Solvable self-avoiding walk and polygon models with large growth rates

Nicholas Beaton
Department of Mathematics and Statistics
University of Saskatchewan, Saskatoon

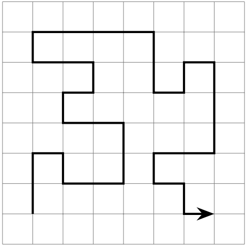
Axel Bacher
Laboratoire d'Informatique de Paris Nord
Université Paris 13

3 June 2015

5th Biennial Canadian Discrete and Algorithmic Mathematics Conference
University of Saskatchewan, Saskatoon

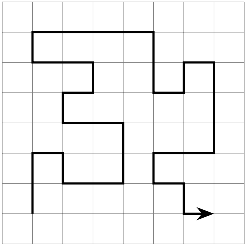
Introduction

Let c_n be the number of self-avoiding walks of length n on the square lattice $\mathbb{Z}^2$.



Introduction

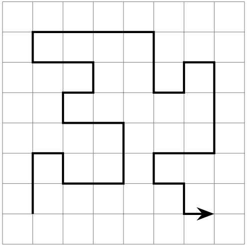
Let c_n be the number of self-avoiding walks of length n on the square lattice $\mathbb{Z}^2$.



c_n is known up to $n = 79$.

Introduction

Let c_n be the number of self-avoiding walks of length n on the square lattice $\mathbb{Z}^2$.



c_n is known up to $n = 79$.

A very hard problem!

Asymptotics

We still know some things about c_n. Because any SAW of length $m + n$ can be split into two smaller ones,

$$c_{m+n} \leq c_m c_n.$$

Asymptotics

We still know some things about c_n. Because any SAW of length $m + n$ can be split into two smaller ones,

$$c_{m+n} \leq c_m c_n.$$

Theorem (Hammersley 1957)

The limit

$$\lim_{n \to \infty} c_n^{1/n} = \mu$$

exists and is equal to $\inf_{n \geq 0} c_n^{1/n}$.

Asymptotics

We still know some things about c_n. Because any SAW of length $m+n$ can be split into two smaller ones,

$$c_{m+n} \leq c_m c_n.$$

Theorem (Hammersley 1957)

The limit

$$\lim_{n\to\infty} c_n^{1/n} = \mu$$

exists and is equal to $\inf_{n\geq 0} c_n^{1/n}$.

That is, the rate of growth of c_n is exponential:

Corollary

$$c_n = e^{o(n)}\mu^n.$$

Asymptotics

The constant μ is called the growth constant (sometimes connective constant), and depends on the lattice in question. On the square lattice,

$$\mu \approx 2.63815853031.$$

It is not known exactly for any regular lattice in ≥ 2 dimensions, except the honeycomb (hexagonal) lattice, where $\mu = \sqrt{2+\sqrt{2}}$ (Duminil-Copin & Smirnov 2012).

Asymptotics

The constant μ is called the growth constant (sometimes connective constant), and depends on the lattice in question. On the square lattice,

$$\mu \approx 2.63815853031.$$

It is not known exactly for any regular lattice in ≥ 2 dimensions, except the honeycomb (hexagonal) lattice, where $\mu = \sqrt{2+\sqrt{2}}$ (Duminil-Copin & Smirnov 2012).

In low dimension the subexponential term is not known exactly, but it is widely expected to follow a power law.

Conjecture

In 2 dimensions,

$$c_n \sim An^{\gamma-1}\mu^n$$

The constant A is the amplitude and depends on the lattice, while the exponent γ should only depend on the dimension. In $d = 2$ there is good reason to expect $\gamma = 43/32$, while for $d \geq 5$ it is known that $\gamma = 1$.

Behaviour of the generating function

Define the generating function

$$C(z) = \sum_{n \geq 0} c_n z^n.$$

Behaviour of the generating function

Define the generating function

$$C(z) = \sum_{n \geq 0} c_n z^n.$$

Then $z_c = \mu^{-1}$ is the radius of convergence of $C(z)$, and (in 2d) we should have

$$C(z) \sim A'(1 - \mu z)^{-43/32}$$

for $z \sim z_c$ and some constant A'.

Solvable subclasses

We don't have an expression for c_n or the generating function $C(z)$.

Solvable subclasses

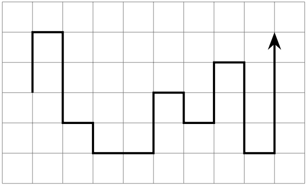
We don't have an expression for c_n or the generating function $C(z)$.

Instead, can look for subclasses of SAWs which are solvable. They may

- shed light on the overall SAW problem
- lead to physical models for which more precise information can be obtained
- lead to new techniques for enumeration, analysis, etc.
- be interesting in their own right!

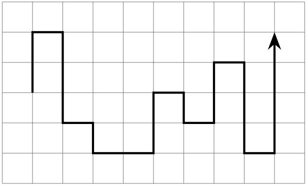
A simple example: partially directed walks

The simplest classes are obtained by forbidding one or more step directions. eg. a NES-partially directed walk can step $\uparrow, \rightarrow$ and $\downarrow$ but not $\leftarrow$.



A simple example: partially directed walks

The simplest classes are obtained by forbidding one or more step directions. eg. a NES-partially directed walk can step $\uparrow, \rightarrow$ and $\downarrow$ but not $\leftarrow$.

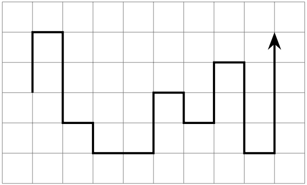


Easy to construct these recursively:

- either a walk has no $\rightarrow$ steps, and so is empty or just $\uparrow$ steps or just $\downarrow$ steps; or
- a walk has a last $\rightarrow$ step, and can be decomposed uniquely into a shorter walk concatenated with $\rightarrow$ and then a (possibly empty) sequence of $\uparrow$ or $\downarrow$.

