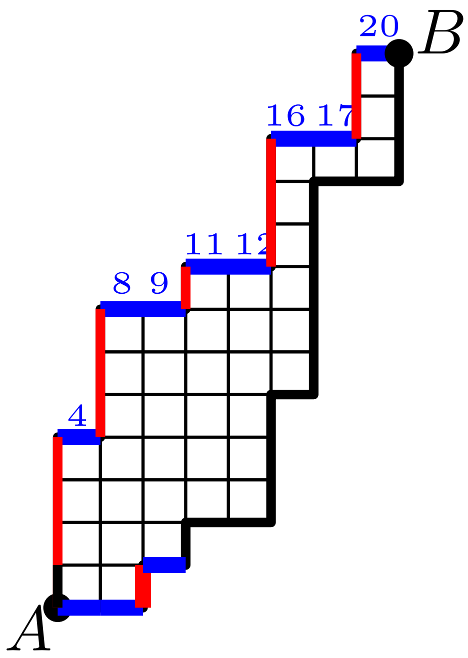
From parallelogram polyominoes of size $n+1$
to square permutations of size n

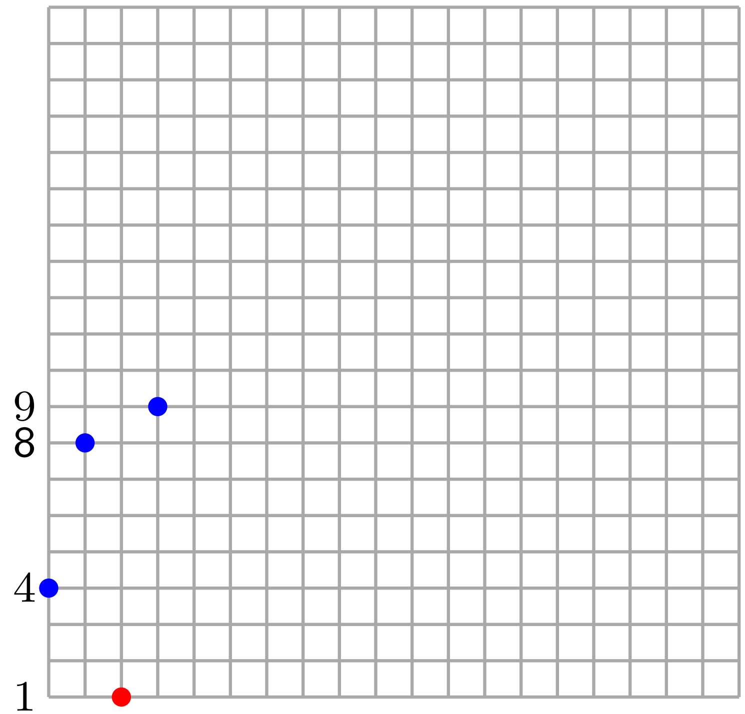




Catalan numbers and bijections

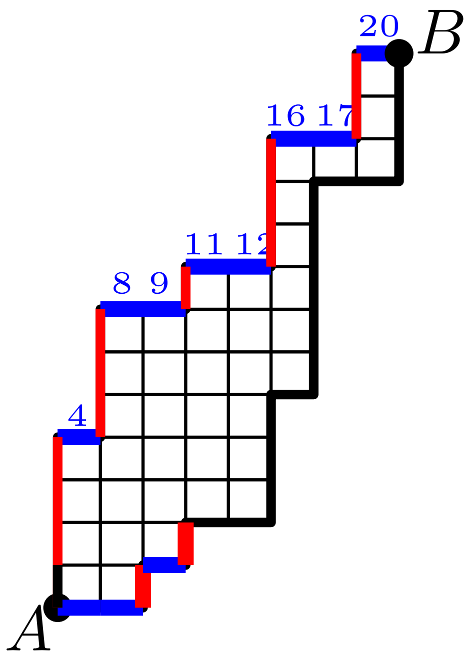
From parallelogram polyominoes of size $n+1$
to square permutations of size n

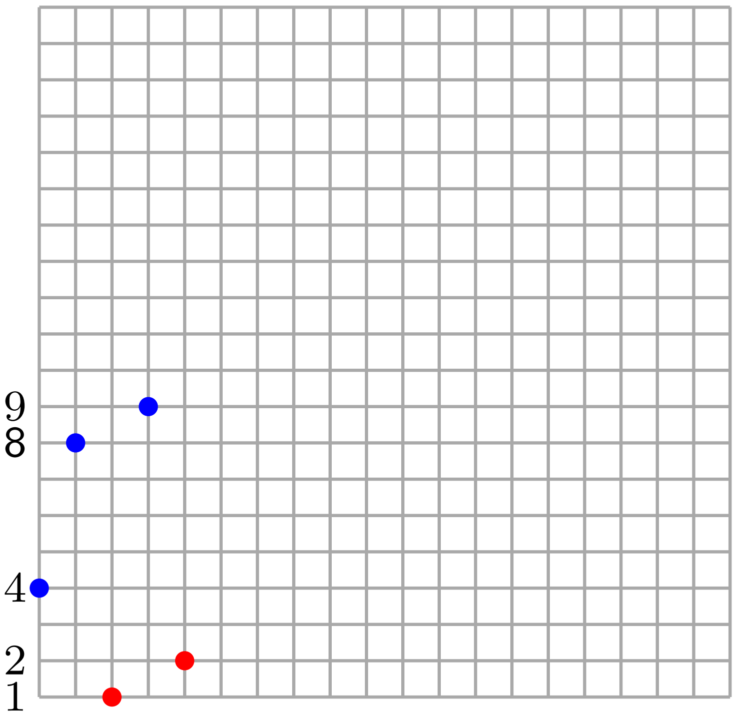




Catalan numbers and bijections

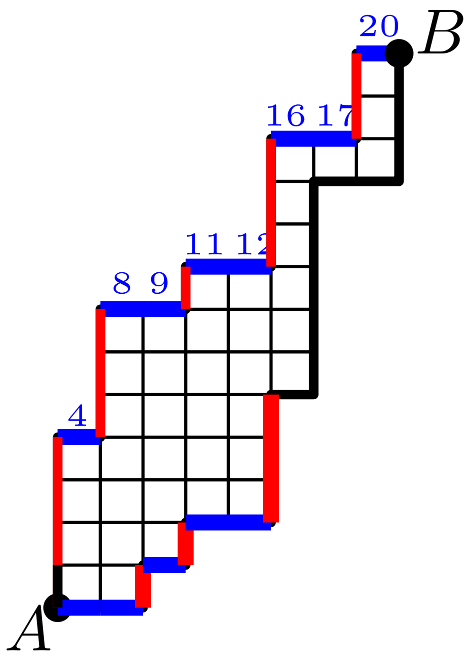
From parallelogram polyominoes of size $n + 1$
to square permutations of size n

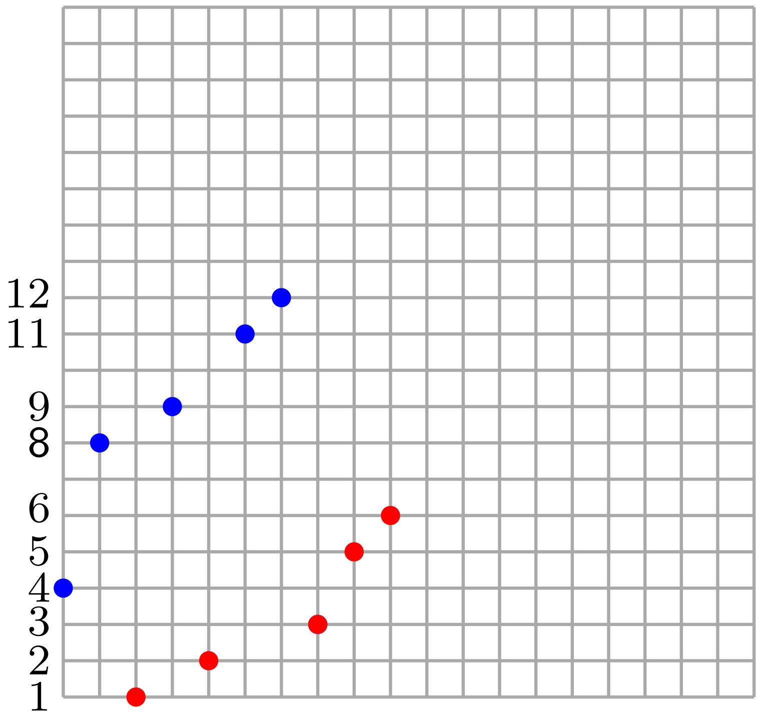




Catalan numbers and bijections

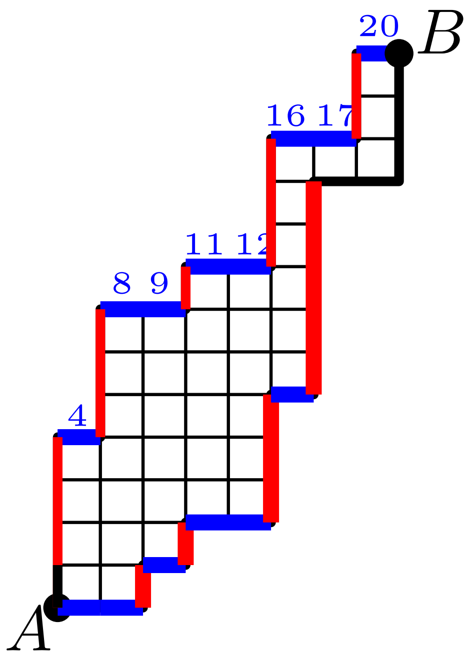
From parallelogram polyominoes of size $n+1$
to square permutations of size n

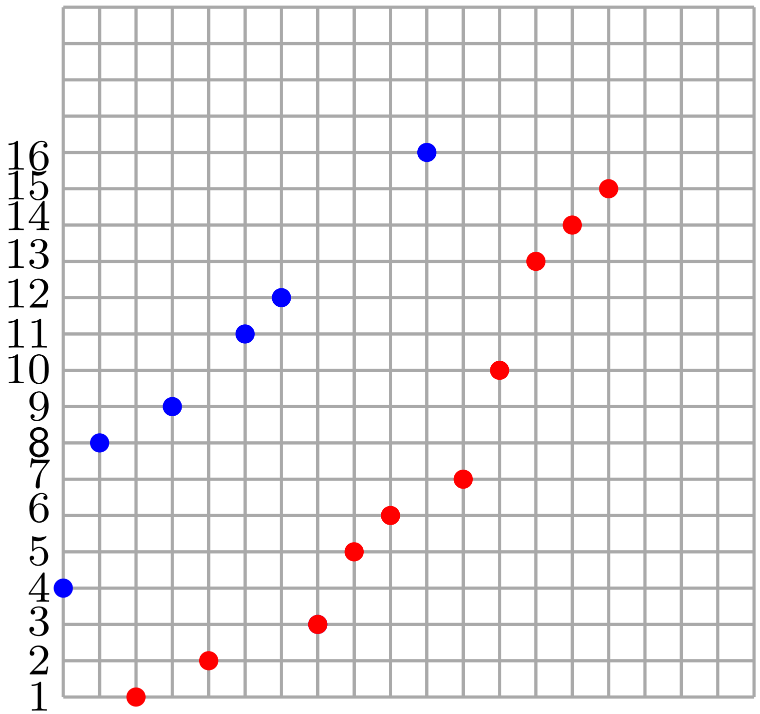




Catalan numbers and bijections

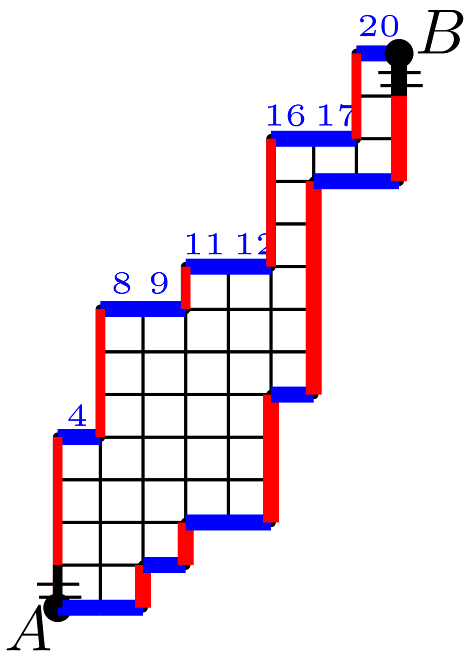
From parallelogram polyominoes of size $n+1$
to square permutations of size n

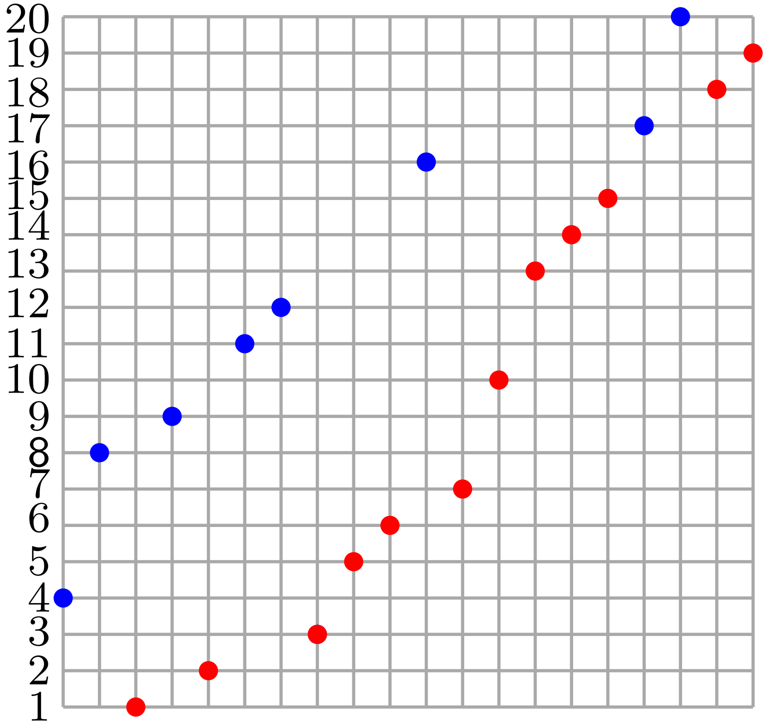




Catalan numbers and bijections

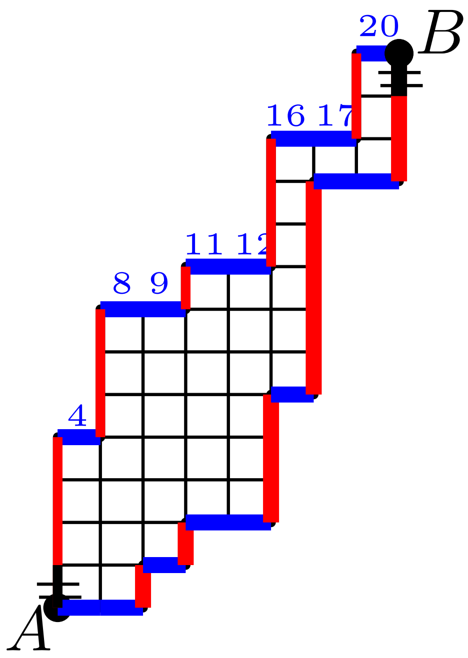
From parallelogram polyominoes of size $n+1$
to square permutations of size n

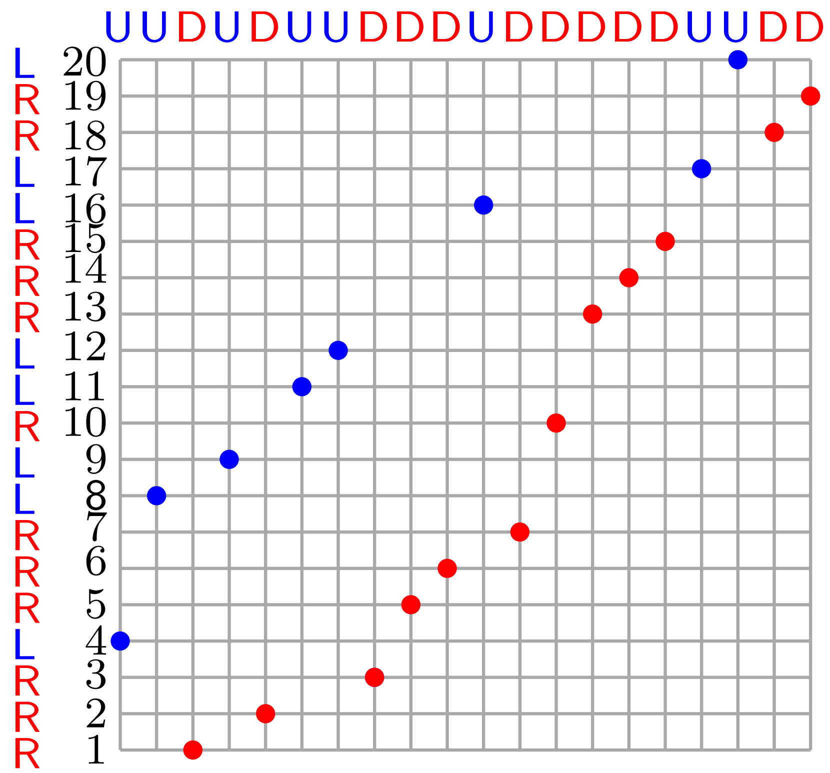




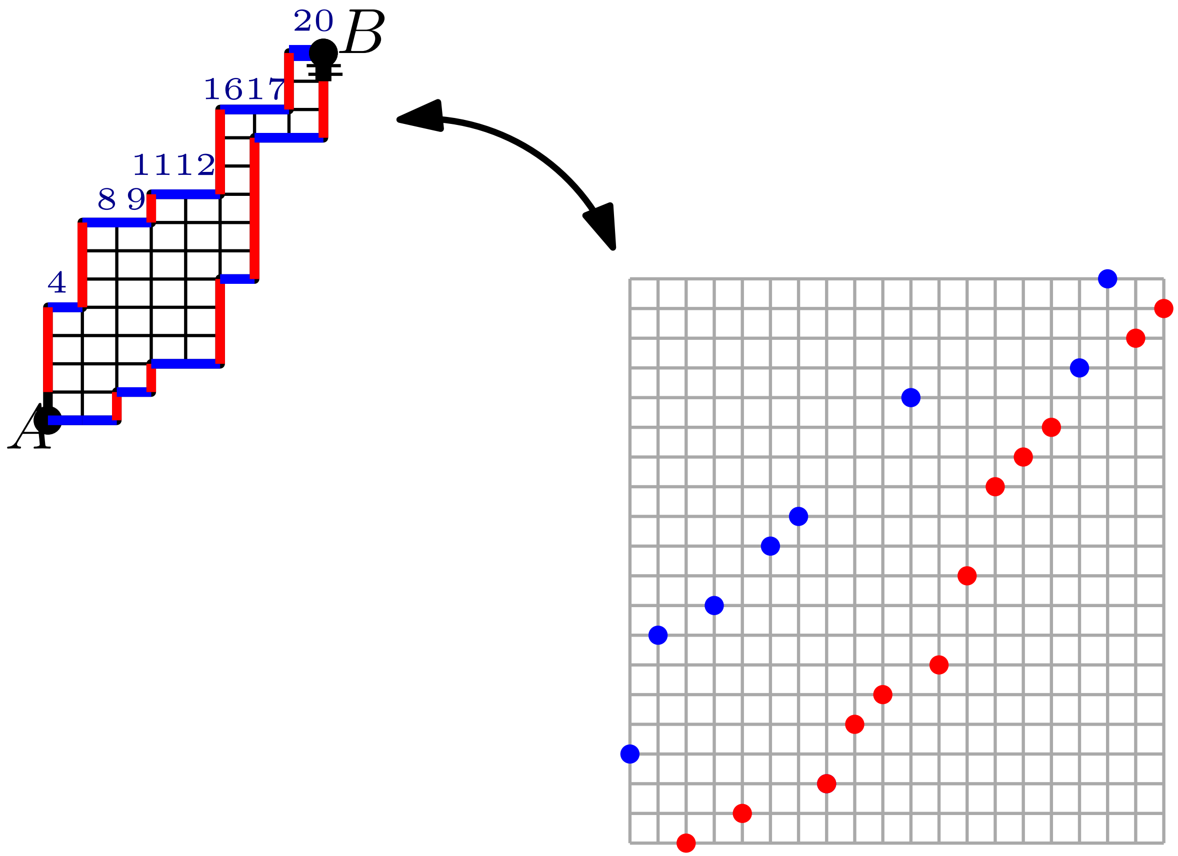
Catalan numbers and bijections

From parallelogram polyominoes of size $n+1$
to square permutations of size n

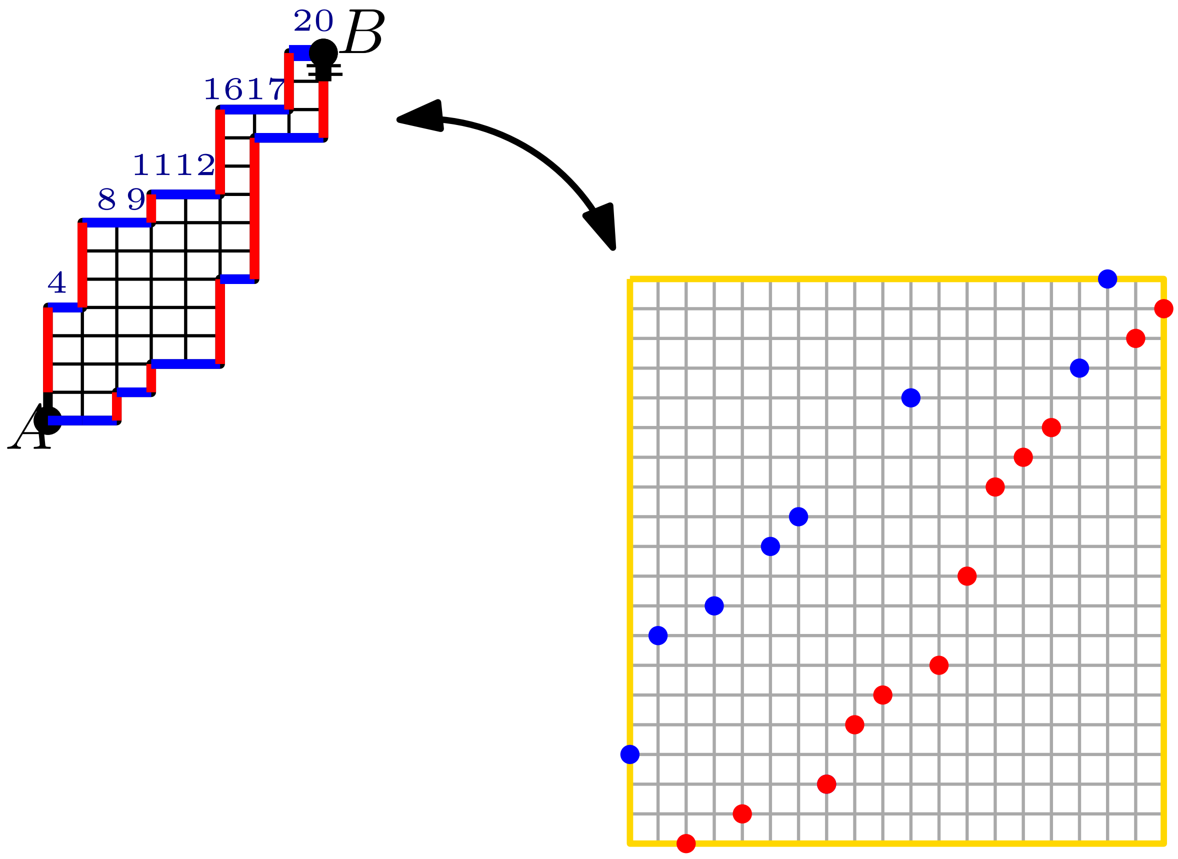




Catalan numbers and bijections

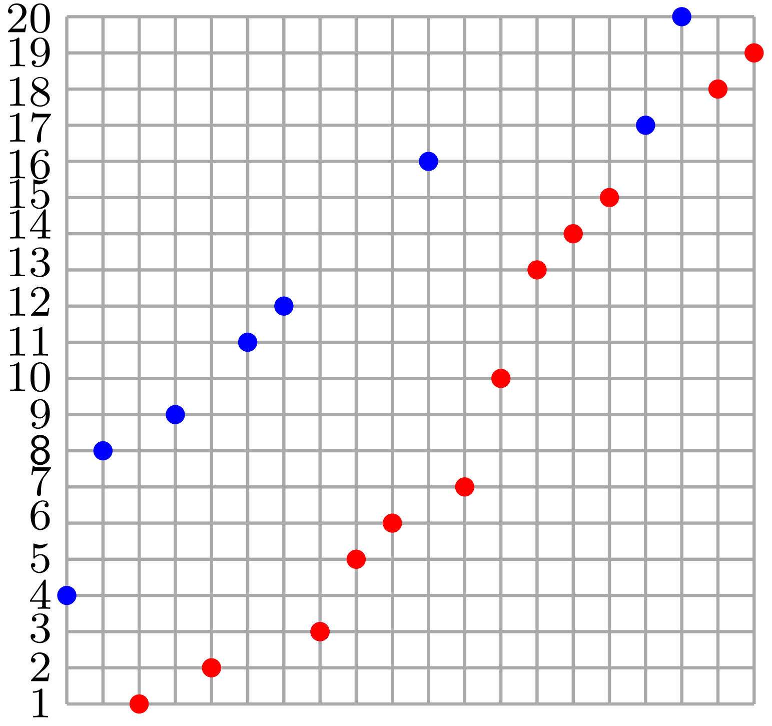


Catalan numbers and bijections



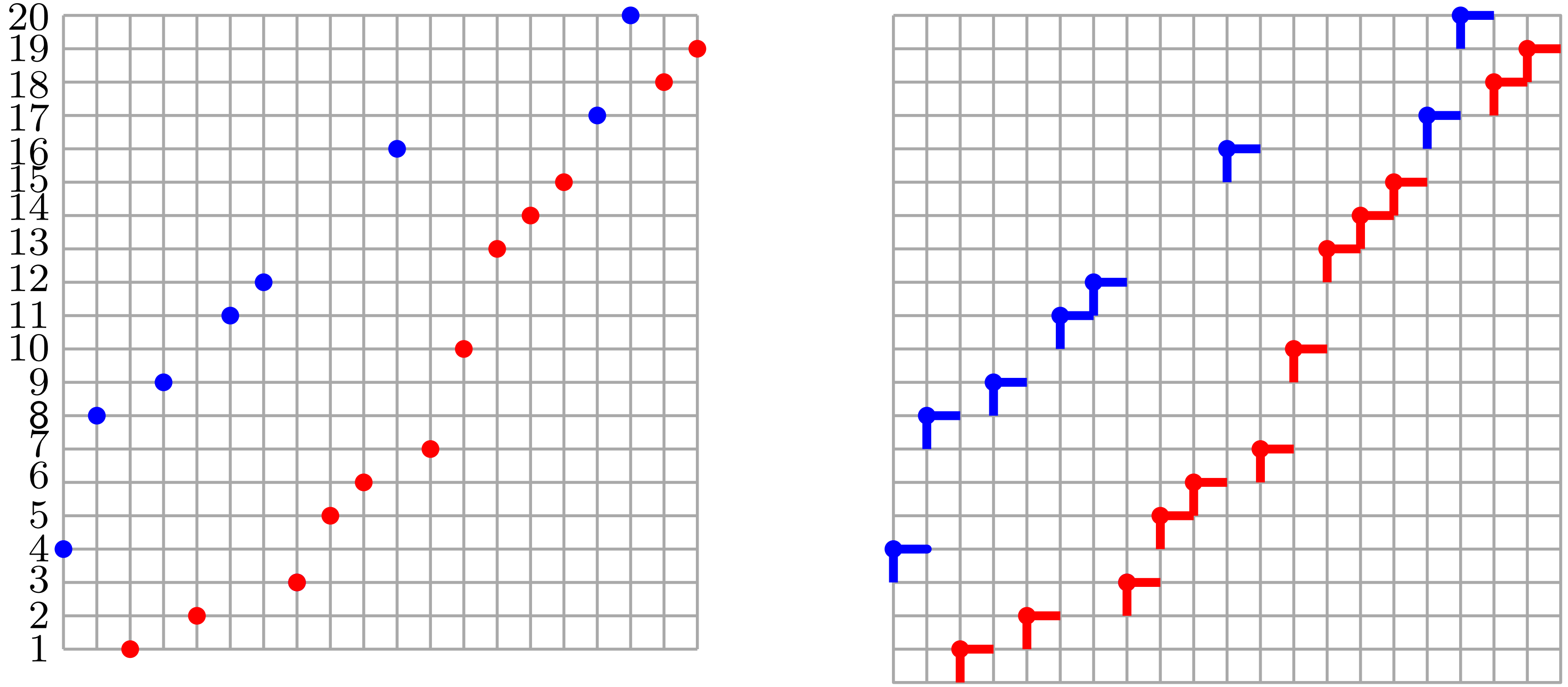
Catalan numbers and bijections

From parallel permutations of size n
to parallelogram permutominoes of size $n+1$.



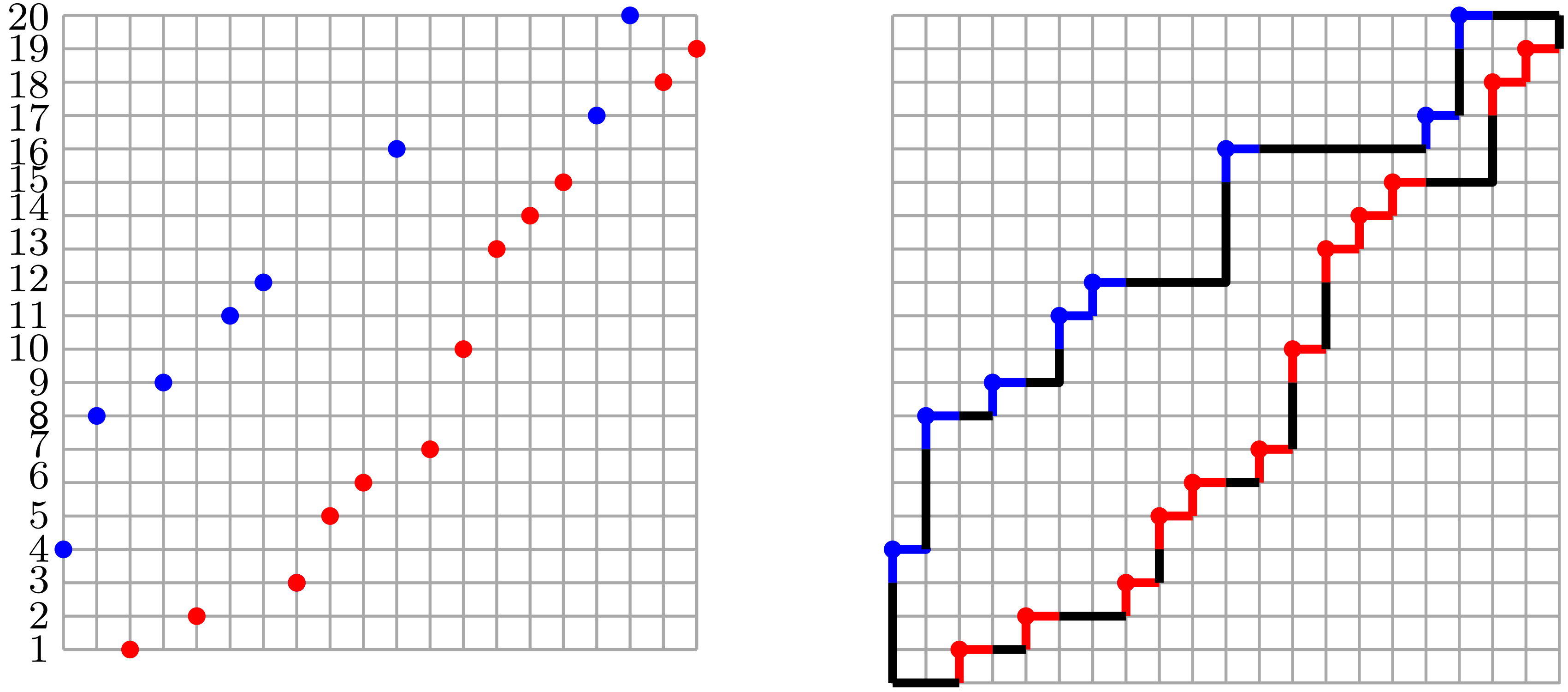
Catalan numbers and bijections

From parallel permutations of size n
to parallelogram permutominoes of size $n+1$.

