# Some problems in the counting of lattice animals, polyominoes, polygons and walks

**Andrew Daniel Rechnitzer**

Department of Mathematics and Statistics
The University of Melbourne
Parkville Victoria 3010
Australia

August 2000

Submitted in total fulfilment of the requirements
of the degree of Doctor of Philosophy

Printed on acid-free paper

## Abstract

The enumeration of polyominoes and lattice site and bond animals is a famous problem in enumerative combinatorics. General polyominoes and animals of arbitrary size have not been enumerated on any regular lattice in two or more dimensions; in fact after more than 40 years of intensive study there are still very few rigorous results. In light of the difficulty of these problems it is not unreasonable to ask what we may do in order to make some progress towards a solution. Three possible options we explore in this thesis are:

- analyse the problem numerically,
- determine properties of the solution without actually obtaining the solution, and
- solve similar problems.

Roughly speaking this thesis is divided into five problems falling across these three areas.

- We develop a method for the investigation of certain properties of anisotropic generating functions of families of bond animals on the square lattice, including many important unsolved models. It has been suggested (on the basis of numerical evidence) that some of these properties are intimately linked with the *solvability* of a model. In the case of *self-avoiding polygons* we are able to use this method and refine its results to prove certain facts about the analytic structure of the generating function. These imply that the anisotropic generating function of *self-avoiding polygons* is not *differentiably-finite*.
- The most elegant technique (to date) for the enumeration of directed animals is based on a correspondence between these animals and certain heaps of dimers, called pyramids. By extending this correspondence we are able to define and solve several new classes of polyominoes on the square and triangular lattices that are exponentially more numerous than any previously solved.
- The study of polyominoes has mainly focused on their enumeration according to their most basic geometric properties — area and perimeter — and a large number of models have been enumerated according to both of these parameters. By contrast there are very few results concerning the *site-perimeter* of polyominoes; site-perimeter is of considerable interest to physicists since it plays an important role in the study of percolation models. Using a variation of the column-by-column construction known as the "Temperley method", we solve bargraph polygons according to their site-perimeter; all previous site-perimeter results concern families of convex polygons.
- We investigate functional symmetries, known as "*reciprocity*" and "*inversion relations*", in the generating functions of a number of families of polygons, including several families of column-convex polygons, three-choice polygons and staircase polygons

with a staircase hole. In so doing, we establish a connection between the reciprocity results known to combinatorialists and the inversion relations used by physicists to solve models in statistical mechanics. For several classes of convex polygons, the inversion (reciprocity) relation, in conjunction with certain symmetry and analyticity properties, completely determines the anisotropic perimeter generating function.

- We investigate restricted self-avoiding walk models in three dimensions; in particular we examine the effect that small changes to the model have on its asymptotic behaviour. The two-dimensional case has been well studied and previously it has been argued that the *universality class* (*i.e.* the dominant asymptotic behaviour) of the restricted walk model is determined primarily by the symmetries of the restricting rule. In order to search for a connection between the symmetries of the rules and possible new universality classes in three-dimensions, we have chosen a number of different restricted self-avoiding walk models and analysed them using computer enumeration and series analysis. We argue that no such link exists in three dimensions and we conjecture that there exists only one important universality class.

# Declaration

This is to certify that:

1. *the thesis comprises only my original work except where indicated in the preface,*
2. *due acknowledgement has been made in the text to all other material used,*
3. *the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.*

Andrew Rechnitzer

# Preface

This thesis was written under the supervision of


- Professor Anthony J. Guttmann and Dr. Aleksander L. Owczarek
  Department of Mathematics and Statistics
  The University of Melbourne
  Victoria 3010, Australia.
  `tonyg@ms.unimelb.edu.au, aleks@ms.unimelb.edu.au`

- Dr. Mireille Bousquet-Mélou
  CNRS, LaBRI, Université Bordeaux 1
  351 cours de la Libération
  33405 Talence Cedex, France.
  `bousquet@labri.u-bordeaux.fr`


## Chapter 3

- Work done by Andrew Rechnitzer (AR). We thank M. Bousquet-Mélou (MBM), A.J. Guttmann (AJG) and A.L. Owczarek (ALO) for their valuable discussions.

- AR thanks I. Jensen (IJ) and AJG for providing their enumeration data.

- Much of the work in this chapter has been accepted as a communication to be given at the LaCIM 2000 conference in Montreal.

- A paper based on this chapter and chapter 7 is in preparation [137].

## Chapter 4

- Most of this chapter is original work done equally by AR and MBM. We thank J.-P. Allouche for his valuable discussions.

- Some of the work was presented at the FPSAC 1999 conference in Barcelona [31].

- Sections 4.2.1 and 4.2.2 review the theory of heaps of dimers due to Viennot [156] and its application to the enumeration of directed animals.

- Section 4.4 is the work of MBM.

- The contents of this chapter are currently being written as a paper by AR and MBM [29].

## Chapter 5

- Most of this chapter is a review of known techniques.
- The interpretation of the "integral Temperley" method in terms of Hadamard products and its extension to the "Hadamard-Temperley" method is due to AR.

## Chapter 6

- Work done by equally AR and MBM. We thank ALO for his useful discussions about phase transitions and polymers.
- The contents of this chapter are currently being written as a paper with MBM [30].

## Chapter 7

- Work done by AR. We thank MBM and AJG for their useful discussions.
- AR thanks IJ and AJG for providing their enumeration data.
- Some of the work of this chapter has been accepted by the LaCIM 2000 conference.
- Section 7.4 is a review of classes of formal power series.
- Theorem 7.19 is due to MBM.
- A manuscript based on this chapter and chapter 3 is in preparation [137].

## Chapter 8

- Work done by AR, MBM, AJG and W. P. Orrick (WPO).
- The work in this chapter has appeared as a paper [28].
- Section 8.4 is mostly the work of MBM.
- Section 8.5 is mostly the work of WPO.

## Chapter 9

- Work done equally by AR and ALO.
- The work in this chapter has appeared as a paper [138].
- Enumeration data generated by AR, and analysed by AR under the guidance of ALO.

## Acknowledgements

- **for supervision above and beyond the call of duty:** Aleks, Mireille and Tony.
- **for coffee drinking even in the face of impending kidney damage:** Murray, Tommy, Nick, Anthony, Mark, Henry, Trond, Will, Lois and Kate.
- **for standing by and watching me butcher the French language:** Cedric, Manu, Philipe, Gilles, Olivier, Driss and Akka.
- **for stopping me going insane in the last few weeks of the thesis:** Aleks and Rachel.
- **for too much overkill:** Peetee.
- **all things geeky and technical:** Russell, Chris, Rafal, Rhys, Tibor, Nick and Mark.
- **administrative excellence:** Yolanda (the fixer) and Averil.
- **DNA:** my parents.
- **cats:** Sheera, Krista and Kitty.
- **patience:** Rachel.
- **for having a very fine-toothed comb:** my thesis examiners.

# Contents

# List of Figures

# List of Tables

# CHAPTER 1

## Introduction

## 1.1

# Placing tiles

Consider an infinite square grid[1], and some unit square tiles.

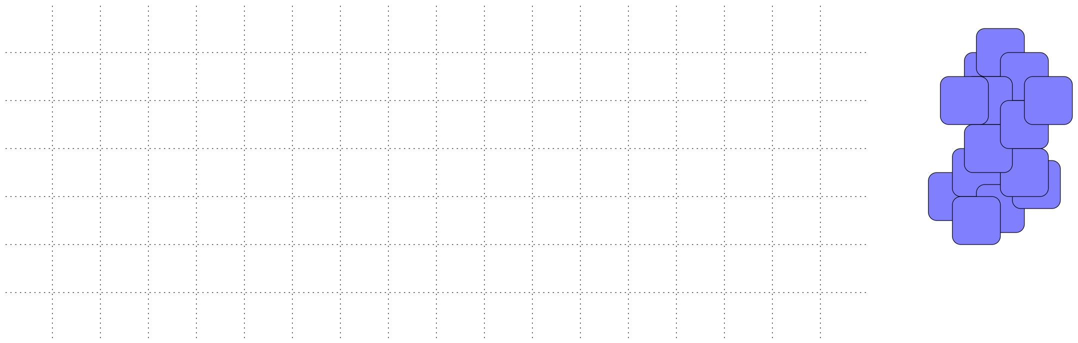


Figure 1.1: A portion of the infinite square grid, and some square tiles.

This thesis (and a good number of other theses and articles) is motivated by the simple question:

**Question 1.** ***How many ways can we place a finite number of tiles on the square grid?***

Rather than leaping in at the deep end, let us start with a simpler question:

**Question 2.** ***How many ways can we place a single tile on the square grid?***

Figure 1.2: A placement of a single tile on the square grid.

[1]Due to limitations of space, we cannot show all of it here.

Since there are an infinite number of cells on the square grid, and a single tile can be placed on any of them, there are an infinite number of ways we can place a single tile. All of these different ways, however, are equivalent under translation — there is only a single way to colour a single cell (up to translation). In the same way, both of the arrangements in

Figure 1.3: These two arrangements of five tiles are equivalent up to translation.

figure 1.3 are equivalent under some translation. In light of this, let us modify question 1:

**Question 3.** ***How many ways (up to translation) can we place a finite number of tiles on the square grid?***

We know that there is only a single way to place a single tile (up to translation). What about placing two tiles? Consider the three arrangements in figure 1.4. The tiles in the first two arrangements (left and centre) touch, while in the last the tiles are separated by at least a single lattice spacing. It is not difficult to see that there are only two *connected* arrangements of two tiles. Let us define *connectedness* a little more carefully:

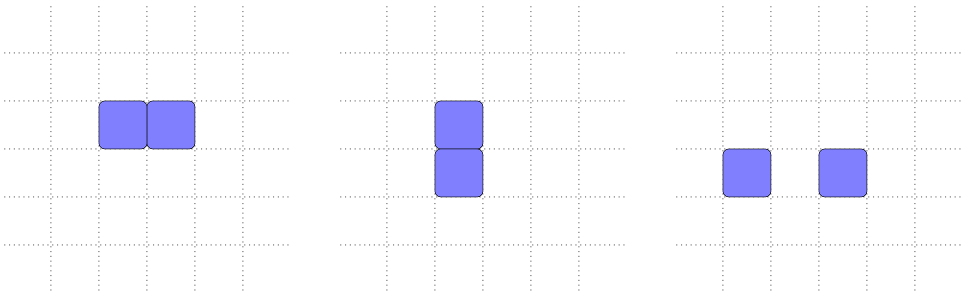


Figure 1.4: The first two arrangements are the only possible connected arrangements of two tiles. Disconnected arrangements pose problems — there are an infinite number of disconnected arrangements of two tiles.

**Definition 1.1.**
We say that an arrangement of tiles, $P$, on the square grid is *connected* if for any two tiles in $P$ there exists a path (made up of unit steps north, east, south or west) from one tile to the other, that stays completely within $P$ (see figure 1.5).

If an arrangement of $n$ tiles is connected, then we call it an $n$-celled ***polyomino***, or a ***polyomino*** of area $n$. Unless explicitly stated otherwise, we will consider polyominoes defined up to translation (*i.e.* if polyomino #1 can be mapped to polyomino #2 under some translation, then we will consider them to be the same).

The name, *polyomino*, was introduced by Golomb [74], and is the generalisation of *dominoes*.

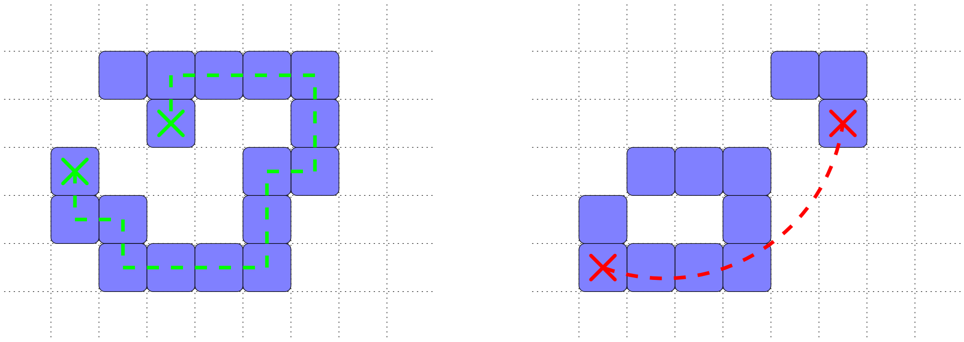


Figure 1.5: The arrangement on the left is connected, while that on the right is not.

**Lemma 1.1.** *For finite $n$, there is always a finite number of polyominoes of area $n$. More specifically, there cannot be more than $\binom{n^2}{n}$ polyominoes of area $n$.*

*Proof.* A polyomino of area $n$ must always be contained within a square "box" of side-length $n$; this box contains $n^2$ cells. Every polyomino is a choice of $n$ cells from within this box and so there are at most $\binom{n^2}{n}$ polyominoes of area $n$. ∎

On the other hand, we can easily create an infinite number of *disconnected* arrangements of two tiles — take the single arrangement of a single tile, and then place another tile $n$ cells to its right, where $n$ is some integer strictly greater than 1. Using similar reasoning, it is not hard to see that there will always be an infinite number of disconnected arrangements of $n$ tiles (for finite $n \geq 2$) .

Since there are always a finite number (up to translation) of *polyominoes* of area $n$, and an infinite number of disconnected arrangements of $n$ tiles, we modify question 3:

**Question 4.** ***How many different polyominoes (up to translation) of area $n$ can be constructed on the square grid?***

Before we consider what is known about the number of polyominoes, let us first consider what we would accept as a solution to question 4.

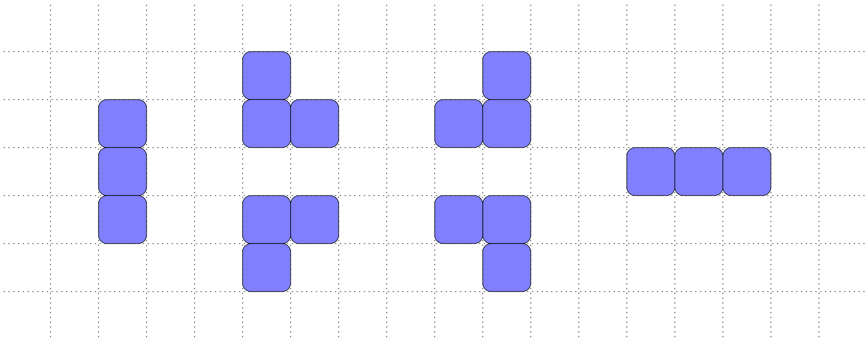


Figure 1.6: The 6 possible polyominoes of area 3.

**Question 5.** ***What constitutes a solution to question 4?***

Rather than delving into a deep discussion of computational complexity and mathematical philosophy, we will attempt to answer this question in a heuristic[2] way.

Any answer to question 4 must be a two step process of the form:

**1** ⇾ choose some $n \in \mathbb{N}$.

**2** ⇾ put this $n$ into some sort of *machinery*[3] and after some *finite* amount of time it returns the number, $c_n$, of polyominoes of area $n$.

This "definition" of a solution clearly allows a very wide range of possibilities — ideally we would like to find the "best" possible solution, however this is *really* hard to define. It is much easier to start with the "worst" type of solution. Perhaps the "worst" way[4] in which we can compute $c_n$ is to use brute force to list all of the polyominoes with $n$ cells; $c_n$ is then the length of the list[5]. For many problems no better method is forthcoming and this "worst" method is in fact the best available.

Any method that is "faster" than this "worst" approach (particularly when $n$ is a large number) is, in some weak way, a solution; combinatorialists consider a *nice* solution to be one of the following:

- a closed-form expression for $c_n$ that is "easy to evaluate",
- a closed-form expression for the generating function,

$$f(q) = \sum c_n q^n \qquad \text{or} \qquad e(q) = \sum c_n \frac{q^n}{n!}$$

  that is "easy to evaluate"[6], or

[2] *i.e.* hand-waving.

[3] It could be a mathematical, virtual, quantum... it really doesn't matter.

[4] Other than giving up.

[5] Any method that is "slower" than this really is worthless.

[6] Since $c_n = \frac{1}{2\pi i} \oint \frac{f(q)}{q^{n+1}} \mathrm{d}q$, we can recover $c_n$ from the expression of $f(q)$ or $e(q)$.

- an algorithm or recurrence that computes $c_n$ "efficiently" (and may be equivalent to a closed-form expression for $c_n$ or the generating function).

By an expression being "easy to evaluate" we mean that evaluating the expression should be substantially faster than the brute force method — if we wish to compute all the elements in a set $S$, then a solution of the form

$$|S| = \sum_{x \in S} 1,$$

is equivalent to brute force and so is almost completely worthless.

Another slightly different way of "solving" a polyomino problem (which we do not explore in this thesis) is to somehow map the original problem to a new problem that can be solved. A solution to the new problem may not give us a means of calculating $c_n$, but it may give us exact values of other quantities of interest, such as growth constants and critical exponents (see section 1.2.2 below).

As an example of the different forms of a solution, let us consider the simpler question of permutations:

**Question 6.** ***What is the number,*** $\mathrm{s_n}$***, of ways of ordering*** $\mathrm{n}$ ***objects?***

The "worst" solution to this problem of computing $s_n$ would be to list all the possible permutations using a computer program (or, heaven forbid, by hand). A better solution would be to show that the numbers $s_n$ satisfy the recurrence

$$\begin{aligned} s_0 &= 1, \\ s_n &= n s_{n-1} \qquad \text{for } n \geq 1. \end{aligned} \tag{1.1}$$

Solving this recurrence gives a closed-form expression for $s_n$ that is very easy to evaluate:

$$s_n = n! \tag{1.2}$$

Alternatively we could solve the recurrence for the (exponential) generating function,

$$\sum_{n \geq 0} s_n \frac{q^n}{n!} = \sum_{n \geq 0} q^n = \frac{1}{1-q}. \tag{1.3}$$

Any solution to question 4 that took one of these three "compact" and "efficient" forms (*i.e.* a closed form expression for the coefficients or the generating function, or an efficient algorithm) would be most welcome. In this thesis we will almost exclusively consider solutions in terms of generating functions.

**Question 7.** ***What do we know about the number of polyominoes?***

In one dimension, it is easy to see that there is only a single $n$-celled polyomino for any given $n$. As soon as one ventures into two or more dimensions, however, the answer is not so simple; it may come as a surprise that the answer to question 4 remains elusive even after more than 40 years of intensive study [75, 104, 151]. Arguably the strongest *rigorous* result on the number of polyominoes concerns their asymptotics. Let $c_n$ denote the number of polyominoes of area $n$ on the square lattice. A concatenation argument [105] shows that

$$c_{n+m} \geq c_n \, c_m \qquad\qquad \forall n, m \in \mathbb{N}.$$

Combining this with a better upper bound than that given in lemma 1.1 implies that there exists a constant $\mu$, sometimes called *Klarner's constant*, or more generally *growth constant* and *connective constant*, such that

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

The exact value of $\mu$ is unknown, though numerical studies [44, 99] have shown that $\mu \simeq 4.06$. The best published bounds[7] for $\mu$ [99, 107] are

$$3.9 < \mu < 4.65.$$

It is a measure of the complexity of polyomino enumeration that after more than *forty years*, the rigorous bounds on $\mu$ are so wide (we only know it to within around 20% of the value predicted by numerical work).

The existence of $\mu$ tells us that the number, $c_n$, of polyominoes of area $n$ grows exponentially with $n$; for any $\bar{\mu} < \mu$ we have (for sufficiently large $n$) $c_n > \bar{\mu}^n$. *There are lots of them!* It also allows us be a little more specific about what we would require of a good solution to question 4.

Since the number of polyominoes grows exponentially, the time taken by a brute-force approach must also grow exponentially (time $= O(\mu^n)$). Consequently, any method that evaluates $c_n$ in either bounded time (time $= O(1)$) or polynomial time (time $= O(n^\alpha)$, for some $\alpha \in \mathbb{R}^+$) would be a very good answer to question 4. Even an exponential time algorithm (time $= O(\lambda^n)$, for some $1 < \lambda < \mu$) would be, in some weak sense, a solution. The best "solution" to date (in two dimensions) is an enumeration algorithm based on the finite lattice method [44] — this still requires exponential time and *space*[8], but *is* exponentially faster than brute force enumeration.

Since the enumeration of polyominoes (and related objects) in one dimension is trivial, in this thesis we mostly consider enumeration problems in two dimensions (on the square, triangular and honeycomb lattices).

[7]This topic is evolving rapidly; see Steve Finch's web page on mathematical constants for up-to-date information ( `http://www.mathsoft.com/asolve/constant/constant.html` )

[8]*i.e.* the amount of computer memory required to compute $c_n$ also grows exponentially with $n$.

# 1.2 If counting polyominoes is hard, what else can we do?

Ideally[9] we would like to be able to write in this introduction something along the lines of:

"*In chapter k we give our polynomial time algorithm for the computation of the number of polyominoes of area n.*"

The history of polyomino enumeration on the square lattice would suggest that our attempts at finding a solution are likely to be frustrated[10], and indeed a quick look at the table of contents will tell you that we cannot make any such claim (without lying).

**Question 8.** ***If we cannot solve the problem then what can we do?***

Three possible options are:

- analyse the problem numerically,
- determine properties of the solution without actually obtaining the solution, and
- solve similar problems.

Roughly speaking, this thesis is divided into three different areas, each exploring one of these possibilities.

## 1.2.1 Numerical analysis

Many problems are too difficult to be tackled analytically, and often the most appropriate avenue to take is some sort of numerical analysis or random simulation. Even if we are unable to solve a model, it is frequently possible to write a reasonably fast computer algorithm to provide us with the first $N$ coefficients of the generating function, $c_1, c_2, \dots, c_N$, either exactly[11] or to a good approximation[12].

It is sometimes possible, from the first $N$ terms (if they are exact), to somehow "see" the pattern in the coefficients and hypothesise a solution (which can then be proved by other means). More usual is to use the data to explore the asymptotic behaviour of the model. One can make some assumption about the asymptotic form of the coefficients such as

$$c_n \sim A\mu^n n^\gamma \qquad \text{as } n \to \infty,$$

[9] In a perfect world where our brains are larger... well even this may not be enough; no-one has proved that a good solution exists.

[10] This is an understatement of the extreme difficulty that has been encountered by those venturing into this area of enumerative combinatorics.

[11] In which case $N$ is probably a small integer, probably well under 100.

[12] In which case $N$ is probably much larger, but the error bounds on the estimates of the coefficients also become larger and larger with $N$.

and then fit the data (exact or approximate) to this form and so obtain estimates of various quantities such as the growth constant or the mean geometric size[13] of the objects. Indeed for many purposes the asymptotic behaviour of a model is arguably more interesting than the existence of an exact solution (particularly if we are modelling some sort of physical problem). Of course, an exact solution is more useful than a knowledge of the asymptotic behaviour, since one can (usually) extract the asymptotic behaviour from a solution but not *vice-versa*[14].

### 1.2.2 Properties of the solution

In many cases it is possible to determine various properties of the solution to the problem we are interested in without actually obtaining that solution. Asymptotic behaviour is one of the most important. For example, the existence of the growth constant, $\mu$, tells us that the coefficients, $c_n$ must be of the form

$$c_n = \mu^n \theta(n), \tag{1.4}$$

for some function $\theta(n)$ that satisfies

$$\lim_{n\to\infty} \theta(n)^{1/n} = 1.$$

It is believed (but not yet proved) that $c_n$ should actually behave like

$$c_n \sim A\mu^n n^{\gamma-1} \qquad \text{as } n \to \infty, \tag{1.5}$$

where $\gamma$ is called a *critical exponent*. This asymptotic form is expected to hold for polyominoes and a great number of other models.

In the case of the self-avoiding walk (SAW)[121] the exact value of the exponent $\gamma$ is known in two dimensions (but not rigorously) because of a connection, first observed by de Gennes [47], with a magnet model called the $N$-vector model. In the limit $N \to 0$ this magnet model (in $d$-dimensions) reduces to the self-avoiding walk (in $d$-dimensions). Nienhuis [125] was able to calculate the exponents of the $N$-vector model in two dimensions for general $N$, and found that $\gamma = \frac{43}{32}$ (when $N = 0$). This same technique also gives the SAW length exponent $\nu = 3/4$ (see below), and the growth constant on the honeycomb lattice as $\mu = \sqrt{2+\sqrt{2}}$.

Some knowledge of the analytic structure of the generating function may also be of use; there exist many techniques for "discovering" solutions from the first few terms of the generating function (such as NEWGRQD [80], GFUN and MGFUN [92]). These techniques rely on the solution satisfying simple differential or algebraic equations. If we can show that solution has certain properties that mean it cannot satisfy such an equation then we will not be able to "guess" the solution using these techniques, no matter how many terms we obtain.

[13] *i.e.* on average how wide are they? And how does this width change with $n$?

[14] Extracting asymptotic behaviour from a solution is not necessarily trivial — indeed for the problems described in chapter 6, we expect the asymptotics to be rather simple, but the form of the solution makes *proving* this rather difficult.

### 1.2.3 Solve similar problems

Perhaps the most obvious thing that we can do when presented with a hard problem is to first try to solve a similar, but simpler problem. By solving simpler models we develop techniques and ideas that help us understand the original problem. In the study of polyominoes (arguably) the most successful line of research has been the enumeration of simpler subsets of polyominoes. By enforcing additional restrictions, such as directedness or convexity, the set of polyominoes can be reduced until it is solvable. Many of these similar problems are interesting in their own right. Consider, for example, *directed* polyominoes (we will describe them in detail in chapter 4). Directed polyominoes must contain a single tile called the *root* or *source* from which all the other tiles can be reached by paths taking only north and east steps (see figure 1.7); this particular model has been solved. *Directedness* arises naturally in models which are subject to strong forces such as gravity, and can also be used to model time dependence (since objects cannot move backwards in time). Dhar [54, 55] showed that certain directed polyomino models are connected to certain gas models. *i.e.* it is important for reasons other than that it lets us solve the model — it is not just a consolation prize.

In the next section we list some variations of the basic polyomino problem that have been studied. It should be noted that subclasses of general polyominoes (certainly all the solved subclasses) have *smaller* growth constants (directed polyominoes have a growth constant $\mu = 3$). Part of the challenge of enumerating subclasses is to enumerate *large* subclasses with growth constants close to those of general polyominoes.

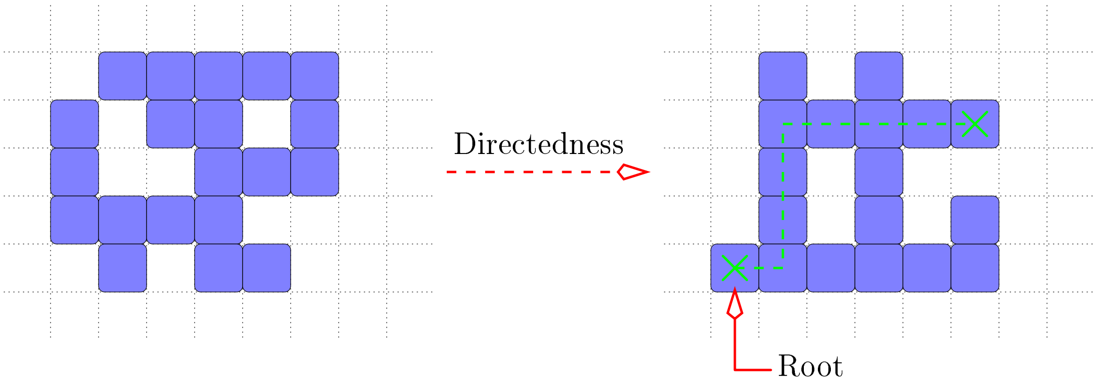



Figure 1.7: If there exists a single *root* cell, from which all the cells of the polyomino can be reached by a directed path, then the polyomino is a *directed polyomino*.

Let us now examine some of these objects related to polyominoes.

## 1.3

# A taxonomy of the animal kingdom

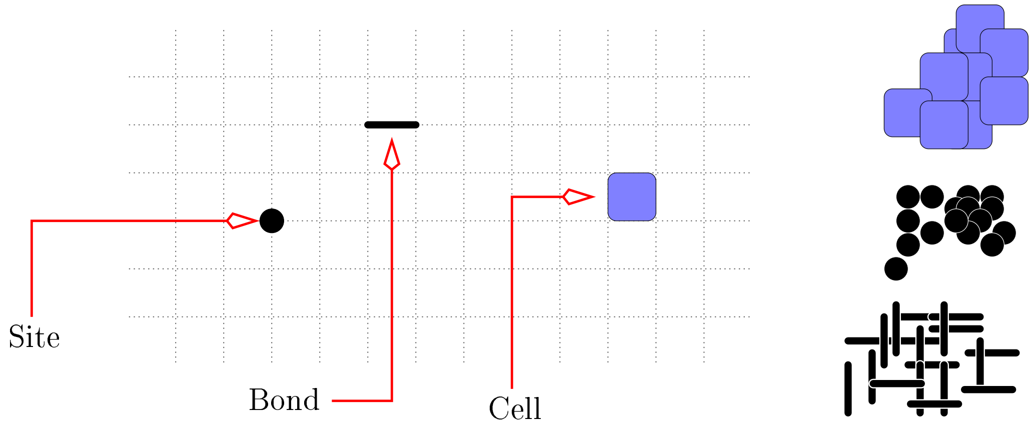



Figure 1.8: Basic bits of grids

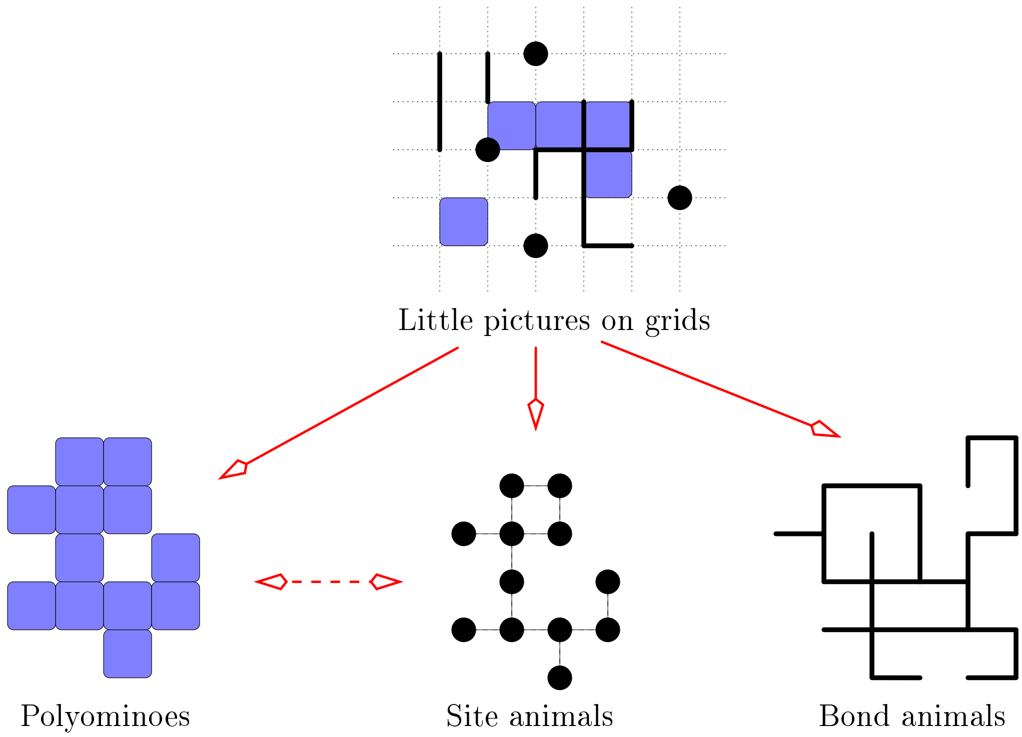



Figure 1.9: Little pictures on grids give rise to polyominoes, site animals and bond animals. Site animals and polyominoes are equivalent under a duality transformation.

Consider again the square grid (see figure 1.8). On this grid there are three basic graph-theoretic objects — faces, vertices, edges. In the polyomino (and associated) literature, these objects are usually referred to as *cells*, *sites* and *bonds*, respectively. Just as we constructed polyominoes from cells, we can construct all sorts of connected objects on the grid from combinations of cells, sites and bonds. Let us first consider objects constructed from just one of these three things.

**Definition 1.2.**
In the same way that a polyomino is a connected union of cells, we define:

- a ***site animal*** is a connected union of sites, and
- a ***bond animal*** is a connected union of bonds.

Site animals and polyominoes are closely related objects — if we replace each cell of a polyomino with a site at its centre, a polyomino is replaced with a site animal on the *dual lattice*[15] (see figure 1.10). Consequently if a family of polyominoes can be enumerated according to their area, then the corresponding family of site animals is also solved.

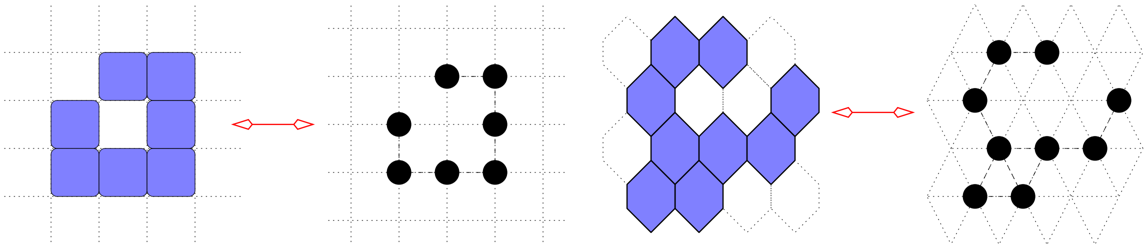


Figure 1.10: Polyominoes with square and hexagonal cells, and the corresponding site animals on the dual lattices.

In this section we describe some of the most commonly studied site animals, bond animals and polyominoes; *self-avoiding polygons* or SAPs (see [121] for example) take a special place in this classification, since they are defined both as bond animals and polyominoes (see figure 1.11). If we consider the perimeter of the polygon then:

**Definition 1.3.**
A ***self-avoiding polygon*** is a bond animal for which every vertex visited by the animal is of degree 2. Alternatively it is the embedding of a simple closed loop into the square grid.

On the other hand, if we consider the interior of the polygon then:

**Definition 1.4.**
A ***self-avoiding polygon*** is a polyomino with no holes; all cells that are not a part of the polygon, must be connected to $\infty$.

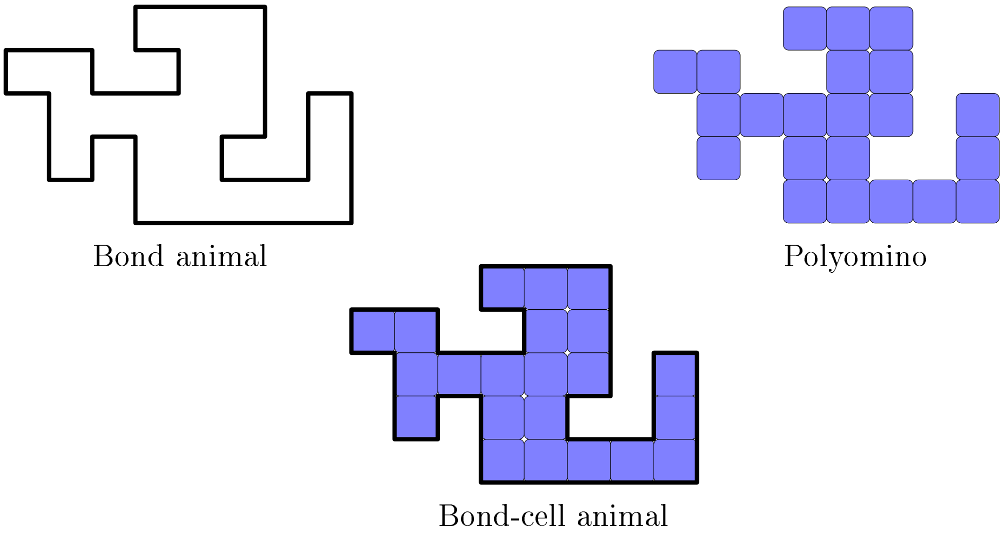



Figure 1.11: Polygons: If we consider only the perimeter, then a polygon is a bond animal. If we consider the interior (area), then a polygon is a polyomino. If we consider both area and perimeter, then a polygon is somehow a *bond-cell* animal.

Polyominoes, site-animals, bond-animals and self-avoiding polygons remain unsolved in two dimensions and higher; the best solutions remain either brute-force methods or algorithms that are exponentially faster than brute-force, but still require exponential time. It is not certain that any of these models do have a “nice” solution; no-one has proved that $c_n$ can be computed in polynomial time. Imposing topological restrictions on these models allows us to make some progress; if we simplify the possible animal and polyomino “topologies” then we can find solutions. Indeed all solved models have strict topological restrictions; with the exception of only two[16], all are either *directed* or *convex* or both (see below and chapter 2).

Below we give a taxonomy of these models, both solved and unsolved, which we have divided into three parts:

- site animals and polyominoes — enumerated according to area (number of sites or cells),
- bond animals — enumerated according to the number of bonds,
- polygon models — enumerated according to perimeter (number of bonds) and area (number of cells).

Since polygon models can be considered both as polyominoes and bond-animals, it makes sense to describe them separately. Further since there are a large number of polygon models

[15] The square grid is dual to itself, while the dual of triangular grid is the hexagonal grid (and *vice versa*).

[16] Spiral walks on the square and triangular lattices [90, 100, 157, 15, 101] and 3-choice polygons on the square lattice [42].

to describe we have given the majority of their definitions in chapter 2. One could construct more complicated objects, however we do not explore this possibility in this thesis.

By far the majority of polyomino and animal results are for the square lattice, and it should be assumed (unless stated otherwise) that the definitions and results described below are for the square lattice.

### 1.3.1 Site animals and polyominoes

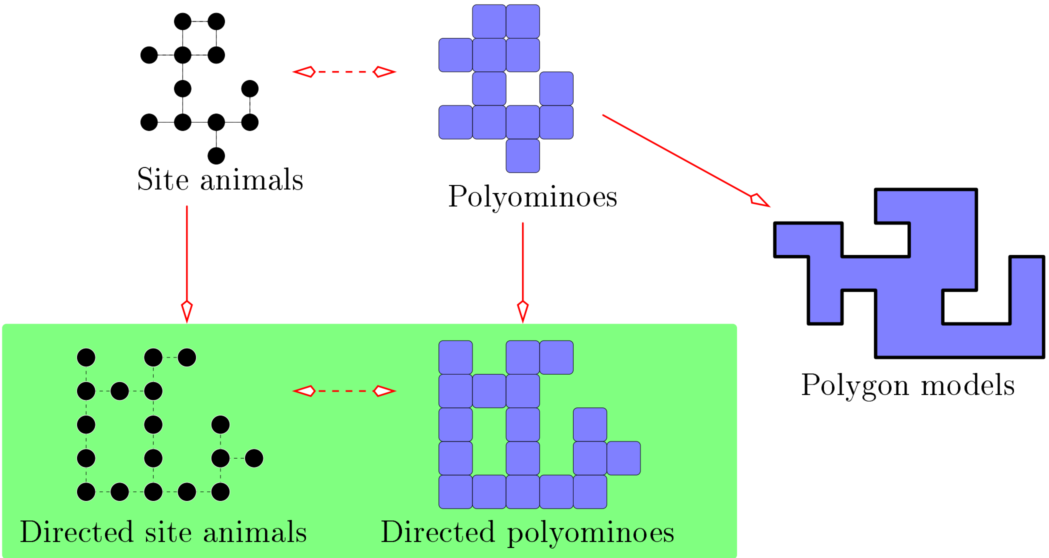



Figure 1.12: Polyomino and site animal taxonomy. Directed site animals and polyominoes have been solved. Polyominoes also give rise to polygon models.

There is only one solved family of site animals and polyominoes (that is not a family of polygons) — directed polyominoes[17]:

**Definition 1.5.**
A polyomino is a ***directed polyomino*** if there exists a cell, called the *root* or *source*, from which all other cells can be reached by a path (remaining inside the polyomino) that takes only north and east steps. The definition of *directed site animals* is similar.

Directed animals enumerated according to *area* were first solved on the square and triangular lattices[18] by Dhar [54, 55] (using a formal connection with lattice gas models) and Gouyou-Beauchamps & Viennot [76] (by more combinatorial methods). The proof was substantially simplified by Bétréma and Penaud [12, 13, 129]; this proof has been extended to take the right half-width into account (see chapter 4). The generating function of directed animals, counted

[17] One can also solve directed polyominoes whose cells lie on or below the line $y = x$; these arise in the solution of directed polyominoes.

[18] Using lattice duality square and hexagonally celled directed polyominoes are also solved.

according to their area and number of cells only supported from below[19] has been solved (both on the square and triangular lattices) using the connection between two-dimensional directed animals and one-dimensional gas models [25]. Directed animals on the hexagonal lattice (and polyominoes on the triangular lattice) have not been solved (possible reasons for this are discussed in [46]).

One of the few animal results in higher dimensions are directed site animals on a cubic lattice. Dhar [55] showed that a model of directed animals on the cubic lattice (in which nearest-neighbour and next-nearest neighbour steps are permitted) is equivalent to a statistical mechanical model called the hard-hexagon model which was solved by Baxter [8].

As noted above almost all solved models are either directed or convex or both. The largest families (in terms of their growth constants) with these properties — directed animals and column-convex polyominoes[20] (see chapter 2) — have been solved. Consequently if we are to find or invent larger classes of solvable animals and polyominoes, we must look beyond convexity and directedness. In chapter 4 we define and solve three new larger classes of triangular lattice animals (and two on the square lattice) that are neither convex nor directed; the starting point for this is the beautiful mapping between directed animals and *pyramids of dimers* first observed by Viennot.

### 1.3.2 Bond animals

Like polyominoes and site-animals, bond animals remain unsolved in two dimensions and higher, it therefore makes sense to consider bond animals with simpler topologies (see figure 1.13):

**Definition 1.6.**

- ***Directed bond animals*** are bond animals in which all bonds can be reached from the vertex at the origin by a path (remaining inside the animal) that takes only north and east steps.
- ***Lattice trees*** are bond animals that contain no closed loops.
- ***Self-avoiding walks*** (SAWs) are bond animals with a *linear* topology. Alternatively they are bond animals, such that every vertex visited by the animal is of degree 2, excepting two vertices (the endpoints) which are of degree 1.

Unfortunately these restrictions have not been sufficient to lead to a solution in any dimension higher than one. It is curious that directed bond animals are yet to be solved when directed site animals have been solved for almost two decades; we discuss possible reasons for this in chapter 3.

Self-avoiding walks (SAWs) are of considerable importance as a model of linear polymers (see below) and have been intensively studied for many years. Even though no exact solution

[19] A cell is supported only from below if there is another cell directly below it, but no cell directly on its left.

[20] Column-convex polyominoes are always polygons.

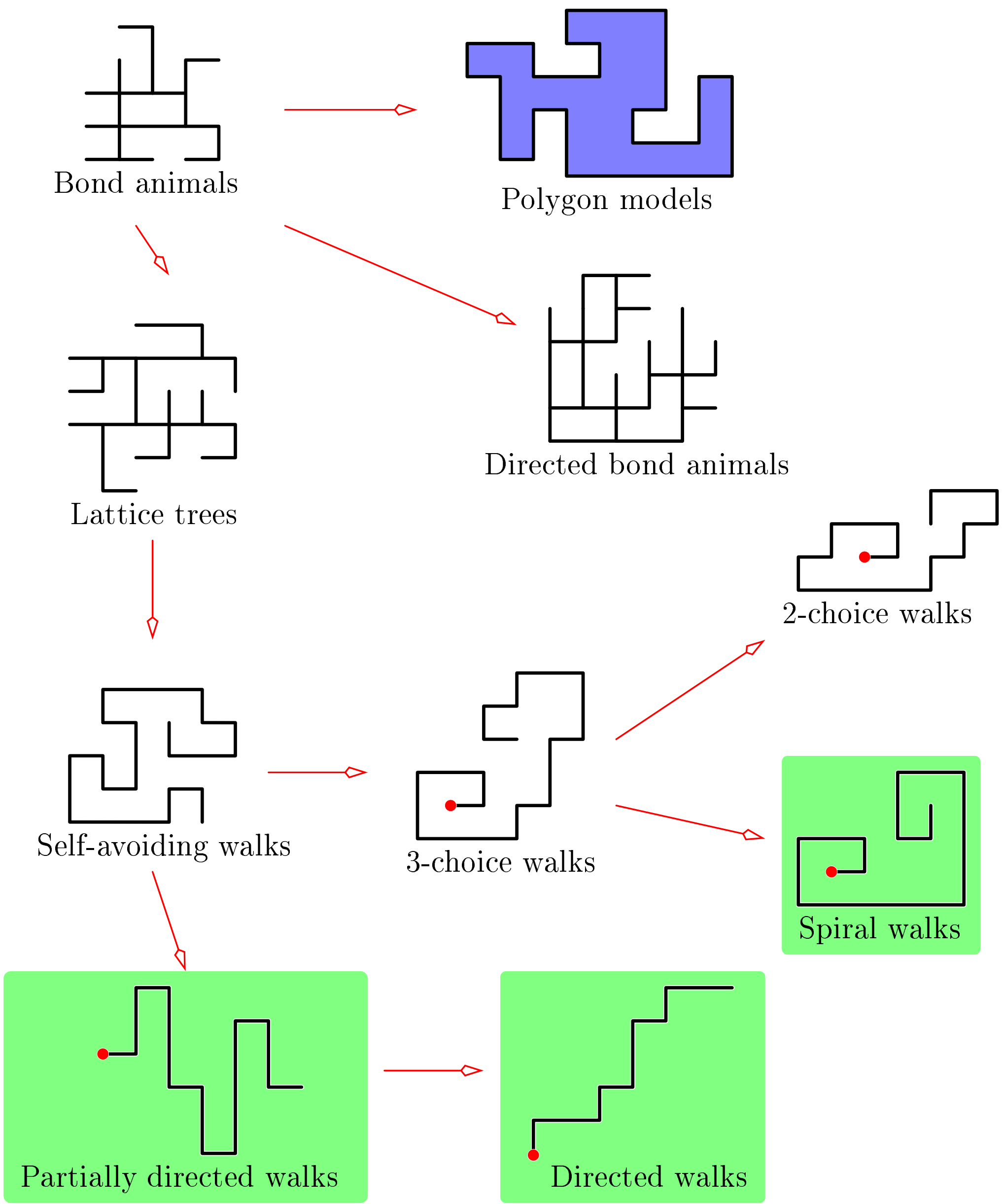



Figure 1.13: The bond animal kingdom — the only solved models (other than polygon models) are spiral walks (on the square lattice, and one of the three possible models on the triangular lattice), partially directed walks and directed walks.

has been forthcoming, a great deal is known about them, by both rigorous and numerical methods (see [121] and references within) — for example, the connection between SAWs and the $N$-vector model [47] gives (non-rigorously) both the critical exponents $\nu$ and $\gamma$ in two dimensions, and the growth constant on the honeycomb lattice [125].

Many modified walk models have been introduced to mimic various physical situations and to allow for easier analysis. These modifications can lead to changes in the asymptotic behaviour of the model. A large family of modified SAW models are *two-step-restricted walks* (TSRWs). A TSRW is constructed by starting at the origin and adding bonds so that the walk is self-avoiding and so that the possible directions of the $n^{th}$ bond (or step) depend upon the direction of the $(n-1)^{th}$ bond according to some rule (consequently the bonds of a TSRW are ordered). In two dimensions, a wide ranging study of TSRW [86] found (using computer enumeration and numerical methods) that the asymptotic behaviour of these TSRWs can be linked to the symmetries of the restricting rule. We describe this survey in chapter 9. Some of the more interesting[21] restricted walks are:

**Definition 1.7.**

- 3-***choice walks*** are TSRW for which clockwise turns are forbidden after steps in the $\pm x$ direction; *i.e.* after an east step it is not possible to step south, and after a west step it is not possible to step north.
- 2-***choice walks*** are 3-choice walks that are forbidden to make more than one successive step in the $\pm y$ direction.
- ***Spiral walks*** are oriented SAWs in which all clockwise turns are forbidden; *i.e.* after an east step it is not possible to step south, after a north step it is not possible to step east, *etc.*
- ***Partially directed walks*** are SAWs whose intersection with any vertical line is connected. Equivalently the walk is forbidden to step west.
- ***Directed walks*** are SAWs that consist entirely of north and east steps.

These walks are discussed in more detail in chapter 9. The only “non-trivial” walk models to have been solved are partially directed walks, directed walks [136] and spiral walks. Two different spiral walk models have been solved; spiral walks on the square lattice [90, 100, 157, 15], and one model[22] of spiral walks on the triangular lattice [101] (although the asymptotics of the other two is known [150]). The growth constants of 3-choice and 2-choice walks are known [158].

In chapter 9 we examine the behaviour of three dimensional TSRW. There are $2^{30} = 1073741824$ TSRW models in three dimensions, and so we have taken only a *very small* subset of these, chosen so as to examine the relationship between asymptotic behaviour of the walk and the symmetries of the restricting rule. Given the extreme difficulty of finding rigorous results for any bond animal models, we have used computer enumeration and numerical methods in this study.

[21] On the square lattice there are 4 directions that each bond may take, and then 3 possible directions for the following bond (since immediate reversal of direction would violate self-avoidance), and so there are 12 possible *two-step configurations*. The different TSRW models are obtained by allowing or disallowing each of these 12 configurations. Hence there are $2^{12} = 4096$ different TSRW models on the square lattice! Most of these are essentially either zero- or one-dimensional.

[22] Since there are both $60°$ and $120°$ turns on the triangular lattice, there are three possible models of spiral walks (depending on which of these turns is allowed). — the solved model allows only $120°$ turns.

### 1.3.3 Polygons

While self-avoiding polygons remain unsolved, a number of topologically restricted families of polygons have been solved; these fall into three categories:

- 3-choice polygons,
- diagonally-convex directed polygons, and
- the many families of column-convex polygons.

**Definition 1.8.**

- A 3-***choice polygon*** is a polygon whose perimeter obeys the 3-choice walk rule (see above); *i.e.* the perimeter is constructed in the same manner as a 3-choice walk, excepting that the walk must return to the origin (see figure 1.14).
- A polygon is *diagonally-convex* if its intersection with lines of the form, $x + y = k$ (with $k \in \mathbb{Z}$), is connected. A polygon is a *diagonally-convex directed polygon* if it is diagonally-convex and also a directed polyomino.
- A polygon is *column-convex* if its intersection with any vertical line is connected. The definition of *row-convex* is similar.

It is worth noting that while any column-convex polyomino is necessarily a polygon, a diagonally-convex polyomino may not be a polygon (the polyomino on the left of figure 1.10 is diagonally convex but not a polygon).

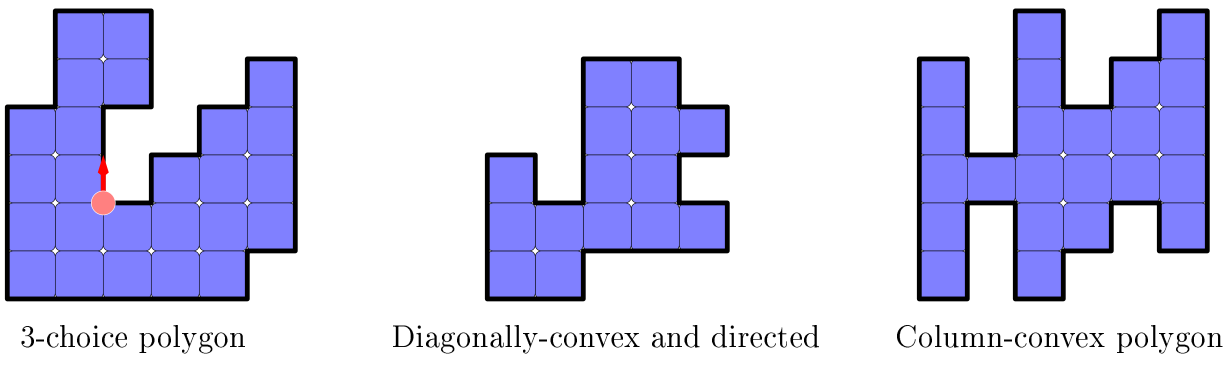



Figure 1.14: Examples of a 3-choice polygon, a diagonally-convex directed polygon and a column-convex polygon. The perimeter of the 3-choice polygon is a 3-choice walk that returns to the origin (which is highlighted).

In the next chapter we give the definitions of all the column-convex families of polygons: column-convex polygons, directed column-convex polygons, bargraphs, fully convex polygons[23], directed convex polygons, staircase polygons, stacks, Ferrers diagrams and rectangles.

Diagonally convex directed polygons, 3-choice polygons and the many families of column-convex polygons have all been enumerated according to their *anisotropic* perimeter (number of vertical and horizontal bonds) and area (see tables 2.1 and 2.2 in the next chapter). Further, they have been enumerated by both of these parameters simultaneously; for example, we know that there are exactly 4 column-convex polygons with an area of 3, a horizontal perimeter of 4 and a vertical perimeter of 4 (these are the "L" shaped polyominoes in figure 1.6). Almost all of these solutions rely on the same two methods:

- by splitting a polygon into two smaller polygons at a point where it is especially thin, one can construct equations for the generating function that can then (hopefully) be solved — this method is sometimes referred to as "wasp-waist factorisation". A similar method consists of mapping polygons to words in algebraic languages and then using the grammar of that language to determine the generating function.

- by using a column-by-column construction, one can find a (solvable) recurrence or functional equation for column-convex polygons (diagonally convex directed polygons can be constructed diagonal-by-diagonal). This method was used by Temperley to enumerate the largest family of column-convex polygons according to their area [151] and is sometimes known as the "Temperley method"; we describe this technique in chapter 5.

While all solved polygon models have been enumerated according to their area and (anisotropic) perimeter, there are very few results concerning the enumeration of polygons according to other parameters. One parameter of considerable importance in the modelling of random media is site-perimeter (see section 1.4.1 below). All the polygons which have been enumerated according to their site-perimeter are both row and column convex; in these cases the techniques used to take site-perimeter into account are the same as those used to enumerate perimeter and so the perimeter and site-perimeter solutions are of the same nature. When we consider polygons that are column-convex but not row-convex, we find that the site-perimeter and perimeter behave differently, and we can no longer use the same methods. In chapter 6 we use a variation of the column-by-column construction (which we describe in chapter 5), to extend site-perimeter enumeration results to the simplest family of non-convex polygons, *bargraphs*.

From the above discussion it should be clear that finding families of animals, polyominoes or polygons that we can actually solve is a non-trivial exercise. Guttmann and Enting [83] noticed that the anisotropic generating functions[24] of most solved statistical mechanical models have a very simple analytic structure, while those of unsolved models are considerably

[23] Polygons that are both row-convex and column-convex are called *convex* polygons — this is an ambiguous convention.

[24] In the case of polygon models and bond-animals, this generating function enumerates animals according to the number of vertical and horizontal bonds.

more complicated (we will be more precise about what we mean by this in chapter 3). They proposed that by examining the first few terms of the anisotropic generating function one could test the "solvability" of a model; in particular one could test if the generating function is likely to be expressible in terms of "nice" functions such as *differentiably finite* functions (see chapter 7). Applications of this technique show that many bond animal problems we would like to solve, including bond animals and self-avoiding polygons, are not solvable in terms of these nice functions [82, 97].

In chapter 3 we describe a "squashing" technique, which we call *haruspicy*, that allows us to examine aspects of the analytic structure of anisotropic perimeter generating functions of bond-animals, regardless of whether the solution is known in closed form. For self-avoiding polygons (see chapter 7) we are able to use this technique to sharpen the numerical results of Guttmann and Enting's test into proof — *i.e.* we prove that the anisotropic perimeter generating function of SAPs is not a differentiably finite function. In chapter 8 we describe a type of functional symmetry called *reciprocity* or *inversion relations*. We are able to demonstrate that a number of solved and unsolved polygon models satisfy such symmetries and we show that in certain favourable circumstances these symmetries can be used to solve the model.

## 1.4 Why do we want to count them?

It is quite likely that almost everyone who has dabbled in polyomino and animal enumeration has been asked by a friend or relative (or even asked themselves):

**Question 9.** ***Why count polyominoes and animals?***

In the recent past the mathematics of counting, *enumerative combinatorics*, has undergone something of a renaissance; arguably the main force behind this has been the role of combinatorics in many problems found at the interface between mathematics and other sciences — particularly statistical physics, computer science and theoretical chemistry. Many different types of animals and polyominoes can be found in abundance at this interface.

To give the flavour of the sorts of models and problems in which animals and polyominoes can be found, we will describe how animal enumeration problems can be found in the theory of integer partitions, and in the mathematical treatment of magnets, polymers and random media.

### 1.4.1 Percolation models

Percolation theory was introduced by Broadbent and Hammersley [38] as a mathematical model (or a collection of mathematical models) of random media. Consider the square grid; let each site be "occupied" with probability $p$ or "vacant" with probability $(1 - p)$, independently of all the other sites on the lattice. A "percolation cluster" on this grid is a collection of nearest-neighbour occupied sites — a site animal! The number of occupied sites within a cluster is called its "mass" (see figure 1.15). Instead of occupying the sites on the lattice, we can also consider occupying the bonds or a combination of both — giving rise to site-percolation, bond-percolation and site-bond-percolation. Percolation clusters can (and have) been used to model a wide array of phenomena; from oil deposits to forest fires (see [148, 79] and references therein)

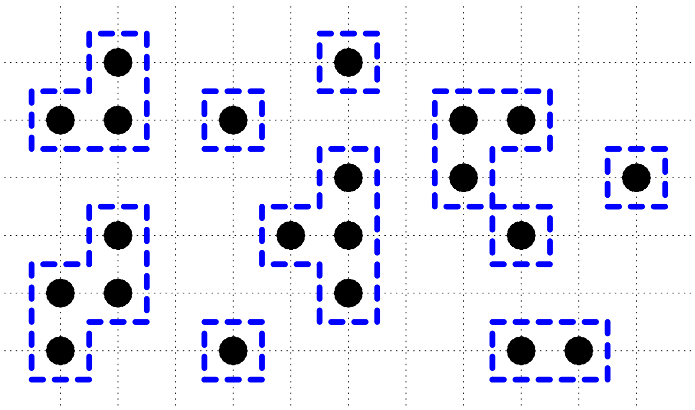


Figure 1.15: Percolation clusters: five clusters of mass 1, one of mass 2, two of mass 3 and two of mass 4.

As the occupation probability, $p$, is increased from zero, the average or typical clusters become larger (both in terms of the number of sites and geometric size). If we define $P(p)$, "the percolation probability", to be the probability that the origin is part of an *infinite* cluster, then we find that below a certain value $p = p_c$, called the critical probability, $P(p)$ is always zero. As $p$ is increased above $p_c$, the percolation probability becomes non-zero (see figure 1.16). This change in behaviour is an example of a *phase transition*.

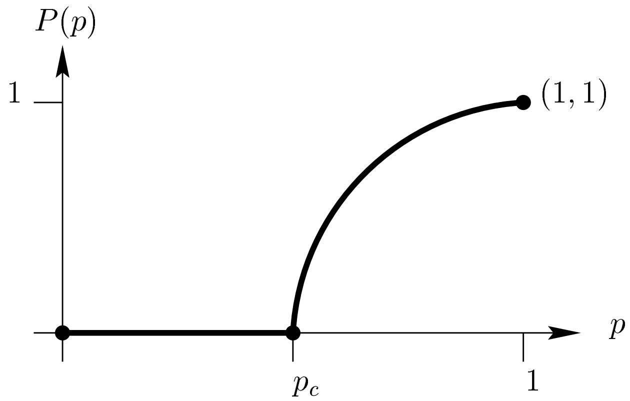



Figure 1.16: The expected behaviour of the percolation probability, $P(p)$.