A simple example: partially directed walks

The simplest classes are obtained by forbidding one or more step directions. eg. a NES-partially directed walk can step $\uparrow, \rightarrow$ and $\downarrow$ but not $\leftarrow$.



Easy to construct these recursively:

- either a walk has no $\rightarrow$ steps, and so is empty or just $\uparrow$ steps or just $\downarrow$ steps; or
- a walk has a last $\rightarrow$ step, and can be decomposed uniquely into a shorter walk concatenated with $\rightarrow$ and then a (possibly empty) sequence of $\uparrow$ or $\downarrow$.

This can be written as an equation involving the generating function $P(z)$:

$$P(z) = 1 + \frac{2z}{1-z} + z\left(1 + \frac{2z}{1-z}\right)P(z).$$

A simple example: partially directed walks

So

$$P(z) = \frac{1+z}{1-2z-z^2}$$
$$= 1 + 3z + 7z^2 + 17z^3 + 41z^4 + 99z^5 + 239z^6 + \dots$$

A simple example: partially directed walks

So

$$P(z) = \frac{1+z}{1-2z-z^2}$$
$$= 1 + 3z + 7z^2 + 17z^3 + 41z^4 + 99z^5 + 239z^6 + \dots$$

and the number c_n^{PDW} of PDWs of length n is

$$c_n^{\mathrm{PDW}} = \frac{\left(-2+\sqrt{2}\right)\left(1-\sqrt{2}\right)^n + \left(2+\sqrt{2}\right)\left(1+\sqrt{2}\right)^n}{2\sqrt{2}}$$
$$\sim \frac{1}{2}\left(1+\sqrt{2}\right)\left(1+\sqrt{2}\right)^n.$$

A simple example: partially directed walks

So

$$P(z) = \frac{1+z}{1-2z-z^2}$$
$$= 1 + 3z + 7z^2 + 17z^3 + 41z^4 + 99z^5 + 239z^6 + \dots$$

and the number c_n^{PDW} of PDWs of length n is

$$c_n^{\text{PDW}} = \frac{\left(-2+\sqrt{2}\right)\left(1-\sqrt{2}\right)^n + \left(2+\sqrt{2}\right)\left(1+\sqrt{2}\right)^n}{2\sqrt{2}}$$
$$\sim \frac{1}{2}\left(1+\sqrt{2}\right)\left(1+\sqrt{2}\right)^n.$$

So the growth rate here is $1+\sqrt{2} \approx 2.4142$.

A simple example: partially directed walks

So

$$P(z) = \frac{1+z}{1-2z-z^2}$$
$$= 1 + 3z + 7z^2 + 17z^3 + 41z^4 + 99z^5 + 239z^6 + \dots$$

and the number c_n^{PDW} of PDWs of length n is

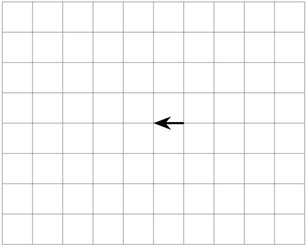
$$c_n^{\text{PDW}} = \frac{\left(-2+\sqrt{2}\right)\left(1-\sqrt{2}\right)^n + \left(2+\sqrt{2}\right)\left(1+\sqrt{2}\right)^n}{2\sqrt{2}}$$
$$\sim \frac{1}{2}\left(1+\sqrt{2}\right)\left(1+\sqrt{2}\right)^n.$$

So the growth rate here is $1+\sqrt{2} \approx 2.4142$.

Question: How close can we get to $\mu \approx 2.63815853031$ with a solvable model?

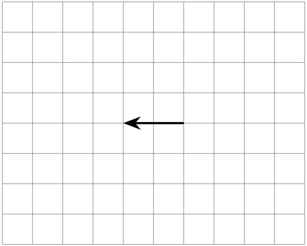
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



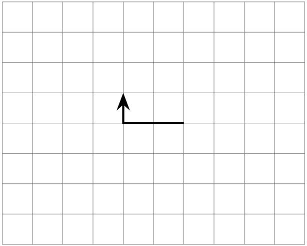
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



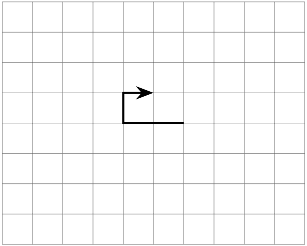
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



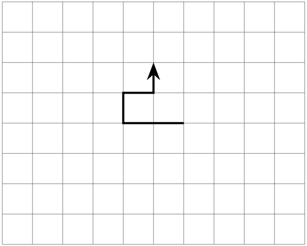
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



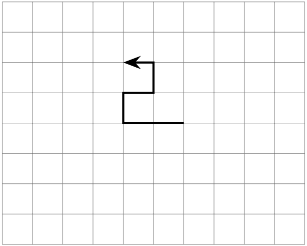
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



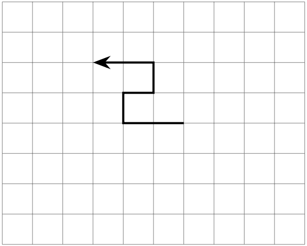
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



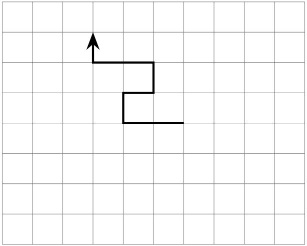
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



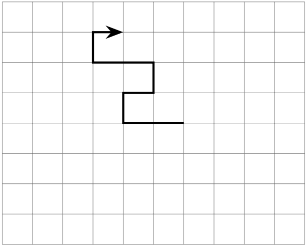
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