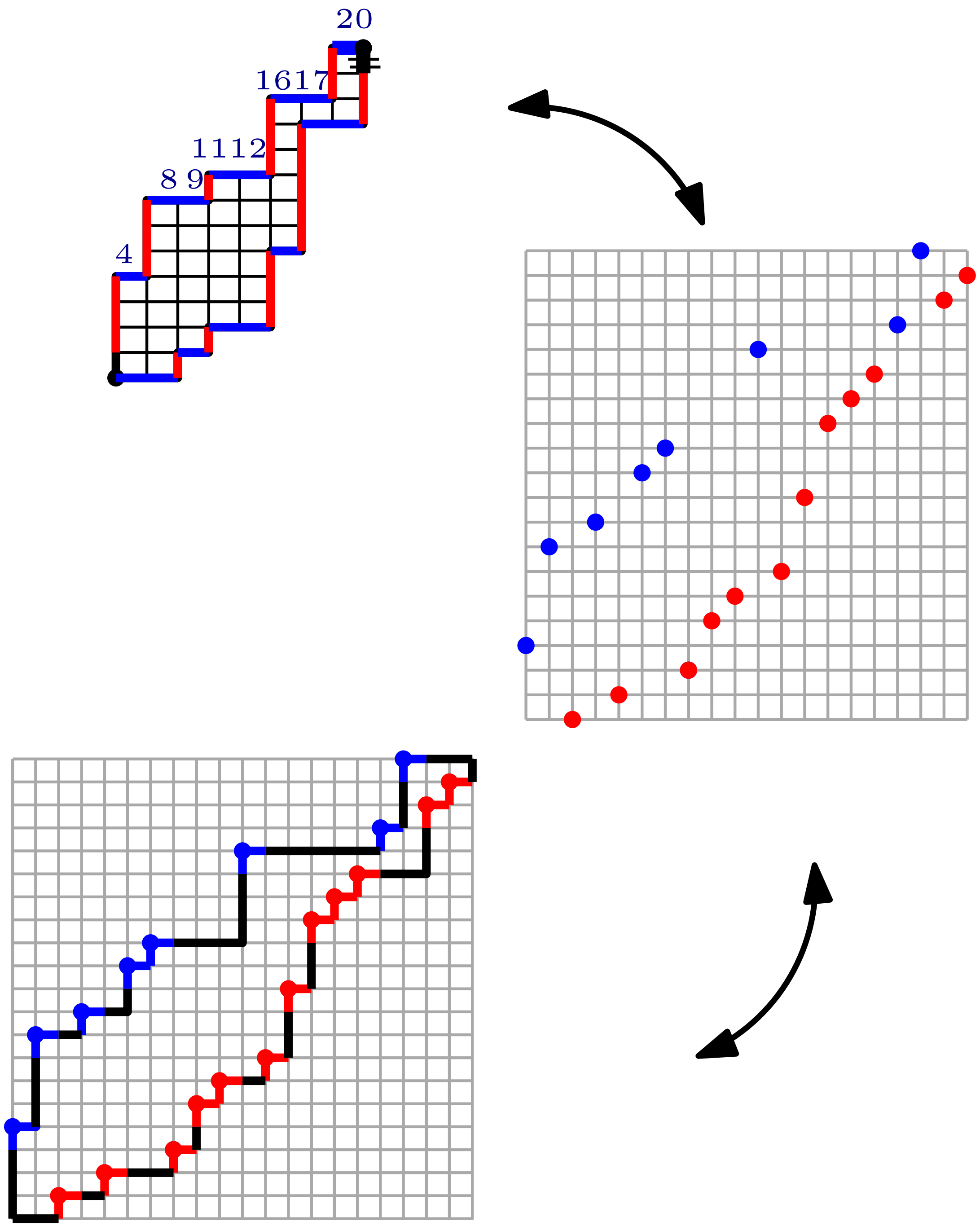


Catalan numbers and bijections

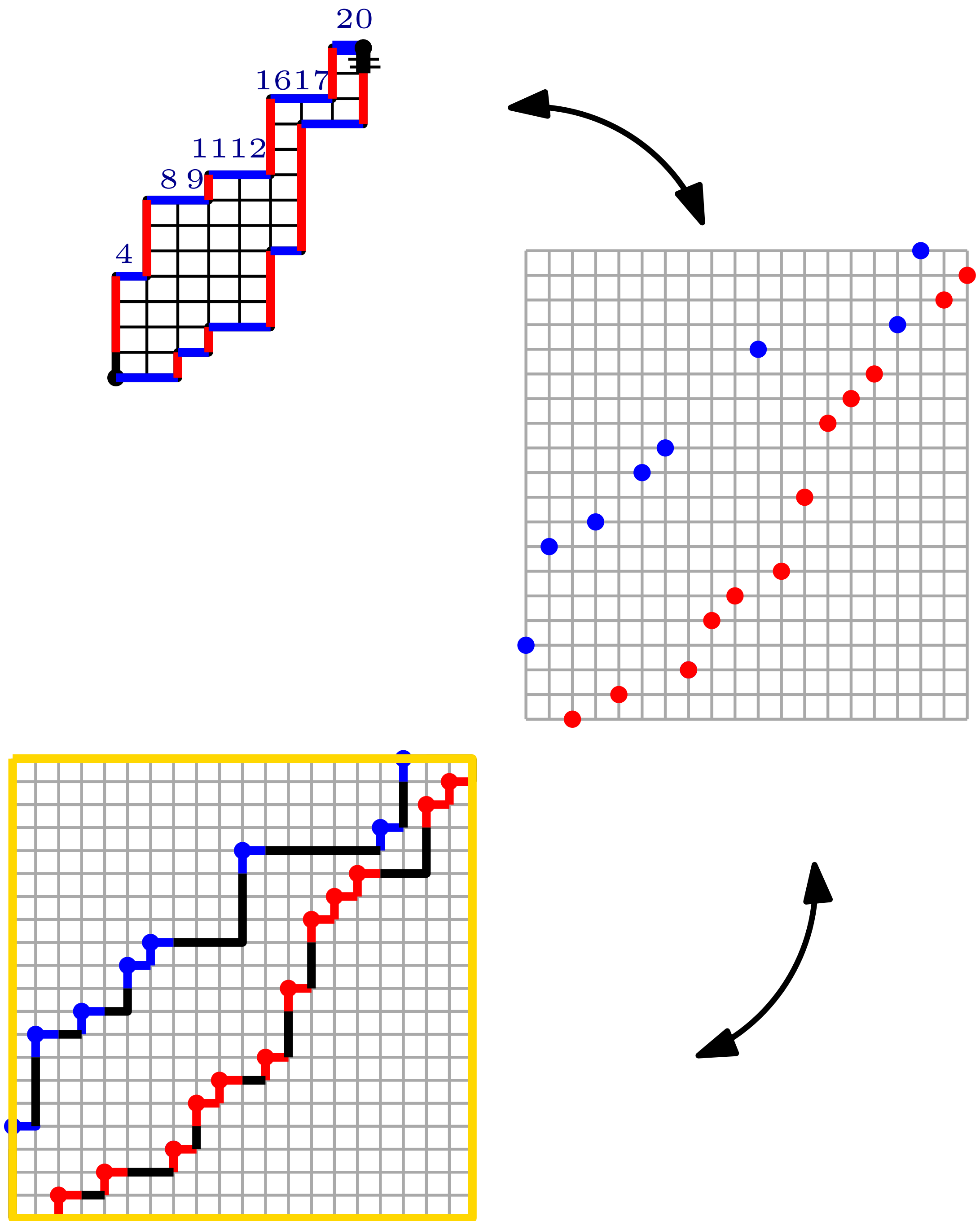
From parallel permutations of size n
to parallelogram permutominoes of size $n+1$.



Catalan numbers and bijections

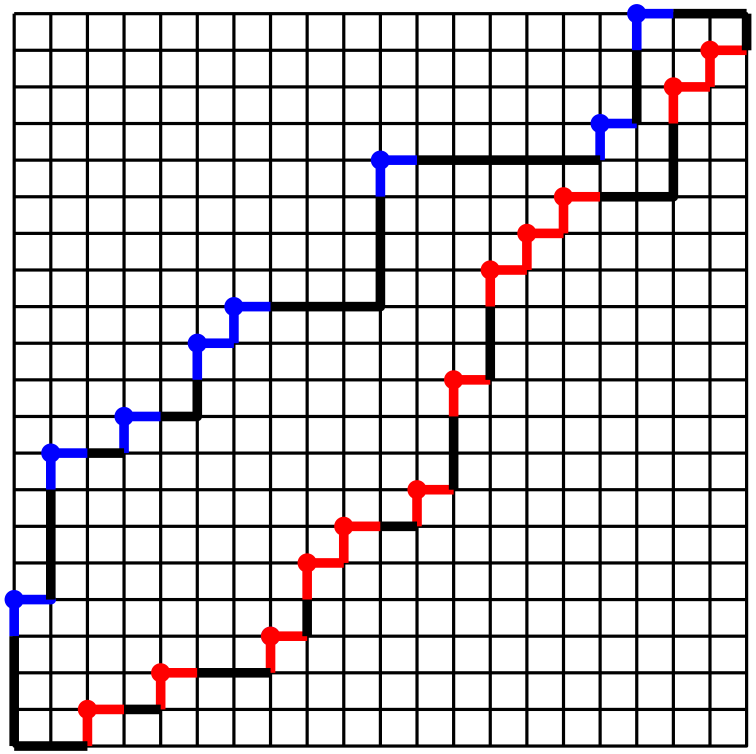


Catalan numbers and bijections



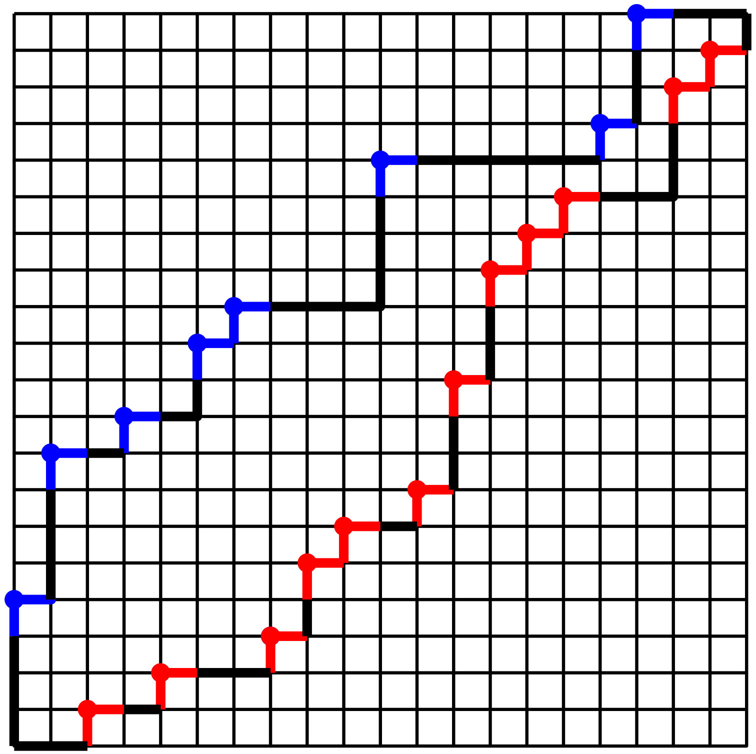
Catalan numbers and bijections

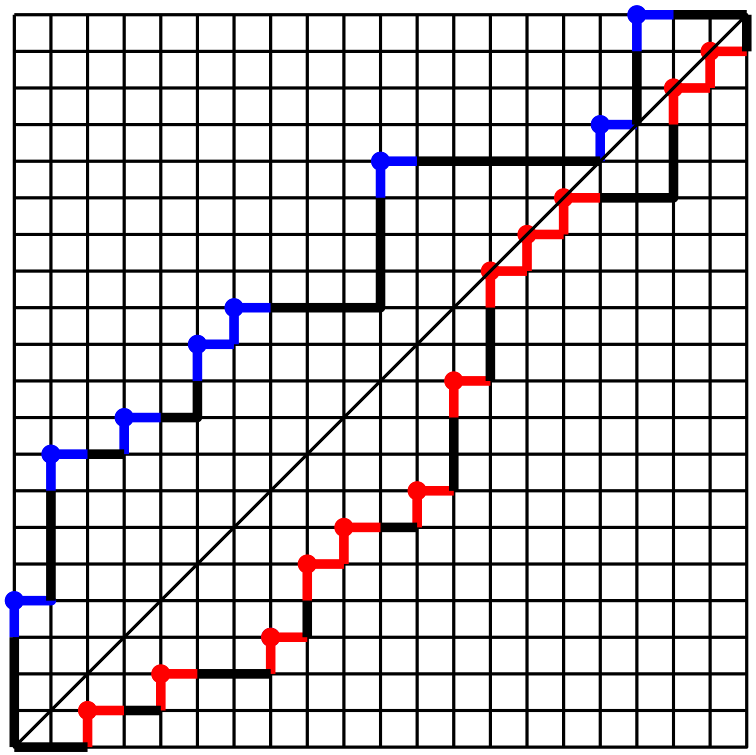
From parallelogram permutominoes of size n
to Dyck paths of semi-perimeter n.



Catalan numbers and bijections

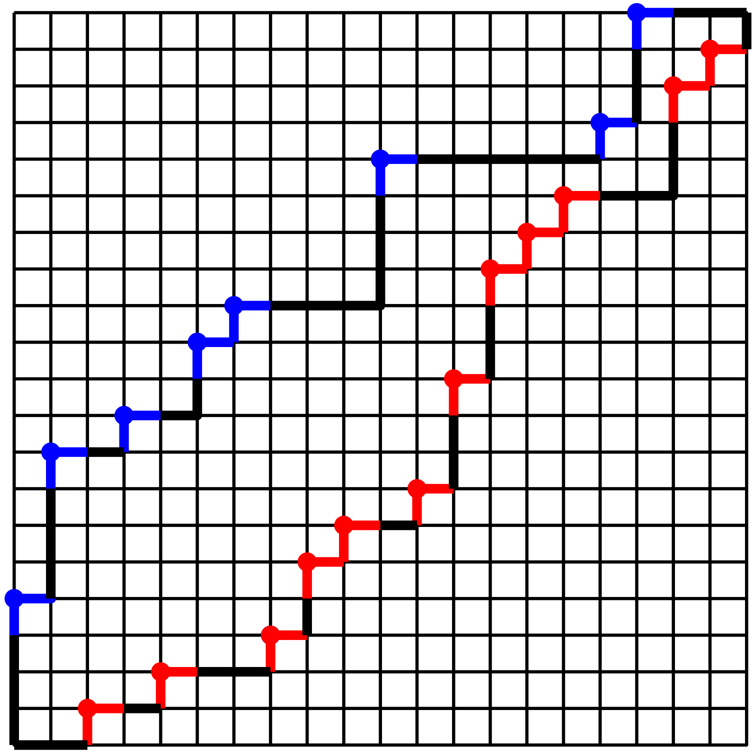
From parallelogram permutominoes of size n
to Dyck paths of semi-perimeter n.

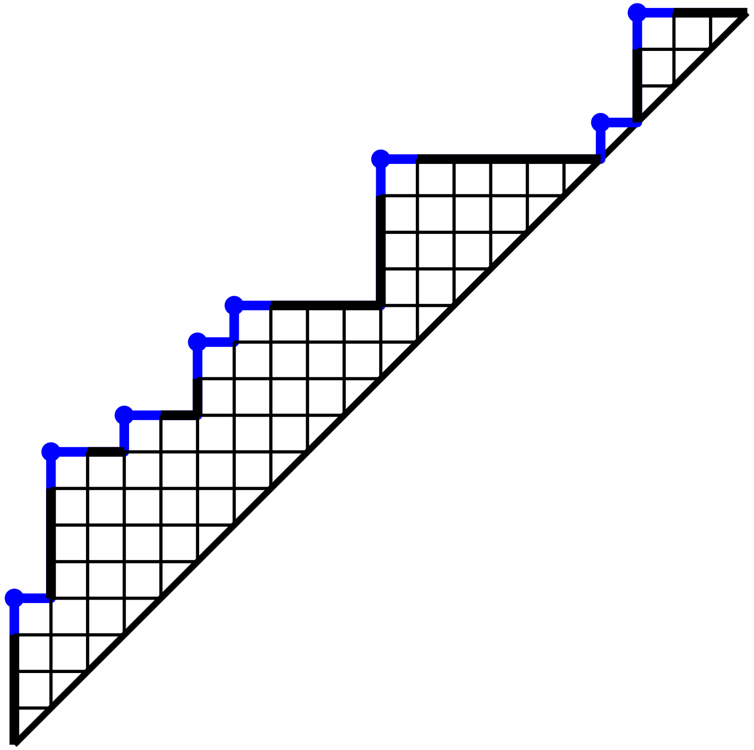




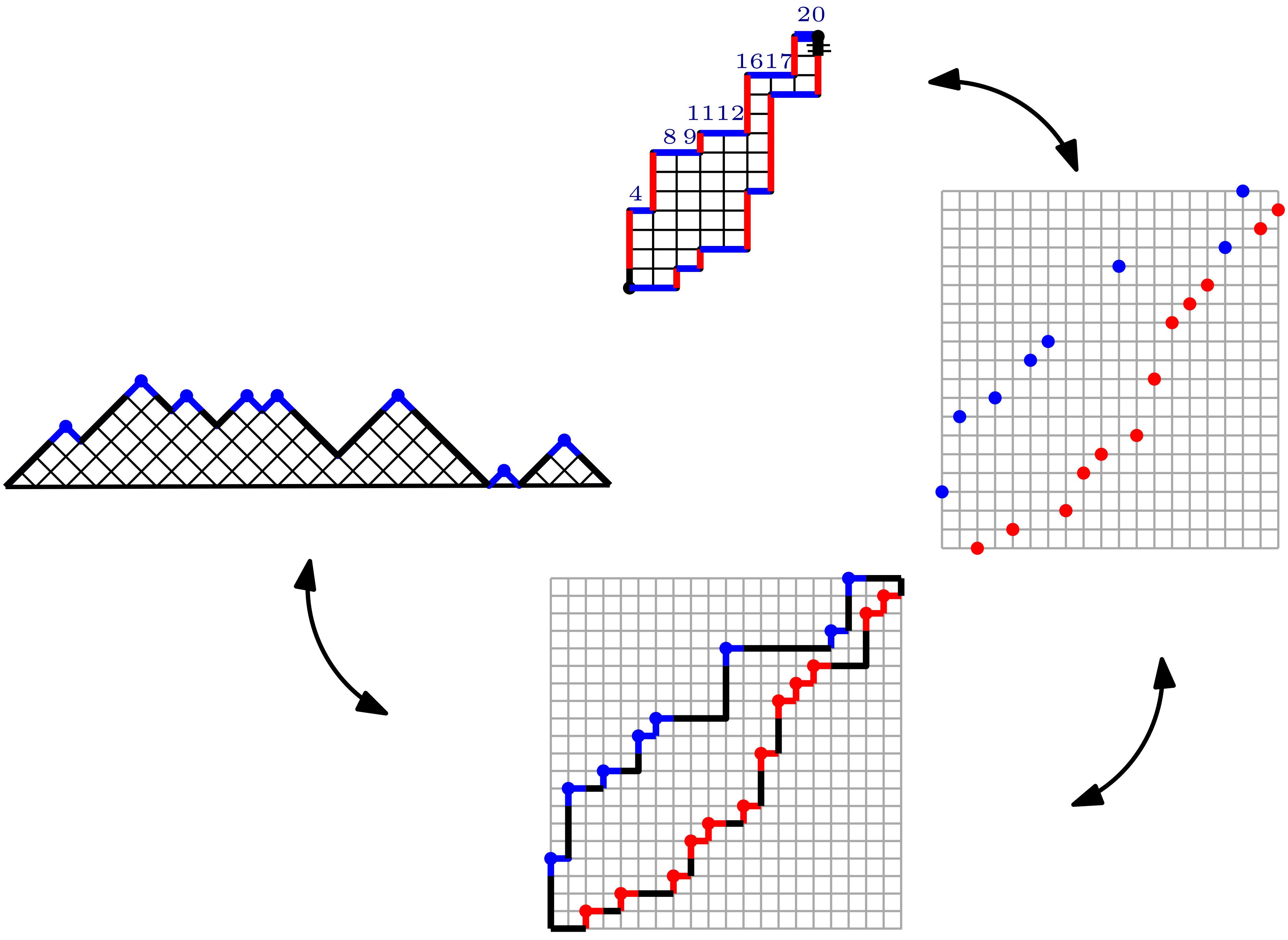
Catalan numbers and bijections

From parallelogram permutominoes of size n
to Dyck paths of semi-perimeter n.

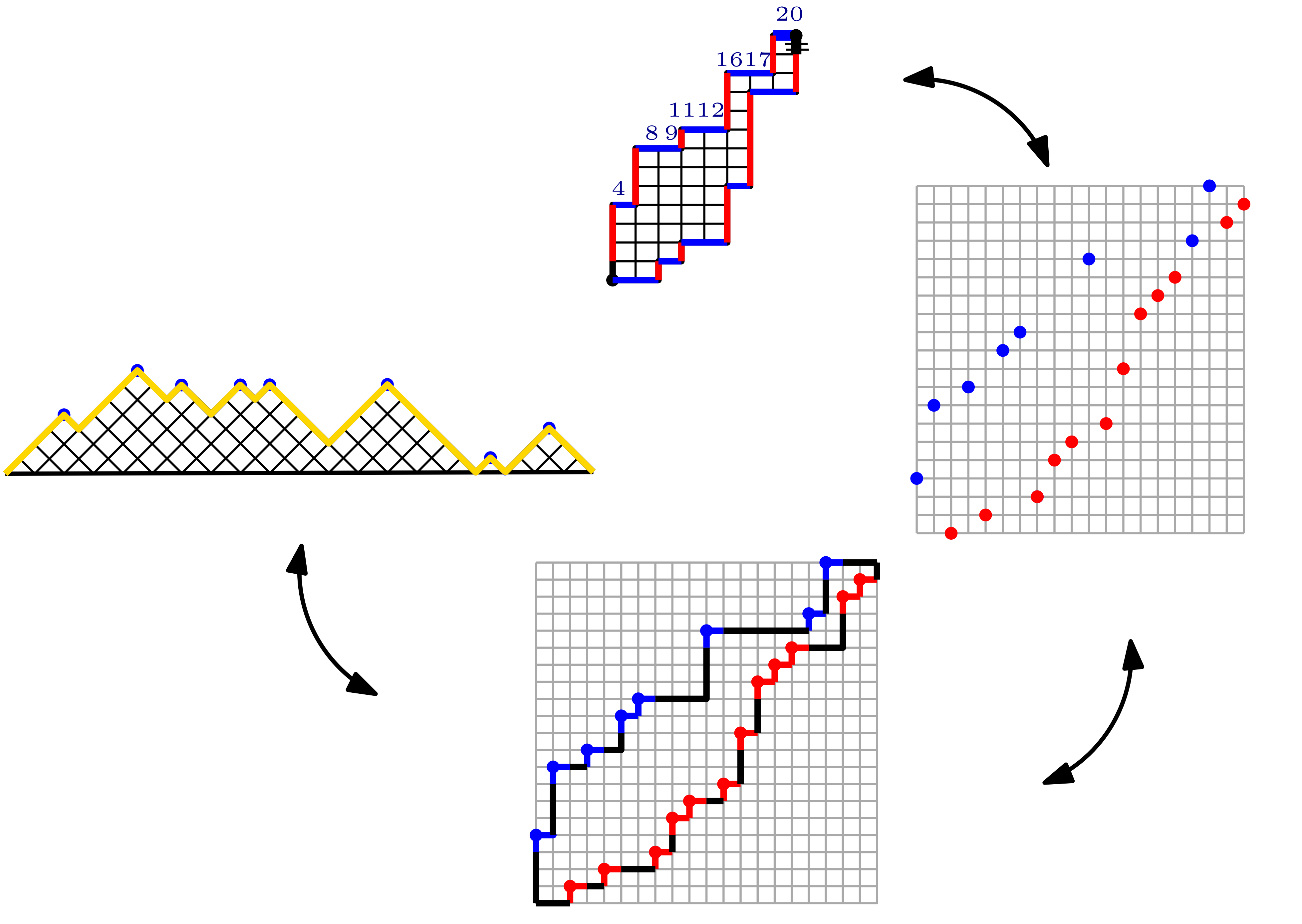




Catalan numbers and bijections

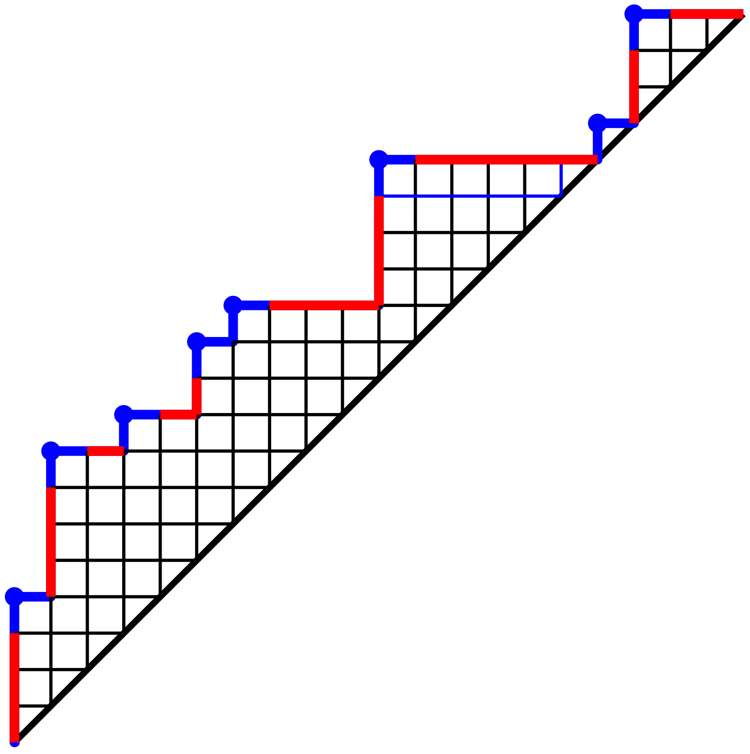


Catalan numbers and bijections



Catalan numbers and bijections

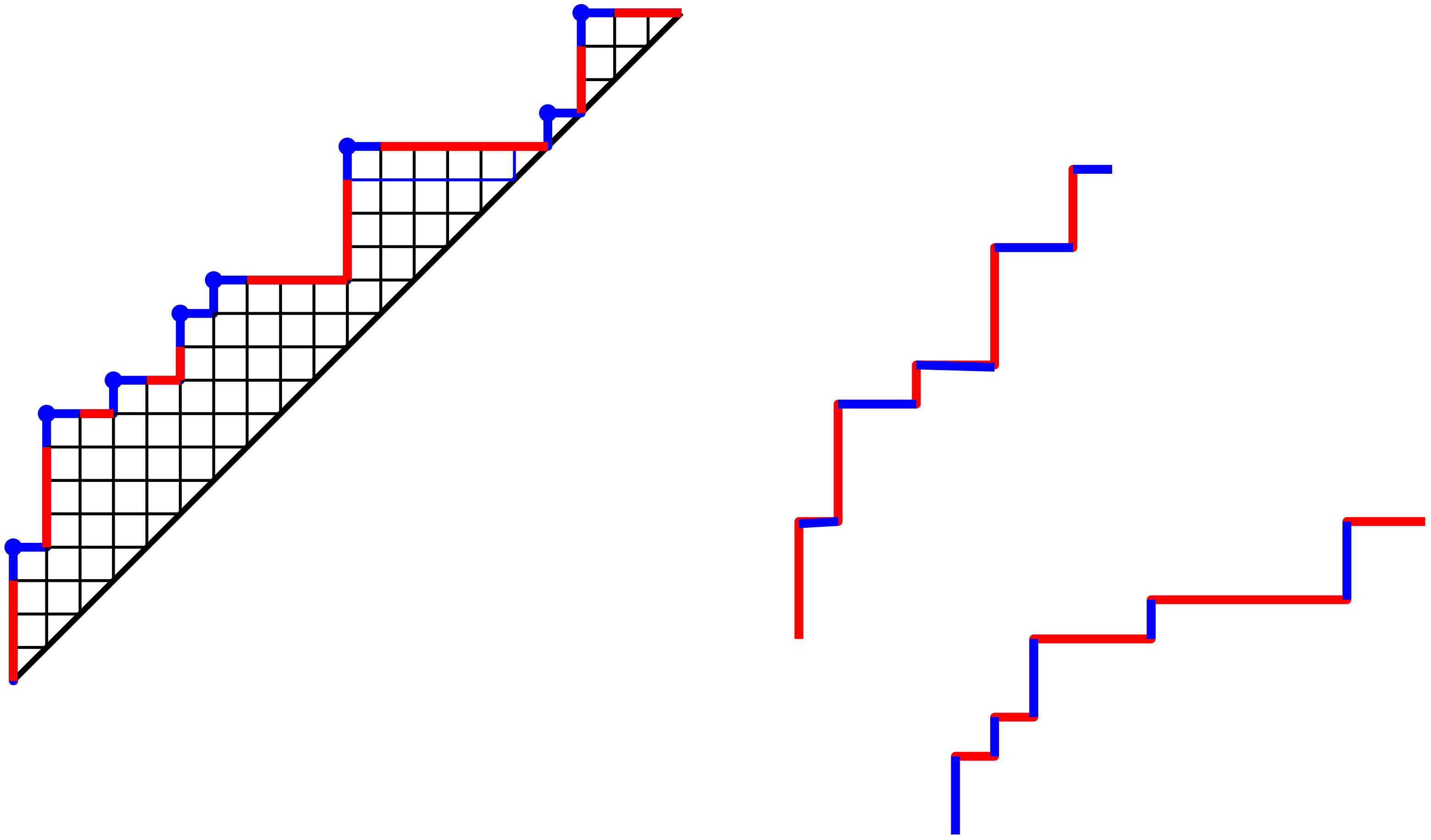
From Dyck paths of semi-perimeter n
to parallelogram polyominoes of semi-perimeter $n+1$.



