27 February 2025

Lattice Paths with Flexible Boundaries: Patterns, Automata, and Counting

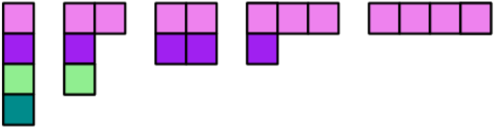
Enumerative Combinatorics and Effective Aspects of Differential Equations

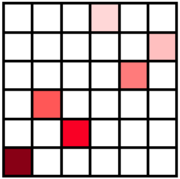
WARWICK
THE UNIVERSITY OF WARWICK

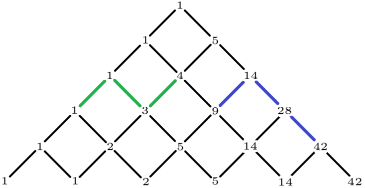
Outline

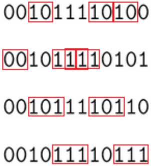
1. Patterns in lattice paths via automata
2. Patterns in the quarter plane
3. Lattice paths with dynamic boundary

Combinatorial structures where patterns are commonly studied:









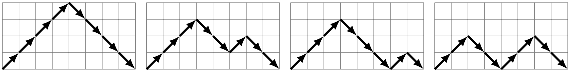
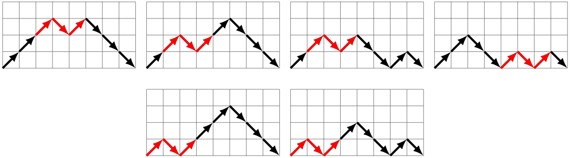
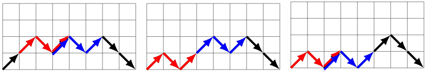
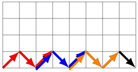
(links with sorting algorithms, logic, number theory, bioinformatics, . . .)

Typical questions:

- What is number of structures of size n with j occurrences of the pattern?
- Is there a nice formula for the generating function?
- Asymptotic behaviour, limit laws?
- Generation of these objects?

Simple example: Dyck paths with pattern ↗↘↗

$$D(z, v) = 1 + z^2 + (v+1)z^4 + (v^2 + 2v + 2)z^6 + (v^3 + 3v^2 + 6v + 4)z^8 + \cdots$$

# P	Dyck paths of length 8 with # P occurrences of ↗↘↗	
0		$4u^0$
1		$6u^1$
2		$3u^2$
3		u^3

Ad hoc method (standard combinatorial decomposition)

$$\mathcal{D} \;=\; \varepsilon + \nearrow \mathcal{D} \searrow \mathcal{D}$$

$$D(z,v) = 1 + {\color{red}v}z^2(D(z,v)-1) + z^2 + z^2(D(z,v)-1)D(z,v)$$

$$\varepsilon \qquad {\color{red}\nearrow\searrow}\, \mathcal{D}\setminus\{\varepsilon\} \qquad \nearrow\searrow \qquad \nearrow \mathcal{D}\setminus\{\varepsilon\} \searrow \mathcal{D}$$

Ad hoc method (standard combinatorial decomposition)

$$\mathcal{D} \;=\; \varepsilon + \nearrow \mathcal{D} \searrow \mathcal{D}$$

$$D(z,v) = 1 + vz^2(D(z,v) - 1) + z^2 + z^2(D(z,v) - 1)D(z,v)$$

$$\varepsilon \qquad \nearrow\searrow \mathcal{D} \setminus \{\varepsilon\} \qquad \nearrow\searrow \qquad \nearrow \mathcal{D} \setminus \{\varepsilon\} \searrow \mathcal{D}$$

$$D(z,v) = \frac{1 + z^2 - z^2 v - \sqrt{(1 + z^2 - vz^2)(1 - 3z^2 - vz^2)}}{2z^2}$$

- $D(z,1)$: standard generating function for Dyck paths.
- $D(z,0)$: generating function for Dyck paths with no occurrence of the pattern.
- $\frac{\partial}{\partial v} D(z,v)\Big|_{v=1}$: generating function for the total number of occurrences of a pattern.

Ad hoc method (standard combinatorial decomposition)

$$\mathcal{D} = \varepsilon + \nearrow \mathcal{D} \searrow \mathcal{D}$$

$$D(z,v) = 1 + vz^2(D(z,v)-1) + z^2 + z^2(D(z,v)-1)D(z,v)$$

$$\varepsilon \qquad \nearrow\searrow \mathcal{D}\setminus\{\varepsilon\} \qquad \nearrow\searrow \qquad \nearrow \mathcal{D}\setminus\{\varepsilon\} \searrow \mathcal{D}$$

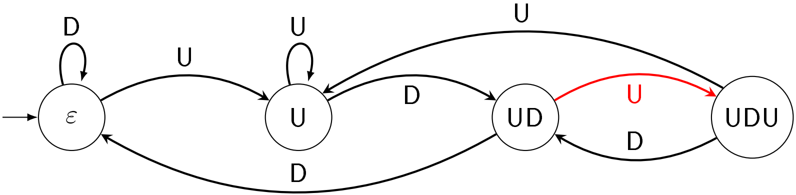
$$D(z,v) = \frac{1+z^2-z^2v-\sqrt{(1+z^2-vz^2)(1-3z^2-vz^2)}}{2z^2}$$

$D(z,1)$:	standard generating function for Dyck paths.
$D(z,0)$:	generating function for Dyck paths with no occurrence of the pattern.
$\frac{\partial}{\partial v}D(z,v)\Big\rvert_{v=1}$:	generating function for the total number of occurrences of a pattern.

Doing this in general is unpleasant!

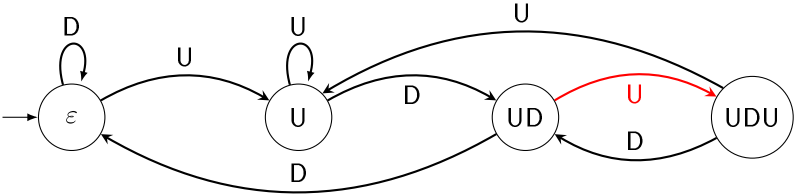
General method: via automata/walks on graphs

First step: build an automaton encoding the structure of the pattern (here UDU).

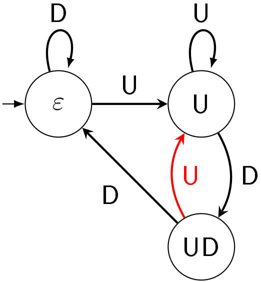


General method: via automata/walks on graphs

First step: build an automaton encoding the structure of the pattern (here UDU).



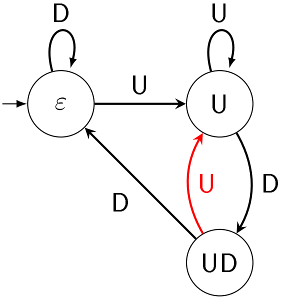
The associated minimal automaton (De Bruijn) is:



Each time the red transition is used indicates an occurrence of the pattern UDU.

Unconstrained Dyck paths with ↗↘↗

Second step: set up the adjacency matrix A.



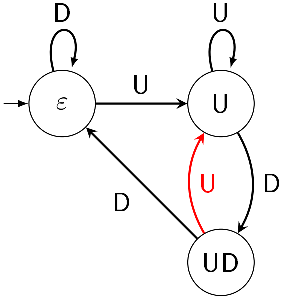
$$A = \begin{bmatrix} u^{-1} & u & 0 \\ 0 & u & u^{-1} \\ u^{-1} & vu & 0 \end{bmatrix}$$

(u marks height, v pattern).
Generating function for all possible paths
(z marks length):

$$I + zA + (zA)^2 + (zA)^3 + \cdots = (I - zA)^{-1}$$

Unconstrained Dyck paths with 

Second step: set up the adjacency matrix A.