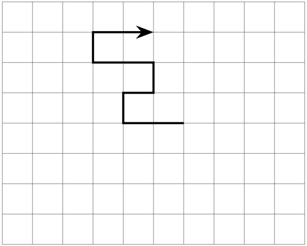
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



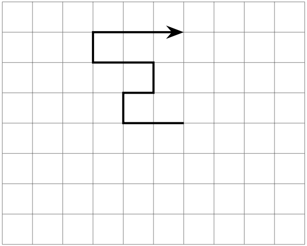
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



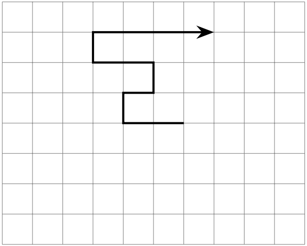
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



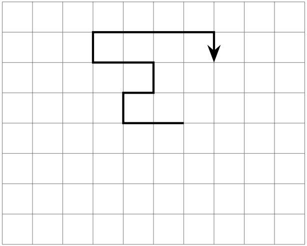
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



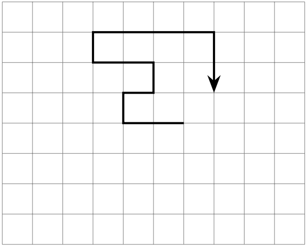
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



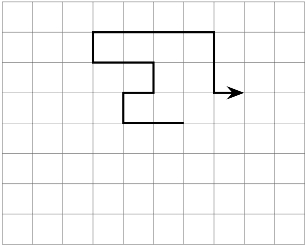
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



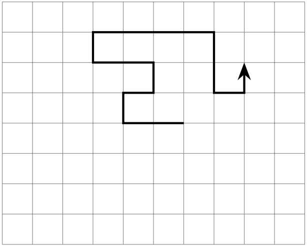
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



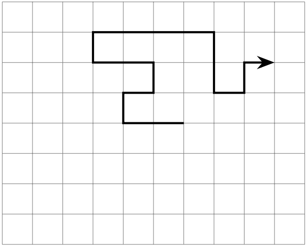
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



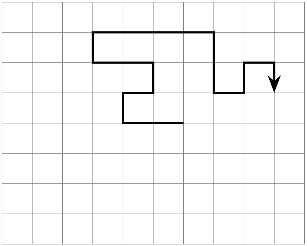
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



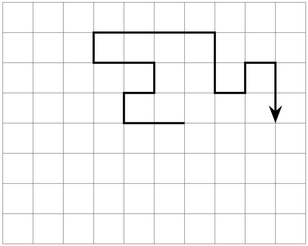
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



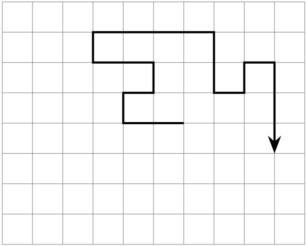
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



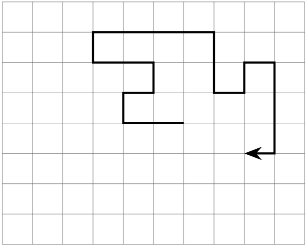
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



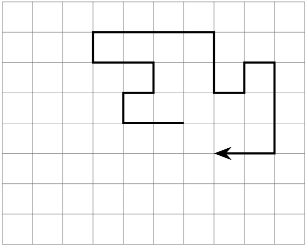
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



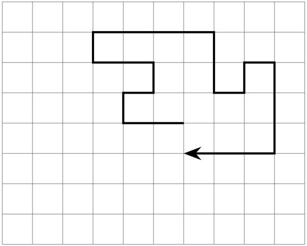
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



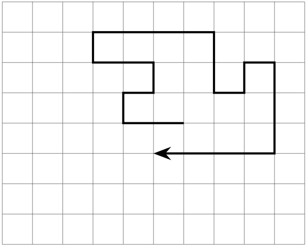
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



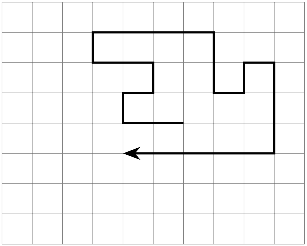
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



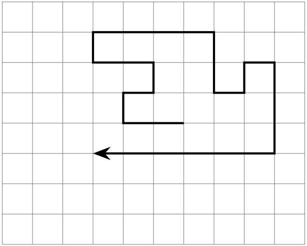
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



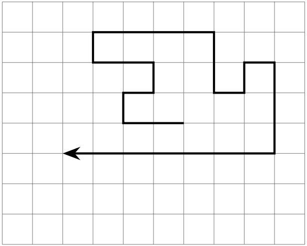
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



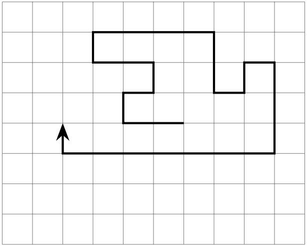
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



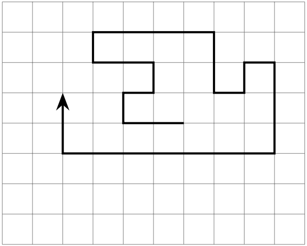
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



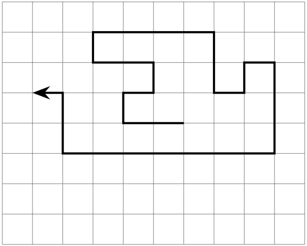
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



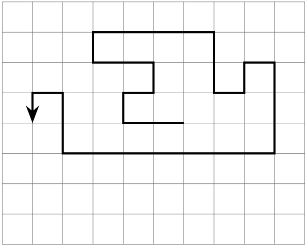
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