Catalan numbers and bijections

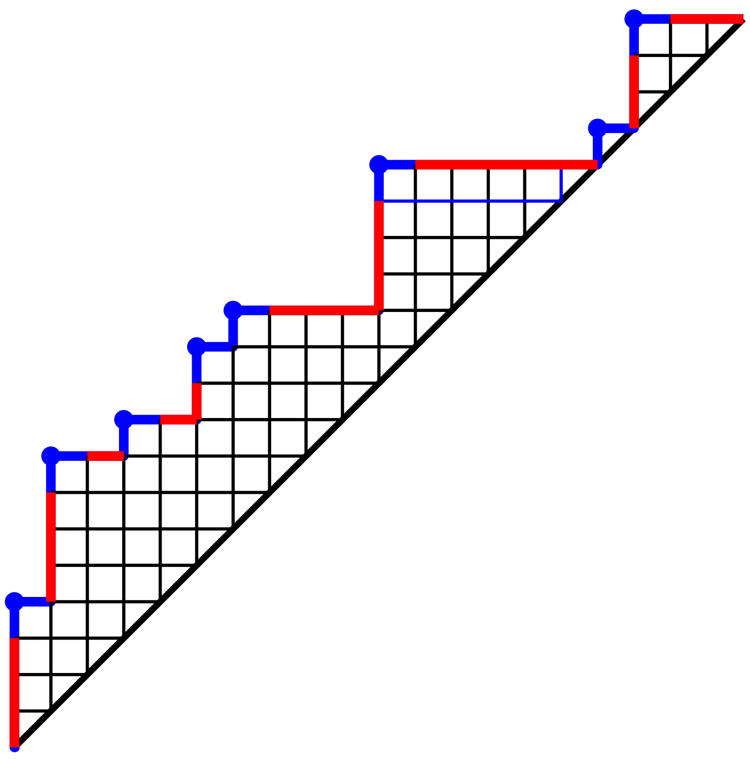
From Dyck paths of semi-perimeter n

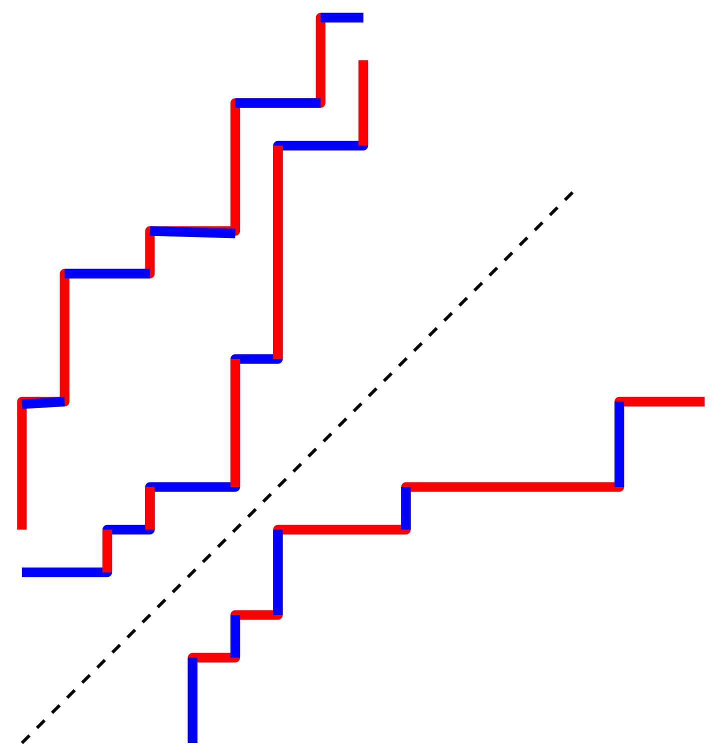
to parallelogram polyominoes of semi-perimeter $n + 1$.



Catalan numbers and bijections

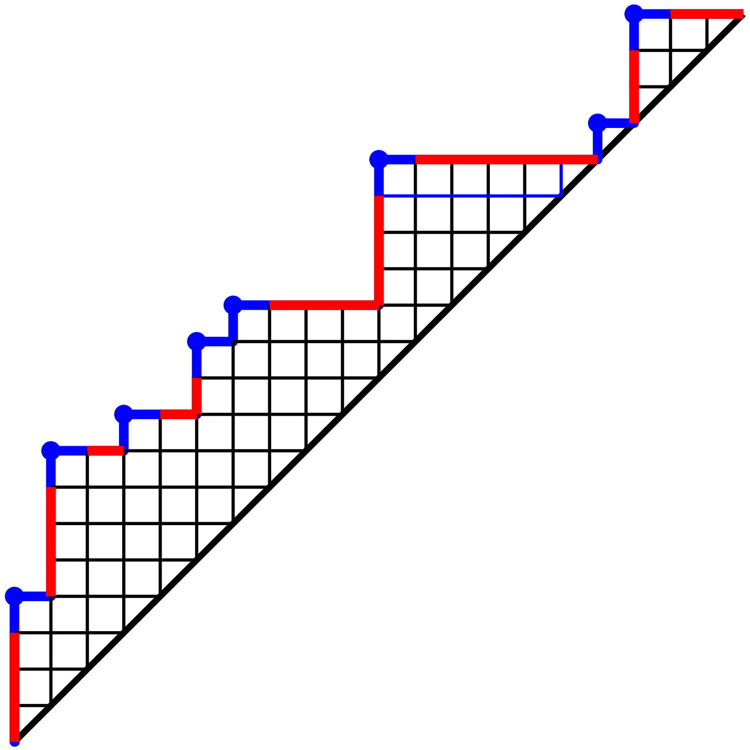
From Dyck paths of semi-perimeter n
to parallelogram polyominoes of semi-perimeter $n+1$.

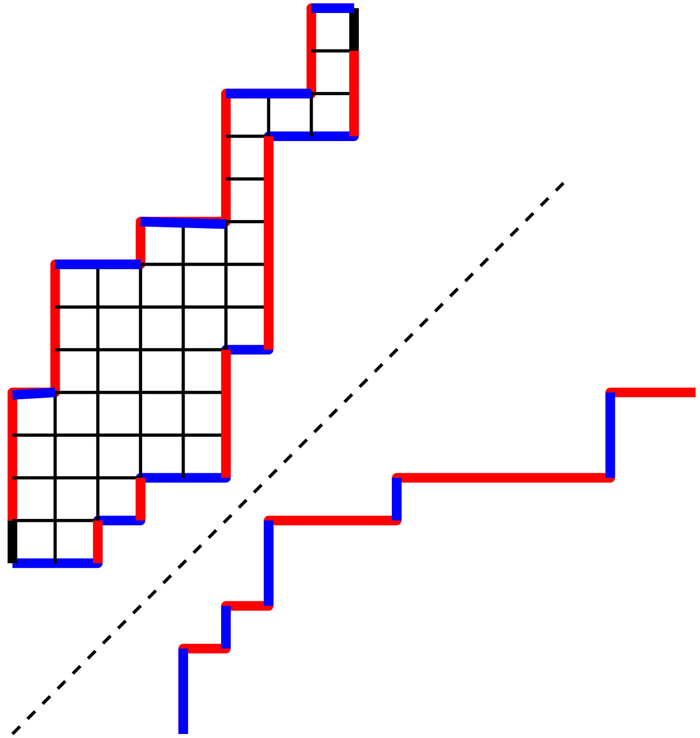




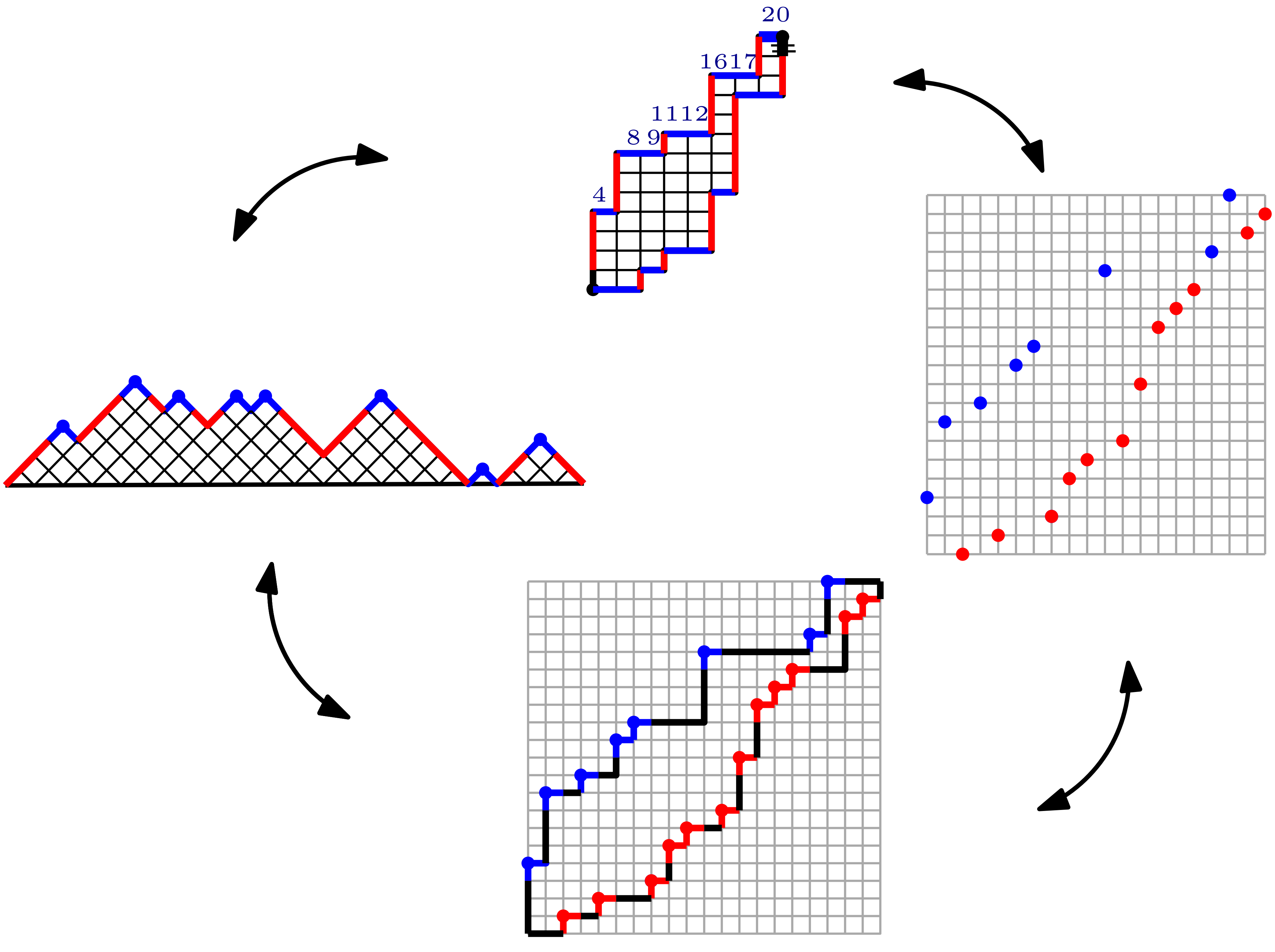
Catalan numbers and bijections

From Dyck paths of semi-perimeter n
to parallelogram polyominoes of semi-perimeter $n+1$.





Catalan numbers and bijections



Combinatorial interpretations

2) Interpretation of formulas with differences

Interpretation of Delest Viennot formula

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

$$C(t) = \frac{t^2}{1-4t}(2 + t + \frac{2t}{1-4t}) - \frac{4t^3}{(1-4t)^{3/2}}$$

Interpretation of Delest Viennot formula

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

$$C(t) = \frac{t^2}{1-4t}(2+t+\frac{2t}{1-4t}) - \frac{4t^3}{(1-4t)^{3/2}}$$

Delest / Viennot (84): famous success of Schützenberger methodology

$$\mathcal{C}_n \; = \mathcal{A}_n \cup \mathcal{B}_n$$

with $\mathcal{A}_n$, $\mathcal{B}_n$ and $\mathcal{A}_n \cap \mathcal{B}_n$ encoded by algebraic langages.

$\Rightarrow \qquad |\mathcal{C}_n| \; = |\mathcal{A}_n| + |\mathcal{B}_n| \; - \; |\mathcal{A}_n \cap \mathcal{B}_n|$

or $\qquad C(t) = F_A(t) + F_B(t) - F_{A\cap B}(t)$

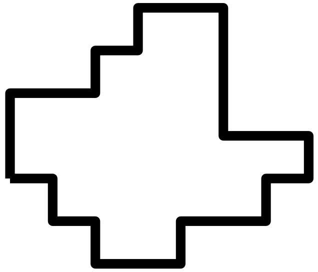
but it does not really explain the form "rational gf − algebraic gf"

Interpretation of Delest Viennot formula

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

$$C(t) = \frac{t^2}{1-4t}\left(2+t+\frac{2t}{1-4t}\right) - \frac{4t^3}{(1-4t)^{3/2}}$$

Bousquet-Mélou / Guttmann (97):



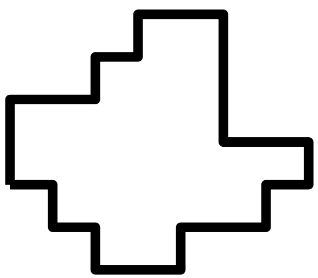
convex polyominoes

Interpretation of Delest Viennot formula

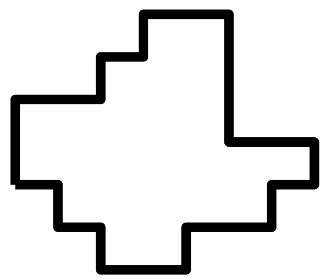
$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

$$C(t) = \frac{t^2}{1-4t}\left(2+t+\frac{2t}{1-4t}\right) - \frac{4t^3}{(1-4t)^{3/2}}$$

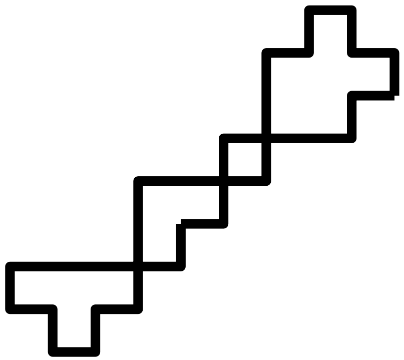
Bousquet-Mélou / Guttmann (97):



=



−



convex polyominoes

=

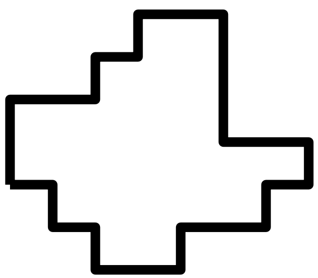
self-avoiding convex polygons = all convex polygons − self intersecting polygons

Interpretation of Delest Viennot formula

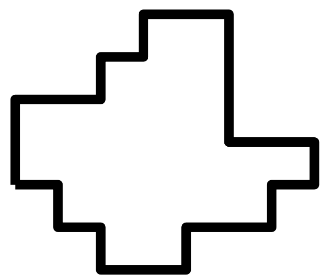
$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

$$C(t) = \tfrac{t^2}{1-4t}(2+t+\tfrac{2t}{1-4t}) - \tfrac{4t^3}{(1-4t)^{3/2}}$$

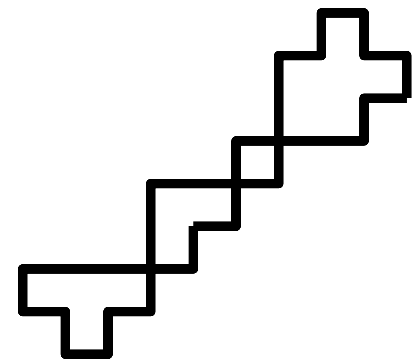
Bousquet-Mélou / Guttmann (97):



$=$



$-$



convex polyominoes

$=$

self-avoiding convex polygons $=$ all convex polygons $-$ self intersecting polygons

$=$

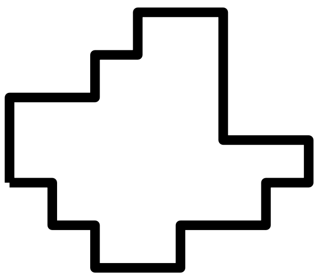
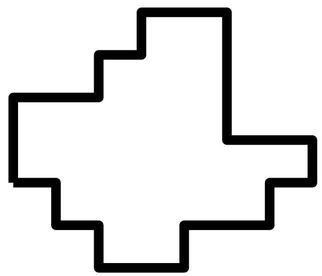
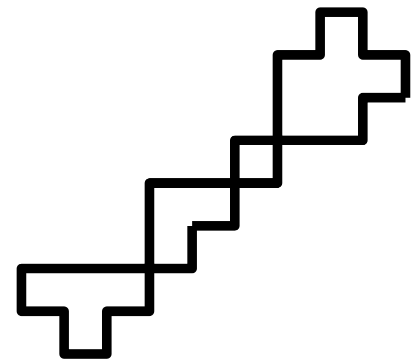
$$2 \times \mathcal{D} \times \mathrm{Seq}(\mathcal{P}) \times \mathcal{D}$$

Interpretation of Delest Viennot formula

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

$$C(t) = \tfrac{t^2}{1-4t}(2+t+\tfrac{2t}{1-4t}) - \tfrac{4t^3}{(1-4t)^{3/2}}$$

Bousquet-Mélou / Guttmann (97):

 $=$  $-$ 

convex polyominoes

$=$

self-avoiding convex polygons $=$ all convex polygons $-$ self intersecting polygons

$=$

$2 \times \mathcal{D} \times \mathrm{Seq}(\mathcal{P}) \times \mathcal{D}$

$$C(t) = L(t) - 2 \cdot D(t) \cdot \tfrac{1}{1-P(t)} \cdot D(t)$$

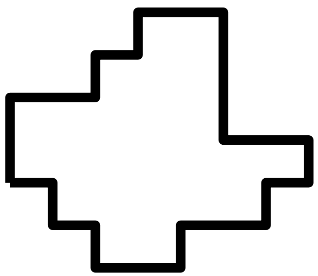
rational gf — algebraic gf

Same for convex permutominoes

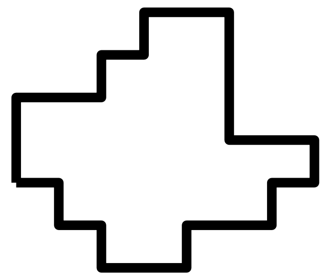
$$|\mathcal{CT}_n| = (2n+4)\,4^{n-3} - (2n-3)\binom{2n-4}{n-2}$$

$$CT(t) = \tfrac{t^2}{1-4t}(2 + \tfrac{2t}{1-4t}) - \tfrac{t^2}{(1-4t)^{3/2}}$$

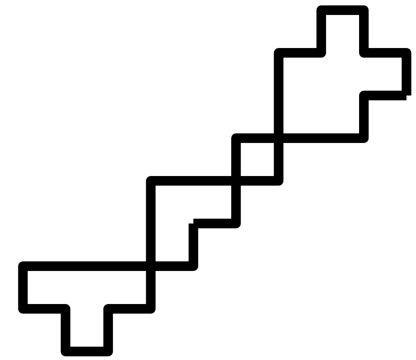
Disanto/D./Pinzani/Rinaldi (12)



=



−



convex permutominoes

=

self-avoiding convex permupolygons = all convex permupolygons − self intersecting permupolygons

$$CT(t) \quad = \quad R(t) \quad - \quad 2 \cdot DT(t) \cdot \tfrac{1}{1-PT(t)} \cdot DT(t)$$

rational gf — algebraic gf

What for square permutations ?

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

$$|\mathcal{S}_n| = (2n+4)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

$$|\mathcal{CT}_n| = (2n+4)\,4^{n-3} - (2n-3)\binom{2n-4}{n-2}$$

What for square permutations ?

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

done

$$|\mathcal{S}_n| = (2n+4)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

$$|\mathcal{CT}_n| = (2n+4)\,4^{n-3} - (2n-3)\binom{2n-4}{n-2}$$

What for square permutations ?

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

done

$$|\mathcal{S}_n| = (2n+4)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

$$|\mathcal{CT}_n| = (2n+4)\,4^{n-3} - \quad (2n-3)\binom{2n-4}{n-2}$$

$$= 2n\,4^{n-3} - \quad \left((2n-3)\binom{2n-4}{n-2} - 4^{n-2}\right)$$

What for square permutations ?

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

done

$$|\mathcal{S}_n| = (2n+4)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

now

$$|\mathcal{CT}_n| = (2n+4)\,4^{n-3} - (2n-3)\binom{2n-4}{n-2}$$

$$= 2n\,4^{n-3} - \left((2n-3)\binom{2n-4}{n-2} - 4^{n-2}\right)$$

What for square permutations ?

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

done

$$|\mathcal{S}_n| = (2n+4)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

now

$$|\mathcal{CT}_n| = (2n+4)\,4^{n-3} - (2n-3)\binom{2n-4}{n-2}$$

$$= 2n\,4^{n-3} - \left((2n-3)\binom{2n-4}{n-2} - 4^{n-2}\right)$$

$$S(t) = \frac{t^2}{1-4t}\left(2 + \frac{2t}{1-4t}\right) - 4\frac{t^3}{(1-4t)^{3/2}}$$

$$CT(t) = \frac{t^2}{1-4t}\left(2 + \frac{2t}{1-4t}\right) - \frac{t^2}{(1-4t)^{3/2}}$$

What for square permutations ?

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

done

$$|\mathcal{S}_n| = (2n+4)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

now

$$|\mathcal{CT}_n| = (2n+4)\,4^{n-3} - \quad (2n-3)\binom{2n-4}{n-2}$$

$$= 2n\,4^{n-3} - \quad \left((2n-3)\binom{2n-4}{n-2} - 4^{n-2}\right)$$

$$S(t) = \frac{t^2}{1-4t}\left(2 + \frac{2t}{1-4t}\right) - 4\frac{t^3}{(1-4t)^{3/2}}$$

$$CT(t) = \frac{t^2}{1-4t}\left(2 + \frac{2t}{1-4t}\right) - \frac{t^2}{(1-4t)^{3/2}}$$

First we need an interpretation of the rational part.

Combinatorial interpretations

2) Interpretation of formulas with differences

A code for square permutations

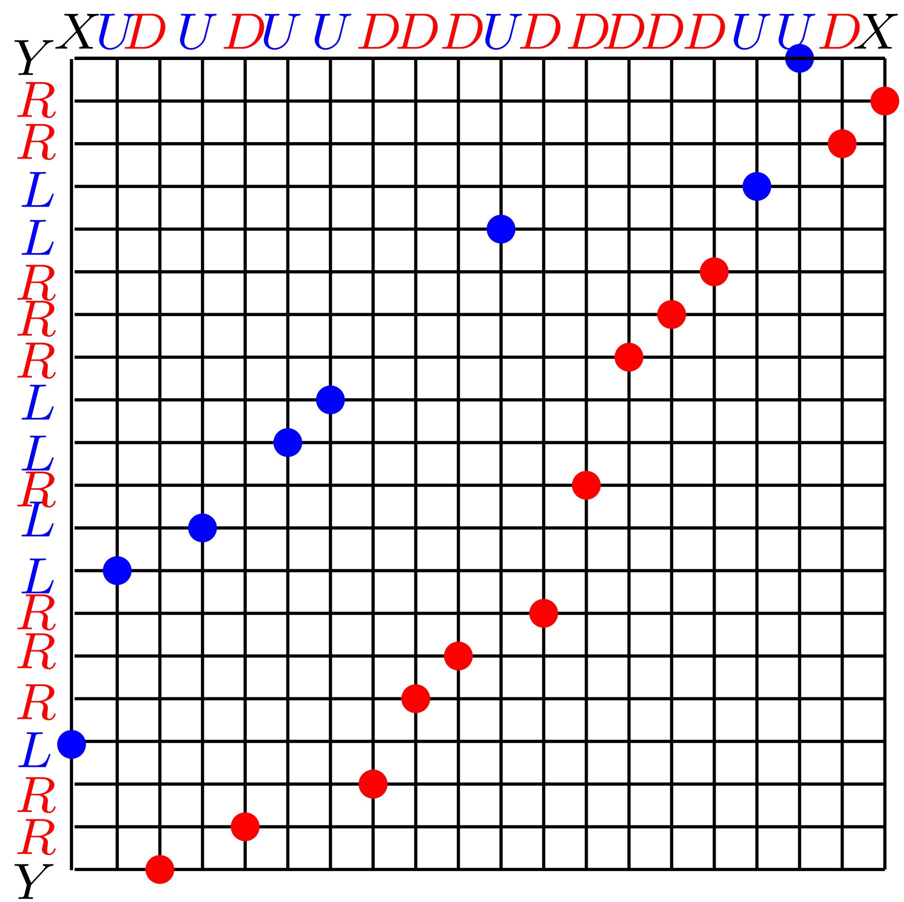
A code for square permutations

Recall the encoding we have used in the bijection between parallel permutations and parallelogram polyominoes:

The permutation was encoded by two words

- $u_1 \dots u_n$ (horizontal code)
- and $v_1 \dots v_n$ (vertical code)

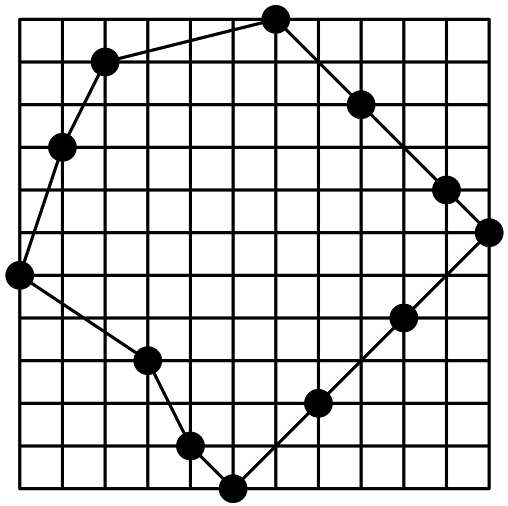
of the same length



A code for square permutations

We extend the encoding to general square permutations by constructing

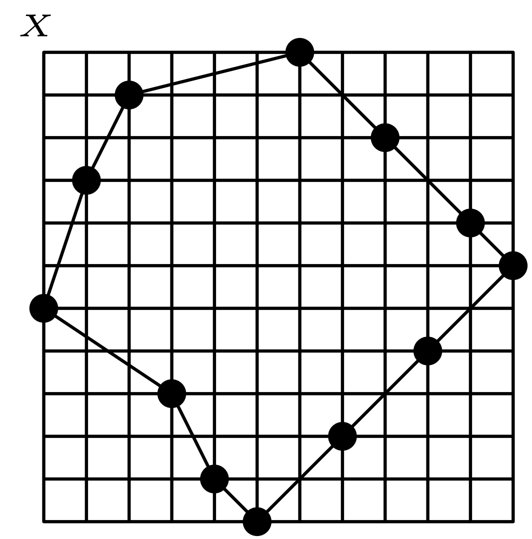
- the horizontal word $u_1 \ldots u_n$



A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \ldots u_n$

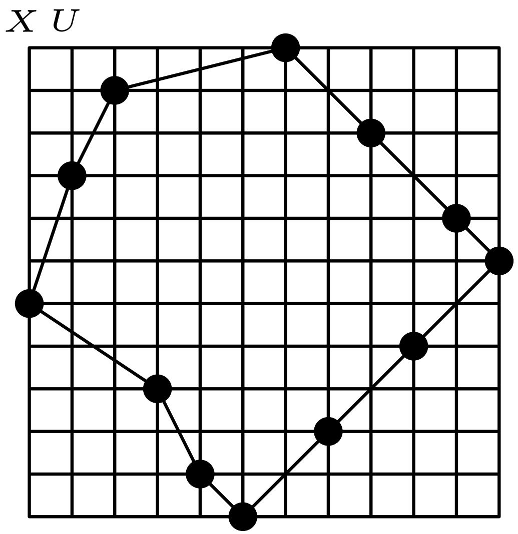


$u_i = X$ for extremal points in the vertical borders of the bounding box.

A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \dots u_n$



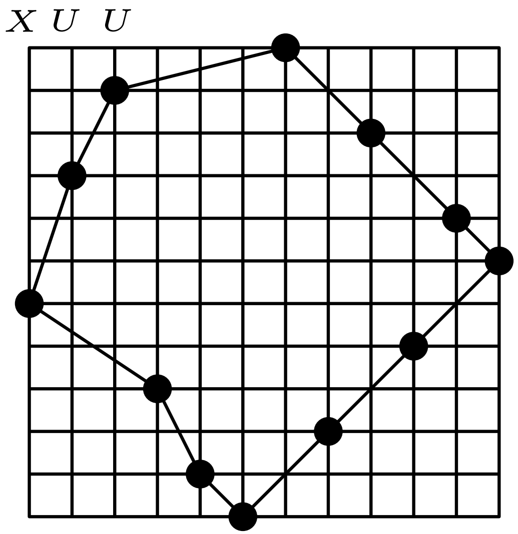
$u_i = X$ for extremal points in the vertical borders of the bounding box.

$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \dots u_n$



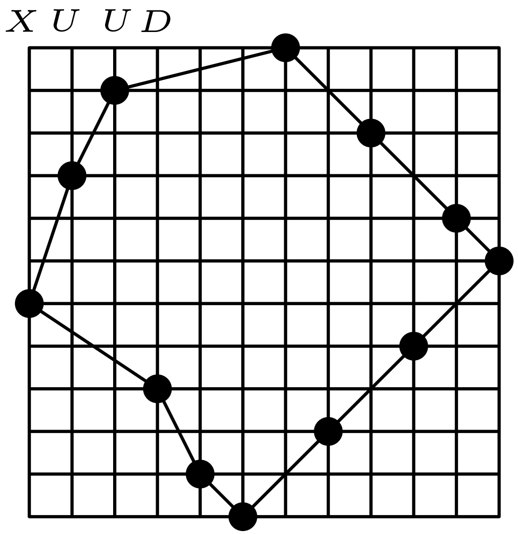
$u_i = X$ for extremal points in the vertical borders of the bounding box.

$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \dots u_n$



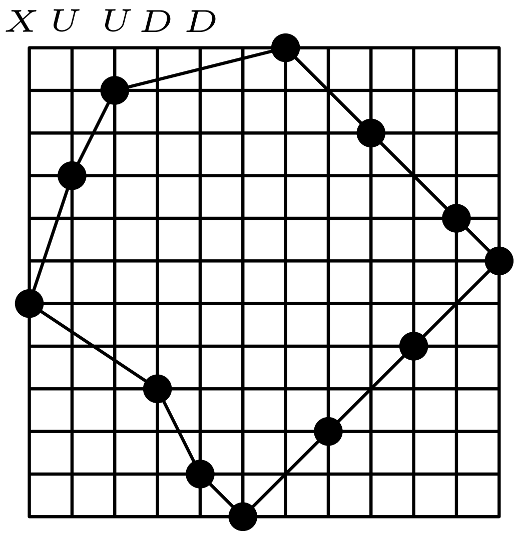
$u_i = X$ for extremal points in the vertical borders of the bounding box.

$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \dots u_n$



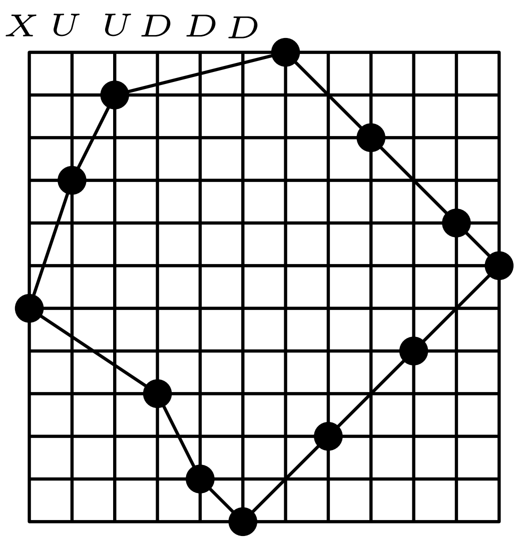
$u_i = X$ for extremal points in the vertical borders of the bounding box.