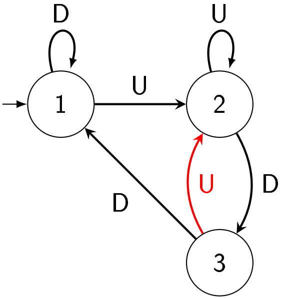
$$A = \begin{bmatrix} u^{-1} & u & 0 \\ 0 & u & u^{-1} \\ u^{-1} & vu & 0 \end{bmatrix}$$

(u marks height, v pattern).
Generating function for all possible paths (z marks length):

$$I + zA + (zA)^2 + (zA)^3 + \cdots = (I - zA)^{-1}$$

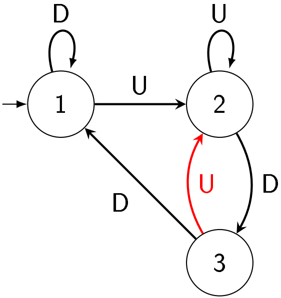
$$G(z,u,v) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} (I - zA)^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$G(z,u,v) = \frac{(vz^2 - z^2 - 1)u}{vuz^2 - vz^3 + u^2 z - uz^2 + z^3 - u + z}$$



However, this automaton does not keep track of height. For this, we use a stack:

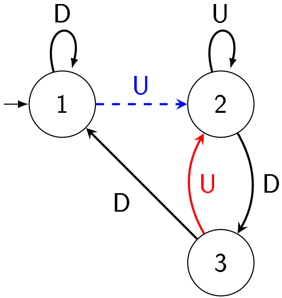
Read U: push U
Read D: pop U
Cannot pop from an empty stack!



However, this automaton does not keep track of height. For this, we use a stack:

Read U: push U
Read D: pop U
Cannot pop from an empty stack!

Word: UUUDUDDUDU

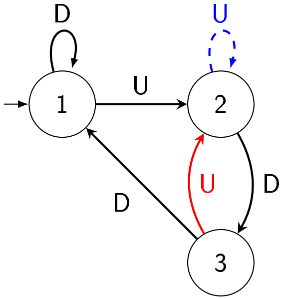


However, this automaton does not keep track of height. For this, we use a stack:

Read U: push U
Read D: pop U
Cannot pop from an empty stack!

U			...

Word: UUUDUDDUDU

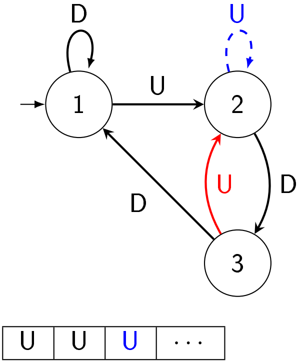


U	U		...

Word: UUUDUDDUDU

However, this automaton does not keep track of height. For this, we use a stack:

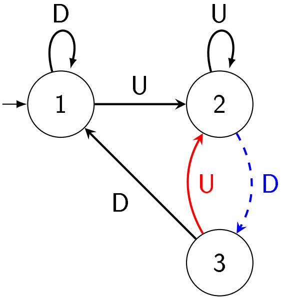
Read U: push U
Read D: pop U
Cannot pop from an empty stack!



However, this automaton does not keep track of height. For this, we use a stack:

Read U: push U
Read D: pop U
Cannot pop from an empty stack!

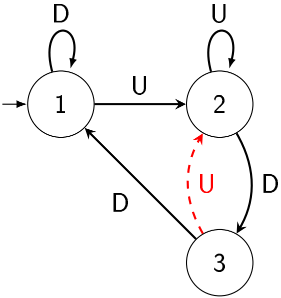
Word: UUUDUDDUDU



However, this automaton does not keep track of height. For this, we use a stack:

Read U: push U
Read D: pop U
Cannot pop from an empty stack!

Word: UUUDUDDUDU

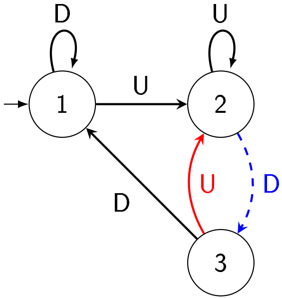


U	U	U	···

Word: UUUDUDDUDU

However, this automaton does not keep track of height. For this, we use a stack:

Read U: push U
Read D: pop U
Cannot pop from an empty stack!

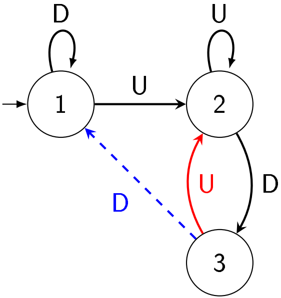


U	U		...

Word: UUDUDDUDU

However, this automaton does not keep track of height. For this, we use a stack:

Read U: push U
Read D: pop U
Cannot pop from an empty stack!

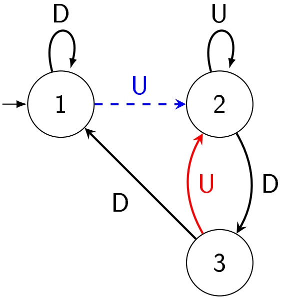


However, this automaton does not keep track of height. For this, we use a stack:

Read U: push U
Read D: pop U
Cannot pop from an empty stack!

U			···

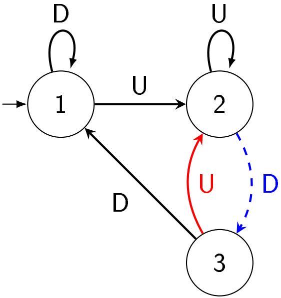
Word: UUUDUDDUDU



However, this automaton does not keep track of height. For this, we use a stack:

Read U: push U
Read D: pop U
Cannot pop from an empty stack!

Word: UUUDUDDUDU

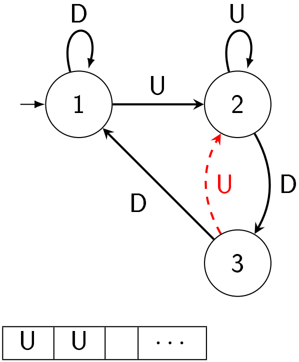


However, this automaton does not keep track of height. For this, we use a stack:

Read U: push U
Read D: pop U
Cannot pop from an empty stack!

U			...

Word: UUUDUDDUDU



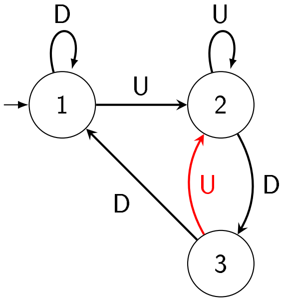
However, this automaton does not keep track of height. For this, we use a stack:

Read U: push U
Read D: pop U
Cannot pop from an empty stack!

Word: UUUDUDDUDU

Positivity constraint: add a stack to the automaton!

Third step: positivity constraint at the level of the functional equation.



Adjacency matrix (u marks height):

$$A = \begin{bmatrix} u^{-1} & u & 0 \\ 0 & u & u^{-1} \\ u^{-1} & 0 & 0 \end{bmatrix}$$

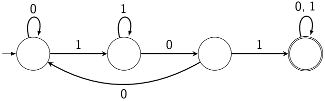
$M_i(z,u)$ is the generating function for meanders terminating in state i.