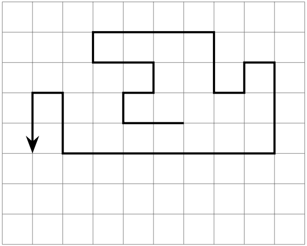
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



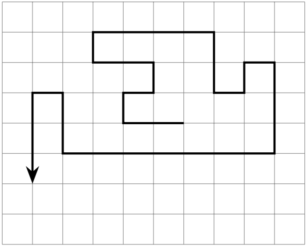
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



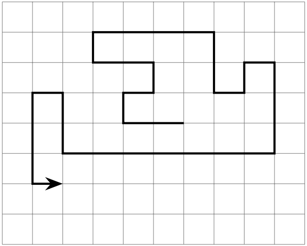
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



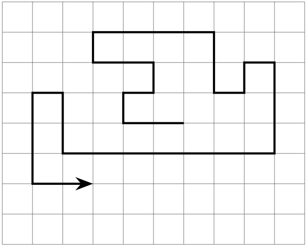
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



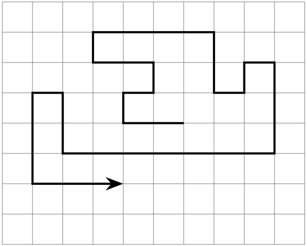
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



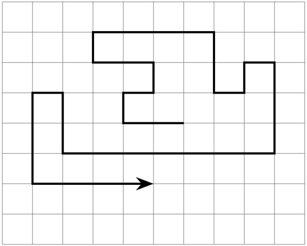
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



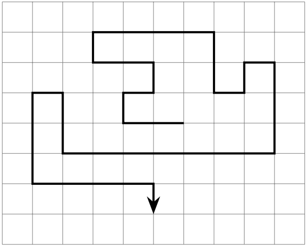
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



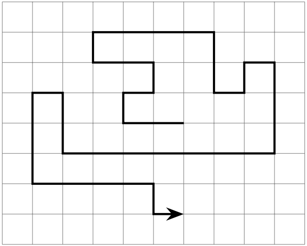
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



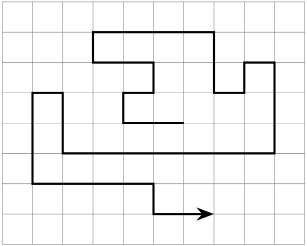
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



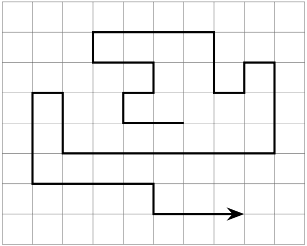
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



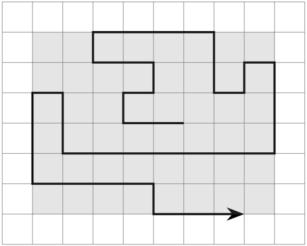
Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



Prudent walks

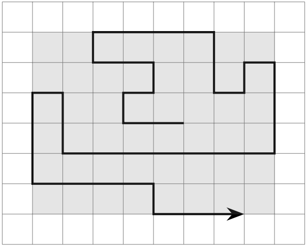
A prudent walk is a SAW which never steps towards an already-occupied vertex.



The end of a prudent walk always lies on the boundary of its bounding box, and this allows for a sub-classification:

Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.

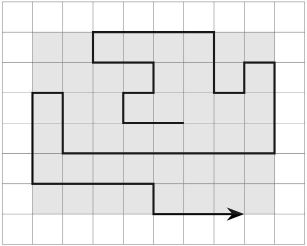


The end of a prudent walk always lies on the boundary of its bounding box, and this allows for a sub-classification:

- 1-sided: after each step, endpoint is on E side of box
- 2-sided: after each step, endpoint is on N or E sides
- 3-sided: after each step, endpoint is on N, E or W sides
- 4-sided: no restriction

Prudent walks

A prudent walk is a SAW which never steps towards an already-occupied vertex.



The end of a prudent walk always lies on the boundary of its bounding box, and this allows for a sub-classification:

- 1-sided: after each step, endpoint is on E side of box
- 2-sided: after each step, endpoint is on N or E sides
- 3-sided: after each step, endpoint is on N, E or W sides
- 4-sided: no restriction

1-sided prudent SAWs are also PDWs. 2- and 3-sided have been solved (Duchi 2005, Bousquet-Mélou 2010), 4-sided remains unsolved.

Prudent walks

Theorem (Duchi 2005, Bousquet-Mélou 2010)

The numbers of 2- and 3-sided prudent walks are asymptotically

$$c_n^{\text{2-pru}} \sim A_2 \kappa^n \qquad \text{and} \qquad c_n^{\text{3-pru}} \sim A_3 \kappa^n$$

where $\kappa \approx 2.48119$ *is the root of a cubic polynomial and* A_2, A_3 *are positive constants.*

Prudent walks

Theorem (Duchi 2005, Bousquet-Mélou 2010)

The numbers of 2- and 3-sided prudent walks are asymptotically

$$c_n^{2\text{-}pru} \sim A_2 \kappa^n \qquad \text{and} \qquad c_n^{3\text{-}pru} \sim A_3 \kappa^n$$

where $\kappa \approx 2.48119$ *is the root of a cubic polynomial and* A_2, A_3 *are positive constants.*

The generating function $C^{2\text{-pru}}(z)$ is algebraic (the root of a quadratic with coefficients in $\mathbb{Z}[z]$), and is solved with the kernel method.

The generating function $C^{3\text{-pru}}(z)$ is non-D-finite (cannot be written as the solution of a linear ODE with coefficients in $\mathbb{Z}[z]$), and is solved with the iterated kernel method.