$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \dots u_n$



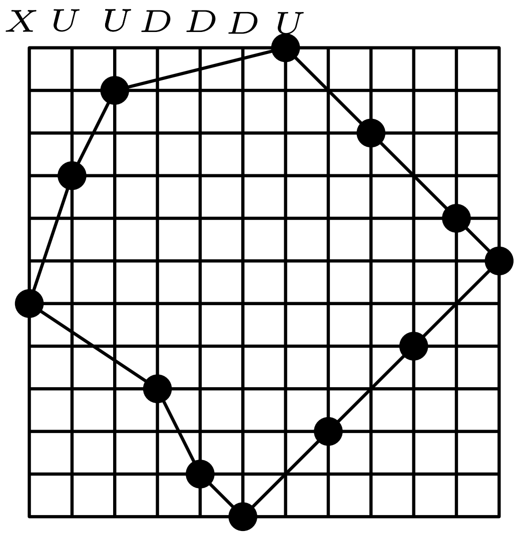
$u_i = X$ for extremal points in the vertical borders of the bounding box.

$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \ldots u_n$



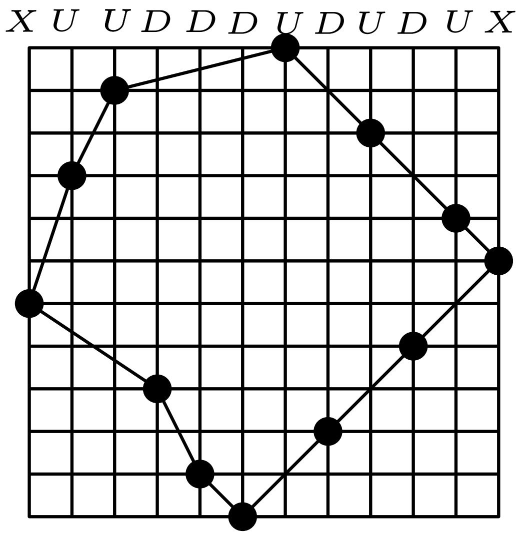
$u_i = X$ for extremal points in the vertical borders of the bounding box.

$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \ldots u_n$
- the vertical word $v_1 \ldots v_n$



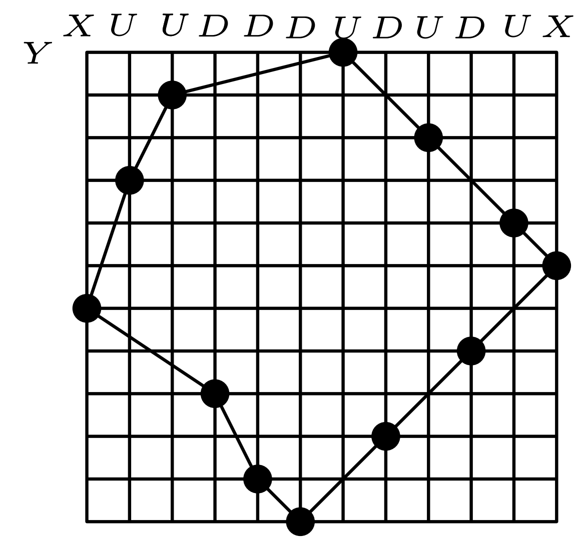
$u_i = X$ for extremal points in the vertical borders of the bounding box.

$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \dots u_n$
- the vertical word $v_1 \dots v_n$



$u_i = X$ for extremal points in the vertical borders of the bounding box.

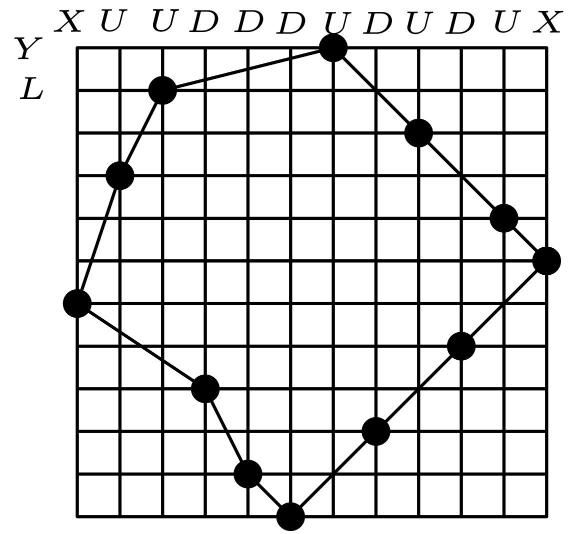
$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

$v_i = Y$ for extremal points in the horizontal borders of the bounding box.

A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \ldots u_n$
- the vertical word $v_1 \ldots v_n$



$u_i = X$ for extremal points in the vertical borders of the bounding box.

$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

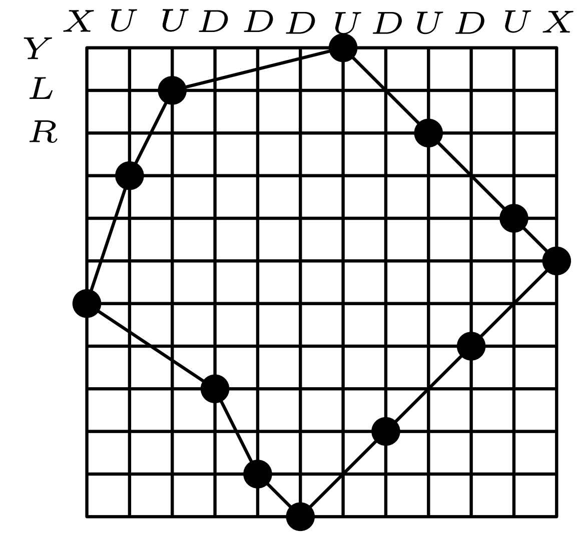
$v_i = Y$ for extremal points in the horizontal borders of the bounding box.

$$v_i = \begin{cases} L, & \text{if } (i, \sigma(i)) \text{ is left point which is not an upper right one} \\ R, & \text{otherwise} \end{cases}$$

A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \dots u_n$
- the vertical word $v_1 \dots v_n$



$u_i = X$ for extremal points in the vertical borders of the bounding box.

$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

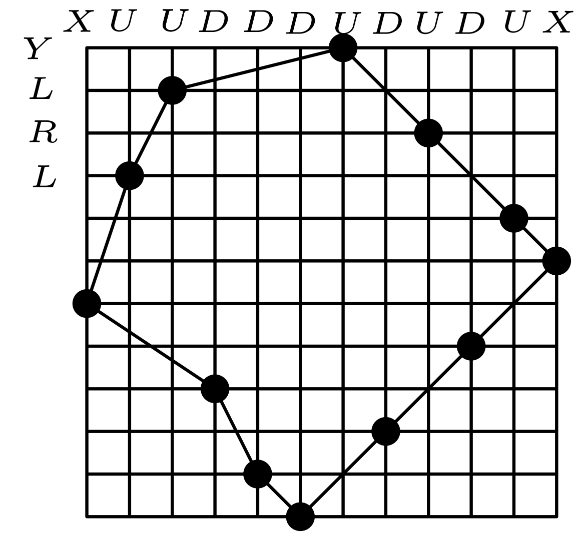
$v_i = Y$ for extremal points in the horizontal borders of the bounding box.

$$v_i = \begin{cases} L, & \text{if } (i, \sigma(i)) \text{ is left point which is not an upper right one} \\ R, & \text{otherwise} \end{cases}$$

A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \dots u_n$
- the vertical word $v_1 \dots v_n$



$u_i = X$ for extremal points in the vertical borders of the bounding box.

$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

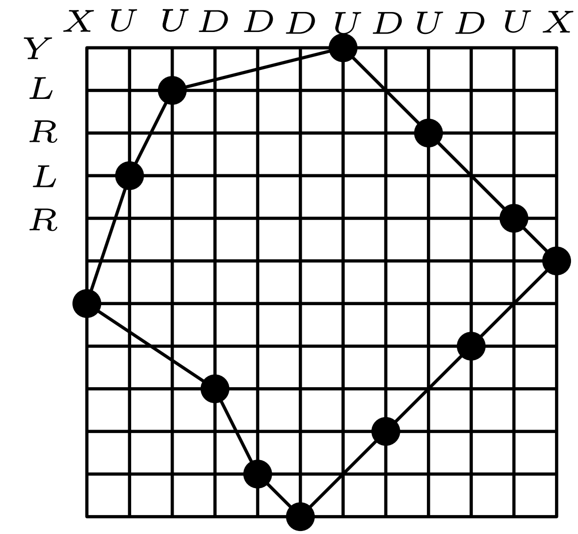
$v_i = Y$ for extremal points in the horizontal borders of the bounding box.

$$v_i = \begin{cases} L, & \text{if } (i, \sigma(i)) \text{ is left point which is not an upper right one} \\ R, & \text{otherwise} \end{cases}$$

A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \dots u_n$
- the vertical word $v_1 \dots v_n$



$u_i = X$ for extremal points in the vertical borders of the bounding box.

$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

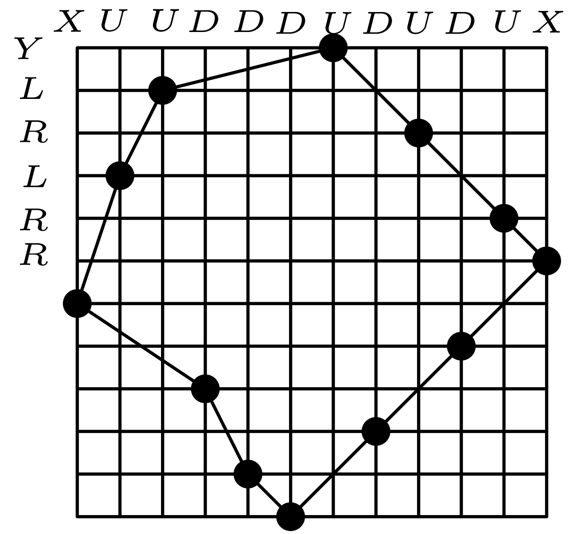
$v_i = Y$ for extremal points in the horizontal borders of the bounding box.

$$v_i = \begin{cases} L, & \text{if } (i, \sigma(i)) \text{ is left point which is not an upper right one} \\ R, & \text{otherwise} \end{cases}$$

A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \dots u_n$
- the vertical word $v_1 \dots v_n$



$u_i = X$ for extremal points in the vertical borders of the bounding box.

$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

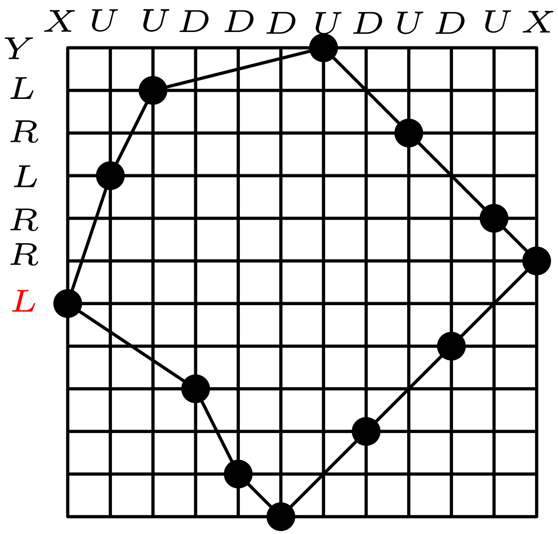
$v_i = Y$ for extremal points in the horizontal borders of the bounding box.

$$v_i = \begin{cases} L, & \text{if } (i, \sigma(i)) \text{ is left point which is not an upper right one} \\ R, & \text{otherwise} \end{cases}$$

A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \dots u_n$
- the vertical word $v_1 \dots v_n$



$u_i = X$ for extremal points in the vertical borders of the bounding box.

$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

$v_i = Y$ for extremal points in the horizontal borders of the bounding box.

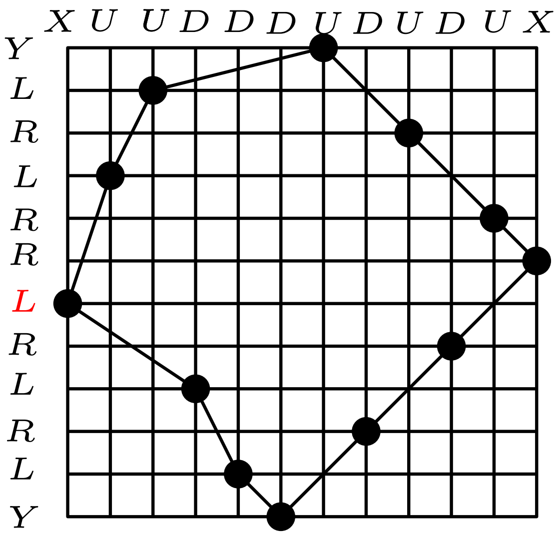
$$v_i = \begin{cases} L, & \text{if } (i, \sigma(i)) \text{ is left point which is not an upper right one} \\ R, & \text{otherwise} \end{cases}$$

A code for square permutations

We extend the encoding to general square permutations by constructing

- the horizontal word $u_1 \dots u_n$
- the vertical word $v_1 \dots v_n$

and by marking one letter L or X.



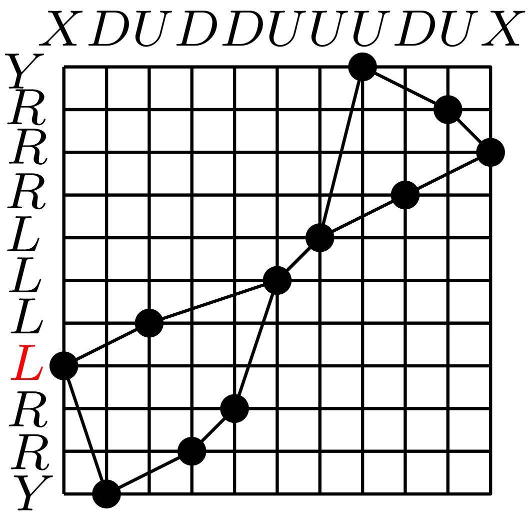
$u_i = X$ for extremal points in the vertical borders of the bounding box.

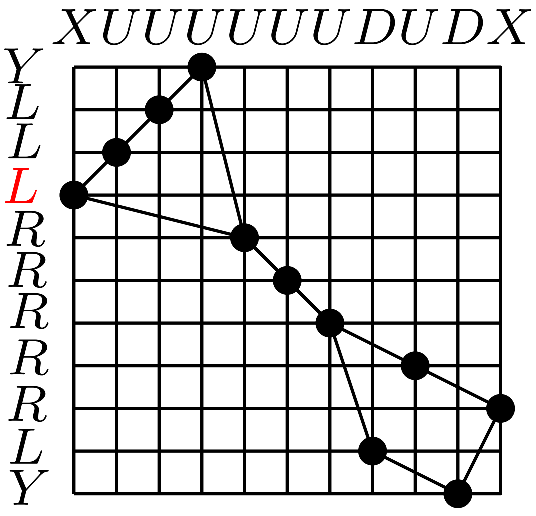
$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

$v_i = Y$ for extremal points in the horizontal borders of the bounding box.

$$v_i = \begin{cases} L, & \text{if } (i, \sigma(i)) \text{ is left point which is not an upper right one} \\ R, & \text{otherwise} \end{cases}$$

A code for square permutations





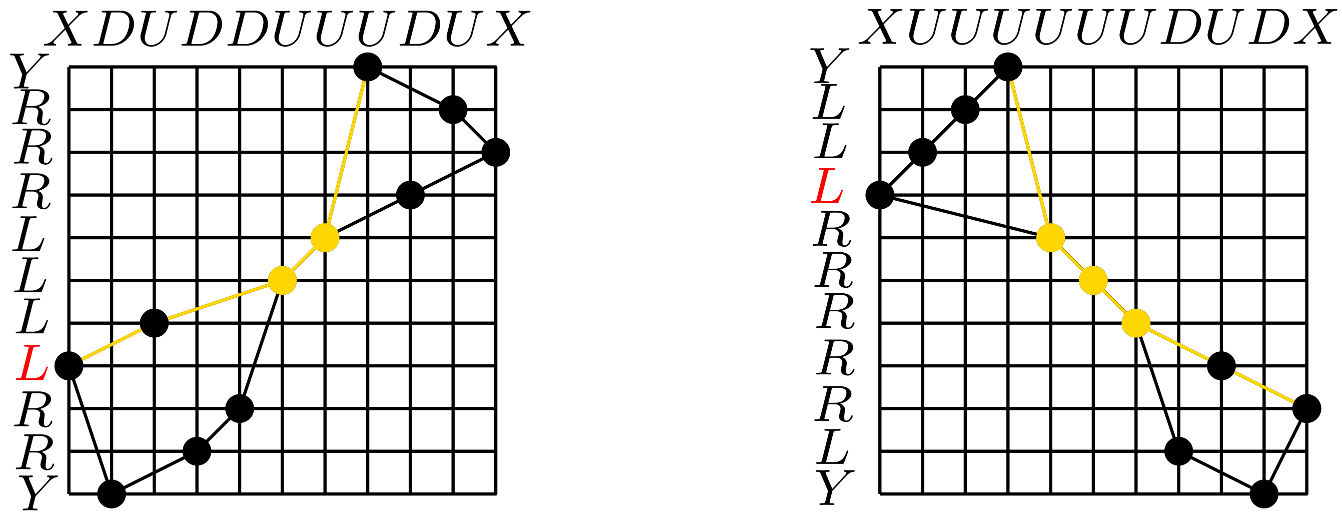
$u_i = X$ for extremal points in the vertical borders of the bounding box.

$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

$v_i = Y$ for extremal points in the horizontal borders of the bounding box.

$$v_i = \begin{cases} L, & \text{if } (i, \sigma(i)) \text{ is left point which is not an upper right one} \\ R, & \text{otherwise} \end{cases}$$

A code for square permutations



$u_i = X$ for extremal points in the vertical borders of the bounding box.

$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point} \\ D, & \text{otherwise} \end{cases}$$

$v_i = Y$ for extremal points in the horizontal borders of the bounding box.

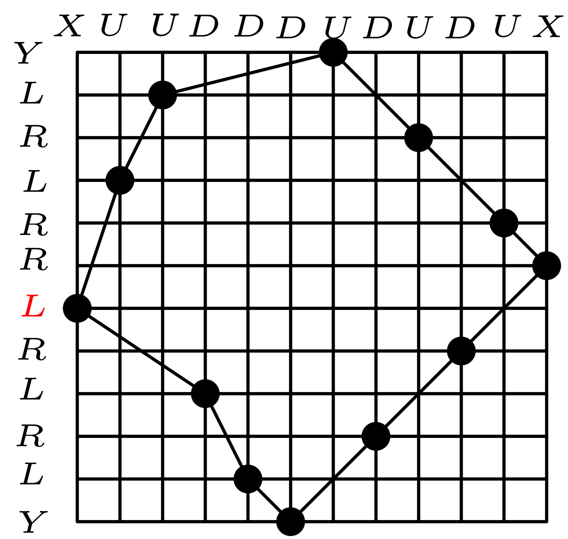
$$v_i = \begin{cases} L, & \text{if } (i, \sigma(i)) \text{ is left point which is not an upper right one} \\ R, & \text{otherwise} \end{cases}$$

A code for square permutations

Let $\mathcal{W} = \mathcal{A}^*$ the set of (bi)words on the alphabet $\mathcal{A} = \{U, D\} \times \{L, R\}$

Then our codes can be viewed as elements of the set $\mathcal{M}$ of marked biwords (w, m) of the form:

- $w = (X, Y) \cdot w' \cdot (X, Y)$ with $w' \in \mathcal{W}$
- $1 \leq m \leq n$ and $v_m = (L, Y)$



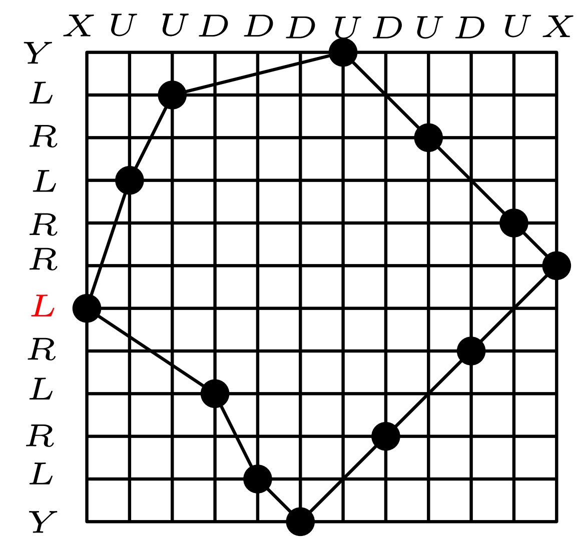
The set of marked words $\mathcal{M}$ gives an interpretation for rational part of the formula. Indeed

$$|\mathcal{M}_n| = (n-2) \cdot 2^{n-3} \cdot 2^{n-2} + 2 \cdot 2^{n-2} \cdot 2^{n-2} = (2n+4) \cdot 4^{n-3}$$

$$M(t) = \frac{t^2}{1-4t}(2 + \frac{2t}{1-4t})$$

A code for square permutations

Let $\mathcal{W} = \mathcal{A}^*$ the set of (bi)words on the alphabet $\mathcal{A} = \{U, D\} \times \{L, R\}$



Then our codes can be viewed as elements of the set $\mathcal{M}$ of marked biwords (w, m) of the form:

- $w = (X, Y) \cdot w' \cdot (X, Y)$ with $w' \in \mathcal{W}$
- $1 \leq m \leq n$ and $v_m = (L, Y)$

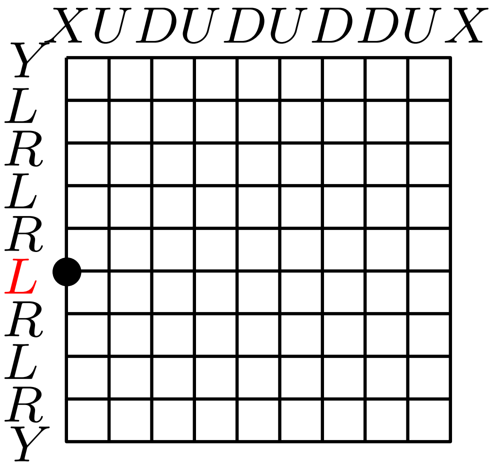
The set of marked words $\mathcal{M}$ gives an interpretation for rational part of the formula. Indeed

$$|\mathcal{M}_n| = (n-2) \cdot 2^{n-3} \cdot 2^{n-2} + 2 \cdot 2^{n-2} \cdot 2^{n-2} = (2n+4) \cdot 4^{n-3}$$

$$M(t) = \frac{t^2}{1-4t}\left(2 + \frac{2t}{1-4t}\right)$$

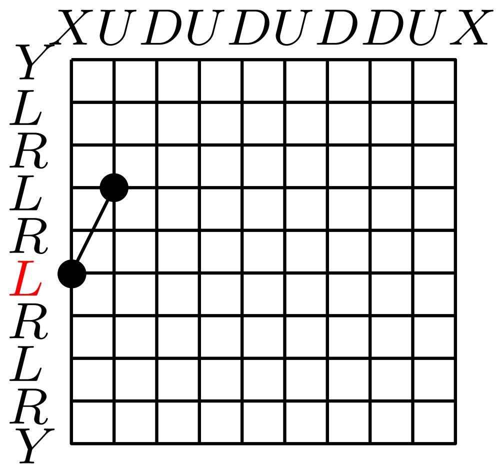
Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



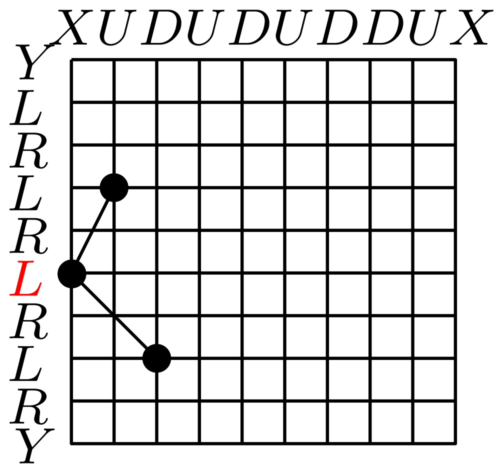
Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



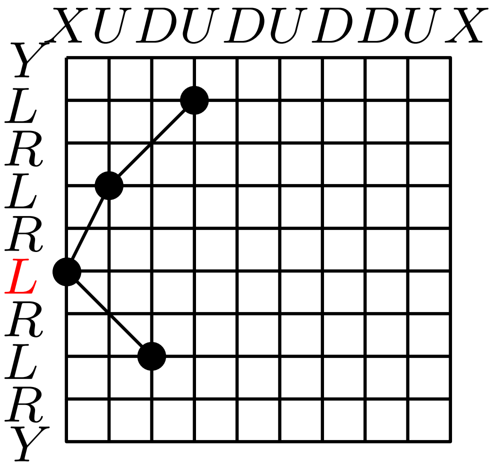
Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



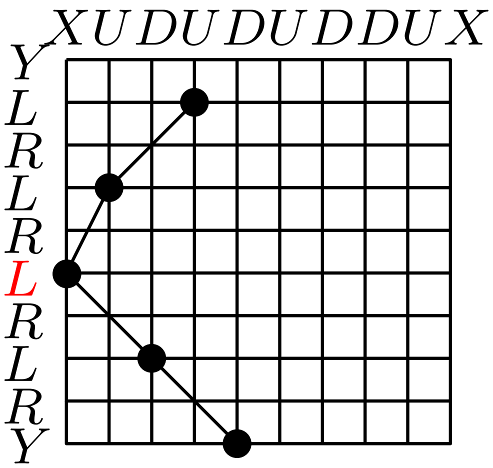
Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



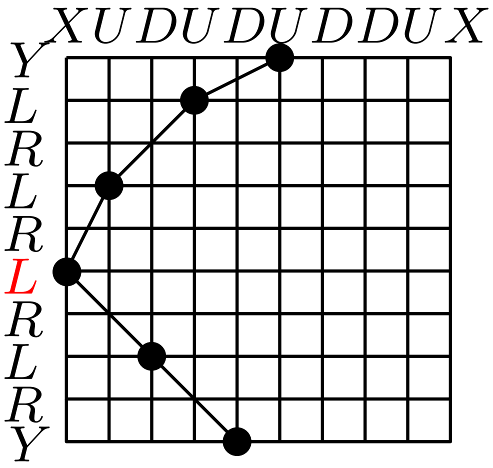
Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



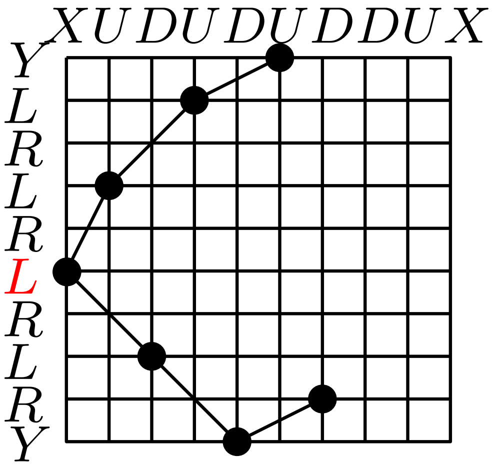
Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



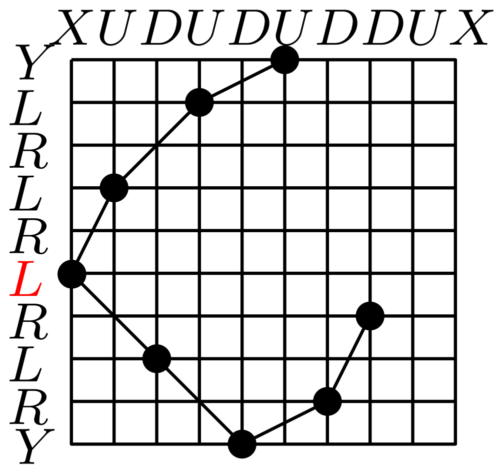
Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



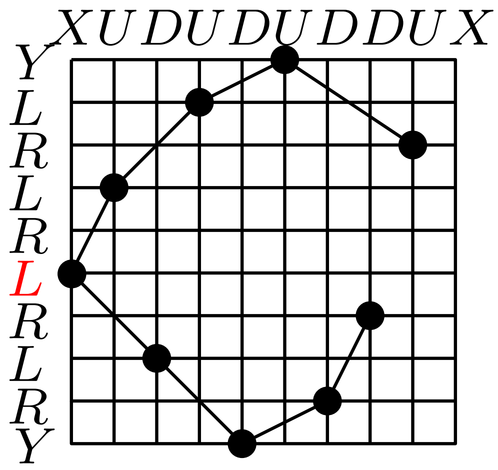
Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



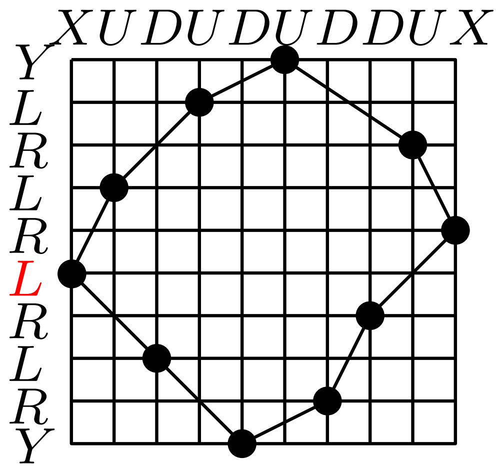
Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



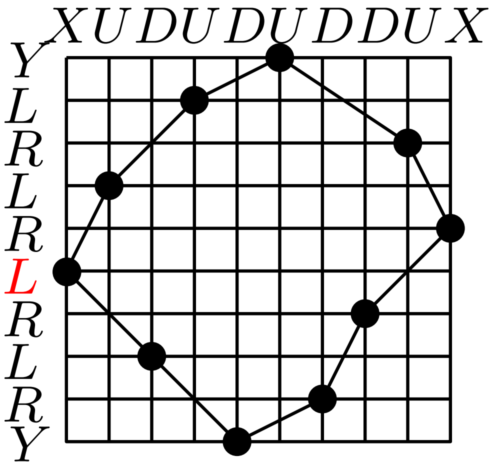
Decoding a word of $\mathcal{M}$

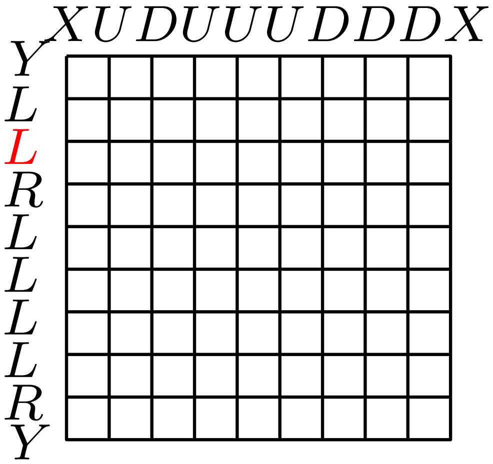
Consider a marked bi-word $(w, m) \in \mathcal{M}$



Decoding a word of $\mathcal{M}$

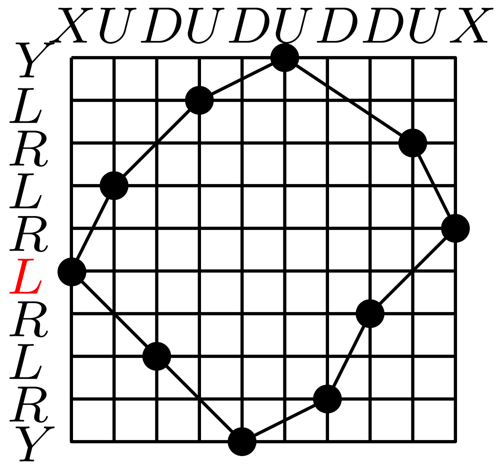
Consider a marked bi-word $(w, m) \in \mathcal{M}$