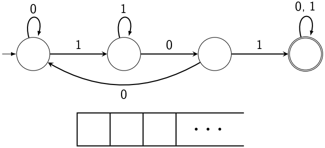
$$(M_1, M_2, M_3) = (1,0,0) + z(M_1, M_2, M_3)A - z\{u^{<0}\}((M_1, M_2, M_3)A).$$

$$(M_1, M_2, M_3)(I - zA) = (1,0,0) - z((M_1(z,0), M_2(z,0), M_3(z,0))A).$$

$\rightsquigarrow$ vectorial kernel method [Asinowski, Bacher, Banderier, Gittenberger, 2019].

Generating functions

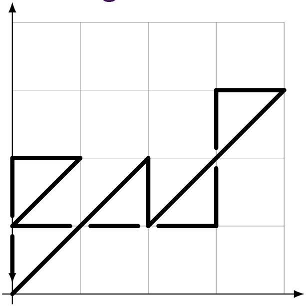
Deterministic Finite Automaton	Rational Generating Function
	$G(z) = \frac{p(z)}{q(z)}$

Pushdown Automaton (subclass)	Algebraic Generating Function
	$P(z, G(z)) = 0$

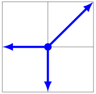
- Unambiguous context-free grammar $\implies$ algebraic
  [Chomsky-Schützenberger, 1963]
- 'Shuffles' of context-free languages $\implies$ D-finite
  [Mishna, Zabrocki, 2008]
- Two-way reversal bounded counter machines $\implies$ D-finite
  [Bostan, Carayol, Koechlin, Nicaud, 2020]

Outline

Unravelling a formula!

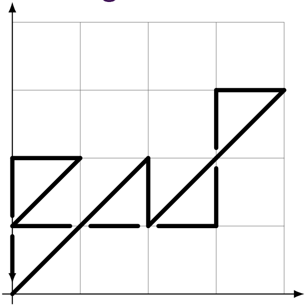


$\mathcal{S} = \{(1,1),(0,-1),(-1,0)\}.$

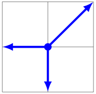


Excursion: $(0,0)$ to $(0,0)$
Meander: $(0,0)$ to anywhere

Unravelling a formula!



$$\mathcal{S} = \{(1,1),(0,-1),(-1,0)\}.$$

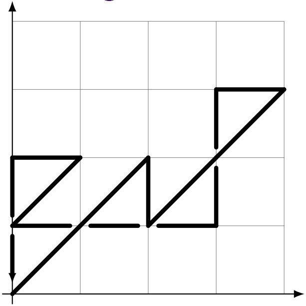


Excursion: $(0,0)$ to $(0,0)$
Meander: $(0,0)$ to anywhere

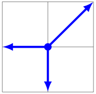
How many such excursions are there?

$$\frac{4^n}{(n+1)(2n+1)}\binom{3n}{n}$$

Unravelling a formula!



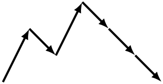
$\mathcal{S} = \{(1,1),(0,-1),(-1,0)\}.$



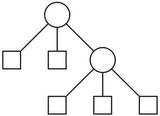
Excursion: $(0,0)$ to $(0,0)$
Meander: $(0,0)$ to anywhere

How many such excursions are there?

$$\frac{4^n}{(n+1)(2n+1)}\binom{3n}{n}$$

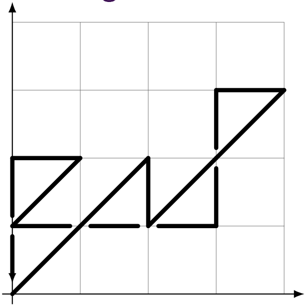


$\mathcal{S} = \{(1,2),(1,-1)\}$

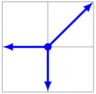


Ternary trees

Unravelling a formula!



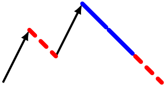
$\mathcal{S} = \{(1,1),(0,-1),(-1,0)\}.$



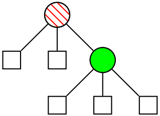
Excursion: $(0,0)$ to $(0,0)$
Meander: $(0,0)$ to anywhere

How many such excursions are there?

$$\frac{4^n}{(n+1)(2n+1)}\binom{3n}{n}$$

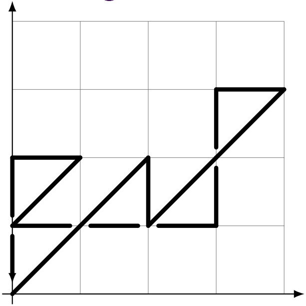


$\mathcal{S} = \{(1,2),(1,-1)\}$

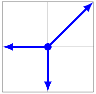


Ternary trees

Unravelling a formula!



$\mathcal{S} = \{(1,1),(0,-1),(-1,0)\}$.



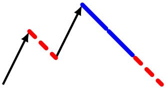
Excursion: $(0,0)$ to $(0,0)$
Meander: $(0,0)$ to anywhere

How many such excursions are there?

$$\frac{4^n}{(n+1)(2n+1)}\binom{3n}{n}$$

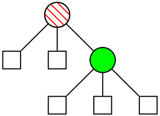
Equivalence classes!

2-colouring



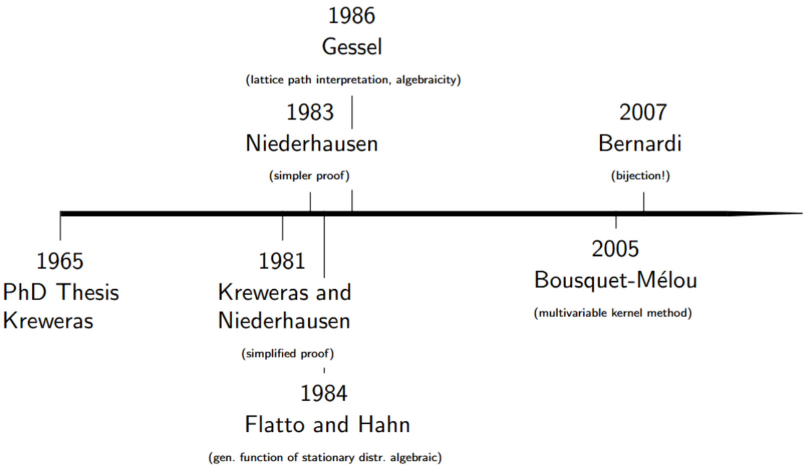
$\mathcal{S} = \{(1,2),(1,-1)\}$

4-colouring



Ternary trees

A brief history of Kreweras walks



Nature of Kreweras generating function?

Kreweras walks are algebraic – can they then be represented by a pushdown automaton?

Nature of Kreweras generating function?

Kreweras walks are algebraic – can they then be represented by a pushdown automaton?

Lemma (Pumping lemma for context-free languages)

If a language L can be represented by some pushdown automaton, there exists some integer $p \geq 1$ (called a "pumping length") such that every word w in L that has a length of p or more symbols (i.e. with $|w| \geq p$) can be written as

$$w = uvzxy$$

with subwords u, v, z, x and y, such that

- $|vx| \geq 1$,
- $|vzx| \leq p$, *and*
- $uv^n zx^n y \in L$ *for all* $n \geq 0$.

Nature of Kreweras generating function?

Kreweras walks are algebraic – can they then be represented by a pushdown automaton?