Prudent walks

Theorem (Duchi 2005, Bousquet-Mélou 2010)

The numbers of 2- and 3-sided prudent walks are asymptotically

$$c_n^{\text{2-pru}} \sim A_2 \kappa^n \qquad \text{and} \qquad c_n^{\text{3-pru}} \sim A_3 \kappa^n$$

where $\kappa \approx 2.48119$ is the root of a cubic polynomial and A_2, A_3 are positive constants.

The generating function $C^{\text{2-pru}}(z)$ is algebraic (the root of a quadratic with coefficients in $\mathbb{Z}[z]$), and is solved with the kernel method.

The generating function $C^{\text{3-pru}}(z)$ is non-D-finite (cannot be written as the solution of a linear ODE with coefficients in $\mathbb{Z}[z]$), and is solved with the iterated kernel method.

Conjecture (Dethridge & Guttmann 2008)

The number of 4-sided prudent walks is asymptotically

$$c_n^{\text{4-pru}} \sim A_4 \kappa^n$$

for some positive constant A_4.

Perimeter walks

Can generalise prudent walks by maintaining the bounding box condition while relaxing the prudent condition, to get perimeter walks. Then 2-sided perimeter walks are solvable.

Theorem (B. 2012 (PhD Thesis))

The number of 2-sided perimeter walks is asymptotically

$$c_n^{\text{2-per}} \sim B_2 \tau^n$$

where $\tau \approx 2.50400$ *is probably not algebraic and* B_2 *is a positive constant.*

Perimeter walks

Can generalise prudent walks by maintaining the bounding box condition while relaxing the prudent condition, to get perimeter walks. Then 2-sided perimeter walks are solvable.

Theorem (B. 2012 (PhD Thesis))

The number of 2-sided perimeter walks is asymptotically

$$c_n^{\text{2-per}} \sim B_2 \tau^n$$

where $\tau \approx 2.50400$ *is probably not algebraic and* B_2 *is a positive constant.*

The generating function $C^{\text{2-per}}(z)$ is almost certainly non-D-finite.

Self-avoiding bridges

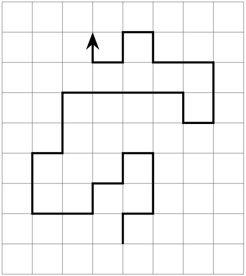
To take things further, we need another definition.

Self-avoiding bridges

To take things further, we need another definition.

A (2d) self-avoiding bridge is a SAW whose

- first vertex has strictly minimal y-coordinate
- final vertex has (not strictly) maximal y-coordinate

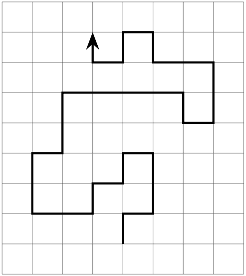


Self-avoiding bridges

To take things further, we need another definition.

A (2d) self-avoiding bridge is a SAW whose

- first vertex has strictly minimal y-coordinate
- final vertex has (not strictly) maximal y-coordinate



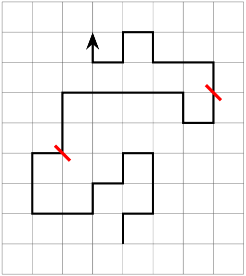
Any two bridges can be concatenated to form a longer bridge.

Self-avoiding bridges

To take things further, we need another definition.

A (2d) self-avoiding bridge is a SAW whose

- first vertex has strictly minimal y-coordinate
- final vertex has (not strictly) maximal y-coordinate



Any two bridges can be concatenated to form a longer bridge.

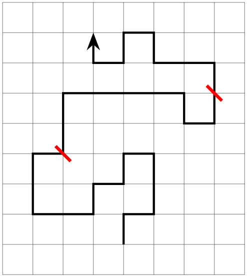
$\Rightarrow$ Any bridge can be uniquely decomposed into a sequence of irreducible bridges.

Self-avoiding bridges

To take things further, we need another definition.

A (2d) self-avoiding bridge is a SAW whose

- first vertex has strictly minimal y-coordinate
- final vertex has (not strictly) maximal y-coordinate



Any two bridges can be concatenated to form a longer bridge.

$\Rightarrow$ Any bridge can be uniquely decomposed into a sequence of irreducible bridges.

Note that we do not consider the empty walk to be a bridge.

Self-avoiding bridges

Since every bridge can be uniquely factorised as a sequence of irreducible bridges,

Self-avoiding bridges

Since every bridge can be uniquely factorised as a sequence of irreducible bridges,

$$B(z) = \frac{I(z)}{1 - I(z)}$$

where $B(z)$ is the generating function of bridges and $I(z)$ is the generating function of irreducible bridges.

Self-avoiding bridges

Since every bridge can be uniquely factorised as a sequence of irreducible bridges,

$$B(z) = \frac{I(z)}{1 - I(z)}$$

where $B(z)$ is the generating function of bridges and $I(z)$ is the generating function of irreducible bridges.

This idea can be exploited to get larger solvable classes: Define a class of walks (bridges) whose irreducible components satisfy some set of properties, which allow them to be solved.

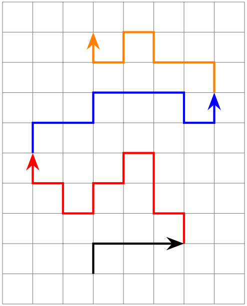
Weakly directed walks

A weakly directed walk is a SAW which is partially directed between any two visits to a horizontal line.

Weakly directed walks

A weakly directed walk is a SAW which is partially directed between any two visits to a horizontal line.