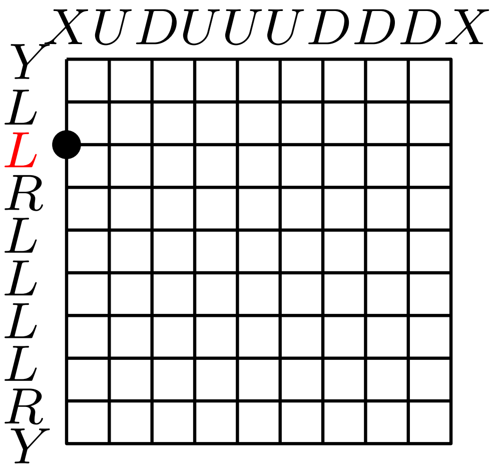




Decoding a word of $\mathcal{M}$

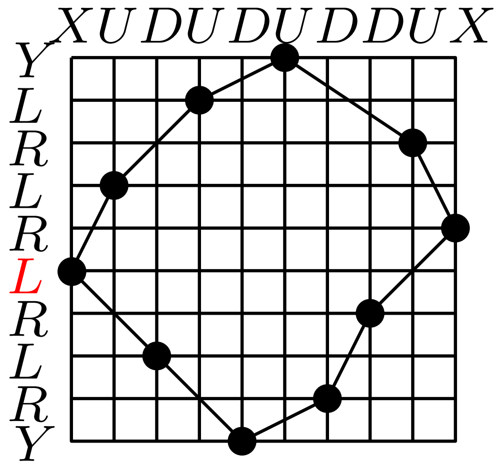
Consider a marked bi-word $(w, m) \in \mathcal{M}$

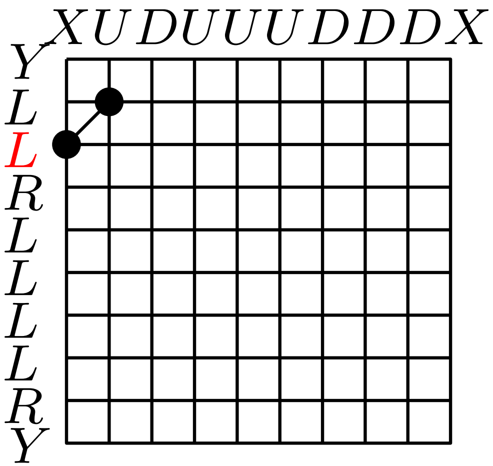




Decoding a word of $\mathcal{M}$

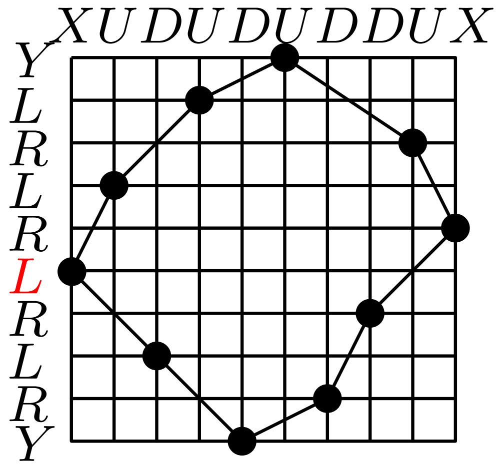
Consider a marked bi-word $(w, m) \in \mathcal{M}$

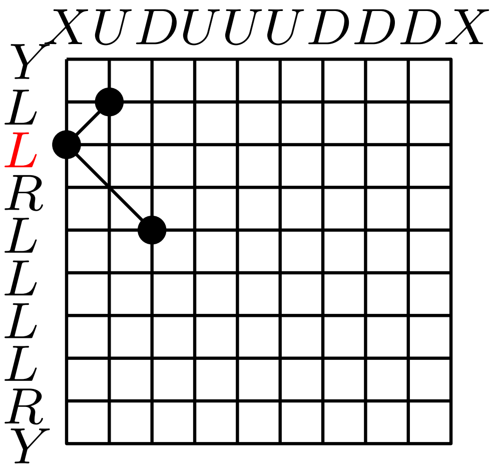




Decoding a word of $\mathcal{M}$

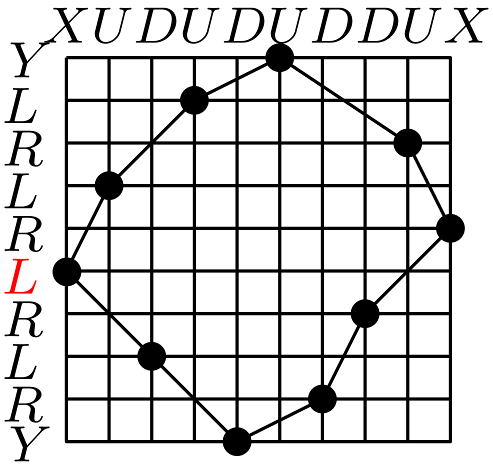
Consider a marked bi-word $(w, m) \in \mathcal{M}$

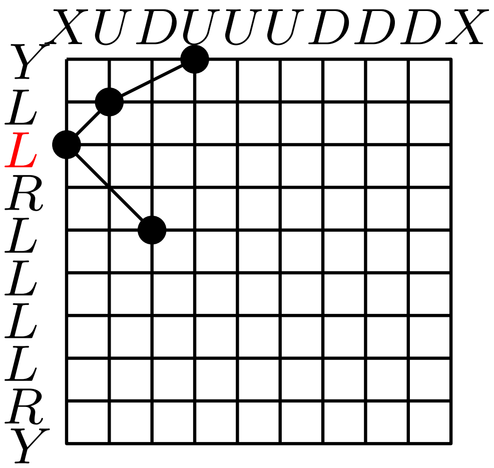




Decoding a word of $\mathcal{M}$

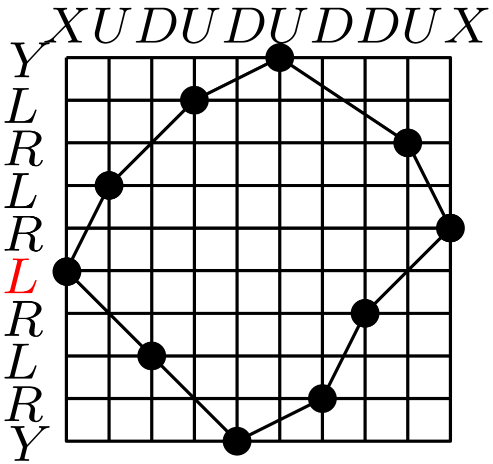
Consider a marked bi-word $(w, m) \in \mathcal{M}$

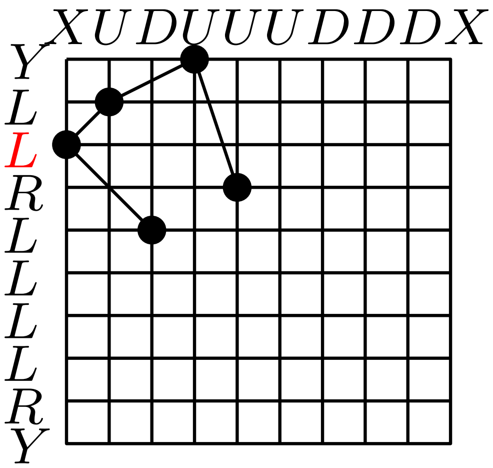




Decoding a word of $\mathcal{M}$

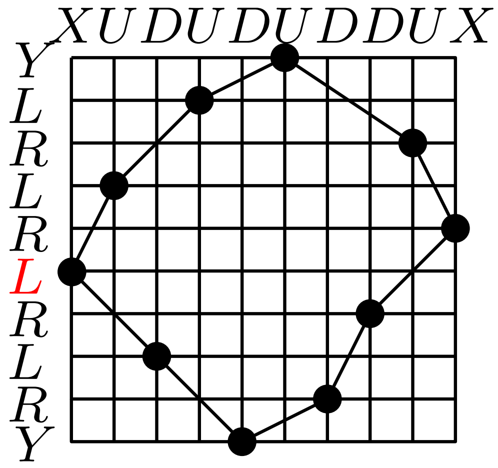
Consider a marked bi-word $(w, m) \in \mathcal{M}$

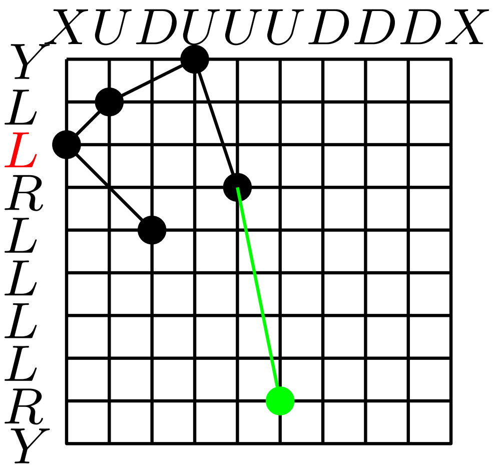




Decoding a word of $\mathcal{M}$

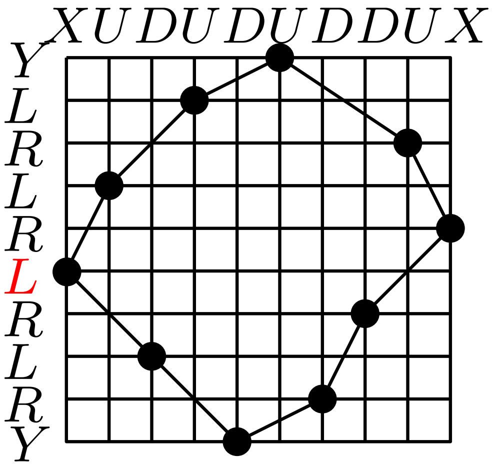
Consider a marked bi-word $(w, m) \in \mathcal{M}$

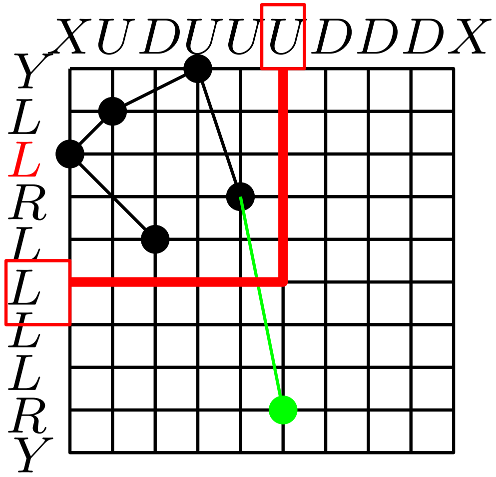




Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$

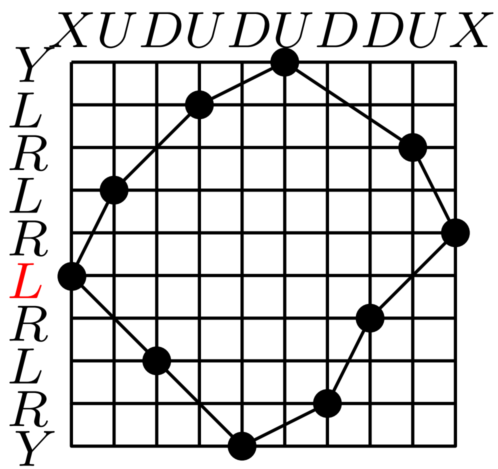


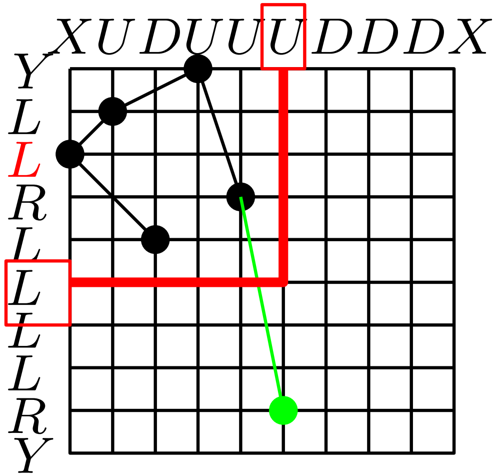


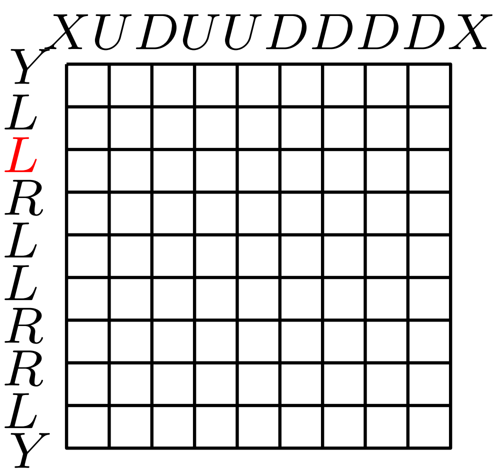
$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



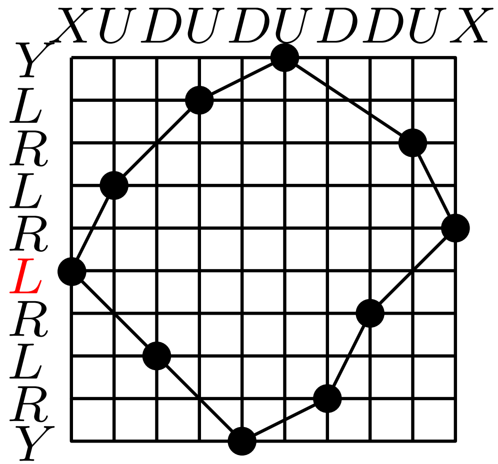


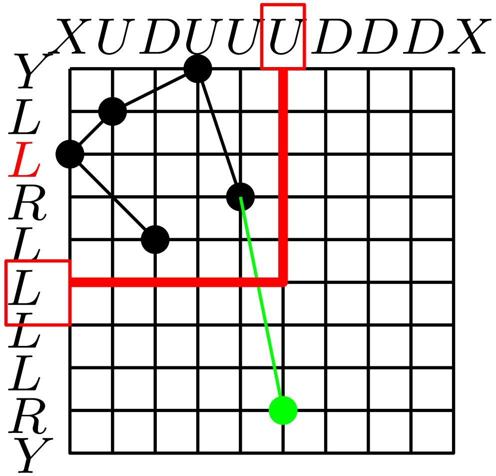


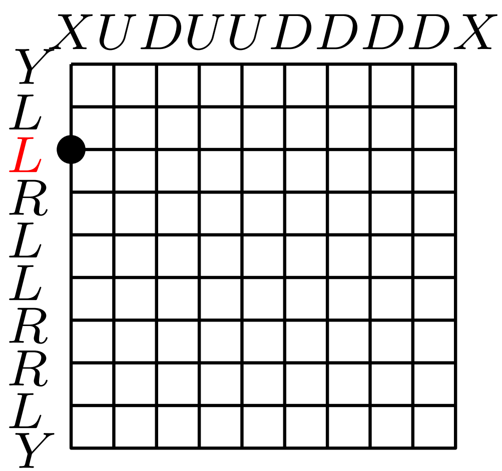
$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



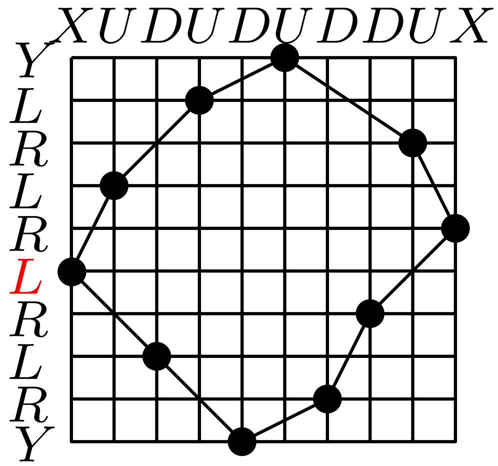


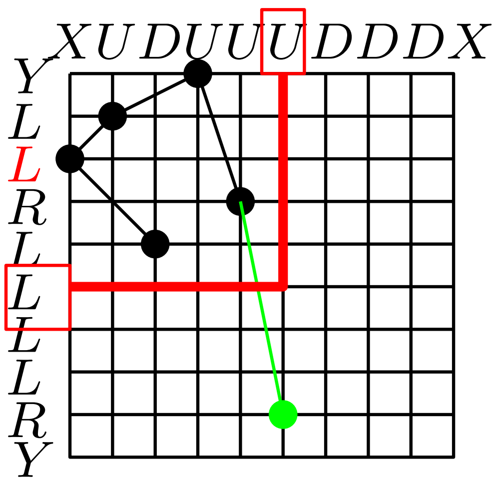


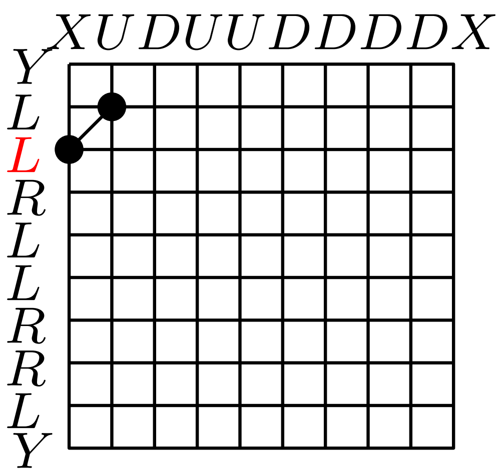
$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



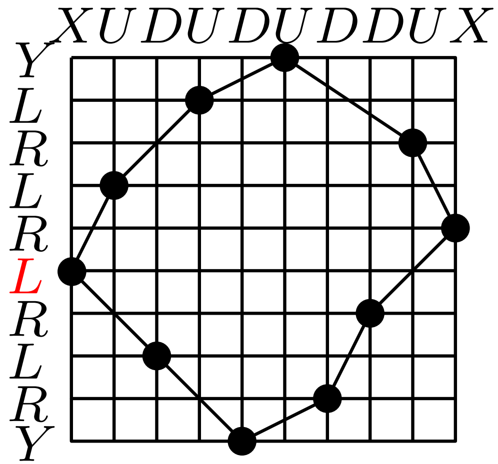


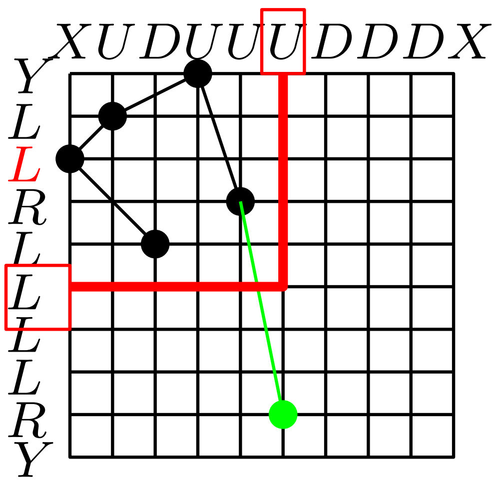


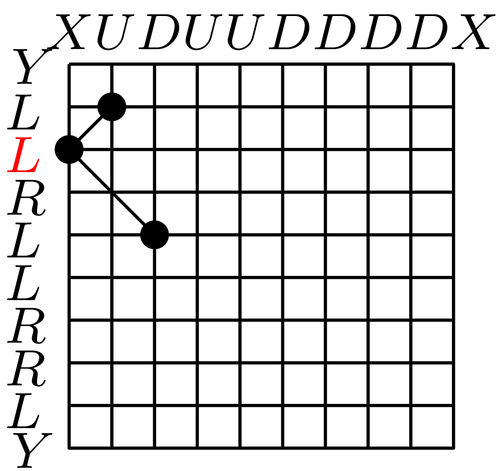
$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



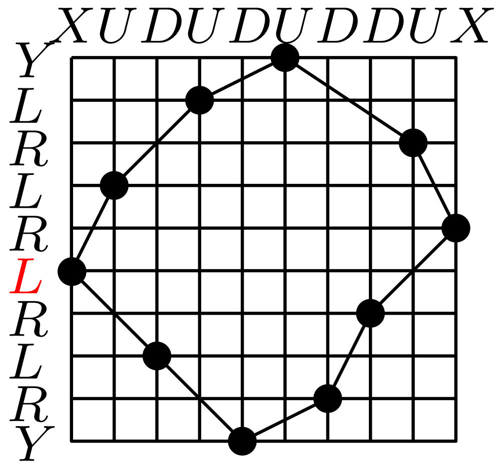


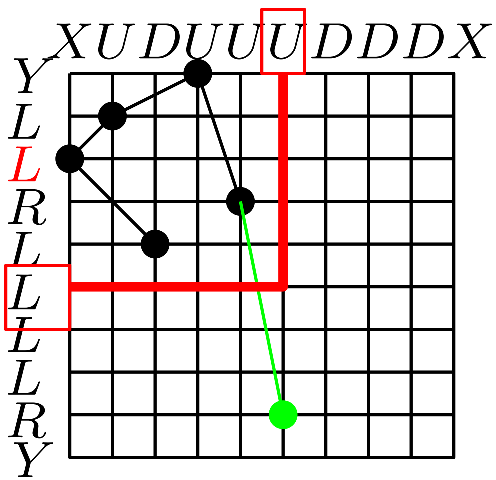


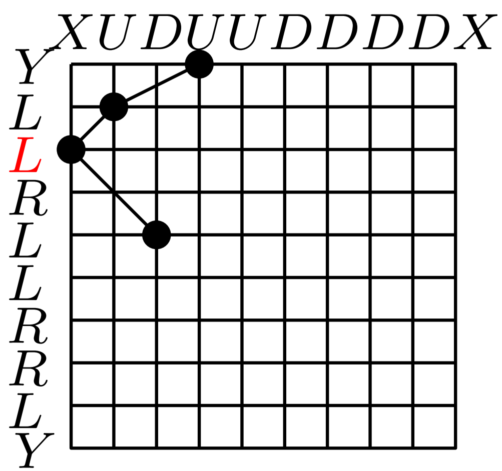
$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



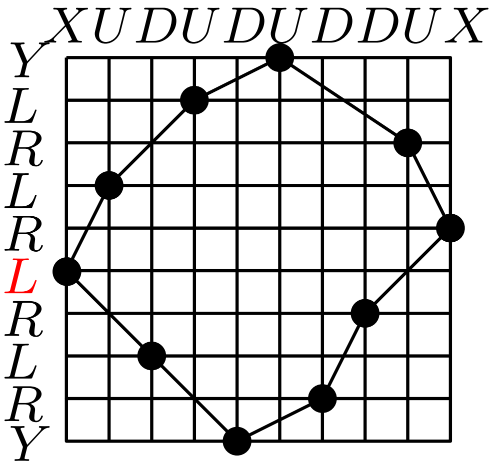


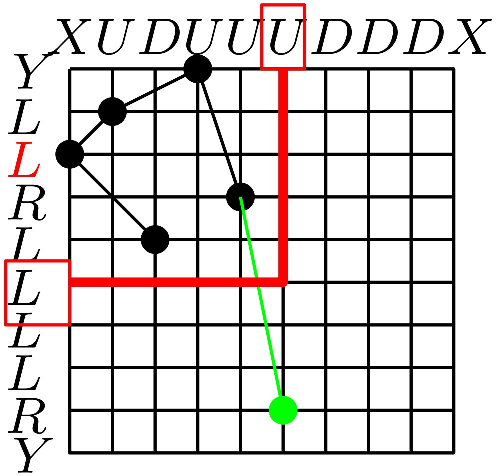


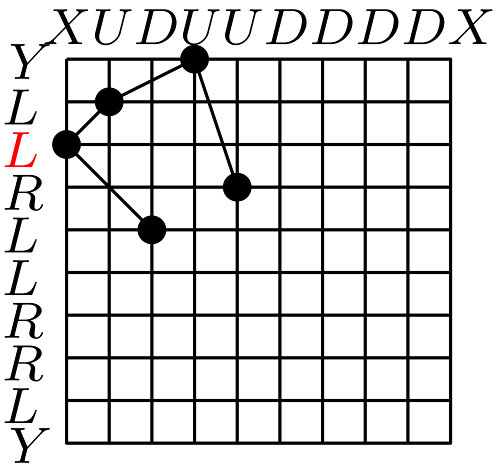
$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



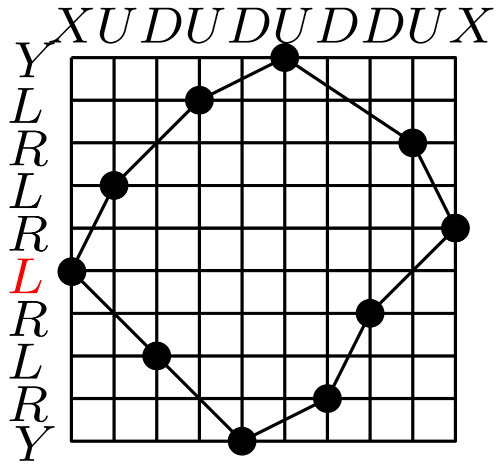


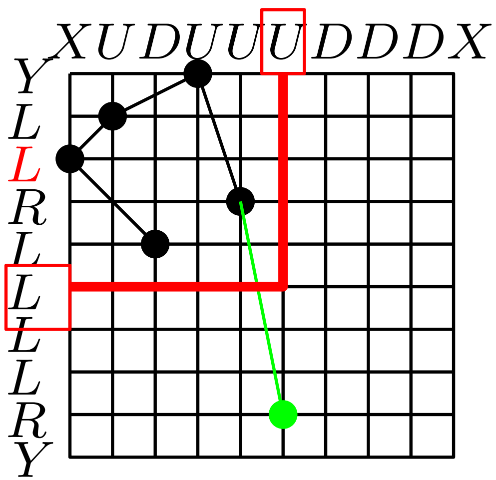


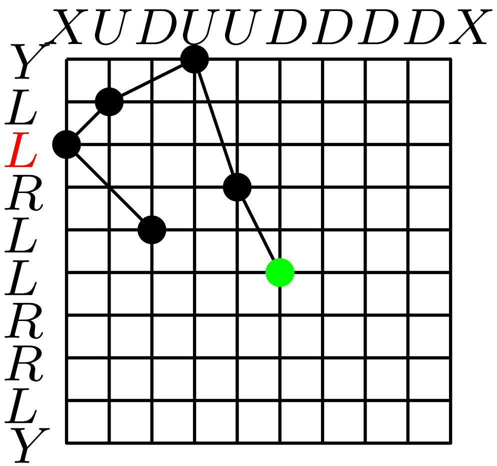
$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



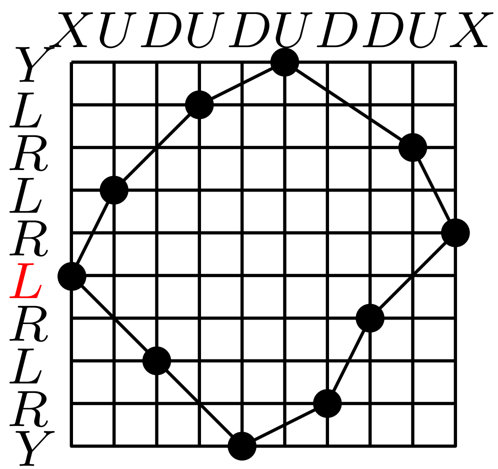


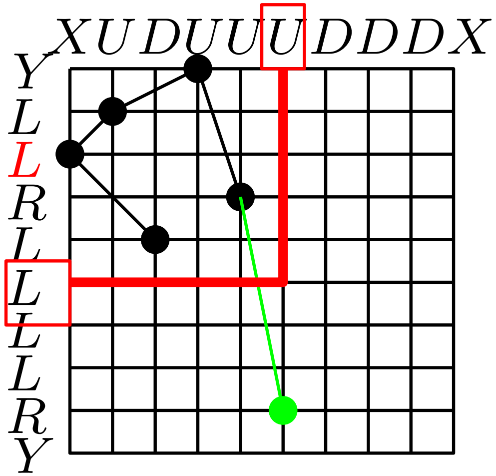


$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

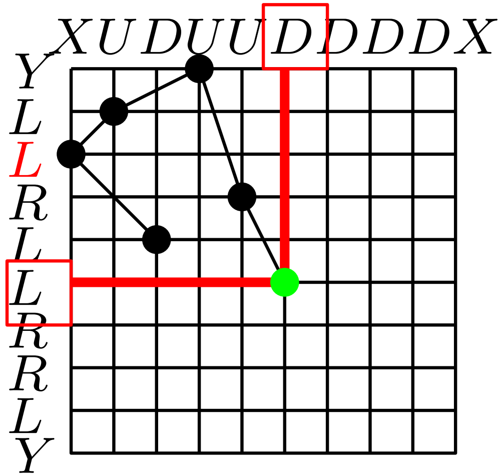
Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$





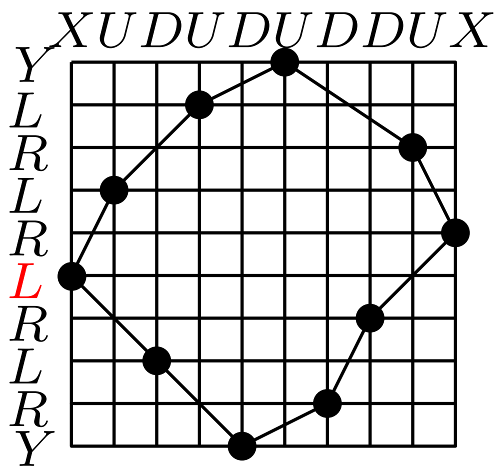
$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

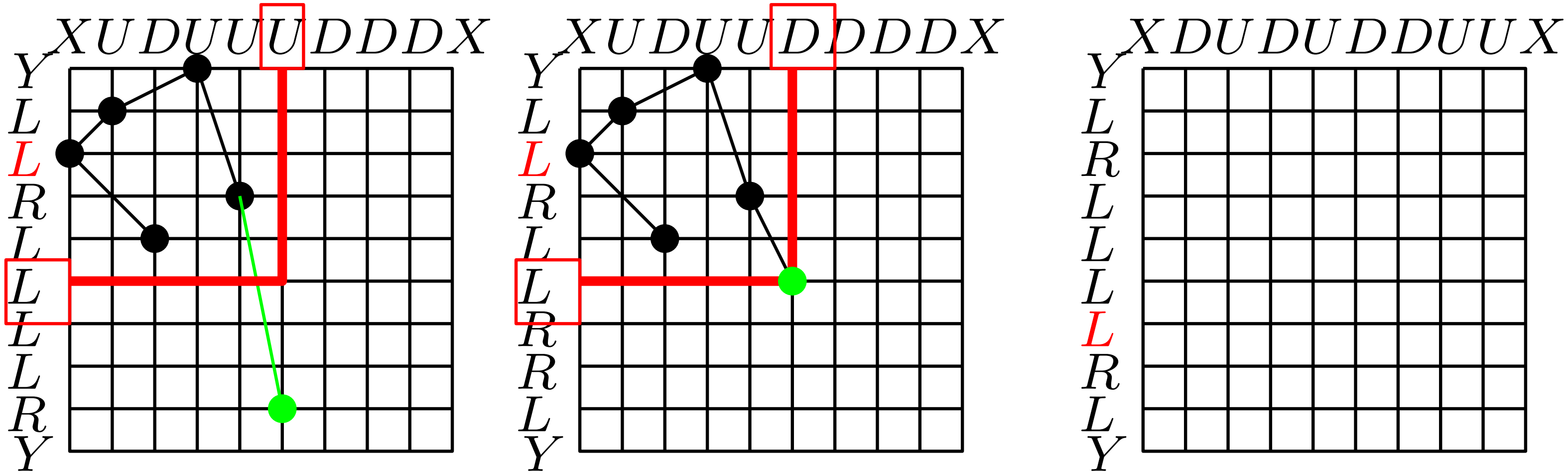


$\mathcal{T}^{\nwarrow} \cdot (D, L) \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