Lemma (Pumping lemma for context-free languages)

If a language L can be represented by some pushdown automaton, there exists some integer $p \geq 1$ (called a "pumping length") such that every word w in L that has a length of p or more symbols (i.e. with $|w| \geq p$) can be written as

$$w = uvzxy$$

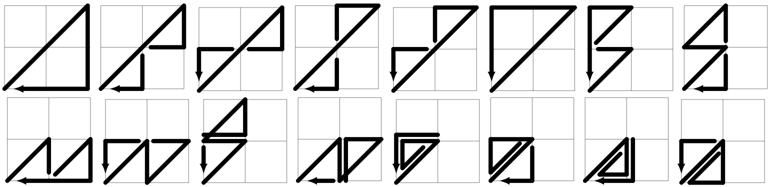
with subwords u, v, z, x and y, such that

- $|vx| \geq 1$,
- $|vzx| \leq p$, *and*
- $uv^n zx^n y \in L$ *for all* $n \geq 0$.

Applying the pumping lemma to $a^p b^p c^p$ shows that Kreweras walks cannot be encoded in a pushdown automaton. $\rightsquigarrow$ Are there other ways to approach studying patterns in walks in the quarter plane?

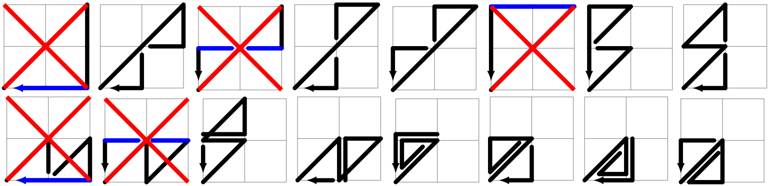
Patterns in walks in the quarter plane: A humble start

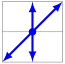
Kreweras excursions of length 6



Patterns in walks in the quarter plane: A humble start

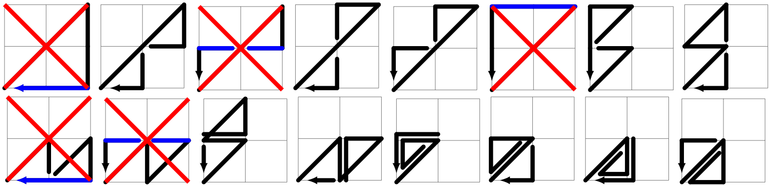
Kreweras excursions of length 6 avoiding the pattern ←←.



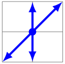
1, 2, 11, 85, 782, 8004, . . . OEIS A135404: "Gessel excursions" 

Patterns in walks in the quarter plane: A humble start

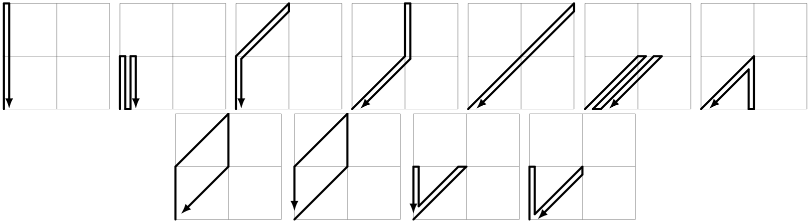
Kreweras excursions of length 6 avoiding the pattern ←←.



1, 2, 11, 85, 782, 8004, . . . OEIS A135404: "Gessel excursions"

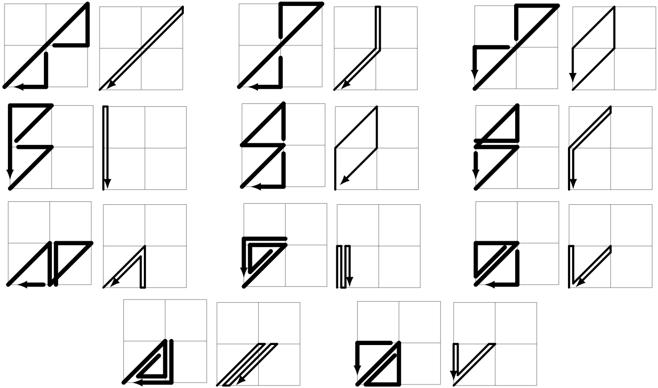


Gessel excursions of length 4.



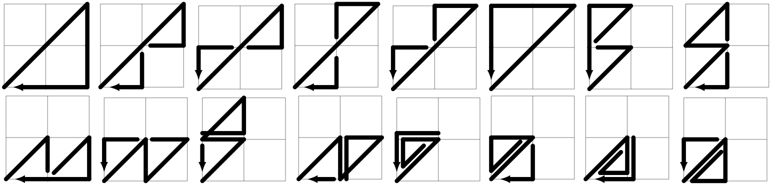
Patterns in walks in the quarter plane: A humble start

‘Shortcut’ bijection (Asinowski, Banderier, S.):



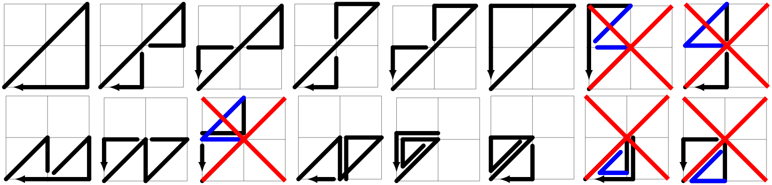
What about other patterns?

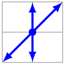
Kreweras excursions of length 6



What about other patterns?

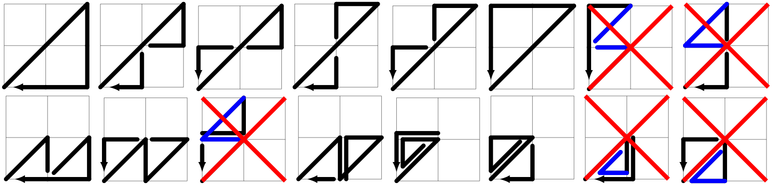
Kreweras excursions of length 6 avoiding the pattern .

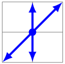


1, 2, 11, 85, 782, 8004, . . . OEIS A135404: "Gessel excursions" 

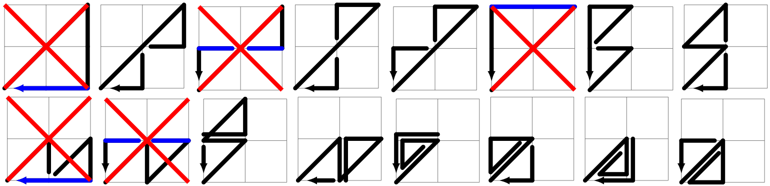
What about other patterns?

Kreweras excursions of length 6 avoiding the pattern .



1, 2, 11, 85, 782, 8004, . . . OEIS A135404: "Gessel excursions" 

Kreweras excursions of length 6 avoiding the pattern .



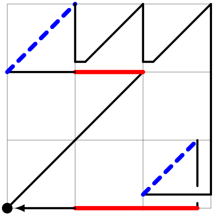
What about other patterns?

Proposition (Asinowski, Banderier, S.)

The number of Kreweras excursions of length $3n$ with

$$\text{“}k \leftarrow\leftarrow \text{ and } \ell \swarrow\!\!\nearrow\text{”} = \text{“}\ell \leftarrow\leftarrow \text{ and } k \swarrow\!\!\nearrow\text{”}.$$

- Mark each $\leftarrow$ followed by a $\leftarrow$ (local indices $a_1, \ldots, a_k$).
- Mark each $\nearrow$ preceded by a $\leftarrow$ (local indices $b_1, \ldots, b_\ell$).