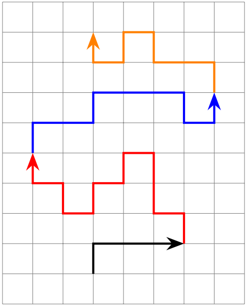
They are (essentially) self-avoiding bridges whose irreducible bridge components are partially directed.



Weakly directed walks

A weakly directed walk is a SAW which is partially directed between any two visits to a horizontal line.

They are (essentially) self-avoiding bridges whose irreducible bridge components are partially directed.



If $I^{\text{PDW}}(z)$ is the generating functions of irreducible partially directed bridges, then

$$B^{\text{WD}}(z) = \frac{I^{\text{PDW}}(z)}{1 - I^{\text{PDW}}(z)}.$$

Weakly directed walks

$I^{\mathrm{PDW}}(z)$ has been solved (Bacher & Bousquet-Mélou 2011).

Theorem (Bacher & Bousquet-Mélou 2011)

The number of weakly directed bridges is asymptotically

$$b_n^{WD} \sim C\sigma^n$$

where $\sigma \approx 2.5447$ is probably not algebraic. The generating function $B^{WD}(z)$ is non-D-finite.

Weakly directed walks

$I^{\mathrm{PDW}}(z)$ has been solved (Bacher & Bousquet-Mélou 2011).

Theorem (Bacher & Bousquet-Mélou 2011)

The number of weakly directed bridges is asymptotically

$$b_n^{WD} \sim C\sigma^n$$

where $\sigma \approx 2.5447$ *is probably not algebraic. The generating function* $B^{WD}(z)$ *is non-D-finite.*

Can we do better?

Weakly prudent bridges

An s-sided weakly prudent bridge is a bridge whose irreducible bridge components are s-sided prudent or co-prudent (prudent in the reverse direction) walks, or reflections/rotations thereof.

Weakly prudent bridges

An *s-sided weakly prudent bridge* is a bridge whose irreducible bridge components are *s*-sided prudent or co-prudent (prudent in the reverse direction) walks, or reflections/rotations thereof.

1-sided are weakly directed.

We solve the 2-sided case.

3-sided and 4-sided are the same thing, but remain unsolved.

Functional equations

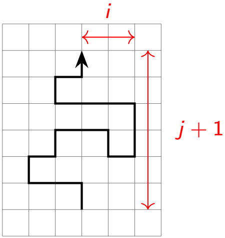
Prudent bridges are a special case of positive prudent walks: those whose first vertex has strictly minimal y-coordinate (upper half plane).

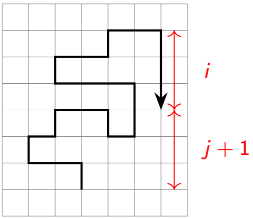
Functional equations

Prudent bridges are a special case of positive prudent walks: those whose first vertex has strictly minimal y-coordinate (upper half plane).

To construct positive prudent walks recursively, need two additional measurements:

- distance i from endpoint to NE corner of box
- distance $j+1$ from endpoint to bottom of box



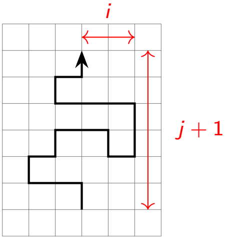


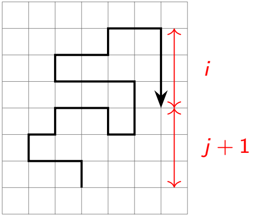
Functional equations

Prudent bridges are a special case of positive prudent walks: those whose first vertex has strictly minimal y-coordinate (upper half plane).

To construct positive prudent walks recursively, need two additional measurements:

- distance i from endpoint to NE corner of box
- distance $j+1$ from endpoint to bottom of box





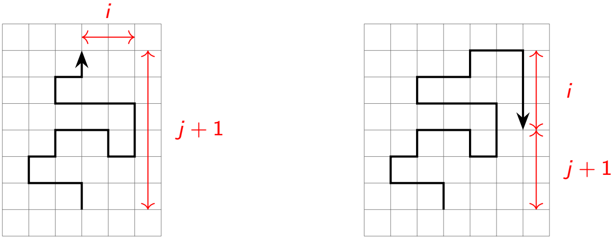
Define $N^{+}(z;u,v)$ and $E^{+}(z;u,v)$ to be the generating functions for those positive walks ending on the N or E side, with u conjugate to i and v conjugate to j. The variables u and v are catalytic.

Functional equations

Prudent bridges are a special case of positive prudent walks: those whose first vertex has strictly minimal y-coordinate (upper half plane).

To construct positive prudent walks recursively, need two additional measurements:

- distance i from endpoint to NE corner of box
- distance $j+1$ from endpoint to bottom of box



Define $N^+(z;u,v)$ and $E^+(z;u,v)$ to be the generating functions for those positive walks ending on the N or E side, with u conjugate to i and v conjugate to j. The variables u and v are catalytic.

The bridges are those counted by N^+.

Functional equations

Do something similar to PDWs (more complicated!) to get

$$\left(1 - \frac{zu}{u - zv} - \frac{z^2 u}{v - zu}\right) E^+(z;u,v) = z - \frac{z^2 v}{u - zv} E^+(z;zv,v) - \frac{z^2 u}{v - zu} E^+(z;u,zu) + zvN^+(z;z,v)$$

$$\left(1 - \frac{zuv}{u - z} - \frac{z^2 uv}{1 - zu}\right) N^+(z;u,v) = \frac{z}{1 - zu} - \frac{z^2 v}{u - z} N^+(z;z,v) + zE^+(z;zv,v)$$

Functional equations

Do something similar to PDWs (more complicated!) to get

$$\left(1 - \frac{zu}{u - zv} - \frac{z^2 u}{v - zu}\right) E^+(z; u, v) = z - \frac{z^2 v}{u - zv} E^+(z; zv, v) - \frac{z^2 u}{v - zu} E^+(z; u, zu) + zvN^+(z; z, v)$$

$$\left(1 - \frac{zuv}{u - z} - \frac{z^2 uv}{1 - zu}\right) N^+(z; u, v) = \frac{z}{1 - zu} - \frac{z^2 v}{u - z} N^+(z; z, v) + zE^+(z; zv, v)$$

This can be solved with the iterated kernel method. (Ugly!)