As an example consider an orchard, in which all of the trees are arranged on the vertices of a large square grid. Suppose that a tree will become infected with a nasty disease with probability $p$ if one of its neighbours is infected. Say a few of the trees in the orchard become infected and the disease spreads through the orchard. After a few days the disease has stopped spreading and there are a number of clusters of diseased trees. Obviously, in this scenario we wish to minimise the size of these clusters; if we know how $p$ varies with the distance between trees (presumably it decreases as we place trees further apart), how far apart do we have to plant the trees to stop the epidemic from destroying a large portion of the orchard? We could just place the trees a long way apart (a small value of $p$), but then there would not be many trees in the orchard and we would go broke. For the orchard to be viable we need lots of trees, and so we need to choose a large value of $p$, but not so large that the orchard is endangered. If $p$ is chosen below the critical probability, then we know that the average cluster size will be quite small, and the disease cannot spread far. On the other hand, if $p$ is above the critical probability, then there is a non-zero probability that the disease could spread to a significant part of the orchard. Hence we must choose $p$ as close to $p_c$ as we dare, while still ensuring that $p$ is less than $p_c$.

The percolation probability, $P(p)$, can be written in terms of site animals enumerated according to their area and another parameter, the *site-perimeter*. The site-perimeter of a site animal is the number of nearest-neighbour sites that are not part of the animal (see figure 1.17). Let us write the generating function of all *finite* site animals, $A$, enumerated

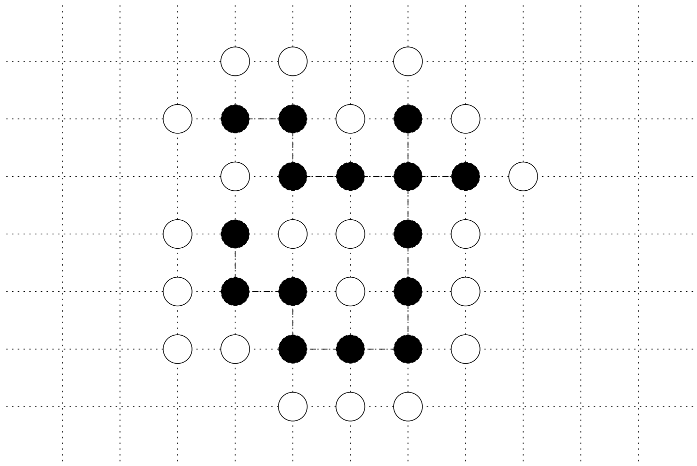


Figure 1.17: A site animal and its site-perimeter. This animal contains 15 sites, and has a site-perimeter of 21.

according to both area and site-perimeter as

$$G(p,q) = \sum_A p^{|A|} q^{\mathrm{site.p}(A)} \tag{1.6}$$

where $|A|$ and site.p($A$) denote, respectively, the area and site-perimeter of an animal $A$. The probability that the origin is part of a given cluster $A$ is equal to $|A|p^{|A|}(1-p)^{\mathrm{site.p}(A)}$, since $|A|$ sites must be occupied, site.p($A$) sites must be vacant, and there are exactly $|A|$

ways of placing the animal over the origin. Consequently the probability that the origin is part of a finite cluster is (expanding in $q = 1 - p$ about $q = 0$)

$$\sum_A |A|(1-q)^{|A|} q^{\text{site.p}(A)} = (1-q) \left( \frac{\partial G}{\partial p} \right) \Big|_{p=1-q} \tag{1.7}$$

and the percolation probability is the probability that the origin is *not* part of a finite cluster (nor vacant):

$$P(p) = 1 - q - (1-q) \left( \frac{\partial G}{\partial p} \right) \Big|_{p=1-q}. \tag{1.8}$$

So a knowledge of $G(p,q)$ would give us an expression for $P(p)$. Unfortunately computing the site-perimeter of polyominoes is a very difficult problem; to date, only polygons that are fully convex (rectangles, Ferrers diagrams, stacks, staircase polygons, directed convex polygons and convex polygons) have been solved by site-perimeter. In chapter 6 we find the site-perimeter generating function of the simplest family of non-convex polygons — bargraph polygons.

### 1.4.2 Partitions of integers and their generalisations

One of the simplest and most natural mathematical operations is the adding of integers. Integer partitions describe the ways in which a positive integer can be written as the sum of other positive integers. For example the number 4 can be written as

$$\begin{array}{lllll} & & 4, & & \\ & & 3+1, & 1+3, & \\ 4 & = & 2+2, & & \\ & & 2+1+1, & 1+2+1, & 1+1+2, \\ & & 1+1+1+1 & & \end{array}$$

So there are 8 ways of writing the number 4 as a sum of positive integers. These 8 different sums are *compositions*.

**Definition 1.9.**
A *composition* of a positive integer $n$ is an ordered finite sequence of positive integers $\lambda_1, \dots, \lambda_k$ such that $\sum_{i=1..k} \lambda_i = n$.

Implicit in this definition is that we care about the order in which the summands appear; the composition $3+1$ is not the same as $1+3$. Since addition is commutative, this distinction seems a little artificial, and it would be more natural to consider the compositions $3+1$ and $1+3$ to be the same object. One way of ensuring this is to require the summands to be in non-increasing order — this leads us to *partitions*.

**Definition 1.10.**
A *partition* of a positive integer $n$ is a finite sequence of *non-increasing* positive integers $\lambda_1, \dots, \lambda_k$ such that $\sum_{i=1..k} \lambda_i = n$.

Using this we find that there are 5 partitions of the number 4.

$$4 \quad = \quad \begin{array}{lll} 4, & 3+1, & 2+2, \\ 2+1+1, & 1+1+1+1 & \end{array}$$

There is a vast array of combinatorial and algebraic structure to be explored in the theory of partitions, far more than we have time for here (see [4] for more on this subject). Instead let us consider the graphical representation of partitions and compositions.

Let us represent a positive integer, $k$, by a column of $k$ cells. A sequence of positive integers is represented by a sequence of columns. In this way every composition can be uniquely represented by a sequence of columns of cells — the total number of cells is the total of the composition (see figure 1.18). These sequences of columns are bargraph polygons

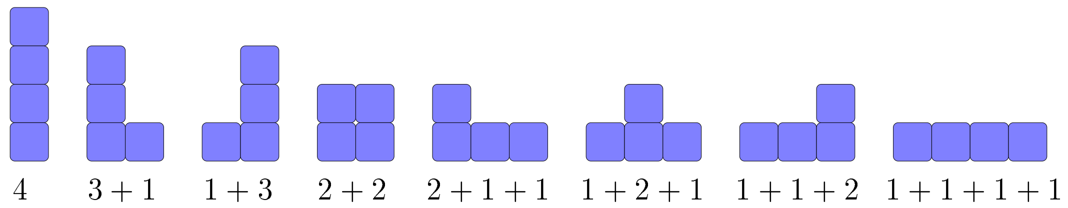



Figure 1.18: Representing the 8 compositions of the number 4 by sequences of columns.

and so the number of compositions of $n$ is equal to the number of bargraph polygons of area $n$.

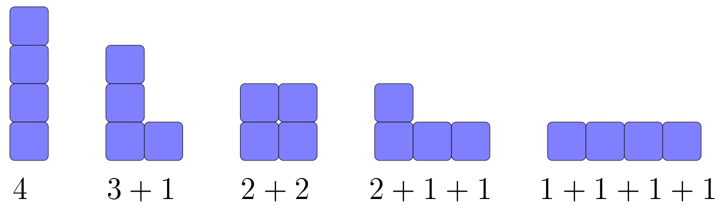



Figure 1.19: Representing the 5 partitions of the number 4 by sequences of columns.

If we consider partitions instead of compositions, then the sequence of columns must be non-increasing — this gives us Ferrers diagrams (see figure 1.19). Generalising this idea, we consider other subsets with other conditions; for example, if we require that the sequence be unimodal then we obtain stack polygons. Polyominoes can be seen as a generalisation of these objects.

### 1.4.3 Polymer models

One of the most transparent[25] applications of animals is to the modelling of polymers in solution. A *polymer* is a large molecule that is made up of many *monomers* connected

[25]In the sense of being a very direct application of animals and polyominoes.

together by chemical bonds; depending on how the monomers are connected to each other polymers can have very complicated topologies.

Polymers in dilute solution undergo a *phase transition* known as the *coil-globule* transition (see figure 1.20). At low temperatures (or in a poor solvent) the attractive interactions between monomers pull the polymer into a dense ball-like configuration or "globule". At high temperatures (or in a good solvent) the interactions are mediated by the solvent molecules, and the typical configurations are open coils. At a specific temperature, the $\theta$-point, the polymers undergo a *phase transition* (much like when water boils into vapour).

These phases (coil and globule) are characterised by the asymptotic behaviour of the average size[26] of a polymer containing $n$ monomers. If we consider the radius of gyration, $R_g(n)$, of a polymer (which is the average distance of a monomer from the polymer's centre of mass), then the average radius of gyration[27] of polymers with $n$ monomers is expected to behave as

$$\langle R_g \rangle_n \sim An^\nu \qquad \text{as } n \to \infty \tag{1.9}$$

where the *critical exponent*, $\nu$, is expected to be the same for all (mathematical or real) linear polymers. The value of $\nu$ changes between the phases; for linear polymers in three dimensions the best numerical estimates of $\nu$ are:

- in the swollen phase: $\nu \approx 0.588$,
- in the collapsed phase: $\nu = 1/3$.

It should be noted that if the polymer behaved like a random walk then one would have $\nu = 1/2$.

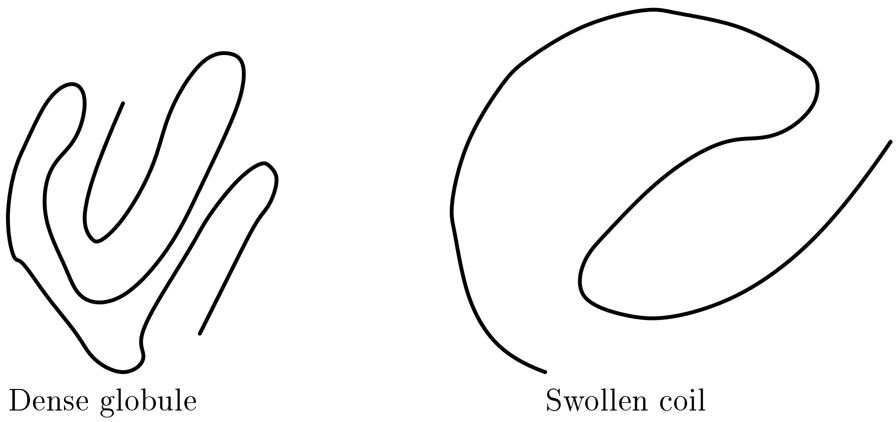



Figure 1.20: In the collapsed phase the polymer forms dense ball-like "globules", while in its swollen phase it forms more open coils.

Consider a linear polymer (each monomer is connected to only two others). In a real polymer the monomers occupy positions in continuous space ($\mathbb{R}^3$), and the bonds between them

[26] Not the number of monomers, rather a measure of the space occupied by the polymer.
[27] Other measures of the size of a polymer are expected to behave similarly.

are constrained to have only certain angles (depending upon the nature of the monomers). We can simplify this situation by embedding the polymer into discrete space ($\mathbb{Z}^3$), requiring that the monomers exist at integer coordinates only a single lattice spacing apart. Since the monomers cannot occupy the same space, the polymer embedded into discrete space is in fact a self-avoiding walk. To account for interactions between monomers, one can study the number of nearest-neighbour contacts (the number of points at which the walk passes within one lattice spacing of itself); by favouring or disfavouring these contacts one can study attractive and repulsive interactions — see figure 1.21. Lattice trees and bond animals can be used to model polymers with more complicated topologies.

The self-avoiding walk model was introduced by Orr [127] to explore the geometric properties of linear polymers in a good solvent. At first glance it appears as though this model is far too simple to have any hope of modelling such a complex situation, however the phenomenon of *universality* tells us many quantities are not dependent on the specific details of the system, rather they are determined only by its *universality class*. The *universality class* is determined by the very general properties of the system, such as its dimension, and not by the very specific details (such as the type of lattice). All systems (real or mathematical) within the same universality class share the same dominant asymptotic behaviour close to a phase transition — and so any member of the universality class can be used to determine this behaviour. For example, the critical exponent, $\nu$, in any given phase is expected to be the same for all linear polymers in three dimensions, regardless of precise details of the system — $\nu$ for SAWs is exactly the same as $\nu$ for almost any [28] linear polymer! On the other hand, other quantities such as the exact location of the $\theta$-point are not universal, and can only be determined by experiment.

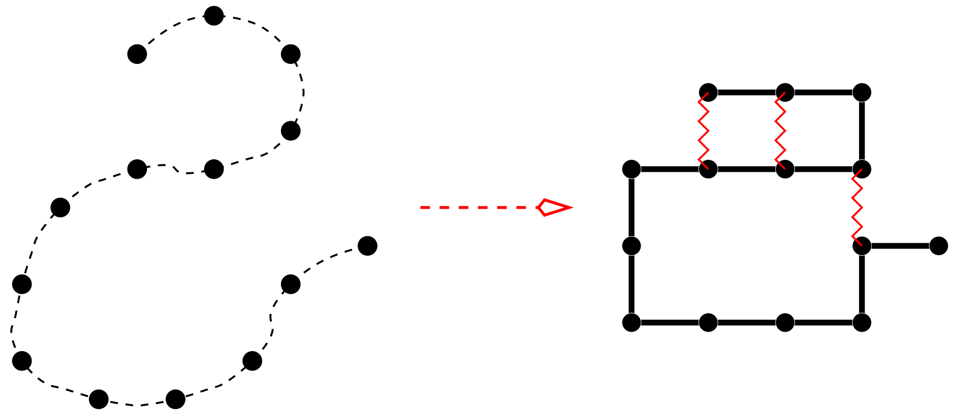


Figure 1.21: Embedding a polymer into discrete space ($\mathbb{Z}^3$) gives an interacting self-avoiding walk.

[28] Any linear polymer that is flexible and has short-range interactions.

### 1.4.4 The Ising model and counting graphs

The last example we will consider is the Ising model. Unlike the previous three examples in which animal and polyomino enumeration arise quite directly, one must venture a little way beyond the definition of the Ising model before the animals can be found lurking.

The Ising model is perhaps the most famous and widely studied model in statistical mechanics. It models the effect of temperature and external magnetic fields on the properties of a magnet. When a piece of iron is placed in a magnetic field it becomes magnetised. If the magnet is then heated, the strength of the iron's magnetic field weakens until it disappears — the temperature at which this happens is called the Curie temperature. This change in behaviour is much like the transition water undergoes when it evaporates, and is known as a *phase transition*. Phase transitions are also observed in combinatorial models and exhibit themselves as changes in their asymptotic behaviour.

The Ising model was solved in one dimension by Ising [93], but was shown not to undergo any phase transition. Onsager [126] solved[29] the two dimensional Ising model (in the case of no external magnetic field), and it was shown to undergo a phase transition like those exhibited by real magnets. The three dimensional Ising model remains unsolved.

Let us demonstrate how animals arise in this model. Consider a finite portion of the square lattice with magnetic "spins" placed at the vertices. These spins could be vectors, scalars or even quantum mechanical spin operators. The Ising model considers only the simplest case; each spin is either spin up ($\sigma_P = +1$) or spin down ($\sigma_P = -1$) (see figure 1.22).

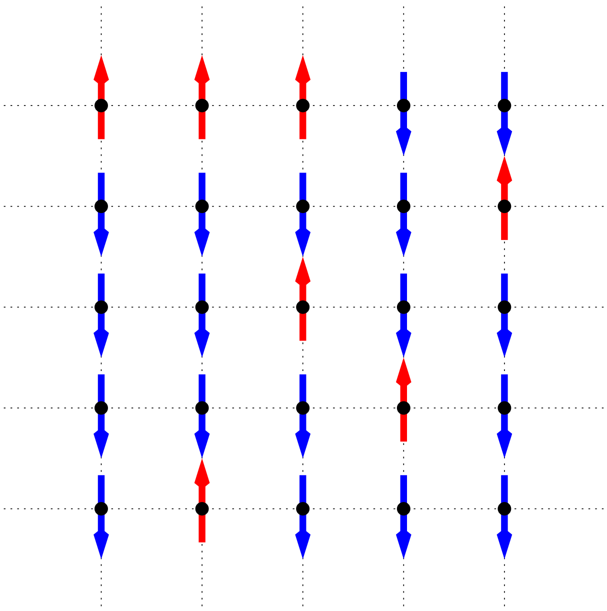


Figure 1.22: Ising model

[29]It has not been entirely solved; some properties of the model have been found — the free energy and spontaneous magnetisation — while others — most notably the susceptibility — remain unknown.

The interaction energy of a configuration $\{\sigma\}$ is defined to be

$$E\{\sigma\} = -J \sum_{P,Q}^{*} \sigma_P \sigma_Q - H \sum_P \sigma_P, \tag{1.10}$$

where $\sum_{P,Q}^{*}$ denotes the sum over all pairs of nearest neighbour spins, $P$ and $Q$[30], $J$ is the "coupling constant" that defines the strength of the interaction between spins, and $H$ represents the interaction between the spins and the external magnetic field (if there is one).

The key to any statistical mechanical problem is the computation of the *partition function* which is given by

$$Z_N = \sum_{\{\sigma\}} \exp\left(-\beta E(\sigma)\right), \tag{1.11}$$

where the sum is over all possible configurations, $\sigma$, of the system, $E(\sigma)$ is the energy of a given configuration, $N$ is the number of sites in the lattice, $\beta = 1/kT$, $k$ is Boltzmann's constant and $T$ is the absolute temperature. A knowledge of the partition function of a system is sufficient to find many other relevant thermodynamic quantities, such as the internal energy and entropy.

Onsager's solution of the two dimensional Ising model in zero external field ($H = 0$) is far from trivial, and we will not discuss it in this thesis (except for a brief discussion of transfer matrices in chapter 8). Van der Waerden [153] showed how the evaluation of the partition function can be translated into a bond animal enumeration problem. We give an outline of this approach.

When $H = 0$ the partition function becomes

$$Z_N = \sum_{\{\sigma\}} \prod_{P,Q}^{*} \exp(\nu \sigma_P \sigma_Q), \tag{1.12}$$

where $\nu = J/kT$ and $\prod_{P,Q}^{*}$ denotes the product over all pairs of nearest-neighbour spins, $P$ and $Q$. It is not difficult to verify that

$$\begin{aligned} \exp(\nu \sigma_P \sigma_Q) &= \begin{cases} e^{-\nu} & \text{if } \sigma_P \neq \sigma_Q \\ e^{+\nu} & \text{if } \sigma_P = \sigma_Q \end{cases} \\ &= (\cosh \nu)\Big(1 + \sigma_P \sigma_Q \tanh \nu\Big). \end{aligned} \tag{1.13}$$

Using this we can rewrite the partition function as

$$Z_N = (\cosh \nu)^B \sum_{\{\sigma\}} \prod_{P,Q}^{*} \left(1 + \omega \sigma_P \sigma_Q\right), \tag{1.14}$$

where $\omega = \tanh \nu$, and $B$ is the number of bonds in the lattice. From here it is not hard to rewrite this as a sum over graphs on the square grid.

[30] Since we consider only a finite portion of the lattice, the number of nearest-neighbour pairs is finite, and so the sum is convergent.

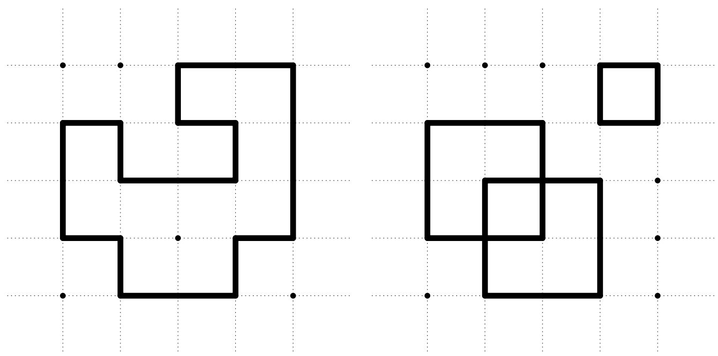


Figure 1.23: Typical graphs that contribute to the partition function — some of these graphs are self-avoiding polygons. It is possible for the graphs to be disconnected.

Let us consider each bond on the grid to be either occupied or vacant, and let $\mathcal{G}$ be the set of all possible combinations of occupied bonds (each combination forms a graph with vertex set equal to the $N$ sites in the lattice). Note that though these graphs are similar to bond animals, they are not bond animals, since they can be disconnected and they are not translationally invariant. The partition function can be rewritten as

$$Z_N = (\cosh\nu)^B \left( \sum_{G\in\mathcal{G}} \omega^{|G|} \prod_P \sum_{\sigma_P=\pm1} \sigma_P^{\deg_G(P)} \right), \tag{1.15}$$

where $\prod_P$ is a product over all $N$ sites, $|G|$ is the number of occupied bonds in $G$ and $\deg_G(P)$ is the number of occupied bonds incident on the vertex $P$ in the graph $G$. Now, for any spin

$$\sum_{\sigma_P=\pm1} \sigma_P^n = \begin{cases} 0 & \text{if } n \text{ is odd} \\ 2 & \text{if } n \text{ is even} \end{cases} \tag{1.16}$$

and so when we sum over the possible spin configurations in equation (1.15), *any* graph, $G$, that has a vertex of odd degree will contribute zero while all other graphs will contribute $2^N$. Let us define $\mathcal{G}'$ to be the subset of $\mathcal{G}$ such that for every $G \in \mathcal{G}'$ every vertex in $G$ has even degree. If we define $n(r)$ to be the number of graphs in $\mathcal{G}'$ that contain $r$ bonds, then the partition function is

$$Z_N = 2^N(\cosh\nu)^B \sum_{r=0}^{\infty} n(r)\omega^r, \tag{1.17}$$

where $n(0) = 1$. So the problem has been reduced to finding the generating function of a set of graphs, which are related to bond animals (though they are not restricted to be connected, nor are they defined up to translation). These graphs (see figure 1.23) look like self-avoiding polygons, or groups of overlapping self-avoiding polygons — in fact, self-avoiding polygons were introduced by Temperley [151] as a special case of these graphs.

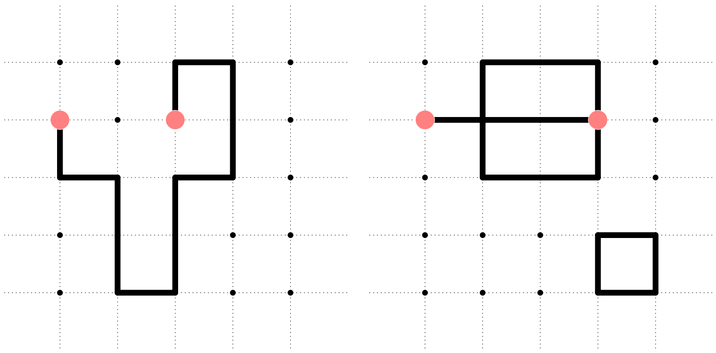


Figure 1.24: Typical graphs that contribute to the correlation function — some may be self-avoiding walks. The vertices of odd degree are highlighted.

Another quantity of interest is the pair correlation function, $\langle \sigma_k \sigma_l \rangle$, which is defined by

$$\langle \sigma_k \sigma_l \rangle = Z_N^{-1} \sum_{\{\sigma\}} \sigma_k \sigma_l \prod_{P,Q}^{*} \exp(\nu \sigma_P \sigma_Q). \tag{1.18}$$

This function is a measure of the degree of order within the state of the system — if there is long range order in the system, then separated spins will tend to be in the same state and the pair correlation function will take a value bounded away from 0. Following the same argument used for the partition function, we can rephrase the pair correlation function as the generating function of a new set of graphs in which every vertex is of even degree *excepting* $k$ and $l$, which must be of odd degree (see figure 1.24). In the same way that self-avoiding polygons arose in the partition function, we find that self-avoiding walks arise in the graphs contributing to the pair correlation function.

The $N$-vector model discussed above is a generalisation of the Ising model, in which the spins can take one of $N$ states. These $N$ states are equally spaced unit vectors in $\mathbb{R}^N$, and the interaction energy of a configuration $\{\sigma\}$ is defined to be

$$E\{\sigma\} = -J \sum_{P,Q}^{*} \vec{\sigma_P} \cdot \vec{\sigma_Q}, \tag{1.19}$$

and so when $N = 2$ we arrive back at the Ising model. In the limit $N \to 0$ the $N$-vector model is equivalent (in some sense) to self-avoiding walks [47].

# 1.5 In this thesis

Now that we have given a brief introduction to the problem of polyomino and animal enumeration, let us describe the structure and contents of this thesis in more detail. In chapter 2 we review the many polygon models that have been solved; all of these models, with a single exception, are convex in some way. We give a table of the first[31] solutions of these models, and comment on the analytic structure of these solutions. Material covered in this thesis can be broken down into roughly four different themes; techniques, exact solutions, properties of solutions, and application of numerical methods.

- ***Techniques:***
  - **Chapter 3 — Haruspicy and anisotropic generating functions:** We present a *new* method for examining some of the properties of the anisotropic generating function of many commonly studied families of bond animals. In particular if one considers the anisotropic generating function $P(x, y)$, then the coefficient of $y^n$ is a rational function whose denominator is a product of cyclotomic polynomials. The method can determine which cyclotomic factors can occur in these denominators and also bound their exponents; we compute bounds for the exponents of certain cyclotomic factors in several bond-animal models and find that these bounds do agree with the available data. It also can be used to compute the generating functions of bond animals with a small finite number of vertical bonds. In chapter 7 we use these haruspicy methods to show that the anisotropic perimeter generating function of self-avoiding polygons is not D-finite.
  - **First half of chapter 4 — Towards larger classes of polyominoes:** The general aim of this chapter is the definition (or invention) of new classes of square lattice animals that would be both large and exactly enumerable. The starting point for this work is a bijection between *directed* animals and certain *heaps of dimers*, called *pyramids*, which was described by Viennot more than 10 years ago. This correspondence greatly simplifies the enumeration of directed animals. Roughly speaking, the first half of this chapter is a *review* of the theory of heaps (in particular heaps of dimers), and the application of this theory to the enumeration of directed animals.
  - **Chapter 5 — Building polyominoes incrementally:** One of the most successful methods of polyomino enumeration is the "*column-by-column*" construction which is sometimes known as the "*Temperley method*". This method can be split into two different steps: $(i)$ finding a recurrence for the coefficients or an equation satisfied by the generating function, and $(ii)$ solving this recurrence or equation. In this chapter we *review* a number of the different approaches to this technique and introduce one more.

[31] *i.e.* a table of who solved what, and when.

- ***Exact solutions:***
  - **Second half of chapter 4 — Towards larger classes of polyominoes:** In the second half of this chapter we show how heaps of dimers can be used to enumerate certain classes of polyominoes. We define two natural classes of heaps that are supersets of pyramids and are in bijection with certain classes of square lattice site animals; we are able to enumerate these classes exactly. The first class has an algebraic generating function and a growth constant of 3.5, while the other has a non-D-finite generating function and a growth constant of 3.58... The growth constants of directed animals and column-convex animals are 3 and 3.20... respectively, and so both the growth constants of these new classes are considerably closer to that of general site-animals, which is expected to be approximately 4.06. We obtain similar results for triangular lattice animals.
  - **Chapter 6 — The site-perimeter of bargraphs:** Though many polygon models have been enumerated according to their area and perimeter, very few have been enumerated according to their site-perimeter, which is a parameter that plays an important role in percolation theory. For fully convex polygons, taking site-perimeter into account is not substantially more difficult than the ordinary perimeter, and similar techniques can be used to find a solution. For families of column-convex polygons (that are not convex), on the other hand, extending perimeter enumeration techniques to take site-perimeter into account is not trivial. In this chapter, we use a variation of the standard column-by-column construction, the Temperley method, to extend site-perimeter enumeration results to the simplest family of column-convex polygons that are not convex — bargraph polygons. The generating function is of a form not previously seen in polyomino enumeration problems.

- ***Properties of solutions:***
  - **Chapter 7 — The solvability of self-avoiding polygons:** We apply the haruspicy techniques developed in chapter 3 to the self-avoiding polygon generating function, $P(x, y)$. In particular we bound the exponent of any cyclotomic factor occurring in the denominator of the coefficient of $y^n$ (for any given $n$). These bounds are in excellent agreement with existing numerical work. We are then able to sharpen these bounds into equalities for certain cyclotomic factors appearing in the denominators of coefficients of certain powers of $y$. This is sufficient to show that the set of singularities of the coefficients of all powers of $y$ is dense on the unit circle, which in turn implies that the anisotropic SAP generating function is not solvable in terms of D-finite functions.
  - **Chapter 8 — Reciprocity and inversion relations:** We derive self-reciprocity properties for a number of polyomino generating functions, including several families of column-convex polygons, three-choice polygons and staircase polygons with a staircase hole. In so doing, we establish a connection

between the reciprocity results known to combinatorialists and the inversion relations used by physicists to solve models in statistical mechanics. For several classes of convex polygons, the inversion (reciprocity) relation, augmented by certain symmetry and analyticity properties, completely determines the anisotropic perimeter generating function.

- ***Application of numerical methods:***
  - **Chapter 9 — Two step restricted walks:** In two dimensions the universality classes (or asymptotic behaviour) of SAWs on the square lattice, restricted by allowing only certain two-step configurations to occur within each walk, has been argued to be determined primarily by the symmetry of the set of allowed rules. In three dimensions early work tentatively found one (undirected) universality class different to that of unrestricted SAW on the simple cubic lattice. This rule was a natural generalisation of the square lattice 'spiral' self-avoiding walk to three dimensions. In this chapter we use numerical techniques to examine a variety of three-dimensional SAW models. These models are carefully chosen with different step restrictions, so as to search for a connection between the symmetry of the rules and possible new universality classes.

The work and manuscripts of chapters 4, 8 and 9 were originally papers with other authors as indicated in the preface. The work of chapters 3 and 7 is wholly my own.

# CHAPTER 2

# Polygon models

# 2.1

# Polygon models

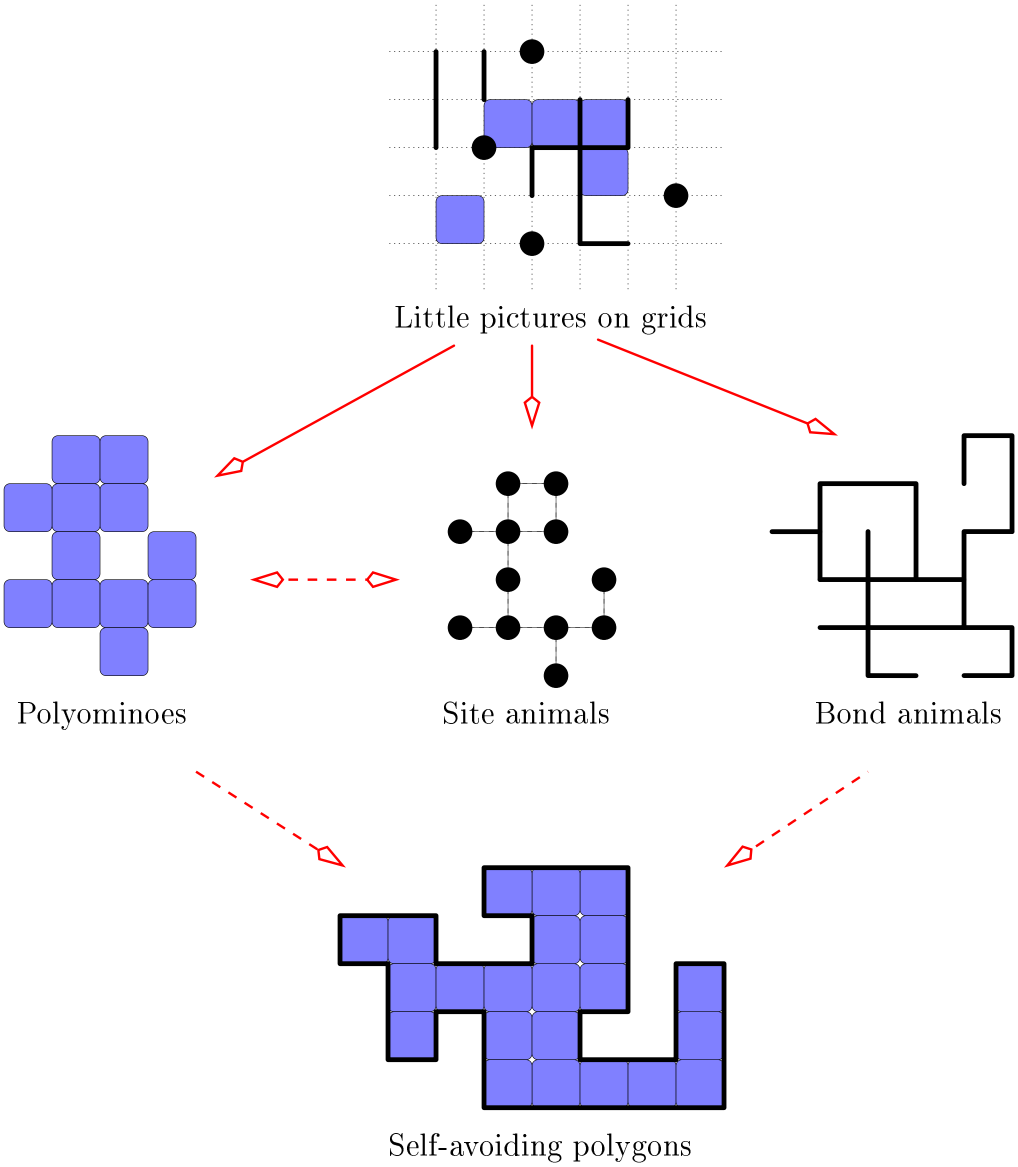



Figure 2.1: Little pictures on grids give rise to polyominoes, site animals and bond animals. Depending on which parameters are considered, self-avoiding polygons are both bond animals and polyominoes.

In the previous chapter we gave a taxonomy of some of the most commonly studied site animals, bond animals, polyominoes and self-avoiding polygons. In this chapter we focus on the definitions of the solved families of polygons on the square lattice (see figure 2.2).

### 2.1.1 Polygon models — definitions

Self-avoiding polygons were introduced by Temperley [151] in 1956. They have not been solved either by area or perimeter. By studying only those SAPs that obey certain topological restrictions, we obtain classes that can be solved. As noted in chapter 1, all solved subclasses (except one) are either *directed* or *convex* or both.

**Definition 2.1.**

- A polygon is *directed* if its cells form a directed polyomino.
- A polygon is *convex* with respect to a direction $\vec{v}$ if its intersection with lines in the direction of $\vec{v}$ is connected.

On the square lattice it is usual to consider column-convexity, row-convexity and diagonal convexity; a little care must be taken with diagonal convexity, and we say that a polygon is diagonally convex if the intersection of a polygon and lines of the form $x+y=k$ (with $k \in \mathbb{Z}$) is connected. It is also usual to refer to a polygon that is column-convex and row-convex as being *convex*[1].
Let us start with the only solved polygon that is not convex:

**Definition 2.2.**
A 3-***choice polygon*** is a polygon whose perimeter is a 3-choice walk, *i.e.* the perimeter is constructed in the same manner as a 3-choice walk — there are no clockwise turns after a step in the $\pm x$ directions. The walk must return to the origin (which is highlighted in figure 2.2).

3-choice polygons have been solved by perimeter and area; more specifically, the area-perimeter generating function is not known but there does exist a polynomial time algorithm for the computation of its coefficients [42]. It is also trivial to alter this algorithm to obtain an area / perimeter solution for 2-choice polygons (which are defined in a similar way to 3-choice polygons).

*Column-convex polygons* were introduced by Temperley [151] as an approximation of self-avoiding polygons that could be solved. Families of *column-convex* or *vertically convex* polygons have now been well studied, many subclasses have been enumerated according to both their area and (vertical and horizontal) perimeter[2] simultaneously.

**Definition 2.3 (Column-convex).**

- A polygon is ***column-convex*** when a vertical line of the form $x = k + \frac{1}{2}$ (with $k \in \mathbb{Z}$), either intersects the perimeter of the polygon in exactly two points or not at all. Equivalently, considering the cells within the polygon, the intersection of the

[1] This convention is a little ambiguous — better to talk of *fully convex* polygons.

[2] The vertical perimeter is the number of bonds running north-south. Horizontal perimeter is the number of bonds running east-west.

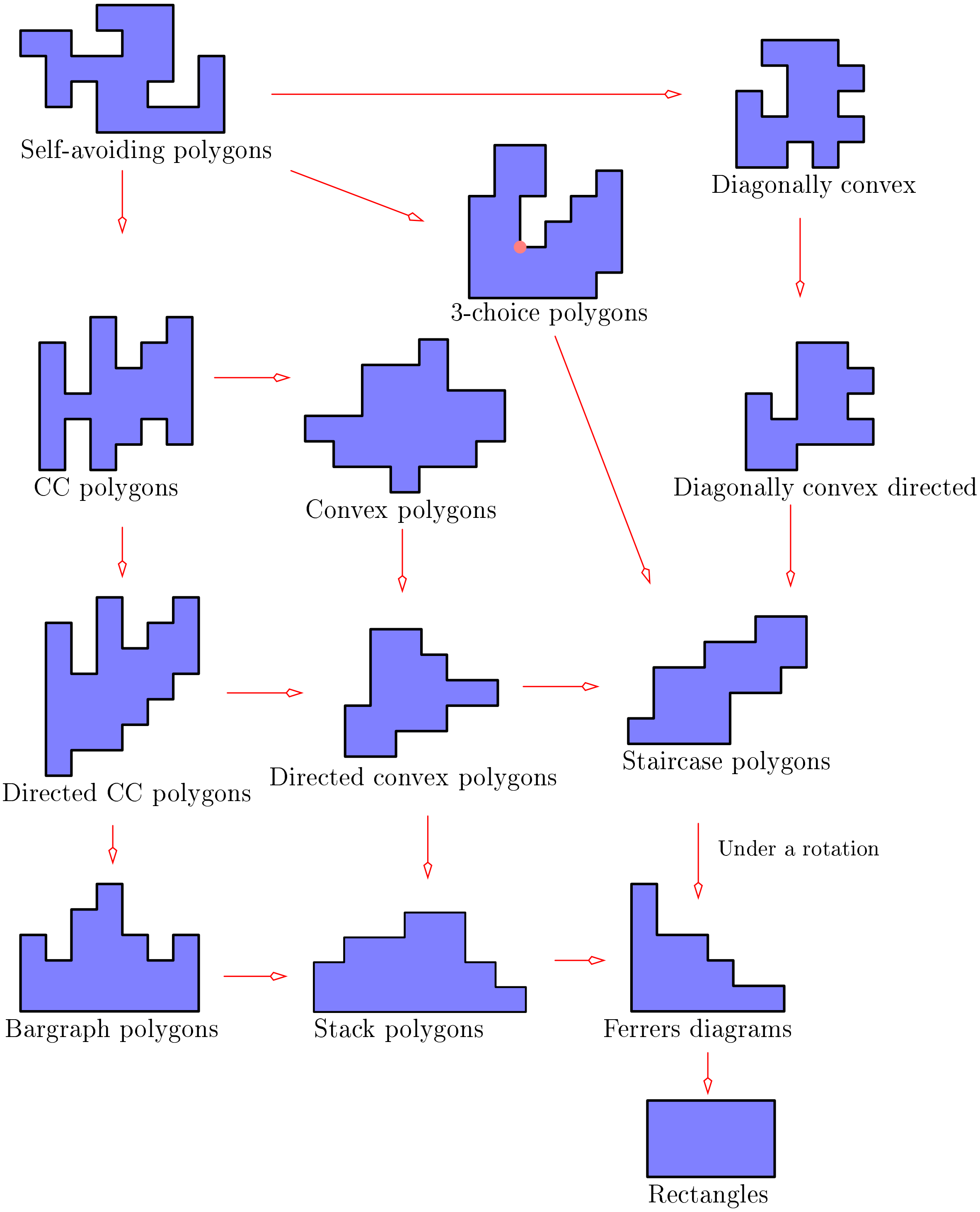



Figure 2.2: The polygon taxonomy. SAPs and diagonally convex polygons remain unsolved. All the other classes have been enumerated according to their area and perimeter (simultaneously). The point at which the perimeter of the 3-choice polygon violates the 3-choice walk rule (which is also the origin) is highlighted.

polygon and any vertical line is either empty or connected. *Row-convex polygons* are defined similarly.

- A polygon is ***directed column-convex*** if it is both column-convex, and every cell in the polygon can be reached from the *root* cell by a path taking only north and east steps. Equivalently it is a column-convex polygon whose lower side takes only north and east steps.

- A polygon is a ***bargraph polygon*** if it is both column-convex, and its lower side consists entirely of horizontal steps. *i.e.* it is a composition.

Enforcing convexity both horizontally and vertically gives *fully convex polygons*.

**Definition 2.4 (Column- & row-convex).**

- A ***convex polygon*** is both row-convex and column-convex. Equivalently its perimeter is equal to the perimeter of its minimum bounding rectangle.

- A ***directed convex polygon*** is both convex and every cell in the polygon can be reached from the *root* cell by a path taking only north and east steps.

- A ***stack-polygon*** is a convex polygon whose lower side consists entirely of horizontal steps.

- A ***staircase polygon*** is a convex polygon whose lower and upper sides are paths taking only north and east steps.

- A ***Ferrers diagram*** is a stack-polygon whose columns (from left to right) are non-increasing.

- A ***rectangle*** — this is reasonably obvious.

Another (more difficult) family of convex polygons are convex with respect to the leading diagonal (in the direction $\vec{v} = \vec{\imath} + \vec{\jmath}$):

**Definition 2.5.**

- A ***diagonally convex polygon*** is convex with respect to the direction $\vec{v} = \vec{\imath} + \vec{\jmath}$.

- A ***diagonally convex directed polygon*** is a diagonally convex polygon that is also directed.

Diagonally convex polygons have not been solved either by area or perimeter. Polygons that are both diagonally convex and directed, on the other hand, have been solved; the area and perimeter generating function is known.

### 2.1.2 Polygon models — solutions

Since polygons can be defined both as bond animals and as polyominoes, one can try to enumerate them according to perimeter and area separately, as well as perimeter and area *simultaneously.* We have listed in tables 2.1 and 2.2 the solutions by area, perimeter, and area and perimeter simultaneously. Multiple listing indicate effective simultaneous publication or that the method and / or the form of the solution are different.

A few remarks are in order:

- All these solved polygon models, with the exception of 3-choice polygons, have algebraic perimeter generating functions[3]. For example, the staircase polygon generating function, $S(x)$, is given by

$$S(x) = \frac{1}{2} - x^2 - \frac{1}{2}\sqrt{1-4x^2}, \tag{2.1}$$

  and so satisfies

$$x^4 + (2x^2-1)S(x) + S(x)^2 = 0. \tag{2.2}$$

  Rectangles, Ferrers diagrams and stacks actually have simple rational perimeter generating functions. All polygon models that have been enumerated according to their perimeter have also been enumerated anisotropically — according to their vertical and horizontal perimeter.

- The polygon models that are column-convex but not row-convex (bargraphs, directed column-convex and column-convex polygons) have rational area generating functions. This is somehow because the height of each column is independent of all the other columns. This independence is lost when row-convexity is also enforced, and the area generating functions of families of fully convex polygons are $q$-series, which have a more complicated analytic structure (see chapter 7). For example the bargraph area generating function, $B(q)$, is given by:

$$B(q) = \frac{q}{1-2q}, \tag{2.3}$$

  and so has a single pole at $q = \frac{1}{2}$. On the other hand, the area generating function of rectangles, $R(q)$, is given by:

$$R(q) = \sum_{n\geq 1} \frac{q^n}{1-q^n}, \tag{2.4}$$

  which has a natural boundary on $|q| = 1$ and so is not rational, algebraic or D-finite (see the proof of proposition 4.8 in chapter 4). Consequently it is far harder to "guess" the generating function of rectangles than that of bargraphs from the first few terms of their series expansions.

[3] *i.e.* if the generating function is denoted $f(x)$, then there exists some polynomial in two variables, $P(x_1, x_2)$, such that $P(f, x) = 0$.

| Model | Who and What |
|---|---|
| Rectangles | • **Perimeter & Area:** Obvious |
| Ferrers diagrams | • **Perimeter:** Obvious<br>• **Area:** Euler 1748 [59]<br>• **Perimeter & Area:** Prellberg & Owczarek 95 [134] |
| Stack polygons | • **Perimeter:** Obvious<br>• **Area:** Auluck 51 [5]<br>• **Perimeter & Area:** Prellberg & Owczarek 95 [134] |
| Staircase polygons | • **Perimeter:** Levine 59 [110]<br>Pólya 69 [132]<br>• **Area:** Klarner & Rivest 74 [108]<br>• **Perimeter & Area:** Brak & Guttmann 90 [34]<br>Lin & Tzeng 91 [115]<br>Bousquet-Mélou & Viennot 92 [32]<br>Prellberg & Brak 95 [133]<br>Fédou & Rouillon 95 [61]<br>Bousquet-Mélou 96 [23] |
| Directed convex polygons | • **Perimeter:** Lin & Chang 88 [114]<br>• **Perimeter & Area:** Bousquet-Mélou & Viennot 92 [32]<br>Bousquet-Mélou & Fédou 95 [26]<br>Bousquet-Mélou 96 [23] |
| Convex polygons | • **Perimeter:** Delest & Viennot 84 [53]<br>Kim 88 [103]<br>Lin & Chang 88 [114]<br>Gessel 90 [73]<br>• **Perimeter & Area:** Bousquet-Mélou 91 [18, 19, 21]<br>Lin 91[113]<br>Bousquet-Mélou & Fédou 95 [26]<br>Bousquet-Mélou 96 [23] |

Table 2.1: Solutions to convex polygons

| Model | Who and What |
|---|---|
| Bargraph polygons | • **Area:** Obvious<br>• **Perimeter:** Feretić 96 [64]<br>• **Perimeter & Area:** Owczarek & Prellberg 93 [128]<br>Prellberg & Brak 95 [133] |
| Directed column-convex polygons | • **Perimeter:** Delest & Dulucq 93 [49]<br>Feretić 93 [62]<br>Joyce & Guttmann 94 [102]<br>• **Area:** Moser, Klarner 65 [104]<br>• **Perimeter & Area:** Prellberg & Brak 95 [133]<br>Bousquet-Mélou 96 [23]<br>Feretić 2000 [67] |
| Column-convex polygons | • **Perimeter:** Delest 88 [50]<br>Brak, Guttmann & Enting 90 [35]<br>Lin 90 [112]<br>Feretić 96 [64]<br>• **Perimeter on other lattices:** (Hexagonal) Lin & Wu 90 [116]<br>(Hexagonal) Feretić & Svrtan 93 [68]<br>(Checkerboard) Tzeng & Lin 91 [152]<br>• **Area:** Temperley 56 [151]<br>Klarner 65 [104]<br>Pólya 69 [132]<br>• **Area on other lattices:** (Hexagonal) Klarner 67 [105]<br>(Triangular & hypercubic) Forgacs & Privman 87 [71]<br>(Checkerboard) Feretić 95 [63]<br>• **Perimeter & Area:** Brak & Guttmann 90 [34]<br>Lin & Tzeng 91 [115]<br>Feretić & Svrtan 93[68]<br>Bousquet-Mélou 96 [23]<br>• **Perimeter and Area on other lattices:** (Hexagonal) Feretić & Svrtan 93 [68] |
| 3-choice polygons | • **Perimeter & Area:** Delest, Conway & Guttmann 97 [42] (solution is polynomial-time algorithm) |
| Diagonally convex directed polygons | • **Perimeter:** Delest & Fédou 89 [51]<br>Penaud 90 [130]<br>• **Area:** Privman & Švrakić 88 [135]<br>• **Perimeter & Area:** Bousquet-Mélou 96 [24]<br>Feretić 99 [66] |

Table 2.2: Solutions to column-convex polygons, 3-choice polygons and diagonally convex directed polygons.

- The combined area and perimeter generating functions for solved models are $q$-series[4]. This is even true in the cases of bargraphs, column-convex, and directed column-convex polygons whose area generating functions are rational and perimeter generating functions are algebraic — the combined perimeter and area generating functions are considerably more complicated.

- The techniques that work in two dimensions generally do not continue to work in higher dimensions, and consequently there are not very many higher dimensional results; we list them below:

  – Plane partitions by volume[5] — MacMahon [120].

  – Staircase polygons by perimeter on hypercubic lattices — Guttmann and Prellberg 93 [85].

  – Higher dimensional column-convex polygons by area — Forgacs & Privman 87 [71].

  – Three dimensional (fully) convex polygons by perimeter — Bousquet-Mélou & Guttmann 97 [27].

In this thesis we do not add any new classes of polygons to tables 2.1 and 2.2. Our addition to the world of solved polygon models is the enumeration of bargraph polygons according to their site-perimeter (in chapter 6). All of the classes of polygons in the above tables have been enumerated according to their area and (anisotropic) perimeter, but only families of fully convex polygons (rectangles, Ferrers diagrams, stacks, staircase polygons, directed convex polygons and convex polygons) have been enumerated according to their site-perimeter. For families of convex polygons the site-perimeter can be enumerated using the techniques used for ordinary perimeter, but this approach breaks down for families of column-convex polygons. In order to take the site-perimeter into account, we use a variation of the column-by-column construction (which we describe in chapter 5).

Even though we do not add many entries to these tables, we will make use of the polygons we have defined above. They will arise as examples of applications of the haruspicy techniques in chapter 3, and also as examples of reciprocity and inversion relations in chapter 8. In chapter 5 we will discuss column-by-column constructions by which all families of column-convex polygons can be enumerated.

[4] Though the area and perimeter generating function of 3-choice polygons is not known in closed form, it may be possible to show that it is not D-finite using the techniques described in chapters 3 and 7.

[5] Plane partitions are a generalisation of Ferrers diagrams to three dimensions — they are arrangements of columns of cubes in the first quadrant of the plane, such that the heights of the columns are non-increasing in the $+x$ and $+y$ directions.

# Part I

# Techniques and exact solutions

# CHAPTER 3

# Haruspicy and anisotropic generating functions

**Definition 3.1 (From [39]).**
**haruspicy:** /həˈrʌspɪsi/ Divination by inspection of the entrails of sacrifices.

**Question 10.** *Why "haruspicy?"*

**Answer.** *Because we will examine the properties of anisotropic generating functions (some of which are as yet unsolved) by examining the configuration of bonds within some specially prepared bond animals.*

3.1

# Properties of anisotropic generating functions

The enumeration of polyominoes and lattice animals is arguably one of the most famous problems in combinatorics and considering the intensive study that these models have been subjected to over their 40+ year history, it is perhaps a little surprising that the number of rigorous results concerning these models is very small, and that the number of models without severe topological restrictions that have been solved exactly (either implicitly or explicitly) is smaller still. In this chapter we will only consider bond animals on the square lattice; we will write "animal" to mean "square lattice bond animal".

Ideally we would like to enumerate a family of bond animals, such as self-avoiding polygons, or directed bond animals enumerated according to the number of bonds contained. As we noted in chapter 1, the history of animals does not instill us with great hope of being able to enumerate a large family of animals; to date, only those models with severe topological restrictions have been solved (see chapters 1 and 2). Rather than commencing an almost certainly doomed attempt to find the solution of a less restricted family of animals we will attempt to determine some of the *properties* of the solution by examining their *anisotropic* generating function.

## 3.1.1 Anisotropic generating functions

The *isotropic* generating function of a family of animals enumerates the animals according to the total number of bonds. The *anisotropic* generating function, on the other hand, distinguishes between horizontal and vertical bonds. For a given bond animal $B$, we denote the number of horizontal (resp. vertical) bonds it contains by $|B|_{\Leftrightarrow}$ (resp. $|B|_{\Updownarrow}$).

Let $\mathcal{G}$ be a set of bond animals on the square lattice. Let us enumerate the elements of $\mathcal{G}$ according to the number of horizontal and vertical bonds and form the *anisotropic* generating function:

$$\begin{aligned} gf(\mathcal{G}) &= \sum_{Q\in\mathcal{G}} x^{|Q|_{\Leftrightarrow}} y^{|Q|_{\Updownarrow}} \\ &= \sum_{n,m=0}^{\infty} c_{n,m}\, x^n y^m, \end{aligned}$$

where $c_{n,m}$ is the number of elements of $\mathcal{G}$ containing exactly $n$ horizontal bonds and $m$ vertical bonds.

If the animal is a polygon, then the numbers of vertical and horizontal bonds are always even numbers, and so rather than introducing extraneous factors of 2 we will enumerate families of polygons according to their horizontal and vertical *half-perimeters* (being exactly half the number of horizontal and vertical bonds). Rather than defining different notation for polygons we will simply take $|B|_{\Leftrightarrow}$ and $|B|_{\Updownarrow}$ to mean the horizontal and vertical half-perimeters (respectively) of any polygon $B$.

Let us start by summing this generating function to make it a power series in $y$ with coefficients that are power series in $x$. Writing $\mathcal{G}_n = \{B \in \mathcal{G} : |B|_{\Updownarrow} = n\}$ we have

$$gf(\mathcal{G})(x,y) = \sum_{n\geq 1} y^n \sum_{Q\in\mathcal{G}_n} x^{|Q|_{\Leftrightarrow}} = \sum_{n\geq 1} H_n(x) y^n.$$

The coefficient of $y^n$ in the above generating function, $H_n(x)$, is a power series in $x$ that enumerates all bond animals in $\mathcal{G}$ containing $n$ vertical bonds (the set $\mathcal{G}_n$), according to the number of horizontal bonds.

Very generally speaking it is not too difficult to extend isotropic techniques (be they exact solutions or numerical expansions) to the anisotropic case, so one can obtain the anisotropic generating function without having to do much more work. The anisotropic generating function is somehow a more manageable object than the isotropic. Splitting the set of animals $\mathcal{G}$, into separate simpler subsets, $\mathcal{G}_n$, gives us smaller pieces, each of which is easier to study than the whole.

As noted above, the only families of bond animals that have been solved are those with severe topological restrictions. If one seeks to understand the *isotropic* generating function then one must somehow examine *all* possible topologies or configurations[1] that can occur in $\mathcal{G}$. On the other hand, if we examine the generating function of $\mathcal{G}_n$, then the number of different topologies that can occur is always finite. For example, consider self-avoiding polygons with $2n$ vertical bonds. If $n = 1$ all configurations are rectangles. If $n = 2$, then all configurations are vertically *and* horizontally convex, while if $n = 3$ all configurations are vertically *or* horizontally convex. The anisotropy allows one to study the effect that these configurations have on the generating function in a more controlled manner.

Similarly, instead of trying to study the properties of the whole generating function (that may not be known), the anisotropy breaks the generating function into separate simpler pieces, $H_n(x)$, that can be calculated exactly. By studying the properties of these coefficients we can obtain some idea of the properties of the generating function as a whole. This procedure has been proposed [83] as a test of the solvability of a model (see below), and has been applied to a wide variety of models [82].

It has also been observed [28] that the anisotropic generating functions of many (solved and unsolved) families of animals obey a type of functional symmetry known as *reciprocity* or *inversion relations*. These are well known in statistical mechanics and in certain cases can lead to an exact solution. These symmetries are not exhibited in the isotropic generating function. We discuss these properties in chapter 8.

### 3.1.2 A definition and some examples

In the work that follows we will show that the generating functions, $H_n(x)$, can be expressed in terms of generating functions of the form $\frac{x^k}{1-x^k}$. The polynomial $(1 - x^k)$ can be factored into *cyclotomic polynomials*.

[1] We deliberately use imprecise words here. We will be more precise below.

**Definition 3.2.**
The cyclotomic polynomials, $\Psi_k(x)$, are the factors of the polynomials $(1-x^n)$. They are defined by $(1-x^n) = \prod_{k|n} \Psi_k(x)$. Hence for any given integers $\{c_i\}$, we have the following factorisation

$$\prod_{n\geq 1}(1-x^n)^{c_n} = \prod_{n\geq 1}\prod_{k|n} \Psi_k(x)^{c_n} = \prod_{k\geq 1} \Psi_k(x)^{\sum_{d\geq 1} c_{kd}}$$

The first few cyclotomic polynomials are below in bold.

$$\begin{array}{lll}
\Psi_1: & (1-x) = & \mathbf{(1-x)} \\
\Psi_2: & (1-x^2) = & (1-x)\mathbf{(1+x)} \\
\Psi_3: & (1-x^3) = & (1-x)\mathbf{(1+x+x^2)} \\
\Psi_4: & (1-x^4) = & (1-x)(1+x)\mathbf{(1+x^2)} \\
\Psi_5: & (1-x^5) = & (1-x)\mathbf{(1+x+x^2+x^3+x^4)} \\
\Psi_6: & (1-x^6) = & (1-x)(1+x)(1+x+x^2)\mathbf{(1-x+x^2)}
\end{array}$$

Note that $(1-x^6) = \Psi_1\Psi_2\Psi_3\Psi_6$, since $1, 2, 3$ and $6$ all divide 6. We call $\Psi_k(x)$ the $k$th cyclotomic polynomial, and say that its *order* is $k$.

Let us consider three examples of these anisotropic generating functions; staircase polygons, 3-choice polygons and self-avoiding polygons.

**Example 3.1 (Staircase polygons).**
The anisotropic generating function of staircase polygons is known in closed form [110]:

$$P(x,y) = \frac{1}{2}\left(1-x-y-\sqrt{(1-x-y)^2-4xy}\right).$$

Expanding $P(x,y)$ as a power series in $y$ gives

$$\begin{aligned} P(x,y) &= \frac{x}{1-x}y + \frac{x}{(1-x)^3}y^2 + \frac{x(1+x)}{(1-x)^5}y^3 \\ &\quad + \frac{x(1+3x+x^2)}{(1-x)^7}y^4 + \frac{x(1+6x+6x^2+x^3)}{(1-x)^9}y^5 + \cdots \end{aligned}$$

The coefficients, $H_n(x)$, have the following properties:

- $H_n(x)$ is a rational function of $x$,
- the degree of the numerator of $H_n(x)$ is $(n-1)$, for all $n \geq 2$,
- the denominator of $H_n(x)$ is $(1-x)^{2n-1}$, and
- the coefficients of the numerators are positive, symmetric and unimodal.

All other polygon models for which a closed form solution is known (all of which are subsets of column-convex polygons) have similar properties (though the symmetry of numerator coefficients is lost in some cases).

**Example 3.2 (3-choice polygons).**
Though the anisotropic generating function of 3-choice polygons, $P(x, y)$, is not known in closed form (nor is the isotropic generating function), it has been "solved" in the sense that there exists a polynomial time (and space) algorithm that calculates its coefficients [42].

Resumming $P(x, y)$ as a power series in $y$, we find that the coefficients $H_n(x)$ have the following properties:

- $H_n(x)$ is a rational function of $x$,
- the degree of the numerator of $H_n(x)$ is less than the degree of the denominator,
- the denominator of $H_n(x)$ has been shown [16] to be

$$\begin{array}{ll} (1-x)^{2n-1}(1+x)^{2n-7}, & n \text{ even} \\ (1-x)^{2n-1}(1+x)^{2n-8}, & n \text{ odd} \end{array}$$

  and hence contains only the first two cyclotomic factors, and
- the coefficients of the numerators are positive and unimodal, but not symmetric.

**Example 3.3 (Self-avoiding polygons).**
The anisotropic generating function of self-avoiding polygons, $P(x, y)$, remains elusive, and to date the best algorithm for computing the coefficients of $P$ is the finite lattice method, which still requires exponential time [57], but is exponentially more efficient than direct enumeration.

Expanding $P(x, y)$ as a power series in $y$, one finds [98] that the coefficients (which have been computed up to order 14), $H_n(x)$ have the following properties:

- $H_n(x)$ is a rational function of $x$,
- the degree of the numerator of $H_n(x)$ is equal to the degree of its denominator,
- the coefficients of the numerators are positive and unimodal, but not symmetric,

•if we write the denominator of $H_n(x)$ as $D_n(x)$, then the first ten are:

$$\begin{aligned}
D_1(x) &= (1-x) \\
D_2(x) &= (1-x)^3 \\
D_3(x) &= (1-x)^5 \\
D_4(x) &= (1-x)^7 \\
D_5(x) &= (1-x)^9(1+x)^2 \\
D_6(x) &= (1-x)^{11}(1+x)^4 \\
D_7(x) &= (1-x)^{13}(1+x)^6(1+x+x^2) \\
D_8(x) &= (1-x)^{15}(1+x)^8(1+x+x^2)^3 \\
D_9(x) &= (1-x)^{17}(1+x)^{10}(1+x+x^2)^5 \\
D_{10}(x) &= (1-x)^{19}(1+x)^{12}(1+x+x^2)^7(1+x^2)
\end{aligned}$$

This suggests that a new cyclotomic factor enters every third coefficient, and that it enters with exponent 1, and increases by 2 in each subsequent coefficient (with the exception of $(1+x)$ which has exponent 1 less than this pattern predicts).