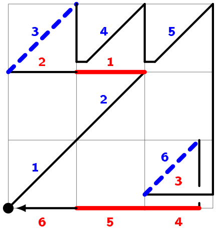
What about other patterns?

Proposition (Asinowski, Banderier, S.)

The number of Kreweras excursions of length $3n$ with

$$\text{“}k \leftarrow\leftarrow \text{ and } \ell \swarrow\!\nearrow\text{”} = \text{“}\ell \leftarrow\leftarrow \text{ and } k \swarrow\!\nearrow\text{”}.$$

- Mark each $\leftarrow$ followed by a $\leftarrow$ (local indices $a_1, \ldots, a_k$).
- Mark each $\nearrow$ preceded by a $\leftarrow$ (local indices $b_1, \ldots, b_\ell$).



(1, 4, 5)

(3, 6)

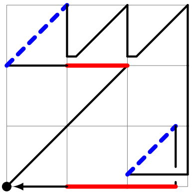
What about other patterns?

Proposition (Asinowski, Banderier, S.)

The number of Kreweras excursions of length $3n$ with

$$\text{“}k \leftarrow\leftarrow \text{ and } \ell \nearrow\!\!\swarrow\text{”} = \text{“}\ell \leftarrow\leftarrow \text{ and } k \nearrow\!\!\swarrow\text{”}.$$

1. For $i = 1 \dots k$, remove the $\leftarrow$ step with index a_i and insert it immediately before the $\nearrow$ step with index $a_i + 1$.



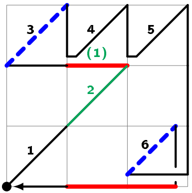
What about other patterns?

Proposition (Asinowski, Banderier, S.)

The number of Kreweras excursions of length $3n$ with

$$\text{“}k \leftarrow\leftarrow \text{ and } \ell \swarrow\nearrow\text{”} = \text{“}\ell \leftarrow\leftarrow \text{ and } k \swarrow\nearrow\text{”}.$$

1. For $i = 1 \dots k$, remove the $\leftarrow$ step with index a_i and insert it immediately before the $\nearrow$ step with index $a_i + 1$.



move (1) before 2

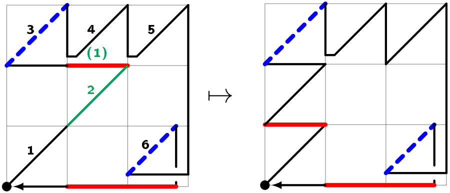
What about other patterns?

Proposition (Asinowski, Banderier, S.)

The number of Kreweras excursions of length $3n$ with

$$\text{“}k \leftarrow\leftarrow \text{ and } \ell \nearrow\!\leftarrow\text{”} = \text{“}\ell \leftarrow\leftarrow \text{ and } k \nearrow\!\leftarrow\text{”}.$$

1. For $i = 1 \ldots k$, remove the $\leftarrow$ step with index a_i and insert it immediately before the $\nearrow$ step with index $a_i + 1$.



move (1) before 2

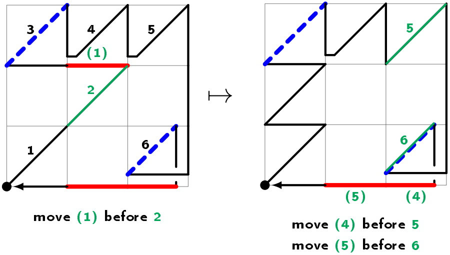
What about other patterns?

Proposition (Asinowski, Banderier, S.)

The number of Kreweras excursions of length $3n$ with

$$\text{“}k \leftarrow\!\leftarrow \text{ and } \ell \swarrow\!\!\nearrow\text{”} = \text{“}\ell \leftarrow\!\leftarrow \text{ and } k \swarrow\!\!\nearrow\text{”}.$$

1. For $i = 1 \ldots k$, remove the $\leftarrow$ step with index a_i and insert it immediately before the $\nearrow$ step with index $a_i + 1$.



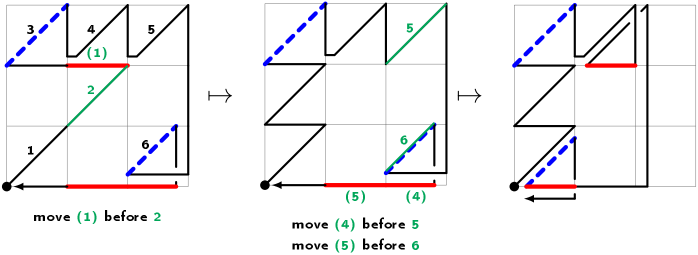
What about other patterns?

Proposition (Asinowski, Banderier, S.)

The number of Kreweras excursions of length $3n$ with

$$\text{“}k \leftarrow\leftarrow \text{ and } \ell \nearrow\!\leftarrow\text{”} = \text{“}\ell \leftarrow\leftarrow \text{ and } k \nearrow\!\leftarrow\text{”}.$$

1. For $i = 1 \ldots k$, remove the $\leftarrow$ step with index a_i and insert it immediately before the $\nearrow$ step with index $a_i + 1$.



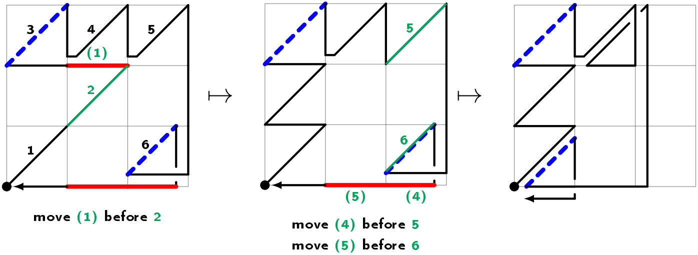
What about other patterns?

Proposition (Asinowski, Banderier, S.)

The number of Kreweras excursions of length 3n with

$$\text{“}k \leftarrow\leftarrow \text{ and } \ell \nearrow\!\!\leftarrow\text{”} = \text{“}\ell \leftarrow\leftarrow \text{ and } k \nearrow\!\!\leftarrow\text{”}.$$

1. For $i = 1 \dots k$, remove the $\leftarrow$ step with index a_i and insert it immediately before the $\nearrow$ step with index $a_i + 1$.



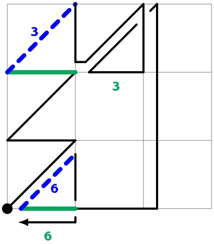
What about other patterns?

Proposition (Asinowski, Banderier, S.)

The number of Kreweras excursions of length $3n$ with

$$\text{“}k \leftarrow\leftarrow \text{ and } \ell \nearrow\!\leftarrow\text{”} = \text{“}\ell \leftarrow\leftarrow \text{ and } k \nearrow\!\leftarrow\text{”}.$$

2. For $j = \ell \dots 1$, remove the $\leftarrow$ step before the $\nearrow$ step with index b_j and insert it immediately before the $\leftarrow$ step with index b_j.



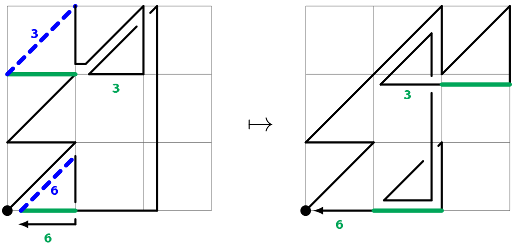
What about other patterns?

Proposition (Asinowski, Banderier, S.)