Functional equations

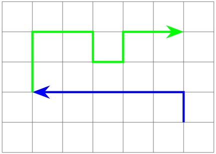
Do something similar to PDWs (more complicated!) to get

$$\left(1 - \frac{zu}{u - zv} - \frac{z^2 u}{v - zu}\right) E^+(z; u, v) = z - \frac{z^2 v}{u - zv} E^+(z; zv, v) - \frac{z^2 u}{v - zu} E^+(z; u, zu) + zvN^+(z; z, v)$$

$$\left(1 - \frac{zuv}{u - z} - \frac{z^2 uv}{1 - zu}\right) N^+(z; u, v) = \frac{z}{1 - zu} - \frac{z^2 v}{u - z} N^+(z; z, v) + zE^+(z; zv, v)$$

This can be solved with the iterated kernel method. (Ugly!)

The generating function of 2-sided prudent bridges is $N^+(z; 1, 1)$. We want to get at irreducible bridges. This is not so obvious – the irreducible components of a prudent bridge must be prudent, but if we concatenate two prudent bridges the result may not be prudent:



More functional equations

So we can't just directly factorize prudent bridges into irreducible prudent bridges.
Have to take a longer route.

More functional equations

So we can't just directly factorize prudent bridges into irreducible prudent bridges.
Have to take a longer route.

Have to consider the subset of bridges which end at the north-east corner of their box.

More functional equations

So we can't just directly factorize prudent bridges into irreducible prudent bridges.
Have to take a longer route.

Have to consider the subset of bridges which end at the north-east corner of their box.

Then, define a slightly different factorisation...

More functional equations

So we can't just directly factorize prudent bridges into irreducible prudent bridges. Have to take a longer route.

Have to consider the subset of bridges which end at the north-east corner of their box.

Then, define a slightly different factorisation...

Use the iterated kernel method on a slightly different functional equation...

More functional equations

So we can't just directly factorize prudent bridges into irreducible prudent bridges. Have to take a longer route.

Have to consider the subset of bridges which end at the north-east corner of their box.

Then, define a slightly different factorisation...

Use the iterated kernel method on a slightly different functional equation...

Then manipulate some generating functions...

...then combine with co-prudent, use inclusion-exclusion to account for those bridges which are both prudent and co-prudent (these are in fact partially directed). Likewise for reflections/rotations.

...then combine with co-prudent, use inclusion-exclusion to account for those bridges which are both prudent and co-prudent (these are in fact partially directed). Likewise for reflections/rotations.

The first few terms of the series for the irreducible objects are

$$I^{\text{2-WP}}(z) = z + 2z^2 + 2z^3 + 2z^4 + 2z^5 + 4z^6 + 10z^7 + 26z^8 + 56z^9 + 116z^{10} + O(z^{11})$$

...then combine with co-prudent, use inclusion-exclusion to account for those bridges which are both prudent and co-prudent (these are in fact partially directed). Likewise for reflections/rotations.

The first few terms of the series for the irreducible objects are

$$I^{\text{2-WP}}(z) = z + 2z^2 + 2z^3 + 2z^4 + 2z^5 + 4z^6 + 10z^7 + 26z^8 + 56z^9 + 116z^{10} + O(z^{11})$$

Our generating function is then

$$B^{\text{2-WP}}(z) = \frac{I^{\text{2-WP}}(z)}{1 - I^{\text{2-WP}}(z)}$$

Growth constant

Generating function is solved but complicated – easier to generate & analyse long series (6144 terms) using transfer matrix techniques.

Growth constant

Generating function is solved but complicated – easier to generate & analyse long series (6144 terms) using transfer matrix techniques.

Theorem (Bacher & B. 2014)

The number of 2-sided weakly prudent bridges is asymptotically

$$b_n^{2\text{-}WP} \sim D\phi^n$$

where $\phi \approx 2.57817$ *(known to 101 digits) is probably not algebraic.*

Growth constant

Generating function is solved but complicated – easier to generate & analyse long series (6144 terms) using transfer matrix techniques.

Theorem (Bacher & B. 2014)

The number of 2-sided weakly prudent bridges is asymptotically

$$b_n^{2\text{-}WP} \sim D\phi^n$$

where $\phi \approx 2.57817$ *(known to 101 digits) is probably not algebraic.*

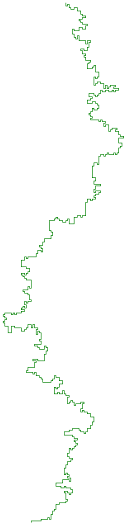
Similarly to 3-sided prudent and weakly directed:

Conjecture (Bacher & B. 2014)

The generating function $B^{2\text{-}WP}(z)$ *of weakly prudent bridges is not D-finite.*

Further results & future work

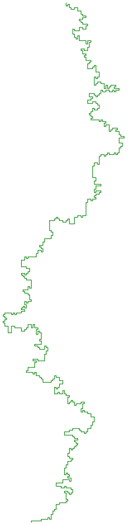
Can generate random weakly prudent bridges with a critical Boltzmann sampler.