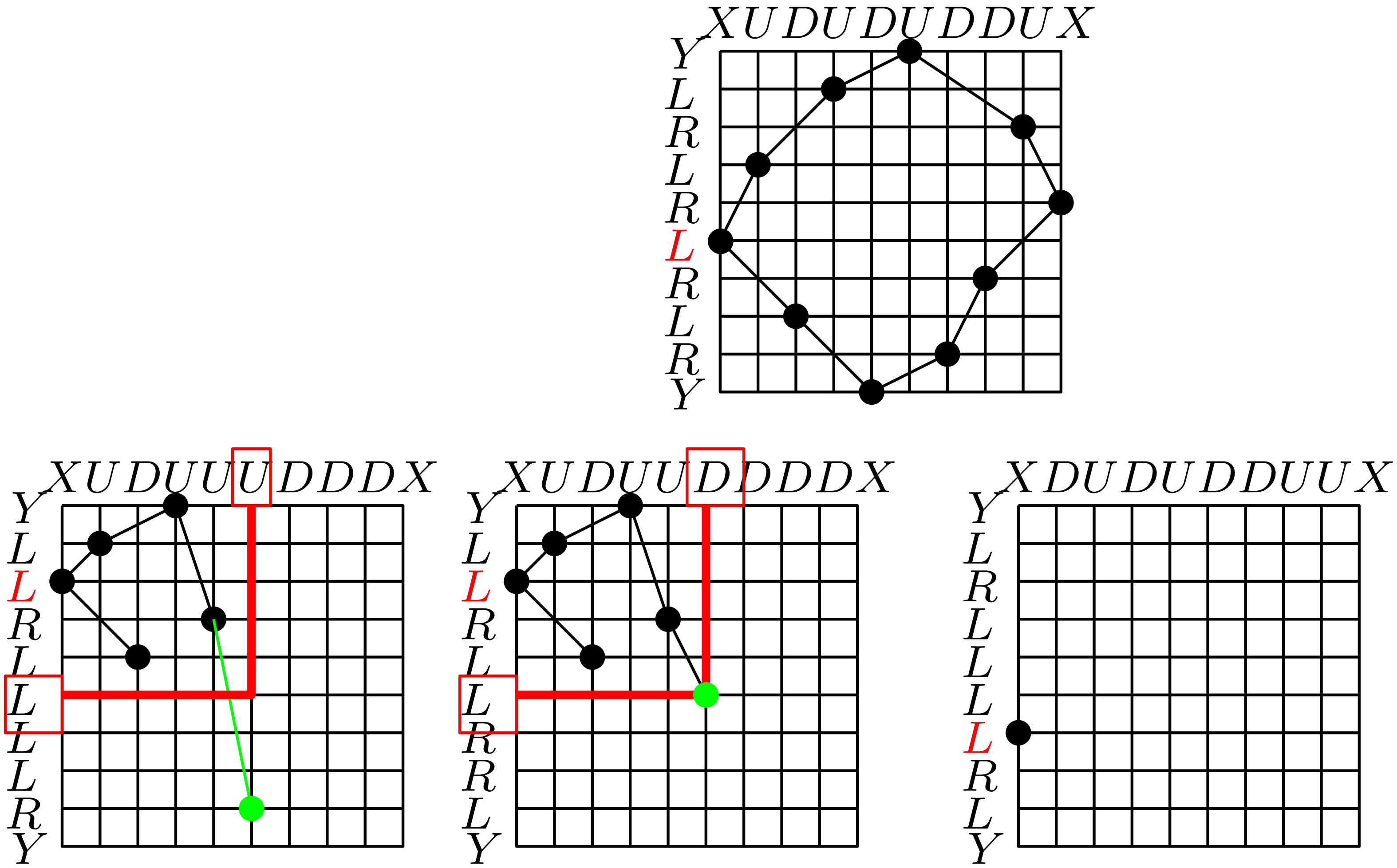


$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

$\mathcal{T}^{\nwarrow} \cdot (D, L) \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$

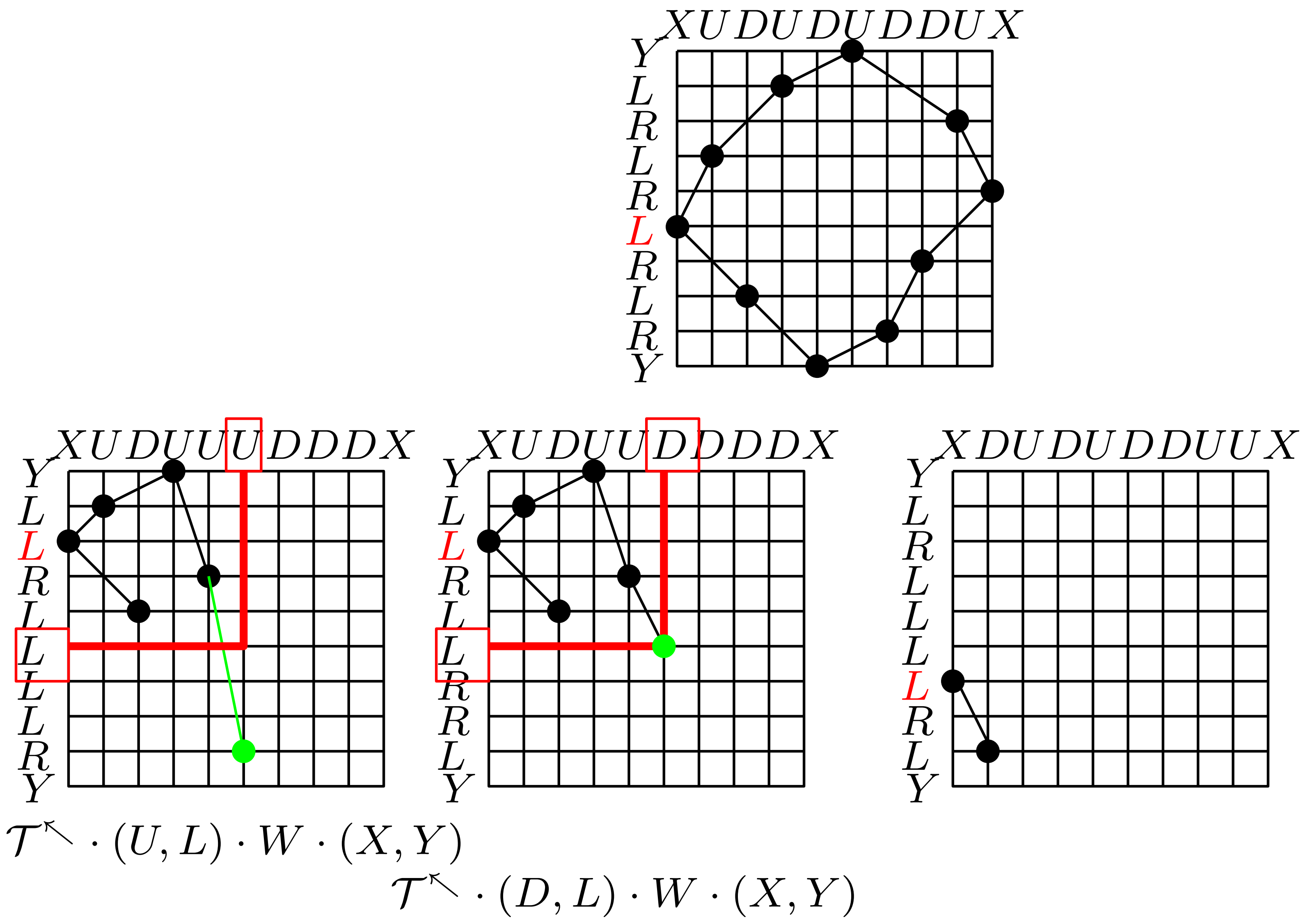


$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

$\mathcal{T}^{\nwarrow} \cdot (D, L) \cdot W \cdot (X, Y)$

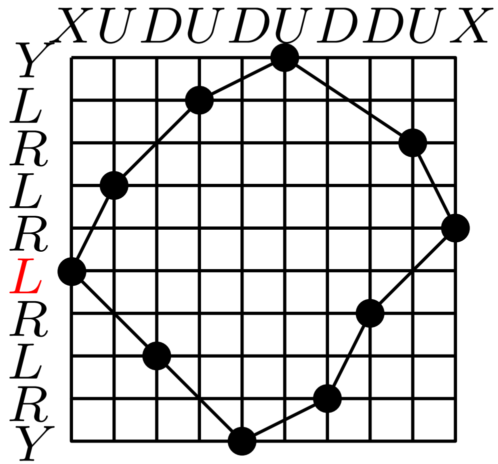
Decoding a word of $\mathcal{M}$

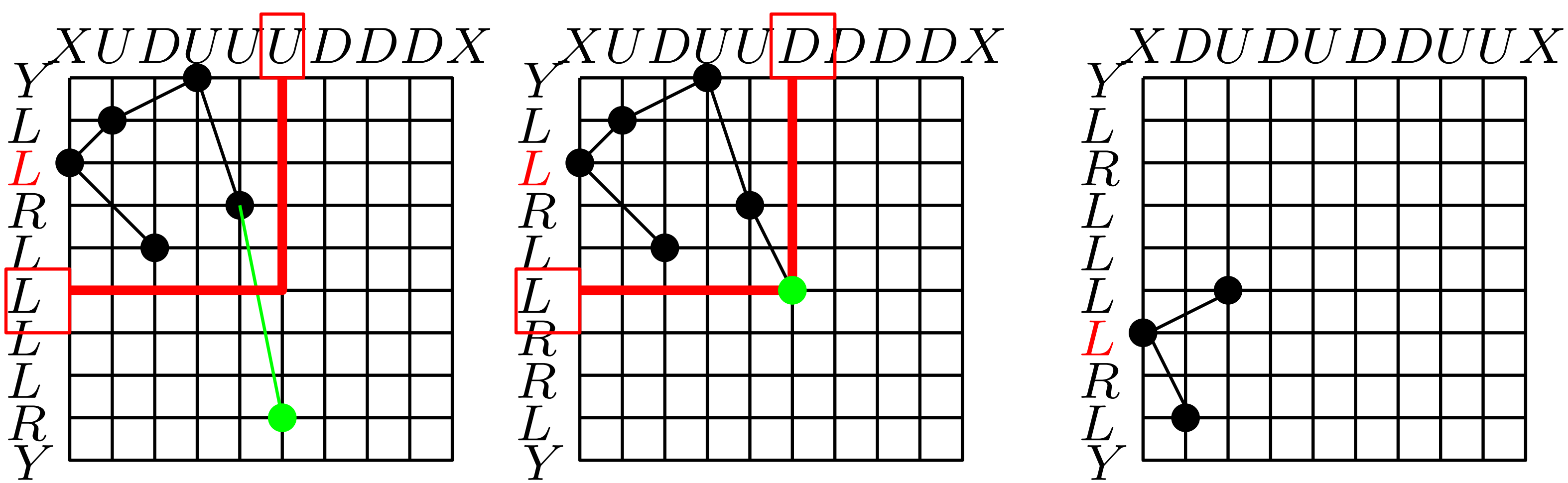
Consider a marked bi-word $(w, m) \in \mathcal{M}$



Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



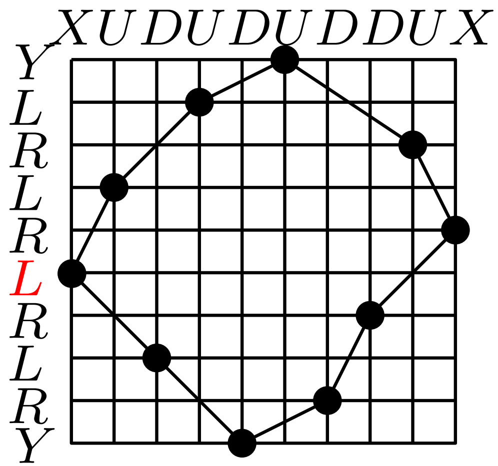


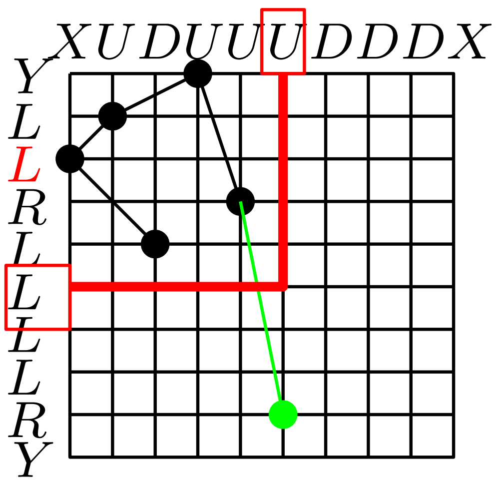
$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

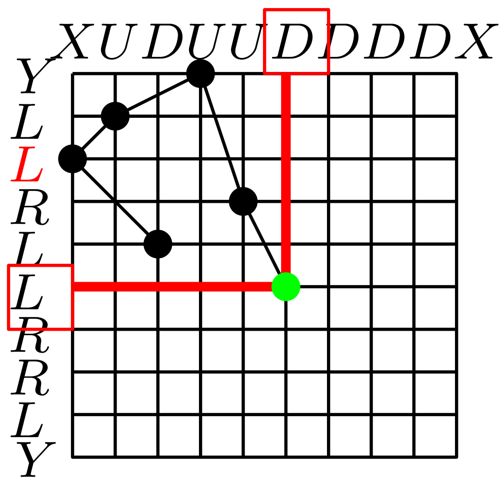
$\mathcal{T}^{\nwarrow} \cdot (D, L) \cdot W \cdot (X, Y)$

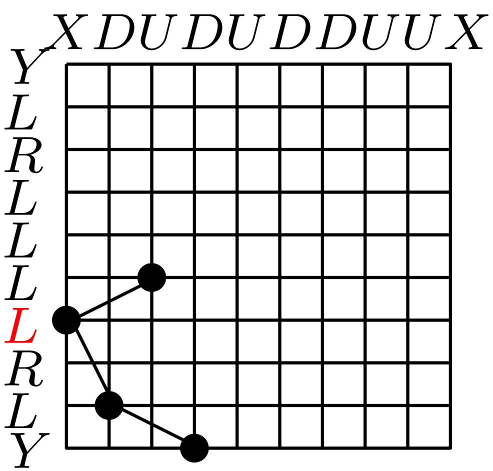
Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$







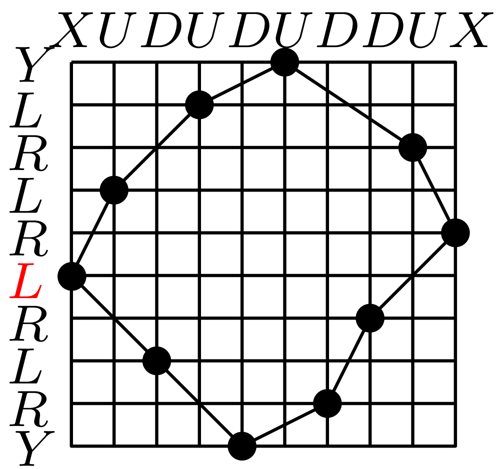


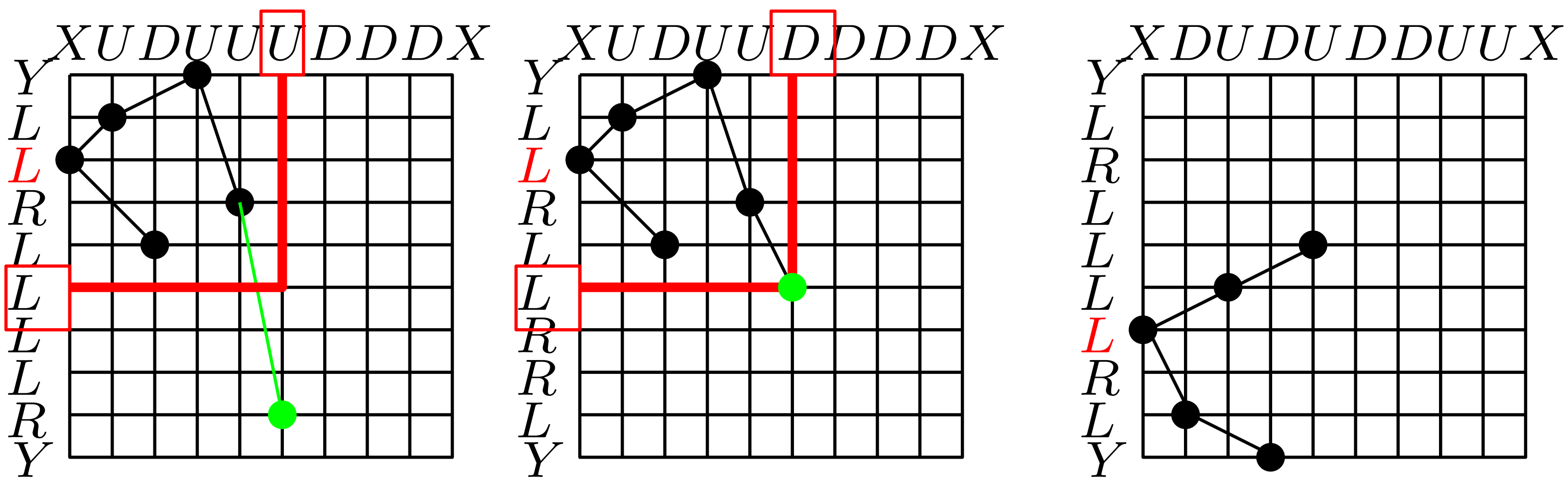
$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

$\mathcal{T}^{\nwarrow} \cdot (D, L) \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



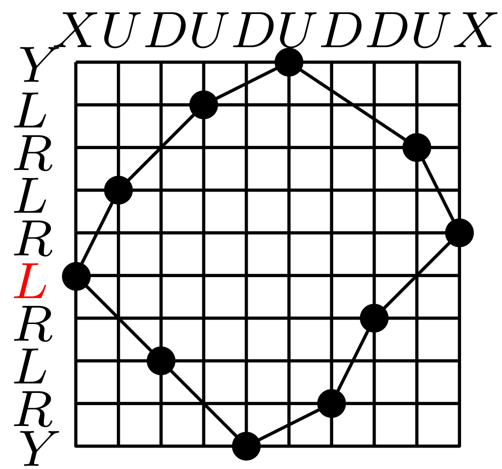


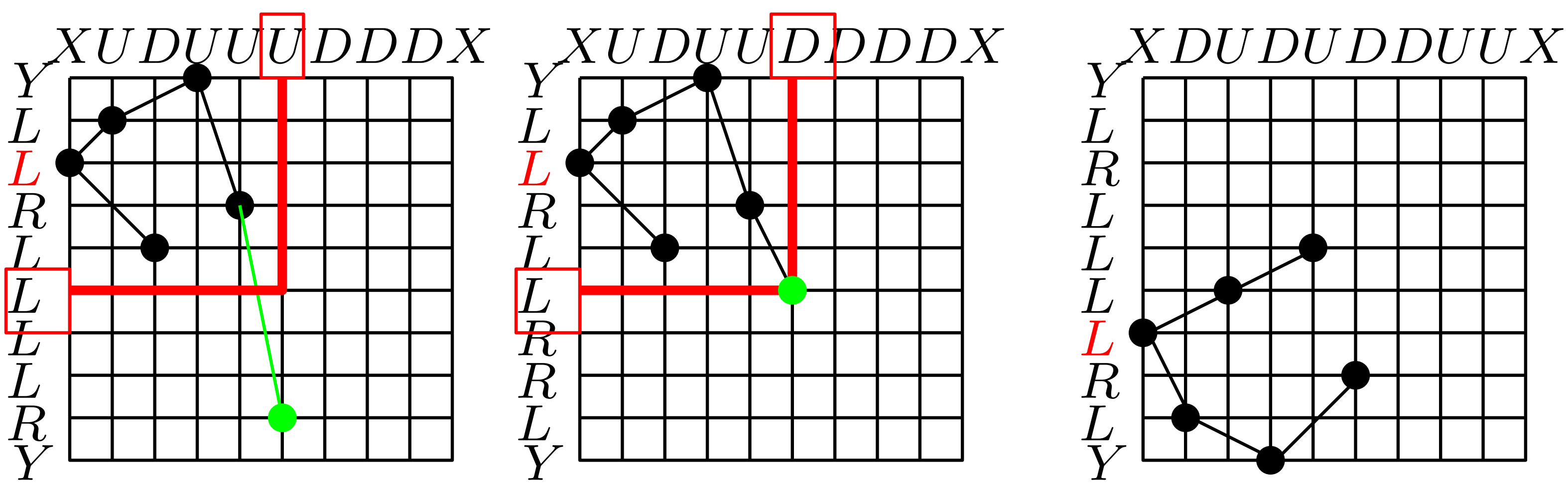
$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

$\mathcal{T}^{\nwarrow} \cdot (D, L) \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



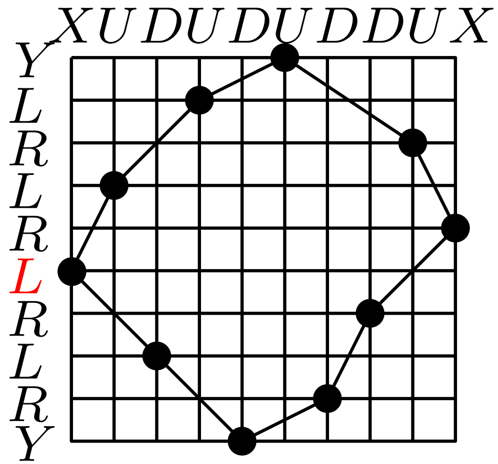


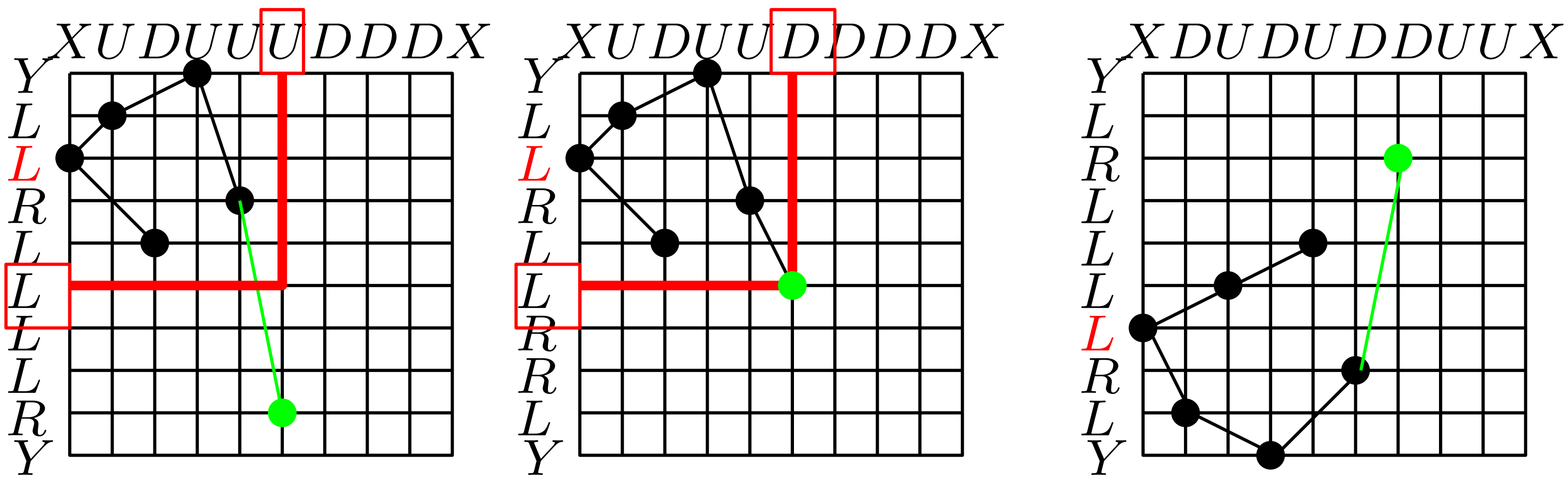
$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

$\mathcal{T}^{\nwarrow} \cdot (D, L) \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



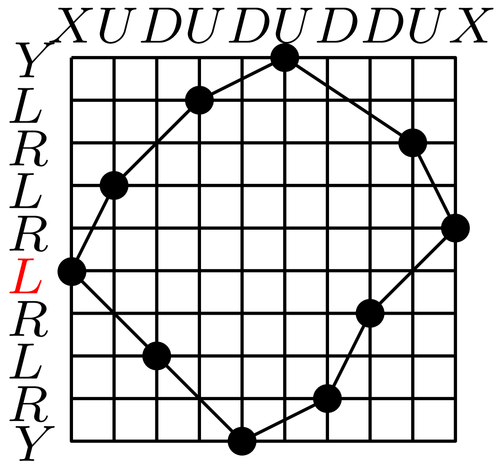


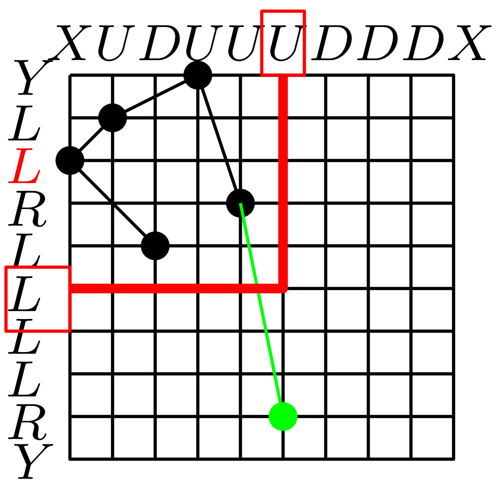
$$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$$

$$\mathcal{T}^{\nwarrow} \cdot (D, L) \cdot W \cdot (X, Y)$$

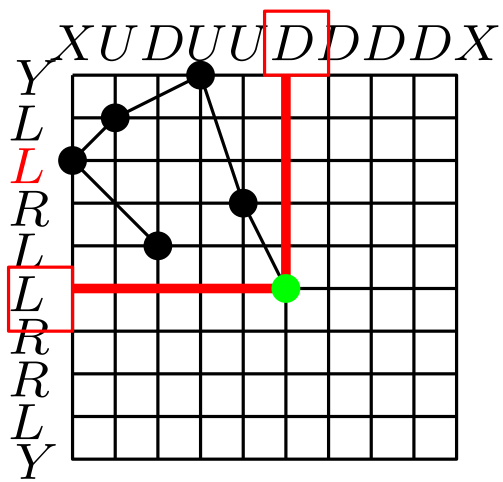
Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$

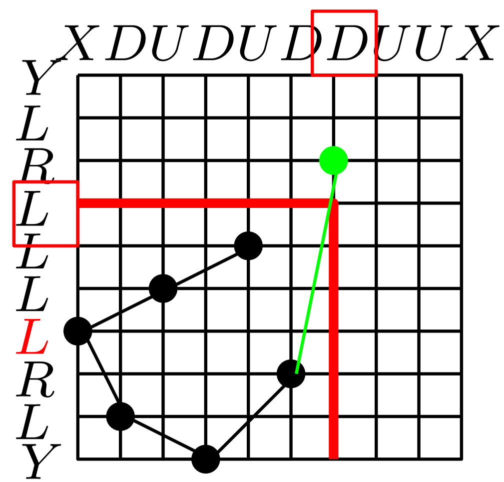




$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$



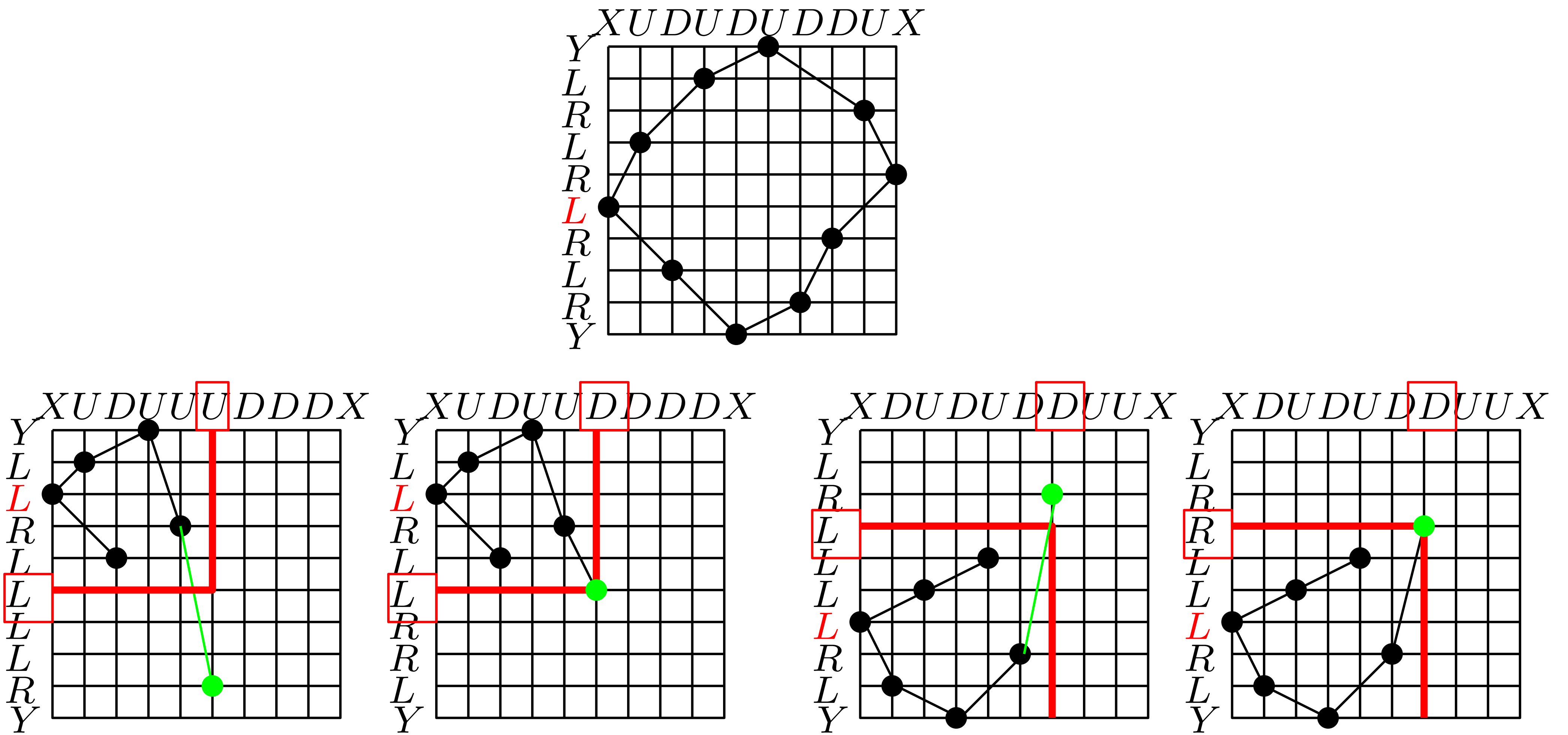
$\mathcal{T}^{\nwarrow} \cdot (D, L) \cdot W \cdot (X, Y)$



$\mathcal{T}^{\swarrow} \cdot (D, L), \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