The number of Kreweras excursions of length $3n$ *with*

$$\text{“}k \leftarrow\leftarrow \text{ and } \ell \swarrow\!\!\nearrow\text{”} = \text{“}\ell \leftarrow\leftarrow \text{ and } k \swarrow\!\!\nearrow\text{”}.$$

2. For $j = \ell \ldots 1$, remove the $\leftarrow$ step before the $\nearrow$ step with index b_j and insert it immediately before the $\leftarrow$ step with index b_j.



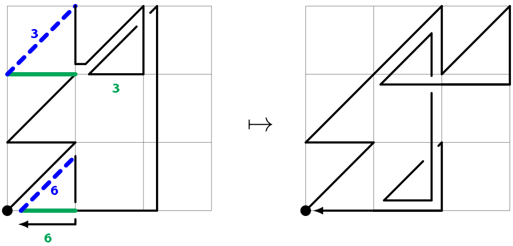
What about other patterns?

Proposition (Asinowski, Banderier, S.)

The number of Kreweras excursions of length 3n with

$$\text{“}k \leftarrow\!\leftarrow \text{ and } \ell \swarrow\!\nearrow\text{”} = \text{“}\ell \leftarrow\!\leftarrow \text{ and } k \swarrow\!\nearrow\text{”}.$$

2. For $j = \ell \ldots 1$, remove the $\leftarrow$ step before the $\nearrow$ step with index b_j and insert it immediately before the $\leftarrow$ step with index b_j.



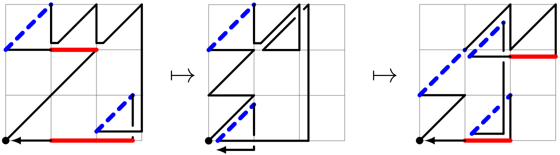
What about other patterns?

Proposition (Asinowski, Banderier, S.)

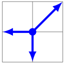
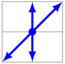
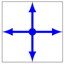
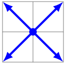
The number of Kreweras excursions of length 3n with

$$\text{“}k \leftarrow\leftarrow \text{ and } \ell \swarrow\!\!\nearrow\text{”} = \text{“}\ell \leftarrow\leftarrow \text{ and } k \swarrow\!\!\nearrow\text{”}.$$

- Mark each $\leftarrow$ followed by a $\leftarrow$ (local indices $a_1, \ldots, a_k$).
- Mark each $\nearrow$ preceded by a $\leftarrow$ (local indices $b_1, \ldots, b_\ell$).
- For $i = 1 \ldots k$, remove the $\leftarrow$ step with index a_i and insert it immediately before the $\nearrow$ step with index $a_i + 1$.
- For $j = \ell \ldots 1$, remove the $\leftarrow$ step before the $\nearrow$ step with index b_j and insert it immediately before the $\leftarrow$ step with index b_j.



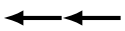
Kreweras excursions avoiding patterns of length 2

			
Kreweras	Gessel	Pólya	Diagonal

Pattern p	**#** $\{$**Kreweras excursions of length** $3n$ **avoiding** $p\}$	**OEIS**	**In bijection with**
	1, 2, 11, 85, 782, 8004, . . .	A135404	Gessel excursions
	1, 2, 11, 85, 782, 8004, . . .	A135404	Gessel excursions
	1, 1, 5, 37, 332, 3343, . . .	None	Gessel excursions ending with $\downarrow$
	1, 2, 10, 70, 588, 5544, . . .	A005568	Pólya excursions
	1, 1, 4, 25, 196, 1764, . . .	A001246	Diagonal excursions

Coincidence?

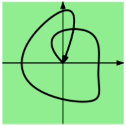
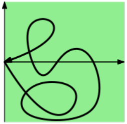
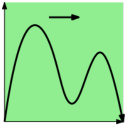
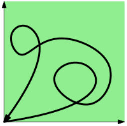
Nature of generating functions in specific cases

Pattern p	# {Kreweras excursions of length $3n$ avoiding p}	OEIS	Nature of gen. func.
	1, 2, 11, 85, 782, 8004, ...	A135404	Algebraic
	1, 2, 11, 85, 782, 8004, ...	A135404	Algebraic
	1, 1, 5, 37, 332, 3343, ...	None	Algebraic
	1, 2, 10, 70, 588, 5544, ...	A005568	D-finite
	1, 1, 4, 25, 196, 1764, ...	A001246	D-finite

First three models: Algebraic
(using [Kauers–Koutschan–Zeilberger 2009] and [Bostan–Kauers 2009])
but not context-free [Banderier–Drmota 2015]: $4^n n^{-2/3}$ asymptotics.
Last two models: D-finite, but not algebraic.

Pólya walks: $C_n C_{n+1} \sim 4 \frac{16^n}{\pi n^3}$. Such asymptotics involving a n^{-3} factor are not compatible with the rather constrained asymptotics of algebraic function coefficients.

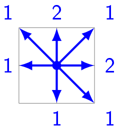
Generating functions: walks avoiding a pattern

For walks in $\mathbb{Z}^2$ avoiding a pattern, generating function is rational.  complement of regular expression	For walks in $\mathbb{N} \times \mathbb{Z}$ avoiding a pattern, generating function is algebraic.  one counter automaton
For directed walks in $\mathbb{N}^2$ avoiding a pattern, generating function is algebraic.  vectorial kernel method [Asinowski, Bacher, Banderier, Gittenberger, Roitner]	For non-directed walks in $\mathbb{N}^2$ avoiding a pattern, gen. function is not necessarily algebraic.  Classification needed!

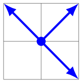
Generating functions: walks avoiding a pattern

For non-directed walks in $\mathbb{N}^2$ avoiding a pattern, the generating function is not necessarily algebraic (or D-finite!).

Kauers–Yatchak's model, 2015
(steps are with multiplicity)

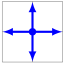


Mishna–Rechnitzer's model, 2009



Forbidding some steps in the top (algebraic) model leads to the bottom (differentially transcendental) model.

What about other models?



Polya

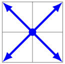
Avoiding SS, conjecturally:

$$
\begin{aligned}
&-6(x+5)G(x)+3(3x^3-43x^2-73x+5)G'(x)\\
&+(11x^4-141x^3-171x^2+13x)G''(x)\\
&+(2x^5-26x^4-26x^3+2x^2)G^{(3)}(x)=0
\end{aligned}
$$

Avoiding NS, conjecturally:

$$
\begin{aligned}
&(-6x^2-24x-6)G(x)+(-12x^4+54x^3-30x^2-90x+6)G'(x)\\
&+(3x^6-48x^5+84x^4+42x^3-87x^2+6x)G''(x)\\
&+(x^7-15x^6+14x^5+14x^4-15x^3+x^2)G^{(3)}(x)=0.
\end{aligned}
$$

What about other models?