Further results & future work

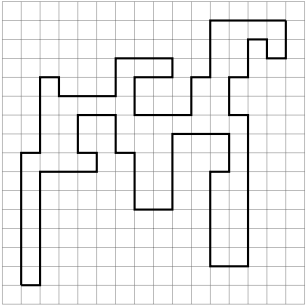
Can generate random weakly prudent bridges with a critical Boltzmann sampler.

Further generalisations? Solving walk models with more than 2 catalytic variables is often very difficult. (eg. the 3/4-sided prudent or 2-sided perimeter versions would require 3 catalytic variables).



Self-avoiding polygons

A (unrooted, undirected) self-avoiding polygon is a closed loop on the lattice.



It is known that these have the same growth rate as SAWs. If p_{2n} is the number of polygons of perimeter $2n$, then

Theorem (Hammersley 1961)

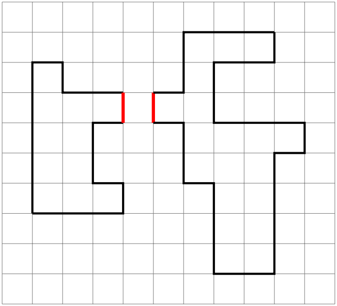
The limit

$$\lim_{n\to\infty} p_{2n}^{1/2n}$$

exists and is equal to μ*, the growth rate of SAWs on the lattice.*

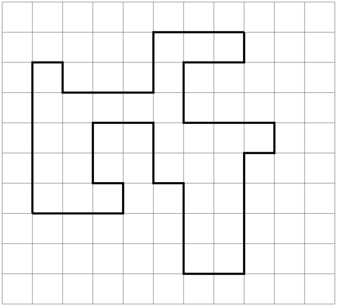
Concatenating polygons

Polygons can be freely concatenated, like self-avoiding bridges. We identify the highest edge on the right side of one polygon with the lowest edge on the left side of the other, and delete them both:



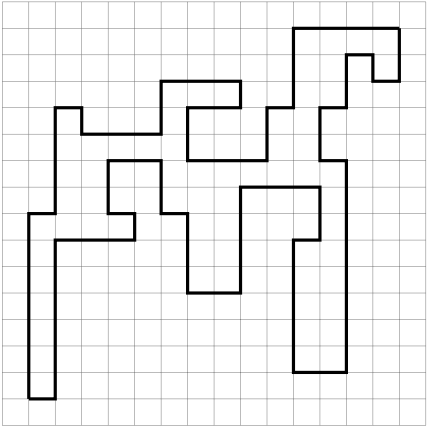
Concatenating polygons

Polygons can be freely concatenated, like self-avoiding bridges. We identify the highest edge on the right side of one polygon with the lowest edge on the left side of the other, and delete them both:



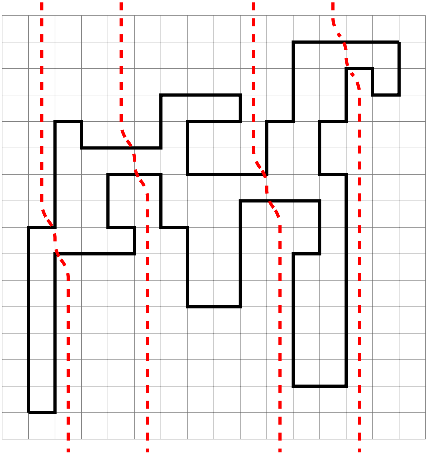
Irreducible polygons

Just as with bridges, we can then define an irreducible polygon to be one that cannot be written as the concatenation of two smaller polygons. Every polygon then has a unique factorization into irreducible components:



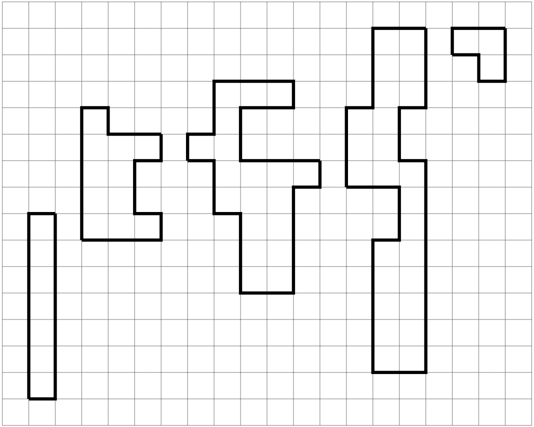
Irreducible polygons

Just as with bridges, we can then define an irreducible polygon to be one that cannot be written as the concatenation of two smaller polygons. Every polygon then has a unique factorization into irreducible components:



Irreducible polygons

Just as with bridges, we can then define an irreducible polygon to be one that cannot be written as the concatenation of two smaller polygons. Every polygon then has a unique factorization into irreducible components:



Irreducible polygons

To employ the same idea as with bridges, we then want to solve the largest possible class of irreducible polygons, and then concatenate them to generate something new.

Irreducible polygons

To employ the same idea as with bridges, we then want to solve the largest possible class of irreducible polygons, and then concatenate them to generate something new.

Note that the concatenation of two column-convex polygons is also column-convex. Most promising candidate is then column- and/or row-convex polygons, for which convexity is usually not preserved after concatenation.

Irreducible polygons

To employ the same idea as with bridges, we then want to solve the largest possible class of irreducible polygons, and then concatenate them to generate something new.

Note that the concatenation of two column-convex polygons is also column-convex. Most promising candidate is then column- and/or row-convex polygons, for which convexity is usually not preserved after concatenation.

Still a work in progress!

Reference

Bacher & B., *Weakly prudent self-avoiding bridges*, Proceedings of FPSAC 2014 (Chicago, USA), 827-838.

Reference

Bacher & B., *Weakly prudent self-avoiding bridges*, Proceedings of FPSAC 2014 (Chicago, USA), 827-838.

Thank you!