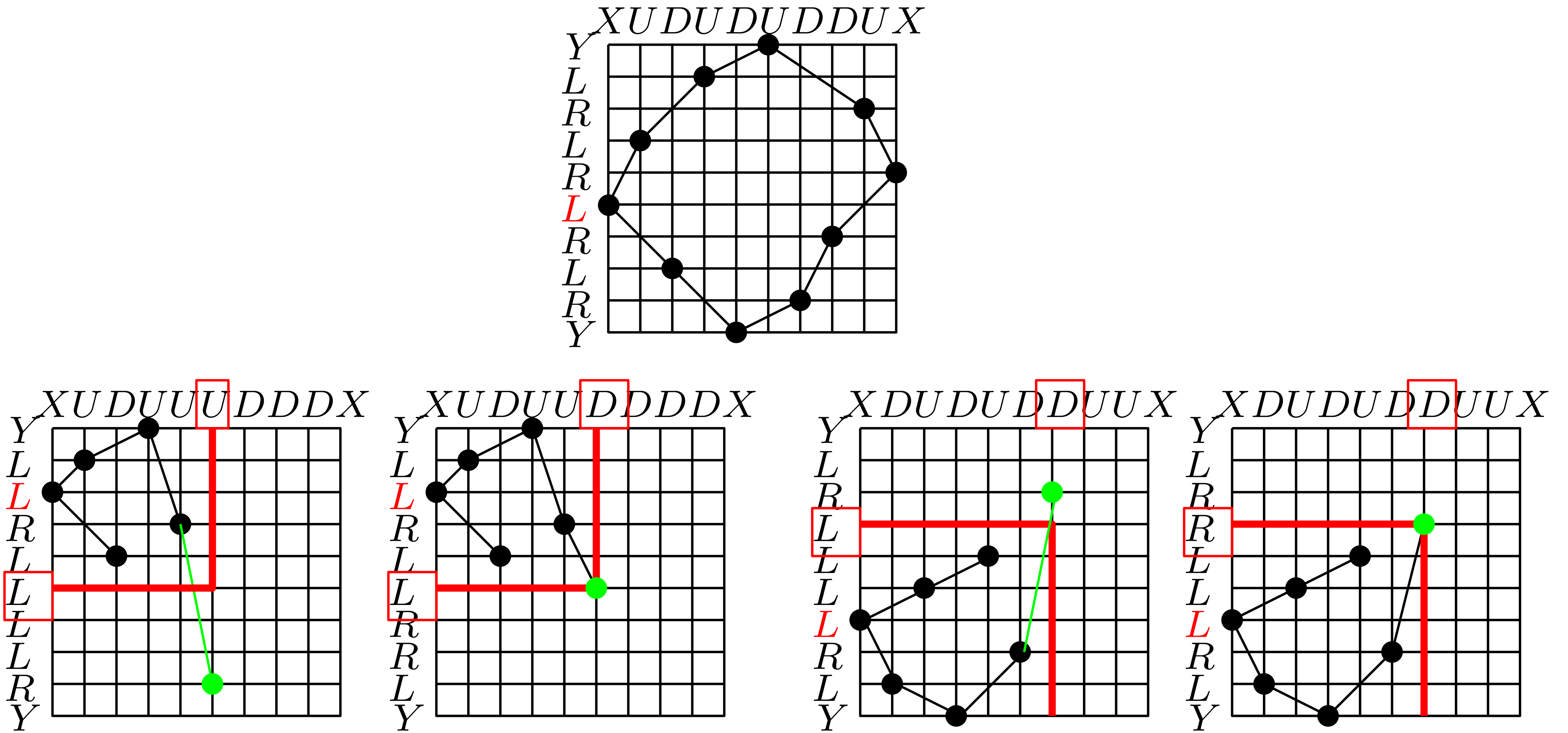
$\mathcal{T}^{\nwarrow} \cdot (D, L) \cdot W \cdot (X, Y)$

$\mathcal{T}^{\swarrow} \cdot (D, L), \cdot W \cdot (X, Y)$

$\mathcal{T}^{\swarrow} \cdot (D, R), \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

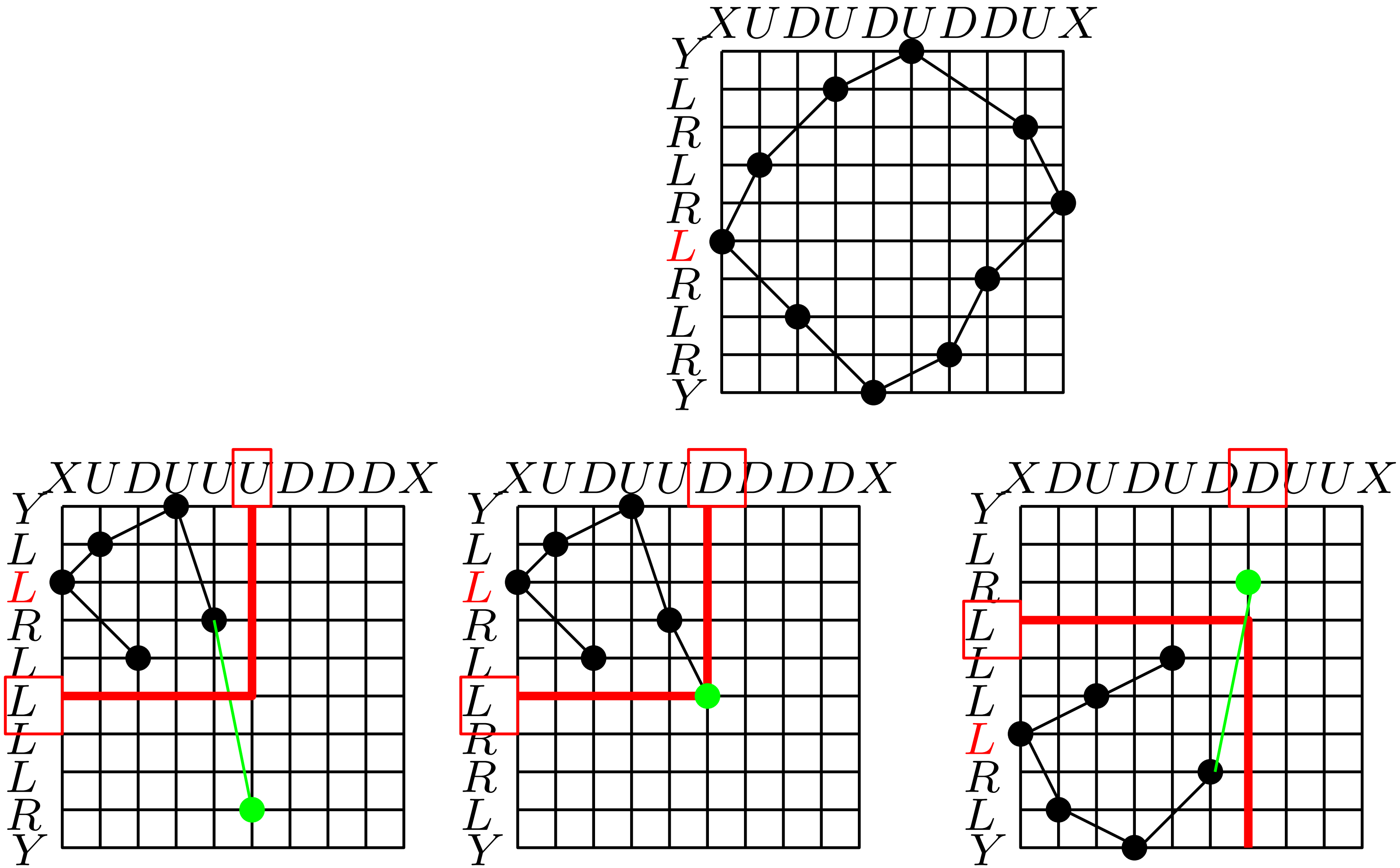
$\mathcal{T}^{\nwarrow} \cdot (D, L) \cdot W \cdot (X, Y)$

$\mathcal{T}^{\swarrow} \cdot (D, L), \cdot W \cdot (X, Y)$

$\mathcal{T}^{\swarrow} \cdot (D, R), \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$



$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

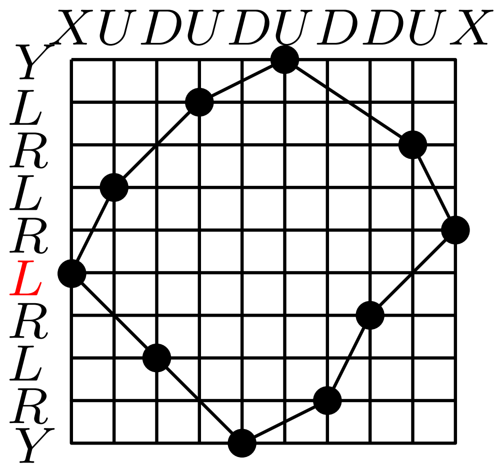
$\mathcal{T}^{\nwarrow} \cdot (D, L) \cdot W \cdot (X, Y)$

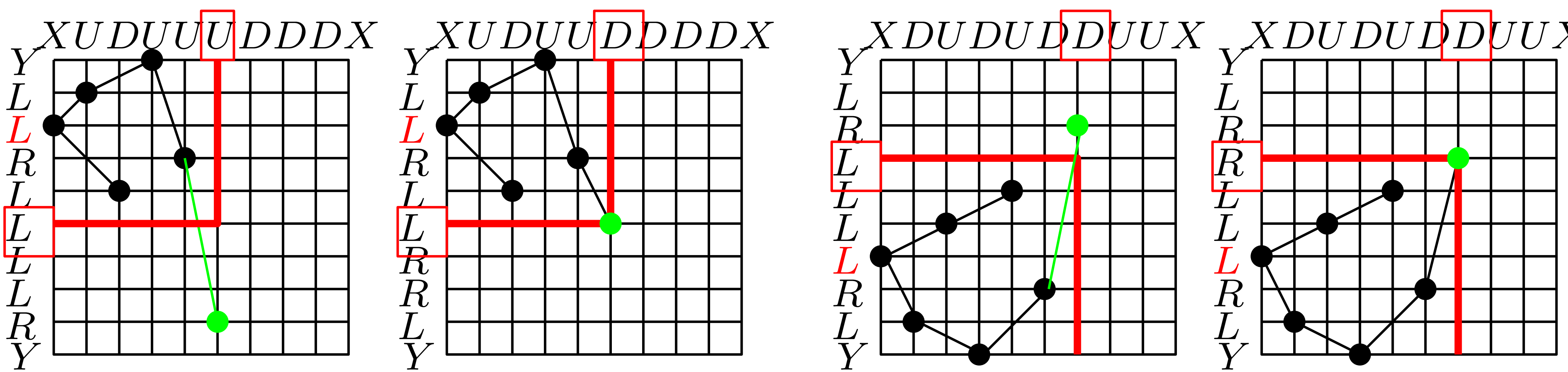
$\mathcal{T}^{\swarrow} \cdot (D, L), \cdot W \cdot (X, Y)$

$\mathcal{T}^{\swarrow} \cdot (D, R), \cdot W \cdot (X, Y)$

Decoding a word of $\mathcal{M}$

Consider a marked bi-word $(w, m) \in \mathcal{M}$





$\mathcal{T}^{\nwarrow} \cdot (U, L) \cdot W \cdot (X, Y)$

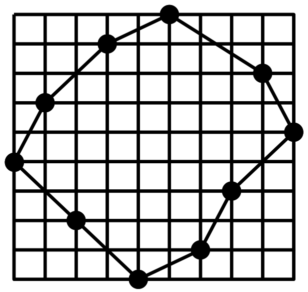
$\mathcal{T}^{\nwarrow} \cdot (D, L) \cdot W \cdot (X, Y)$

$\mathcal{T}^{\swarrow} \cdot (D, L), \cdot W \cdot (X, Y)$

$\mathcal{T}^{\swarrow} \cdot (D, R), \cdot W \cdot (X, Y)$

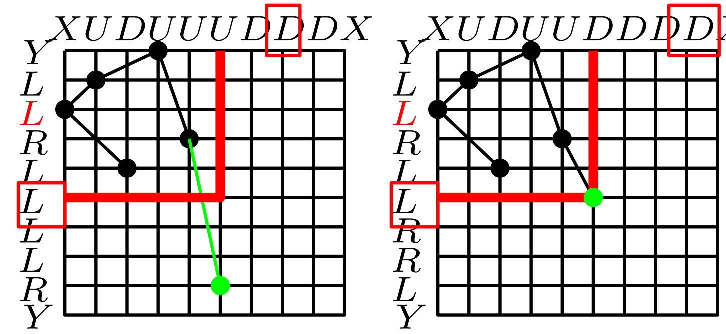
$$\mathcal{S} = \mathcal{M} - \mathcal{T}^{\nwarrow} \cdot \{(U, L), (D, L)\} \cdot W \cdot (X, Y) - \mathcal{T}^{\swarrow} \cdot \{(D, L), (D, R)\} \cdot W \cdot (X, Y)$$

Interpretation of the difference

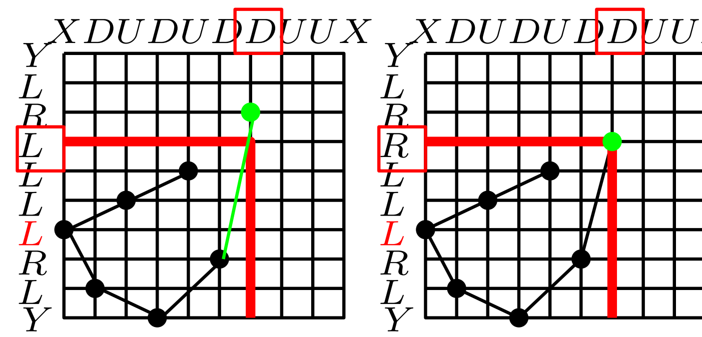


$$= \begin{matrix} XUDUDUDDUX \\ YRLR{\color{red}L}RLRLY \end{matrix}$$

$-$



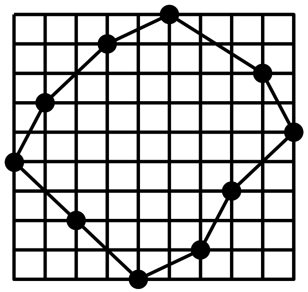
$-$



$$\mathcal{S} \quad = \quad \mathcal{M} \quad - \quad \mathcal{T}^{\nwarrow} \cdot \{(U,L),(D,L)\} \cdot W \cdot (X,Y)$$

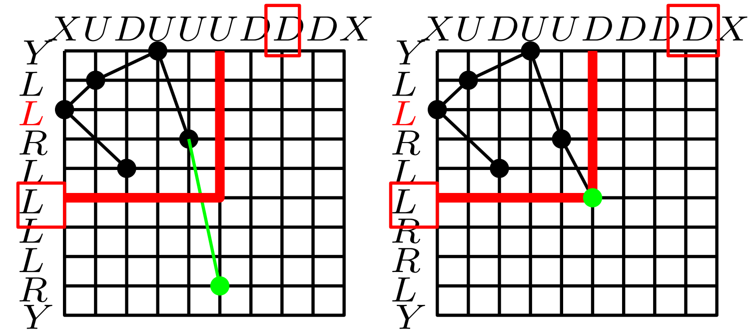
$$- \quad \mathcal{T}^{\swarrow} \cdot \{(D,L),(D,R)\} \cdot W \cdot (X,Y)$$

Interpretation of the difference

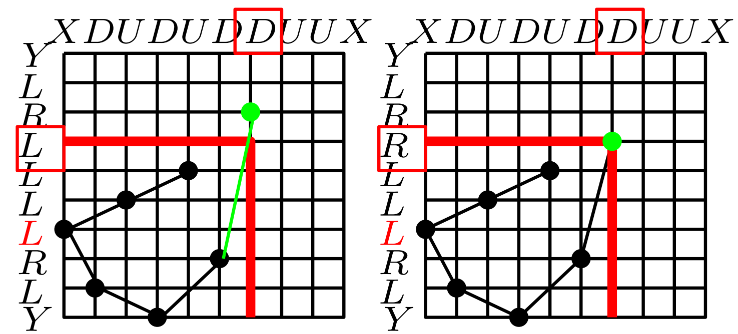


$$= \begin{array}{c} XUDUDUDDUX \\ YRLR\color{red}{L}RLRLY \end{array}$$

$-$

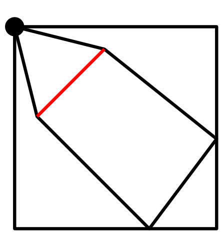


$-$



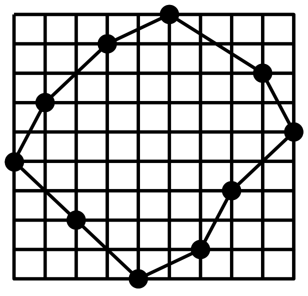
$$\mathcal{S} \quad = \quad \mathcal{M} \quad - \quad \mathcal{T}^{\nwarrow} \cdot \{(U,L),(D,L)\} \cdot W \cdot (X,Y)$$

$$- \quad \mathcal{T}^{\swarrow} \cdot \{(D,L),(D,R)\} \cdot W \cdot (X,Y)$$

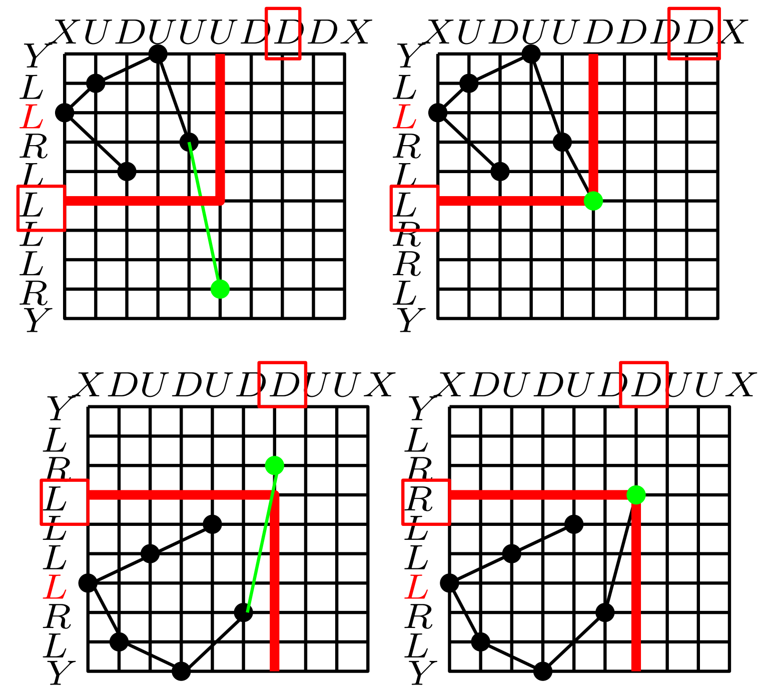


$$\equiv \quad \bullet\mathcal{T}^{\searrow} = \{(w, \color{red}{n})\} - \bullet\mathcal{P} \cdot \{(U,L),(D,L)\} \cdot W \cdot (X,Y)$$

Interpretation of the difference

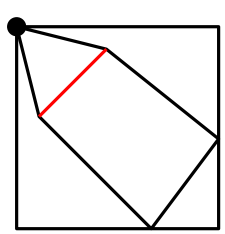


$=$ $XUDUDUDDUX$ / $YRLRLRLRLY$ (the fifth L in red) $-$ / $-$



$$\mathcal{S} \quad = \quad \mathcal{M} \quad - \quad \mathcal{T}^{\nwarrow} \cdot \{(U,L),(D,L)\} \cdot W \cdot (X,Y)$$

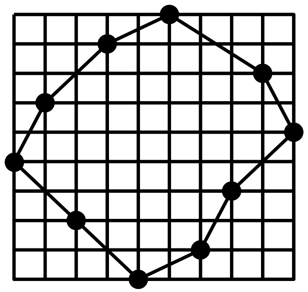
$$- \quad \mathcal{T}^{\swarrow} \cdot \{(D,L),(D,R)\} \cdot W \cdot (X,Y)$$



$$\equiv \quad \bullet\mathcal{T}^{\searrow} = \{(w,n)\} - \bullet\mathcal{P} \cdot \{(U,L),(D,L)\} \cdot W \cdot (X,Y)$$

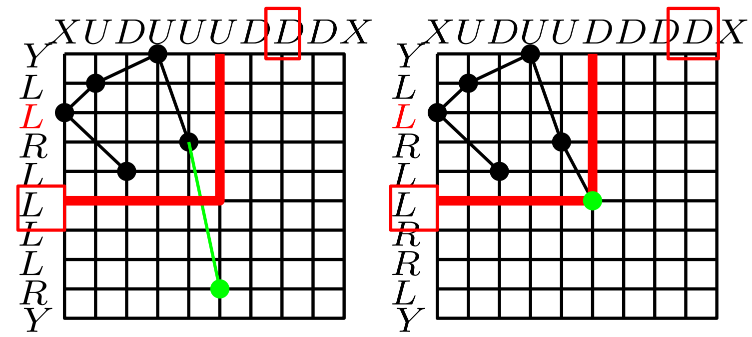
$\mathcal{P}$ is Catalan via previous bijection.

Interpretation of the difference

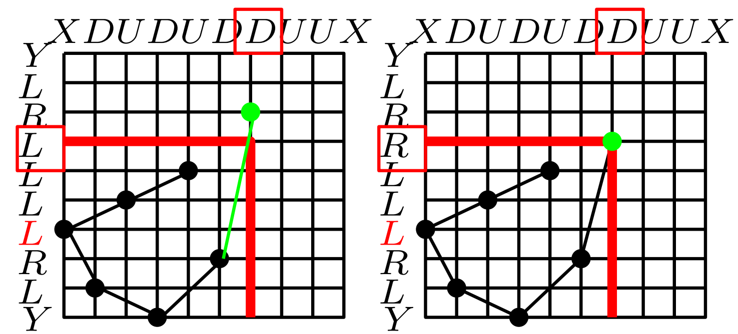


$=$ $\begin{array}{c} XUDUDUDDUX \\ YRLRLRLRLY \end{array}$

$-$

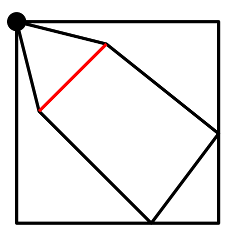


$-$



$$\mathcal{S} \quad = \quad \mathcal{M} \quad - \quad \mathcal{T}^{\nwarrow} \cdot \{(U,L),(D,L)\} \cdot W \cdot (X,Y)$$

$$- \quad \mathcal{T}^{\swarrow} \cdot \{(D,L),(D,R)\} \cdot W \cdot (X,Y)$$



$$\equiv \quad \bullet\mathcal{T}^{\searrow} = \{(w,n)\} - \bullet\mathcal{P} \cdot \{(U,L),(D,L)\} \cdot W \cdot (X,Y)$$

$\mathcal{P}$ is Catalan via previous bijection.

rational gf

algebraic gf

Translating into equation for generating functions

Recall that $\mathcal{W}(t) = \sum_{\overline{w}\in\mathcal{W}} t^{n(\overline{w})} = \frac{1}{1-4t}$ and $\mathcal{M}(t) = \frac{2t^2}{1-4t} + \frac{2t^3}{(1-4t)^2}$

while $\mathcal{P}^{\nwarrow}(t) = \mathcal{P}^{\nearrow}(t) = \frac{1-2t-\sqrt{1-4t}}{2t}$ and $\mathcal{T}^{\nwarrow}(t) = \mathcal{T}^{\nearrow}(t) = \mathcal{T}^{\searrow}(t)$

so that the combinatorial interpretation

$$\begin{aligned} \mathcal{S} \quad = \quad \mathcal{M} \quad & - \quad \mathcal{T}^{\nwarrow} \cdot \{(U,L),(D,L)\} \cdot W \cdot (X,Y) \\ & - \quad \mathcal{T}^{\swarrow} \cdot \{(D,L),(D,R)\} \cdot W \cdot (X,Y) \end{aligned}$$

$$\bullet\mathcal{T}^{\searrow} \; = \; \{(w,n)\} - \quad \bullet\mathcal{P} \cdot \{(U,L),(D,L)\} \cdot W \cdot (X,Y)$$

$\mathcal{P}$ is Catalan via previous bijection.

translates into the following expressions for generating functions:

$$t\mathcal{T}^{\searrow}(t) \quad = \quad t^2\mathcal{W}(t) \quad - \quad 2t^3\mathcal{P}^{\nwarrow}(t)\mathcal{W}(t)$$

and

$$\begin{aligned} \mathcal{S}(t) \quad & = \quad \mathcal{M}(t) \quad - \quad 4t^2\mathcal{T}^{\searrow}(t)\mathcal{W}(t) \\ & = \frac{2t^2}{1-4t} + \frac{2t}{(1-4t)^2} \quad - \quad \frac{4t^3}{(1-4t)^{3/2}} \end{aligned}$$

Combinatorial interpretations

2) Interpretation of formulas with differences

Extension to convex permupolygons

An extended code for convex permutominoes

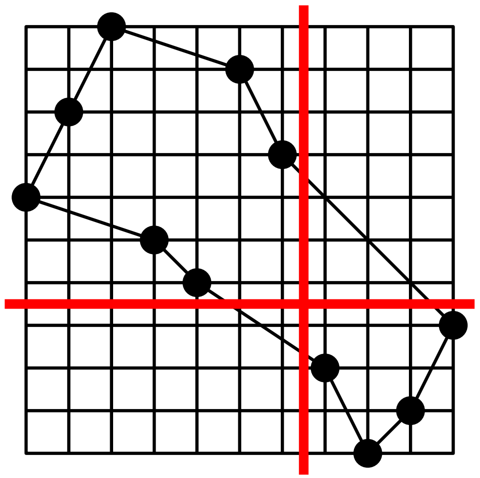
We use a bijection between co-undecomposable square permutations with colored fix points and permutominoes.

We extend the previous encoding to undecomposable square permutations with colored fix points.

An extended code for convex permutominoes

We use a bijection between co-undecomposable square permutations with colored fix points and permutominoes.

We extend the previous encoding to undecomposable square permutations with colored fix points.

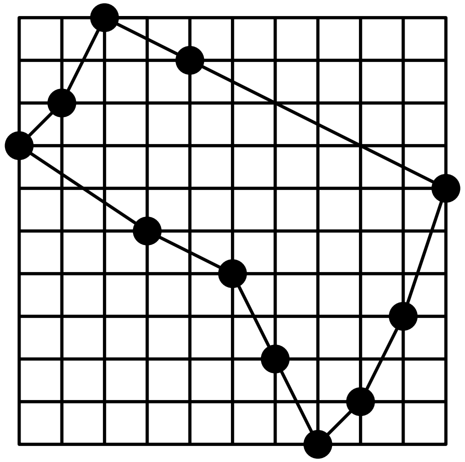


A co-decomposable permutation σ:
there exists two permutations π of $\{1, \ldots, k\}$ and π' of $\{1, \ldots, \ell\}$ such that $\sigma = \pi_1 + \ell, \pi_2 + \ell, \ldots, \pi_k + \ell, \pi'_1, \pi'_2, \ldots, \pi'_\ell$.

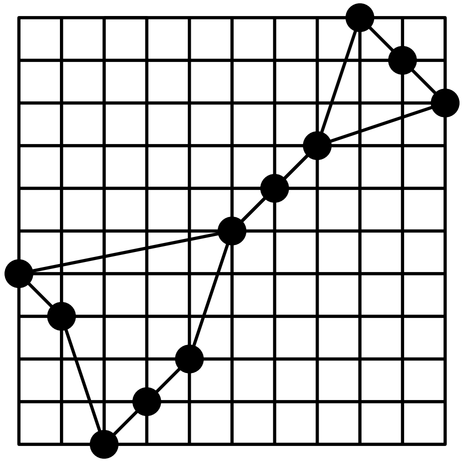
An extended code for convex permutominoes

We use a bijection between co-undecomposable square permutations with colored fix points and permutominoes.

We extend the previous encoding to undecomposable square permutations with colored fix points.



A co-indecomposable square permutation σ without fix points $\sigma(i) = i$.

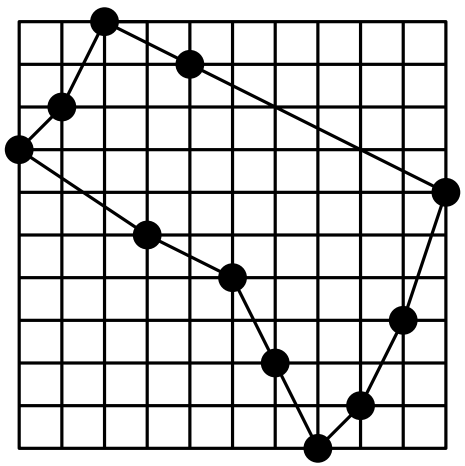


A co-indecomposable square permutation σ with three fixed points.

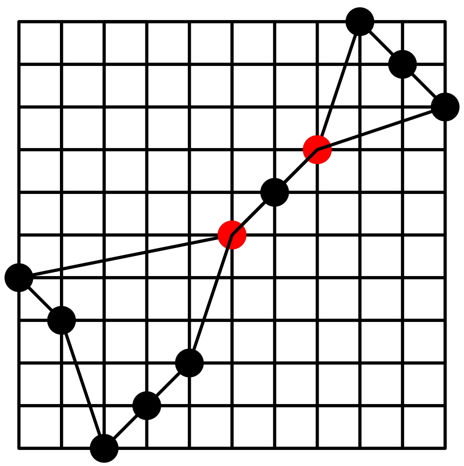
An extended code for convex permutominoes

We use a bijection between co-undecomposable square permutations with colored fix points and permutominoes.

We extend the previous encoding to undecomposable square permutations with colored fix points.



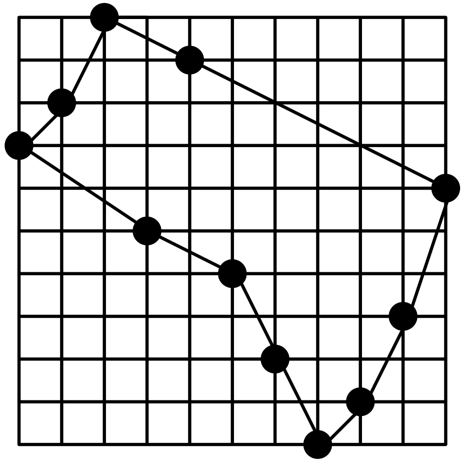
A co-indecomposable square permutation σ without fix points $\sigma(i) = i$.

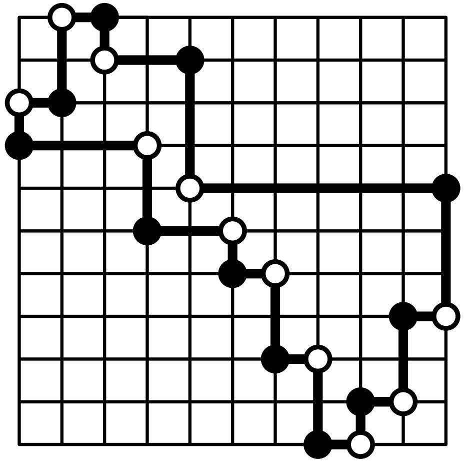


A co-indecomposable colored square permutation σ with three fixed points.