The denominator structure of self-avoiding polygons is starkly different to that of the previous two (solvable) examples. The denominators of staircase polygons contain only a single cyclotomic factor, $(1-x)$, while those of 3-choice polygons contain only $(1-x)$ and $(1+x)$, so $H_n(x)$ contains only a finite number of poles as $n \to \infty$. On the other hand, if one extrapolates from the observed pattern of self-avoiding polygon denominators, then any cyclotomic factor will eventually appear, and so $H_n(x)$ has a finite set of set of poles that becomes dense on $|x| = 1$ as $n \to \infty$. This has strong (pessimistic) implications for the analytic nature and solvability of the self-avoiding polygon generating function (which we will discuss below).

# 3.2

# Haruspicy, squashing animals and partial orders

Consider the set of self-avoiding polygons that contain 2 vertical bonds (see figure 3.1) — this is simply the set of all rectangles of height 1. The horizontal half-perimeter generating function of these polygons is clearly $\sum_{n\geq 1} x^n = \frac{x}{1-x}$.

The smallest polygon (or the minimal polygon) in this set is the unit square. We can then obtain the other polygons from the unit square by "stretching" or "growing" the horizontal bonds (see figure 3.2). The unit square has generating function simply given by $x$, stretching the horizontal bonds to length $n$ gives an $n$ by 1 rectangle that contributes $x^n$ to the generating function. Summing over all possible "stretches" gives $\sum_{n\geq 1} x^n = \frac{x}{1-x}$ as required.

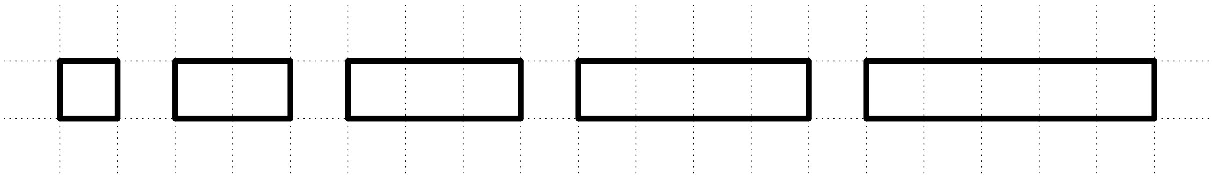


Figure 3.1: Self-avoiding polygons containing exactly 2 vertical bonds.

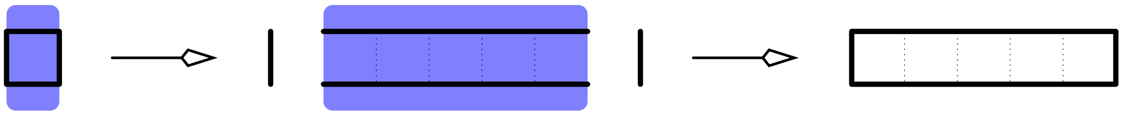


Figure 3.2: Stretching or growing the horizontal bonds of the unit square will give any self-avoiding polygon with 2 vertical bonds.

By reversing the stretching process, we can think of squashing the rectangles into shorter and shorter rectangles until we reach the unit square. This squashing process gives a (total) order on this set. The smallest element of this set under this order is the unit square. This idea can be extended to other animals, and we will introduce two different ways of "squashing" general animals. By examining the contents of these "squashed" animals and "stretching" them we can determine certain properties of anisotropic generating functions (hence "haruspicy").

### 3.2.1 Columns, sections and partial orders

**Definition 3.3.**
We will define a *column* of a given animal to be the horizontal bonds within a single horizontal lattice spacing of the animal. See figure 3.3. If the column contains $k$ horizontal bonds we say it is a $k$-column. The number of $k$-columns in an animal, $A$, is denoted by $\gamma_k(A)$.

**Definition 3.4.**
We construct the *section lines* of an animal in the following way. Draw horizontal lines from the extreme left and the extreme right of the lattice towards the animal so that the lines run through the middle of each lattice cell. The lines are terminated when they first touch a vertical bond (see figure 3.4).

Cut the lattice along each section line from infinity until it terminates at a vertical bond. Then from this vertical bond cut vertically in both directions until another section line is reached. In this way the animal is split into *pages* (see figure 3.4); we consider the vertical bonds along these vertical cuts to lie *between* pages, while the other vertical bonds lie *within* the pages.

We call a *section* the set of horizontal bonds within a single column of a given page. Equivalently, it is the set of horizontal bonds of a column of an animal between two neighbouring section lines. A section with $k$ horizontal bonds is a $k$-section. The number of $k$-sections in an animal, $A$, is denoted by $\sigma_k(A)$.

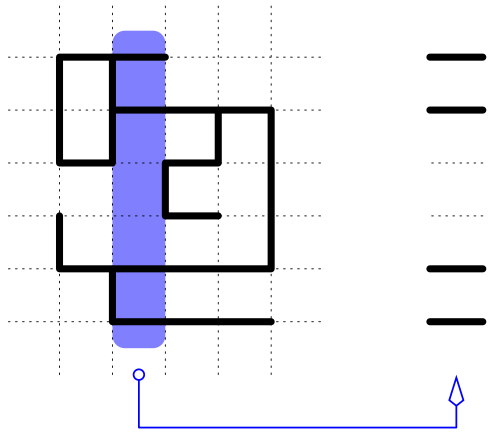


Figure 3.3: A 4-column of a bond animal.

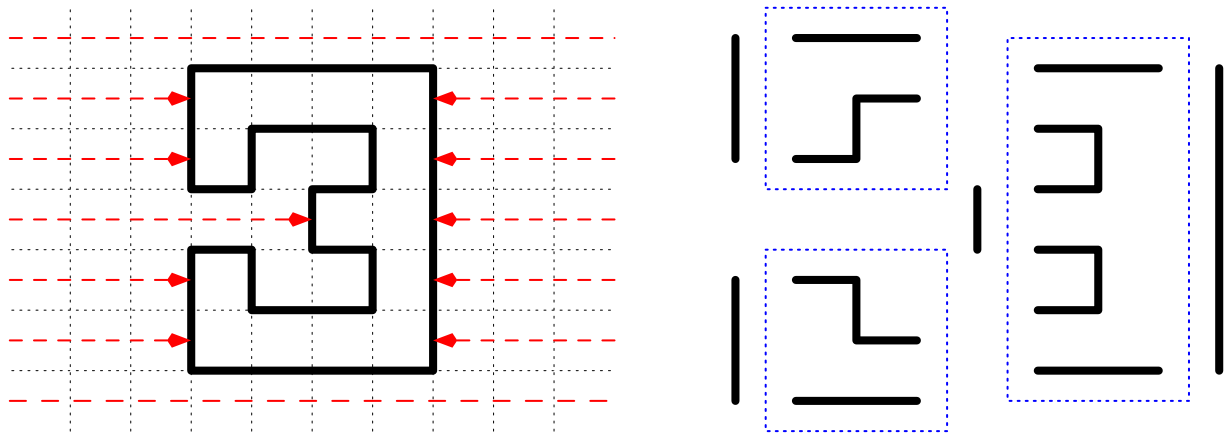


Figure 3.4: *Section lines* (the heavy dashed lines) split the animal (in this example it is a polygon) into *pages*. Each column in a page is a *section*. This polygon is split into 3 pages, each containing 2 sections. 10 vertical bonds lie between pages and 4 vertical bonds lie within the pages.

**Definition 3.5.**
We say that a column is a *duplicate column* if the column immediately on its left (without loss of generality) is identical (see figure 3.5) and there are no vertical bonds between them. We similarly define a *duplicate section*.

One can squash or reduce animals by *deletion* of duplicate columns by slicing the animal on either side of the duplicate column, removing the column and recombining the animal, as illustrated in figure 3.5. By reversing the column deletion process we define *duplication* of a column. We define *section-deletion* and *section-duplication* in an analogous manner.

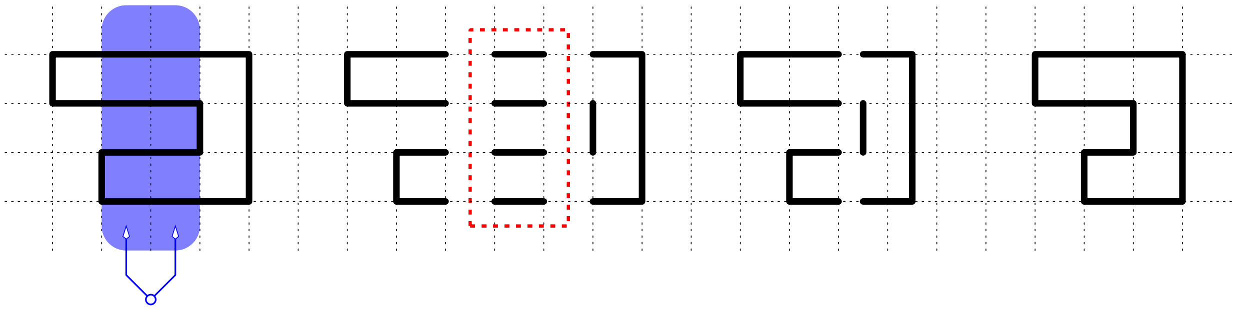


Figure 3.5: Surgery. The process of column deletion. The two indicated columns are identical. Slice either side of the duplicate and separate the polygon into three pieces. The middle piece, being the duplicate, is removed and the remainder of the polygon is recombined. Reversing the steps leads to column duplication.

Let us take an animal, $B$, and delete one (or more) of its columns or sections to obtain another animal, $A$. We define two partial orders $\preceq_c$ and $\preceq_s$ on the set $\mathcal{G}_n$ of animals with $n$ vertical bonds.

**Definition 3.6.**
For any two animals $P, Q \in \mathcal{G}_n$, we define the binary relations $\preceq_c$ and $\preceq_s$ by stating that:

- $P \preceq_c Q$ if $P = Q$ or $P$ can be obtained from $Q$ by a sequence of column deletions and
- $P \preceq_s Q$ if $P = Q$ or $P$ can be obtained from $Q$ by a sequence of section deletions.

See figure 3.6 for example.

From this definition we immediately obtain the following lemma

**Lemma 3.1.** *The binary relations $\preceq_c$ and $\preceq_s$ are partial orders on the set of animals.*

*Proof.* Let $A$, $B$ and $C$ be animals. A partial order must be reflexive, anti-symmetric and transitive. We state the proof for $\preceq_c$ — the proof for $\preceq_s$ is identical.

- **reflexive** — by definition $A \preceq_c A$.
- **anti-symmetric** — If $A \preceq_c B$, then either $A = B$ or $|A|_\Leftrightarrow < |B|_\Leftrightarrow$. Similarly if $B \preceq_c A$ then either $A = B$ or $|A|_\Leftrightarrow > |B|_\Leftrightarrow$. Hence if $A \preceq_c B$ and $B \preceq_c A$ then $A = B$.

- **transitive** — If $A \preceq_c B$ then there exists a sequence of column-deletions that takes $B$ to $A$. Similarly if $B \preceq_c C$, then there exists another sequence of column-deletions that takes $C$ to $B$. Concatenating these gives a sequence of column-deletions that takes $C$ to $A$, and hence $A \preceq_c C$.

■

### 3.2.2 Minimal animals and equivalence relations

If we take an animal and start to remove duplicate columns (or sections), then we cannot reduce the animal to nothing. At some point we must reach an animal that contains no duplicate column. This animal we call a column-minimal animal. A little more formally we may write:

**Definition 3.7.**
A *column-minimal animal*, $A$, is an animal such that for all animals $B$ satisfying $B \preceq_c A$, then $B = A$. *i.e.* $A$ cannot be squashed any further. We define a *section-minimal animal* in a similar way.

If we reduce some animal $A$ by a sequence of column deletions to a column-minimal animal $B$, and also by a sequence of section-deletions to a section-minimal animal $C$, are $B$ and $C$ the same?

**Lemma 3.2.** *If an animal, $P$, is section-minimal then it is also column-minimal. The converse is false.*

*Proof.* If an animal has duplicate columns it must have duplicate sections, so if an animal does not have a duplicate section it cannot have a duplicate column (*modus tollens*). On the other hand, one can readily construct a column-minimal animal that has duplicate sections (see figure 3.6 for example). ■

Consequently if $A$ reduces to $B$ by column deletions and to $C$ by section deletions, we will always have $C \preceq_s B$, but not always $B = C$. Section-deletion is a stronger reduction.

◁ ◁ ⋄ ▷ ▷

Let us take some animal $C$ and squash it down to a column-minimal animal $A$, by some sequence of column deletions. It is natural to ask whether changing the order of the column-deletions changes the column-minimal animal that we will reach. Fortunately each animal $C$ reduces to a *unique* column-minimal animal $A$, and also to a unique section-minimal animal $B$. However $A$ and $B$ are not always the same animal.

**Lemma 3.3.** *Every animal $C$ reduces by column-deletions to a unique column-minimal animal. Similarly every animal reduces by section-deletions to a unique section-minimal animal. The column-minimal animal and section-minimal animal reached from $C$ need not be the same.*

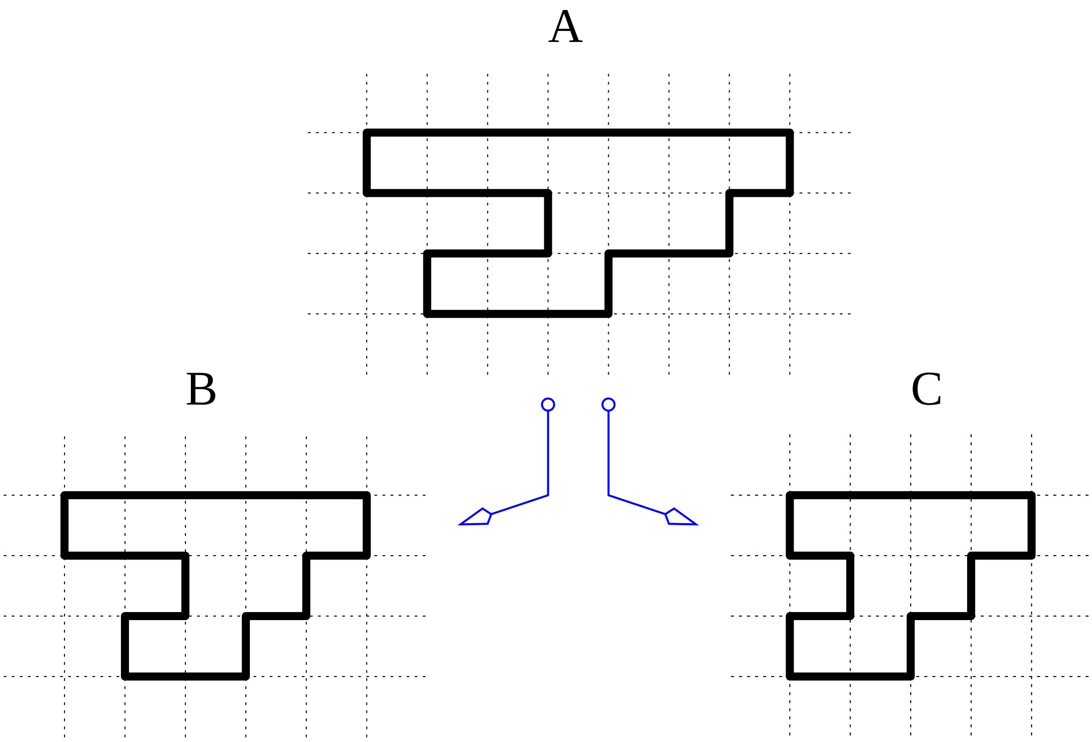



Figure 3.6: Polygon $A$ is reduced by a sequence of column deletions to polygon $B$ (which is column-minimal) and a sequence of section deletions to polygon $C$ (which is section-minimal). $B$ can be reduced to $C$ by section deletions, and hence is not minimal with respect to $\preceq_s$.

*Proof.* The columns of any animal, $C$, can be encoded (from left to right) as a sequence of columns, $(c_1^{\alpha_1}, c_2^{\alpha_2}, \ldots, c_j^{\alpha_j})$, where $c_i^{\alpha_i}$ indicates $\alpha_i$ repetitions of the column $c_i$. Enforcing the additional constraint that $c_i \neq c_{i+1}$ will ensure the uniqueness of the $\alpha_i$. Removing all duplicate columns will reduce $C$ to some animal $A$, that is encoded by the sequence $(c_1^1, c_2^1, \ldots, c_j^1)$. Clearly this is unique.

To prove the same result for section-deletion, one considers the sections within each given page of the animal.

To see that the section-minimal animal and the column-minimal animal reached from $C$ are not necessarily the same animal, consider the animal in figure 3.6. ∎

◁ ◁ ◇ ▷ ▷

Since every animal reduces to a unique minimal element by column deletion (or section deletion), the set of animals can be written as the disjoint union of posets, each of which contains a single minimal animal. Using this idea we can construct two equivalence relations on the set of animals:

**Definition 3.8.**
We say that two animals, $A$ and $B$, are *column-equivalent* if both $A$ and $B$ reduce to the same column-minimal animal. In this case we write $A \approx_c B$. Similarly we say that two animals, $A$ and $B$, are *section-equivalent* if both $A$ and $B$ reduce to the same section-minimal animal. In this case we write $A \approx_s B$.

See figure 3.7.

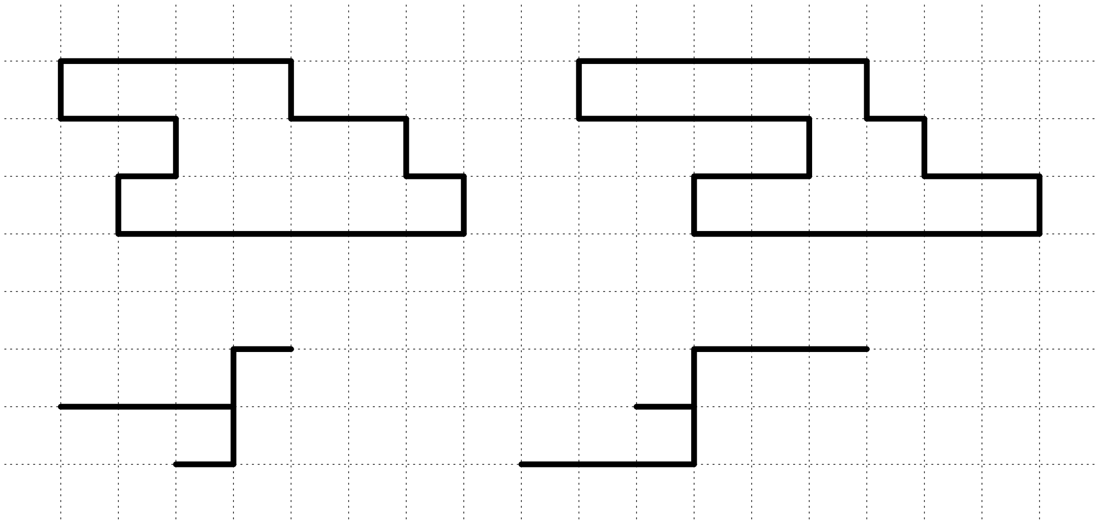


Figure 3.7: The top two animals are column-equivalent (and so also section-equivalent), while the bottom two are section-equivalent but *not* column-equivalent.

**Lemma 3.4.** *Column-equivalence and section-equivalence are equivalence relations.*

*Proof.* An equivalence relation must be reflexive, symmetric and transitive.

- **reflexive** Clearly $A$ reduces to the same minimal element as itself, so $A \approx_c A$.
- **symmetric** If $A \approx_c B$, then $A$ and $B$ reduce to the same minimal animal and clearly $B \approx_c A$.
- **transitive** If $A \approx_c B$, then $A$ and $B$ reduce to the same minimal animal. If $B \approx_c C$ then $B$ and $C$ reduce to the same minimal animal. Since each animal reduces to a unique minimal animal (by column or section deletions) $A$ and $C$ must reduce to the same animal, and hence $A \approx_c C$.

**Definition 3.9.**
Since column-equivalence and section-equivalence are equivalence relations, we can partition the set of animals into equivalence classes each of which can be characterised by the column-minimal (or section-minimal) animal within the class. We refer to the equivalence class of a column-minimal (resp. section-minimal) animal, $A$, as the *column-expansion* (resp. *section-expansion* ) of $A$. We write:

$$\begin{aligned} \mathcal{E}_c(A) &= \{B \in \mathcal{G} \mid A \preceq_c B\}, & (3.1) \\ \mathcal{E}_s(A) &= \{B \in \mathcal{G} \mid A \preceq_s B\}. & (3.2) \end{aligned}$$

We write the horizontal bond generating function of the expansion of a minimal element, $A$, as

$$\begin{aligned} G_c(A) &= \sum_{B \in \mathcal{E}_c(A)} x^{|B|_{\Leftrightarrow}} \qquad \text{if } A \text{ is column-minimal,} & (3.3) \\ G_s(A) &= \sum_{B \in \mathcal{E}_s(A)} x^{|B|_{\Leftrightarrow}} \qquad \text{if } A \text{ is section-minimal.} & (3.4) \end{aligned}$$

Since $\mathcal{G}_n$ is partitioned into equivalence classes, its generating function, $H_n(x)$, can be written as the sum of the generating function of each equivalence class.

**Lemma 3.5.** *Let $\mathcal{M}_c$ and $\mathcal{M}_s$ be the sets of column-minimal animals and section-minimal animals (respectively) of $\mathcal{G}_n$, then*

$$\begin{aligned} H_n(x) &= \textstyle\sum_{B \in \mathcal{G}_n} x^{|B|_{\Leftrightarrow}} = \displaystyle\sum_{A \in \mathcal{M}_c} G_c(A) \\ &= \sum_{A \in \mathcal{M}_s} G_s(A) \end{aligned}$$

*Proof.* This follows directly from the fact that each animal in $\mathcal{G}_n$ is an element of the expansion of exactly one minimal animal. Since the expansions are disjoint the result follows. ∎

◁ ◁ ◇ ▷ ▷

Let us consider a set of bond animals, $\mathcal{G}_n$, the elements of which contain exactly $n$ vertical bonds. How many equivalence classes (or minimal animals) are there in $\mathcal{G}_n$? The exact number depends upon the exact family of animals. We can show that the number of minimal bond animals with $n$ vertical bonds is finite.

**Lemma 3.6.** *If $\mathcal{G}_n$ is a set of animals with $n$ vertical bonds, then the set of minimal elements in $\mathcal{G}_n$ (w.r.t. either partial order) is finite.*

*Proof.* Due to lemma 3.2 every section-minimal animal is column minimal, so it suffices to prove the above lemma for column-minimal animals. We first show that all column-minimal animals with $n$ vertical bonds have finite height and width.

Let $P$ be a column-minimal animal in $\mathcal{G}_n$. Obviously $P$ cannot contain more than $n$ rows. Since there are no duplicate columns in $P$, between each pair of columns of $P$ there must either be a vertical bond, or one horizontal bond must terminate — leaving a vertex of degree 1. See figure 3.8.

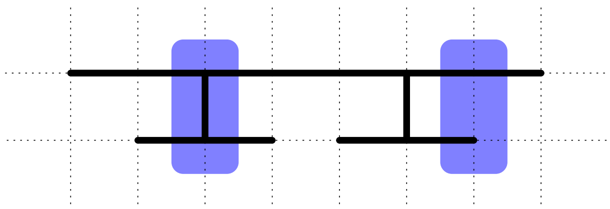


Figure 3.8: Between consecutive columns there must be either a vertical bond or a vertex of degree 1.

Let us bound the number of vertices of degree 1. If there are no vertical bonds, then there can be at most 2 vertices of degree 1. Each vertical bond can be attached to at most 4 lines of horizontal bonds. Hence there can be at most 4 vertices of degree 1 connected to each vertical bond (by lines of horizontal bonds). Consequently $P$ can contain at most $4n+2$ vertices of degree 1. Between each pair of columns there must be either one of these vertices or a vertical bond, so the number of columns in $P$ is bounded by $5n+1$.

Hence every minimal animal in $\mathcal{G}_n$ fits inside a box of height $n$ and width $5n+1$. Since there are only a finite number of bonds inside this box there can be only a finite number of column-minimal animals. ∎

## 3.3

# Dense families of animals and generating functions

Consider again the set of all self-avoiding polygons containing 2 vertical bonds. Clearly the column- and section-minimal animal is the unit square. The equivalence class of this animal is the original set, and so has generating function $\frac{x}{1-x}$.

If we now consider the set of polygons having only 2 vertical bonds and odd horizontal half-perimeter, the minimal element is the same, but now the generating function is $\frac{x}{1-x^2}$. Worse still is the subset of polygons which have 2 vertical bonds and *prime* horizontal half-perimeter; it still has the same minimal element, but has a very nasty generating function.

So rather than have to deal with such a possible variety of generating functions for the expansion of same minimal element, we will restrict ourselves to *dense* families of animals.

### 3.3.1 Dense families of animals

**Definition 3.10.**
A set of animals, $\mathcal{G}$ is *dense* if:

- for every column-minimal animal, $A \in \mathcal{G}$, all the elements of the column expansion of $A$ are contained in $\mathcal{G}$.
- for every section-minimal animal, $A \in \mathcal{G}$, all the elements of the section expansion of $A$ are contained in $\mathcal{G}$.

Equivalently a set of animals, $\mathcal{G}$, is *dense* if for every column-minimal (resp. section-minimal) animal, $A \in \mathcal{G}$ every column (resp. section) can be duplicated any number of times *independently* of every other column (resp. section). Hence if $A$ is encoded by a sequence of columns (or sections) $(c_1, \ldots, c_j)$ and for every $\alpha = (\alpha_1, \ldots, \alpha_j) \in \mathbb{Z}^{+j}$, there exists an animal $\in \mathcal{G}$ encoded by the sequence of columns (or sections) $(c_1^{\alpha_1}, \ldots, c_j^{\alpha_j})$.

Most families of animals that are studied on the square lattice are dense. Self-avoiding polygons on the *hexagonal lattice* are often considered (particularly for the purpose of computer aided enumeration) as polygons on the square lattice with the additional restriction that vertical bonds can only be placed according to a brick-work pattern — *i.e.* every second vertical edge (along both horizontal and vertical lines) is disallowed. Removing a duplicate column from such a polygon gives a polygon that violates the brick-work rule[2]. See figure 3.9.

### 3.3.2 Generating functions of dense families of animals

Now that we are restricting ourselves to dense families of animals, we find that the generating function of an equivalence class has a simple form.

**Lemma 3.7.** *If $P$ is a column-minimal (resp. section-minimal) animal in a* dense *family of animals then its expansion has the following generating function:*

$$G_c(P) = \prod_k \left(\frac{x^k}{1-x^k}\right)^{\gamma_k(P)} \tag{3.5}$$

$$\left( resp.\ G_s(P) = \prod_k \left(\frac{x^k}{1-x^k}\right)^{\sigma_k(P)} \right) \tag{3.6}$$

i.e. *equivalence classes have nice generating functions.*

[2]Constructing a dense family of animals on the brick-work lattice, that is making the theory of this chapter work for animals on the brick-work lattice, will be the subject of future work. It can probably be done simply by requiring duplicate sections and columns be removed in pairs.

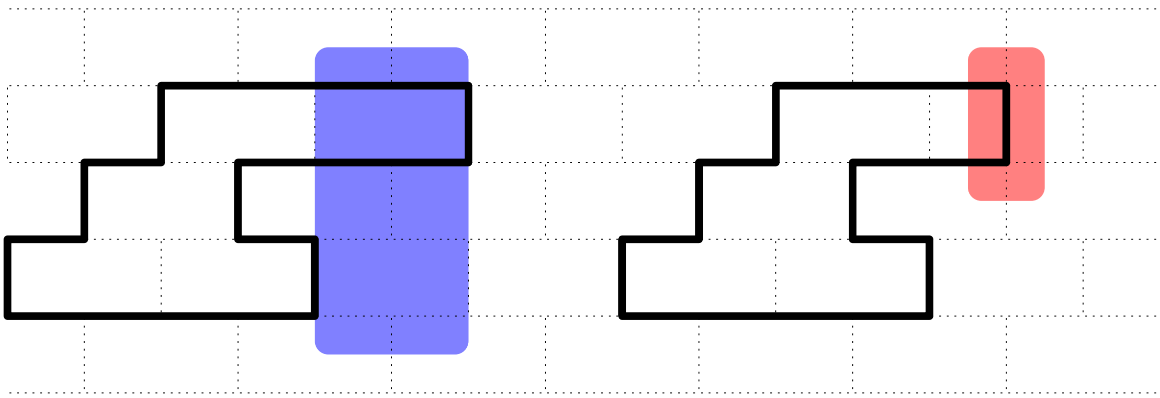


Figure 3.9: The highlighted columns in the left hand polygon are duplicates. Removing one of the duplicate columns results in the polygon on the right which has a vertical bond where there is no edge on the brick-work lattice.

*Proof.* We state the proof for column-minimal animals. Let $P$ be a column-minimal animal, it can be encoded as a sequence of columns $(c_1, \dots, c_j)$, with $c_i \neq c_{i+1}$. Since $P$ is part of a *dense* family of animals, given any $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_j) \in \mathbb{Z}^{+j}$ there exists an animal $Q$ encoded by a sequence of columns $(c_1^{\alpha_1}, \dots c_j^{\alpha_j})$.

So

$$\begin{aligned}
\mathcal{E}_c(P) &= \bigcup_{\boldsymbol{\alpha}} \{c_1^{\alpha_1}, \dots, c_j^{\alpha_j}\} \\
G_c(P) &= \prod_i \sum_{\alpha_i} (x^{|c_i|_\Leftrightarrow})^{\alpha_i} \\
&= \prod_i \frac{x^{|c_i|_\Leftrightarrow}}{1 - x^{|c_i|_\Leftrightarrow}} \qquad (3.7)
\end{aligned}$$

where $|c_i|_\Leftrightarrow$ is the number of horizontal bonds in $c_i$. The result follows. The proof for $G_s(P)$ can be constructed in a similar way; instead of treating the animal as a whole, one considers the section configurations in each page. ∎

Directly from this we can deduce some of the properties of the coefficient of $y^n$ in the anisotropic generating function of a set of animals:

**Theorem 3.8.** *If $P(x, y) = \sum_{n \geq 0} H_n(x) y^n$ is the anisotropic generating function of some dense family of animals, $\mathcal{G}$, then*

- *$H_n(x)$ is a rational function,*
- *the degree of the numerator of $H_n(x)$ cannot be greater than the degree of its denominator, and*
- *the denominator of $H_n(x)$ is a product of cyclotomic polynomials.*

*Proof.* From lemma 3.5, $H_n(x)$ is the sum of the generating functions of the expansions of each of the minimal elements in $\mathcal{G}_n$. Lemmas 3.6 and 3.7 imply that this sum is a finite sum of rational functions with the desired properties. The result follows. ∎

◁ ◁ ◇ ▷ ▷

We can then sharpen the above result to determine which cyclotomic factors can appear in the denominator of $H_n(x)$, by noting that the denominator of $G_s(A)$ can only contain the cyclotomic factor $\Psi_k(x)$ if $A$ contains a $K$-section, where $K$ is some integer multiple of $k$.

### 3.3.3 The poles of $H_n(x)$

**Theorem 3.9 (Poles, columns and sections).** *If $H_n(x)$ has a denominator factor $\Psi_k(x)$, then $\mathcal{G}_n$ must contain a column-minimal animal containing a $K$-column for some $K \in \mathbb{Z}^+$ divisible by $k$. Further if $H_n(x)$ has a denominator factor $\Psi_k(x)^\alpha$, then $\mathcal{G}_n$ must contain a column-minimal animal that contains $\alpha$ columns that are $K$-columns for some (possibly different) $K \in \mathbb{Z}^+$ divisible by $k$.*

*Similar results hold for $k$-sections and section-minimal animals.*

*Proof.* The proof is identical for both partial orders. We state it here for section-minimal animals.

Let $\mathcal{M} = \{M_i\}$ be the set of section-minimal animals $\in \mathcal{G}_n$.

$$\begin{aligned} H_n(x) &= \sum_i G_s(M_i) \\ &= \sum_i \prod_K \left(\frac{x^K}{1-x^K}\right)^{\sigma_K(M_i)} \\ &= \sum_i x^{|M_i|_{\Leftrightarrow}} \prod_k \Psi_k(x)^{-\sum_d \sigma_{kd}(M_i)} \\ &= \frac{< \text{a big mess} >}{\prod_k \Psi_k(x)^{\mu_k}} \end{aligned}$$

where $\mu_k \leq \max_i\{\sum_d \sigma_{kd}(M_i)\}$ (this is an inequality since the numerator and denominator could share common cyclotomic factors). Consequently if there is no minimal element $M_i$ containing a $K$-section (for some $K$ divisible by $k$) then $\mu_k = 0$, and the denominator cannot contain $\Psi_k(x)$.

Similarly, if for all $M_i \in \mathcal{M}$ the sum, $\sum_{d\geq 1} \sigma_{kd}(M_i) < \alpha$, (*i.e.* there is no minimal animal that contains at least $\alpha$ columns that are $K$-columns for $K$ divisible by $k$) then $\mu_k < \alpha$.

∎

Using the above theorem we can get some idea of which cyclotomic factors will occur in the denominator of $H_n(x)$; in particular, we can bound the order of the cyclotomic factors that occur. *i.e.* we want to find some bound $m$, which will generally be a function of $n$, such that if $\Psi_k(x)$ occurs in the denominator of $H_n(x)$, then $k \leq m$.

**Corollary 3.10 (Denominators with $\preceq_s$ and $\preceq_c$).** *Let $\mathcal{G}$ be a dense family of animals, and let $\mathcal{M}_c$ (resp. $\mathcal{M}_s$) be the set of column-minimal (resp. section-minimal) animals of $\mathcal{G}_n$. Let*

$$\begin{aligned} c &= \max\{k \mid \exists A \in \mathcal{M}_c \text{ with } \gamma_k(A) > 0\} \\ s &= \max\{k \mid \exists A \in \mathcal{M}_s \text{ with } \sigma_k(A) > 0\} \end{aligned}$$

*Then $s \leq c$. Moreover, if $\Psi_k$ is a factor of the denominator of $H_n(x)$ then $k \leq s$.*

*Proof.* According to theorem 3.9, if there is a factor of $\Psi_k(x)$ in the denominator of $H_n(x)$, then there must be a minimal animal that has a $K$-section (for some $K$ divisible by $k$). Take a section-minimal animal, $A$, with an $s$-section. Since $A$ is section-minimal, it is also column-minimal, and so contains a column with at least $s$ horizontal bonds, so $s \leq c$. ∎

◁ ◁ ◇ ▷ ▷

Corollary 3.10 means that if we can show that no animal in $\mathcal{G}_n$ contains a column or section with more than $k$ horizontal bonds, then the denominator of $H_n(x)$ can only contain cyclotomic factors of order $\leq k$. Further, it implies that if we wish to attempt to find such a bound, it is better to find the maximum number of horizontal bonds occurring in a section (the number $s$), rather than the maximum number of horizontal bonds in a column (the number $c$), since $s \leq c$, and gives tighter bounds on the order of the cyclotomic factors that can occur.

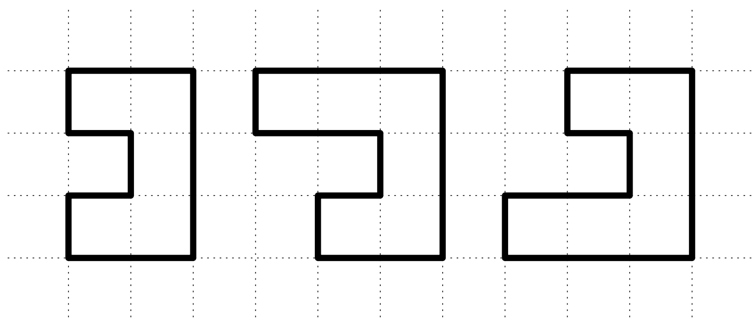


Figure 3.10: Polygons $P$, $Q$ and $R$ (from left to right) are all column-minimal, but $Q$ and $R$ reduce to $P$ under $\preceq_s$. So while $P$ contains only 1-sections, all contain 1 and 2 columns.

To illustrate corollary 3.10, consider the polygons in figure 3.10 enumerated by their horizontal half-perimeters. One can see that

$$\mathcal{E}_s(P) = \mathcal{E}_c(P) \uplus \mathcal{E}_c(Q) \uplus \mathcal{E}_c(R)$$

and hence

$$G_s(P) = G_c(P) + G_c(Q) + G_c(R)$$

where

$$\begin{aligned}
G_s(P) &= \frac{x^3}{(1-x)^3} \\
G_c(P) &= \frac{x^3}{(1-x)^2(1+x)} \\
G_c(Q) &= \frac{x^4}{(1-x)^3(1+x)} \\
G_c(R) &= \frac{x^4}{(1-x)^3(1+x)}
\end{aligned}$$

So the column-minimal polygons *suggest* the existence of a higher order cyclotomic factor, $(1+x)$, which the section-minimal polygon does not. Summing over all minimal diagrams *will* give the same generating function, but there will be *more* cancellations using column-minimal animals.

## 3.4 Applications

Theorem 3.9 tells us that if there is no animal in $\mathcal{G}_n$ that contains a $k$ section (or a $K$-section for any $K$ being an integer multiple of $k$), then the denominator of $H_n(x)$ does not contain a factor of $\Psi_k(x)$. Further, if there is no *minimal animal* that contains $\alpha$ $k$-sections in $\mathcal{G}_n$ (or a total of $\alpha$ $K$-sections for any $K$ being an integer multiple of $k$), then the denominator of $H_n(x)$ cannot contain a factor of $\Psi_k(x)^\alpha$.

**Corollary 3.11.** *We have the following results on the coefficient denominators in the anisotropic generating functions of various families of* dense *animals:*

- *The coefficient of $y^n$ in the anisotropic generating function of any subset of column convex polygons can only contain denominator factors $(1-x)$.*
- *The coefficient of $y^n$ in the anisotropic generating function of any subset of row convex polygons can only contain denominator factors $(1-x)$.*
- *The coefficients of $y^n$ in the anisotropic generating function of 3-choice polygons can only contain denominator factors $(1-x)$ and $(1+x)$.*

- *The factor $\Psi_k(x)$ for $k > 1$ cannot enter the denominator of $H_n(x)$ of spiral, 3-choice and 2-choice walks for $n \leq \binom{k}{2}$. The factor $\Psi_1(x) = (1 - x)$ can enter in $H_0(x)$.*

- *For any* dense *family of animals containing $n$ vertical bonds, $\mathcal{G}_n$, the horizontal bond generating function $H_n(x)$, cannot contain the denominator factor $\Psi_k(x)$ if $n < 2k - 2$.*

- *The exponent of $\Psi_k(x)$ in the denominator of $H_{2k-2}(x)$ is at most $k$.*

*Proof.* We claim that all of the above families of animals are dense. We proceed by showing how many vertical bonds are required to construct an animal that contains a given number of $k$-sections, and then the results follow by application of theorem 3.9.

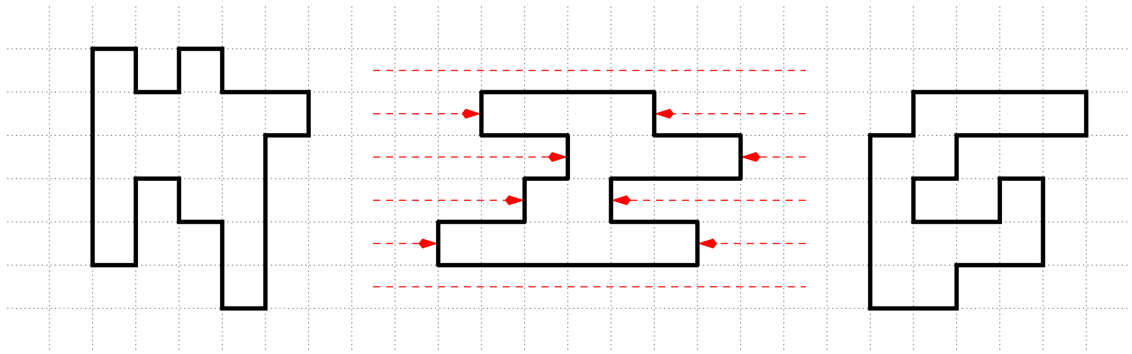


Figure 3.11: A column convex polygon, a row-convex polygon and a 3-choice polygon.

Consider the polygons given in figure 3.11.

- Column convex polygons by definition can only have 2 horizontal bonds in each column and hence only contain 1-sections.

- Row convex polygons containing $k$ rows, can have $k$-columns, but they are restricted to only have 1-sections due to row convexity.

- Each column of a 3-choice polygon can contain at most 4 horizontal bonds, and so all 3-choice polygons contain only 1- and 2-sections.

Consider the walks given in figure 3.12.

- The first occurrence of a $k$-section ($k > 1$) in a spiral, 3-choice or 2-choice walk (*i.e.* with a minimal number of vertical bonds) occurs in an outwardly spiraling configuration such as those drawn in figure 3.12. It is straightforward to calculate that the minimum number of vertical bonds for such a configuration is $1 + (1 + 2 + \cdots + (k - 1)) = \binom{k}{2} + 1$. A walk containing a 1-section can be constructed with no vertical bonds.

Consider the animals drawn in figure 3.13.

- To construct a $k$-section, at least $2k-2$ section lines need to be blocked, each requiring a single vertical bond.
- An animal containing a $k$-section and exactly $2k-2$ vertical bonds must be of height $k-1$. In such an animal if two $k$-sections are in adjacent columns they must be identical, and so the animal is not section-minimal.

  From the $k-1$ vertical bonds on the left (and similarly on the right) there can be at most $k$ lines of horizontal bonds towards the right (and left). One of these lines from the left must connect to one of the lines from the right, leaving $2k-2$ lines of horizontal bonds that can terminate in a vertex of degree 1. Hence between the vertical bonds on the left of the $k$-section and those on the right, there can be at most $2k-2$ degree one vertices.

  Between each pair of column there must be at least one of these degree 1 vertices, so there can be a total of $2k-1$ sections (between the vertical bonds). Not all of these can be $k$-sections, since if two $k$-sections are next to each other there will be no degree 1 vertex between them. So there can be $k$ $k$-sections, with $k-1$ not-$k$-sections between them.

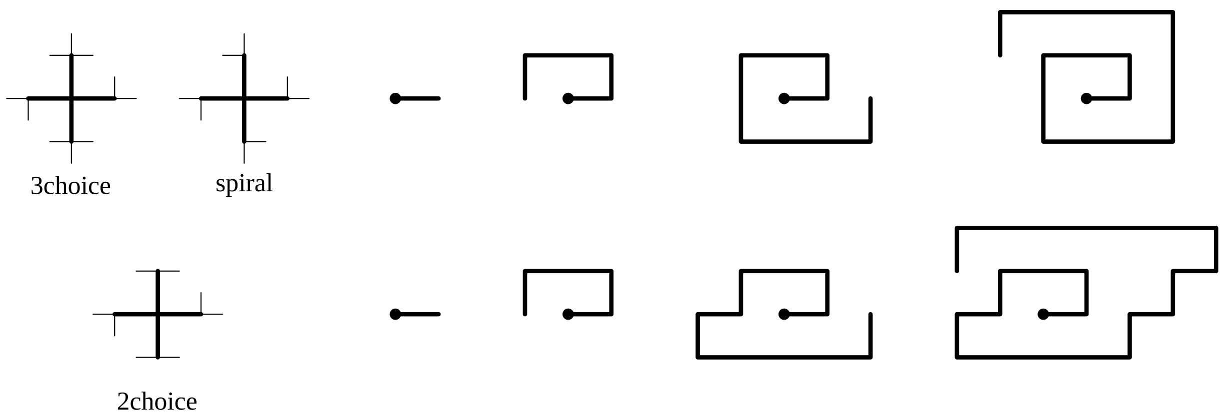



Figure 3.12: Spiral, 3-choice and 2-choice walks. The walk rules are given on the left. The spiral rule, for example, shows that after an east step the possible continuing steps are either east or north *etc.*, while the 2-choice walk rule shows that after a north step only an east or west continuing step is possible. An example of a section-minimal configuration containing $k$-sections ($k = 1, 2, 3, 4$) is given on the right.

■

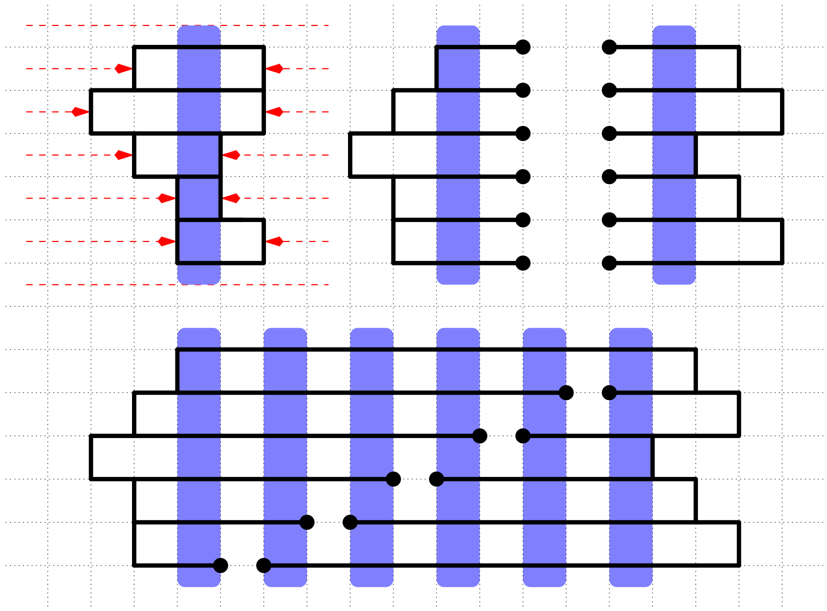


Figure 3.13: A single 6-section requires 10 vertical bonds. Between the left of the rightmost 6-section and the right of the leftmost 6-section there can be at most 12 vertices of degree 1, but at least two of these are required to connect the left of the animal to its right.

## 3.5 Proceeding further

The main problem with the above theory is that it only provides upper bounds for the exponent of cyclotomic factors that can appear in $H_n(x)$; it does not demonstrate that there are no cancellations between numerators and denominators. Consequently, it is very pertinent to consider the quality of these bounds — how tight are they?

In the case of the exactly solved models (column convex, row convex polygons) expansion of the solution shows that the denominator of $H_n(x)$ contains only factors of $(1-x)$, and it has been shown that the denominators of 3-choice polygons contain factors of $(1-x)$ and $(1+x)$ but no other factors [16].

Unsolved models must be examined numerically and the author thanks I. Jensen for providing his recent unpublished series for bond animals, directed bond animals, lattice trees and self-avoiding polygons [97]. These indicate (as far as has been calculated) that the upper bounds of corollary 3.11 are equalities; *e.g.* for bond animals the factor $\Psi_k(x)$ first appears in the denominator of $H_{2k-2}(x)$ with exponent *exactly equal* to $k$, while (see chapter 7) for SAPs $\Psi_k(x)$ appears in $H_{3k-2}(x)$ with exponent *exactly equal* to 1 (except when $k=2$). This indicates that the denominators predicted by the above techniques are very close to the exact denominators (sharing no common factors with the numerators). This gives some hope that one might be able to *prove* that there are no cancellations using some extension of these techniques. We have made some progress in this direction (see chapter 7).

### 3.5.1 Finding the exact exponent and D-finite functions

Proving that there are no cancellations at all between numerators and (predicted) denominators appears to be a very difficult problem. It is far simpler to consider the first occurrence of $\Psi_k(x)$. We have seen in the proof of corollary 3.11 that the animals that contain a $k$-section and the minimum number of vertical bonds, have relatively simple characterisations, and on the basis of this it is not unreasonable to hope that they (or a simple superset of them) can be enumerated. Once enumerated, analysis of their generating function will show whether or not there are any cancellations, which will determine the exponent of $\Psi_k(x)$.

We have done exactly this for self-avoiding polygons. We have shown that the factor $\Psi_k(x)$ appears in the denominator of $H_{3k-2}(x)$ with exponent exactly equal to 1 (excepting when $k=2$). This implies that the set of singularities of the coefficient of $y^n$ become dense on the unit circle, $|x|=1$, as $n\to\infty$. Numerical evidence [97, 83, 82] would suggest such behaviour in many unsolved models.

Such a nasty analytic property means that the *anisotropic* SAP generating function, $P(x,y)$, is not solvable in terms of algebraic functions (those satisfying an algebraic equation), nor D-finite functions [144, 117]. *i.e.* when considered a power series in $y$ with coefficients that are power series in $x$, it does not satisfy any linear differential equation with polynomial coefficients[3].

In contrast, all explicitly solved column-convex animals are algebraic (see [23] for example) and 3-choice polygons are D-finite [16]. This points to a very marked difference in analyticity between the models that we really would like to solve and the ones that we have been able to solve to date. This difference between the denominators of solved and unsolved models led Guttmann & Enting to propose examination of the anisotropic generating function as a numerical test of the solvability of a model [83]. If one finds that $H_n(x)$ has only a few poles for all $n$, then the model is probably solvable, while if the poles of $H_n(x)$ become dense as $n \to \infty$, then the model probably has a non-D-finite solution and is going to be much harder to tackle.

### 3.5.2 Patterns in the exponents

In all the models that have been observed, we find that the exponent of a given cyclotomic factor, $\Psi_k(x)$ in the denominator of $H_n(x)$ seems to obey a very simple pattern. For column-convex polygons (as well as 3-choice polygons and SAPs), the exponent of $(1-x)$ in $H_n(x)$ is simply $2n-1$. For self-avoiding polygons, the exponent of $\Psi_k(x)$, with $k \neq 2$, in the denominator of $H_{3k-2+l}(x)$ is simply $2l+1$. For bond animals and lattice trees, the exponent of $\Psi_k(x)$ with $k > 1$ in the denominator of $H_{2k-2+l}(x)$ is $k+2l$.

We have seen (in the proof of corollary 3.11) that the animals that contain a $k$-section and the minimum number of vertical bonds can be characterised quite simply. From these animals it is quite easy to then construct animals with more $k$-sections. See figure 3.14.

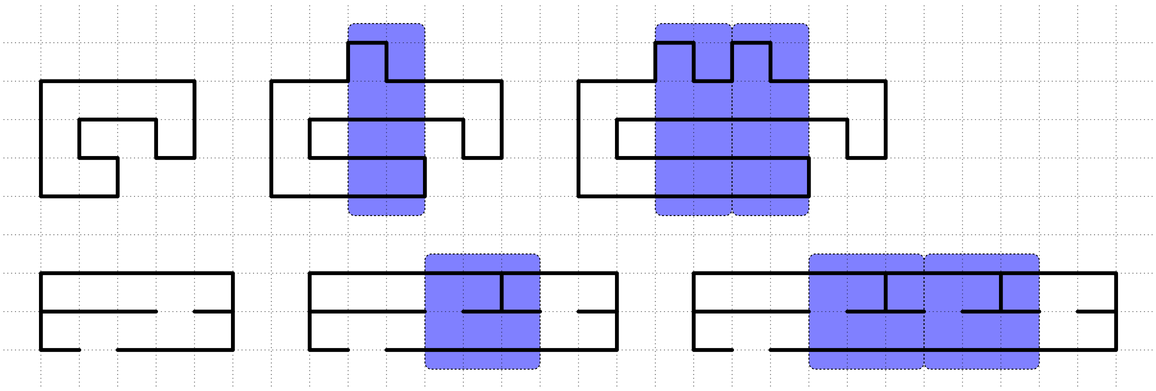


Figure 3.14: Building an animal with more $k$-sections from simpler animals, by inserting a simple bond configuration (highlighted).

[3] What this result implies about the *isotropic* generating function $P(x,x)$ is unclear; it is (unfortunately) all too easy to construct a non-D-finite function $P(x,y)$ for which $P(x,x)$ is D-finite or even rational.

One can cut the animals on one side of the $k$-section, and then insert some simple configuration of bonds that contains more $k$-sections and one (or in the case of polygons two) vertical bond(s). Repeating this process gives:

- a self-avoiding polygon with $(6k - 4 + 2l)$ vertical bonds and $(2l + 1)$ $k$-sections and
- a bond animal (or lattice tree) with $(2k - 2 + l)$ vertical bonds and $(k + 2l)$ $k$-sections.

Of course we are yet to show that no animal with more $k$-sections and the same number of vertical bonds exists, but it is a step in the right direction. We have proved that no such self-avoiding polygon exists (see chapter 7).

### 3.5.3 Magnetic models

Many thermodynamic functions in lattice models of magnets, such as the Ising model susceptibility, can be interpreted as the enumeration of families of bond animals. These families of bond animals are very different to those discussed above, most notably because they can be disconnected, but also because the weights are more complicated (they can even be negative). These weights can lead to very different functions; for the Ising model the specific heat is D-finite, the magnetisation is algebraic, while the susceptibility is thought to be non-D-finite [82].

# CHAPTER 4

# Towards larger classes of polyominoes

## 4.1

# Introduction

All the subclasses of polyominoes that have been solved to date have at least one of the following two properties: *convexity* or *directedness*. A polyomino is *column-convex* if its intersection with any vertical line is connected (Fig. 4.1.$a$); it is *directed* if any cell can be reached from a fixed cell, called the *source* or *root*, by a north-east directed path that only visits cells of the animal (Fig. 4.1.$b$). The most general polyominoes with these properties have been solved exactly (column-convex and directed polyominoes respectively): this shows that convexity limits the growth constant $\mu$ to 3.2..., while directedness limits it to 3. Table 4.1 gives more details, together with the nature of the associated *generating function* $\sum_n a_n x^n$.

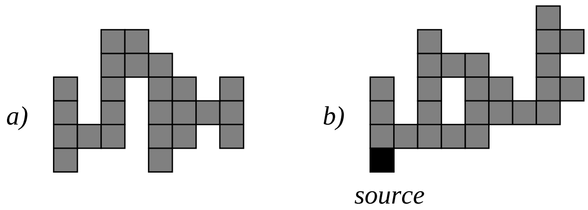



Figure 4.1: A column-convex polyomino and a directed polyomino.

| Model | $\mu$ | Nature of the GF | Who solved it (first) |
|---|---|---|---|
| Rectangles | 1 | $q$-series | obvious |
| Ferrers Diagrams (Partitions) | 1 | $q$-series | Euler [59] |
| Stacks | 1 | $q$-series | Auluck [5] |
| Staircase (Parallelogram) | 2.30... | $q$-series | Klarner & Rivest [108] |
| Directed Convex | 2.30... | $q$-series | Bousquet & Viennot [32] |
| Convex | 2.30... | $q$-series | Bousquet & Fédou [26] |
| Bargraph (Compositions) | 2 | rational | obvious |
| Directed Column Convex | 2.62... | rational | Moser, Klarner [104] |
| **Column Convex** | **3.20...** | **rational** | Temperley [151] |
| Diagonally convex directed | 2.66... | $q$-series | Privman & Švrakić [135] Bousquet [24], Feretić [66] |
| **Directed** | **3** | **algebraic** | Dhar [54] |

Table 4.1: Some of the solved subclasses of square lattice polyominoes and their growth constants.

In order to advance this line of polyomino enumeration one must *invent* new exactly solvable classes of animals. Since the largest directed and convex classes have been solved, if these new classes are to be larger, they cannot be restricted by directedness or convexity. In this chapter we define two new classes of square lattice animals, that are neither directed[1] nor convex. Each of them is in one-to-one correspondence with a natural class of *heaps of dimers*. These objects, defined in Section 4.2.1, have already proved useful in the enumeration of directed animals, by suggesting a canonical way to recursively factor them into two smaller directed animals (see Section 4.2.2 for details). The same kind of factorisation allows us to enumerate our first new class of animals, called *stacked directed animals*: the generating function is algebraic (like for directed animals), and the growth constant is 3.5 (Section 4.2.3). The second new class (*multi-directed animals*) is more general, also related to heaps, but resists the factorisation method. To solve this second model, we use a different approach, based on a column-by-column construction of animals (Section 4.3). This construction yields a functional equation for their generating function, which we solve explicitly: the generating function is no longer algebraic (and not even holonomic, see below for definitions). The growth constant is 3.58... (Section 4.4). This construction also works for directed animals, and allows us to take into account their width, which we prove to be a non-algebraic parameter (Section 4.4.2).

It is interesting to note that the enumeration of classes of polyominoes having a convexity property can also be attacked via two methods: a *wasp-waist factorisation* that consists in splitting the polyomino into two smaller ones, at a place where it is especially thin (see for instance [22, 133]), and a column-by-column construction, sometimes called Temperley's method (see, e.g., [23, 151]). This is somehow paralleled by the existence of two constructions for heaps, both of which we discuss in this chapter.

All our square lattice results have analogues for triangular lattice animals (polyominoes with hexagonal cells). The growth constants of the two new classes we define are respectively 4.5 and 4.58..., and surpass the constants 3.86... and 4 obtained by counting column-convex animals and directed animals on the triangular lattice [105, 54]. The largest class we enumerate is equinumerous with a class of animals introduced by Klarner [105], for which he simply gives a lower bound: details are given at the end of Section 4.3.1. The number of $n$-celled animals in each family, for small values of $n$, is given at the end of the chapter in Tables 3 (square lattice) and 4 (triangular lattice).

We are tempted to write that the classes of animals we define and count in this chapter are "the largest classes of animals ever counted exactly". This sentence has to be written with a few caveats: the lower bound on Klarner's constant $\mu \geq 3.9$ mentioned in chapter 1 suggests that one has been able to enumerate classes of animals whose growth constant is significantly larger than our "record" of 3.58... The classes of animals that provide such lower bounds have usually a rational generating function that *could* be evaluated exactly; but the details of this rational function are less interesting than the value of its smallest singularity (the inverse of which is the growth constant). The description of these classes usually depends on a parameter $k$: a simple example is provided by polyominoes of width at most $k$, whose generating function can be evaluated by transfer matrix techniques [161]; another example

[1] The word "directed" appears in the name of each of these animals (even though they are not directed) because each of them is in some sense constructed from a sequence of directed animals.

is based on a factorisation of polyominoes into *prime* polyominoes: the generating function for polyominoes whose prime factors have area less than $k$ is $(1 - P_k(x))^{-1}$, where $P_k(x)$ enumerates prime polyominoes of area less than $k$. We refer to [159] for variations on this factorisation. While certainly interesting as approximants of the real problem, the dependence on $k$, and the obvious linear structure of these classes of polyominoes makes them not very interesting in themselves.

We conclude this introduction with a few definitions, starting with a convention: since dimers are subject to gravity, which usually acts downwards, it is convenient to rotate the square lattice by $\pi/4$, and to represent the triangular lattice as a square lattice with diagonals (see Fig. 4.4). Hence the preferred direction of directed animals will now be north: in a directed animal, the source is connected to any given cell by a path consisting only of north east and north west steps.

The (area) *generating function* for a class $\mathcal{A}$ of animals is $\sum_n a_n x^n$, where $a_n$ denotes the number of animals of $\mathcal{A}$ having area $n$. We shall often enumerate animals according to several parameters, like the area and width, which will give rise to multivariate generating functions, such as

$$\sum_{n,k} a_{n,k} x^n u^k,$$

where $a_{n,k}$ is the number of animals of $\mathcal{A}$ having area $n$ and width $k$.

Given a ring $\mathbb{L}$ and $n$ indeterminates $x_1, \ldots, x_n$, we denote by

- $\mathbb{L}[x_1, \ldots, x_n]$ the ring of polynomials in $x_1, \ldots, x_n$ with coefficients in $\mathbb{L}$,
- $\mathbb{L}[[x_1, \ldots, x_n]]$ the ring of formal power series in $x_1, \ldots, x_n$ with coefficients in $\mathbb{L}$,

and if $\mathbb{L}$ is a field, we denote by

- $\mathbb{L}(x_1, \ldots, x_n)$ the field of rational functions in $x_1, \ldots, x_n$ with coefficients in $\mathbb{L}$.