Diagonal

Avoiding NE SE, conjecturally:

$$(1470x^2 - 936x + 216)G'(x) + (3570x^3 - 3309x^2 + 900x - 36)G''(x)$$
$$+ (1701x^4 - 1975x^3 + 552x^2 - 28x)G^{(3)}(x)$$
$$+ (189x^5 - 257x^4 + 72x^3 - 4x^2)G^{(4)}(x) = 0$$

Avoiding NW SE, conjecturally:

$$(4x - 4)G(x) + (2x^4 - 26x^3 - 70x^2 - 54x + 4)G'(x)$$
$$+ (4x^5 - 47x^4 - 117x^3 - 61x^2 + 5x)G''(x)$$
$$+ (x^6 - 12x^5 - 26x^4 - 12x^3 + x^2)G^{(3)}(x) = 0$$

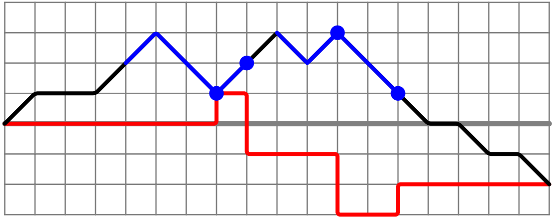
Outline

1. Patterns in lattice paths via automata
2. Patterns in the quarter plane
3. Lattice paths with dynamic boundary

Dynamic boundary problem

Currently, studying a pattern in lattice paths is a static procedure.
Experiment: What happens if the occurrence of a pattern changes the "future" of the path?

- $\mathcal{S} = \{\mathsf{U}, \mathsf{L}, \mathsf{D}\}$ with $\mathsf{U} = (1,1)$, $\mathsf{L} = (1,0)$, $\mathsf{D} = (1,-1)$.
- Pattern UDD shifts the boundary upwards by 1 unit.
- Pattern DU shifts the boundary downwards by 2 units.
- A path must end on the boundary line in its current position.



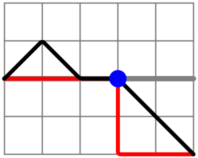
Dynamic boundary problem

Dynamic boundary walks (in one-dimension) are:

- Algebraic
- Able to be modelled with pattern-avoidance problems

Interesting relationships exist between dynamic boundary lattice paths and other combinatorial objects.

Motzkin paths, $\mathcal{S} = \{\mathrm{U}, \mathrm{D}, \mathrm{L}\}$
DL shifts the boundary by -2
UL shifts the boundary by 2.
Set of excursions of length n: $\mathrm{M}(n)$



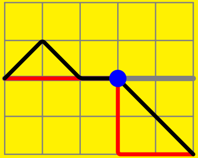
Shifted Young tableau: i-th row begins in position i
Entries: numbers from 1 to n, increasing along rows and columns
Height ≤ 3
Set shifted Young tableaux of size n: $\mathrm{sYT}(n)$

1	2	4
	3	5
		6

Shifted Young tableau: i-th row begins in position i
Entries: numbers from 1 to n, increasing along rows and columns
Height ≤ 3
Set shifted Young tableaux of size n: $\mathrm{sYT}(n)$

1	2	4
	3	5
		6

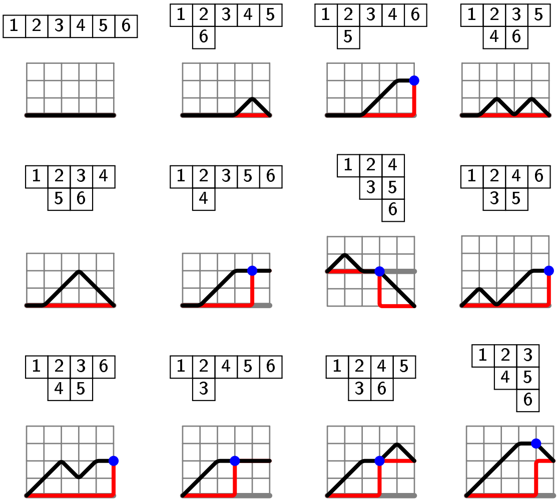
$\mathrm{M}(n) \simeq \mathrm{sYT}(n+1)$ **(Asinowski, Brosch, S.):**



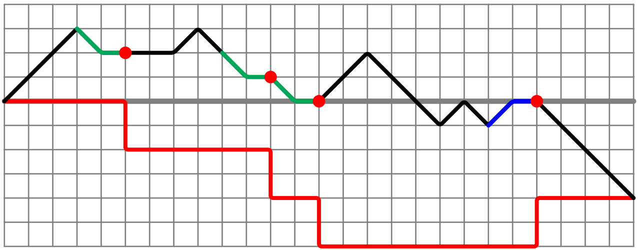
$\simeq$

1	2	4
	3	5
		6

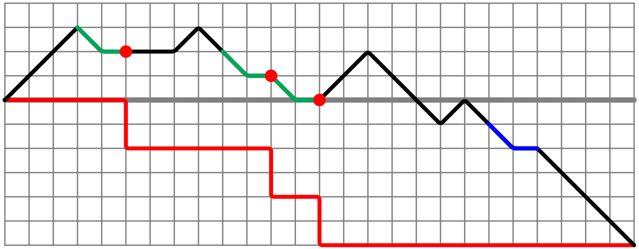
$\mathrm{M}(n) \simeq \mathrm{sYT}(n+1)$



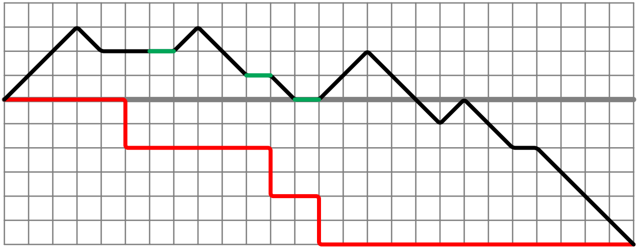
Where the pattern UL occurs, transform it into DL.



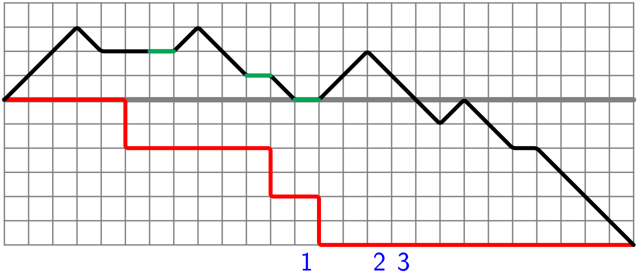
Where the pattern DL occurs, mark the L step. If it is followed by a contiguous sequence of L, remove the marking and mark the final L in this contiguous sequence.



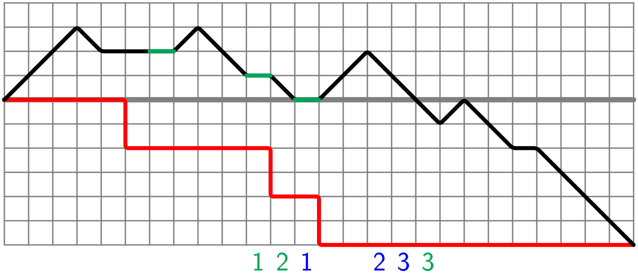
From right to left in the path, map a marked L to 1 and the next (left to right) two D steps to 2 and 3 (in order).



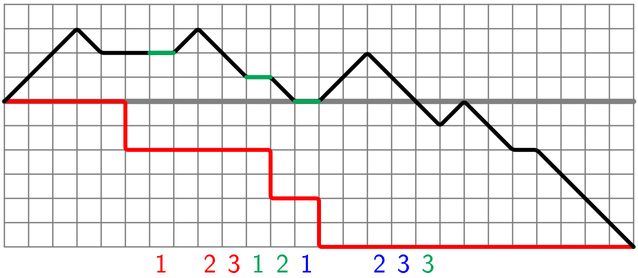
From right to left in the path, map a marked L to 1 and the next (left to right) two D steps to 2 and 3 (in order).