An extended code for convex permutominoes

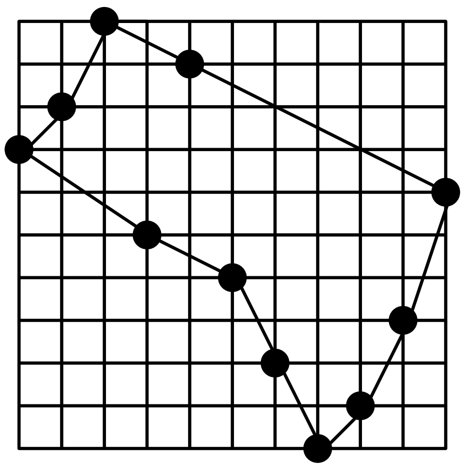
A bijection between co-indecomposable colored square permutations and convex permutominoes. (Bernini *et al.*, 2007)

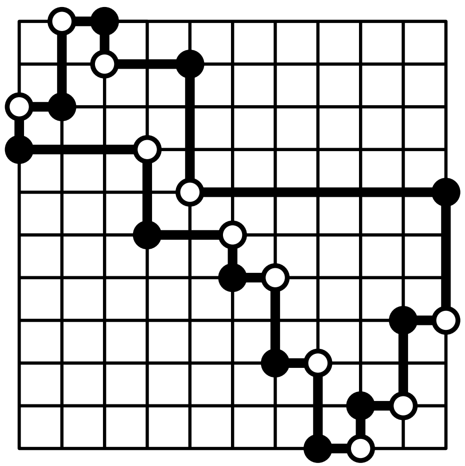




An extended code for convex permutominoes

A bijection between co-indecomposable colored square permutations and convex permutominoes. (Bernini *et al.*, 2007)

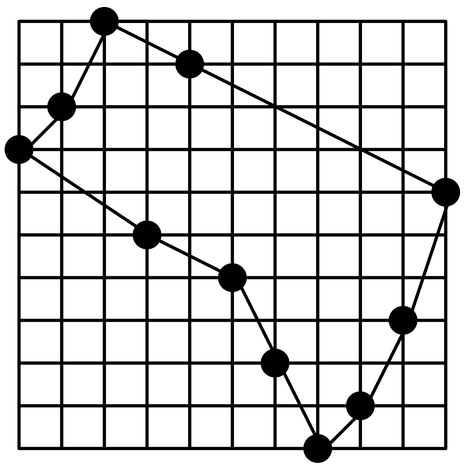


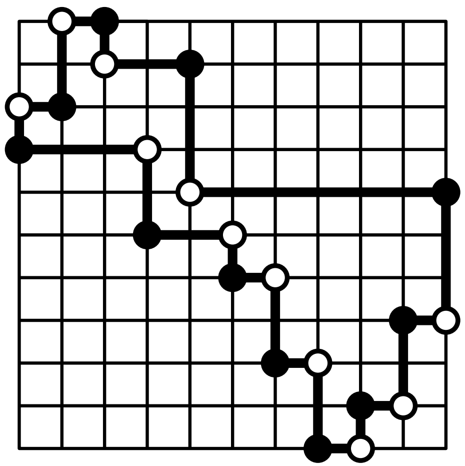


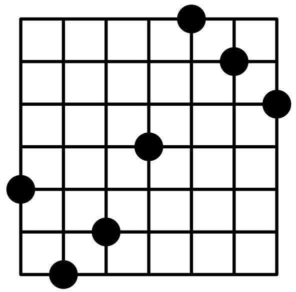
The square permutation corresponds to the black permutation in the associated convex permutomino.

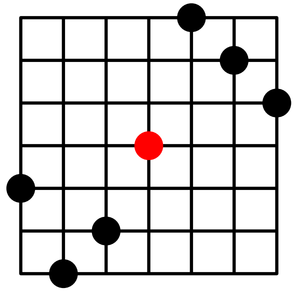
An extended code for convex permutominoes

A bijection between co-indecomposable colored square permutations and convex permutominoes. (Bernini *et al.*, 2007)







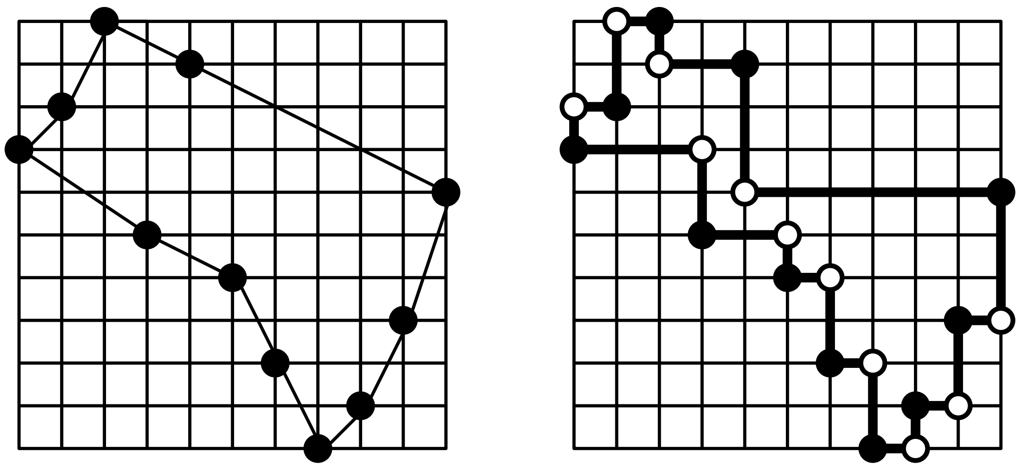


The square permutation corresponds to the black permutation in the associated convex permutomino.

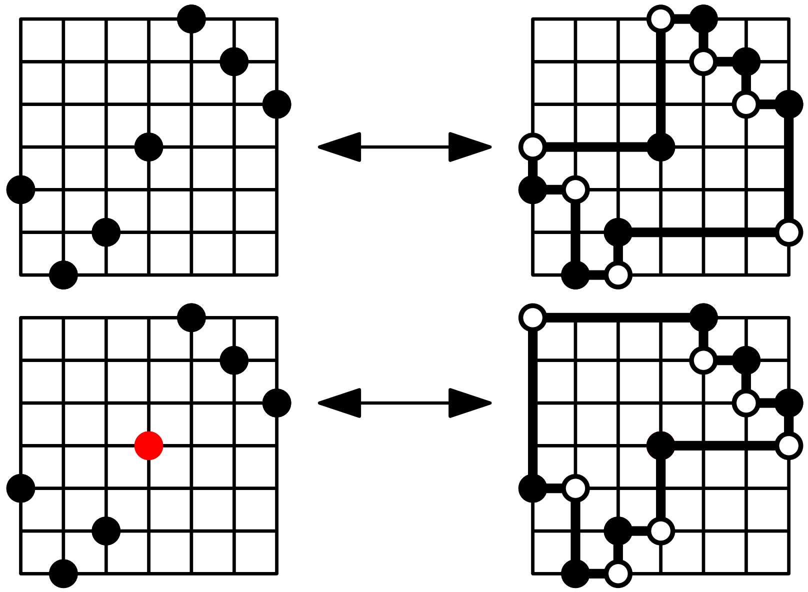
What happens if we have fix points?

An extended code for convex permutominoes

A bijection between co-indecomposable colored square permutations and convex permutominoes. (Bernini *et al.*, 2007)



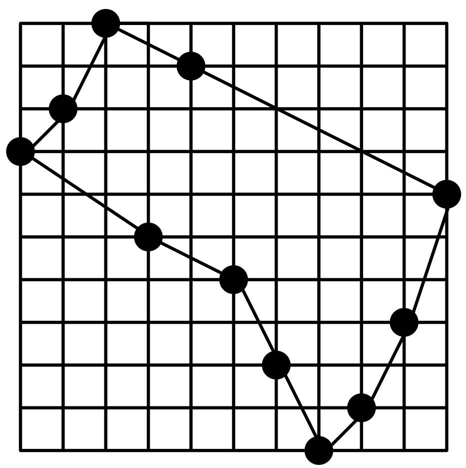
The square permutation corresponds to the black permutation in the associated convex permutomino.

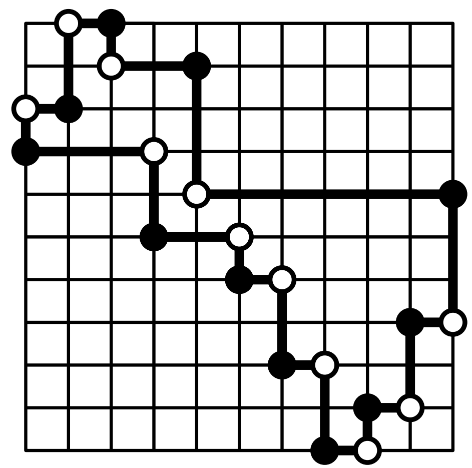


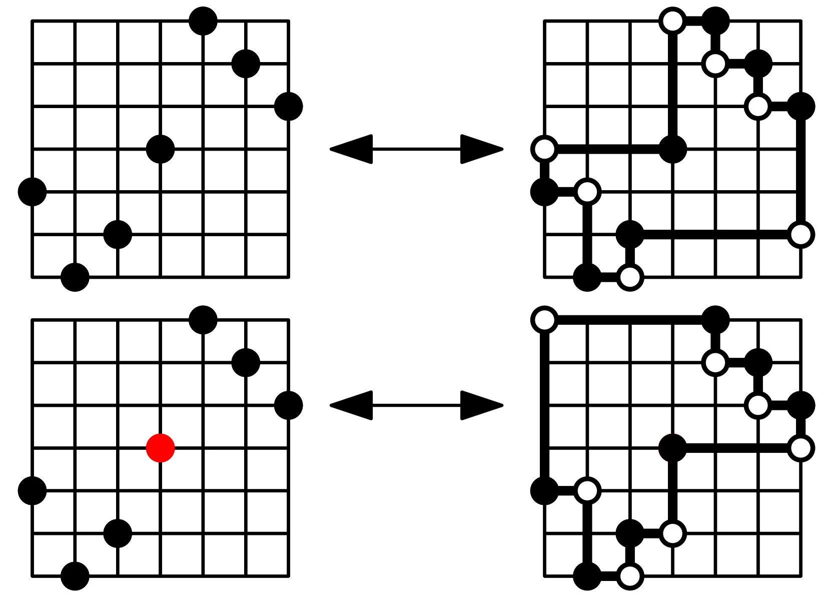
What happens if we have fix points?
It depends if they are colored or not.

An extended code for convex permutominoes

A bijection between co-indecomposable colored square permutations and convex permutominoes. (Bernini *et al.*, 2007)



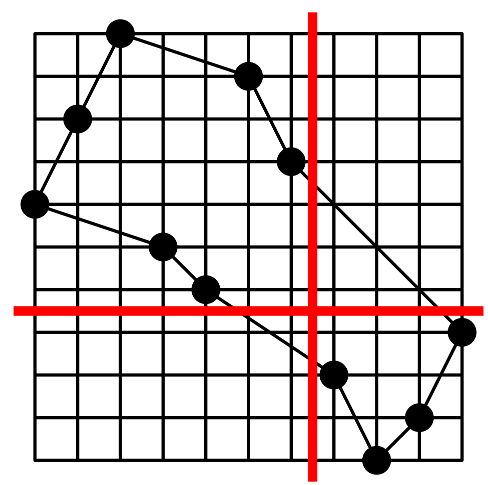




The square permutation corresponds to the black permutation in the associated convex permutomino.

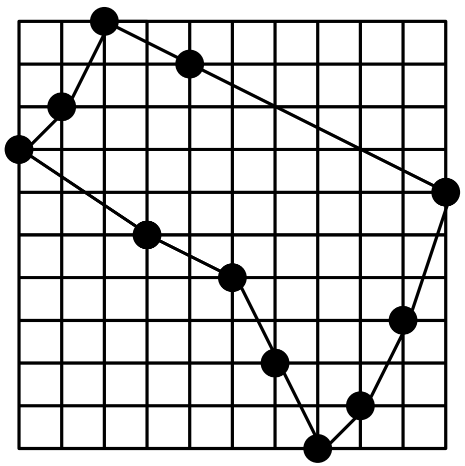
What happens if we have fix points?
It depends if they are colored or not.

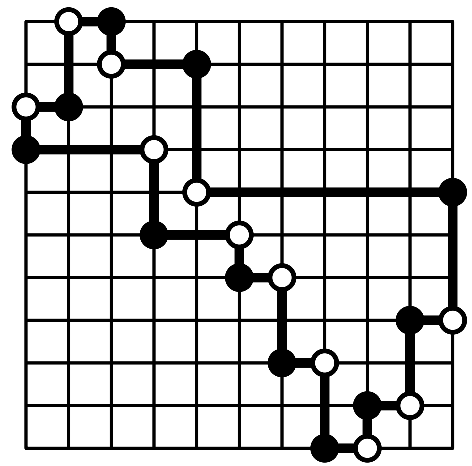
Why are co-decomposable permutation forbidden?

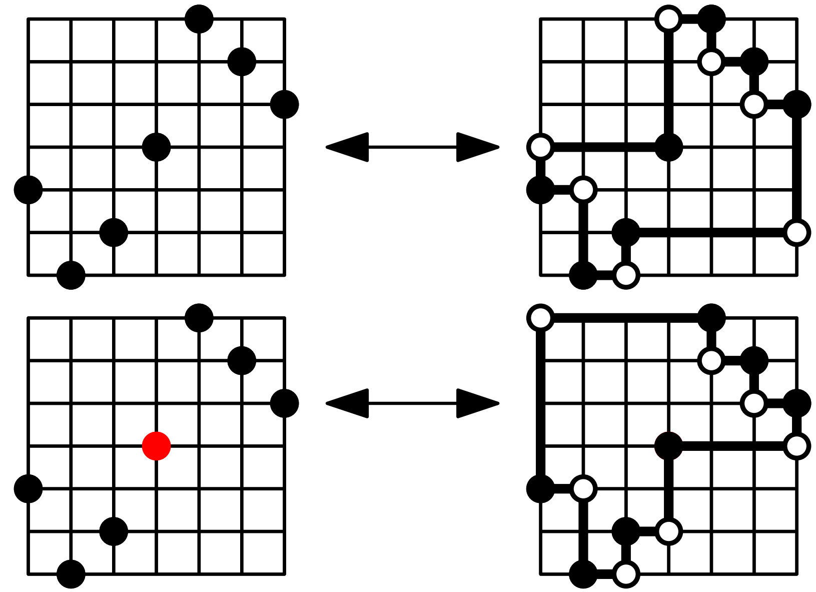


An extended code for convex permutominoes

A bijection between co-indecomposable colored square permutations and convex permutominoes. (Bernini *et al.*, 2007)





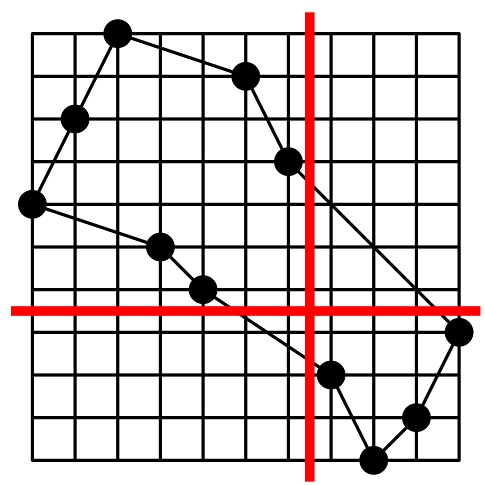


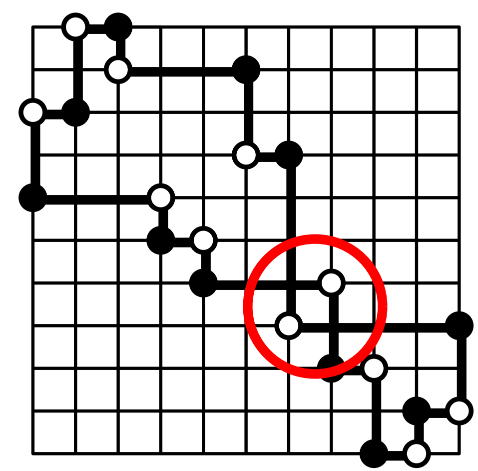
The square permutation corresponds to the black permutation in the associated convex permutomino.

What happens if we have fix points? It depends if they are colored or not.

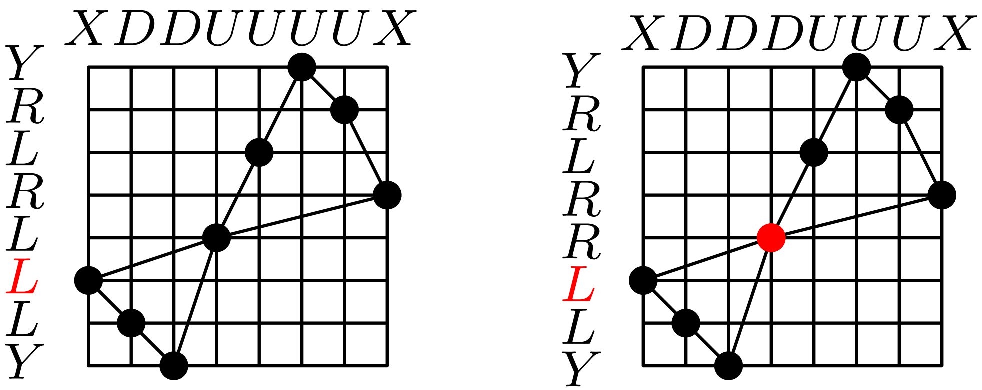
Why are co-decomposable permutation forbidden?

They lead to self intersecting permutominoes.





An extended code for convex permutominoes



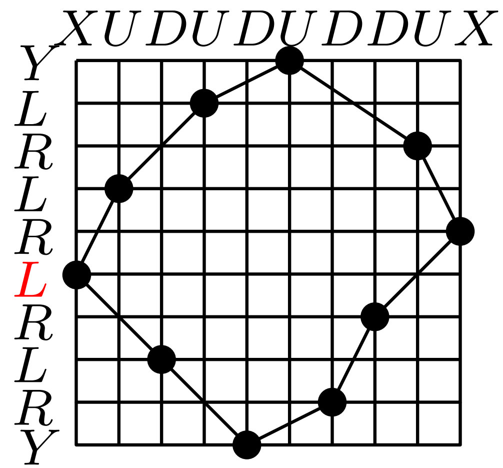
$u_i = X$ for extremal points in the vertical borders of the bounding box.

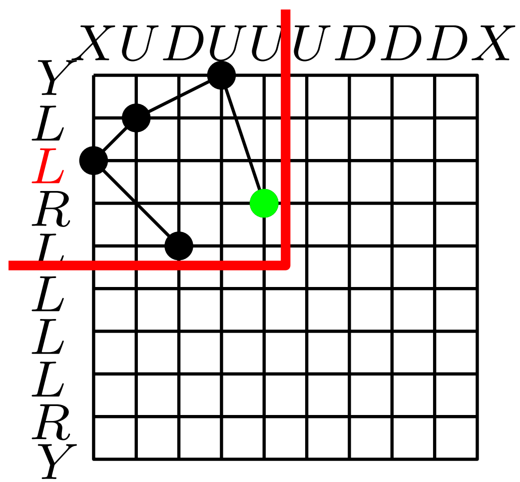
$$u_i = \begin{cases} U, & \text{if } (i, \sigma(i)) \text{ is an upper point that is not colored} \\ D, & \text{otherwise} \end{cases}$$

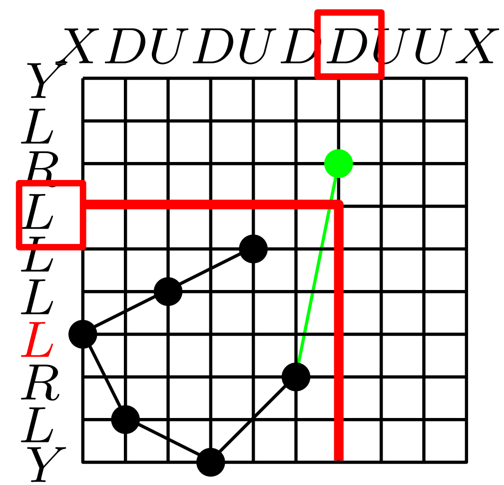
$v_i = Y$ for extremal points in the horizontal borders of the bounding box.

$$v_i = \begin{cases} L, & \text{if } (i, \sigma(i)) \text{ is left point which is not an upper right one or is not colored} \\ R, & \text{otherwise} \end{cases}$$

Decoding a word of $\mathcal{M}$







$$\mathcal{CT} = \mathcal{M} - \mathcal{DT}^{\nwarrow +} \cdot W \cdot (X, Y) - \mathcal{DT}^{\searrow} \cdot \{(D, L)\} \cdot W \cdot (X, Y)$$

Similarly we get the decompositions for $\mathcal{DT}^{\nwarrow +}$ and $\mathcal{DT}^{\searrow}$

By translating into equations as for square permutations we obtain:

$$\mathcal{CT}(t) = \frac{2t^2}{1-4t} + \frac{2t}{(1-4t)^2} - \frac{t^2}{(1-4t)^{3/2}}$$

Final remarks

The 4^{n-3} extra polyominoes...

Enumeration of permutations according to record types

Permutations with few internal points

The 4^{n-3} extra polyominoes

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

done

$$|\mathcal{S}_n| = (2n+4)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

done

$$|\mathcal{CT}_n| = (2n+4)\,4^{n-3} - (2n-3)\binom{2n-4}{n-2}$$

$$= 2n\,4^{n-3} - \left((2n-3)\binom{2n-4}{n-2} - 4^{n-2}\right)$$

The 4^{n-3} extra polyominoes

$$|\mathcal{C}_n| = (2n+5)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

done

$$|\mathcal{S}_n| = (2n+4)\,4^{n-3} - 4(2n-5)\binom{2n-6}{n-3}$$

done

$$|\mathcal{CT}_n| = (2n+4)\,4^{n-3} - (2n-3)\binom{2n-4}{n-2}$$

$$= 2n\,4^{n-3} - \left((2n-3)\binom{2n-4}{n-2} - 4^{n-2}\right)$$

it remains to give a combinatorial interpretation of

$$|\mathcal{C}_n| - |\mathcal{S}_n| = 4^{n-3}$$

Records and internal points

H. Wilf raised the question of enumerating permutations with respect to the numbers of upper-left, upper-right, down-left down-right points (LR-min, LR-max, RL-min, RL-max).

The standard generating tree for all permutations allows to control only two parameters, *e.g.* the numbers of upper right and lower right points.

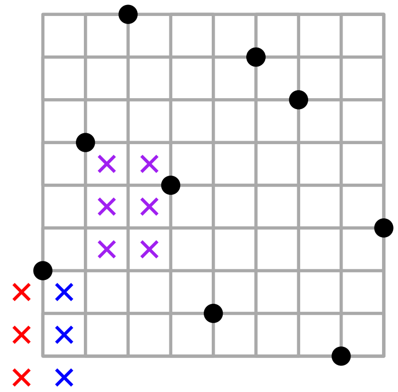
Instead our operator θ allows to control all four parameters in the case of square permutations.

Theorem [D. / Poulalhon 2008]
The refined generating series $S(u, v, w, z; t)$ of square permutations with respect to the number of points of each type is algebraic.

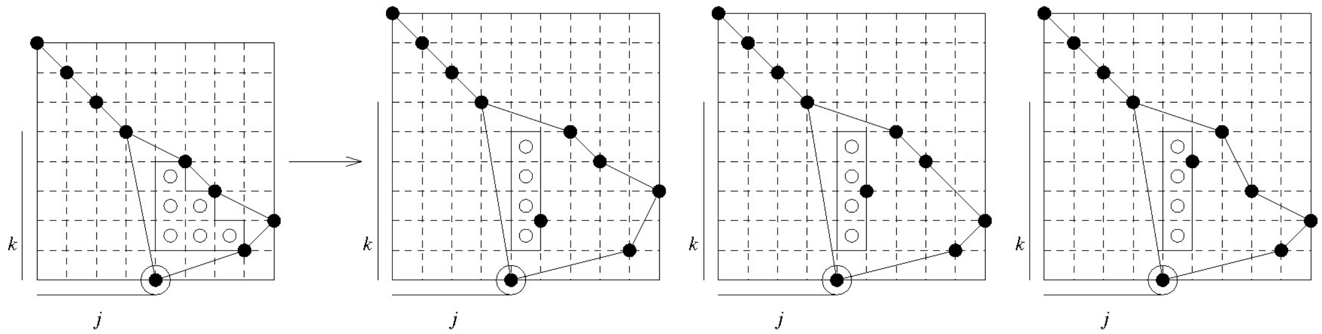
Records and internal points

Moreover the operator θ can be complemented with an operator θ' that introduces internal points one at a time.

This is done by defining a suitable set of internal active sites:



and describing the associated generating tree:



AxiomePermutation:

$Perm$	$Class$	u	v	w	x	y	z	$loop?$	$monomial$	sum
2,1	A, Des, U	1	1	0	0	0	0		uv	
1,2	B, Ang, C	0	0	0	0	0	0		1	

RegleA:

$Perm$	$Class$	u	v	w	x	y	z	$loop?$	$monomial$	sum
$IC(1,1,s)$	B, Ang, C	0	0	0	$k+l-1$	$k-1$	j		$(xy)^k z^j x^l (xy)^{-1}$	
$IC(i+1,1,s)$	B, Asc, C	i	$j+1$	0	$k+l-i$	$k-i$	j	$i=1..k-1$	$(xy)^k (vz)^j x^l v (u/xy)^i$	$\sum(u/xy)^i = \frac{(u/xy)-(u/xy)^k}{1-(u/xy)}$
$IC(k+i+1,1,s)$	B, Asc, U	k	$j+1$	i	$l-i$	1	$j-1-i$	$i=0..l-1$	$u^k (vz)^j x^l vyz^{-1} (w/xz)$	
$IC(k+l+1,1,s)$	B, Asc, U	k	$j+1$	l	0	0	0		$u^k v^j w^l v$	
$IC(1,2,s)$	A, Des, U	$k+l+1$	1	0	$k+l-1$	$k-1$	j		$(uxy)^k z^j (ux)^l uv (xy)^{-1}$	
$IC(i+1,2,s)$	C, Des, U	$k+l-i+1$	1	0	$k+l-i$	$k-i$	j	$i=1..k-1$	$(uxy)^k z^j (ux)^l uv (uxy)^{-i}$	
$IC(k+1+i,2,s)$	C, Des, U	$l+1-i$	1	0	$l-i$	1	$j-1-i$	$i=0..l-1$	$z^j (ux)^l z^j (uvy/z)(uxz)^{-i}$	
$IC(k+l+1,2,s)$	A, Des, U	k	$j+1$	$l+1$	0	0	0		$u^k v^j w^j vw$	

RegleB0

$Perm$	$Class$	u	v	w	x	y	z	$loop?$	$monomial$	sum
$IC(1,1,s)$	B, Ang, C	0	0	0	0	0	0		1	
$IC(1,2,s)$	B, Des, C	1	1	0	0	0	0		uv	

RegleB with $l = 0$ and covered

$Perm$	$Class$	u	v	w	x	y	z	$loop?$	$monomial$	sum
$IC(1,1,s)$	B, Ang, C	0	0	0	k	k	j		$(xy)^k z^j$	
$IC(1+i,1,s)$	B, Asc, C	i	$j+1$	0	$k+1-i$	$k+1-i$	j	$i=1..k$	$(xy)^k (vz)^j vxy (u/xy)^i$	
$IC(1,2,s)$	B, Des, C	$k+1$	1	0	k	k	j		$(uxy)^k z^j uv$	
$IC(1+i,2,s)$	D, Des, C	$k-i+1$	1	0	$k-i+1$	$k-i+1$	j	$i=1..k$	$(uxy)^k z^j uvxy (uxy)^{-i}$	

RegleB otherwise

$Perm$	$Class$	u	v	w	x	y	z	$loop?$	$monomial$	sum
$IC(1,1,s)$	B, Ang, C	0	0	0	$k+l$	$k-1$	j		$(xy)^k z^j x^l / y$	
$IC(1+i,1,s)$	B, Asc, C	i	$j+1$	0	$k+l+1-i$	$k-i$	j	$i=1..k-1$	$(xy)^k (vz)^j x^l vx (u/xy)^{-i}$	
$IC(k+1+i,1,s)$	B, Asc, U	k	$j+1$	i	$l+1-i$	1	$j-1-i$	$i=0..l$	$u^k (vz)^j x^l vxy/z (w/xz)^i$	
$IC(1,2,s)$	B, Des, C	$k+l+1$	1	0	$k+l$	$k-1$	j		$(uxy)^k z^j (ux)^l uv/y$	
$IC(1+i,2,s)$	D, Des, C	$k+l-i+1$	1	0	$k+l-i+1$	$k-i$	j	$i=1..k-1$	$(uxy)^k z^j (ux)^l uvx (uxy)^{-i}$	
$IC(k+1+i,2,s)$	D, Des, C	$l+1-i$	1	0	$l+1-i$	1	$j-1-i$	$i=0..l$	$z^j (ux)^l uvxy/z (uxz)^{-i}$	

ReglеC

Perm	*Class*	*u*	*v*	*w*	*x*	*y*	*z*	*loop?*	*monomial*	*sum*
$IC(s[j+1]+i,1,s)]$	D, Asc, C		i	$j+1$	0	$k+l-i$	$k-i$	$l+1$	$i=1..k-1$	
$IC(s[j+1]+k+i,1,s)]$	D, Asc, U	k	$j+1$	i	$l-i$	1	$l-i$	$i=0..l-1$		
$IC(s[j+1]+k+l,1,s)]$	D, Asc, U	k	$j+1$	l	0	0	0			
$IC(s[j+1]+i,2,s)]$	C, Des, U	$k+l-i+1$	1	0	$k+l-i$	$k-i$	$l+1$	$i=1..k-1$		
$IC(s[j+1]+k+i,2,s)]$	C, Des, U	$l-i+1$	1	0	$l-i$	1	$l-i$	$i=0..l-1$		
$IC(s[j+1]+k+l,2,s)]$	C, Des, U	k	$j+1$	$l+1$	0	0	0			

RegleD with $l=0$ and covered

Perm	*Class*	*u*	*v*	*w*	*x*	*y*	*z*	*loop?*	*monomial*	*sum*
$IC(s[j+1]+i,1,s)]$	D, Asc, C	i	$j+1$	0	$k-i+1$	$k-i+1$	j	$i=1..k-1$		
$IC(s[j+1]+k,1,s)]$	D, Asc, C	k	$j+1$	0	1	1	j			
$IC(s[j+1]+i,2,s)]$	D, Des, C	$k+l-i+1$	1	0	$k-i+1$	$k-i+1$	j	$i=1..k-1$		
$IC(s[j+1]+k,2,s)]$	D, Des, C	1	1	0	1	1	j			

RegleD otherwise

Perm	*Class*	*u*	*v*	*w*	*x*	*y*	*z*	*loop?*	*monomial*	*sum*
$IC(s[j+1]+i,1,s)]$	D, Asc, C	i	$j+1$	0	$k+l-i+1$	$k-i$	j	$i=1..k-1$		
$IC(s[j+1]+k+i,1,s)]$	D, Asc, U	k	$j+1$	i	$l+1-i$	1	$j-1-i$	$i=0..l$		
$IC(s[j+1]+i,2,s)]$	D, Des, C	$k+l-i+1$	1	0	$k+l-i+1$	$k-i$	j	$i=1..k-1$		
$IC(s[j+1]+k+i,2,s)]$	D, Des, C	$l+1-i$	1	0	$l+1-i$	1	$j-1-i$	$i=0..l$		

Interior points:

```
if inizio = Ang  then interieur := RegleI(classe,inizio,covered,k,j,l,p,q,r,sigma) fi;
if inizio = Des  then interieur := RegleI(classe,inizio,covered,k+1,j,l,p,q,r,sigma) fi;
if inizio = Asc  then interieur := RegleI(classe,inizio,covered,k,j+1,l,p,q,r,sigma) fi;
```

RegleI

```
[seq([seq([classe,inizio,covered,k,j,l,p+1,p+1,i,
             IC(max(2,min(s[1],s[2]))+t,2+i,s),X],t=1..p)],
      i=1..r-p+q),
         seq([seq([classe,inizio,covered,k,j,l,p+1,p+1-i,r-p+q+i,
               IC(max(2,min(s[1],s[2]))+t,2+r-p+q+i,s),X],t=1..(p-i))],
            i=1..p-q)]
```

Conclusion

The formula for square permutations thus extends to the following:

Theorem (Disanto, D. Rinaldi, Schaeffer)
For all $i \geq 0$ the generating function $S^{(i)}(t)$ of permutations with i internal points is rational in the Catalan series.
More generally the refined generating function with respect to the four types of points is algebraic.

A natural question could be to give a similar statement extending the results for convex polyominoes or convex permutominoes...

Merci !