A formal power series $F$ in $\mathbb{L}[[x_1, \ldots, x_n]]$ is said to be *algebraic* if there exists a non-trivial polynomial $P$ in $n+1$ variables, with coefficients in $\mathbb{L}$, such that $P(x_1, \cdots, x_n, F) = 0$. It is said to be *holonomic* (or *D-finite*) if the partial derivatives of $F$ with respect to the variables $x_i$ span a finite dimensional vector space over $\mathbb{L}(x_1, \ldots, x_n)$ (see [117]). In other words, for $1 \leq i \leq n$, the series $F$ satisfies a non-trivial partial differential equation of the form

$$\sum_{k=0}^{d_i} P_{k,i}(x_1, \cdots, x_n) \frac{\partial^k F}{\partial x_i^k} = 0,$$

where $P_{k,i}(x_1, \cdots, x_n)$ is a polynomial in the $x_j$.

Any algebraic series is holonomic (the converse is false). Any derivative of a holonomic series $F$ is holonomic, as is any (well-defined) specialisation of $F$. The classical theory of linear differential equations implies that a holonomic series $F(x)$ with complex coefficients has only finitely many singularities.

# 4.2

# Factorisation of heaps of dimers

## 4.2.1 Heaps of dimers and animals

Heaps of pieces are objects that give a convenient geometric representation of the elements of a partially commutative monoid[2] . This representation is due to Viennot [156]. Intuitively, a heap of dimers is obtained by dropping a finite number of dimers towards the horizontal axis. Each dimer falls until it touches the horizontal axis or another dimer (Fig. 4.2). The heap is *strict* if no dimer has another dimer directly above it. It is *connected* if its orthogonal projection on the horizontal axis is connected. The *width* of a connected heap is the number of non-empty columns. The dimers that touch the axis are *minimal*. A heap having only one minimal dimer is called a *pyramid*. If, moreover, this minimal dimer is the rightmost one, the heap is a *half-pyramid*. The *right width* of a pyramid is the number of nonempty columns to the right of the minimal dimer. The left width is defined similarly. A pyramid with zero right width is hence a half-pyramid.

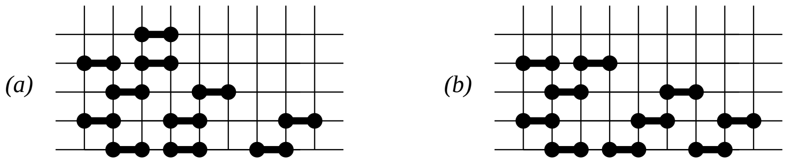



Figure 4.2: Two heaps of dimers; each has three minimal dimers.

For instance, the heap of Fig. 4.2.$a$ is neither strict nor connected, while the heap of Fig. 4.2.$b$ is both strict and connected, and has width 8. Two examples of pyramids can be found on Fig. 4.4; both have right width and left width 2. A (schematic) half-pyramid is shown on Fig. 4.6.

Take an animal $A$ on the square or triangular lattice; replace each cell by a dimer, and let the dimers fall. We call $V(A)$ the resulting heap of dimers. Note that $V(A)$ is connected (Fig. 4.3). It was observed by Viennot [155] that this mapping induces a bijection between directed animals on the square (resp. triangular) lattice and strict (resp. general) pyramids of dimers (Fig. 4.4). The correspondence $V$ will be the leading thread of this chapter: all the classes of animals we are going to enumerate will be in bijection with natural families of heaps of dimers.

We can associate with any strict heap $H$ an infinite set $\mathcal{E}(H)$ of general heaps by replacing each dimer of $H$ by a column of dimers of any positive height (Fig. 4.5). Conversely, given an element $H'$ of $\mathcal{E}(H)$, we can recover $H$ by compressing all columns of $H'$. Consequently,

[2]A monoid is a set with a multiplication operation that is closed on the set. *e.g.* the set of words constructed from the three letters, "$a$", "$b$" and "$c$", together with the concatenation operation is a monoid. A monoid is partially commutative if the product of some pairs of elements in the set is commutative.

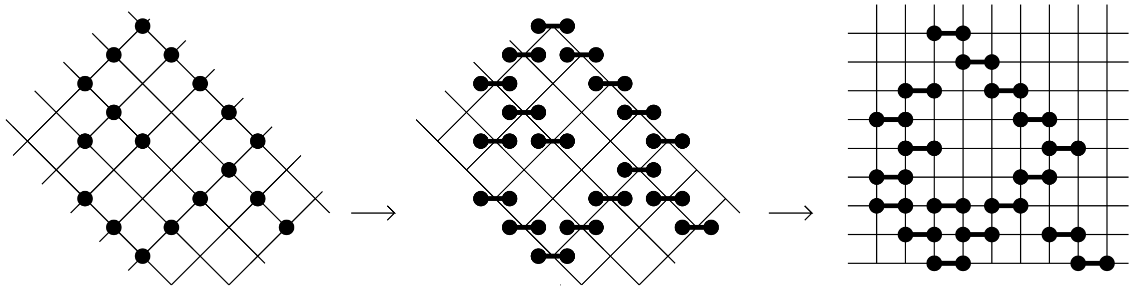


Figure 4.3: From animals to connected heaps: the transformation $V$.

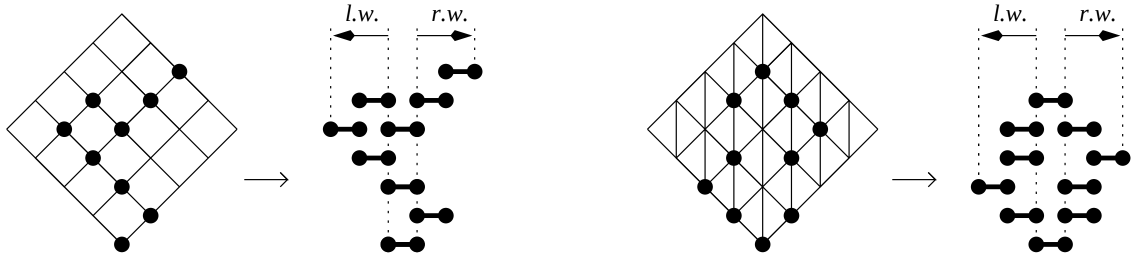



Figure 4.4: From directed animals to pyramids: the square lattice and the triangular lattice.

if $F(x)$ is the generating function of a class $\mathcal{F}$ of strict heaps, then the generating function for heaps of $\mathcal{E}(\mathcal{F})$ is $F(x/(1-x))$.

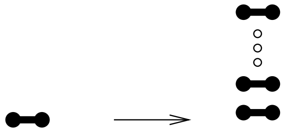


Figure 4.5: From strict heaps to general heaps.

Almost all the classes $\mathcal{A}$ of animals we are going to count in this chapter will have a square lattice version $\mathcal{A}_s$ and a triangular lattice version $\mathcal{A}_t$ (with one exception). The animals of $\mathcal{A}_s$ will be in bijection with a set $\mathcal{F}_s$ of strict heaps, and the animals of $\mathcal{A}_t$ will be in bijection with heaps of a set $\mathcal{F}_t$. These sets are related by the above mapping, $\mathcal{F}_t = \mathcal{E}(\mathcal{F}_s)$ and hence, the generating functions of the corresponding animals will always be related by $A_t(x) = A_s(x/(1-x))$.

### 4.2.2 Directed animals

Directed animals on the square and triangular lattices were first enumerated by Dhar [54, 55], followed by Gouyou-Beauchamps and Viennot [76]. However, both proofs are complicated when compared to the simplicity of the generating functions (see Proposition 4.1 below). In contrast, the notion of heaps provides extremely simple proofs, unveiled by Bétréma and

Penaud in [12, 13, 129] (which we present below). The key ingredient is a *monoid structure* on the set of heaps (the product of two heaps is obtained by putting one heap above the other and dropping its pieces). From unambiguous factorisations of pyramids, one derives algebraic equations for their generating function.

**Proposition 4.1.** *The generating functions $Q_s(x)$ and $Q_t(x)$ for strict and general half-pyramids (respectively) are given by*

$$Q_s(x) = \frac{1 - x - \sqrt{(1+x)(1-3x)}}{2x} \quad \textit{and} \quad Q_t(x) = Q_s\left(\frac{x}{1-x}\right) = \frac{1-2x-\sqrt{1-4x}}{2x}.$$

*Denoting the generating function for either set of half-pyramids by $Q$, the generating function for the corresponding pyramids, counted by their number of dimers ($x$) and their right width ($v$) is*

$$P(x,v) = \frac{Q(x)}{1 - vQ(x)}. \tag{4.1}$$

*In particular, the generating functions*

$$P_s(x,1) = \frac{1}{2}\left(\sqrt{\frac{1+x}{1-3x}} - 1\right) \quad \textit{and} \quad P_t(x,1) = P_s\left(\frac{x}{1-x}, 1\right) = \frac{1}{2}\left(\frac{1}{\sqrt{1-4x}} - 1\right)$$

*count directed animals on the square and triangular lattices respectively. Consequently, the number of $n$-celled directed animals on the square (resp. triangular) lattice is asymptotic to*

$$\frac{1}{\sqrt{3\pi}}\, 3^n n^{-1/2} \qquad \left(\textit{resp.}\ \ \frac{1}{2\sqrt{\pi}}\, 4^n n^{-1/2}\right).$$

*The average width of the corresponding pyramids is asymptotic to*

$$6\sqrt{3\pi}\; n^{1/2} \qquad \left(\textit{resp.}\ \ 16\sqrt{\pi}\; n^{1/2}\right).$$

*Proof.* These expressions are obtained by factoring pyramids in a canonical way. Let us begin with the factorisation of strict half-pyramids. Consider a strict half-pyramid $H$ having several dimers. If there is only one dimer (the minimal one) in the rightmost column, then $H$ is the product of its minimal dimer and a half-pyramid (Fig. 4.6). Otherwise, by pushing upwards the lowest non-minimal dimer of the rightmost column, we factor $H$ into two half-pyramids and a minimal dimer. This gives $Q_s(x) = x + xQ_s(x) + xQ_s(x)^2$. This equation is readily solved, yielding the expression of $Q_s(x)$ given in the proposition. Observe that the expansion operation of Fig. 4.5 does not change the right width and so preserves the property of being a half-pyramid; this gives the announced result for $Q_t(x)$. Alternatively, we could directly factor general half-pyramids to obtain an algebraic equation satisfied by $Q_t(x)$.

A pyramid (be it strict or general) is either a half-pyramid, or the product of a half-pyramid and a pyramid (Fig. 4.7). With the notations of the proposition, this implies that $P(x,v) = Q(x)(1 + vP(x,v))$, both for the strict and general pyramid model.

The asymptotic results follow from the general correspondence between the position and nature of singularities of a generating function and the asymptotic behaviour of its coefficients [70]. The average *right* width is obtained by differentiating (4.1) with respect to $v$, and the average width is of course twice the average half-width plus one. ■

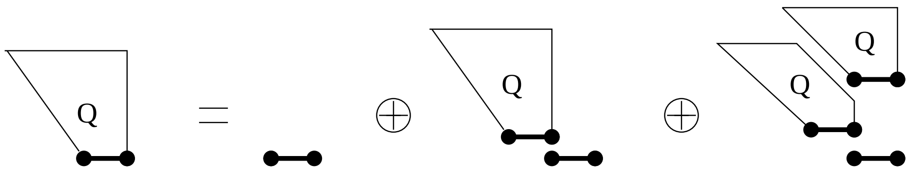



Figure 4.6: The factorisation of strict half-pyramids.

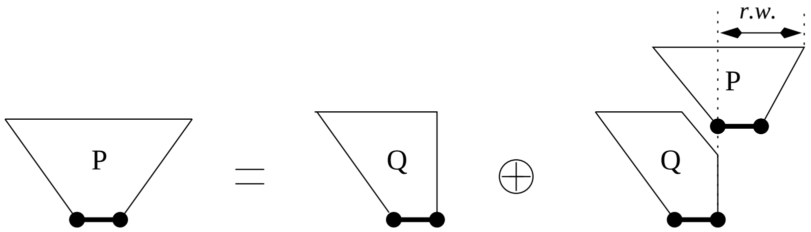



Figure 4.7: The factorisation of pyramids.

### 4.2.3 Multi-directed animals and stacked directed animals

Encouraged by the success of the mapping between directed animals and pyramids, it is natural to ask whether we can extend this mapping to other larger classes of heaps and animals. The map $V$, that transforms animals into connected heaps, can be shown to be surjective, and using the following argument one can describe a natural class of triangular lattice animals that are in bijection with connected heaps of dimers.

Let $H$ be a connected heap. We proceed by induction on the number of minimal pieces of $H$. If $H$ is a pyramid, let $\overline{V}(H)$ be the corresponding triangular lattice directed animal. If $H$ has $k$ minimal dimers, with $k > 1$, let us push upwards the $(k-1)$ leftmost minimal dimers: this yields a connected heap $H'$, placed "far above" a remaining pyramid $P_k$ (in figure 4.8 the heap $H'$ is the product of the two pyramids, $P_1$ and $P_2$). Replace $H'$ and $P_k$ by their corresponding animals $\overline{V}(H')$ and $\overline{V}(P_k)$, and push back $\overline{V}(H')$ downwards until it connects to $\overline{V}(P_k)$. Let $\overline{V}(H)$ be the resulting animal (Fig. 4.8).

In this way, we (recursively) define a class of triangular lattice animals that is in one-to-one correspondence with connected heaps of dimers. Since these animals are somehow constructed from a sequence of directed animals, we call them *multi-directed animals*.

The case of square lattice animals is a bit more delicate. The transformation $V$ maps some of them to non-strict heaps (Fig. 4.3). But, clearly, not all connected heaps can be obtained

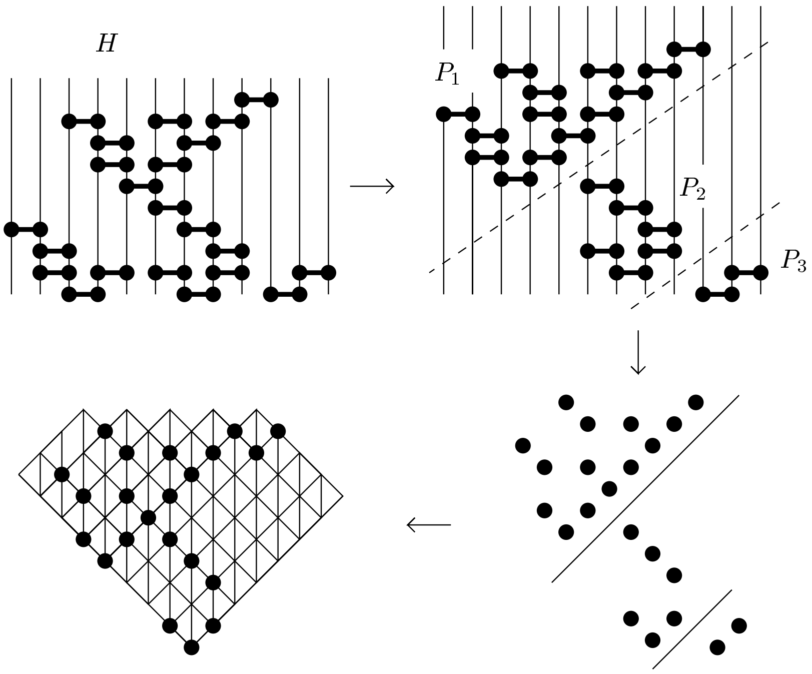



Figure 4.8: From connected heaps to animals: the transformation $\overline{V}$.

in this way: for instance, the column of height 2 has no pre-image among square lattice animals. It seems that the image by $V$ of square lattice animals has no simple description. However all is not lost: if we apply $\overline{V}$ to a *strict* heap $H$ we obtain a *square lattice* animal: hence $\overline{V}$ induces a one-to-one correspondence between strict connected heaps and square lattice multi-directed animals.

Take a connected heap $H$ with $k$ minimal pieces. The multi-directed animal $\overline{V}(H)$ is formed by the concatenation of $k$ directed animals. Let us denote by $P_1, P_2, \ldots, P_k$, from left to right, the corresponding pyramidal factors of $H$. The fact that $H$ is connected means that for $1 < i \leq k$, the projection of $P_i$ onto the horizontal axis intersects the projection of some $P_j$, for $j < i$. In the example of Fig. 4.8, the projections of $P_2$ and $P_3$ intersect the projection of $P_1$.

Let us call *stacked pyramids* the connected heaps such that for $1 < i \leq k$, the horizontal projection of $P_i$ intersects the horizontal projection of $P_{i-1}$. Let us call *stacked directed animals* the corresponding animals (Fig. 4.9). We define the *right width* of a stacked pyramid to be the right width of its rightmost pyramidal factor.

It turns out that these objects are easier to enumerate than connected heaps (but less general): the recursive description of stacked pyramids easily translates into algebraic equations for their generating function, which again turn out to be quadratic. In terms of growth constants, this simple model is already larger than all previous exactly solved models.

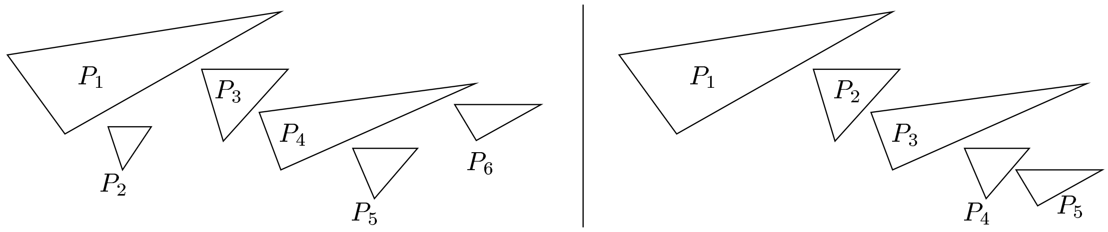



Figure 4.9: The structure of multi-directed animals and stacked directed animals. Each triangle stands for a directed animal.

**Proposition 4.2.** *Let $Q(x) \equiv Q$ denote the generating function for strict (resp. general) half-pyramids. Let $P(x, v)$ denote the generating function for strict (resp. general) pyramids, where the variable $v$ enumerates the right width. Then the generating function for strict (resp. general) stacked pyramids, counted by their number of dimers ($x$), right width ($v$) and number of minimal dimers ($t$), is*

$$S(x, v, t) = \frac{tP(x, v)}{1 - tP(x, 1)^2} = \frac{tQ(1 - Q)^2}{(1 - vQ)[(1 - Q)^2 - tQ^2]}.$$

*In particular, the area generating function for strict (resp. general) stacked pyramids is:*

$$\begin{aligned} S_s(x) &= \frac{(1 - 2x)(1 - 3x) - (1 - 4x)\sqrt{(1 - 3x)(1 + x)}}{2x(2 - 7x)}, \\ S_t(x) &= S_s\left(\frac{x}{1 - x}\right) = \frac{(1 - 3x)(1 - 4x) - (1 - 5x)\sqrt{1 - 4x}}{2x(2 - 9x)}. \end{aligned}$$

*Consequently, the number of $n$-celled stacked directed animals on the square (resp. triangular) lattice is asymptotic to*

$$\frac{3}{28}\, 3.5^n \qquad \left(\text{resp. } \frac{1}{12}\, 4.5^n\right).$$

*The average number of minimal dimers in the corresponding stacked pyramids, which is a lower bound on their width, is asymptotic to*

$$\frac{3}{28}\, n \qquad \left(\text{resp. } \frac{1}{12}\, n\right).$$

*The width is trivially bounded above by $n$.*

*Proof.* We follow the description of $\overline{V}$ given above. Each stacked pyramid $H$ is:

- either a single pyramid (if it has only one minimal piece),
- or the product of a pyramid $P \equiv P_k$ and a stacked pyramid $H'$ placed above $P$ (see Fig. 4.10). The number of ways the pyramid $P$ can be placed is equal to the right width of $H'$. The right width of $H$ is then the right width of $P$.

This shows that

$$S(x,v,t) = tP(x,v)\left(1 + S'_v(x,1,t)\right).$$

We differentiate this equation with respect to $v$ and then set $v$ to 1 in order to compute the derivative $S'_v(x,1,t)$. We finally obtain

$$S(x,v,t) = \frac{tP(x,v)}{1 - tP'_v(x,1)},$$

and we use Proposition 4.1 to obtain the announced expression of $S(x,v,t)$. The asymptotic results follow as in the proof of Proposition 4.1.

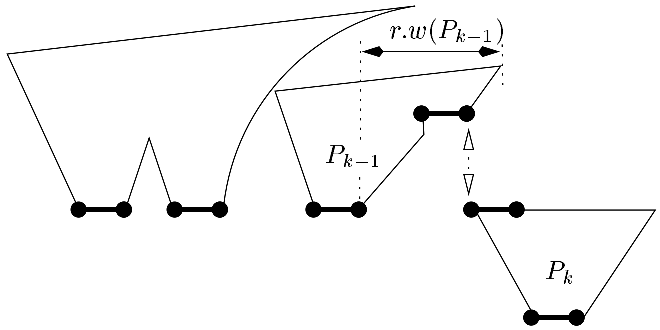



Figure 4.10: A recursive construction of stacked pyramids.

An alternative construction consists in adding a new pyramid $P \equiv P_1$ to the above left of a stacked pyramid $H'$, rather than to its below-right (see Fig. 4.11). The pyramid $P$ can be placed in a number of ways equal to its right width. This yields directly

$$S(x,v,t) = tP(x,v) + tS(x,v,t)P'_v(x,1).$$

■

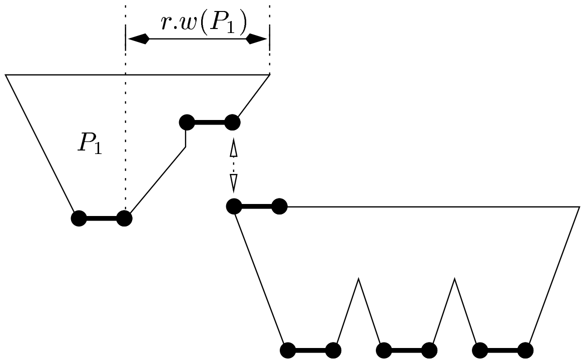



Figure 4.11: Another recursive construction of stacked pyramids.

For comparison, we state here the result for connected heaps. It will only be proved in Section 4.4.

**Proposition 4.3.** *Let $Q(x) \equiv Q$ denote the generating function for strict (resp. general) half-pyramids. Then the generating function for strict (resp. general) connected heaps of dimers is*

$$C(x) = \frac{Q}{(1-Q)\left[1 - \sum_{k\geq 1} \frac{Q^{k+1}}{1-Q^k(1+Q)}\right]}.$$

*This series is not algebraic, nor even D-finite. The number of strict (resp. general) connected heaps having $n$ dimers is asymptotic to $A\mu^n$ for some positive constant $A$, with $\mu = 3.58789436...$ (resp. $\mu = 4.58789436...$).*

Once again, due to the mapping between strict and general heaps (see figure 4.5 and surrounding text) we note that it is sufficient to prove this result for general connected heaps: with obvious notations,

$$C_s(x) = C_t(x/(1+x)) \quad \text{and} \quad Q_s(x) = Q_t(x/(1+x)).$$

Why is the result for connected heaps so much more complicated than the result for stacked pyramids? Why can't we apply the method of Proposition 4.2? After all, a connected heap is also constructed recursively by adding a new pyramid to the below-right of another connected heap. Let us explain why the approach of Proposition 4.2 does not work for connected heaps.

Take a pyramid $P \equiv P_k$, and a connected heap $H'$ having $k-1$ minimal dimers. The number of ways we can place $P$ to the below-right of $H'$ so as to form a new connected heap $H$ having $k$ minimal dimers is the *modified right width* of $H'$, which we define to be the number of nonempty columns to the right of the rightmost minimal dimer of $H'$ (Fig. 4.12). If we are to proceed by the same construction then we have to take the modified right width into account in the enumeration. So the next question is: what is the modified right width of the connected heap $H$ we have obtained? If $P$ has been placed in $i$th position (the leftmost position corresponding to $i = 1$), then, with obvious notations,

$$m.r.w(H) = r.w(P) + \max(0, m.r.w(H') - i - w(P)),$$

and this rather complicated formula, combined with the fact that the width of pyramids is an inherently non-algebraic parameter (see Section 4.4.2), explains why the first construction we used in the proof of Proposition 4.2 will not work for connected heaps.

In the proof of Proposition 4.2, we described a second construction of stacked pyramids, that consisted of adding a new pyramid $P \equiv P_1$ to the above-left of another stacked pyramid. The principle of this construction also works for connected heaps, but, when we try to use it,

- we actually have to count *all* heaps, because the connectivity is not necessarily preserved when removing the leftmost pyramidal factor from a connected heap (Fig. 4.8);
- we have to take into account a width-like parameter whose transformation during the construction depends simultaneously on the right width and left width of the pyramid $P_1$.

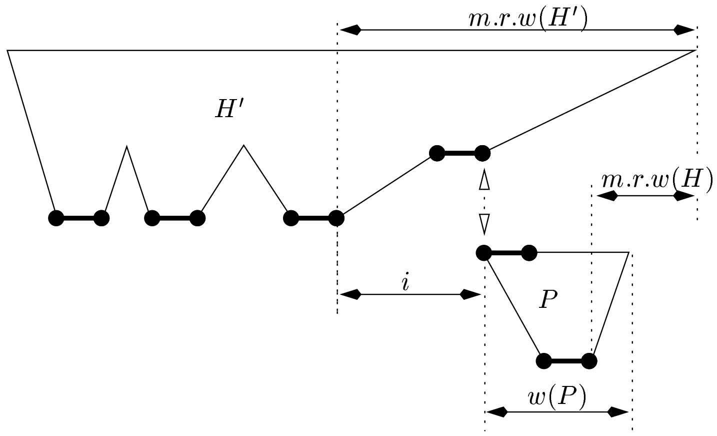



Figure 4.12: Recursive construction of connected heaps.

For these reasons, it is perhaps not too surprising that in order to prove Proposition 4.3 we will have to make a slight detour and examine heaps of fixed width.

4.3

# Heaps of fixed width: rational generating functions

In this section, we focus on general heaps and triangular lattice animals. The expansion/contraction procedure of Fig. 4.5 can be used to provide analogous results for square lattice animals. We attack the enumeration of heaps via a recursive construction that is even more elementary than those of Section 4.2: it simply consists of building a heap column by column. This approach provides functional equations for the corresponding generating functions and allows us to take into account the *width* of heaps (Section 4.3.1). This approach does *not* work for stacked pyramids, but suggests a new class of connected heaps, which we call *saw-tooth heaps*. We solve the functional equations in Section 4.3.2 and give combinatorial explanations for the form of the solutions in Section 4.3.3.

## 4.3.1 Establishing functional equations

Let $\mathbf{Q}(x, y, u)$ be the generating function for half-pyramids, where $u$ counts the width, $y$ the dimers in the rightmost column, and $x$ the other dimers. Similarly, let $\overline{\mathbf{Q}}(x, y, u)$ be the generating function for half-pyramids, where $y$ now counts dimers of the *leftmost* column. Let $\mathbf{P}(x, y, u, v)$ be the generating function for pyramids, where $u$ counts the left width, $v$ the right width, $y$ the dimers in the rightmost column, and $x$ the other dimers (Fig. 4.13). Finally, let $\mathbf{C}(x, y, u)$ count connected heaps by their width ($u$), number of dimers in the rightmost column ($y$), and number of remaining dimers ($x$).

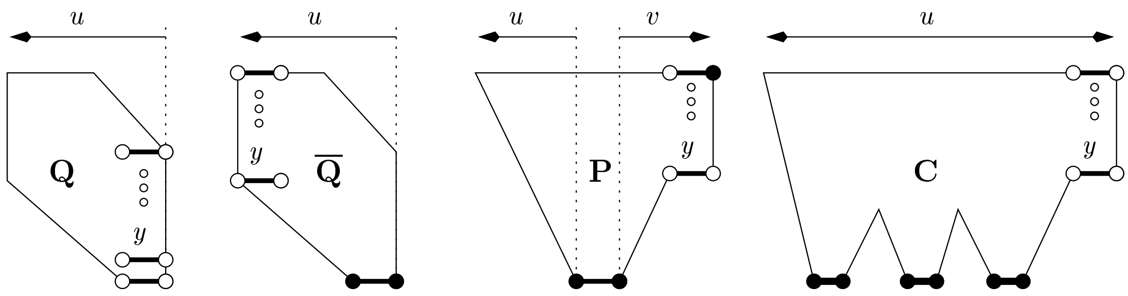



Figure 4.13: The various generating functions.

Constructing these classes of heaps column by column yields the following functional equations. For simplicity, we denote $\mathbf{Q}(x, y, u) \equiv \mathbf{Q}(y)$, etc.

**Proposition 4.4.** *The heap generating functions defined above are governed by the following functional equations:*

$$\begin{aligned}
\mathbf{Q}(y) &= \frac{uy}{1-y} + \frac{uy}{1-y}\mathbf{Q}\left(\frac{x}{1-y}\right), \\
\overline{\mathbf{Q}}(y) &= \frac{uy}{1-y} + u\overline{\mathbf{Q}}\left(\frac{x}{1-y}\right) - u\overline{\mathbf{Q}}(x), \\
\mathbf{P}(y) &= \frac{1}{u}\mathbf{Q}(y) + v\mathbf{P}\left(\frac{x}{1-y}\right) - v\mathbf{P}(x), \\
\mathbf{C}(y) &= \frac{uy}{1-y} + \frac{u}{1-y}\mathbf{C}\left(\frac{x}{1-y}\right) - u\mathbf{C}(x).
\end{aligned}$$

*Proof.* Let us begin with the equation governing $\mathbf{Q}$. Take a half-pyramid, and delete the dimers in its rightmost column; then the remaining dimers (if any) form a new half-pyramid. Conversely, any half-pyramid $H$ of width $m \geq 2$ can be grown from a half-pyramid $H'$ of width $m-1$ by creating a new column to the right of $H'$. More precisely:

1. we insert a (possibly empty) column of dimers to the above-right of each dimer in the rightmost column of $H'$. This corresponds to the substitution $y \longmapsto \frac{x}{1-y}$ in the generating function $\mathbf{Q}$;

2. we insert a nonempty column of dimers to the below-right of the minimal dimer of $H'$. The bottom dimer of this column will be the minimal piece of $H$. This operation is represented by multiplying by $\frac{y}{1-y}$.

This procedure, illustrated by Fig. 4.14, generates all half-pyramids of width at least two. The term $uy/(1-y)$ accounts for half-pyramids of width one.

Let us now prove the equation governing $\overline{\mathbf{Q}}$. This time, we construct half-pyramids of width $m \geq 2$ by creating a new column to the *left* of a pyramid $H'$ of width $m-1$. More precisely, we insert a (possibly empty) column of dimers to the above-left of each dimer in

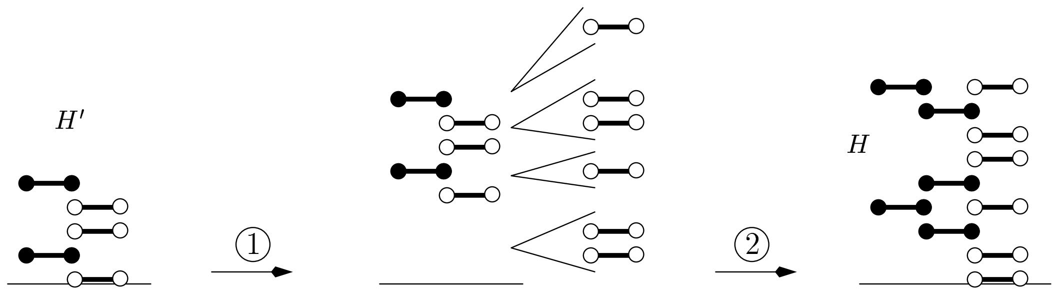



Figure 4.14: Constructing half-pyramids column by column.

the leftmost column of $H'$. This corresponds to the substitution $y \longmapsto \frac{x}{1-y}$ in the generating function $\overline{\mathbf{Q}}$. Since each of these columns of dimers we have just added can be of zero height, it is possible that we might not have added any dimers at all. To avoid this unwanted case, we simply subtract $u\overline{\mathbf{Q}}(x, x, u)$.

The two other equations are proved in a similar manner.

It is worth noting that one can also find these equations using the Hadamard-Temperley method described in chapter 5 — in fact this is how we first discovered them, before this more combinatorial explanation. ∎

By symmetry, the series $\mathbf{C}(x, y, u)$ also counts connected heaps by their width ($u$), *leftmost* dimers ($y$) and remaining dimers ($x$). Let us compare the functional equations defining the series $\overline{\mathbf{Q}}$ and $\mathbf{C}$. They reflect a recursive construction of half-pyramids (resp. connected heaps) where a new column of dimers is created at each step to the left of the object. The difference between the two equations comes from the fact that, in the construction of connected heaps, we can create new minimal dimers by adding a new column of dimers to the below-left of the lowest leftmost dimer, whereas this is forbidden when we construct half-pyramids. If we bound the height of this new column of dimers by $h$, we obtain a family of heaps that interpolates between half-pyramids ($h = 0$) and connected heaps ($h = \infty$). In particular, for $h = 1$, we obtain a class of heaps whose generating function $\mathbf{N}(x, y, u) \equiv \mathbf{N}(y)$ satisfies:

$$\mathbf{N}(y) = \frac{uy}{1-y} + u(1+y)\mathbf{N}\left(\frac{x}{1-y}\right) - u\mathbf{N}(x). \tag{4.2}$$

The lower edge of these heaps is constrained so that (when drawn from left to right) it may only grow upwards diagonally, but is able to grow straight down, something like the edge of a saw-tooth. Because of this similarity, we call these heaps *saw-tooth heaps* (Fig. 4.15).

**Remark.** Let $c_k(n; a_k)$ denote the number of connected heaps of width $k$ having $n$ dimers, $a_k$ of which lie in the rightmost column. The equation defining $\mathbf{C}(x, y, u)$, given in Proposi-

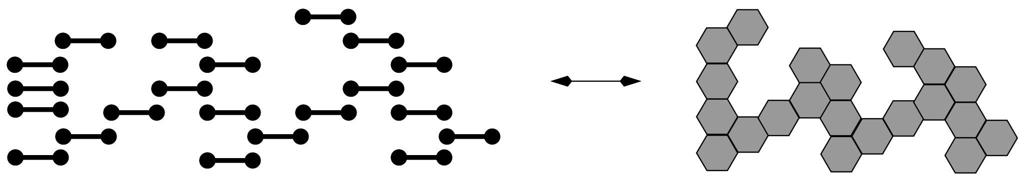


Figure 4.15: A saw-tooth heap and the corresponding honeycomb lattice polyomino.

tion 4.4, is equivalent to the following recursion:

$$c_k(n; a_k) = \begin{cases} 1 & \text{if } k = 1 \text{ and } a_k = n, \\ \displaystyle\sum_{a_{k-1}=1}^{n-a_k} \binom{a_{k-1}+a_k}{a_{k-1}} c_{k-1}(n-a_k; a_{k-1}) & \text{if } k > 1 \text{ and } a_k < n, \\ 0 & \text{otherwise.} \end{cases}$$

Equivalently, the number of connected heaps of width $k$ having $a_i$ dimers in the $i$th column, for $1 \leq i \leq k$, is

$$\binom{a_1+a_2}{a_1}\binom{a_2+a_3}{a_2}\cdots\binom{a_{k-1}+a_k}{a_{k-1}}. \tag{4.3}$$

In [105, Section 4], Klarner describes a class of *square lattice* animals counted by a sequence

$$b(n) = \sum f(a_1, a_2) f(a_2, a_3) \cdots f(a_{k-1}, a_k), \tag{4.4}$$

where the sum extends over all compositions $(a_1, a_2, \cdots, a_k)$ of $n$ in positive parts, and the generating function for the numbers $f(a, b)$ is given by

$$\sum_{a,b\geq 1} f(a,b) x^a y^b = \frac{1-xy}{1-x-y+x^2y^2} - \frac{1-xy}{(1-x)(1-y)}.$$

Then, he mentions the existence of an analogous class of animals *on the triangular lattice*, which are enumerated by (4.4) with $f(a,b) = \binom{a+b}{a}$. The description of these two families is a little ambiguous, and while studying them we realized that (i) the triangular lattice model was equivalent to (general) connected heaps of dimers and (ii) this model could be solved "more exactly" than the treatment in Klarner's paper where he only gives Eq. (4.3) and the lower bound of 4 on the growth constant. These realisations were the starting point of this work. Let us underline that the class of *square lattice* animals described by Klarner is *not* equivalent to strict connected heaps: the corresponding growth constant is approximately 3.72. Hence his class of square lattice animals is larger (and probably more interesting) than the class of strict connected heaps. Unfortunately, we have so far not been able to solve the associated functional equation.

### 4.3.2 Solving functional equations

The first idea that comes to mind to solve the equations of Proposition 4.4 – iteration – turns out to be the right one. Let us, for instance, start iterating the equation ruling $\mathbf{Q}$:

$$
\begin{aligned}
\mathbf{Q}(y) &= \frac{uy}{1-y}\left[1+\mathbf{Q}\left(\frac{x}{1-y}\right)\right] \\
&= \frac{uy}{1-y}+\frac{u^2xy}{(1-y)(1-x-y)}\left[1+\mathbf{Q}\left(\frac{x(1-y)}{1-x-y}\right)\right] \\
&= \frac{uy}{1-y}+\frac{u^2xy}{(1-y)(1-x-y)} \\
&\quad +\frac{u^3x^2y}{(1-x-y)(1-2x-y(1-x))}\left[1+\mathbf{Q}\left(\frac{x(1-x-y)}{1-2x-y(1-x)}\right)\right] \\
&= \frac{uy}{1-y}+\frac{u^2xy}{(1-y)(1-x-y)}+\frac{u^3x^2y}{(1-x-y)(1-2x-y(1-x))} \\
&\quad +\frac{u^4x^3y}{(1-2x-y(1-x))(1-3x+x^2-y(1-2x))}\left[1+\mathbf{Q}\left(\frac{x(1-2x-y(1-x))}{1-3x+x^2-y(1-2x)}\right)\right].
\end{aligned}
$$

After a few iterations, it becomes clear that the polynomials involved here are the *Fibonacci polynomials* $F_n$, defined by $F_0 = F_1 = 1$, and for $n \geq 2$,

$$F_n = F_{n-1} - xF_{n-2}. \tag{4.5}$$

This is not surprising, since these polynomials arise in the enumeration of Dyck paths of bounded height [109, Section 6], and a general bijection between walks on a graph and heaps of cycles [156, Section 6] puts Dyck walks of height $m$ in one-to-one correspondence with half-pyramids of width $m$. Another way of understanding the role of these polynomials is based on general heap enumeration results and on the fact that $F_n$ counts *trivial heaps of dimers* on a segment of $n$ vertices (equivalently $F_n$ is the matching polynomial of a segment of $n$ vertices). This approach will be detailed in the next subsection. To state our results concisely, it is convenient to introduce the bivariate polynomials $\tilde{F}_n(x,y) \equiv \tilde{F}_n$ defined (in terms of the Fibonacci polynomials) for $n \geq 2$ by

$$\tilde{F}_n = F_{n-1} - yF_{n-2}, \tag{4.6}$$

with $\tilde{F}_0 = \tilde{F}_1 = 1$. By convention, $F_i = \tilde{F}_i = 0$ if $i < 0$, so that (4.5) and (4.6) hold for $n \geq 1$. Observe that $\tilde{F}_n(x,x) = F_n$.

**Proposition 4.5.** *The heap generating functions* $\mathbf{Q}, \overline{\mathbf{Q}}, \mathbf{P}, \mathbf{C}$ *and* $\mathbf{N}$ *defined above are given by:*

$$\begin{aligned}
\mathbf{Q}(x,y,u) &= \sum_{n\geq 1} \frac{yx^{n-1}u^n}{\tilde{F}_n \tilde{F}_{n+1}}, \\
\overline{\mathbf{Q}}(x,y,u) &= \sum_{n\geq 1} \frac{yx^{n-1}u^n}{F_n \tilde{F}_{n+1}}, \\
\mathbf{P}(x,y,u,v) &= \sum_{m\geq 0}\sum_{n\geq 0} yx^n u^m v^n \left[\frac{F_m F_{m+1}}{F_{m+n+1}\tilde{F}_{m+n+2}} - \frac{F_{m-1}F_m}{F_{m+n}\tilde{F}_{m+n+1}}\right] \\
&= \sum_{m\geq 0} \frac{u^m x^m y}{\tilde{F}_{m+1}\tilde{F}_{m+2}} + \sum_{m\geq 0}\sum_{n\geq 1} x^{m+1}u^m v^n \left[\frac{\tilde{F}_n \tilde{F}_{n+1}}{\tilde{F}_{m+n+1}\tilde{F}_{m+n+2}} - \frac{F_{n-1}F_n}{F_{m+n}F_{m+n+1}}\right], \\
\mathbf{N}(x,y,u) &= \sum_{n\geq 1} \frac{x^{n-1}yu^n}{\tilde{F}_{n+1}} \prod_{i=1}^{n-1}\left(1 + \frac{F_i}{F_{i+1}}\right), \\
\mathbf{C}(x,y,u) &= \frac{\displaystyle\sum_{n\geq 0} \frac{u^n}{\tilde{F}_{n+1}}}{\displaystyle\sum_{n\geq 0} \frac{u^n}{F_n}} - 1.
\end{aligned}$$

*Observe that the last expression defines* $\mathbf{C}(x,y,u)$ *as a power series in* $u$ *with rational coefficients in* $x$ *and* $y$*, but that it is not well-defined when we attempt to set* $u = 1$*.*

*Proof.* The easiest but least illuminating way of proving these identities consists of verifying that the series defined by the above expressions satisfy the functional equations of Proposition 4.4 and Eq. (4.2). This verification is straightforward, all the identities resulting from the definition of the polynomials $F_n$ and $\tilde{F}_n$, and from the fact that for $n \geq 1$,

$$\tilde{F}_n\left(x, \frac{x}{1-y}\right) = \frac{\tilde{F}_{n+1}}{1-y}.$$

To prove that the first expression of $\mathbf{P}(x,y,u,v)$ satisfies the third equation of Proposition 4.4, we also need the identity

$$F_m\tilde{F}_{m+1} - F_{m-1}\tilde{F}_{m+2} = F_m^2 - F_{m-1}F_{m+1} = x^m, \qquad m \geq 0, \tag{4.7}$$

which is easily proved by induction on $m$.

The uniqueness of these solutions follows from the linearity of the functional equations. ∎

This is, perhaps, a little dissatisfying and it is far more interesting to describe how one can *discover* these identities. Clearly, the first two can be conjectured by inspection: we iterate the corresponding equation a few times to obtain the coefficient of $u^n$, for small values of

$n$, and the pattern soon becomes clear. The same approach also works for **N**. Iterating the equation defining **C**, however, does not suggest any simple pattern for the numerator:

$$\mathbf{C}(y) = \frac{uy}{1-y} + \frac{u^2xy(2-x-y)}{(1-x)(1-y)(1-x-y)} \\ + \frac{u^3x^2y(4-9x-6y+4x^2+10xy+2y^2-2x^2y-2xy^2)}{(1-x)(1-2x)(1-y)(1-x-y)(1-2x-y+xy)} + O(u^4).$$

Let us explain how to obtain the expression of **C**. It is more convenient to start with the series $\overline{\mathbf{C}}(x,y,u) \equiv \overline{\mathbf{C}}(y) = 1 + \mathbf{C}(x,y,u)$, which satisfies

$$\overline{\mathbf{C}}(y) = 1 + \frac{u}{1-y}\overline{\mathbf{C}}\left(\frac{x}{1-y}\right) - u\overline{\mathbf{C}}(x). \tag{4.8}$$

Let us introduce two operators $\Phi$ and $\Psi$ that act on power series of $\mathbb{Q}[[x,y,u]]$:

$$\Phi(\mathbf{S}(x,y,u)) = \frac{u}{1-y}\mathbf{S}\left(x, \frac{x}{1-y}, u\right) \quad \text{and} \quad \Psi(\mathbf{S}(x,y,u)) = -u\mathbf{S}(x,x,u).$$

Observe that, if **S** does not depend on $y$, then

$$\Phi(\mathbf{ST}) = \mathbf{S}\Phi(\mathbf{T}) \quad \text{and} \quad \Psi(\mathbf{ST}) = \mathbf{S}\Psi(\mathbf{T}).$$

Using these operators Eq. (4.8) can be rewritten as

$$\overline{\mathbf{C}}(y) = 1 + (\Phi + \Psi)\overline{\mathbf{C}}(y).$$

Iterating formally Eq. (4.8) provides

$$\overline{\mathbf{C}}(y) = \sum_{n\geq 0}(\Phi+\Psi)^n(1) = \frac{1}{1-\Phi-\Psi}(1).$$

Equivalently, we can consider $\overline{\mathbf{C}}(y)$ to be the sum of every word in the language $\{\Phi,\Psi\}^*$ acting on 1,

$$\overline{\mathbf{C}}(y) = \sum_{w\in\{\Phi,\Psi\}^*} w(1).$$

The language $\{\Phi,\Psi\}^*$ can be written as the disjoint union of $\Phi^*$ and $\{\Phi,\Psi\}^*\Psi\Phi^*$. It is easy to prove by induction on $n \geq 0$ that

$$\Phi^n(1) = \frac{u^n}{\tilde{F}_{n+1}} \quad \text{and} \quad \Psi\Phi^n(1) = -\frac{u^{n+1}}{F_{n+1}}.$$

Observe that $\Psi\Phi^n(1)$ is independent of $y$. Applying these facts gives

$$\begin{aligned}
\overline{\mathbf{C}}(y) &= \sum_{w\in\Phi^*} w(1) + \sum_{w\in\{\Phi,\Psi\}^*\Psi\Phi^*} w(1) \\
&= \sum_{n\geq 0}\Phi^n(1) + \frac{1}{1-\Phi-\Psi}\left(\sum_{n\geq 0}\Psi\Phi^n(1)\right) \\
&= \sum_{n\geq 0}\frac{u^n}{\tilde{F}_{n+1}} - \frac{1}{1-\Phi-\Psi}\left(\sum_{n\geq 0}\frac{u^{n+1}}{F_{n+1}}\right) \\
&= \sum_{n\geq 0}\frac{u^n}{\tilde{F}_{n+1}} - \left(\sum_{n\geq 0}\frac{u^{n+1}}{F_{n+1}}\right)\frac{1}{1-\Phi-\Psi}(1) \\
&= \sum_{n\geq 0}\frac{u^n}{\tilde{F}_{n+1}} - \sum_{n\geq 0}\frac{u^{n+1}}{F_{n+1}}\,\overline{\mathbf{C}}(y).
\end{aligned}$$

This provides the expression of $\mathbf{C} = \overline{\mathbf{C}} - 1$ given in Proposition 4.5.

We can proceed similarly for solving the functional equation that defines the pyramid generating function $\mathbf{P}(x,y,u,v) \equiv \mathbf{P}(y)$. This equation can be written

$$\mathbf{P}(y) = \sum_{m\geq 0}\frac{u^m x^m y}{\tilde{F}_{m+1}\tilde{F}_{m+2}} + (\Phi+\Psi)\mathbf{P}(y)$$

where the operators $\Phi$ and $\Psi$ are now defined by

$$\Phi(\mathbf{S}(x,y,u,v)) = v\mathbf{S}\left(x, \frac{x}{1-y}, u, v\right) \qquad \text{and} \qquad \Psi(\mathbf{S}(x,y,u,v)) = -v\mathbf{S}(x,x,u,v).$$

Clearly, $\mathbf{P}(y) = \sum_{m\geq 0} u^m \mathbf{P}_m(y)$ where

$$\mathbf{P}_m(y) = \frac{x^m y}{\tilde{F}_{m+1}\tilde{F}_{m+2}} + (\Phi+\Psi)\mathbf{P}_m(y). \tag{4.9}$$

It is easy to prove by induction on $n$ that for $n \geq 1$ and $m \geq 0$,

$$\Phi^n\left(\frac{x^m y}{\tilde{F}_{m+1}\tilde{F}_{m+2}}\right) = \frac{v^n x^{m+1}\tilde{F}_n\tilde{F}_{n+1}}{\tilde{F}_{m+n+1}\tilde{F}_{m+n+2}}$$

and for $n \geq 0$ and $m \geq 0$,

$$\Psi\Phi^n\left(\frac{x^m y}{\tilde{F}_{m+1}\tilde{F}_{m+2}}\right) = -\frac{v^{n+1}x^{m+1}F_nF_{n+1}}{F_{m+n+1}F_{m+n+2}}.$$

Moreover, $\Phi(1) = v$ and $\Psi(1) = -v$, so that

$$\frac{1}{1-\Phi-\Psi}(1) = 1.$$

The iteration of Eq. (4.9) provides

$$\begin{aligned}
\mathbf{P}_m(x,y) &= \frac{1}{1-\Phi-\Psi}\left(\frac{x^m y}{\tilde{F}_{m+1}\tilde{F}_{m+2}}\right)\\
&= \sum_{n\geq 0}\Phi^n\left(\frac{x^m y}{\tilde{F}_{m+1}\tilde{F}_{m+2}}\right)+\frac{1}{1-\Phi-\Psi}\left(\sum_{n\geq 0}\Psi\Phi^n\left(\frac{x^m y}{\tilde{F}_{m+1}\tilde{F}_{m+2}}\right)\right)\\
&= \frac{x^m y}{\tilde{F}_{m+1}\tilde{F}_{m+2}}+\sum_{n\geq 1}\frac{v^n x^{m+1}\tilde{F}_n\tilde{F}_{n+1}}{\tilde{F}_{m+n+1}\tilde{F}_{m+n+2}}-\sum_{n\geq 0}\frac{v^{n+1}x^{m+1}F_nF_{n+1}}{F_{m+n+1}F_{m+n+2}}\frac{1}{1-\Phi-\Psi}(1)\\
&= \frac{x^m y}{\tilde{F}_{m+1}\tilde{F}_{m+2}}+\sum_{n\geq 1}\frac{v^n x^{m+1}\tilde{F}_n\tilde{F}_{n+1}}{\tilde{F}_{m+n+1}\tilde{F}_{m+n+2}}-\sum_{n\geq 1}\frac{v^{n}x^{m+1}F_{n-1}F_{n}}{F_{m+n}F_{m+n+1}}\\
&= \frac{x^m y}{\tilde{F}_{m+1}\tilde{F}_{m+2}}+\sum_{n\geq 1}v^n x^{m+1}\left[\frac{\tilde{F}_n\tilde{F}_{n+1}}{\tilde{F}_{m+n+1}\tilde{F}_{m+n+2}}-\frac{F_{n-1}F_n}{F_{m+n}F_{m+n+1}}\right].
\end{aligned}$$

This yields the second expression of $\mathbf{P}(x,y,u,v)$ given in Proposition 4.5.

### 4.3.3 Combinatorial interpretation

So far, the heaps we have considered were implicitly defined up to a translation, just as animals themselves are defined up to a translation. This will no longer be the case in this subsection, where the axis on which the minimal dimers lie will be graded.

In this context, it is convenient to consider heaps as words of a partially commutative monoid, as explained in [156]. Let $A\subset\mathbb{Z}$, and let $A^*$ be the free monoid on $A$, that is, the set of words $a_1\cdots a_k$ on the alphabet $A$, equipped with the concatenation product: $a_1\cdots a_k\circ b_1\cdots b_\ell = a_1\cdots a_k b_1\cdots b_\ell$. We consider the congruence on $A^*$ defined by the following partial commutations: for $i,j$ in $\mathbb{Z}$,

$$ij\equiv ji \iff |i-j|>1.$$

The elements of the quotient monoid $A^*/\equiv$ can be represented graphically by heaps of dimers above a graded axis: each letter $i$ corresponds to a dimer whose projection on the horizontal axis is centred at $i$.

The results concerning pyramids and half-pyramids stated in Proposition 4.5 are consequences of a simple inversion lemma, due to Viennot [156] that actually holds in any partially commutative monoid. Let $x_1,\dots,x_m$ be $m$ formal variables, and let us endow the letters of $A$ with a weight

$$w: A\mapsto\mathbb{R}[x_1,\dots,x_m].$$

The weight of a word is defined to be the product of the weights of its letters. We assume that the series $\sum_{u\in A^*}w(u)$ is a well-defined formal power series in the $x_i$. We denote by $|u|$ the number of letters (dimers) of a word $u$.

We say that a heap is *trivial* if all its dimers are minimal. Given $B\subseteq A$, we denote by $\mathcal{T}(B)$ the set of trivial heaps whose letters (or dimers) belong to $B$.

**Lemma 4.6 (Viennot).** *Let $B \subset A$. The generating function for heaps of $A^*/\equiv$ having all their minimal dimers in $B$ is*

$$\sum_{\substack{u \in A^*/\equiv \\ \min(u) \subset B}} w(u) = \frac{\sum_{T \in \mathcal{T}(A \setminus B)} (-1)^{|T|} w(T)}{\sum_{T \in \mathcal{T}(A)} (-1)^{|T|} w(T)}.$$

### Half-pyramids and pyramids

We first review briefly the derivation of the first three results of Proposition 4.5 based on the above lemma.

To recover the first result of Proposition 4.5, we note that any half-pyramid of width $\leq n$ lies on the segment $[1-n, 0]$, with its minimal dimer centred at 0, and so we take $A = [1-n, 0]$ and $B = \{0\}$. We choose $w(0) = y$ and $w(i) = x$ for $i < 0$. Using the recurrence relation satisfied by the Fibonacci polynomials, it is easy to check by induction on $n$ that for $n \geq 0$,

$$\sum_{T \in \mathcal{T}(A)} (-1)^{|T|} w(T) = \tilde{F}_{n+1}$$

and that

$$\sum_{T \in \mathcal{T}(A \setminus B)} (-1)^{|T|} w(T) = F_n.$$

By Lemma 4.6, the generating function for half-pyramids of minimal dimer 0 and width *at most* $n$ is $F_n/\tilde{F}_{n+1}$. This series counts, among others, the empty heap. The generating function for half-pyramids of minimal dimer 0 and width *exactly* $n$ is then

$$\frac{F_n}{\tilde{F}_{n+1}} - \frac{F_{n-1}}{\tilde{F}_n} = \frac{F_n \tilde{F}_n - F_{n-1}\tilde{F}_{n+1}}{\tilde{F}_n \tilde{F}_{n+1}} = \frac{x^{n-1} y}{\tilde{F}_n \tilde{F}_{n+1}},$$

where we have made use of (4.7).

For the second result of Proposition 4.5, we change the weights to $w(1-n) = y$ and $w(i) = x$ for $i > 1-n$. Then the generating function for half-pyramids of minimal dimer 0 and width *at most* $n$ is $\tilde{F}_n/\tilde{F}_{n+1}$, and the generating function for half-pyramids of minimal dimer 0 and width *exactly* $n$ is

$$\frac{\tilde{F}_n}{\tilde{F}_{n+1}} - \frac{F_{n-1}}{F_n} = \frac{x^{n-1} y}{F_n \tilde{F}_{n+1}}.$$

Finally, to evaluate the generating function of pyramids of left width $m$ and right width $n$, we take $A = [-m, n]$ and $B = \{0\}$. We choose $w(n) = y$ and $w(i) = x$ for $i < n$. Then the generating function for pyramids of minimal dimer 0, left width at most $m$ and right width at most $n$ is

$$\frac{F_{m+1} \tilde{F}_{n+1}}{\tilde{F}_{m+n+2}},$$

and the generating function for pyramids of minimal dimer 0, left width $m$ and right width $n$ is

$$\frac{F_{m+1}\tilde{F}_{n+1}}{\tilde{F}_{m+n+2}} - \frac{F_m\tilde{F}_{n+1}}{\tilde{F}_{m+n+1}} - \frac{F_{m+1}F_n}{F_{m+n+1}} + \frac{F_mF_n}{F_{m+n}}.$$

If we group the first two terms on the one hand, and the last two terms on the other hand, we obtain the second expression of $\mathbf{P}(x, y, u, v)$ given in Proposition 4.5, thanks to the following identity, valid for $m \geq 0$:

$$F_{m+1}\tilde{F}_{m+n+1} - F_m\tilde{F}_{m+n+2} = \begin{cases} yx^m & \text{if } n = 0, \\ x^{m+1}\tilde{F}_n & \text{otherwise.} \end{cases}$$

If we group the first and third terms on the one hand, and the second and fourth on the other hand, we obtain the first expression of $\mathbf{P}(x, y, u, v)$ given in Proposition 4.5, thanks to the following identity, valid for $m \geq -1, n \geq 0$:

$$\tilde{F}_{n+1}F_{m+n+1} - F_n\tilde{F}_{m+n+2} = x^n yF_m.$$

### Connected heaps

Let us now give a combinatorial explanation of the connected heap result. Let $\overline{\mathbf{C}}(x, y, u) = \mathbf{C}(x, y, u)+1$ denote the generating function for (possibly empty) connected heaps, considered up to a translation, then Proposition 4.5 states that

$$\sum_{n\geq 0} \frac{u^n}{\tilde{F}_{n+1}} = \overline{\mathbf{C}}(x, y, u) \sum_{m\geq 0} \frac{u^m}{F_m}. \tag{4.10}$$

If we scan any general heap from left to right, we find that it is made up of several connected heaps, with some non-zero number of empty columns between them. The above identity, roughly speaking, comes from the fact that any general heap can be split into its rightmost connected component and a remaining smaller heap. Let us make this idea more precise.

Take $A = [0, n-1]$, $w(n-1) = y$ and $w(i) = x$ for $i < n-1$. Then $1/\tilde{F}_{n+1}$ is the generating function for all heaps on the alphabet $A$. We shall describe a bijection $f$ that maps any heap $H$ on the alphabet $A$, onto a triple $(m, H', C)$, where $0 \leq m \leq n$, $H'$ is a heap on the alphabet $\{0, 1, \ldots, m-2\}$, and $C$ is a connected heap of width $n-m$ whose rightmost dimers are centred at $n-1$. Moreover, this bijection will be such that the weight of $H$ is the product of the weights of $H'$ and $C$. The existence of such a bijection implies (4.10).

We now describe $f$ in details (Fig. 4.16). We take $C$ to be the connected component of $H$ that contains all the letters $n-1$ occurring in $H$. If there is no such letter in $H$, then $C$ is empty. Let $n-m$ be the width of $C$, with $0 \leq m \leq n$. Having deleted $C$ from $H$, we are left with a heap $H'$ on the alphabet $[0, m-2]$. We take $f(H) = (m, H', C)$.

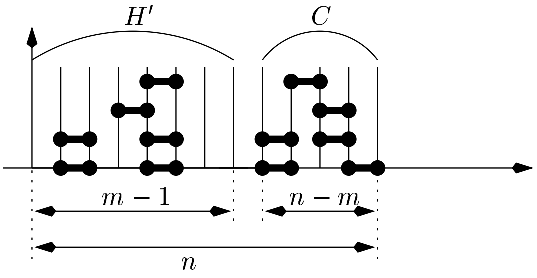



Figure 4.16: The combinatorial interpretation for connected heaps.

## 4.4 Analytic structure of the generating functions, and asymptotic results

It is natural to ask whether one can recover the simple form of the results of Section 4.2 from the more complicated form of the results of Proposition 4.5. This *is* possible, and in order to do so we have to express the Fibonacci polynomials in terms of the generating function for Catalan numbers:

$$Q = \frac{1 - 2x - \sqrt{1-4x}}{2x} = \sum_{n\geq 1} \frac{1}{n+1}\binom{2n}{n} x^n, \tag{4.11}$$

which is also the generating function for half-pyramids. The recursion defining the Fibonacci polynomials is equivalent to

$$\sum_{n\geq 0} F_n u^n = \frac{1}{1-u+xu^2}.$$

Expanding this rational function in partial fractions of $u$ and taking the coefficient of $u^n$ yields, for $n \geq 0$,

$$F_n = \frac{1-Q^{n+1}}{(1-Q)(1+Q)^n}. \tag{4.12}$$

Moreover, $x$ and $Q$ are related by

$$x = \frac{Q}{(1+Q)^2}. \tag{4.13}$$

Using these identities, we can recover from Proposition 4.5 the results of Section 4.2 about half-pyramids and pyramids. For instance, the generating function for half-pyramids is

$$\begin{aligned}
\mathbf{Q}(x,x,1) &= \sum_{n\geq 1} \frac{x^n}{F_n F_{n+1}} \\
&= (1-Q^2)\sum_{n\geq 1} \frac{Q^n(1-Q)}{(1-Q^{n+1})(1-Q^{n+2})} \\
&= (1-Q^2)\sum_{n\geq 1} \left(\frac{Q^n}{1-Q^{n+1}} - \frac{Q^{n+1}}{1-Q^{n+2}}\right) \\
&= (1-Q^2)\frac{Q}{1-Q^2} = Q, \qquad (4.14)
\end{aligned}$$

as stated in Proposition 4.1.