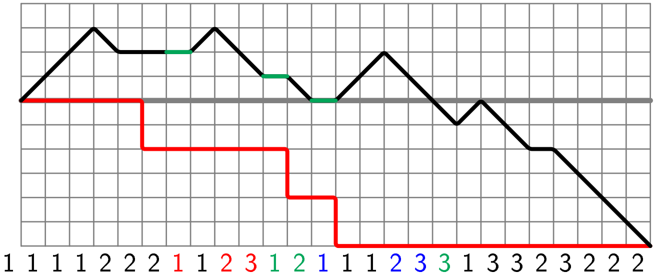
From right to left in the path, map a marked L to 1 and the next (left to right) two D steps to 2 and 3 (in order).



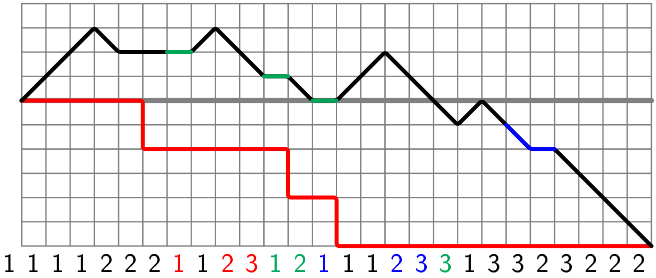
From right to left in the path, map a marked L to 1 and the next (left to right) two D steps to 2 and 3 (in order).



Recursively [Eu, 2010]: First step L, mark with 1. First step U (mark with 1), if the next non-U is D (mark with 2) or L (find next D and mark with 2 and 3 respectively). Repeat on unnumbered steps.



If the i-th element is numbered j, add the number i to the j-th row of the tableau.



1	2	3	4	8	9	12	14	15	16	20
	5	6	7	10	13	17	23	25	26	27
		11	18	19	21	22	24			

Steps:

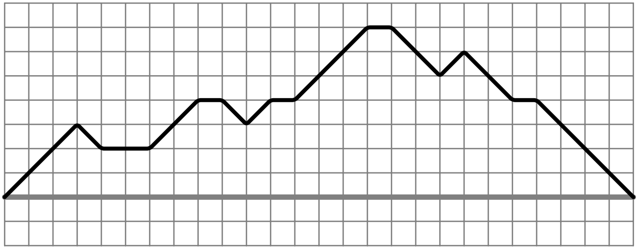
1. Where the pattern DL occurs, mark the L step.
2. Where the pattern UL occurs, transform it into DL.
3. From right to left in the path, for every marked L:
    1. If it is followed by a contiguous sequence of L, remove the marking and mark the rightmost L in this contiguous sequence.
    2. Then, map the marked L to 1 and the next (left to right) two D steps to 2 and 3 (in order).
4. Recursively [Eu, 2010]: First step L, mark with 1. First step U (mark with 1), if the next non-U is D (mark with 2) or L (find next D and mark with 2 and 3 respectively). Repeat on unnumbered steps.
5. If the i-th element is numbered j, add the number i to the j-th row of the tableau.

How do we reverse the procedure?

1 1 1 1 2 2 2 1 1 2 3 1 2 1 1 1 2 3 3 1 3 3 2 3 2 2 2

How do we reverse the procedure?

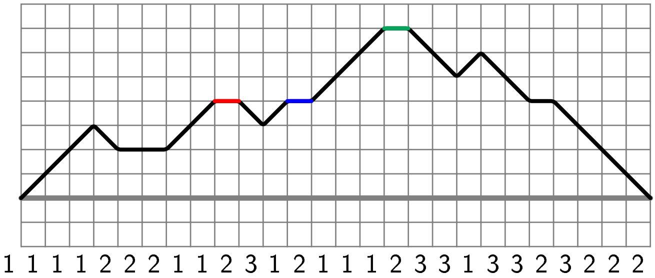
1 1 1 1 2 2 2 1 1 2 3 1 2 1 1 1 2 3 3 1 3 3 2 3 2 2 2



1 1 1 1 2 2 2 1 1 2 3 1 2 1 1 1 2 3 3 1 3 3 2 3 2 2 2

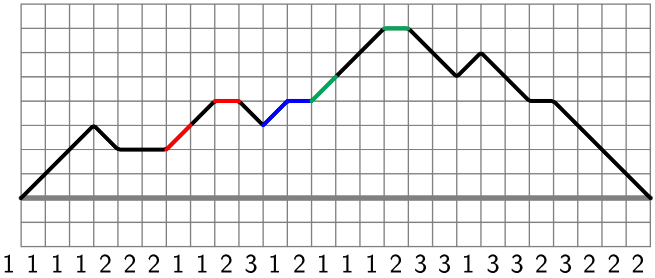
How do we reverse the procedure?

1 1 1 1 2 2 2 1 1 2 3 1 2 1 1 1 2 3 3 1 3 3 2 3 2 2 2



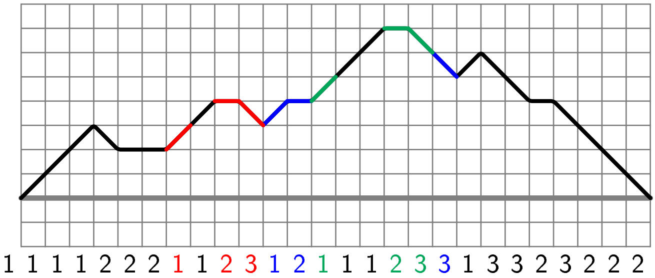
How do we reverse the procedure?

1 1 1 1 2 2 2 1 1 2 3 1 2 1 1 1 2 3 3 1 3 3 2 3 2 2 2



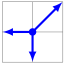
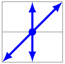
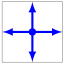
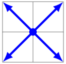
How do we reverse the procedure?

1 1 1 1 2 2 2 1 1 2 3 1 2 1 1 1 2 3 3 1 3 3 2 3 2 2 2



What about dynamic boundaries in the quarter plane?

Exhaustive searches carried out on

			
Kreweras	Gessel	Pólya	Diagonal

Some interesting relationships exist, e.g.:
Kreweras walks

- Start at $(0, 0)$, boundary is $y = 0$, $x = 0$, end on the current boundary.
- NE W shifts boundary by $(1, 2)$.
- W W shifts boundary by $(-2, -1)$.
- Boundary shifts just before final step of a pattern.
- Enumerated by

$$\frac{2^n}{2n+1}\binom{3n}{n}.$$

Future directions

- Find a "Proof from The Book" for the Kreweras enumeration! Perhaps by following some patterns...
- Given a stepset and a pattern, determine the nature of the generating function of the resulting pattern-avoiding walk. Even some small but reasonably general results in this direction...(small steps?)
- Develop automata-based methods for studying patterns in walks with two-way reversal bounded counter machines (D-finite). Assisting in classifying which pattern-avoiding walks remain D-finite?
- Find an example of a walk in $\mathbb{N}^2$ and a pattern for which some adapted kernel method would work. A highly symmetric walk?
- Pumping lemma-type results for D-finite languages?

Future directions

- Find a "Proof from The Book" for the Kreweras enumeration!
  Perhaps by following some patterns...
- Given a stepset and a pattern, determine the nature of the generating function of the resulting pattern-avoiding walk.
  Even some small but reasonably general results in this direction... (small steps?)
- Develop automata-based methods for studying patterns in walks with two-way reversal bounded counter machines (D-finite).
  Assisting in classifying which pattern-avoiding walks remain D-finite?
- Find an example of a walk in $\mathbb{N}^2$ and a pattern for which some adapted kernel method would work.
  A highly symmetric walk?
- Pumping lemma-type results for D-finite languages?

Thank you!


RESTAURANT
PARKING