### 4.4.1 Connected heaps and saw-tooth heaps

More generally, all the series of Proposition 4.5 can be expressed in terms of $u, v, y$ and $Q$, using Eqs. (4.12) and (4.13). In particular, we can now complete the enumeration of connected heaps and saw-tooth heaps.

**Proposition 4.7.** *Let $Q(x) \equiv Q$ denote the generating function for half-pyramids. Then the generating function for connected heaps of dimers is*

$$\mathbf{C}(x,x,1) = \frac{Q}{(1-Q)\left[1 - \sum_{k\geq 1} \frac{Q^{k+1}}{1-Q^k(1+Q)}\right]},$$

*as stated in Proposition 4.3. Consequently it is not D-finite. The number of connected heaps having $n$ dimers (or multi-directed animals having $n$ cells) is asymptotic to $A\mu^n$ for some positive constant $A$, with $\mu = 4.58789436...$ The average width of these heaps is asymptotic to $Bn$ for some positive constant $B$.*

*The generating function $\mathbf{N}(x,x,1)$ for saw-tooth heaps of dimers is*

$$\mathbf{N}(x,x,1) = (1-Q)\sum_{n\geq 1} \frac{Q^n}{(1+Q)^{n-1}(1-Q^{n+2})} \prod_{i=1}^{n-1}\left(1 + \frac{(1+Q)(1-Q^{i+1})}{1-Q^{i+2}}\right),$$

*and is not D-finite either. The number of saw-tooth heaps having $n$ dimers is asymptotic to $A\mu^n$ for some positive constant $A$, with $\mu = \sqrt{5}+2 = 4.236....$ The average width of these heaps is asymptotic to $Bn$ for some positive constant $B$.*

*Proof.* Let us start from the following expression of $\mathbf{C}(x,x,u)$ derived from Proposition 4.5 (remembering that $\tilde{F}_n(x,x) = F_n$):

$$\begin{aligned}\mathbf{C}(x,x,u) &= \frac{1}{u} - 1 - \left[\sum_{n\geq 0} \frac{u^{n+1}}{F_n}\right]^{-1} \\ &= \frac{1}{u} - 1 - \frac{1}{1-Q}\left[\sum_{n\geq 1} \frac{u^n(1+Q)^{n-1}}{1-Q^n}\right]^{-1}. \end{aligned} \tag{4.15}$$

We cannot replace $u$ by 1 directly; let us isolate the part that becomes singular at $u=1$:

$$\begin{aligned}\mathbf{C}(x,x,u) &= \frac{1}{u} - 1 - \frac{1}{1-Q}\left[\sum_{n\geq 1} u^n(1+Q)^{n-1}\left(1+\frac{Q^n}{1-Q^n}\right)\right]^{-1} \\ &= \frac{1}{u} - 1 - \frac{1-u(1+Q)}{u(1-Q)}\left[1 + (1-u(1+Q))\sum_{n\geq 1} \frac{u^{n-1}Q^n(1+Q)^{n-1}}{1-Q^n}\right]^{-1}.\end{aligned}$$

Hence, at $u=1$,

$$\mathbf{C}(x,x,1) = \frac{Q}{(1-Q)\left[1-\sum_{n\geq 1}\frac{Q^{n+1}(1+Q)^{n-1}}{1-Q^n}\right]}.$$

Finally, to recover the expression of Proposition 4.7, we observe that

$$\begin{aligned}\sum_{n\geq 1}\frac{Q^{n+1}(1+Q)^{n-1}}{1-Q^n} &= \sum_{n\geq 1} Q^{n+1}(1+Q)^{n-1}\sum_{k\geq 0} Q^{nk} \\ &= \frac{Q}{1+Q}\sum_{k\geq 0}\sum_{n\geq 1}\left(Q^{k+1}(1+Q)\right)^n \\ &= \sum_{k\geq 1}\frac{Q^{k+1}}{1-Q^k(1+Q)}.\end{aligned}$$

Let us prove that $\mathbf{C}(x,x,1)$ is not D-finite. Let $f(q) = q/[(1-q)g(q)]$, where

$$g(q) = 1 - \sum_{k\geq 1}\frac{q^{k+1}}{1-q^k(1+q)}.$$

Hence $\mathbf{C}(x,x,1) = f(Q(x))$, where $Q(x)$ is the algebraic function of $x$ given by (4.11). Conversely,

$$f(q) = \mathbf{C}(q/(1+q)^2, q/(1+q)^2, 1).$$

It is not difficult to see that if $F(x)$ is D-finite and $G(x)$ is algebraic, then $F(G(x))$ is still D-finite [144, Theorem 2.7]. Hence $\mathbf{C}(x,x,1)$ is D-finite if and only if $f(q)$ is D-finite. We will show that this is not the case.

The series $g(q)$ is meromorphic on $\{q : |q| < 1\}$, with simple poles at all values of $q$ satisfying

$$q^k(1+q) = 1, \quad |q| < 1.$$

These values can be seen to accumulate on $\{e^{i\theta}, -2\pi/3 \leq \theta \leq 2\pi/3\}$ as $k \to \infty$ (Fig. 4.17). Hence $f(q)$ is meromorphic on $\{q : |q| < 1\}$, and has an accumulation of zeroes on $\{e^{i\theta}, -2\pi/3 \leq \theta \leq 2\pi/3\}$. Consequently, any point of this set is a singularity of $f$ (an analytic continuation of $f$ would be zero on a portion of the unit circle, which is impossible for a non-trivial holomorphic function). As a D-finite function has only finitely many singularities, $f(q)$ cannot be D-finite.

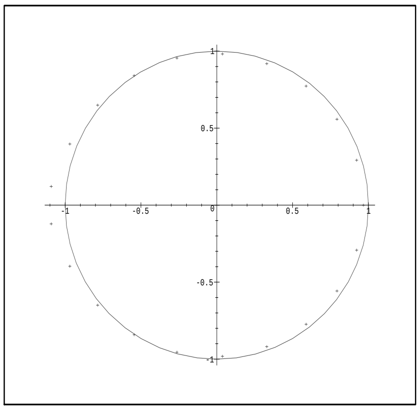


Figure 4.17: The zeroes of $q^k(1+q) - 1$ for $k = 20$.

Let us write $\mathbf{C}(x, x, 1) = \sum_n c_n x^n$, so that $c_n$ is the number of connected heaps having $n$ dimers. In order to determine the asymptotic behaviour of $c_n$, we need to study the *dominant* singularities of $\mathbf{C}(x, x, 1) = f(Q(x))$, that is, the singularities of smallest modulus. The singularities of $\mathbf{C}(x, x, 1)$ are to be found among the singularities of $Q(x)$ and the values of $x$ such that $Q(x)$ is a singularity of $f$. The series $Q(x)$ has a unique singularity at $x = 1/4$. What about $f$?

The series $f(q)$ is meromorphic on $|q| < 1$. Its singularities in this domain are isolated poles corresponding to the zeroes of $g$. The series $1 - g(q) = \sum_{k\geq 1} \frac{q^{k+1}}{1-q^k(1+q)}$ has positive coefficients, is holomorphic on $|q| < (\sqrt{5}-1)/2$ and tends to infinity as $q$ tends to $(\sqrt{5}-1)/2$. This implies that $g(q)$ has a unique zero of minimal modulus, denoted $q_c$, which can be estimated numerically: $q_c = 0.4727898832...$ Moreover, this zero is simple: $g'(q_c) < 0$. This zero is the unique dominant singularity of $f$. It is a simple pole.

Now $Q(x)$ has positive coefficients, $Q(0) = 0$ and $Q(1/4) = 1$. Therefore there exists $x_c \in [0, 1/4]$ such that $Q(x_c) = q_c$. This implies that $\mathbf{C}(x, x, 1) = f(Q(x))$ has a unique dominant singularity, $x_c = q_c/(1+q_c)^2 = 0.2179649141...$, which is a simple isolated pole. It follows that $c_n \sim A(4.58789436...)^n$ as $n \to \infty$.

Finally, let us study the average width of connected heaps. We differentiate the expression of $\mathbf{C}(x, x, u)$ given in Eq. (4.15) with respect to $u$. We apply to this series the same treatment that led to the expression of $\mathbf{C}(x,x,1)$. We obtain:

$$\begin{aligned}\mathbf{C}'_u(x,x,1) &= \frac{1}{1-Q}\frac{1+\sum_{n\geq 1}\frac{nQ^{n+2}(1+Q)^{n-1}}{1-Q^n}}{\left[1-\sum_{n\geq 1}\frac{Q^{n+1}(1+Q)^{n-1}}{1-Q^n}\right]^2}-1 \\ &= \frac{1}{1-Q}\frac{1+\sum_{k\geq 1}\frac{Q^{k+2}}{(1-Q^k(1+Q))^2}}{\left[1-\sum_{k\geq 1}\frac{Q^{k+1}}{1-Q^k(1+Q)}\right]^2}-1. \qquad (4.16)\end{aligned}$$

We have seen that $\mathbf{C}(x,x,1)$ has a unique dominant singularity, which is a simple pole, at $x_c = q_c/(1+q_c)^2$, where $q_c$ is the smallest positive solution of $1=\sum_{k\geq 1}\frac{q^{k+1}}{1-q^k(1+q)}$. Similarly, we derive from (4.16) that $\mathbf{C}'_u(x,x,1)$ has a unique dominant singularity, which is a double pole, at $x_c$. Hence the coefficient of $x^n$ in $\mathbf{C}'_u(x,x,1)$ is asymptotic to $Bnx_c^{-n}$. The result follows.

Let us now turn our attention to saw-tooth animals. We obtain the expression of $\mathbf{N}(x,x,1)$ from Proposition 4.5 by replacing $x$ and the Fibonacci polynomials $F_i$ by their expressions in terms of $Q$ (see Eqns. (4.12) and (4.13)). This leads us to introduce the following power series in the formal variable $q$:

$$\begin{aligned}f(q) &= (1-q)\sum_{n\geq 1}\frac{q^n}{(1+q)^{n-1}(1-q^{n+2})}\prod_{i=1}^{n-1}\left[1+\frac{(1+q)(1-q^{i+1})}{1-q^{i+2}}\right] \\ &= (1-q)\sum_{n\geq 1}\frac{q^n(2+q)^{n-1}}{(1+q)^{n-1}(1-q^{n+2})}\prod_{i=1}^{n-1}\left[1-\frac{(1-q^2)q^{i+1}}{(2+q)(1-q^{i+2})}\right].\end{aligned}$$

Again, $\mathbf{N}(x,x,1)$ is D-finite if and only if $f(q)$ is D-finite. We are going to prove that this is not the case, as $f(q)$ is meromorphic inside the unit circle, with infinitely many poles.

We first want to put aside the product in the expression of $f(q)$. The equation $(1-q^2)q^{i+1} = (2+q)(1-q^{i+2})$ can be rewritten

$$q^i = \frac{1+2q^{-1}}{1+2q}.$$

It is not difficult to see that $|1+2q^{-1}|\geq|1+2q|$ if and only if $|q|\leq 1$. This implies that for $i\geq 0$, all roots of the above equation lie on the unit circle. Moreover, for $|q|<1$, the series $\sum_i q^{i+1}/(1-q^{i+2})$ is convergent. Consequently, as $n$ goes to infinity,

$$\lim_{n\to\infty}\prod_{i=1}^{n-1}\left[1-\frac{(1-q^2)q^{i+1}}{(2+q)(1-q^{i+2})}\right] = \prod_{i\geq 1}\left[1-\frac{(1-q^2)q^{i+1}}{(2+q)(1-q^{i+2})}\right]\equiv W(q),$$

and the function $W(q)$ is a holomorphic function with no zero inside the unit circle. Hence, for $|q| < 1$, we can write

$$\begin{aligned} f(q) &= (1-q)W(q)\sum_{n\geq 1}\frac{q^n(2+q)^{n-1}}{(1+q)^{n-1}(1-q^{n+2})}\prod_{i\geq 1}\left[1-\frac{(1-q^2)q^{n+i}}{(2+q)(1-q^{n+i+1})}\right]^{-1} \\ &= (1-q)W(q)\sum_{n\geq 1}\frac{q^n(2+q)^{n-1}}{(1+q)^{n-1}}g(q^n,q), \end{aligned}$$

where

$$g(z,q) = \frac{1}{1-zq^2}\prod_{i\geq 1}\left[1-\frac{(1-q^2)zq^i}{(2+q)(1-zq^{i+1})}\right]^{-1}$$

is a holomorphic function of $z$ for $|z| < 1$. Let us expand $g(z,q)$ in $z$:

$$g(z,q) = \sum_{k\geq 0} z^k g_k(q),$$

where $g_k(q)$ is a rational function of $q$, of denominator $(2+q)^k$, that remains positive on the interval $(0,1)$. In the expression of $f(q)$, we replace $g(q^n,q)$ by its expansion, exchange the summations on $n$ and $k$, perform the summation on $n$ and end up with

$$f(q) = (1-q)W(q)\sum_{k\geq 0}\frac{g_k(q)q^{k+1}}{1-\frac{q^{k+1}(2+q)}{1+q}}.$$

We can now work out the analytic structure of $f$. The above expression shows that $f(q)$ is meromorphic inside the unit circle and has only simple poles, that are to be found among the roots of $q^{k+1}(2+q) = 1+q$, $k \geq 0$. For $k \geq 0$, the solution of smallest modulus of the equation $q^{k+1}(2+q) = 1+q$, denoted $q_k$, belongs to $[0,1]$, and increases to 1 as $k$ goes to infinity. As $g_k(q) > 0$ for $q \in (0,1)$, we see that each $q_k$ is indeed a simple pole of $f$. Hence $f$ has infinitely many singularities, and cannot be D-finite.

The radius of $f$ is given by its smallest pole, $q_0 = (\sqrt{5}-1)/2$. As the only singularity of $Q(x)$ is at $x = 1/4$, this implies that $\mathbf{N}(x,x,1)$ has a unique dominant singularity, $x_c = q_0/(1+q_0)^2 = \sqrt{5}-2$, which is a simple pole. Hence the number of $n$-celled saw-tooth animals is asymptotic to $A(\sqrt{5}+2)^n$.

The study of the average width of saw-tooth heaps is extremely close to what we have already done. With the notations used above, we have

$$\begin{aligned} \mathbf{N}'_u(x,x,1) &= (1-Q)\sum_{n\geq 1}\frac{nQ^n(2+Q)^{n-1}}{(1+Q)^{n-1}(1-Q^{n+2})}\prod_{i=1}^{n-1}\left[1-\frac{(1-Q^2)Q^{i+1}}{(2+Q)(1-Q^{i+2})}\right] \\ &= (1-Q)W(Q)\sum_{n\geq 1}\frac{nQ^n(2+Q)^{n-1}}{(1+Q)^{n-1}}g(Q^n,Q), \\ &= (1-Q)W(Q)\sum_{k\geq 0}\frac{g_k(Q)Q^{k+1}}{\left(1-\frac{Q^{k+1}(2+Q)}{1+Q}\right)^2}. \end{aligned}$$

From this, we conclude that $\mathbf{N}'_u(x,x,1)$ has a unique dominant singularity, which is a double pole at $x_c = \sqrt{5} - 2$. The result follows. ∎

### 4.4.2 Half-pyramids and pyramids: what parameters are D-finite?

Thanks to the results obtained in the previous section on the enumeration of heaps of given width, we can show that certain series are neither algebraic nor D-finite; the generating function for pyramids counted by their width and number of dimers for instance. In what follows, the derivation of series with respect to the variable $a$ is denoted $D_a$. Recall the definition of the series $\mathbf{Q}(x,y,u), \overline{\mathbf{Q}}(x,y,u)$ and $\mathbf{P}(x,y,u,v)$, illustrated in Fig. 4.13.

**Proposition 4.8.** *The series $\mathbf{Q}(x,y,1)$ and $\mathbf{P}(x,y,1,v)$ are algebraic.*
*The series $D_u\mathbf{Q}(x,x,1)$ and $D_y\overline{\mathbf{Q}}(x,0,1)$ are not D-finite. Consequently, the series $\mathbf{Q}(x,x,u)$ (counting half-pyramids by their width and number of dimers) and $\overline{\mathbf{Q}}(x,y,1)$ (counting half-pyramids by their number of dimers, distinguishing those in the leftmost column) are not D-finite either.*
*The series $u\mathbf{P}(x,x,u,u)$, which counts pyramids by their width and number of dimers, is not D-finite, although its derivative with respect to $u$, evaluated at $u=1$, is algebraic.*

*Proof.* To prove the first two results, we use the factorisation of (general) half-pyramids and pyramids described in Section 4.2. For half-pyramids, we have to add to the three cases of Fig. 4.6 a fourth case, where there is a dimer directly above the minimal one. We obtain:

$$\begin{aligned} bq(x,y,1) &= y + y\mathbf{Q}(x,y,1) + y\mathbf{Q}(x,x,1) + y\mathbf{Q}(x,x,1)\mathbf{Q}(x,y,1) \\ &= y(1+\mathbf{Q}(x,x,1))(1+\mathbf{Q}(x,y,1)). \end{aligned}$$

Then, Fig. 4.7 gives the following equation for pyramids:

$$\mathbf{P}(x,y,1,v) = \mathbf{Q}(x,y,1) + v\mathbf{Q}(x,x,1)\mathbf{P}(x,y,1,v).$$

From these two equations, we obtain

$$\mathbf{Q}(x,x,1) = Q, \quad \mathbf{Q}(x,y,1) = \frac{y(1+Q)}{1-y(1+Q)} \quad \text{and } P(x,y,1,v) = \frac{y(1+Q)}{[1-y(1+Q)][1-vQ]}.$$

We could also derive these results from the expressions of Proposition 4.5, as we have done at the beginning of the section for $\mathbf{Q}(x,x,1)$, but this would be more tedious and less combinatorial.

Let us now focus on the "negative" results. From Proposition 4.5,

$$D_u\mathbf{Q}(x,x,1) = \sum_{n\geq 1} \frac{nx^n}{F_n F_{n+1}}.$$

We replace $x$ and $F_n$ by their expressions in terms of $Q$ and obtain, after a few reductions comparable to (4.14):

$$D_u\mathbf{Q}(x,x,1) = (1-Q^2)\sum_{n\geq 1} \frac{Q^n}{1-Q^{n+1}}.$$

Similarly, we compute

$$D_y\overline{\mathbf{Q}}(x,0,1) = [y]\overline{\mathbf{Q}}(x,y,1) = (1-Q^2)^2 \sum_{n\geq 1} \frac{Q^{n-1}}{(1-Q^{n+1})^2}.$$

In order to prove that these series are not D-finite, we have to prove that the divisor generating functions

$$\sum_{n\geq 1} \frac{q^n}{1-q^n} = \sum_{m\geq 1} q^m \sum_{d|m} 1 \qquad \text{and} \qquad \sum_{n\geq 1} \frac{q^n}{(1-q^n)^2} = \sum_{m\geq 1} q^m \sum_{d|m} d$$

are not D-finite. These series have integer coefficients, and have radius of convergence 1. By a theorem of Pólya-Carlson (a series with integer coefficients that converges inside the unit disk is either rational or has the unit circle as a natural boundary [41]), these series are either rational, or not D-finite. But these series are actually known to be irrational (this can be proven [3] by adapting an argument of [2]). Hence they are not D-finite.

Let us now study pyramids. Let $\overline{\mathbf{P}}(x,u) = u\mathbf{P}(x,x,u,u)$ be their generating function according to their number of dimers and width. The series $D_u\overline{\mathbf{P}}(x,1)$ counts pyramids where a column is marked, and *is algebraic*. This can be derived from the expression of $\mathbf{P}(x,y,u,v)$ given in Proposition 4.5, but is more easily obtained via the factorisation of pyramids described in Fig. 4.18. This factorisation gives:

$$\begin{aligned} D_u\overline{\mathbf{P}}(x,1) &= \overline{\mathbf{P}}(x,1) + 2\overline{\mathbf{P}}(x,1)^2 \\ &= \frac{Q}{1-Q} + 2\frac{Q^2}{(1-Q)^2} \\ &= \frac{Q(1+Q)}{(1-Q)^2}. \end{aligned}$$

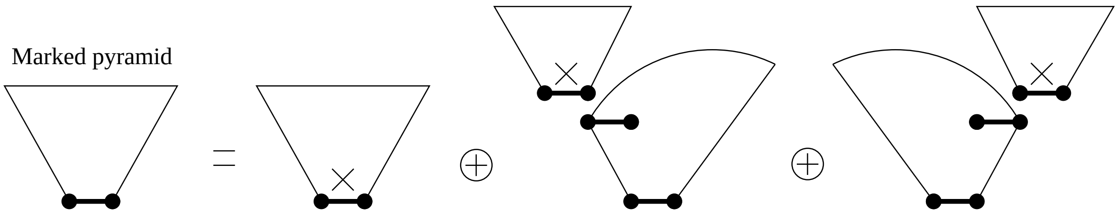



Figure 4.18: The generating function for pyramids with a marked column is algebraic.

From the first expression of $\mathbf{P}(x,y,u,v)$ given in Proposition 4.5, we find that the generating function $\mathbf{P}_N$ for pyramids of width $N$ is

$$\mathbf{P}_N = [u^N]\overline{\mathbf{P}}(x,u) = \sum_{m+n=N-1} \frac{Q^{n+1}(1-Q^{m+1})}{1-Q^{N+1}} \left[\frac{1-Q^{m+2}}{1-Q^{N+2}} - \frac{1-Q^m}{1-Q^N}\right].$$

This sum can be easily evaluated:

$$\begin{aligned}\mathbf{P}_N &= \frac{(1-Q^2)Q^N}{(1-Q^N)(1-Q^{N+1})(1-Q^{N+2})}\left[N(1+Q^{N+1})-2q\frac{1-Q^N}{1-Q}\right]\\ &= \frac{Q^N(1+Q)}{1-Q}\left[\frac{N}{1-Q^N}-2\frac{Q(N+1)}{1-Q^{N+1}}+\frac{Q^2(N+2)}{1-Q^{N+2}}\right].\end{aligned}$$

The series $\overline{\mathbf{P}}(x,u)$ can be seen as a formal power series in $u$ with rational coefficients in $x$:

$$\overline{\mathbf{P}}(x,u)=\sum_{N\geq 1}\mathbf{P}_N(x)u^N.$$

If it were D-finite, so would be the following series:

$$\mathbf{S}(q,u)=\overline{\mathbf{P}}\left(\frac{q}{(1+q)^2},u\right)=\sum_{N\geq 1}\mathbf{S}_N(q)u^N$$

with

$$\mathbf{S}_N(q)=\frac{q^N(1+q)}{1-q}\left[\frac{N}{1-q^N}-2\frac{q(N+1)}{1-q^{N+1}}+\frac{q^2(N+2)}{1-q^{N+2}}\right].$$

For $N\geq 3$, observe that any $N$th primitive root of 1 is a pole of $\mathbf{S}_N(q)$. This property, combined with the following theorem, shows that $\mathbf{S}(q,u)$, and hence $\overline{\mathbf{P}}(x,u)=u\mathbf{P}(x,x,u,u)$, cannot be D-finite. ∎

**Theorem 4.9.** *Let $S(q,u)=\sum_n S_n(q)u^n$ be a formal power series in $u$ with coefficients in $\mathbb{C}(q)$. Assume that $S(q,u)$ is D-finite in its main variable $u$. For $n\geq 0$, let $\mathcal{P}_n$ be the set of poles of $S_n(q)$, and let $\mathcal{P}=\cup_n\mathcal{P}_n$. Then $\mathcal{P}$ has only a finite number of limit points.*

*Proof.* By assumption, the sequence $S_n(q)$ satisfies a linear recurrence relation with coefficients in $\mathbb{C}[q,n]$:

$$a_0(q,n)S_n(q)=a_1(q,n)S_{n-1}(q)+\cdots+a_k(q,n)S_{n-k}(q),$$

for $n\geq n_0$, with $a_i(q,n)\in\mathbb{C}[q,n]$, and $a_0(q,n)\not\equiv 0$. Consequently, $S_n(q)$ can be written as a rational function of denominator

$$D_n(q)=I(q)\prod_{m=1}^{n}a_0(q,m),$$

where $I(q)$ is a polynomial accounting for the denominators of the $S_m(q)$, $m<n_0$. Let $\ell\in\mathbb{C}$ be a limit point of the set $\mathcal{P}$: there exists a sequence $(q_i,n_i)$ such that $q_i\to\ell$, $n_i\to\infty$ and $a_0(q_i,n_i)=0$ for all $i$. Let the degree of $a_0(q,n)$ in $n$ be $d$, and let us write

$$a_0(q,n)=\sum_{k=0}^{d}b_k(q)n^k.$$

Then

$$0 = \frac{a_0(q_i, n_i)}{n_i^d} = \sum_{k=0}^{d} b_k(q_i) n_i^{k-d},$$

and this converges to $b_d(\ell)$ as $i$ goes to infinity. Hence all limit points of $\mathcal{P}$ are roots of the polynomial $b_d(q)$. This concludes the proof. ∎

4.5

## Summary of the results.

We have enumerated exactly two new families of square lattice animals: stacked directed animals and multi-directed animals. Both include directed animals. Each of these families is in one-to-one correspondence with a natural set of (strict) heaps of dimers. The corresponding growth constants are 3.5 and 3.58... respectively, improving on the growth constant 3 of directed animals.

By removing the condition that heaps should be strict, we have enumerated exactly two analogous families of triangular lattice animals. The growth constants are 4.5 and 4.58... respectively, which should be compared to the growth constant 4 of directed animals on the triangular lattice. We have also enumerated a third new class of animals, called saw-tooth animals, that have a growth constant of $2 + \sqrt{5} = 4.2...$. They have no simple square lattice counterpart and are a bit more artificial than the other two classes.

We have used two kinds of recursive constructions for heaps of dimers: the first one concatenates two heaps, and is well-suited to the enumeration of directed animals and stacked directed animals. The second builds the heap column-by-column (like the Temperley method of chapter 5), and is well-suited to the enumeration of directed animals and multi-directed animals.

We have highlighted the analytic structure of the generating functions, and the nature of their singularities. The generating function for stacked directed animals is algebraic, while the generating function for multi-directed animals is not D-finite. Both have a simple pole as their dominant singularity. We have also proved that certain series, like the generating function for directed animals counted by their area and width, are not D-finite.

Our results are summarised in Table 2 below. The data in the last line of this table are estimations: see [44, 99] for the square lattice growth constant, [96] for the average width, and [84, 160] for the triangular lattice growth constant. The number of $n$-celled animals in each family, for small values of $n$, is given in Tables 3 (square lattice) and 4 (triangular lattice).

Let us conclude by mentioning an algebraic parameter, defined on directed animals, that seems to resist the two constructions we have used in this work. Let $A$ be a directed animal on the square lattice. We say that a cell $c$ of $A$ is *only supported on the right* if the south-east neighbour of $c$ belongs to $A$, but not its south-west neighbour. Similarly, in a triangular lattice directed animal $A$, we say that a cell $c$ is *only supported on the right* if, in addition

| Model | Growth constant | Average width | Nature of the GF |
|---|---|---|---|
| Column convex<br>polyominoes | 3.20... □<br>3.86... △ | $n$ | rational (Temperley [151])<br>rational (Klarner [105]) |
| Directed animals<br>(pyramids) | 3 □<br>4 △ | $\sqrt{n}$<br>(Gouyou-Viennot [76]) | quadratic (Dhar [54]) |
| Stacked directed animals<br>(stacked pyramids) | 3.5 □<br>4.5 △ | $O(n)$ | quadratic<br>(Proposition 4.2) |
| Saw-tooth animals<br>(saw-tooth heaps) | <br>4.23... △ | $n$ | not D-finite<br>(Proposition 4.7) |
| Multi-directed animals<br>(connected heaps) | 3.58... □<br>4.58... △ | $n$ | not D-finite<br>(Proposition 4.7) |
| All animals | 4.06... □<br>5.18... △ | $n^{0.64...}$ | ? |

Table 4.2: Comparison of the various models.

to the above two conditions, the south neighbour of $c$ does not belong to $A$. For instance, the first animal of Fig. 4.4 has 3 cells only supported on the right, while the second animal has 2 such cells.

It has been proved, using a formal connection between directed animals and one-dimensional gas models, that the generating function for directed animals, counted by their area and number of cells only supported on the right, is algebraic, both on the square and triangular lattices [25]. But this result has, so far, resisted the approach based on factorisations of heaps. Moreover, it seems that the column-by-column construction of Section 4.3 is not well-suited either. We hope that eventually, a transparent, combinatorial proof will be found for this apparently simple result. The new recent construction of Shapiro [143] is nice and simple, but does not seem well-adapted to this parameter, at least at first sight.

| *cells* | *directed* | *column − convex* | *stacked dir.* | *multi − directed* | *all* |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 5 | 6 | 6 | 6 | 6 |
| 4 | 13 | 19 | 19 | 19 | 19 |
| 5 | 35 | 61 | 63 | 63 | 63 |
| 6 | 96 | 196 | 213 | 214 | 216 |
| 7 | 267 | 629 | 729 | 738 | 760 |
| 8 | 750 | 2017 | 2513 | 2571 | 2725 |
| 9 | 2123 | 6466 | 8703 | 9020 | 9910 |
| 10 | 6046 | 20727 | 30232 | 31806 | 36446 |
| 11 | 17303 | 66441 | 105236 | 112572 | 135268 |
| 12 | 49721 | 212980 | 366849 | 399548 | 505861 |
| 13 | 143365 | 682721 | 1280131 | 1421145 | 1903890 |
| 14 | 414584 | 2188509 | 4470354 | 5063254 | 7204874 |
| 15 | 1201917 | 7015418 | 15619386 | 18062902 | 27394666 |
| 16 | 3492117 | 22488411 | 54595869 | 64505148 | 104592937 |
| 17 | 10165779 | 72088165 | 190891131 | 230547424 | 400795844 |
| 18 | 29643870 | 231083620 | 667590414 | 824547052 | 1540820542 |
| 19 | 86574831 | 740754589 | 2335121082 | 2950565215 | 5940738676 |
| 20 | 253188111 | 2374540265 | 8168950665 | 10562978104 | 22964779660 |

Table 4.3: Square lattice data for the indicated families of animals enumerated according to their area.

| *cells* | *directed* | *column − convex* | *stacked directed* | *multi − directed* | *all* |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 3 | 3 | 3 | 3 | 3 |
| 3 | 10 | 11 | 11 | 11 | 11 |
| 4 | 35 | 42 | 44 | 44 | 44 |
| 5 | 126 | 162 | 184 | 184 | 186 |
| 6 | 462 | 626 | 789 | 790 | 814 |
| 7 | 1716 | 2419 | 3435 | 3450 | 3652 |
| 8 | 6435 | 9346 | 15100 | 15242 | 16689 |
| 9 | 24310 | 36106 | 66806 | 67895 | 77359 |
| 10 | 92378 | 139483 | 296870 | 304267 | 362671 |
| 11 | 352716 | 538841 | 1323318 | 1369761 | 1716033 |
| 12 | 1352078 | 2081612 | 5911972 | 6188002 | 8182213 |
| 13 | 5200300 | 8041537 | 26455294 | 28031111 | 39267086 |
| 14 | 20058300 | 31065506 | 118528793 | 127253141 | 189492795 |
| 15 | 77558760 | 120010109 | 531540891 | 578694237 | 918837374 |
| 16 | 300540195 | 463614741 | 2385375732 | 2635356807 | 4474080844 |
| 17 | 1166803110 | 1791004361 | 10710619014 | 12015117401 | 21866153748 |
| 18 | 4537567650 | 6918884013 | 48112492938 | 54831125131 | 107217298977 |
| 19 | 17672631900 | 26728553546 | 216195753066 | 250418753498 | 527266673134 |
| 20 | 68923264410 | 103255896932 | 971744791032 | 1144434017309 | 2599804551168 |

Table 4.4: Triangular lattice data for the indicated families of animals enumerated by area. It is interesting to note that even though directed animals have a larger connective constant than column-convex animals, there are more $n$-celled column-convex animals than $n$-celled directed animals up until $n = 43$.

# CHAPTER 5

# Building with sequences

# 5.1 Building objects with blocks

One of the most successful methods for the enumeration of polyominoes is a column-by-column construction which is sometimes called the "Temperley Method". All polyominoes for which the generating function is known in closed form (with the exception of convex polygons and directed polyominoes in three dimensions) have solutions that can be found using this technique[1]. The basic idea of Temperley's approach is quite simple; for example, a column-convex polygon can be built up column-by-column by successively "gluing" columns together. This construction is translated into a recurrence for the number of polygons (or a functional equation satisfied by the generating function), which can then be solved.

More generally, there are a wide range of objects that we can consider to be constructed from a sequence of simpler objects that we will call *building blocks*. Take, for example, the following families of lattice objects:

- every bargraph polygon can be mapped to a sequence of connected columns of cells,
- every column-convex polygon can be uniquely mapped to a sequence of connected columns of cells,
- every self-avoiding walk can be uniquely mapped to a sequence of the four possible steps on the square lattice; in the $\pm x$ and $\pm y$ directions.

More formally: Let $\mathcal{O}$ be the set of (lattice) objects that we are studying, and let $\mathcal{P}$ be the set of unrestricted sequences from which the objects are constructed. There exists a decomposition or "ungluing" function, $\mathfrak{U}$, that maps an element $A \in \mathcal{O}$ to a *unique* sequence $\rho \in \mathcal{P}$:

$$\mathfrak{U} : \mathcal{O} \longrightarrow \mathcal{P}. \tag{5.1}$$

We denote the range of $\mathfrak{U}$ to be the set $\bar{\mathcal{P}} \subseteq \mathcal{P}$. The above examples show that the function $\mathfrak{U}$ can be many-to-one (for example, there are three column-convex polygons that can be constructed from two columns of two cells — see figure 5.1), and that $\bar{\mathcal{P}}$ can be a proper subset of $\mathcal{P}$ (not all sequences of steps on the square lattice will be self-avoiding, for example).

## 5.1.1 Counting with sequences

Let us consider two sets of blocks. Let $\mathcal{S}$ be the set of *seed blocks*, and let $\mathcal{B}$ be the set of *building blocks*. The sequences of blocks that we wish to study are those that start with a seed block and then continue with some number (possibly zero) of building blocks; we denote this set $\mathcal{P}$, and it is defined as

$$\mathcal{P} = \biguplus_{k \geq 0} \mathcal{S} \times \mathcal{B}^k, \tag{5.2}$$

[1]Even animals with no convexity restrictions can be enumerated with this method — see chapter 4.

and so any $\rho \in \mathcal{P}$ can be written uniquely as

$$\rho = (\sigma, \beta_1, \beta_2, \ldots, \beta_k), \tag{5.3}$$

with $\sigma \in \mathcal{S}$ and $\beta_i \in \mathcal{B}$. Let $\mathcal{O}$ be some set of lattice objects that decompose into the sequences given in the set $\mathcal{P}$ under an "ungluing" function, $\mathfrak{U}$.

In order to enumerate the elements of $\mathcal{O}$ according to some weighting, $\Omega$, we form the generating function

$$W = \sum_{A \in \mathcal{O}} \Omega(A). \tag{5.4}$$

By using the ungluing function, $\mathfrak{U}$, we can rewrite this generating function in terms of the sequences in $\mathcal{P}$ enumerated according to a new weight function $\omega$,

$$W = \sum_{A \in \mathcal{O}} \Omega(A) = \sum_{\rho \in \mathcal{P}} \omega(\rho). \tag{5.5}$$

The weight function $\omega$ is defined to be

$$\omega(\rho) = \sum_{A \in \mathfrak{U}^{-1}(\rho)} \Omega(A), \tag{5.6}$$

where $\mathfrak{U}^{-1}(\rho) = \{A \in \mathcal{O} \mid \mathfrak{U}(A) = \rho\}$.

For example, let $\mathcal{O}$ be the set of column-convex polygons enumerated by area ($q$) and horizontal and vertical perimeters ($x$ and $y$ respectively), and let $\mathcal{P}$ be the set of sequences of connected columns of cells. Let $\rho$ be the sequence of two columns of two cells. There are three column-convex polygons that can be decomposed to give $\rho$ (see figure 5.1). All have an area of 4 and a horizontal perimeter of 4, but two have a vertical perimeter of 6, while the last has a vertical perimeter of 4. Consequently $\omega(\rho) = x^4 q^4 (y^4 + 2y^6)$.

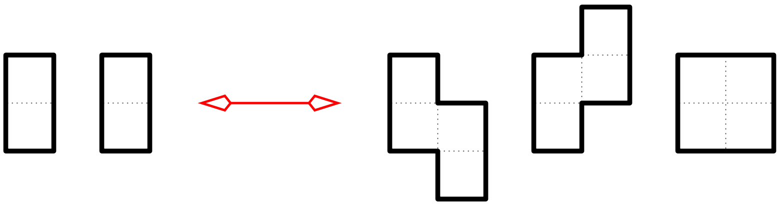


Figure 5.1: Three column-convex polygons can be constructed from two columns of two cells.

Since any given sequence $\rho$ is constructed block-by-block, it is natural to ask how the weight of the sequence develops as the blocks are added; *i.e.* if we add the block $\alpha$ to the end of $\rho$, how does $\omega$ change? Let $\rho = (\sigma, \beta_1, \ldots, \beta_k) \in \mathcal{P}$, and $\alpha \in \mathcal{B}$. We define the *transfer weight* $\tau$ to be

$$\tau = \omega(\rho \circ \alpha)/\omega(\rho), \tag{5.7}$$

where $\rho \circ \alpha$ denotes the sequence obtained by adding $\alpha$ to the end of $\rho$. The transfer weight, $\tau$, is a function of the blocks that make up the sequence. If $\tau$ is a function of the last $i$ blocks in the sequence,

$$\omega(\rho, \alpha)/\omega(\rho) = \tau(\beta_{k+2-i}, \beta_{k+3-i}, \ldots, \beta_k, \alpha), \tag{5.8}$$

then we say that $\omega$ has *depth* $i$. Let $W(\alpha_1, \ldots, \alpha_{i-1})$ be the generating function of all sequences ending in the blocks $\alpha_1, \ldots, \alpha_{i-1}$, enumerated by $\omega$. The definition of $\tau$ gives us the recurrence

$$\begin{aligned} W(\alpha_1, \ldots, \alpha_{i-1}) &= \omega(\alpha_1, \ldots, \alpha_{i-1}) + \sum_{\ell \geq 1} \sum_{\gamma_1, \ldots, \gamma_\ell} \omega(\gamma_1, \ldots, \gamma_\ell, \alpha_1, \ldots, \alpha_{i-1}) \\ &= \omega(\alpha_1, \ldots, \alpha_{i-1}) + \sum_{\ell \geq 1} \sum_{\gamma_1, \ldots, \gamma_\ell} \omega(\gamma_1, \ldots, \gamma_\ell, \alpha_1, \ldots, \alpha_{i-2}) \tau(\gamma_\ell, \alpha_1, \ldots, \alpha_{i-1}) \\ &= \omega(\alpha_1, \ldots, \alpha_{i-1}) + \sum_{\gamma_\ell} W(\gamma_\ell, \alpha_1, \ldots, \alpha_{i-2}) \tau(\gamma_\ell, \alpha_1, \ldots, \alpha_{i-1}) \end{aligned} \tag{5.9}$$

For many interesting problems $\omega$ does not have a finite depth. One important example of this sort of problem is the enumeration of self-avoiding walks:

**Example 5.1 (Self-avoiding walks enumerated by length).**
Let $\mathcal{O}$ be the set of self-avoiding walks (SAWs) on the square lattice, and let $\mathcal{S}$ and $\mathcal{B}$ be the set of four possible steps on the square lattice. The weight function $\Omega$ acting on a SAW $A$ is simply $x^{\text{length}(A)}$.

Since $\mathcal{P}$ contains all sequences of steps, including those that are not self-avoiding, we define $\omega$ to be zero on those sequences that are not self-avoiding. *i.e.* $\omega$ is defined by

$$\omega(\rho) = \begin{cases} x^{\text{length}(\rho)} & \text{if } \rho \text{ is self-avoiding} \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

This function does not obviously simplify, since in order to test whether or not a given sequence is self-avoiding we must consider every step in the sequence at once — it is highly non-local. If we add a new block to the end of given sequence, then the self-avoidance of the new sequence depends upon all the blocks, and so $\omega$ has arbitrary depth (*i.e.* the depth can be equal to the length of the walk).

It is worth noting that one can rephrase the self-avoiding walk problem, so that the weight function has depth 1, but the building blocks and the transfer weights are far from trivial [43].

Complicated, infinite depth, weight functions, like that of self-avoiding walks, arise all too frequently in problems that we would like to be able to solve. Unfortunately very little can be done with them. At the other end of the spectrum are weight functions which have very small depth; consider a depth 1 weight function:

$$\omega(\rho) = \omega(\sigma, \beta_1, \beta_2, \ldots, \beta_k) = \omega(\sigma)\tau(\beta_1) \cdots \tau(\beta_k). \tag{5.11}$$

In this case, adding a new block to a sequence, simply multiplies the weight of the sequence by the weight of the new block, independent of all other blocks in the sequence. In such cases the generating function is very easily computed:

**Lemma 5.1.** *Let $\omega(\rho)$ be the weight function defined by equation (5.11). The generating function, $W = \sum_{\rho\in\mathcal{P}} \omega(\rho)$ satisfies*

$$W = S + WB, \tag{5.12}$$

*where $S$ and $B$ are the generating function of $\mathcal{S}$ and $\mathcal{B}$ enumerated according to $\tau$. Solving the equations gives*

$$W = \frac{S}{1-B}. \tag{5.13}$$

*Proof.* Let $\mathcal{P}_n = \mathcal{S} \times \mathcal{B}^{n-1}$, and let $W_n$ be the associated generating function. Since $\mathcal{P}_1 = \mathcal{S}$ and $\mathcal{P}_n = \mathcal{P}_{n-1} \times \mathcal{B}$, the generating functions, $W_n$, satisfy the following recurrence

$$\begin{aligned} W_1 &= S, & (5.14)\\ W_{n+1} &= \sum_{\rho\in\mathcal{P}_n} \sum_{\beta_{n+1}\in\mathcal{B}} \omega(\rho \circ \beta_{n+1}) \\ &= \sum_{\rho\in\mathcal{P}_n} \omega(\rho) \sum_{\beta_{n+1}\in\mathcal{B}} \tau(\beta_{n+1}) \\ &= W_n B. & (5.15) \end{aligned}$$

Summing over $n$ gives the equation satisfied by $W$. ∎

**Example 5.2 (Bargraphs enumerated by area).**
Let $\mathcal{O}$ be the set of bargraphs. Since every bargraph is constructed from a sequence of columns, the seed blocks, $\mathcal{S}$, and building blocks, $\mathcal{B}$, are both equal to the set of columns of cells, $\Gamma$. More specifically $\Gamma = \{\gamma_i\}_{i\geq 1}$, where $\gamma_i$ is the connected column of $i$ cells.

The weighting $\Omega$ acting on a bargraph, $A$, is defined to be $q^{\mathrm{area}(A)}$. Since the area of a bargraph is the sum of the areas of its columns, we define weight function $\omega$ acting on a sequence of columns $\rho = (\gamma_{a_1}, \gamma_{a_2}, \ldots, \gamma_{a_k})$ to be

$$\begin{aligned} \omega(\rho) &= \omega(\gamma_{a_1}, \gamma_{a_2}, \ldots, \gamma_{a_k}) \\ &= q^{\sum_{i=1}^k a_i} = \prod_{i=1}^k q^{a_i} \\ &= \prod_{i=1}^k \tau(\gamma_{a_i}), & (5.16) \end{aligned}$$

*i.e.* the weight of the sequence is simply the product of the weights of each column, and so we can apply lemma 5.1. The generating function of $\Gamma$ with respect to $\tau$ is

$$C(q) = \sum_{n\geq 1} \tau(\gamma_n) = \sum_{n\geq 1} q^n = \frac{q}{1-q}, \tag{5.17}$$

and so by lemma 5.1:

$$W(q) = \frac{C(q)}{1 - C(q)} = \frac{\frac{q}{1-q}}{1 - \frac{q}{1-q}} = \frac{q}{1-2q}. \tag{5.18}$$

Broadly speaking, weight functions of the sort addressed by lemma 5.1 are too simple to be applicable to any interesting problems[2]. As noted above, many problems that we would like to solve, such as self-avoiding walks, have infinite depth weight functions that we can do barely anything with.

Luckily one can find interesting problems, that have finite depth weight functions. In fact, there are a wide range of interesting (and frequently *solvable*) problems that have depth 2 weight functions:

$$\omega(\rho) = \omega(\sigma)\tau(\sigma, \beta_1)\tau(\beta_1, \beta_2)\dots\tau(\beta_{k-1}, \beta_k). \tag{5.19}$$

Such a function defines the weight of the sequence as the product of weights on adjacent blocks. The remainder of the chapter focuses on the solution of models with depth 2 weight functions.

### 5.1.2 From depth 2 weight functions to solutions

As noted above, there are many interesting problems that have weights that simplify into the form of equation (5.19). Even in the case of very general weights, $\tau(\alpha, \beta)$, this recurrence can be solved by translating it into a matrix equation — this is essentially the *transfer matrix method*, which is a classical and powerful combinatorial technique (chapter 4, section 7 of [147]); to quote Stanley (*ibid.*) :

"*The transfer matrix method, like the Principle of Inclusion-Exclusion and the Möbius inversion formula, has simple theoretical underpinnings but a very wide range of applicability.*"

In some sense the ideas we have described above are a generalisation of the transfer matrix method.

Take an element $\rho = (\sigma, \beta_1, \dots \beta_k) \in \mathcal{P}$, and define a weight function $\omega$ such that

$$\omega(\rho) = \omega(\sigma)\tau(\sigma, \beta_1)\tau(\beta_1, \beta_2)\dots\tau(\beta_{k-1}, \beta_k). \tag{5.20}$$

For any $\alpha \in \mathcal{S} \cup \mathcal{B}$ define the set $\mathcal{P}_\alpha$ as those sequences in $\mathcal{P}$ that end in the block $\alpha$. The generating function of this set is

$$W_\alpha = \sum_{\rho \in \mathcal{P}_\alpha} \omega(\rho). \tag{5.21}$$

Since any element of $\mathcal{P}_\alpha$ is either the single block $\alpha$, or the concatenation of any sequence and the single block $\alpha$ we have the following equation:

$$W_\alpha = \omega(\alpha) + \sum_\beta W_\beta \tau(\beta, \alpha). \tag{5.22}$$

[2]This is particularly unfortunate, since they are very easy to solve!!

If we can solve this equation, then the generating function $W$ can be recovered by summing over all $\alpha$.

Let us index the set $\mathcal{S} \cup \mathcal{B} = \{\alpha_i \mid i \in \mathcal{I}\}$, then we can rewrite equation (5.22) as the following matrix equation

$$\vec{W} = \vec{S} + \vec{W} M, \tag{5.23}$$

where $\vec{W}_i = W_{\alpha_i}$, $\vec{S}_i = \omega(\alpha_i)$ and $M_{i,j} = \tau(\alpha_i, \alpha_j)$. Solving this equation for $\vec{W}$ gives

$$\vec{W} = \vec{S}\,(I - M)^{-1}. \tag{5.24}$$

The problem of computing $W$, has been transformed into one of finding $(I - M)^{-1}$. If the index set, $\mathcal{I}$, is finite, then one can invert the matrix by hand or computer — many numerical enumeration results for lattice animals have been obtained this way (see chapter 4, section 7 of [147] for example).

For many polyomino problems, including that of column-convex polygons (which we shall use as an example throughout this chapter), we can find and solve equation (5.22) without matrix inversion — this approach is sometimes called the "Temperley method".

### 5.1.3 Building column-by-column

It is very difficult to characterise all the models that are amenable to the Temperley method[3]; like a great number of methods in combinatorics, we should consider it to be a *technique* that *may* be useful in solving a model, rather than a theorem with a clear domain of applicability. When we approach a new polyomino problem, we need to determine whether or not the Temperley method[4] is actually applicable; if so — great!, if not then we have to look for some other method.

The application of the technique can be split into two separate problems. The first is to find a recurrence satisfied by the number of polygons (or a functional equation satisfied by the associated generating function), while the second problem is the resolution of this recurrence. Indeed this is true of other techniques we might use to find a solution (such as wasp-waist factorisations or heap techniques), but the character of the "Temperley method" is defined by the way in which we find the recurrence; by constructing polygons in an incremental way, *column-by-column*, we naturally obtain a recurrence like that of equation (5.22).

In the next two chapters we describe four variants of this *column-by-column* approach, and we should start by stating that

***All of these methods are essentially the same!***

By various manipulations one can show that the mathematics behind all four methods is identical — the differences between them lie in the aesthetics of the derivation of the recurrence (or functional equation) and the subsequent resolution of that recurrence (or functional equation).

[3] Though it is safe to say that all "natural" families of column-convex polygons can be enumerated with it — see [23] for example.

[4] We will get around to defining it in a moment.

In section 5.2 we first review Temperley's original method [151]. This method allows one to find, with relative ease, a recurrence on the number of polygons. Though this recurrence is easily solved in some cases (such as column-convex polygons enumerated according to their area), in general it is not easily solved. When a solution *is* forthcoming it can be in a form that is not very aesthetically pleasing.

By summing these recurrences, they can be translated into functional equations. These functional equations can be interpreted combinatorially, and most importantly these equations can be solved far more easily and the solution is in a better form (we describe the resolution of the functional equations in section 5.5). We continue section 5.2 by describing Bousquet-Mélou's "functional Temperley" method [23] which derives these functional equations by direct combinatorial construction (rather than deriving and then summing a recurrence).

In section 5.3 we describe a less well known version of the Temperley method — the "integral Temperley" method, introduced by Klarner [105, 106]. This variant translates the recurrence into constant-term integral equations satisfied by the generating function of the polygons. Solving such equations is, in general, highly non-trivial, however we show how the constant-term integrals can be replaced with Hadamard products, which in turn can be reduced to the same functional equations found by the functional Temperley method.

We also give a different combinatorial interpretation of the constant-term integrals (and Hadamard products) that changes the problem of deriving the functional equations into one of finding the generating functions of small sets of polygons (such as column-convex polygons of width 2) — we call this variation the Hadamard-Temperley method. And while it is arguable if this simplifies the derivation of functional equations of families of column-convex polygons enumerated by area and perimeter, in certain situations it does make things easier (for example, the equation satisfied by bargraph polygons enumerated by site-perimeter in chapter 6 or that satisfied by *2-4-2* polygons in chapter 7).

In section 5.5 we show how the functional equations found by the functional-Temperley and Hadamard-Temperley methods can be solved (following Bousquet-Mélou's paper [23]). In section 5.6 we solve the non-trivial example of staircase polygons enumerated by area, perimeter and site-perimeter.

## 5.2

## The Temperley method

In this section we describe the two best known versions of the Temperley method, Temperley's original method and a variant of it due to Bousquet-Mélou, which we will call the "functional Temperley method".

### 5.2.1 Original Temperley

Every column-convex polygon can be mapped to a sequence of columns of cells, and so we define $\mathcal{S} = \mathcal{B} = \Gamma = \{\gamma_i\}$, the set of all connected columns of cells. We define the *left height* and *right height* of a column-convex polygon to be the number of cells in its left-most and right-most columns respectively.

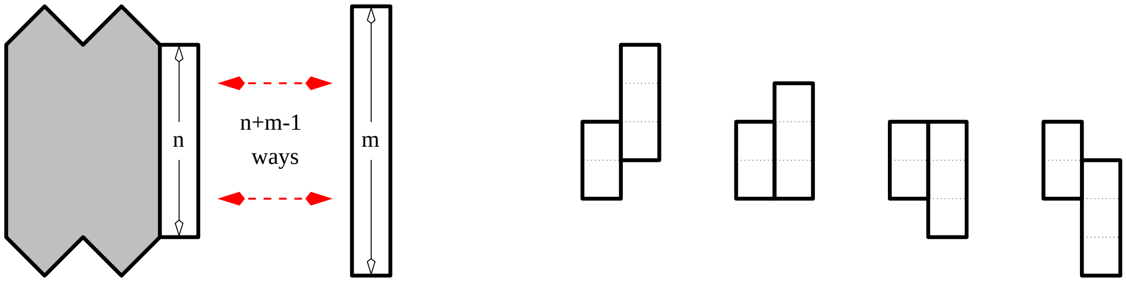



Figure 5.2: Gluing a column of height $m$ onto a polygon with right height $n$ can be done in $n+m-1$ ways. The 4 ways that a column of height 3 can be glued to a column of height 2.

Let $A$ be a column-convex polygon of right height $n$. Gluing a column of $m$ cells to the rightmost column of $A$ increases the area by $m$, and further the column can be "glued" in one of $n+m-1$ different ways (see figure 5.2). Consequently we define the transfer weight to be $\tau(\gamma_n, \gamma_m) = (n+m-1)q^m$.

If we define $G_n(q)$ to be the generating function of all column-convex polygons of right-height $n$, then in analogy with equation 5.22 we find

$$G_n(q) = q^n \left( 1 + \sum_{m\geq 1} (m+n-1) G_m(q) \right). \tag{5.25}$$

Taking differences we find that

$$\begin{aligned} G_n(q) - qG_{n-1}(q) &= q^n \sum_{m\geq 1} G_m(q) \qquad \text{and so,} & (5.26) \\ G_n(q) - 2qG_{n-1}(q) + q^2 G_{n-2}(q) &= 0, & (5.27) \end{aligned}$$

which is valid for $n \geq 3$. This is a constant coefficient recurrence with repeated characteristic roots, and so has a general solution of the form

$$G_n = A(q)q^n + B(q)nq^n, \tag{5.28}$$

where $A(q)$ and $B(q)$ are arbitrary functions of $q$ that are independent of $n$. Clearly

$$\begin{aligned} \sum_{n\geq 1} G_n(q) &= \frac{qA(q)}{1-q} + \frac{qB(q)}{(1-q)^2} & (5.29) \\ \sum_{n\geq 1} nG_n(q) &= \frac{qA(q)}{(1-q)^2} + \frac{q(1+q)B(q)}{(1-q)^3} & (5.30) \end{aligned}$$

Substituting $n = 1, 2$ into equation (5.25) we find that

$$G_1(q) \quad = qA(q) + qB(q) \quad = q\left(1 + \sum_{n\geq 1} nG_n(q)\right) \tag{5.31}$$

$$G_2(q) \quad = q^2A(q) + 2q^2B(q) \quad = q^2\left(1 + \sum_{n\geq 1} G_n + \sum_{n\geq 1} nG_n(q)\right) \tag{5.32}$$

Substituting equations (5.29) and (5.30) into equations (5.31) and (5.32) and solving gives $A(q)$ and $B(q)$. Putting everything back together gives the area generating function

$$G(q) = \sum_{n\geq 1} G_n(q) = \frac{q(1-q)^3}{1-5q+7q^2-4q^3}. \tag{5.33}$$

This is Temperley's solution of column-convex polygons, enumerated by area.

**Remark.**
It is interesting to note (see [147]) that the generating function (or the equivalent recurrence for the coefficients) of column-convex polygons enumerated by area was known to Pólya in 1938 (but unpublished until 1969). These polygons remained the largest solved class of polyominoes (since every polygon is a polyomino) on the square lattice for about 60 years (until very recent work — see chapter 4).

This same method can be pushed further to yield the area and perimeter generating function. Temperley computed the transfer weights (enumerating both perimeter and area) and from it found the recurrence

$$\begin{aligned} G_{n+2} - 2q(1+y^2)G_{n+1} &+ q^2(1+4y^2+y^4)G_n \\ -2q^3(y^2+y^4)G_{n-1} &+ q^4y^4G_{n-2} = x^2q^{n+2}(1-y^2)^2G_n, \end{aligned} \tag{5.34}$$

where $G_n$ now enumerates column-convex polygons of right height $n$ by their horizontal and vertical perimeters ($x$ and $y$ resp.) and their area ($q$).

Setting $q = 1$ reduces this to a constant coefficient recurrence, but due to the complexity of the roots of the characteristic equation Temperley was only able to find an *implicit* solution. Temperley's method did not resurface until Brak, Guttmann and Enting [35] managed to find the algebraic generating function in closed form with the aid of a computer algebra package, *Mathematica*$^{TM}$. Delest [50] was able to find the explicit solution a few years previous to this, via algebraic language techniques, but the method is less direct and substantially more complicated. Feretić was able to greatly simplify the anisotropic perimeter generating function [65].

When $q \neq 1$ this recurrence is no longer a constant coefficient recurrence and more powerful methods must be used than those above. Brak and Guttmann [34] have solved the recurrence for general $q$, but the techniques they used are far from trivial, and the form of the solution is not elegant.

### 5.2.2 Functional Temperley

The mathematics at the heart of Temperley's method and the functional Temperley method described by Bousquet-Mélou [23] are essentially identical. As we have seen above, Temperley's methodology computes the weight function explicitly and translates the construction into recurrences satisfied by the coefficients, $G_n(q)$. If we sum this recurrence then we obtain a functional equation. Consider again the recurrence satisfied by column-convex polygons:

$$G_n(q) = q^n \left( 1 + \sum_{m \geq 1} (m+n-1) G_m(q) \right).$$

Multiply both sides of this equation by $s^n$ and sum over $n$:

$$\begin{aligned} P(s;q) = \sum_{n\geq 1} G_n(q) s^n &= \sum_{n\geq 1} s^n q^n \left( 1 + \sum_{m\geq 1} (m+n-1) G_m(q) \right) \\ &= \frac{sq}{1-sq} + \sum_{n,m\geq 1} (n-1) s^n q^n G_m(q) + \sum_{n,m \geq 1} s^n q^n m G_m(q) \\ &= \frac{sq}{1-sq} + \frac{s^2q^2}{(1-sq)^2} P(1;q) + \frac{sq}{1-sq} \left( \frac{\partial P}{\partial s} \right)(1,q), \end{aligned} \tag{5.35}$$

where $P(s;q) = \sum_{n\geq 1} G_n(q) s^n$, is the generating function of column-convex polygons enumerated by their area ($q$) and right height ($s$).

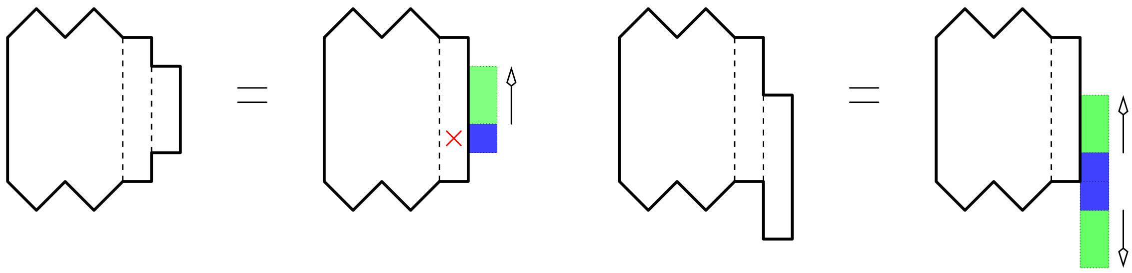
The "functional Temperley method" is a way of deriving this same equation more directly. As per Temperley's original method, we consider how a new column can be added to an existing polygon. Rather than considering the number of ways we can "glue" a column of height $n$ onto the end of a polygon with right height $m$, let us consider how a new column can be "grown" on the end of the existing polygon.

Consider figure 5.3. The way in which a column-convex polygon with $n+1$ columns, $B$, is constructed from a column-convex polygon with $n$ columns, $A$, depends upon the relative positions of the lowest cells in the $n$th and $(n+1)$th columns of $B$:

case 1 if the lowest cell in the $n$th column is lower than the lowest cell in the $(n+1)$th column, then we construct $B$ from $A$ by choosing one of the cells in the $n$th column of $A$ and adding a single cell to its right. Then grow some number of cells (possibly zero) above this.

case 2 if the lowest cell in the $n$th column is strictly higher than the lowest cell of the $(n+1)$th column, then we construct $B$ from $A$ by adding a cell to the right of the lowest cell in the $n$th column and then another one directly below this one. Then grow some number of cells (possibly zero) both above and below this.

These two constructions can be *directly* translated into operations on the generating function $P(s;q)$ as defined above.



(a) case 1 (b) case 2

Figure 5.3: Building column-convex polygons. In case 1 we choose a single cell in the rightmost column, place a single cell on its right (the blue cell) and then grow some number of cells above it (the green cells). In case 2 we place two cells at the lower end of the rightmost column (the blue cells), and then grow some number of cells above and below these (the green cells).

case 1 Choosing a single cell in the rightmost column of $A$, is the same as marking a single cell in the rightmost column; the generating function for polygons marked in this way is $\sum nG_n(q)s^n = s\frac{\partial P}{\partial s}(s,q)$. Adding a cell to the right of the mark, gives a new polygon with right height 1, and generating function $sq\left(\frac{\partial P}{\partial s}\right)(1,q)$. Growing some number of cells (possibly zero) above this corresponds to multiplying the generating function by $\frac{1}{1-sq}$.

case 2 Adding two cells to the right of the $n$th column of $A$ gives a new polyomino with right height 2, and the corresponding generating function is $s^2q^2P(1;q)$. Growing some number of cells (possibly zero) above and below this pair of cells, corresponds to multiplying the generating function by $\frac{1}{(1-sq)^2}$.

Since a column-convex polygon is either a single column (whose generating function is $\frac{sq}{1-sq}$) or can be constructed from a column-convex polygon by the column-gluing construction above we arrive back at the functional equation

$$P(s;q) = \frac{sq}{1-sq} + \frac{s^2q^2}{(1-sq)^2}P(1;q) + \frac{sq}{1-sq}\left(\frac{\partial P}{\partial s}\right)(1,q). \tag{5.36}$$

Substituting $s=1$ into the equation and its derivative with respect to $s$ gives

$$P(1;q) = \frac{q}{1-q} + \frac{q^2}{(1-q)^2}P(1;q) + \frac{q}{1-q}\left(\frac{\partial P}{\partial s}\right)(1,q) \tag{5.37}$$

$$\left(\frac{\partial P}{\partial s}\right)(1,q) = \frac{q}{(1-q)^2} + \frac{2q^2}{(1-q)^3}P(1;q) + \frac{q}{(1-q)^2}\left(\frac{\partial P}{\partial s}\right)(1,q). \tag{5.38}$$

Solving this linear system of equations we get Temperley's solution,

$$P(1;q) = \frac{q(1-q)^3}{1-5q+7q^2-4q^3}. \tag{5.39}$$

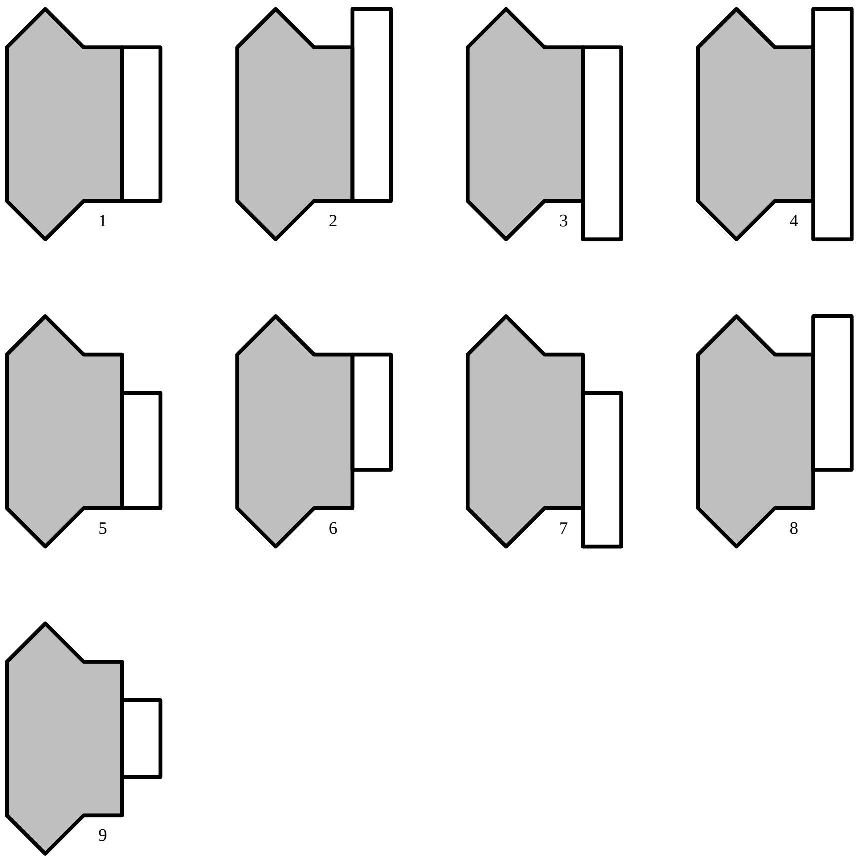



Figure 5.4: The nine cases we need to consider to build a new column.

Let us (take a deep breath and) include the horizontal and vertical perimeter and repeat this method. We will use $P(s)$ as shorthand for $P(s;x,y,q)$. When adding a new column, $b$, to the right of the last column (call it $a$) of a column-convex polygon, the position of the lower (resp. upper) edge of $b$ can be higher, equal or lower relative to the lower (resp. upper) edge of $a$. Consequently there are 9 cases[5] (which with symmetries and simplifications boil down to only 6) we consider — these are illustrated in figure 5.4.

1 The new column is identical to the previous. The horizontal half-perimeter increases by 1, the vertical perimeter does not change, and the area increases by the height of

[5]If we consider the edges of $a$ to be either higher (and possibly equal) or *strictly* lower relative to the edges of $b$, then the 9 cases reduce to 4 — this is why there are only 4 cases to consider in [23].

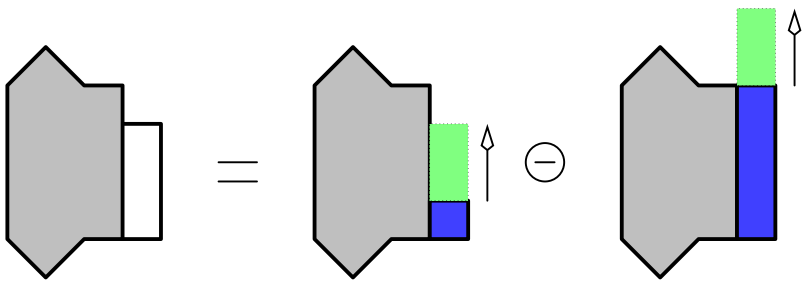


Figure 5.5: Cases 5 & 6. The green cells indicate some number of cells (possibly zero) that are grown.

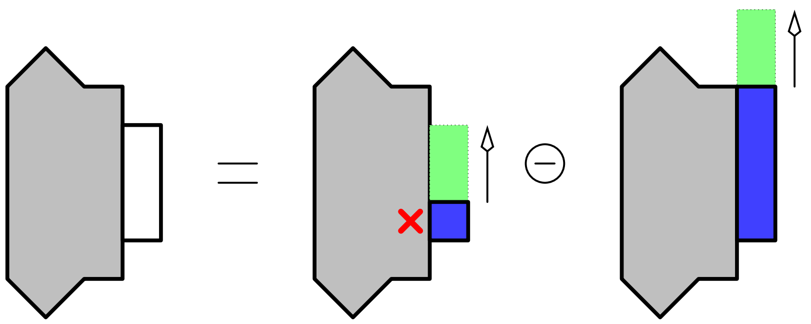


Figure 5.6: Case 9. Again the green cells indicate some number of cells (possibly zero) that are grown.

the column.

$$P(s) \to xP(sq).$$

2,3 As per (1), except now a column of cells is added. The generating function of these cells is $\frac{syq}{1-syq}$, since they increase both the area and the vertical half-perimeter.

$$P(s) \to x\frac{syq}{1-syq}P(sq).$$

4 As per (2,3) except that a column of cells is added at both ends.

$$P(s) \to x\left(\frac{syq}{1-syq}\right)^2 P(sq).$$

5,6 See figure 5.5. To grow this column, add a single cell to the right of the bottom (resp. top) of the existing column — the resulting generating function is $xsqP(1)$. Then a column of cells is grown upwards (resp. downwards). Giving the generating function $x\frac{sq}{1-sq}P(1)$.

In so doing, polygons are created whose top (resp. bottom) edge is too high (resp. low). These are exactly those enumerated in case (1) with a (possibly zero-length) column of cells attached (not contributing to the vertical perimeter).

$$P(s) \to x\frac{sq}{1-sq}P(1) - \frac{1}{1-sq}(\text{case } 1).$$

7,8 As per (5,6) except that now a column of cells is added to the end, each of which increases the area and the vertical half-perimeter. The generating function of these cells is $\frac{syq}{1-syq}$, and hence

$$P(s) \to \frac{syq}{1-syq}(\text{case 5 or 6}).$$

9 See figure 5.6. To grow this block, one must mark (or choose) a cell in the existing column (but not the bottom cell) — the corresponding generating function is $\sum(n-1)G_n(q) = \frac{\partial P}{\partial s}(1) - P(1)$. Place a new cell next to this and then grow a column of cells upwards. The corresponding generating function is $x\frac{sq}{1-sq}\left(\frac{\partial P}{\partial s}(1) - P(1)\right)$.

In so doing, polygons are created whose top edge is too high — these are equivalent to those enumerated in (6) with a (possibly zero-length) column of cells (not contributing to the vertical perimeter).

$$P(s) \to x\frac{sq}{1-sq}\left(\frac{\partial P}{\partial s}(1) - P(1)\right) - \frac{1}{1-sq}(\text{case } 6)$$

In summary

- cases 1,2,3,4 $\Rightarrow \frac{x}{(1-syq)^2}P(sq)$,
- cases 5,6,7,8 $\Rightarrow 2\frac{x}{(1-syq)(1-sq)}\Big(sqP(1) - P(sq)\Big)$, and
- case 9 $\Rightarrow x\frac{sq}{1-sq}\frac{\partial P}{\partial s}(1) + \frac{sq(sq-2)}{(1-sq)^2}P(1) + \frac{1}{(1-sq)^2}P(sq)$.

Adding these contributions together gives the linear functional equation satisfied by $P(s)$:

$$\begin{aligned} P(s) &= \frac{xsyq}{1-syq} + \frac{xsq}{1-sq}\frac{\partial P}{\partial s}(1) + \frac{xs^2q^2(2y-syq-1)}{(1-sq)^2(1-syq)}P(1) \\ &\quad + \frac{xs^2q^2(1-y)^2}{(1-sq)^2(1-syq)^2}P(sq). \end{aligned} \tag{5.40}$$

This method moves the difficulty of manipulating recurrences (as one finds using Temperley's original method) to a combinatorial derivation of a functional equation. The resulting functional equation can then be solved by a (relatively easy) "iteration" technique, which may be applied to a wide range of such equations (we describe iteration and other methods for the solution of functional equations in section 5.5). The solution is given by

$$P(1) = \frac{\mathfrak{A}(1) + \mathfrak{D}(1)\mathfrak{A}'(1) - \mathfrak{A}(1)\mathfrak{D}'(1)}{1 - \mathfrak{B}(1) - \mathfrak{D}'(1) - \mathfrak{D}(1)\mathfrak{B}'(1) + \mathfrak{B}(1)\mathfrak{D}'(1)}, \tag{5.41}$$

where $\mathfrak{A}'(1)$ denotes $\left(\frac{\partial \mathfrak{A}}{\partial s}\right)|_{s=1}$, and

$$\mathfrak{A}(s) = y \sum_{n \geq 1} \frac{x^n s^{2n-1} (1-y)^{2n-2} q^{n^2}}{(sq)_{n-1}^2 (syq)_{n-1} (syq)_n}, \tag{5.42}$$

$$\mathfrak{B}(s) = \sum_{n \geq 1} \frac{x^n s^{2n} (1-y)^{2n-2} q^{n(n+1)} (2y - syq^n - 1)}{(sq)_n^2 (syq)_{n-1} (syq)_n}, \tag{5.43}$$

$$\mathfrak{D}(s) = \sum_{n \geq 1} \frac{x^n s^{2n-1} (1-y)^{2n-2} q^{n^2}}{(sq)_{n-1} (sq)_n (syq)_{n-1}^2}. \tag{5.44}$$

This formula can be simplified further — see theorem 4.8 in [23]. The form of this solution is quite substantially simpler than that found by Brak and Guttmann, and its structure is more amenable to analysis.

## 5.3 Integral Temperley and Hadamard products

Another variant of the Temperley method, though perhaps less well known, was introduced by Klarner [105, 106], in which the column-by-column construction is translated into a constant-term integral equation. We refer to this approach as the "integral Temperley method".

Like Temperley, Klarner computes the weight function for column-convex polygons explicitly and then translates the recurrence of equation 5.22 into an integral equation (which he was able to solve for the area generating function). We will show how these integrals can be replaced by Hadamard products (making the equations a little more rigorous); these Hadamard products equations can be reduced to the same functional equations as one finds with the functional Temperley method.

This would seem to imply that Klarner's integral Temperley method is a round-about way of deriving functional equations, and at the same time appears to be more difficult (thanks to the integrals) than the functional Temperley method. However, we show that Klarner's approach has a geometric interpretation that is quite different to that of the other two approaches. This interpretation changes the problem of the derivation of the functional equations into a problem of computing the generating functions of small sets of polygons (such as column-convex polygons with only two columns). Late in this thesis we make extensive use of this technique to study the site-perimeter of bargraphs (in chapter 6) and to prove certain results concerning the analytic structure of the self-avoiding polygon generating function (in chapter 7).

### 5.3.1 Integral Temperley

Let $p(k,n,a)$ be the number of column-convex polygons with $k$ columns, area $n$ and right-height $a$. Following Temperley's reasoning, the column-by-column construction implies the following recurrence on these numbers

$$\begin{aligned} p(1,n,a) &= \delta(n-a) && (5.45)\\ p(k,n,a) &= \sum_{b\geq 1}(a+b-1)\,p(k-1,n-a,b). && (5.46)\end{aligned}$$

The associated generating functions are

$$\begin{aligned} P_1(s;q) &= \sum_{n\geq 1} s^n q^n = \frac{sq}{1-sq}, && (5.47)\\ P_k(s;q) &= \sum_{a,n\geq 1} p(k,n,a)q^n s^a, && (5.48)\\ P(s;x,q) &= \sum_{k\geq 1} P_k(s;q)x^k. && (5.49)\end{aligned}$$

In order to translate the recurrence in equation (5.46) into a recurrence on the generating functions we consider the generating function of the transfer weights

$$\bar{T}(t,s;q) = \sum_{a,b}\tau(a,b)t^a s^b = \sum_{a,b\geq 1}(a+b-1)q^b t^a s^b = \frac{stq(1-stq)}{(1-t)^2(1-sq)^2}, \quad (5.50)$$

and the product[6]:

$$\begin{aligned} P_k(t;q)\bar{T}(1/t,s;q) &= \left(\sum_{a,n\geq 1} p(k,n,a)q^n t^a\right)\left(\sum_{b,c\geq 1}(b+c-1)t^{-b}s^c q^c\right)\\ &= \sum_{a,b\geq 1} t^{a-b}\left(\sum_{n,c\geq 1} q^{n+c}s^c\,(b+c-1)\,p(k,n,a)\right)\\ &= t^0\left(\sum_{a,n,c\geq 1} q^{n+c}s^c\,(a+c-1)\,p(k,n,a)\right)\\ &\quad +\sum_{a\neq b} t^{a-b}\left(\sum_{n,c\geq 1} q^{n+c}s^c\,(b+c-1)\,p(k,n,a)\right)\\ &= P_{k+1}(s;q)+\sum_{a\neq b} t^{a-b}\left(\sum_{n,c\geq 1} q^{n+c}s^c\,(b+c-1)\,p(k,n,a)\right). \end{aligned}$$

[6] One needs to be a little careful here about what space of series we are working in; there is no product in the space of functions of the form $\sum_{n\in\mathbb{Z}} a_n t^n$. If we consider $\bar{T}(t,s;q)$ to be a series in $q$ with rational coefficients in $s$ and $t$ everything should be fine... Anyway, in a couple of pages we will show nasty products like this can be avoided.

Hence $P_{k+1}(s;q)$ can be expressed as the coefficient of $t^0$ in the product $P_k(t;q)\bar{T}(1/t,s;q)$. The coefficient of $t^0$ in a function $f(t)$ can be calculated by the following integral (provided it has a Taylor expansion around $t=0$):

$$[t^0]f(t) = \frac{1}{2\pi i}\oint f(t)\frac{\mathrm{d}t}{t}. \tag{5.51}$$

Ideally we would then like to write $P_{k+1}(s;q)$ using this integral, however $P_k(t;q)\bar{T}(1/t,s;q)$ does not have a Taylor expansion about $t=0$ (as a power series in $t$ it is of the form $\sum_{n\in\mathbb{Z}} a_n t^n$) and so we should consider the following steps as being purely formal, and let the integral to denote the constant term of the product (which is well defined).

With this in mind, let us rewrite equations (5.47), (5.48) and (5.49) as the following (purely formal) constant-term integral relations:

$$\begin{aligned} P_1(s;q) &= \frac{sq}{1-sq} && (5.52)\\ P_{k+1}(s;q) &= \frac{1}{2\pi i}\oint P_k(t;q)\bar{T}(1/t,s;q)\frac{\mathrm{d}t}{t} && (5.53)\\ P(s;x,q) &= \frac{xsq}{1-sq} + \frac{x}{2\pi i}\oint P(t;x,q)\bar{T}(1/t,s;q)\frac{\mathrm{d}t}{t}. && (5.54)\end{aligned}$$

For convenience let us move the factor of $x$ from outside the integral into the function $\bar{T}(t,s;q)$ to form the function $T(t,s;x,q)$:

$$\begin{aligned} T(t,s;x,q) &= \frac{xstq(1-stq)}{(1-t)^2(1-sq)^2}, \qquad \text{and so} && (5.55)\\ P(s;x,q) &= \frac{xsq}{1-sq} + \frac{1}{2\pi i}\oint P(t;x,q)T(1/t,s;x,q)\frac{\mathrm{d}t}{t}. && (5.56)\end{aligned}$$

In general such integral relations are very difficult to solve. However in this case we can show (a little non-rigorously) that it is equivalent to the linear partial differential equation found by the functional Temperley method (equation (5.36)). Consider the partial fraction form of $T(1/t,s;x,q)/t$ (in the variable $t$):

$$\frac{1}{t}T(1/t,s;x,q) = x\left(\frac{-s^2q^2}{t(1-sq)^2} + \frac{s^2q^2}{(1-sq)^2(t-1)} + \frac{sq}{(1-sq)(t-1)^2}\right). \tag{5.57}$$

Multiplying this by $P(t;x,q)$ gives

$$\begin{aligned} \frac{1}{2\pi i}\oint P(t;x,q)T(1/t,s;x,q)\frac{\mathrm{d}t}{t} &= \frac{-xs^2q^2}{(1-sq)^2}\frac{1}{2\pi i}\oint \frac{P(t;x,q)}{t}\mathrm{d}t\\ &\quad+\frac{xs^2q^2}{(1-sq)^2}\frac{1}{2\pi i}\oint \frac{P(t;x,q)}{t-1}\mathrm{d}t\\ &\quad+\frac{xsq}{1-sq}\frac{1}{2\pi i}\oint \frac{P(t;x,q)}{(t-1)^2}\mathrm{d}t. && (5.58)\end{aligned}$$

At first sight this does not look much better, however Cauchy's formula tells us that integrals of this form can be evaluated easily (since they now have Taylor expansions about $t = 0$):

$$\frac{1}{2\pi i}\oint \frac{m!F(t)}{(t-a)^{m+1}}\,\mathrm{d}t = \left(\frac{\partial^m F}{\partial t^m}\right)\Big|_{t=a}. \tag{5.59}$$

Applying this to the integrals in equation (5.58) (and noting that $P(0; x, q) = 0$) returns us to the linear partial differential equation, equation (5.36). It should be noted that Klarner does not solve equation (5.56) by the above method, rather he manipulates the expression at the level of the coefficients of the generating function to arrive at equations equivalent to equations (5.37) and (5.38). In a later paper, Klarner and Rivest [108], translate an integral recurrence for staircase polygons enumerated by area into a functional equation; this equation is then solved by the iteration method (see section 5.5 below) — the method is not generalised to other problems.

◁ ◁ ◇ ▷ ▷

In the next subsection we will show that the integrals in Klarner's method can be sidestepped, and in the process of doing so we find a more combinatorial interpretation of the $T(t, s, q)$ generating function. This interpretation simplifies the derivation of the functional equations of the functional Temperley method.

### 5.3.2 Replacing the integrals

The evaluation of an integral of the form $\frac{1}{2\pi i}\oint f(t)g(1/t)\frac{\mathrm{d}t}{t}$ is not a trivial matter — it would be much nicer if we could get rid of these integrals completely. Thankfully we are able to replace such integrals with (slightly modified) Hadamard products. If $f(t) = \sum_n f_n t^n$ and $g(t) = \sum_n g_n t^n$ then we note that (formally) we have

$$\frac{1}{2\pi i}\oint f(t)g(1/t)\frac{\mathrm{d}t}{t} = \sum_n f_n g_n. \tag{5.60}$$

This sum is *almost* a Hadamard product.

**Definition 5.1.**
Let $f(t) = \sum_n f_n t^n$ and $g(t) = \sum_n g_n t^n$. The *Hadamard product* $f(t) \odot_t g(t)$ is defined to be

$$h(t) = f(t) \odot_t g(t) = \sum_n f_n g_n t^n. \tag{5.61}$$

Using this notation we can replace the integral, $\frac{1}{2\pi i}\oint f(t)g(1/t)\frac{\mathrm{d}t}{t}$, with $f(t) \odot_t g(t)$ evaluated at $t = 1$. Rather than writing "$f(t) \odot_t g(t)$ evaluated at $t = 1$" we will simply define a *restricted Hadamard product*:

**Definition 5.2.**
Let $f(t) = \sum_{n\geq 0} f_n t^n$ and $g(t) = \sum_{n\geq 0} g_n t^n$. We define the *restricted Hadamard product* (or r-Hadamard product) $f(t) \odot_t g(t)$ to be

$$f(t) \odot_t g(t) = \sum_{n\geq 0} f_n g_n = h(1). \tag{5.62}$$

Using the r-Hadamard product we can rewrite equation (5.56) in the following simpler and more appealing form:

$$P(s;x,q) = \frac{xsq}{1-sq} + P(t;x,q) \odot_t T(t,s;x,q). \tag{5.63}$$

Of course, we still have to be able to evaluate the Hadamard product.

### 5.3.3 Evaluating Hadamard products.

For two general functions $f(t)$ and $g(t)$ the evaluation of $f(t) \odot_t g(t)$ is an extremely non-trivial exercise. However, for certain functions the Hadamard product can be easily evaluated.

**Lemma 5.2 (Easy r-Hadamard products).** *Let $f(t)$ be a formal power series, and choose $\alpha \in \mathbb{C}$ such that $f$ is convergent in the neighbourhood of $t = \alpha$.*

- *The r-Hadamard product is distributive:*

$$f(t) \odot_t (g(t) + h(t)) = f(t) \odot_t g(t) + f(t) \odot_t h(t) \tag{5.64}$$

- *A product rule for the derivative of r-Hadamard products.*

$$\frac{\partial}{\partial s}\left(f(s,t) \odot_t g(s,t)\right) = \left(\frac{\partial f}{\partial s}\right) \odot_t g(s,t) + f(s,t) \odot_t \left(\frac{\partial g}{\partial s}\right) \tag{5.65}$$

- *The r-Hadamard product of $f(t)$ and a simple pole:*

$$f(t) \odot_t \frac{1}{(1-\alpha t)} = f(\alpha). \tag{5.66}$$

- *The r-Hadamard product of $f(t)$ and a higher order pole:*

$$f(t) \odot_t \frac{t^k k!}{(1-\alpha t)^{k+1}} = \frac{\partial^k f}{\partial t^k}(\alpha). \tag{5.67}$$

*Proof.* Each of the above facts can be proved directly from the definition of the r-Hadamard product.

- Let $f(t)=\sum_{n\geq 0} f_n t^n$, $g(t)=\sum_{n\geq 0} g_n t^n$, and $h(t)=\sum_{n\geq 0} h_n t^n$.

$$\begin{aligned}
f(t)\odot_t (g(t)+h(t)) &= \sum_{n\geq 0} f_n\,(g_n+h_n)\\
&= \sum_{n\geq 0} (f_n g_n + f_n h_n)\\
&= f(t)\odot_t g(t) + f(t)\odot_t h(t)
\end{aligned}$$

- Let $f(s,t)=\sum_{n\geq 0} f_n(s)t^n$ and $g(s,t)=\sum_{n\geq 0} g_n(s)t^n$, then

$$\begin{aligned}
\frac{\partial}{\partial s}\Big(f(s,t)\odot_t g(s,t)\Big) &= \frac{\partial}{\partial s}\left(\sum_{n\geq 0} f_n(s)g_n(s)\right)\\
&= \sum_{n\geq 0}\frac{\partial}{\partial s}\left(f_n(s)g_n(s)\right)\\
&= \sum_{n\geq 0}\left(\frac{\partial f_n}{\partial s}g_n(s)+f_n(s)\frac{\partial g_n}{\partial s}\right)\\
&= \left(\frac{\partial f}{\partial s}\right)\odot_t g(s,t) + f(s,t)\odot_t\left(\frac{\partial g}{\partial s}\right)
\end{aligned}$$

- Since $\frac{1}{1-\alpha t}=\sum_{n\geq 0}\alpha^n t^n$ we have

$$\begin{aligned}
f(t)\odot_t\frac{1}{(1-\alpha t)} &= \left(\sum_{n\geq 0} f_n t^n\right)\odot_t\left(\sum_{n\geq 0}\alpha^n t^n\right)\\
&= \sum_{n\geq 0} f_n\alpha^n\\
&= f(\alpha).
\end{aligned}$$

- One can obtain the fourth result simply by differentiating both sides of the third result $k$ times with respect to $\alpha$.

■

It follows directly from the above lemma that the r-Hadamard product of a function, $f(t)$, and any rational function can be rewritten as a linear functional equation in $f(t)$ and its derivatives.

**Corollary 5.3 (r-Hadamard products and linear functional equations).** *Let $f(t)=\sum_{n\geq 0} f_n t^n$ and let $g(t)$ be a rational function $g(t)=\sum_{n\geq 0} g_n t^n$, with poles $t=1/\alpha_j$. The r-Hadamard product $f(t)\odot_t g(t)$ can be expressed as a linear combination of $f(t)$, its derivatives $\left(\frac{\partial^k f}{\partial t^k}\right)$ evaluated at $\alpha_j$, and its derivatives $\left(\frac{\partial^k f}{\partial t^k}\right)$ evaluated at $0$.*

*Proof.* Partial fractions decomposition splits $g(t)$ into a polynomial in $t$ and a sum of rational functions of the form $\frac{1}{(1-\alpha t)^{n+1}}$. These can then be massaged into the forms given in lemma 5.2, namely $\frac{t^k}{(1-\alpha t)^{k+1}}$:

$$\begin{aligned}\frac{1}{(1-\alpha t)^{n+1}} &= \frac{1}{(1-\alpha t)}\frac{1}{(1-\alpha t)^n}\\ &= \frac{1}{1-\alpha t}\left(1+\frac{\alpha t}{1-\alpha t}\right)^n\\ &= \frac{1}{1-\alpha t}\sum_{k=0}^{n}\binom{n}{k}\left(\frac{\alpha t}{1-\alpha t}\right)^k\\ &= \sum_{k=0}^{n}\binom{n}{k}\frac{\alpha^k t^k}{(1-\alpha t)^{k+1}}\end{aligned}$$

Hence if we take $g(t)$ we can decompose it into partial fractions of the form $\frac{t^k}{(1-\alpha t)^{k+1}}$ and some polynomial in $t$. Applying lemma 5.2 to this partial fraction decomposition completes the proof. ∎

**Remark.**
A power series, $f(x)$, is differentiably-finite (or D-finite) if it satisfies a linear ordinary differential equation with polynomial coefficients (see chapter 7). For example, the generating function of staircase polygons, $f(x) = \frac{1}{2}\left(1-x-\sqrt{1-4x}\right)$ satisfies

$$(1-4x)\left(\frac{\mathrm{d}f}{\mathrm{d}x}\right) + 2f - 2x = 0, \tag{5.68}$$

and hence is D-finite. Using lemma 5.2 we can rewrite the above differential equation in terms of r-Hadamard products:

$$\begin{aligned}2f(x) &= f(t)\odot_t \frac{2}{1-tx} &(5.69)\\ (1-4x)\left(\frac{\partial f}{\partial x}\right) &= f(t)\odot_t \frac{(1-4x)t}{(1-tx)^2} &(5.70)\end{aligned}$$

and so $f(x)$ satisfies the equation

$$f(t)\odot_t\left(\frac{2+t-6tx}{(1-tx)^2}\right) - 2x = 0, \tag{5.71}$$

where $\frac{2+t-6tx}{(1-tx)^2} = \frac{(1-4x)t}{(1-tx)^2} + \frac{2}{1-tx}$. In the same way any D-finite function of a single variable, $x$, satisfies an equation of the form

$$f(t)\odot_t T(t,x) + a(x) = 0, \tag{5.72}$$

where $T(t,x)$ is a rational function of $t$ and $x$, and $a(x)$ is a polynomial in $x$.

We note that the function

$$f(x) = \sum_{n\geq 0} x^{2^n}, \tag{5.73}$$

which is not D-finite[7], satisfies

$$f(t) \odot_t \frac{xt(1-x)}{(1-xt)(1-xt^2)} = x, \tag{5.74}$$

and so functions satisfying equations of the form of equation (5.72) are a *superset* of D-finite functions. It is interesting to speculate that we could perhaps define a class of “H-finite” functions satisfying such recurrences. This idea is certainly one that is worthy of further consideration.

◁ ◁ ◇ ▷ ▷

### 5.3.4 Interpreting the r-Hadamard product of generating functions

In the same way that the sum and product of two generating functions have natural combinatorial interpretations in terms of the objects they enumerate, one can find an interpretation of the r-Hadamard product.

Consider two sets of combinatorial objects, $\mathcal{F}$ and $\mathcal{G}$, and say that these two sets of objects be enumerated according to some parameter $X$, associated with a variable $x$. We can write the two generating functions,

$$f(x) = \sum_{n\geq 0} f_n x^n \qquad \text{with } f_n = |\{\varphi \in \mathcal{F} \mid X(\varphi) = n\}| \tag{5.75}$$

$$g(x) = \sum_{n\geq 0} g_n x^n \qquad \text{with } g_n = |\{\gamma \in \mathcal{G} \mid X(\gamma) = n\}|. \tag{5.76}$$

Then

- $h(x) = f(x) + g(x)$ enumerates a new set $\mathcal{H} = \mathcal{F} \uplus \mathcal{G}$, which is the *disjoint* union of $\mathcal{F}$ and $\mathcal{G}$. *i.e.* $\mathcal{H}$ is the set of all the elements of $\mathcal{F}$ and $\mathcal{G}$,
- $h(x) = f(x)g(x)$ enumerates a new set $\mathcal{H} = \mathcal{F} \times \mathcal{G}$. *i.e.* $\mathcal{H}$ is the set of *all* the pairs, $(\varphi, \gamma)$ of an element $\varphi \in \mathcal{F}$ and an element $\gamma \in \mathcal{G}$.

The r-Hadamard product (as defined above), $h = f(x) \odot_x g(x) = \sum_{n\geq 0} f_n g_n$ represents a new set, $\mathcal{H}$, which is all the pairs $(\varphi, \gamma) \in \mathcal{F} \times \mathcal{G}$ such that $X(\varphi) = X(\gamma)$. So while the ordinary product produces *all* pairs of elements, the Hadamard product produces only those pairs of elements which have *matching* values of the parameter $X$.

[7]Since the power series defined by $f(x)$ has integer coefficients and converges inside the unit disc it must either be rational or have a natural boundary on the unit circle (see the proof of proposition 4.8 in chapter 4). It is not difficult to show that it cannot be rational.

### 5.3.5 A geometric picture of Klarner's integral recurrence

Let us re-examine Klarner's integral recurrence armed with Hadamard product and its combinatorial interpretation; we start by rewriting equation 5.56 in terms of the Hadamard product:

$$P(s;x,q) = \frac{xsq}{1-sq} + P(t;x,q) \odot_t T(t,s;x,q). \tag{5.77}$$

We can find a more geometric interpretation of the generating function of the weights, $T$. Let us define $B(t,s;x,q)$ to be the generating function of column-convex polygons of width 2 enumerated by their area ($q$) and the number of cells in each column ($t$ for the left and $s$ for the right). Using Temperley's recurrence we find that

$$B(t,s;x,q) = x^2 \sum_{n_1,n_2 \geq 1} (n_1+n_2-1)(tq)^{n_1}(sq)^{n_2} \tag{5.78}$$

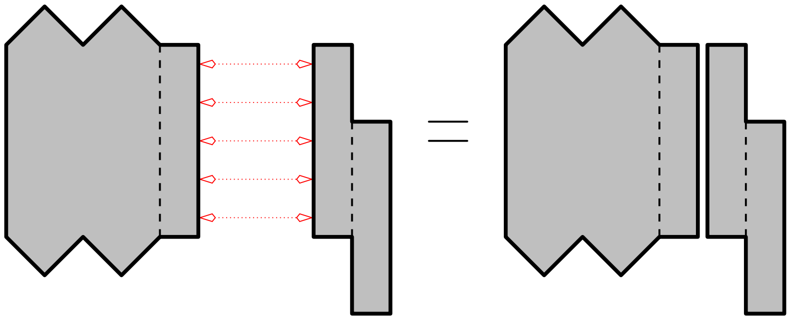
$$= \frac{x^2 stq^2(1-stq^2)}{(1-sq)^2(1-tq)^2}. \tag{5.79}$$

The function $T(t,s;x,q)$, in equation (5.56), is then simply $B(t/q,s;x,q)/x$. This is the generating function of column-convex polyominoes of width 2 enumerated by the number of cells in the left ($t$) and right ($s$) columns, but only the area and horizontal perimeter of *rightmost* column are enumerated. *i.e.* the generating function $T$ enumerates the same polygons as $B$, but it *under-weights* the configurations. This interpretation of the function $T$ is easier to generalise than that of equation (5.77).

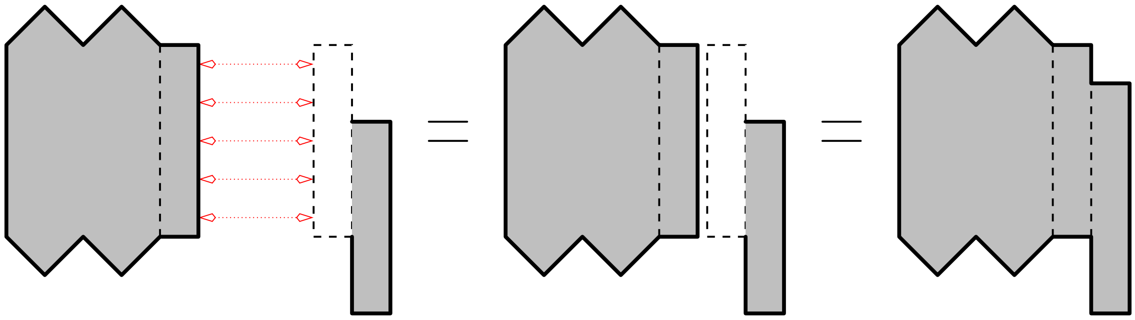
The Hadamard product $P(t;x,q) \odot_t B(t,s;x,q)$ enumerates all pairs $(U,V)$, where $U$ is a column-convex polyomino of width $n$ and $V$ is a column-convex polyomino of width 2, such that the right height of $U$ is equal to $h$ and exactly matches the left height of $V$ (see figure 5.7(a) ). If we "squash" the two polygons together, then a new polygon is created with $n+2$ columns, such that the $n$th and $(n+1)$th columns form a $2\times h$ rectangle. On the other hand, the Hadamard product $P(t;x,q) \odot_t T(t,s;x,q)$ enumerates the same pairs, except now the total area of the pair has been undercounted by $h$ cells, where the right height of $P$ is exactly $h$ (see figure 5.7(b) ). Squashing these two polygons together produces a new polygon with $n+2$ columns such that the $n$th and $(n+1)$th columns form a $2\times h$ rectangle, but now the area of only one of these columns is enumerated. Since one of the two columns in this rectangle is "hollow", we can "squash" further to create a new column-convex polygon with $n+1$ columns. Conversely, any $n+1$-column column-convex polygon can be split into an $n$-column column-convex polygon and an under-weighted column-convex polygon of width 2 (see figure 5.8).

In this way, we can construct any column-convex polygon from a single column (the seed block) and a sequence of under-weighted column-convex polygons of width 2 (the building blocks), which are combined by this "squashing" procedure (see figure 5.9). Klarner [105] describes a similar construction for a set of *polyominoes* (which are equivalent to multi-directed animals — see chapter 4) enumerated by area only, and does not generalise it to other parameters.

This squashing construction can also be made to work for perimeter (and even other parameters — see chapters 6 and 8). Consider figure 5.7. We see that when we squash a



(a) True weight



(b) Underweighted

Figure 5.7: A column-convex polygon and a column-convex polygon of width 2; the right height of the first is equal to the left height of the second. In (a) the two-column polygon is weighted normally, while in (b) the area of the first column of the two-column polygon is ignored. The result of "squashing" the two polygons together is on the right of each figure.

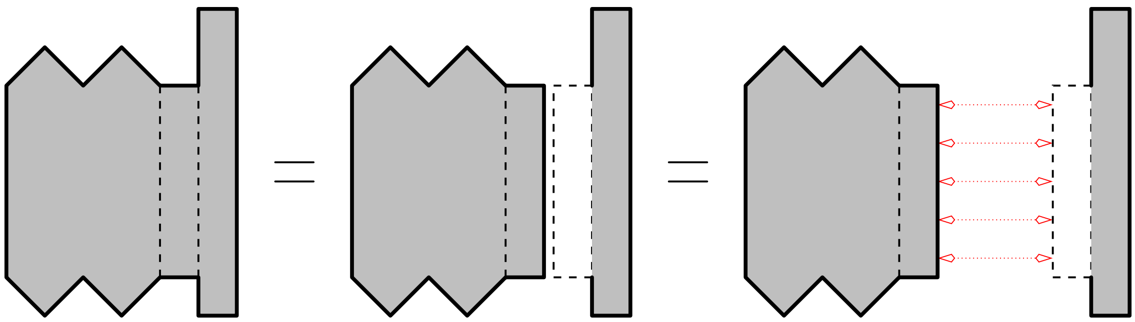


Figure 5.8: Reversing figure 5.7(b), any column-convex polygon of width $(n+1)$ can be split into two polygons, the first an $n$-column column-convex polygon and the second an under-weighted column-convex polygon of width 2.

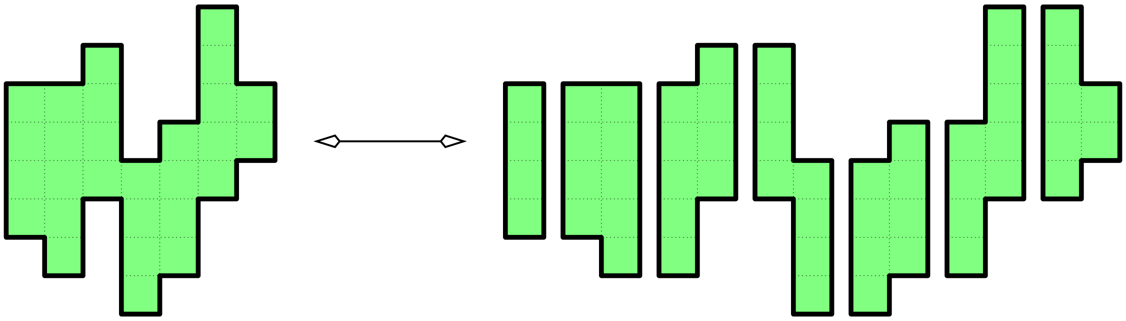


Figure 5.9: Any column-convex polygon can be constructed by "squashing" a single column and a sequence of under-weighted column-convex polygons of width 2.

column-convex polygon of width 2 (with left height $h$) onto the end of an existing polygon (with right height $h$) then we over-count the horizontal half-perimeter by 1, the area by $h$, and the vertical half-perimeter by $h$ (since there will be $2h$ vertical bonds trapped inside the polygon). We can overcome this by underweighting the column-convex polygons of width 2.

The left height of the column-convex polygons of width 2 is also the height of the join, and so the height of the join is enumerated by the variable $t$. Since the amount we need to underweight is a linear function of the join height, we can make a careful substitution for $t$ to make the correction. In particular, if $B(t, s; x, y, q)$ is the generating function of column-convex polygons of width 2, then the correct under-weighted generating function will be $T(t, s; x, y, q) = B(t/yq, s; x, y, q)/x$.

The generating function, $B$, can be found by many methods. We find $B$ (and hence $T$) by isolating all the section-minimal[8] column-convex polygons of width 2 (given in figure 5.10). Expanding these minimal blocks gives the building block generating function[9] and the $T$ function:

$$\begin{aligned} B(t,s;x,y,q) &= x^2[\![ styq^2 ]\!](1+2[\![ syq ]\!]+2[\![ tyq ]\!]+[\![ syq ]\!]^2+[\![ tyq ]\!]^2+2[\![ syq ]\!][\![ tyq ]\!]), \\ T(t,s;x,y,q) &= B(t/yq,s;x,y,q)/x \\ &= -\frac{xs^2y^2q^2}{(1-syq)^2}+\frac{s^2xq^2(2y-syq-1)}{(1-sq)^2(1-syq)(1-t)}+\frac{xsqt}{(1-sq)(1-t)^2} \\ &\quad +\frac{xs^2q^2(1-y)^2}{(1-syq)^2(1-sq)^2(1-sqt)}. \end{aligned} \tag{5.80}$$

The $T$ function defined above, like the $T$ function in equation (5.50), encodes the weight function associated with adding a building block, but now the weight function is contained within the generating function of the building blocks.

[8]Since we want to enumerate animals of finite width instead of finite height, we will consider section lines to be vertical lines, and sections to be the contents of a row between neighbouring section lines. This turns everything in chapter 3 by 90°.

[9]We have written $[\![ f ]\!]$ to mean $\frac{f}{1-f}$ which is the generating function for all nonempty sequences of objects enumerated by $f$.

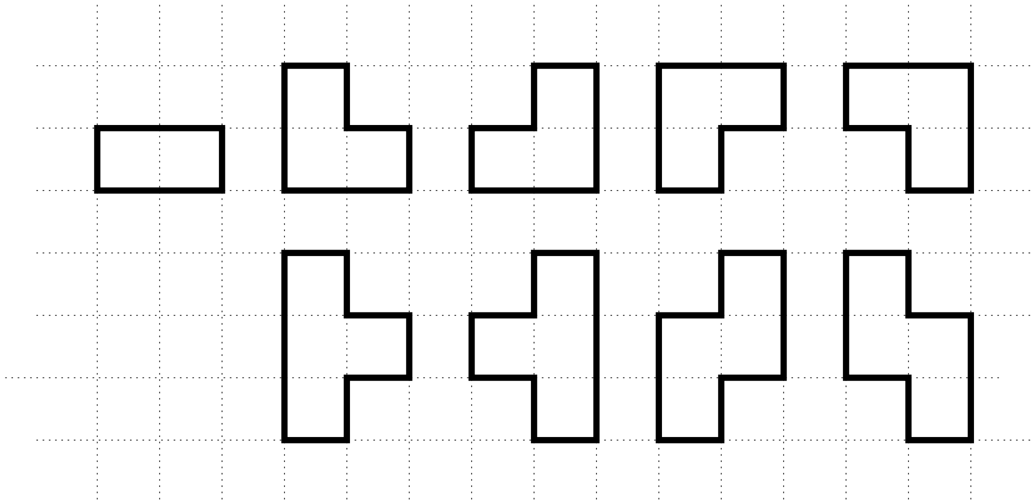


Figure 5.10: Building blocks of column-convex polygons

We can now replace Klarner's integral equation (for the area generating function) with the Hadamard equation (for the area and perimeter generating function)

$$P(s;x,y,q) = \frac{xsyq}{1-syq} + P(t;x,y,q) \odot_t T(t,s;x,y,q), \tag{5.81}$$

where $\frac{xsyq}{1-syq}$ is the generating function of the seed blocks, which are all single columns of cells.

$$\begin{aligned} P(t;x,y,q) \odot_t T(t,s;x,y,q) \quad = \quad & -\frac{xs^2y^2q^2}{(1-syq)^2}\left(P(t;x,y,q) \odot_t t^0\right) \\ & + \frac{s^2xq^2(2y-syq-1)}{(1-sq)^2(1-syq)}\left(P(t;x,y,q) \odot_t \frac{1}{1-t}\right) \\ & + \frac{xsq}{(1-sq)}\left(P(t;x,y,q) \odot_t \frac{t}{(1-t)^2}\right) \\ & + \frac{xs^2q^2(1-y)^2}{(1-syq)^2(1-sq)^2}\left(P(t;x,y,q) \odot_t \frac{1}{1-sqt}\right) \end{aligned} \tag{5.82}$$

Applying lemma 5.2 reduces this Hadamard equation to the linear functional equation (equation 5.40), found by the functional Temperley method.

# 5.4 Seed blocks, building blocks and a few examples of the Hadamard-Temperley method

In the previous section we have shown that the integral Temperley method can be turned into a "Hadamard Temperley" method, which in turn gives the same functional equations as the functional Temperley method (which in turn are equivalent to the recurrences found by Temperley's original method). The superiority of any one of these methods over another lies in the imprecise notions of their ease of use and aesthetics of the form of the solutions.

Using Temperley's original method allows one to find a recurrence quite easily, and in many circumstances these recurrences can be easily solved. It is when one arrives at a recurrence of the form of equation (5.34), that the method becomes difficult. Either summing this recurrence or using the functional-Temperley method we can obtain a functional equation that can be solved more easily.

While finding the functional equations using the functional Temperley method is not trivial, it is not too hard for column-convex families of polygons. For more complicated polyominoes, it becomes increasingly difficult (for some examples see chapter 6 and 7). The problem of finding functional equations using the Hadamard Temperley method, on the other hand, is reduced to one of finding the generating function of the building blocks — and for complicated polygons this can be far easier than computing the functional equation directly.

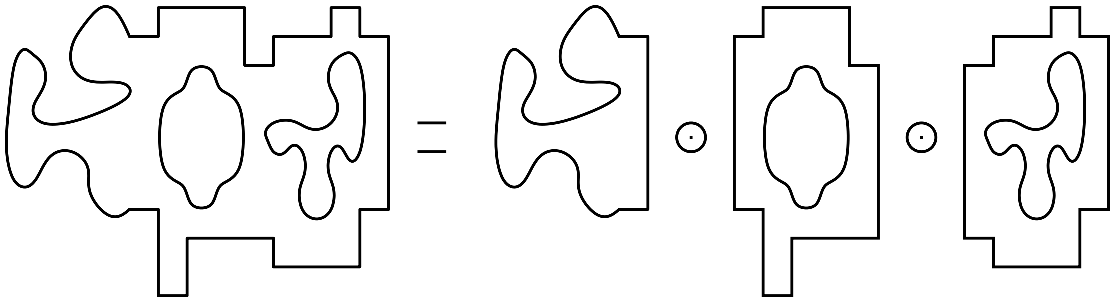


Figure 5.11: Any building blocks are fine, as long as they have a well defined right and left height — we can use this technique to build polyominoes that are non-convex.

## 5.4.1 From seed and building blocks to a recurrence

When we approach a polyomino problem, we need to determine whether or not we can apply a "squashing" construction like that described in section 5.3.5. If we can use this construction then we need to find the sets of seed and building blocks, and their generating functions. These blocks need not be column-convex, nor only 2 columns wide (see figure 5.11 for an extreme example). The only restriction is that the seed blocks have a well defined[10] right height and the building blocks have well defined left and right heights, and if we are to apply lemma 5.2 we require that their generating functions are rational. In this way functional equations for a wide range of objects can be constructed.

Let $\mathcal{S}$ be the set of seed-blocks and let $\mathcal{B}$ be the set of building-blocks. Let $S(s)$ be the generating function of $\mathcal{S}$ enumerated by right height ($s$) and other parameters, and let $B(t,s)$ be the generating function of $\mathcal{B}$ enumerated by left and right heights ($t$ and $s$ resp.) and other parameters. To avoid overweighting the configurations we underweight building blocks and form the function $T(t,s)$. The function $T$ encodes the weights associated with adding a block to a polyomino, playing much the same role that the transfer matrix $M$ does in equation (5.23), and so we call $T$ the *transfer function*.

If we denote the generating function of the polyominoes that we are studying by $P(s)$, then it satisfies the following recurrence

$$P(s) = S(s) + T(t,s) \odot_t P(t) \tag{5.83}$$

If $B$ (and hence $T$) is a rational function of $t$, then we can apply lemma 5.2 to this recurrence and translate it into a linear functional equation for $P(s)$.

### 5.4.2 Construction of seed-blocks and building-blocks

There are many methods that can be used to calculate the generating functions of sets of seed and building blocks. Here we will use the expansions of section-minimal polygons (see chapter 3).

Let $P$ be a section-minimal polygon. In chapter 3 we showed that the horizontal half-perimeter generating function of the expansion of $P$ is given by

$$G_s(P) = \prod_{k\geq 1} [\![ x^k ]\!]^{\sigma_k(P)}, \tag{5.84}$$

where we have written $[\![ f ]\!] = \frac{f}{1-f}$.

Since our seed blocks and building blocks will generally be of finite width, rather than finite height, we turn the definition of a "section" by 90° and redefine section lines to run *vertically* rather than horizontally. Consequently a section is redefined to be the contents of a *row* between two section lines. If a section-minimal polygon, $P$, contains a total of $k$ sections, $a_1, a_2, \ldots, a_k$, and if $f_i$ is the generating function of the contents of $a_i$, then the expansion of the minimal polygon has generating function given by

$$G_s(P) = \prod_{i=1}^{k} [\![ f_i ]\!]. \tag{5.85}$$

The generating function of $\mathcal{S}$ or $\mathcal{B}$ is then simply the sum of the generating functions of the expansions of section-minimal polygons within each set.

[10] The right height (or left height) of a polyomino is "well defined" if its rightmost (or leftmost) column is convex.

### 5.4.3 A few examples

Let us consider a few examples to demonstrate the Hadamard-Temperley method; in order to find the functional equations satisfied by the generating functions of the examples we need to find the sets of seed blocks and building blocks (and their generating functions).

**Example 5.3 (bargraph polygons enumerated by area and perimeter).**
The seed blocks are all single columns of cells, while the building blocks are all two column bargraphs. The section-minimal building blocks are given in figure 5.12. And so

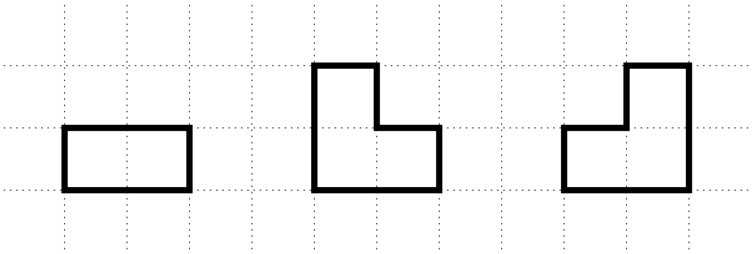


Figure 5.12: Building blocks of bargraph polygons.

$$\begin{aligned} S(s;x,y,q) &= \frac{xsyq}{1-syq}, && (5.86)\\ B(t,s;x,y,q) &= x^2[\![\, styq^2 \,]\!]\,(1+[\![\, syq \,]\!]+[\![\, tyq \,]\!]) \\ &= \frac{x^2styq^2(1-sy^2q^2t)}{(1-syq)(1-tyq)(1-styq^2)}, && (5.87)\\ T(t,s;x,y,q) &= B(t/yq,s;x,y,q)/x \\ &= \frac{qsxy}{1-syq}+\frac{xsq}{(1-sq)(1-t)}-\frac{xsq(1-y)}{(1-syq)(1-sq)(1-stq)}. && (5.88) \end{aligned}$$

So the bargraph generating function, $P(s;x,y,q)$, satisfies

$$P(s;x,y,q)=\frac{xsyq}{1-syq}+\frac{xsq}{1-sq}P(1;x,y,q)-\frac{xsq(1-y)}{(1-syq)(1-sq)}P(sq;x,y,q). \qquad (5.89)$$

**Example 5.4 (staircase polygons enumerated by area and perimeter).**
The seed blocks are all single columns of cells, while the building blocks are all staircase polygons of width 2. The section-minimal building blocks are given in figure 5.13. And so

$$\begin{aligned} S(s;x,y,q) &= \frac{xsyq}{1-syq}, && (5.90)\\ B(t,s;x,y,q) &= x^2[\![\, styq^2 \,]\!]\,(1+[\![\, syq \,]\!]+[\![\, tyq \,]\!]+[\![\, syq \,]\!][\![\, tyq \,]\!]) \\ &= \frac{x^2stq^2y}{(1-sqy)(1-tqy)(1-styq^2)}, && (5.91)\\ T(t,s;x,y,q) &= \frac{xsq}{(1-sq)(1-syq)}\left(\frac{1}{1-t}-\frac{1}{1-stq}\right). && (5.92) \end{aligned}$$

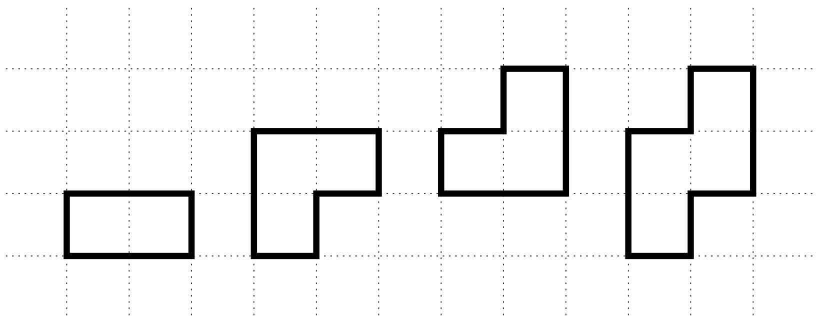


Figure 5.13: Building blocks of staircase polygons.

So the staircase polygon generating function, $P(s; x, y, q)$, satisfies

$$P(s; x, y, q) = \frac{xsyq}{1 - syq} + \frac{xsq}{(1 - sq)(1 - syq)} \Big( P(1; x, y, q) - P(sq; x, y, q) \Big). \tag{5.93}$$

**Example 5.5 (directed column-convex polygons enumerated by area and perimeter).**
Let us take directed column-convex polygons to be the subset of column-convex polygons such that the *lower* edge of the polygon is a path that uses only south and east steps (when traversed from left to right).

The seed blocks are all single columns of cells, while the building blocks are all two-column directed column-convex polygons. The section-minimal building blocks are given in figure 5.14. And so

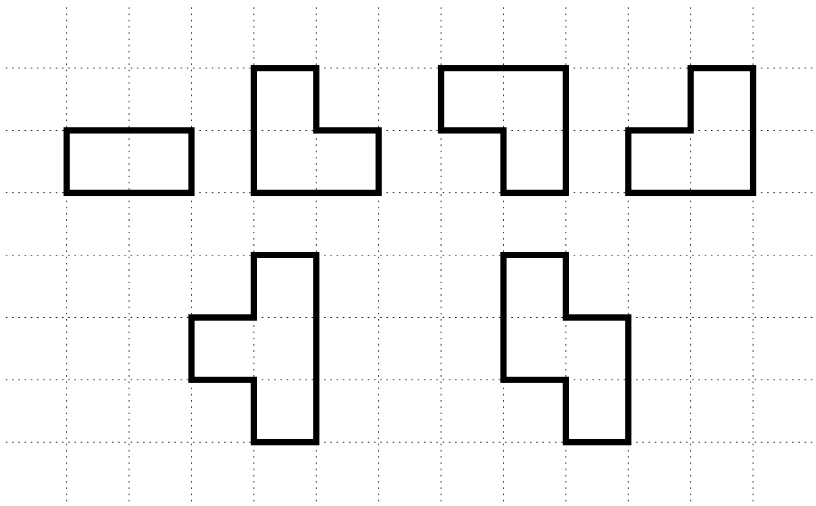


Figure 5.14: Building blocks of directed column-convex polygons.

$$\begin{aligned}
S(s;x,y,q) &= \frac{xsyq}{1-syq}, && (5.94)\\
B(t,s;x,y,q) &= x^2[\![\, styq^2 \,]\!]\left(1+2[\![\, syq \,]\!]+[\![\, syq \,]\!]^2+[\![\, tyq \,]\!]+[\![\, syq \,]\!][\![\, tyq \,]\!]\right)\\
&= \frac{x^2 stq(1-styq)}{(1-sqy)^2(1-tqy)(1-styq^2)}, && (5.95)\\
T(t,s;x,y,q) &= -\frac{xsyq}{(1-syq)^2}+\frac{xsq}{(1-sq)(1-syq)(1-t)}+\\
&\quad \frac{xsq(y-1)}{(1-sq)(1-syq)^2(1-stq)}. && (5.96)
\end{aligned}$$

So the directed column-convex polygon generating function, $P(s;x,y,q)$, satisfies

$$P(s;x,y,q) = \frac{xsyq}{1-syq} + \frac{xsq}{(1-sq)(1-syq)}P(1;x,y,q) + \frac{xsq(y-1)}{(1-sq)(1-syq)^2}P(sq;x,y,q). \tag{5.97}$$

**Example 5.6 (Connected heaps enumerated by the number of dimers).**
In chapter 4 we described several families of heaps that were in bijection with families of site animals on the square and triangular lattices. The generating functions of these heaps satisfy functional equations which we derived by direct combinatorial construction. We first discovered the functional equation for one of these families, *connected heaps*, by the Hadamard-Temperley method. We discovered that we could build connected heaps from a single column of dimers, to which we "glued" a sequence of underweighted two column heaps.

In the next section we describe two different techniques to solve the functional equations that are found using the functional-Temperley or Hadamard-Temperley methods.

5.5

# Solving functional equations

In this section we describe two methods for resolution of linear functional equations. We follow the descriptions given in Bousquet-Mélou's paper [23]. Both of the methods we shall describe, *iteration* and the *kernel method*, solve the functional equations in a way that is not too dissimilar to the way in which one solves a system of linear equations — we attempt to remove the unknowns one by one (by various manipulations and cancellations), until only a single unknown is left. Once we have solved for this unknown, we find the others by back-substitution.

The functional equations we have seen above take one of the two following forms:

$$\begin{aligned}
P(s) &= xa(s)+xb(s)P(1)+xc(s)P(sq), && (5.98)\\
P(s) &= xa(s)+xb(s)P(1)+xc(s)P(sq)+xd(s)\left(\frac{\partial P}{\partial s}\right)(1), && (5.99)
\end{aligned}$$

where we have used $f(s)$ as shorthand to denote $f(s; x, y, q)$. It should be noted that the coefficients of these equations ($a(s)$, $b(s)$, $c(s)$ and $d(s)$) are only singular at $s = 1/q$, $s = 1/q^2$ and $s = 1/yq$.

Let us concentrate upon equation (5.98). In order to obtain an expression for $P(1)$ from this equation we need to eliminate $P(s)$ and $P(sq)$. Perhaps the most obvious way to do this is by making the substitutions $q = 1$ and $s = 1$. Setting $q = 1$, removes the $P(sq)$ term, leaving an equation in $P(1)$ and $P(s)$. Similarly setting $s = 1$ in this equation will leave only $P(1)$, which we can isolate.

If this method of specialisation does work, then $P(1)$ will always be rational. This contradicts many well known results, and so we must conclude that specialisation does not always work. More specifically we find that for many problems only one of these two substitutions is applicable; for example, if we wish to enumerate some set of polygons by area, we cannot set $q = 1$ since doing so removes all information about area from the functional equation. We need the kernel method and iteration to remove unknowns when specialisation fails.

### 5.5.1 Iteration

Let us first consider functional equations of the form of equation (5.98) and then proceed to the more complicated form of equation (5.99), since the technique is almost identical.

Consider setting $q = 1$ in equation (5.98) (though this may not be possible in certain cases, it does seem to be possible in the cases we examine in this thesis and those appearing in [23]). This specialisation removes all information about area. It is not possible (excepting in rare and contrived examples), to solve the functional equation when $q = 1$ and then (somehow) recover the area generating function. Consequently if we wish to enumerate a family of polygons by area we are not able to remove $P(sq)$ from the functional equation by setting $q = 1$. Instead we remove $P(sq)$ using iteration.

Unlike $q$, we are able to set $s$ to (almost) any value we like in the functional equation; let us try to eliminate $P(sq)$ by setting $s = sq$:

$$P(sq) = xa(sq) + xb(sq)P(1) + xc(sq)P(sq^2). \tag{5.100}$$

We can then substitute this back into the original equation to give

$$\begin{aligned} P(s) \;&=\; xa(s) + x^2c(s)a(sq) + \big(xb(s) + x^2c(s)b(sq)\big)\,P(1) \\ &\quad + x^2c(s)c(sq)P(sq^2). \end{aligned} \tag{5.101}$$

And so we have replaced the unknown $P(sq)$ with a new unknown $P(sq^2)$. At first this does not seem to be much of an improvement, because we have not reduced the number of unknowns, however we note that when considered a formal power series in $x$, the coefficient in front of $P(sq^2)$ is $O(x^2)$, while in the original equation the coefficient of $P(sq)$ was $O(x)$.

If we *iterate* this substitution $N$ times we arrive at the equation

$$\begin{aligned} P(s) &= \left(\sum_{n=0}^{N} x^{n+1} c(s)c(sq)\ldots c(sq^{n-1})a(sq^n)\right) + \\ &\left(\sum_{n=0}^{N} x^{n+1} c(s)c(sq)\ldots c(sq^{n-1})b(sq^n)\right) P(1) + \\ &x^{N+1}\Big(c(s)c(sq)\ldots c(sq^N)\Big) P(sq^{N+1}). \end{aligned} \tag{5.102}$$

Now we see that the coefficient in front of $P(sq^{N+1})$ is $O(x^{N+1})$. As $N \to \infty$ this coefficient will converge (as a formal power series) to 0; more specifically it converges to the function $\sum 0x^n$ . Taking the limit $N \to \infty$ of this equation gives

$$\begin{aligned} P(s) &= \left(\sum_{n\geq 0} x^{n+1} c(s)c(sq)\ldots c(sq^{n-1})a(sq^n)\right) + \\ &\left(\sum_{n\geq 0} x^{n+1} c(s)c(sq)\ldots c(sq^{n-1})b(sq^n)\right) P(1) \end{aligned} \tag{5.103}$$

Let us write

$$\mathfrak{A}(s) = \sum_{n\geq 0} x^{n+1} c(s)c(sq)\ldots c(sq^{n-1})a(sq^n) \tag{5.104}$$

$$\mathfrak{B}(s) = \sum_{n\geq 0} x^{n+1} c(s)c(sq)\ldots c(sq^{n-1})b(sq^n) \tag{5.105}$$

We *can* set $s = 1$ (since $a(q^j)$, $b(q^j)$ and $c(q^j)$ are all convergent for $j \geq 0$) and so replace $P(s)$ with $P(1)$, which we can now solve for:

$$\begin{aligned} P(1) &= \mathfrak{A}(1) + \mathfrak{B}(1)P(1) \\ P(1) &= \frac{\mathfrak{A}(1)}{1 - \mathfrak{B}(1)} \end{aligned} \tag{5.106}$$

◁ ◁ ◇ ▷ ▷

**Remark.**
It would be nice to find a combinatorial interpretation of equation (5.106) since it appears to be of a form very similar to equation (5.13).

◁ ◁ ◇ ▷ ▷

If we substitute this expression for $P(1)$ back into equation (5.103), we find $P(s)$:

$$\begin{aligned} P(s) &= \mathfrak{A}(s) + \mathfrak{B}(s)P(1) \\ &= \mathfrak{A}(s) + \frac{\mathfrak{A}(1)\mathfrak{B}(s)}{1 - \mathfrak{B}(1)}. \end{aligned} \tag{5.107}$$

The above method also works for functional equations of the form of equation (5.99). If we define

$$\mathfrak{D}(s) = \left( \sum_{n \geq 0} x^{n+1} c(s) c(sq) \dots c(sq^{n-1}) d(sq^n) \right), \tag{5.108}$$

then by iteration one obtains

$$P(s) = \mathfrak{A}(s) + \mathfrak{B}(s) P(1) + \mathfrak{D}(s) \left( \frac{\partial P}{\partial s} \right)(1) \tag{5.109}$$

This equation is of the same form as equation (5.36) for column-convex polygons enumerated by area, and one proceeds to a solution in an identical manner. We can construct a system of 2 equations in the unknowns $P(1)$ and $P'(1)$, by setting $s = 1$ into the above equation, and also into its derivative with respect to $s$:

$$P(1) = \mathfrak{A}(1) + \mathfrak{B}(1) P(1) + \mathfrak{D}(1) P'(1), \tag{5.110}$$
$$P'(1) = \mathfrak{A}'(1) + \mathfrak{B}'(1) P(1) + \mathfrak{D}'(1) P'(1), \tag{5.111}$$

where we have used $f'(1)$ to denote $\left(\frac{\partial f}{\partial s}\right)|_{s=1}$. Solving this system gives

$$P(1) = \frac{\mathfrak{A}(1) + \mathfrak{D}(1)\mathfrak{A}'(1) - \mathfrak{A}(1)\mathfrak{D}'(1)}{1 - \mathfrak{B}(1) - \mathfrak{D}'(1) - \mathfrak{D}(1)\mathfrak{B}'(1) + \mathfrak{B}(1)\mathfrak{D}'(1)} \tag{5.112}$$

Back-substitution then gives $P(s)$.

**Remark.**
This same method works for similar functional equations with higher order derivatives, $\frac{\partial^k P}{\partial s^k}(1)$, providing the coefficients of the equation are not singular at $s = q^j$, $j \geq 0$. For example, if after iterating the functional equation we obtain an equation of the form

$$P(s) = \mathfrak{A}(s) + \mathfrak{B}(s) P(1) + \mathfrak{D}(s) \left( \frac{\partial P}{\partial s} \right)(1) + \mathfrak{E}(s) \left( \frac{\partial^2 P}{\partial s^2} \right)(1), \tag{5.113}$$

then we can find $P(1)$ by solving the following set of linear equations:

$$\begin{aligned}
P(1) &= \mathfrak{A}(1) + \mathfrak{B}(1) P(1) + \mathfrak{D}(1) P'(1) + \mathfrak{E}(1) P''(1), \\
P'(1) &= \mathfrak{A}'(1) + \mathfrak{B}'(1) P(1) + \mathfrak{D}'(1) P'(1) + \mathfrak{E}'(1) P''(1), \\
P''(1) &= \mathfrak{A}''(1) + \mathfrak{B}''(1) P(1) + \mathfrak{D}''(1) P'(1) + \mathfrak{E}''(1) P''(1).
\end{aligned}$$

### 5.5.2 The kernel method

Though it is possible to solve the functional equations for both perimeter and area, the form of the solutions prevents them from being reduced to a perimeter-only generating function. Consider the solution to column-convex polygons (equation (5.41)); the sums are not convergent for $q = 1$. Instead we must return to the original equation (equation (5.99)), set $q = 1$ and solve again. We quickly find that we cannot set $s = 1$, and we must use the kernel method.

Rather than describe the kernel method in general terms, we shall explore it with the example of staircase polygons; let us again consider the functional equation for staircase polygons enumerated by area, vertical and horizontal half-perimeters:

$$P(s;x,y,q) = \frac{xysq}{1-ysq} + \frac{xsq}{(1-sq)(1-syq)}\Big(P(1;x,y,q) - P(sq;x,y,q)\Big).$$

Setting $q = 1$ in this equation replaces $P(sq)$ with $P(s)$:

$$P(s;x,y,1) = \frac{xsy}{1-sy} + \frac{xs}{(1-s)(1-sy)}\Big(P(1;x,y,1) - P(s;x,y,1)\Big).$$

This leaves two unknowns, $P(s)$ and $P(1)$, and in the process we have lost *all* information about area.

◁ ◁ ◇ ▷ ▷

Since we have successfully removed $P(sq)$ by setting $q = 1$, let us attempt another specialisation to remove $P(s)$; namely let us try to set $s = 1$, and see what goes wrong. Immediately we find that the coefficients in the equation are singular at $s = 1$. We can try to side-step this by multiplying through by $(1 - s)$:

$$(1-s)P(s) = \frac{xsy(1-s)}{1-sy} + \frac{xs}{1-sy}(P(1) - P(s)). \tag{5.114}$$

However when we set $s = 1$, we find

$$\begin{aligned} 0 &= 0 + \frac{x}{1-y}(P(1) - P(1)) \\ 0 &= 0, \end{aligned}$$

and the equation reduces to a tautology[11]. If we could make the substitution $s = 1$ and solve for $P(1)$, then since the equation is linear, we would obtain a *rational* solution. This would contradict previous work (see [48] and references therein) that demonstrates that the perimeter generating function for staircase polygons (and many other families of column-convex polygons) is the solution of an *algebraic* equation (and is an irrational series).

Rather than substituting $s = 1$, let us try to take the limit as $s \to 1$. When we do this, we see that the singularity at $s = 1$ is hiding a derivative:

$$\begin{aligned} P(s) &= \frac{xsy}{1-sy} + \frac{xs}{(1-sy)}\frac{P(1) - P(s)}{1-s} \\ P(1) &= \frac{xy}{1-y} + \frac{x}{1-y}\left(\frac{\partial P}{\partial s}\right)(1). \end{aligned} \tag{5.115}$$

[11] Though a tautology is a true statement (oops — a tautology!), they can be, as indeed they are here, quite useless.

This is actually easier to see if we return to the original Hadamard equation. Substituting $q = 1$ and $s = 1$ into equation (5.92), causes the singularities of $T$ at $t = 1$ and $t = 1/s$ to coalesce.

$$T(t, 1; x, y, 1) = \frac{xt}{(1-y)(1-t)^2}. \tag{5.116}$$

And so taking the Hadamard product, $P(t) \odot_t T(t, 1)$ gives precisely $\frac{x}{1-y}\left(\frac{\partial P}{\partial s}\right)(1)$ (without having to worry about taking limits).

◁ ◁ ◇ ▷ ▷

Feretić and Svrtan [68] showed (for the example of column-convex polygons) that it is possible to recover *algebraic* solutions from the *linear* functional equations (also see [23]). The method they used is known as the *kernel method*. Let us collect all of the $P(s)$ terms together on the left hand side of the equation:

$$\begin{aligned} ((1-sy)(1-s)+xs)P(s) &= xsy(1-s)+xsP(1) \\ (1-(1-x+y)s+ys^2)P(s) &= xs(y(1-s)+P(1)) \end{aligned}$$

To remove the $P(s)$ term, we can set the polynomial in front of it to zero by making a careful choice of $s$. *i.e.* we choose $s = s_0$ such that $s_0$ is a solution of $1-(1-x+y)s+s^2y = 0$ with respect to $s$. Since this equation is quadratic in $s$, there are two solutions:

$$s_{\pm} = \frac{1}{2}\frac{1+y-x \pm \sqrt{1-2y-2x+y^2-2yx+x^2}}{y}. \tag{5.117}$$

The choice of $s_0$ depends upon the convergence of the two solutions in the limit $x, y \to 0$. Since no polygon can have negative perimeter $P(1; x, y, 1)$ must be convergent as $x, y \to 0$, and so we must choose the solution to the above equation that will give us this convergence. We choose $s_0 = s_-$, since it is convergent as $x, y \to 0$. Making the substitution $s = s_0$, sets the left-hand side to zero (this, or the polynomial part of it, is the "kernel" in the "kernel method") and we are left with

$$0 = xs_0\big(y(1-s_0)+P(1)\big),$$

and hence

$$P(1) = y(s_0-1) = \frac{1-y-x-\sqrt{1-2y-2x+y^2-2yx+x^2}}{2}. \tag{5.118}$$

We then can substitute this back into equation (5.114) to find $P(s)$:

$$P(s) = \frac{1}{2}\frac{xs\left(1+y-x-2ys-\sqrt{1-2y-2x+y^2-2yx+x^2}\right)}{1-s-ys+ys^2+xs}. \tag{5.119}$$

This method can also be applied to equation (5.99), however one must use the two solutions of the kernel, each one giving an equation for $P(1)$ and $\frac{\partial P}{\partial s}(1)$. These two equations can then be solved as a system of linear equations.

## 5.6

# Staircase polygons and site-perimeter

Now that we have described the methods for finding and solving linear functional equations, let us find the generating function of a non-trivial set of polyominoes — staircase polygons enumerated by horizontal and vertical half-perimeters, area and site perimeter. The *site perimeter* of a polygon is the number of nearest-neighbour unoccupied cells. It is a quantity of interest in percolation problems (we will give a fuller discussion of site perimeter in chapter 6).

Delest *et al.* [52] first enumerated staircase polygons by their perimeter and site-perimeter, while Dubernard and Dutour [56] solved for area and perimeters using the functional Temperley method. Here we will use the Hadamard Temperley method, iteration and the kernel method to rederive these results by what we hope is a simpler method.

We define the generating function $P(s;x,y,p,q)$ of staircase polygons enumerated by horizontal and vertical half-perimeter ($x$ and $y$ resp.), site perimeter ($p$), area ($q$) and right height ($s$).

### 5.6.1 Blocks and functional equation

To find the functional equation of a set of polyominoes we first need to find the seed and building blocks, and their generating functions. Much like the example of column-convex polygons above, the seed blocks are all single columns of cells, while the building blocks are all two-column staircase polygons. The section-minimal building blocks are shown in figure 5.15.

Expanding the minimal blocks we obtain the generating functions of $\mathcal{S}$ and $\mathcal{B}$:

$$S(s;x,y,p,q) = \frac{xsyp^4q}{1-syp^2q} \tag{5.120}$$

$$\begin{aligned} B(t,s;x,y,p,q) &= x^2p^2[\![\, styp^2q^2 \,]\!] \left(p+[\![\, syp^2q \,]\!]\right)\left(p+[\![\, typ^2q \,]\!]\right) \\ &= \frac{x^2p^6styq^2(1-syp^2q+psyq)(1-tyqp^2+tyqp)}{(1-syp^2q)(1-tyqp^2)(1-styq^2p^2)}. \end{aligned} \tag{5.121}$$

When joining a two-column staircase polygon of left height $h$ to a staircase polygon of right height $h$, the perimeters and area are over-counted — the vertical half-perimeter and area by $h$, the site perimeter by $2h+2$ and the horizontal half-perimeter by 1. So we need to underweight the coefficient of $t^h$ in $B(t,s;x,y,p,q)$ by a factor of $xy^hq^hp^{2h+2}$, and so the transfer function, $T$, is

$$\begin{aligned} T(t,s;x,y,p,q) &= B(t/yqp^2,s;x,y,p,q)/xp^2 \\ &= \frac{px(p-1)(1-syp^2q+psyq)}{(1-syp^2q)} + \frac{pxsq(1-syp^2q+psyq)}{(1-syp^2q)(1-sq)(1-t)} \\ &\quad -\frac{px(1-p+sqp)(1-syp^2q+psyq)}{(1-syp^2q)(1-sq)(1-stq)}. \end{aligned} \tag{5.122}$$

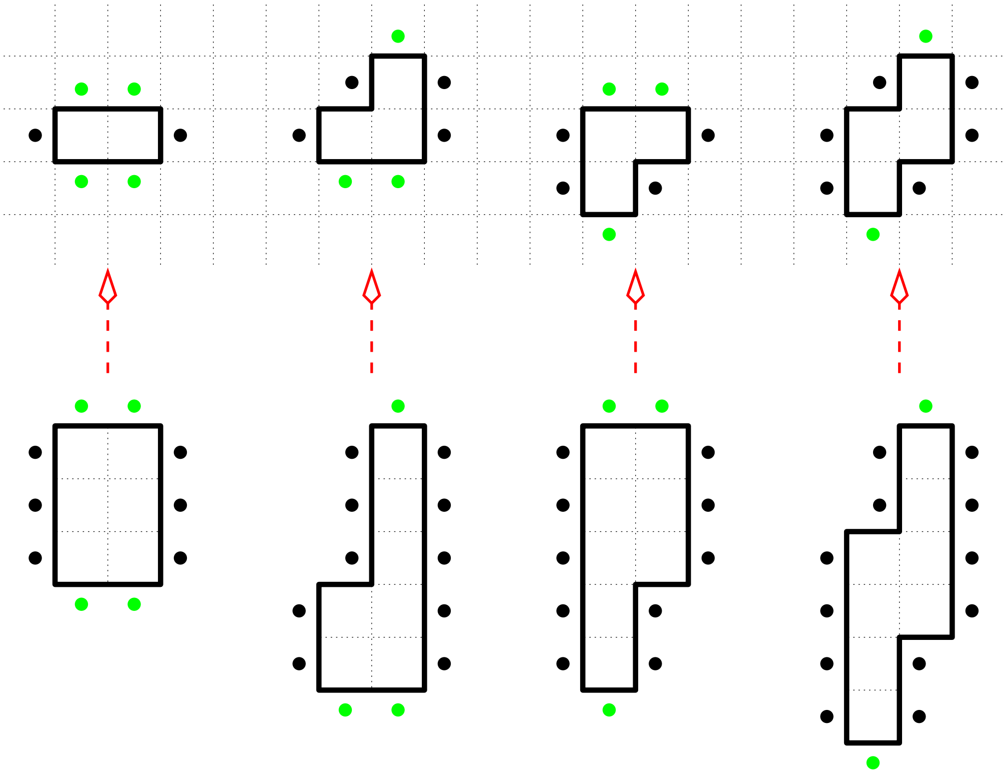


Figure 5.15: Top: The section-minimal building blocks for staircase polygons enumerated by site-perimeter. The dark spots indicate site-perimeter cells that are section-duplicated, while the lighter spots are not duplicated. Bottom: Examples of some building blocks that are section-equivalent to the section-minimal building blocks as indicated.

Hence the staircase polygon generating function, $P(s;x,y,p,q)$, satisfies

$$\begin{aligned} P(s;x,y,p,q) &= \frac{xsyqp^4}{1-syqp^2} + \frac{pxsq(1-syp^2q+psyq)}{(1-syp^2q)(1-sq)}P(1;x,y,p,q) \\ &\quad - \frac{px(1-p+sqp)(1-syp^2q+psyq)}{(1-syp^2q)(1-sq)}P(sq;x,y,p,q) \end{aligned} \tag{5.123}$$

This equation is the same (after a little massaging) as that found by Dubernard and Dutour [56] by the functional Temperley method.

### 5.6.2 The perimeter and site-perimeter generating function

If we wish to concentrate on the perimeter and site-perimeter generating function we can start by throwing away all information about area. We do this by setting $q = 1$:

$$P(s) = \frac{xsyp^4}{1-syp^2} + \frac{pxs(1-syp^2+psy)}{(1-syp^2)(1-s)}P(1) - \frac{px(1-p+ps)(1-syp^2+psy)}{(1-syp^2)(1-s)}P(s), \tag{5.124}$$

where $P(s)$ is now shorthand for $P(s; x, y, p, 1)$. We see that this equation is singular at $s = 1$, and so we cannot simply set $s = 1$ and isolate $P(1)$. This is expected since this would imply that $P(1)$ is rational, which cannot be the case since we know the perimeter generating function is algebraic (and not rational).

Instead of setting $s = 1$, we rearrange the equation so that all the $P(s)$ terms are on the left hand side, and apply the kernel method:

$$P(s)\left(1+\frac{px(1-p+ps)(1-syp^2+psy)}{(1-syp^2)(1-s)}\right) = \frac{xsyp^4}{1-syp^2} + \frac{pxs(1-syp^2+psy)}{(1-syp^2)(1-s)}P(1). \tag{5.125}$$

The left-hand side of this equation can now be set to zero by setting $s = s_0$, where $s_0$ satisfies

$$px(1-p+ps_0)(1-s_0yp^2+ps_0y)+(1-s_0yp^2)(1-s_0)=0, \tag{5.126}$$

and is convergent as $x, y, p \to 0$. The expression for $s_0$ is, not surprisingly, quite a large mess, so we will not state it here. Substituting $s = s_0$ and solving for $P(1)$ gives

$$P(1) = \frac{p^3y(s_0-1)}{1-s_0yp^2+ps_0y}, \tag{5.127}$$

in agreement[12] with [52]. Back-substitution will give an expression for $P(s)$.

### 5.6.3 The perimeter and area generating function

To extract as much information from the functional equation as possible and find the area, perimeter and site-perimeter generating function, $P(s; x, y, p, q) \equiv P(s)$ we must use iteration.

Let us start by giving some short-hand notation for the coefficients; let us write

$$xa(s) = \frac{xsyqp^4}{1-syqp^2}, \tag{5.128}$$

$$xb(s) = \frac{pxsq(1-sypq(p-1))}{(1-syp^2q)(1-sq)}, \tag{5.129}$$

$$xc(s) = -\frac{px(1-p+sqp)(1-sypq(p-1))}{(1-syp^2q)(1-sq)}. \tag{5.130}$$

We note that these coefficients are not singular at $s = q^j$, for $j \geq 0$.

Following the method described in section 5.5.1, iteration gives

$$P(s) = \mathfrak{A}(s) + \mathfrak{B}(s)P(1), \tag{5.131}$$

where $\mathfrak{A}$ and $\mathfrak{B}$ are defined by

$$\mathfrak{A}(s) = sy\sum_{n\geq 1}\frac{(-1)^{n-1}x^np^{n+3}q^n\left((p-1)syqp\right)_{n-1}\prod_{k=1}^{n-1}\left(1-p+sq^k\right)}{(sq)_{n-1}(syp^2q)_n}, \tag{5.132}$$

$$\mathfrak{B}(s) = s\sum_{n\geq 1}\frac{(-1)^{n-1}x^np^nq^{n-1}\left((p-1)syqp\right)_n\prod_{k=1}^{n-1}\left(1-p+sq^k\right)}{(sq)_n(syp^2q)_n}, \tag{5.133}$$

[12] Though, there seems to be slight misprint in [52].

and $(z)_n = \prod_{k=0}^{n-1}(1 - zq^k)$. Since $a(q^j)$, $b(q^j)$ and $c(q^j)$ are all convergent, the functions $\mathfrak{A}$ and $\mathfrak{B}$ are also convergent at $s = 1$. Setting $s = 1$, removing $P(s)$, and solving for $P(1)$ gives

$$P(1) = \frac{\mathfrak{A}(1)}{1 - \mathfrak{B}(1)}, \tag{5.134}$$

which is in agreement with the results of Dubernard and Dutour [56].

## 5.7 Final remarks

We have surveyed a number of variations of the column-by-column construction that is sometimes called the "Temperley method". We have also introduced a new variation of this method which we call the "Hadamard Temperley" method. The strength of this method is that one can proceed directly from the blocks that make up the polyominoes to a linear functional equation satisfied by the generating function. In many cases these equations are easily solved.

It would be nice to be able to *in principle* find a closed form solution to any functional equation found by this technique, however at this point we are unable to do so. Even if a closed form solution is not forthcoming, one can use the functional equation to determine properties of the solution (see chapter 7); the functional equation also allows one to compute the coefficients of the generating function very efficiently.

In the next chapter we will use the Hadamard-Temperley method to find the generating function of bargraph polygons enumerated by site-perimeter in closed form. This is the first set of non-convex polygons for which the site-perimeter is known. This result can then be extended to find a closed form solution for a type of self-interacting partially directed walk of interest to physicists.

# CHAPTER 6

# Enumerating site-perimeter

# 6.1 Site-perimeter

## 6.1.1 What is it

The study of polygons has focussed primarily on their enumeration according to their most basic geometric properties — area (number of cells) and perimeter (number of bonds). Indeed a large number of models have been enumerated according to both of these parameters simultaneously; using a column-by-column construction a number of "natural" families of column-convex polygons can be enumerated according to their area and perimeter (see chapter 5, [23] and references therein), while a number of other models have been enumerated according to area alone (see chapter 4) or perimeter alone [27].

By contrast, there are very few results concerning the *site-perimeter* of polyominoes. Site-perimeter is of considerable interest to physicists since it plays an important role in the study of percolation models (see [148, 79] and references therein); as we noted in chapter 1, the probability that the origin belongs to a given (site) percolation cluster is a function of both the area and site-perimeter of the cluster. Before we go any further, let us reiterate the definition of site-perimeter.

**Definition 6.1.**
The *site-perimeter* of a polyomino is the number of nearest-neighbour vacant cells (see figure 6.1).

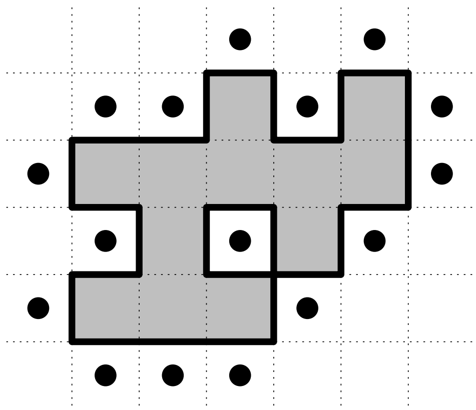


Figure 6.1: An example of a polyomino with an area of 12, a horizontal perimeter of 14, a vertical perimeter of 12 and a site-perimeter of 16.

The area, perimeter and site-perimeter generating function is known [52, 56] for Ferrers diagrams, stack, staircase and directed convex polygons, as well as the the perimeter and site-perimeter generating function of fully convex polygons [122] — all of which are families of fully convex polygons (*i.e.* row and column convex). For any convex family of polygons, the site-perimeter is simply the perimeter minus the number of *inside* corners (see figure 6.2). Using a column-by-column construction, it is relatively easy to extend perimeter and area solutions of convex families of polygons to include the number of inside corners; a simple substitution then gives the site-perimeter generating function. Unfortunately, this approach breaks down for non-convex polygons (we will explain why in section 6.1.3), and no family of non-convex polygons have been enumerated according to their site-perimeter[1].

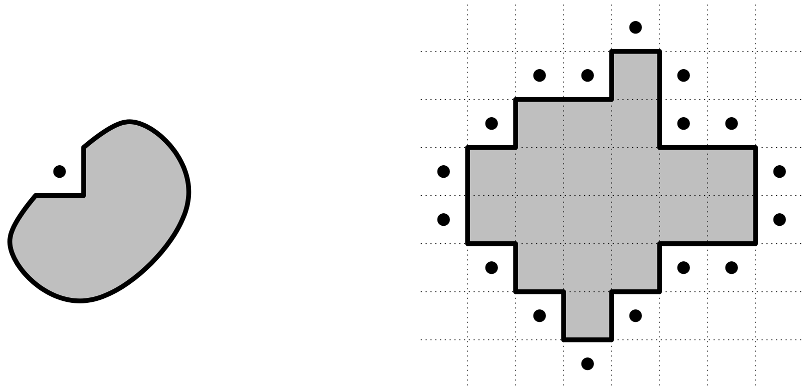


Figure 6.2: The site-perimeter of a convex polygon is equal to the perimeter minus the number of *inside* corners. The polygon on the right has 7 inside corners, a perimeter of 24 and a site-perimeter of 17.

The purpose of this chapter is to extend site-perimeter results to the simplest family of non-convex polygons — *bargraphs*.

### 6.1.2 A few definitions

Before we start let us make a few definitions that will simplify the following work.

**Definition 6.2.** •A ***bargraph polygon*** is a column-convex polygon, such that its lower edge lies on the horizontal axis (and so it is uniquely defined by the heights of its columns).

•A bargraph is *descending* if the height of its rightmost column is strictly less than the height of its second rightmost column. If a bargraph is not descending then it is *ascending* (we consider a single column to be ascending).

•A bargraph is *uniformly ascending* if the height of each column is not lower than those to its left. Similarly a bargraph is *uniformly descending* if the height of each column

[1] Directed column-convex polygons have been enumerated according to their *directed site-perimeter* — being the number of nearest-neighbour vacant cells that lie to the *north* or *east* of cells of the polyomino [49, 62]. Directed site-perimeter plays the same role in *directed percolation* as site-perimeter plays in percolation.

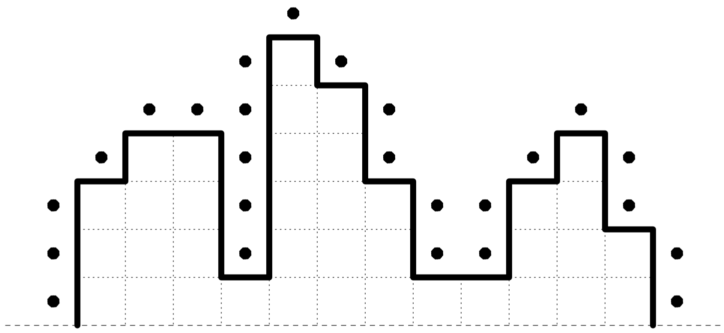


Figure 6.3: A bargraph polygon with site-perimeter indicated. It is usual to consider bargraphs to lie above the horizontal axis, and hence one does not enumerate any bonds that lie horizontally along the axis, nor any site-perimeter that lies below the axis.

is lower than those to its left.

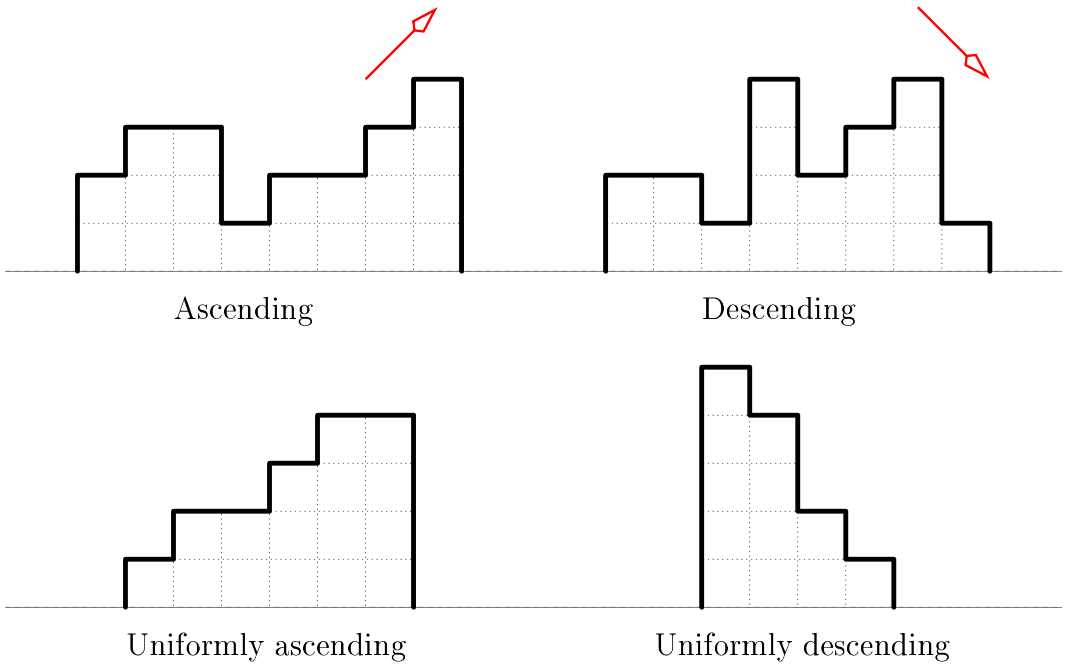



Figure 6.4: Each bargraph is either an ascending bargraph or a descending bargraph.

In the work that follows we will consider generating functions which enumerate a set of polygons, $\mathcal{P}$, according to their left heights and right heights ($t$ and $s$ respectively), area ($q$), number of columns ($x$), vertical half-perimeter ($y$) and site-perimeter ($p$). We use $\omega(P)$ to denote the contribution of a polygon $P \in \mathcal{P}$ to the generating function, *i.e.*

$$\omega(P) = t^{\text{l.h}(P)} s^{\text{r.h}(P)} q^{\text{area}(P)} x^{\text{cols}(P)} y^{\text{v.h.p}(P)} p^{\text{site.p}(P)} \tag{6.1}$$

Since we consider bargraph polygons to lie on the horizontal axis, we consider the site-perimeter of a bargraph to be the number of nearest-neighbour vacant cells *above* the horizontal axis, and similarly we enumerate the number of columns in the bargraph, instead of the horizontal perimeter (see figure 6.3). We can recover full horizontal perimeter and the full site-perimeter[2] by making the substitution,

$$x \longrightarrow x^2 p$$

in the bargraph generating function.

With these definitions out of the way let us examine why it is harder to enumerate polygons according to their site-perimeter than their ordinary perimeter.

### 6.1.3 Why is the site-perimeter hard?

In this subsection we will examine two different methods for the enumeration of bargraphs according to their perimeter and area, and explain why they are difficult to extend to site-perimeter. These methods are:

- a "*wasp-waist*'" factorisation which consists of splitting a bargraph into two smaller bargraphs at an especially thin point, and
- a *column-by-column* construction, which is sometimes referred to as the Temperley method (see chapter 5).

Both of these methods run into difficulties when we try to extend them to include site-perimeter; in both cases it is the same "geometry"[3] that causes the difficulties — a configuration that does not occur in convex polygons. While it is not obvious how the wasp-waist method can be made to work at all, we are able to find functional equations for the generating function by applying a column-by-column construction; these functional equations are considerably more complicated than those we described in chapter 5.

#### Wasp-waist factorisation

The idea of the wasp-waist method is that any bargraph can be split into one or two smaller bargraphs at a point at which it is very thin (like the waist of a wasp). See figure 6.5. Take a bargraph and find the leftmost column of height 1, then the bargraph is:

1. a bargraph whose bottom row has been duplicated (if there is no column of height 1), or
2. a single cell, or

[2] *i.e.* sites above and below the horizontal axis.
[3] We will make this precise below.

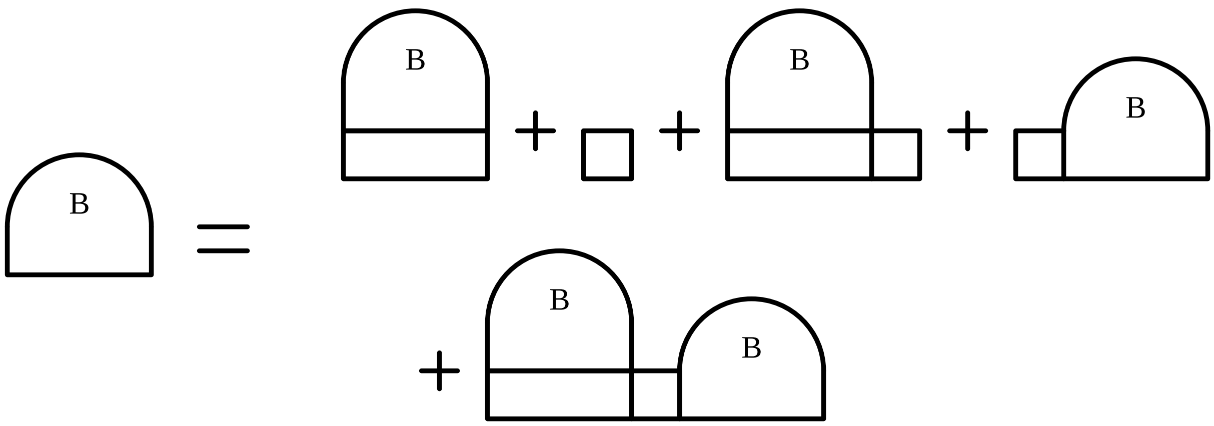



Figure 6.5: Factorisation of bargraphs.

3. a bargraph whose bottom row has been duplicated, with a single cell attached to its right, or

4. a single cell attached to a bargraph, or

5. a bargraph whose bottom row has been duplicated, connected to a single cell, and then connected to another bargraph.

This factorisation directly translates into an algebraic equation; if we define $B(x,y)$ to be the generating function of bargraphs enumerated by the number of columns, and vertical half-perimeter, then $B(x,y)$ satisfies

$$B(x,y) = xy + (x+y+xy)B(x,y) + xB(x,y)^2. \tag{6.2}$$

This can be readily solved to give the perimeter generating function:

$$B(x,y) = \frac{1}{2x}\left(1-x-y-xy-\sqrt{(1-y)((1-x)^2-y(1+x)^2)}\right) \tag{6.3}$$

This factorisation works nicely for perimeter, and can be extended to give a non-linear functional equation for the perimeter and area generating function [128, 133]. Let us try to include the site-perimeter in the 5 parts of this factorisation:

1. duplicating the bottom row of a bargraph gives $yp^2B(x,y,p)$,

2. a single cell has generating function $xyp^3$,

3. duplicating the bottom row and adding a cell gives $xyp^2B(x,y,p)$.

4. *not obvious!*, and

5. *not obvious!*

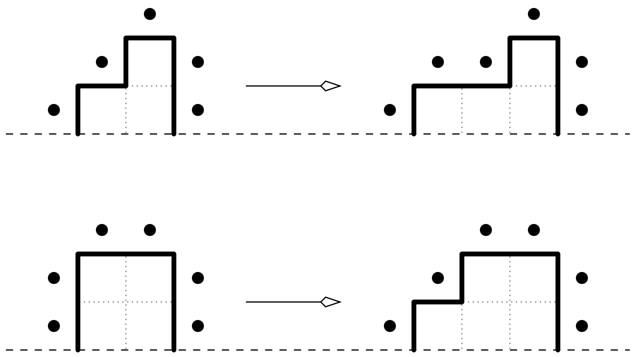


Figure 6.6: Adding a single cell to the left of a bargraph does not always increase the site-perimeter — if the leftmost column is a single cell, then the site-perimeter increases by 1, otherwise it does not change.

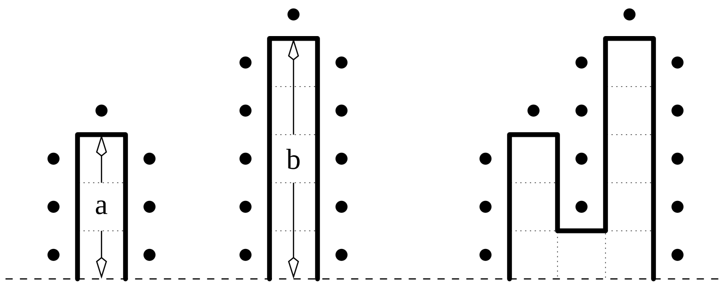



Figure 6.7: The problematic part of the factorisation. The site-perimeter of the final bargraph is not a linear combination of the site-perimeters of the components.

The forth and fifth parts of the factorisation are problematic. Gluing a single cell to the left of a bargraph (see figure 6.6) does not always change the site-perimeter; if the leftmost column of the bargraph consists of a single column, then the site-perimeter will increase by 1, otherwise the site-perimeter does not change. Consider also gluing together two columns, one of height $a$, the other of height $b$, according to the fifth part of the above factorisation (see figure 6.7). The generating function of the first column is $xy^a p^{2a+1}$, the generating function of the second is $xy^b p^{2b+1}$. When these are glued together their generating function is $x^3 y^{a+b-1} p^{1+a+b+\max(a,b)}$. *i.e.* the site perimeter of the product is not a *linear combination* of the heights (or the perimeters) of the components. Consequently we cannot make simple substitutions in the generating functions to obtain the correct site-perimeter, and it is not obvious how the wasp-waist method can be made to work.

### Column-by-column

Consider the construction of a three-column bargraph using the Temperley method. Let the heights of the columns be $a$, $b$, and $c$ (from left to right). In the description of the column-by-column construction in the previous chapter, we made extensive use of the fact that the weight function, $\omega$, factored in the following way:

$$\omega(a,b,c) = \omega(a)\tau(a,b)\tau(b,c). \tag{6.4}$$

For bargraphs enumerated according to their area, number of columns, and vertical half-perimeter, it is easy to verify that this factorisation holds, and that $\tau$ is given by

$$\tau(a,b) = \begin{cases} xq^b y^{b-a} & \text{if } b \geq a, \\ xq^b & \text{otherwise} \end{cases} \tag{6.5}$$

A weight function that factors in this way leads to a recurrence that could be expressed as a functional equation in a single "*active*" variable, $s$, that enumerated the height of the rightmost column. Unfortunately, when we include site-perimeter this factorisation no longer holds. Consider the two bargraphs in figure 6.8. The three column bargraph can be

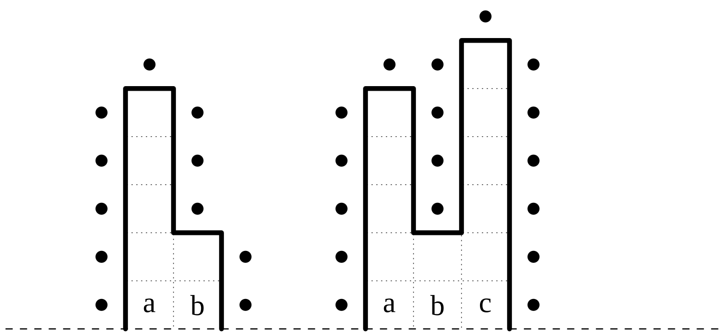



Figure 6.8: Building a three column bargraph from a two column bargraph.

constructed from the two column bargraph by attaching a new column of height $c$.

$$\begin{aligned} \omega(a,b) &= x^2 q^{a+b} y^a p^{2a+1}, & (6.6)\\ \omega(a,b,c) &= x^3 q^{a+b+c} y^{a+c-b} p^{2+a+c+\max(a,c)-b}, \text{ and so} & (6.7)\\ \omega(a,b,c)/\omega(a,b) &= xq^c y^{c-b} p^{1+c+\max(a,c)-a-b} \neq \tau(b,c). & (6.8)\end{aligned}$$

So when we add the new column, the change in weight, $\tau$, is no longer a function of the heights of the *two* rightmost columns, rather it is a function of the *three* rightmost columns. This will always occur when an *ascending* bargraph is constructed by gluing a new column onto a *descending* bargraph.

Since we have to "remember" the heights of two columns the recurrence satisfied by bargraphs will become a functional equation in *two* active variables, $s$ and $r$, where $s$ enumerates the height of the rightmost column, and $r$ enumerates the height of the second rightmost column — the recurrence can be translated into a system of *two* coupled functional equations in *two* active variables; one functional equation describing the column-by-column construction of ascending bargraphs, and the other describing the column-by-column construction of descending bargraphs. However it is possible to reduce this system to a single equation satisfied by ascending bargraphs in a single active variable, $s$. The bargraph generating function can then be expressed in terms of the ascending bargraph solution.

The functional equation for the ascending bargraph generating function does not represent a *column-by-column* construction, rather it corresponds to constructing ascending bargraphs by gluing blocks of *several* columns (any finite number in fact!) at a time. We can derive this functional equation more elegantly[4] using the Hadamard-Temperley method described in the previous chapter. The functional equation can then be solved using a combination of iteration and the kernel method.

### 6.1.4 Wells

We have seen above that the methods used to enumerate bargraphs by perimeter and area both run into problems because of "well" configurations such as those in figure 6.8 (where the height of a column is strictly lower than its immediate neighbours). The well configurations force us to consider the height of the two rightmost columns if we use a column-by-column construction. And since the site-perimeter of a well is not a linear function of the column-heights or perimeter, it is not obvious how we can alter the wasp-waist factorisation so as to be able to avoid the problems caused by such configurations. To show how the Hadamard-Temperley may avoid the problems associated with such configurations, let us consider the construction of bargraphs in which every second column is a well configuration (see figure 6.9). We call these bargraphs "*alternating bargraphs*".

**Definition 6.3.**
A bargraph is *alternating* if it has an odd number of columns, and if the heights of its columns, $c_i$, satisfy the inequality, $c_{2i-1} > c_{2i} < c_{2i+1}$, for $i \geq 1$. Every alternating bargraph is an ascending bargraph.

Like the well configurations in ordinary bargraphs, those in alternating bargraphs pose problems for wasp-waist factorisations and column-by-column constructions. The problem of alternating bargraphs is instructive, however, since it is easier to see how one can apply the block-by-block Hadamard-Temperley method (see figure 6.9); every alternating bargraph can be decomposed into a single column (the seed block), and a sequence of three column alternating bargraphs (the building blocks).

In this way the problematic well configurations are "hidden" inside the building blocks where we do not have to worry about them, and we only have to consider the height of the last column of the bargraph (rather than the last two). The generating functions of these blocks are quite simple and so it is quite simple to find the functional equation satisfied

[4] *i.e.* arriving directly at the single equation without having to first derive and then reduce the system of two equations.

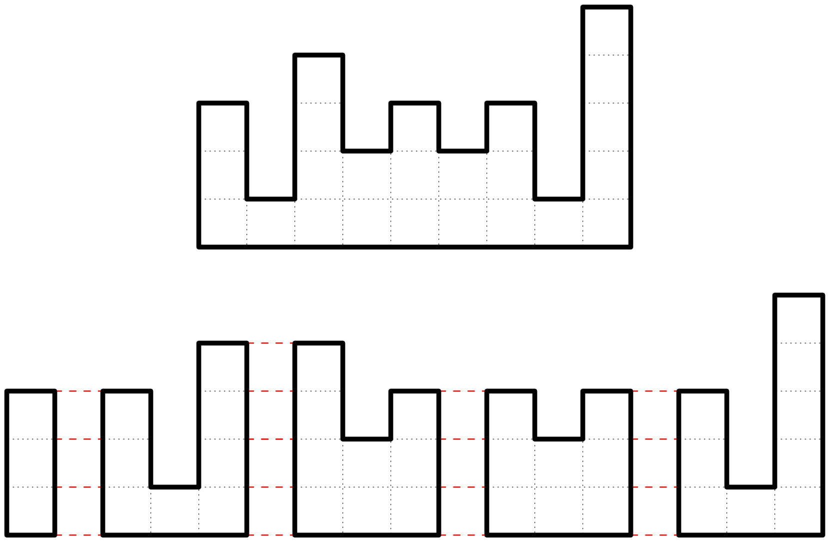


Figure 6.9: One can decompose an alternating bargraph so that the seed block is a single column, and each building block is a three column alternating bargraph.

by the generating function. In section 6.2 we shall describe in detail how this functional equation can be found and solved.

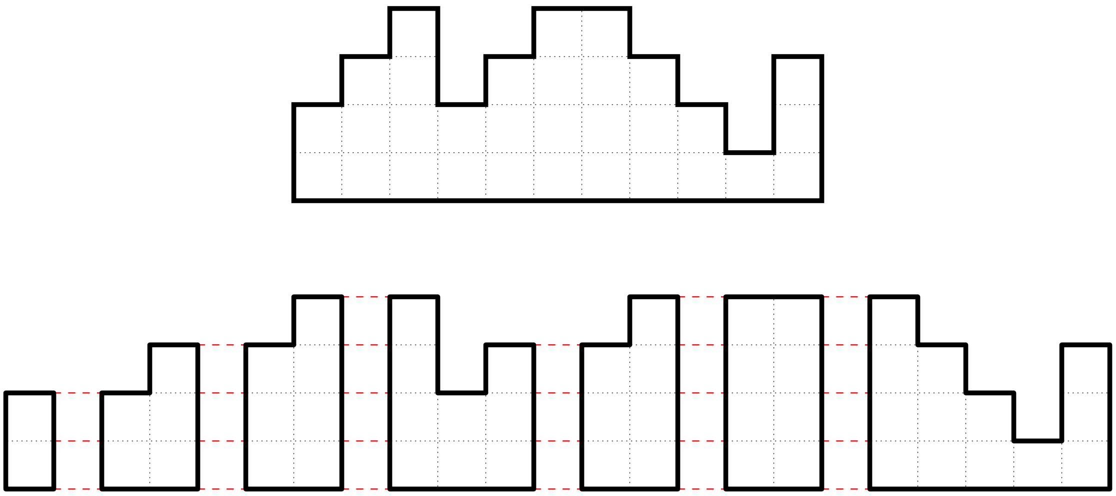


Figure 6.10: Ascending bargraphs can be decomposed so that the building blocks are either two-column ascending bargraphs, and bargraphs that descend into wells.

One can extend the alternating bargraph decomposition to ascending bargraphs. The building blocks are two-column ascending bargraphs, and strictly descending bargraphs that end in wells[5] (see figure 6.10). In section 6.3, we shall show the details of this decomposition,

[5] Strictly speaking, the building blocks are two-column ascending bargraphs and strictly descending bar-

and how it leads to the generating function for all bargraphs.

In section 6.4 we will show how bargraphs enumerated according to their site-perimeter are related to a model of polymer collapse and adsorption — interacting partially directed walks above an adsorbing wall.

# 6.2 Alternating bargraphs

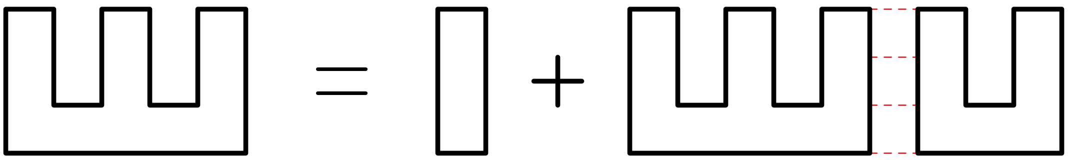


Figure 6.11: Building alternating bargraphs

Alternating bargraphs can be enumerated using the Hadamard Temperley method described in the previous chapter. See figure 6.11. Every alternating bargraph is either:

- a single column, or
- an alternating bargraph to which an underweighted three-column alternating bargraph is glued.

The seed blocks are all single columns, and the building blocks are all three column alternating bargraphs. Let $f(s; x, y, p)$ be the generating function of alternating bargraphs, $S(s; x, y, p)$ be the generating function of the seed blocks, and $T(t, s; x, y, p)$ be the generating function of appropriately underweighted building blocks. Using the Hadamard-Temperley method described in the previous chapter, $f$ satisfies the following Hadamard equation

$$f(s; x, y, p) = S(s; x, y, p) + f(t; x, y, p) \odot_t T(t, s; x, y, p). \tag{6.9}$$

This equation can then be translated into a functional equation, which we can solve.

## 6.2.1 Building blocks and functional equations

In order to apply the Hadamard Temperley method we must find the generating functions of the seed blocks and building blocks. The seed blocks for alternating bargraphs are simply all 1 column bargraphs; their generating function is

$$S(s; x, y, p) = \frac{xsyp^3}{1 - syp^2}. \tag{6.10}$$

graphs that end in either a well or two columns of the same height.

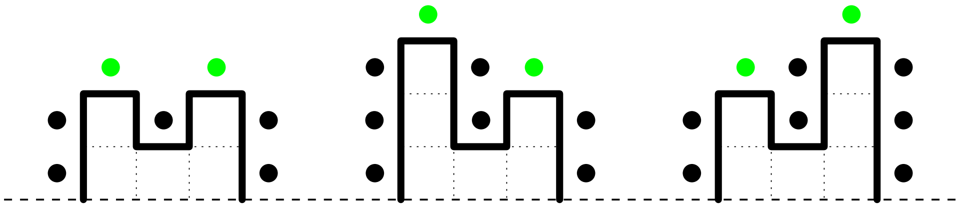


Figure 6.12: The row-minimal building blocks for alternating bargraphs. Dark spots indicate site-perimeter cells that are duplicated, while the lighter spots are not duplicated.

The building blocks are all three column alternating bargraphs. We find their generating functions by expanding the section minimal blocks (see figure 6.12):

$$\begin{aligned} B(t,s;x,y,p) &= p^2x^3[\![\, syp^2t \,]\!][\![\, sy^2p^3t \,]\!] + \\ & p^2x^3[\![\, syp^2t \,]\!][\![\, sy^2p^3t \,]\!][\![\, typ^2 \,]\!] + \\ & p^2x^3[\![\, syp^2t \,]\!][\![\, sy^2p^3t \,]\!][\![\, syp^2 \,]\!]. \end{aligned} \tag{6.11}$$

When a building-block is attached to the end of an alternating bargraph of right height $h$, the site-perimeter is over-counted by $2h+1$, the vertical half-perimeter by $h$ and the number of columns by 1. Underweighing the building blocks gives the transition function

$$\begin{aligned} T(t,s;x,y,p) &= \frac{1}{xp}B(t/yp^2,s;x,y,p) \\ &= \frac{x^2p^2s^2t^2y(1-stp^2y)}{(1-st)(1-stpy)(1-t)(1-sp^2y)} \\ &= \frac{ysp^3x^2}{1-sp^2y} + \frac{x^2yp^2s^2}{(1-spy)(1-s)(1-t)} - \frac{ysp^2x^2(p-1)}{(1-sp^2y)(1-py)(1-spy)(1-stpy)} \\ & + \frac{(p^2y-1)p^2yx^2s}{(1-sp^2y)(1-py)(1-s)(1-st)}. \end{aligned} \tag{6.12}$$

Applying lemma 5.2, we obtain the functional equation satisfied by the alternating bargraph generating function:

$$\begin{aligned} f(s) &= \frac{xysp^3}{1-syp^2} + f(t) \odot_t T(t,s) \\ &= \frac{xsqp^2}{1-sqp} + \frac{x^2s^2pq}{(1-sq)(1-s)}f(1) + \frac{x^2spq(1-p)}{(1-spq)(1-q)(1-sq)}f(sq) \\ & + \frac{x^2spq(pq-1)}{(1-spq)(1-q)(1-s)}f(s), \end{aligned} \tag{6.13}$$

where we have written $q = py$. We will write this equation as

$$f(s) = a(s) + b(s)f(1) + c(s)f(sq) + d(s)f(s), \tag{6.14}$$

with appropriate definitions of $a(s)$, $b(s)$, $c(s)$ and $d(s)$.

**Remark.**
If we enumerate according area (using the variable $z$) as well as the perimeter and site perimeter, then we find an equation of the form

$$f(s) = \tilde{a}(s) + \tilde{b}(s) f(1) + \tilde{c}(s) f(sq) + \tilde{d}(s) f(sz^2). \tag{6.15}$$

It is not obvious how the kernel method could be used to solve this equation. Nor is it straight-forward to apply the iteration method since we will have to contend with an infinite number of unknowns of the form $f(sq^j z^{2k})$.

### 6.2.2 Solving the functional equation

We can find both the perimeter generating function and the perimeter and site-perimeter generating function using the techniques described in the previous chapter.

**$p = 1$, and the kernel method.**

The first step of applying the kernel method to functional equations such as equation (6.14) is to remove the unknown $f(sq)$. In the previous chapter we did this by setting $q = 1$, however equation (6.13) is singular at $q = 1$. We note that the coefficient of $f(sq)$ is zero when $p = 1$, and so we can remove this term by setting $p = 1$ (and so $q \equiv y$), and then applying the kernel method.

When $p = 1$, equation (6.13) reduces to

$$f(s) = \frac{xsy}{1 - sy} + \frac{x^2 s^2 y}{(1 - sy)(1 - s)} f(1) - \frac{yx^2 s}{(1 - sy)(1 - s)} f(s). \tag{6.16}$$

Collect all the $f(s)$ terms on the left-hand side:

$$\frac{ys^2 - (1 + y - x^2 y)s + 1}{(1 - s)(1 - sy)} f(s) = \frac{xsy}{1 - sy} + \frac{x^2 s^2 y}{(1 - sy)(1 - s)} f(1). \tag{6.17}$$

Solving $ys^2 - (1 + y - x^2 y)s + 1 = 0$ with respect to $s$ gives two solutions:

$$s_{\pm} = \frac{1 + y - x^2 y \pm \sqrt{(1 - y - 2xy - x^2 y)(1 - y + 2xy - x^2 y)}}{2y}. \tag{6.18}$$

Only one of these solutions, $s_-$ is convergent as $x, y \to 0$. Substituting $s = s_-$ removes $f(s)$ from the functional equation and gives the following solution for $f(1)$:

$$f(1) = \frac{1 - y + x^2 y - \sqrt{(1 - y - 2xy - x^2 y)(1 - y + 2xy - x^2 y)}}{2x} \tag{6.19}$$

This can then be substituted back into equation (6.16) to obtain $f(s)$.

### Full solution by iteration

We have found the perimeter generating function by using the kernel method, however to find the perimeter and site-perimeter generating function we must use the iteration method. Before we can iterate equation (6.13), we must massage it into an equation of the form

$$f(s) = \alpha(s) + \beta(s)f(1) + \gamma(s)f(sq). \tag{6.20}$$

We can do this by moving all the $f(s)$ terms to the left hand side, and then dividing through by $1 - d(s)$:

$$\begin{aligned} f(s)\Big(1 - d(s)\Big) &= a(s) + b(s)f(1) + c(s)f(sq) \\ f(s) &= \alpha(s) + \beta(s)f(1) + \gamma(s)f(sq), \end{aligned} \tag{6.21}$$

where

$$\alpha(s) = \frac{a(s)}{1-d(s)} = \frac{xsp^2q(1-q)(1-s)}{\eta(s)} \tag{6.22}$$

$$\beta(s) = \frac{b(s)}{1-d(s)} = \frac{x^2s^2pq(1-spq)(1-q)}{(1-sq)\eta(s)} \tag{6.23}$$

$$\gamma(s) = \frac{c(s)}{1-d(s)} = \frac{x^2spq(1-p)(1-s)}{(1-sq)\eta(s)}, \tag{6.24}$$

with

$$\eta(s) = (1-spq)(1-q)(1-s) - x^2spq(pq-1). \tag{6.25}$$

We note that $\alpha(1) = \gamma(1) = 0$ and $\beta(1) = 1$.

Applying the iteration method to equation (6.21) gives

$$\begin{aligned} f(s) = &\left(\sum_{n\geq 0} \gamma(s)\gamma(sq)\cdots\gamma(sq^{n-1})\alpha(sq^n)\right) \\ &+ \left(\sum_{n\geq 0} \gamma(s)\gamma(sq)\cdots\gamma(sq^{n-1})\beta(sq^n)\right) f(1). \end{aligned} \tag{6.26}$$

We must check that these sums converge in the space of formal power series. In particular we must check that the following product converges to the zero function:

$$\lim_{N\to\infty} \prod_{j=0}^{N} \gamma(sq^j) = 0. \tag{6.27}$$

As a formal power series in $x$, the first non-zero coefficient of $\gamma(s)$ is that of $x^2$. Consequently the first non-zero coefficient of $\prod_{j=0}^{N-1}\gamma(sq^j)$ is that of $x^{2N}$, and so the product converges to $\sum 0x^n$.

In the previous chapter, the iteration method removed the unknown $f(sq)$, and left the equation

$$f(s) = \mathfrak{A}(s) + \mathfrak{B}(s)f(1). \tag{6.28}$$

We were able to solve this by setting $s = 1$ and isolating $f(1)$:

$$f(1) = \frac{\mathfrak{A}(1)}{1 - \mathfrak{B}(1)}. \tag{6.29}$$

Substituting this expression for $f(1)$ back into the original equation gave $f(s)$:

$$f(s) = \mathfrak{A}(s) + \frac{\mathfrak{A}(1)\mathfrak{B}(s)}{1 - \mathfrak{B}(1)}. \tag{6.30}$$

Unfortunately things are not quite as simple here. Since $\alpha(1) = \gamma(1) = 0$, and $\beta(1) = 1$, substituting $s = 1$ into equation (6.26) gives the rather useless tautology:

$$f(1) = f(1). \tag{6.31}$$

Though this equation is reassuring, we need to find some other way of removing the $f(s)$ term. We have only used three techniques to remove an unknown from a functional equation; specialisation (*i.e.* setting $q = 1$ or $s = 1$), iteration and the kernel method. We have just seen that specialisation does not work, and it is not clear how we could apply iteration. This leaves the kernel method.

We see[6] that the right-hand side of equation (6.26) has a common denominator factor of $1 - d(s)$. Multiplying both sides by this factor, we find

$$\begin{aligned} \Big(1 - d(s)\Big) f(s) &= a(s) + c(s)\left(\sum_{n\geq 1} \gamma(sq)\gamma(sq^2)\cdots\gamma(sq^{n-1})\alpha(sq^n)\right) + \\ &\quad \left(b(s) + c(s)\left(\sum_{n\geq 1} \gamma(sq)\gamma(sq^2)\cdots\gamma(sq^{n-1})\beta(sq^n)\right)\right) f(1). \end{aligned} \tag{6.32}$$

We then apply the kernel method by finding $s$ such that $1 - d(s) = 0$; this is equivalent to finding the zeros of the equation

$$0 = (1 - spq)(1 - q)(1 - s) - x^2 spq(pq - 1) \tag{6.33}$$

with respect to $s$. Since this equation is quadratic in $s$ there are two solutions[7], however only one of these is convergent as $x, y, p \to 0$, and we will denote this $s_1$, and the other $s_2$.

Setting $s = s_1$ in equation (6.32), makes the left-hand side zero, and we can solve for $f(1)$:

$$f(1) = -\frac{a(s_1) + c(s_1)\left(\sum_{n\geq 1} \gamma(s_1 q)\gamma(s_1 q^2)\cdots\gamma(s_1 q^{n-1})\alpha(s_1 q^n)\right)}{b(s_1) + c(s_1)\left(\sum_{n\geq 1} \gamma(s_1 q)\gamma(s_1 q^2)\cdots\gamma(s_1 q^{n-1})\beta(s_1 q^n)\right)}. \tag{6.34}$$

[6] after several hours of staring! (well... days may be more accurate)!

[7] No I'm not going to state them here since they are both messy.

Again, as above, we have to be careful of the convergence of these sums, in particular we require that $\lim_{N\to\infty}\prod_{j=0}^{N}\gamma(s_1q^j)=0$. If we consider $\gamma(s_1q^j)$ to be a formal power series in $x$, we find that the coefficients of $x^0$ and $x^1$ are zero, and the product must converge to zero.

◁ ◁ ⋄ ▷ ▷

The solution of $f(1)$ can be massaged into a nicer form. We start by noting that $\eta(s)$ can be rewritten in terms of $s_1$ and $s_2$:

$$\eta(s) = (1-q)(1-s/s_1)(1-s/s_2), \tag{6.35}$$

and so

$$\eta(s_1q^n) = (1-q)(1-q^n)(1-\sigma q^n), \tag{6.36}$$

where $\sigma = s_1/s_2$ (we note that $\sigma$ can be expressed in terms of $s_1$ alone). Using this we can simplify $\alpha$, $\beta$ and $\gamma$:

$$\begin{aligned}
\alpha(s_1q^n) &= \frac{xs_1p^2q^{n+1}(1-s_1q^n)}{(1-q^n)(1-\sigma q^n)}, && (6.37)\\
\beta(s_1q^n) &= \frac{x^2s_1^2pq^{2n+1}(1-s_1pq^{n+1})}{(1-s_1q^{n+1})(1-q^n)(1-\sigma q^n)}, && (6.38)\\
\gamma(s_1q^n) &= \frac{x^2s_1pq^{n+1}(1-p)(1-s_1q^n)}{(1-q)(1-s_1q^{n+1})(1-q^n)(1-\sigma q^n)}. && (6.39)
\end{aligned}$$

Let us now simplify the components of the sums in the expression for the generating function:

$$\gamma(s_1q)\gamma(s_1q^2)\cdots\gamma(s_1q^n) = \frac{(1-s_1q)x^{2n}s_1^np^nq^{\binom{n+2}{2}}(1-p)^n}{q(1-s_1q^{n+1})(1-q)^n(q)_n(\sigma q)_n}, \tag{6.40}$$

and so

$$c(s_1)\gamma(s_1q)\gamma(s_1q^2)\cdots\gamma(s_1q^n) = \frac{x^{2n+2}s_1^{n+1}p^{n+1}q^{\binom{n+2}{2}}(1-p)^{n+1}}{(1-s_1pq)(1-s_1q^{n+1})(1-q)^{n+1}(q)_n(\sigma q)_n}. \tag{6.41}$$

Multiplying by $\alpha$ and $\beta$ gives

$$\begin{aligned}
c(s_1)\gamma(s_1q)\gamma(s_1q^2)\cdots\gamma(s_1q^{n-1})\alpha(s_1q^n) &= \frac{x^{2n+1}s_1^{n+1}p^{n+2}q^{\binom{n+2}{2}}(1-p)^n}{(1-s_1pq)(1-q)^n(q)_n(\sigma q)_n} && (6.42)\\
c(s_1)\gamma(s_1q)\gamma(s_1q^2)\cdots\gamma(s_1q^{n-1})\beta(s_1q^n) &= && (6.43)\\
&\frac{x^{2n+2}s_1^{n+2}p^{n+1}q^{\binom{n+3}{2}}(1-p)^n(1-s_1pq^{n+1})}{q^2(1-s_1pq)(1-s_1q^n)(1-s_1q^{n+1})(1-q)^n(q)_n(\sigma q)_n}.
\end{aligned}$$

If we set $n=0$ in equations (6.42) and (6.43) we obtain the expressions for $a(s_1)$ and $b(s_1)$ respectively, and so we can write down the generating function:

$$f(1) = -\frac{pq^2\sum_{n\geq 0}\frac{x^{2n}s_1^np^nq^{\binom{n+2}{2}}(1-p)^n}{(1-q)^n(q)_n(\sigma q)_n}}{xs_1\sum_{n\geq 0}\frac{x^{2n}s_1^np^nq^{\binom{n+3}{2}}(1-p)^n(1-s_1pq^{n+1})}{(1-s_1q^n)(1-s_1q^{n+1})(1-q)^n(q)_n(\sigma q)_n}}. \tag{6.44}$$

This type of solution has not been seen before in polygon enumeration problems. The area generating functions of many column-convex families of polygons (such as staircase polygons) involve $q$-series, being sums of products of *rational* functions. Here we find the solution is a sum of products of *algebraic* functions[8]. Needless to say, the form of this generating function makes it difficult to analyse. We are currently investigating the asymptotics of this expression; preliminary analysis and numerical work indicates that the dominant singularity for all values of $p$, $y$ and $x$ is the square root singularity in the expression of $s_1$, which indicates that the model does not undergo a phase-transition. We will discuss this further in section 6.4.

# 6.3

# Bargraphs

In this section we enumerate all bargraphs according to their perimeter and site-perimeter. We do not do this by a direct construction on all bargraphs, rather we show how the generating function for all bargraphs can be obtained from the generating function of ascending bargraphs by appropriate substitutions. We then give a Hadamard-Temperley construction for ascending bargraphs. The building blocks for this construction are more complicated than those of the alternating bargraph construction, and are no longer required to have width 3 (or even a bounded width). This construction leads to a functional equation that is very similar to that of alternating bargraphs and it can be solved in a similar way.

In the work that follows, let $P(s;x,y,p)$, $R(s;x,y,p)$ and $L(s;x,y,p)$ be the generating function of all bargraphs, ascending bargraphs and descending bargraphs respectively.

## 6.3.1 From ascending bargraphs to all bargraphs.

Consider the descending bargraph in figure 6.13. It can be uniquely decomposed into an ascending bargraph and a *uniformly* descending bargraph (suitably underweighted). The ascending bargraph, on the other hand, can be considered to be itself composed with a single (underweighted) column[9].

This decomposition holds for all bargraphs and we can use the Hadamard-Temperley method to translate it into a functional equation. If $T_d(t,s;x,y,p)$ is the transition function associated with gluing a uniformly descending bargraph or a single column to an ascending bargraph, then

$$P(1;x,y,p) = R(t;x,y,p) \odot_t T_d(t,1;x,y,p). \tag{6.45}$$

[8] Algebraic *irrational* functions. Actually all this "algebraicness" is contained in $s_1$ and $\sigma = s_1/s_2$ (which can be rewritten in terms of $s_1$ since $s_1$ and $s_2$ are solutions of the same quadratic equation).

[9] This is essentially the statement that for a function $f(s)$, with $f(0)=0$, we always have

$$f(s) = f(t) \odot_t \frac{st}{1-st}.$$

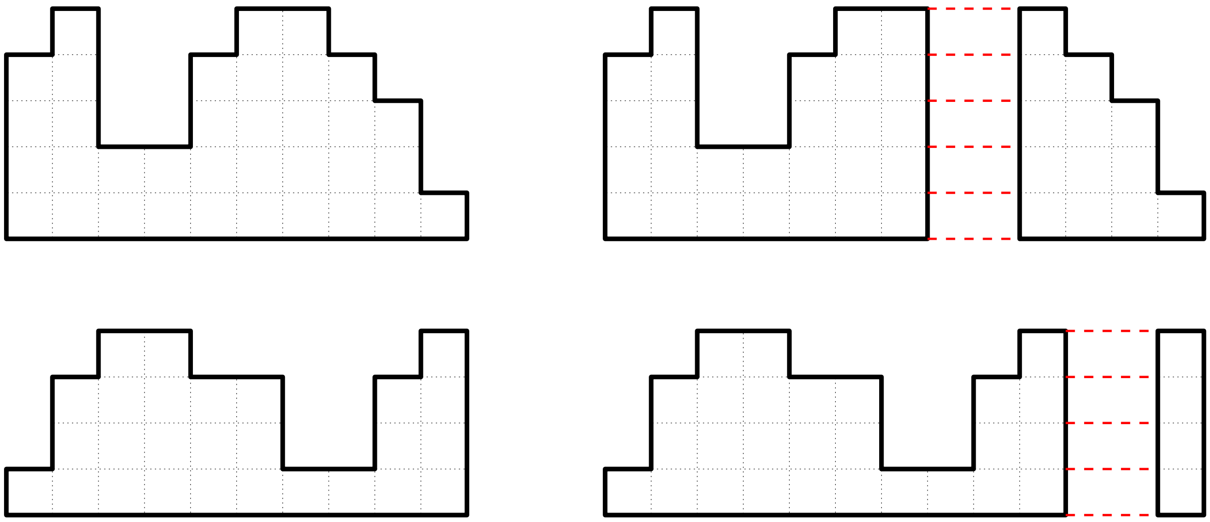


Figure 6.13: Decomposing bargraphs: a descending bargraph can be decomposed into an ascending bargraph and a uniformly descending bargraph, while an ascending bargraph can be decomposed into itself and a single column.

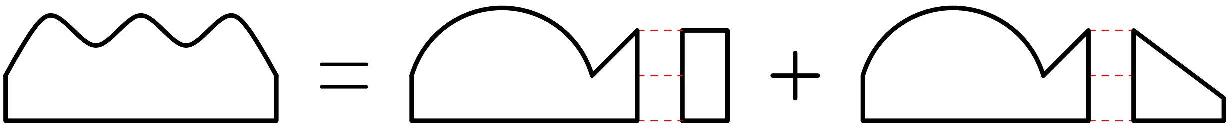


Figure 6.14: All bargraphs can be constructed from ascending bargraphs, by gluing either a single column, or a uniformly descending bargraph.

We compute $T_d$ from the generating function of the building blocks, $B_d$, which is computed by expanding the row-minimal blocks. Unlike in previous cases, these blocks do not have bounded width and hence there are an infinite number of row-minimal blocks. Let $t$ count the left height and $z$ count the right height[10]. We see from figure 6.15 that the generating function of these blocks is

$$\begin{aligned} B_d(t, z; x, y, p) &= xp[\![\, ztyp^2 \,]\!] \sum_{n \geq 0} \left( x[\![\, typ^2 \,]\!] \right)^n \\ &= \frac{xytzp^3(1 - typ^2)}{(1 - typ^2 - xtyp^2)(1 - ztyp^2)}. \end{aligned} \tag{6.46}$$

When we glue a block onto an existing bargraph of right height $h$, we over count the site-perimeter by $2h+1$, the vertical half-perimeter by $h$ and the number of columns by 1. Hence the transition function for descending blocks is $T_d(t, 1; x, y, p) = B_d(t/yp^2, 1; x, y, p)/xp$. The

[10]We use $z$ here instead of $s$ for reasons that will become apparent below.

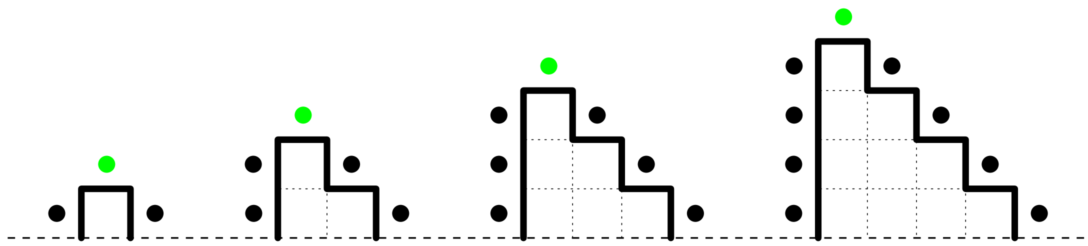


Figure 6.15: The first four section minimal descent blocks.

partial fraction expansion of $T_d$ allows us to rewrite the Hadamard product:

$$\begin{aligned} P(1;x,y,p) &= R(t;x,y,p) \odot_t \frac{t}{1-t-xt} \\ &= R(t;x,y,p) \odot_t \left( \frac{1}{(1+x)(1-t-tx)} - \frac{t^0}{1+x} \right) \\ &= \frac{1}{1+x} R(1+x;x,y,p), \end{aligned} \tag{6.47}$$

where we have made use of the fact that there are no bargraphs with a right height of zero, and so the term $R(0)$ is always equal to zero. Consequently, in order to find the generating function of *all* bargraphs we need only find an expression for $R(1+x)$. Since we can recover the generating function of all bargraphs from the generating function of ascending bargraphs, let us now consider the construction of ascending bargraphs, using the Hadamard-Temperley method.

### 6.3.2 Building blocks for ascending bargraphs

Rather than considering what blocks we need to *add* to an ascending bargraph to obtain another ascending bargraph, let us consider what happens when we *remove* the rightmost column of an ascending bargraph. Three things can occur (see figure 6.16):

- we are left with nothing,
- we are left with an ascending bargraph, or
- we are left with a descending bargraph.

The last case is a problem; rather than leaving a descending bargraph let us continue to remove columns, until the resulting bargraph is ascending. Now let us reverse this process and put back the columns we have just removed. This implies (see figures (6.16) and (6.17)) that an ascending bargraph is either

- a single column,
- an *ascent block* glued to an ascending bargraph, or

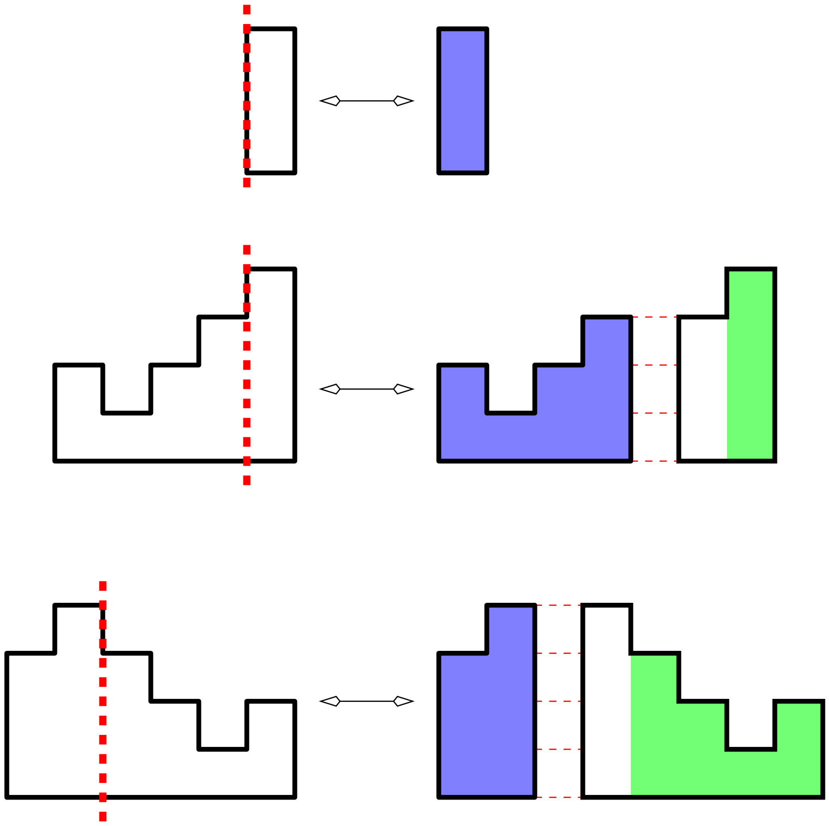


Figure 6.16: The three different cases that can occur when removing columns from an ascending bargraph, and the seed blocks and building blocks these three cases imply. Green indicates the underweighted building block and blue indicates the remaining bargraph.

- a *descent-ascent block* glued to an ascending bargraph.

The first case tells us that the seed blocks are simply all single column bargraphs; their generating function is

$$S(s;x,y,p) = \frac{xsyp^3}{1-syp^2}. \tag{6.48}$$

The second and third cases give us two different types of building blocks. The first we call *ascent blocks*; these correspond to gluing an underweighted two column ascending bargraph. The second type of block we call *descent-ascent blocks*; these correspond to gluing a *uniformly* descending bargraph followed by another column to make the bargraph ascending.

### Descent-ascent blocks

Rather than computing the generating function of descent-ascent blocks directly, we will construct them by gluing "wells" onto the end of descending blocks (since an ascending bargraph may end in two columns of the same height, we must also include this possibility). See figure 6.18.

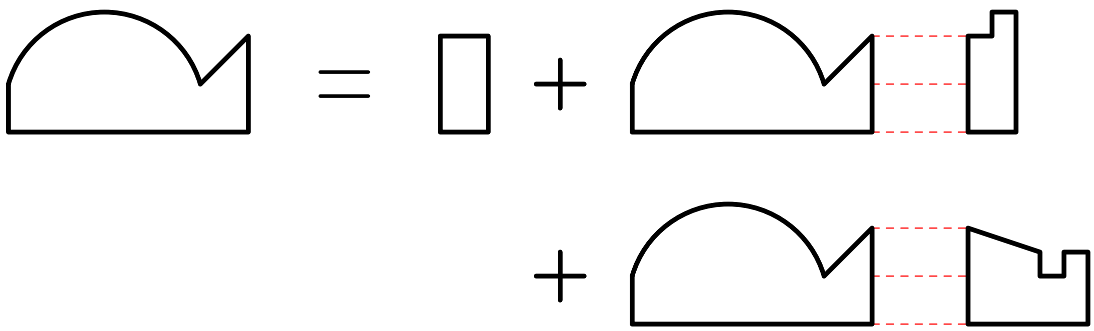


Figure 6.17: Every ascending bargraph is either a single column or can be constructed by gluing an ascending block onto an ascending bargraph, or by gluing a descent-ascent block onto an ascending bargraph.

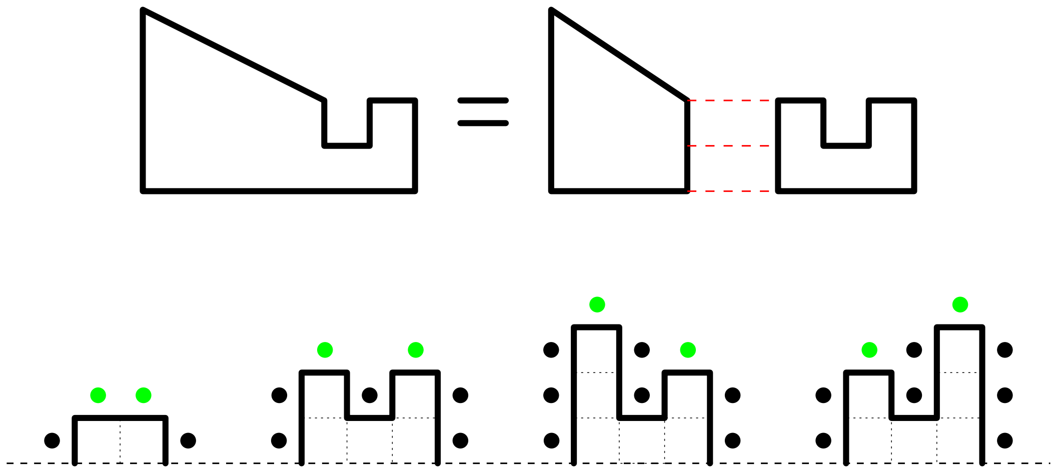


Figure 6.18: We build descent-ascent blocks, by gluing "wells" onto the end of descending blocks. The row-minimal "wells" are given.

The generating function of the wells enumerated by left height ($z$) and right height ($s$), is given by

$$B_w(z, s; x, y, p) = x^2p^2[\![ zsyp^2 ]\!] + x^3p^2[\![ zsyp^2 ]\!][\![ zsy^2p^3 ]\!]\Big(1 + [\![ syp^2 ]\!] + [\![ zyp^2 ]\!]\Big). \tag{6.49}$$

When a well is glued to a descent block of right height $h$, the site-perimeter is over counted by $2h+1$, the vertical half-perimeter by $h$ and the number of columns by 1, and so the well transition function is given by $T_w(z, s; x, y, p) = B_w(z/yp^2, s; x, y, p)/xp$. The generating function of descent-ascent blocks is then

$$\begin{aligned} B_{da}(t, s; x, y, p) &= B_d(t, z; x, y, p) \odot_z T_w(z, s; x, y, p) \\ &= \frac{x^2p^4sty \begin{pmatrix} 1 - yp^2(s+t)y + y^2tp^3s(p-1+x) \\ \qquad + y^3tp^5s(s+t) - y^4t^2p^7s^2(1+x) \end{pmatrix}}{(1-syp^2)(1-tp^3sy^2)(1-tsyp^2)(1-typ^2-xtyp^2)}. \end{aligned} \tag{6.50}$$

**Ascending blocks**

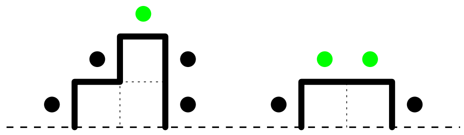


Figure 6.19: The two row-minimal ascending blocks. The block on the right has already been included in the set of descent-ascent blocks.

The ascending blocks are (by comparison) simple — they add a new column to the ascending bargraph such that the right height either increases, or does not change. Hence there are two row-minimal blocks, and they are given in figure 6.19. We must note that the block on the right (corresponding to adding a new column but not changing the right-height) has already been included in the set of descent-ascent blocks (it is obtained by the gluing of the first "well" in figure 6.18, and the first descending block in figure 6.15). Consequently the generating function of ascending blocks is simply given by the expansion of the row-minimal block on the left of figure 6.19:

$$B_a(t, s; x, y, p) = x^2 p [\![ \, tsyp^2 \, ]\!] [\![ \, syp^2 \, ]\!] = \frac{ts^2x^2p^5y^2}{(1 - syp^2)(1 - styp^2)}. \tag{6.51}$$

◁ ◁ ◇ ▷ ▷

The generating function of all building blocks is simply

$$B(t, s; x, y, p) = B_a(t, s; x, y, p) + B_{da}(t, s; x, y, p). \tag{6.52}$$

We do not state this generating function in full because we will state the transition function (which is nearly identical) in the next section. Now that we have the generating functions of the building blocks and seed blocks we can derive the functional equation.

### 6.3.3 Functional equation

When we glue a building block onto the end of an ascending bargraph of right height $h$, the site-perimeter is over counted by $2h + 1$, the vertical half-perimeter by $h$ and the number of columns by 1, so the transition function is given by

$$\begin{aligned}
T(t,s;x,y,p) &= B(t/yp^2,s;x,y,p)/xp \\
&= \frac{xp(xsyp^2+syp^2-1-yps-xpys)}{(1-syp^2)(1+x)}\left(t^0\right) \\
&+ \frac{spx^2(1+x-syp^2-xsyp^2+xpys)}{(1+x)(1+x-s)(1+x-yps)(1-syp^2)}\left(\frac{1}{1-t-tx}\right) \\
&+ \frac{ysp^2x^2(1-p)}{(1-yp)(1+x-yps)(1-syp^2)}\left(\frac{1}{1-tsyp}\right) \\
&+ \frac{xp\begin{pmatrix}(1-s)+yp(s-1)(1+sp-s)\\ \qquad +y^2sp^2(p-1)(x+1-s)\end{pmatrix}}{(1-yp)(x+1-s)(1-syp^2)}\left(\frac{1}{1-st}\right). \qquad (6.53)
\end{aligned}$$

And so (by evaluating Hadamard products) we have the following functional equation for ascending bargraphs:

$$\begin{aligned}
R(s;x,y,p) &= \frac{xsyp^3}{1-syp^2}+R(t;x,y,p)\odot_t T(t,s;x,y,p) \\
&= \frac{xsyp^3}{1-syp^2}+\frac{spx^2(1+x-syp^2-xsyp^2+xpys)}{(1+x)(1+x-s)(1+x-yps)(1-syp^2)}R(1+x) \\
&+\frac{ysp^2x^2(1-p)}{(1-yp)(1+x-yps)(1-syp^2)}R(spy) \\
&+\frac{xp\begin{pmatrix}(1-s)+yp(s-1)(1+sp-s)\\ \qquad +y^2sp^2(p-1)(x+1-s)\end{pmatrix}}{(1-yp)(1+x-s)(1-syp^2)}R(s). \qquad (6.54)
\end{aligned}$$

We make the substitution $q = py$ in the above expression (as we did for alternating bargraphs), yielding an equation of the form

$$R(s) = a(s) + b(s)R(1+x) + c(s)R(sq) + d(s)R(s), \qquad (6.55)$$

where

$$\begin{aligned}
a(s) &= \frac{xsqp^2}{1-sqp}, \qquad (6.56)\\
b(s) &= \frac{x^2sp(1+x-sqp-xsqp+xqs)}{(1-sqp)(1+x-qs)(1+x-s)(1+x)}, \qquad (6.57)\\
c(s) &= \frac{x^2sqp(1-p)}{(1-sqp)(1+x-qs)(1-q)}, \qquad (6.58)\\
d(s) &= \frac{xp\Big((1-s)+q(s-1)(1+sp-s)+q^2s(p-1)(x+1-s)\Big)}{(1-sqp)(1+x-s)(1-q)}. \qquad (6.59)
\end{aligned}$$

### 6.3.4 Solution

Here we will concentrate on the full solution by iteration, rather than the application of the kernel method to obtain the perimeter generating function. The solution to equation (6.55) (by both the kernel and iteration methods) follows exactly the same steps as the solution of alternating bargraphs. We start by juggling the equation into a form to which we can apply the iteration method:

$$R(s) = \alpha(s) + \beta(s)R(1+x) + \gamma(s)R(sq), \tag{6.60}$$

where

$$\alpha(s) = \frac{a(s)}{1-d(s)} = \frac{xsp^2q(1-q)(1+x-s)}{\eta(s)}, \tag{6.61}$$

$$\beta(s) = \frac{b(s)}{1-d(s)} = \frac{x^2ps(1-q)(1+x-sq(p+xp-x))}{(1+x)(1+x-sq)\eta(s)}, \tag{6.62}$$

$$\gamma(s) = \frac{c(s)}{1-d(s)} = \frac{x^2spq(1-p)(1+x-s)}{(1+x-sq)\eta(s)}, \tag{6.63}$$

with

$$\eta(s) = (1-sqp)(1+x-s)(1-q)+ \\ xp\Big((s-1)+q(1-s)(1-s+sp)+q^2s(1-p)(1+x-s)\Big). \tag{6.64}$$

We note here that $\alpha(1+x) = \gamma(1+x) = 0$ and $\beta(1+x) = 1$ (which is very similar to alternating bargraphs).

Iterating equation (6.60) gives

$$R(s) = \left(\sum_{n\geq 0}\gamma(s)\gamma(sq)\cdots\gamma(sq^{n-1})\alpha(sq^n)\right) \\ + \left(\sum_{n\geq 0}\gamma(s)\gamma(sq)\cdots\gamma(sq^{n-1})\beta(sq^n)\right)R(1+x), \tag{6.65}$$

and by similar reasonings to those we used for alternating bargraphs we know that these sums formally converge. We try to remove $R(s)$ by setting $s = 1+x$, and find that the equation reduces to the tautology $R(1+x) = R(1+x)$. We apply the kernel method to overcome this:

$$\Big(1-d(s)\Big)R(s) = a(s) + c(s)\left(\sum_{n\geq 1}\gamma(sq)\gamma(sq^2)\cdots\gamma(sq^{n-1})\alpha(sq^n)\right) + \\ \left(b(s) + c(s)\left(\sum_{n\geq 1}\gamma(sq)\gamma(sq^2)\cdots\gamma(sq^{n-1})\beta(sq^n)\right)\right)R(1+x). \tag{6.66}$$

We remove the $R(s)$ term by finding $s$ such that $d(s)=1$. There are two values, $s_1$ and $s_2$; we choose $s_1$ to be the solution that is convergent as $x,y,p\to 0$. Setting $s=s_1$ and isolating $R(1+x)$ leaves

$$R(1+x) = -\frac{a(s_1)+c(s_1)\left(\sum_{n\geq 1}\gamma(s_1q)\gamma(s_1q^2)\cdots\gamma(s_1q^{n-1})\alpha(s_1q^n)\right)}{b(s_1)+c(s_1)\left(\sum_{n\geq 1}\gamma(s_1q)\gamma(s_1q^2)\cdots\gamma(s_1q^{n-1})\beta(s_1q^n)\right)}. \tag{6.67}$$

We can make this solution a little more palatable by simplifying the product of $\gamma$'s in the same way that we did for alternating bargraphs. Again we note that $\eta(s)$ can be rewritten in terms of $s_1$ and $s_2$:

$$\eta(s) = (1+x-xp)(1-q)(1-s/s_1)(1-s/s_2), \tag{6.68}$$

and so

$$\eta(s_1q^n) = (1+x-xp)(1-q)(1-q^n)(1-\sigma q^n), \tag{6.69}$$

where $\sigma = s_1/s_2$ (again, we note that $\sigma$ can be expressed in terms of $s_1$ alone). Using this we simplify $\alpha$, $\beta$ and $\gamma$:

$$\begin{aligned} \alpha(s_1q^n) &= \frac{xs_1p^2q^{n+1}(1+x-s_1q^n)}{(1+x-xp)(1-q^n)(1-\sigma q^n)}, && (6.70)\\ \beta(s_1q^n) &= \frac{x^2s_1pq^n(1+x-s_1q^{n+1}(p+xp-x))}{(1+x)(1+x-xp)(1+x-s_1q^{n+1})(1-q^n)(1-\sigma q^n)}, && (6.71)\\ \gamma(s_1q^n) &= \frac{x^2s_1pq^{n+1}(1-p)(1+x-s_1q^n)}{(1+x-xp)(1-q)(1+x-s_1q^{n+1})(1-q^n)(1-\sigma q^n)}. && (6.72)\end{aligned}$$

$$\gamma(s_1q)\gamma(s_1q^2)\cdots\gamma(s_1q^n) = \frac{x^{2n}s_1^np^nq^{\binom{n+2}{2}-1}(1-p)^n(1+x-s_1q)}{(1+x-xp)^n(1-q)^n(1+x-s_1q^{n+1})(q)_n(\sigma q)_n}, \tag{6.73}$$

and so

$$c(s_1)\gamma(s_1q)\gamma(s_1q^2)\cdots\gamma(s_1q^n) = \frac{x^{2n+2}s_1^{n+1}p^{n+1}q^{\binom{n+2}{2}}(1-p)^{n+1}}{(1-s_1pq)(1+x-xp)^n(1-q)^{n+1}(1+x-s_1q^{n+1})(q)_n(\sigma q)_n}. \tag{6.74}$$

Multiplying by $\alpha$ and $\beta$ gives

$$c(s_1)\gamma(s_1q)\gamma(s_1q^2)\cdots\gamma(s_1q^{n-1})\alpha(s_1q^n) = \frac{x^{2n+1}s_1^{n+1}p^{n+2}q^{\binom{n+2}{2}}(1-p)^n}{(1-s_1pq)(1+x-xp)^n(1-q)^n(q)_n(\sigma q)_n}, \tag{6.75}$$

$$c(s_1)\gamma(s_1q)\gamma(s_1q^2)\cdots\gamma(s_1q^{n-1})\beta(s_1q^n) = \frac{x^{2n+2}s_1^{n+1}p^{n+1}q^{\binom{n+2}{2}-1}(1-p)^n(1+x-s_1q^{n+1}(p+xp-x))}{(1+x)(1-s_1pq)(1+x-xp)^n(1-q)^n(1+x-s_1q^n)(1+x-s_1q^{n+1})(q)_n(\sigma q)_n}. \quad (6.76)$$

If we set $n=0$ in equations (6.75) and (6.76) we obtain the expressions for $a(s_1)$ and $b(s_1)$ respectively, and so we can write down the generating function

$$R(1+x) = -\frac{(1+x)pq\sum_{n\geq 0}\frac{x^{2n}s_1^n p^n q^{\binom{n+2}{2}}(1-p)^n}{(1+x-xp)^n(1-q)^n(q)_n(\sigma q)_n}}{x\sum_{n\geq 0}\frac{x^{2n}s_1^n p^n q^{\binom{n+2}{2}}(1-p)^n(1+x-s_1q^{n+1}(p+xp-x))}{(1+x-xp)^n(1-q)^n(1+x-s_1q^n)(1+x-s_1q^{n+1})(q)_n(\sigma q)_n}}. \quad (6.77)$$

Substituting this into equation (6.47), we obtain the generating function for all bargraphs enumerated by number of columns, vertical half-perimeter and site-perimeter:

$$P(1;x,y,p) = -\frac{pq\sum_{n\geq 0}\frac{x^{2n}s_1^n p^n q^{\binom{n+2}{2}}(1-p)^n}{(1+x-xp)^n(1-q)^n(q)_n(\sigma q)_n}}{x\sum_{n\geq 0}\frac{x^{2n}s_1^n p^n q^{\binom{n+2}{2}}(1-p)^n(1+x-s_1q^{n+1}(p+xp-x))}{(1+x-xp)^n(1-q)^n(1+x-s_1q^n)(1+x-s_1q^{n+1})(q)_n(\sigma q)_n}}, \quad (6.78)$$

where $q = py$.

Again we find this new $q$-series involving sums of products of *algebraic* functions. We have not yet been able to determine the asymptotics of this generating function, but as per alternating bargraphs, we believe that the dominant singularity is the square root singularity of $s_1$, and that the model does not undergo a phase transition.

## 6.4

# Partially directed, interacting walks

The self-avoiding walk (SAW) is a classical model of long chain polymers in solution (see [121] and references within); by weighting certain combinatorial properties, SAWs can be made to mimic a wide variety of physical situations. Long chain polymers undergo a variety of phase transitions, and here we consider two of these — polymer collapse and adsorption. These two transitions can be studied using the self-avoiding walk by adding *interactions*.

The addition of bulk interactions in the form of a nearest-neighbour attractive force (giving the interacting self-avoiding walk model — ISAWs) gives rise to a collapse transition. When the attractive force is weak, the typical polymer is an open “coil” configuration. On the other hand, when the force is strong, the typical polymer is a dense “globule”. The transition between these two phases is called the “coil-globule” transition and it occurs at a specific temperature called the $\theta$-point.

Introducing a “wall” and a short range attractive force between it and the polymer one can study the adsorption transition. The phases of the polymer in this model are defined by the fraction of monomers (walk bonds) that are in contact with the wall; if the walk is in the free phase then the fraction of contacts is zero (as the length of the walk tends to infinity), while in the adsorbed phase the fraction of contacts is non-zero (as the length of the walk tends to infinity).

If one introduces both self-attraction and adsorption (by weighting certain combinatorial aspects of the configurations), then in two dimensions there are three phases; free, collapsed and adsorbed (in three dimensions there can also be a collapsed-adsorbed phase).

These three models remain unsolved for general self-avoiding walks. As with many animal enumeration problems, one is able to make some progress towards exact solution if one places topological restrictions on the model, in particular that the walk cannot step in the negative $x$ direction. See figure 6.20.

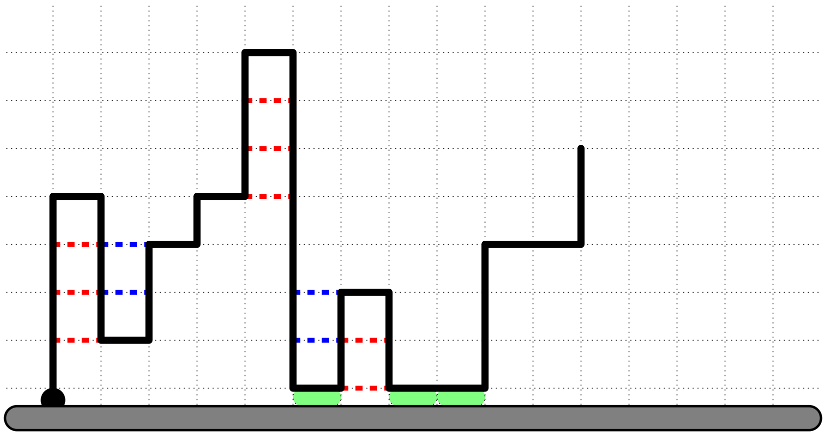


Figure 6.20: A partially directed interacting walk above a wall. We enumerate these walks according to the number of steps, self-interactions and contacts with the wall. This walk contains 40 steps, 12 interactions (of which 4 are left-handed (blue dashed lines) and 8 are right-handed (red dashed lines)), and 3 contacts (highlighted green).

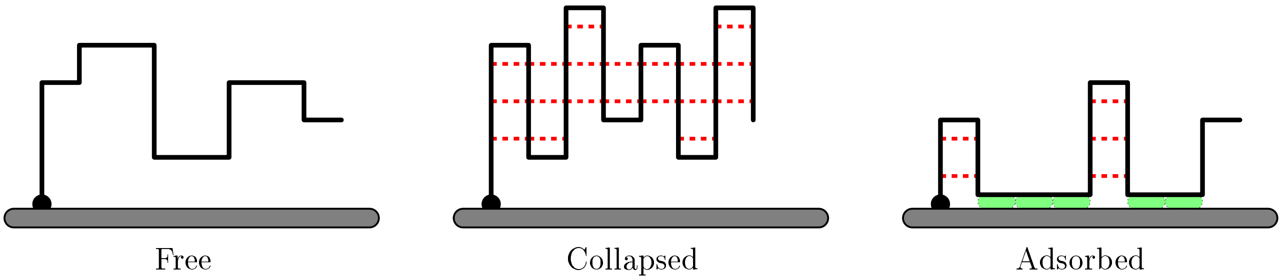



Figure 6.21: The three phases of partially directed interacting walks above an adsorbing wall.

We form the generating function of these walks, $P(x, \kappa, \tau)$, where $x$ enumerates the number of steps, $\tau$ the number of interactions[11] and $\kappa$ the number of contacts. The full

[11]Sorry about the change of notation, but $\tau$ seems to be standard.

generating function has not yet been solved. If the bulk interactions are turned off ($\tau = 1$) and/or the wall is removed, then the model is solvable (see table 6.1). If the interactions *and* the wall are removed, then there is no phase transition. When the interactions *or* the wall is present, then the system does undergo one phase transition (collapse or adsorption) [128, 36].

| **Type of interaction** | **No wall** | **Wall + contact interactions** |
|---|---|---|
| No bulk interactions | Rational<br>No phase transition | Algebraic [128]<br>Free — Adsorbed |
| Left-hand bulk interactions | Rational<br>No phase transition | Horrible $q$-series<br>Free — Adsorbed |
| Full bulk Interactions | $q$-series [36]<br>Free — Collapsed | *Unknown*<br>Free — Collapsed — Adsorbed<br>[154] |

Table 6.1: The effect of self-interaction and an attracting wall on the behaviour of partially directed walks. The phases of the model and the type of generating function are given.

Another possibility is to consider only those interactions that occur on the left-hand side of the walk:

**Definition 6.4.**
If an imaginary walker moves along the bonds of the walk, then *left-hand interactions* are those nearest-neighbour interactions that occur on the left-hand side of the walker. Alternatively, *left-hand interactions* are nearest-neighbour interactions that occur only on the northern side of a partially directed walk.

The only place that left-hand interactions occur in a partially directed walk is in a "well" (see figure 6.20). Consequently they can be enumerated using a similar decomposition to the one we used to enumerate bargraphs — hide the interactions inside the building blocks.

Let us first consider partially directed walks with left-hand interactions without the presence of a wall.

### 6.4.1 Left-hand interactions with no wall

Let $L(s; x, \tau)$ be the generating function of walks that end in *downwards* vertical steps followed by a single horizontal step (like an "L" shape), and let $R(x, \tau)$ be the generating function of all other walks ending in a horizontal step. Using a column-by-column construction one can show that

$$\begin{aligned} R(x,\tau) &= \frac{x}{1-x}\Big(1+R(x,\tau)\Big) + xL(1;x,\tau) + \frac{x^2\tau}{1-x\tau}\Big(L(1;x,\tau)-L(x\tau;x,\tau)\Big) \\ &\qquad + \frac{x^2}{1-x}L(x\tau;x,\tau), \qquad (6.79) \\ L(s;x,\tau) &= \frac{sx^2}{1-sx}\Big(1+L(1;x,\tau)+R(x,\tau)\Big). \qquad (6.80) \end{aligned}$$

These can be solved to give

$$L(1;x,\tau)+R(x,\tau) = \frac{x(x^4\tau+\tau x^3-x^3-x^2\tau-x^2+1)}{1-3x-x^2\tau+3\tau x^3+x^3-x^5\tau+x^2-2x^4\tau+x^4} \qquad (6.81)$$

The generating function of all partially directed walks with left-hand interactions is then

$$P_l(x,\tau) = \frac{L(1;x,\tau)+R(x,\tau)}{x} - 1. \qquad (6.82)$$

The dominant singularity of the generating function is a simple pole given by the zero of the denominator:

$$\tau_c = \frac{1-3x+x^2+x^3+x^4}{x^2(1-3x+x^3+2x^2)}. \qquad (6.83)$$

Since there is no change in the dominant asymptotic behaviour of the coefficients of the generating function, this model does not undergo a phase transition.

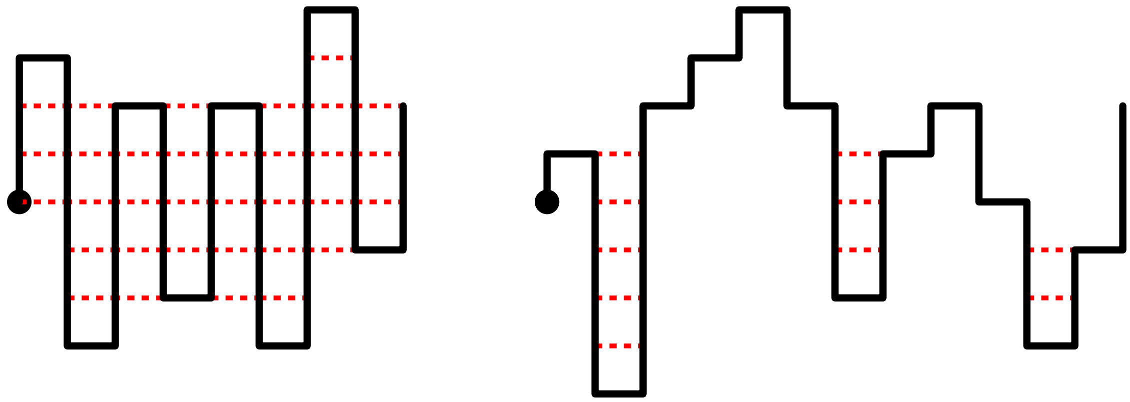


Figure 6.22: (Left) When all interactions are allowed, then both wells and chimneys are encouraged to form, giving rise to very dense configurations. (Right) When only left-hand interactions occur, only well configurations are encouraged, but there is no incentive for these to occur close together.

We can give a heuristic argument as to why this is the case (see figure 6.22). Left-hand interactions favour the formation of wells, but not chimney-configurations (upside-down wells, which are favoured by right-hand interactions). When all interactions occur, then both wells and chimneys are encouraged to form, and the walk is glued into a very dense configuration. On the other hand, when only left-hand interactions occur, only the well configurations are encouraged, and there is no incentive for these configurations to cluster together to form the large "globules" that characterise the collapsed phase, and so the walk does not collapse. Further if partially directed walks with left-hand interactions do not undergo a phase transition, then it is quite unlikely that the presence of a wall will give rise to a collapse transition.

### 6.4.2 Left-hand interactions and a wall

We have not yet explicitly solved this model. Since left-hand interactions occur only in "wells", one can reuse the bargraph and alternating bargraph decomposition. It is not difficult to show that the number of left-hand interactions in a given bargraph (rather than walk) is equal to its perimeter minus the site-perimeter minus the number of inside corners plus the number of wells.

As we have noted above, we do not believe that partially directed walks above a wall with left-hand interactions will display a collapse transition. All in all, this result is quite pessimistic, since the additional work required to enumerate left-hand interactions, as well as the number of steps and contacts, is considerable, and the resulting generating function is (without doubt) a great deal more complicated than that of non-interacting walks above a wall. And yet this jump in complexity is not mirrored by additional complexity in the behaviour of the system.

## 6.5 Conclusions

The site-perimeter of a polygon is arguably a much more difficult statistic to enumerate than its area or (ordinary) perimeter. For convex polygons the site-perimeter of a polygon can be related to its perimeter and the number of inside corners, and so can be enumerated by either a wasp-waist factorisation or a column-by-column construction. However, for even the simplest non-convex family of polygons — bargraphs — the problem is substantially more difficult; it is not obvious how one can apply a wasp-waist factorisation and we must consider the height of the two rightmost columns in order to use a column-by-column construction.

By separating bargraphs into ascending and descending bargraphs, and using the Hadamard-Temperley method we have been able to enumerate bargraph polygons (and another related family of polygons) according to their site-perimeter. The resulting generating function is a new type of complicated $q$-series, and has not been observed before in the enumeration of polygons. Due to the complexity of the generating function we have not yet analysed its asymptotic behaviour, but we believe (on the basis of preliminary analysis and numerical data) that the underlying system does not undergo a phase transition.

We have also explored the links between this model and an unsolved model of polymer collapse and adsorption, self-interacting partially directed walks. Unfortunately, we conclude that a model we are able to solve, partially directed walks with only left-hand interactions, does not undergo a phase transition. Consequently this model does not bring us any close to the behaviour observed in the full model, despite the added complexity in the solution.

# Part II

# Properties of solutions

# CHAPTER 7

# The solvability of self-avoiding polygons

# 7.1

# Anisotropic generating function of SAPs and some observations

In chapter 3 we proved a number of results concerning the structure of anisotropic generating functions of lattice bond animals. Many of these results were *upper bounds* on the exponent of a given cyclotomic factor, $\Psi_k$ in the denominator of $H_n(x)$. We were able to prove that for general bond animals the exponent of $\Psi_k(x)$ in the denominator of $H_n(x)$ is at most zero for $n < 2k-2$ (*i.e.* it cannot appear), and then at most $k$ in the denominator of $H_{2k-2}(x)$ (*i.e.* when it first appears). We did not find upper bounds for the exponent of $\Psi_k(x)$ in the denominator of $H_n(x)$ for general $n$ and $k$.

In this chapter we will examine the anisotropic generating function of self-avoiding polygons in more detail. We will give *upper bounds* for the exponent of $\Psi_k(x)$ in the denominator of $H_n(x)$ for *any* given $n$ and $k$. We will also prove that $\Psi_k(x)$ appears in the denominator of $H_{3k-2}(x)$ with exponent *exactly equal to* 1, for all $k \neq 2$, and so $H_{3k-2}(x)$ is singular at the zeros of $\Psi_k(x)$. We also consider how the singularity structure of the coefficients of the anisotropic generating function effects its solvability.

Let us start by revisiting some of the observed [98] properties of the anisotropic generating function of self-avoiding polygons:

**Observations.**
If we generate (numerically) the anisotropic series $P(x, y)$ up to some finite order in $x$ and $y$, and then consider this to be a formal power series in $y$ with coefficients that are power series in $x$, *i.e.* as $\sum_n H_n(x)y^n$, we observe the following:

1. the coefficients, $H_n(x)$, are rational functions of $x$,

2. the numerator polynomial of $H_n(x)$ has positive unimodal coefficients,

3. the degrees of the numerator and denominator of $H_n(x)$ are equal,

4.the denominator is a product of cyclotomic polynomials:

$$\begin{aligned}
D_1(x) &= (1-x)\\
D_2(x) &= (1-x)^3\\
D_3(x) &= (1-x)^5\\
D_4(x) &= (1-x)^7\\
D_5(x) &= (1-x)^9(1+x)^2\\
D_6(x) &= (1-x)^{11}(1+x)^4\\
D_7(x) &= (1-x)^{13}(1+x)^6(1+x+x^2)\\
D_8(x) &= (1-x)^{15}(1+x)^8(1+x+x^2)^3\\
D_9(x) &= (1-x)^{17}(1+x)^{10}(1+x+x^2)^5\\
D_{10}(x) &= (1-x)^{19}(1+x)^{12}(1+x+x^2)^7(1+x^2)\\
D_{11}(x) &= (1-x)^{21}(1+x)^{14}(1+x+x^2)^9(1+x^2)^3\\
D_{12}(x) &= (1-x)^{23}(1+x)^{16}(1+x+x^2)^{11}(1+x^2)^5\\
D_{13}(x) &= (1-x)^{25}(1+x)^{18}(1+x+x^2)^{13}(1+x^2)^7(1+x+x^2+x^3+x^4)
\end{aligned}$$

5.the cyclotomic factor $\Psi_k(x)$ first enters at $H_{3k-2}(x)$, enters with exponent equal to 1, excepting $\Psi_2(x)$ (which enters at $H_5(x)$ with an exponent of 2).

6.the exponent of $\Psi_k(x)$ in the denominator $H_{3k-2+l}(x)$ is $2l+1$, with the exception of the exponent of $\Psi_2(x)$, which is $2l$.

Unfortunately we are not able to prove all of these observations (in particular those concerning the numerator of $H_n(x)$), however using the partial order, $\preceq_s$, and the Hadamard product methods developed in previous chapters, we are able to show that

1. each coefficient, $H_n(x)$, is a rational function of $x$,
2. the degree of the numerator of $H_n(x)$ is less than or equal to the degree of the denominator,
3. the denominator is a product of cyclotomic polynomials,
4. the cyclotomic factor $\Psi_k(x)$ first enters at $H_{3k-2}(x)$, enters with exponent exactly equal to 1, excepting when $k=2$.
5. the exponent of $\Psi_k(x)$ in the denominator of $H_{3k-2+l}(x)$ is at most $2l+1$.

The first three observations follow directly from theorem 3.8 and indeed are true for any dense family of animals.

By theorem 3.9, we can prove (5) by showing that it is not possible to construct a section-minimal polygon with more than $(2l+1)$ $k$-sections[1] unless one has more than $(6k-4-2l)$

[1] Since we are concerned only with polygons in this chapter, we remind the reader that a $k$-section in a polygon contains $2k$ horizontal bonds.

vertical bonds. To be strictly correct, we need to be a little more careful than this, since a $K$-section will also give rise to a factor of $\Psi_k(x)$ when $K$ is any integer multiple of $k$.

By theorem 3.9, a factor of $\Psi_k(x)$ in the denominator of $H_n(x)$ must be "caused" by a section-minimal polygon in $\mathcal{G}_n$ (*i.e.* having $2n$ vertical bonds) that contains a $k$-section. The section-minimal polygons that first (in the sense of having a minimum number of vertical bonds) contain a $k$-section have a relatively simple characterisation, and we are able to prove (4) by enumerating a simple superset of these animals. Though we do not find an explicit generating function, we are able to use the Hadamard-Temperley techniques developed in chapter 5, to find a functional equation satisfied by the generating function. This functional equation (luckily) is sufficient to show that the exponent of $\Psi_k(x)$ in $H_{3k-2}(x)$ is *exactly* equal to 1 (except when $k = 2$).

## 7.2 Bounding the exponent of $\Psi_k(x)$

### 7.2.1 The first $k$-section, $\Psi_k(x)$ and a little topology

One can consider a self-avoiding polygon to be the embedding of a simple closed loop into the square lattice. We can examine the bonds in a row or column of this polygon by examining the intersection of a line and a simple closed loop:

**Lemma 7.1 (Not quite (but almost) very obvious.).** *If a line cuts a simple closed loop (on the plane) $2k$ times, then the line is naturally divided into $k+1$ segments outside the loop and $k$ segments inside the loop.*

By using this simple observation we can determine the minimum number of vertical bonds that an animal containing a $k$-section must have. Since the line is divided into two types of segment — those inside the loop and those outside we make the following definition.

**Definition 7.1.**
Call the (lattice) cells between consecutive horizontal bonds in a $k$-section a *gap*. If the cells in the gap are within the polygon, call it an *inside gap*, otherwise it must be an *outside gap* (the cells are connected to $\infty$). See figure 7.1.

**Theorem 7.2.** *A $k$-section consists of $k$ inside gaps, and $k-1$ outside gaps. Each inside gap must have at least 1 vertical bond on each side (1 for each cell in the gap). Each outside gap must have 2 vertical bonds on each side (2 for each cell in the gap). Hence a polygon that contains a $k$-section must contain at least $6k-4$ vertical bonds.*

*Proof.* Consider the $k$-section in figure 7.1 and the vertical line through it; the line cuts the polygon $2k$ times. By lemma 7.1 the line can be divided into $k$ segments inside the polygon ($k$ inside gaps) and $k+1$ segments outside the polygon ($k+1$ outside gaps of which 2 lie outside the $k$-section). No section line may pass through one of these gaps (if it did, then the $k$-section is divided into two smaller sections), and so there must be at least one vertical bond on each side of each gap to block the incoming section lines.

Draw a horizontal line through an inside gap. This line must contain at least 3 segments: one inside and two outside. Hence the line cuts the polygon at least twice — and so there must be 2 vertical bonds, one each side (these also block the section lines).

Draw a horizontal line through an outside gap. If the line does not cut the polygon — the polygon is disconnected, which is a contradiction. If the line passes through exactly two vertical bonds, then these must lie on either side of the gap (to block the section lines) and so the gap must be inside the polygon — which contradicts the assumption that the gap is an outside gap. So the line must cut the polygon at least 4 times — hence there are at least 4 vertical bonds, 2 each side.

One can indeed construct a polygon that contains a $k$-section and exactly $6k-4$ vertical bonds (see figure 7.4), and hence this lower bound is tight. ∎

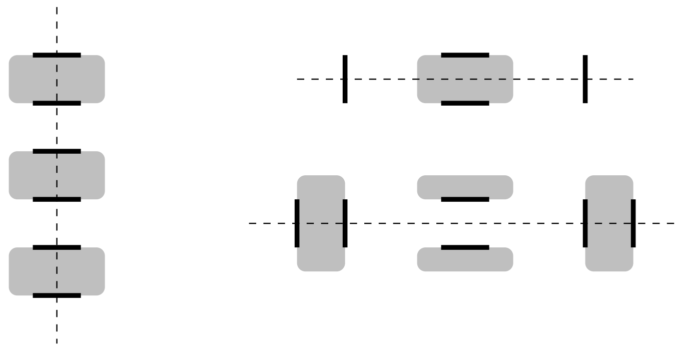


Figure 7.1: A $k$-section has $k$ inside gaps (shaded) and $k-1$ outside gaps. Each inside gap requires 2 vertical bonds, one on each side of the section. Each outside gap requires 4 vertical bonds, 2 on each side of the section.

**Corollary 7.3.** *The factor $\Psi_k(x)$ cannot occur in the denominators $H_n(x)$ for $n < 3k-2$.*

*Proof.* Since a polygon that contains a $k$-section cannot be constructed with less than $6k-4$ vertical bonds, theorem 3.9 implies that $\Psi_k(x)$ cannot occur in the denominators $H_n(x)$ for $n < 3k-2$. ∎

This shows *when* $\Psi_k(x)$ might enter, but does not tell us with what exponent it enters. Hence the need for the following corollary, which we will extend in the following subsection to obtain an upper bound for the exponent of $\Psi_k(x)$ in $H_{3k-2+l}$.

**Corollary 7.4.** *The factor $\Psi_k(x)$ in the denominator of $H_{3k-2}(x)$, cannot occur with an exponent greater than* 1.

*Proof.* If it did occur with an exponent greater than 1, then there would exist a minimal polygon with $6k-4$ vertical bonds and more than one $k$-section. We will show this is false.

If in this polygon there is no more than a single $k$-section in each column then to the left (right) of the left-most (right-most) $k$-section, there must be at least $3k-2$ vertical bonds, and in between each pair of neighbouring $k$-sections there must be at least a single vertical bond. Such a configuration requires more than $6k-4$ vertical bonds.

On the other hand, if there is a column with more than a single $k$-section, this column must contain at least $2k$ inside gaps, and at least $2k-2$ outside gaps. To the left and right of the inside (outside) gaps there must be 2 vertical bonds (4 vertical bonds), totalling no less than $12k-8$ vertical bonds. There must also be a row separating these sections (otherwise all the bonds would belong to the same section); this requires 2 vertical bonds. So the configuration requires at least $12k-6$ vertical bonds. ∎

### 7.2.2 Multiple $k$-sections and the exponent of $\Psi_k(x)$

In this subsection we will extend corollary 7.4 to obtain an upper bound for the exponent of $\Psi_k(x)$ in the denominator of $H_{3k-2+l}(x)$. While the proof we give here is somewhat inelegant it does give the desired bound.

◁ ◁ ◇ ▷ ▷

Let $\mathcal{M} = \{M_i\}$ be the set of section-minimal polygons in $\mathcal{G}_n$, then by theorem 3.9 (in particular see the proof thereof) in chapter 3, the exponent, $\mu_k$, of $\Psi_k(x)$ in the denominator of $H_n(x)$, is bounded above by

$$\mu_k \leq \max_i \left\{ \sum_{d\geq 1} \sigma_{k\cdot d}(M_i) \right\}.$$

For instance an 8-section contributes factors $\Psi_1(x)$, $\Psi_2(x)$, $\Psi_4(x)$ and $\Psi_8(x)$. If $K$ is an integer multiple of $k$, then a $K$-section will contribute a factor of $\Psi_k(x)$; we need a definition to summarise this nicely.

**Definition 7.2.**
We say that a $K$-section is a *$k$-multi-section* if $K$ is an integer multiple of $k$. The generating function of the expansion of any minimal polygon with a $k$-multi-section will contain a factor of $\Psi_k(x)$.

So to determine when $\Psi_k(x)^j$ can occur in a denominator, it is not only necessary to determine the number of vertical bonds required by a polygon with exactly $j$ $k$-sections, but also to determine those polygons with a total of $j$ $k$-multi-sections. In particular we wish to prove that it is not possible to construct a section-minimal polygon containing more than $(2l+1)$ $k$-multi-sections with $(6k-4+2l)$ vertical bonds or less.

The basic idea of the proof is as follows. We first find the maximum number of sections that a minimal polygon of a given height and number of vertical bonds can have. Secondly we determine how many of these must lie outside the extreme left and right $k$-multi-sections. By assuming that all remaining sections are $k$-multi-sections we obtain the upper bound.

### How many pages?

If we wish to determine the maximum number of columns in a polygon, we note that between neighbouring columns of a column-minimal polygon there must be at least one vertical bond. Hence a column-minimal polygon that contains $2V$ vertical bonds can contain at most $2V-1$ columns.

One can show that a section-minimal polygon that contains $2V$ vertical bonds can contain at most $2V-1$ sections, but unfortunately the proof is complicated by the fact that there can be more than one section in a column. In order to deal with this complication it is necessary to examine the *pages* of a polygon. We remind the reader that we make the distinction between vertical bonds that lie within a page (these lie between sections in the same page), and those vertical bonds that lie between pages (these lie between sections in different pages).

**Lemma 7.5.** *Let $P$ be a section-minimal polygon, then if a page within $P$ contains $M$ vertical bonds, then the page contains at most $M+1$ sections. Similarly in any set of $p$ pages of $P$ that contain a total of $M$ vertical bonds there are at most $M+p$ sections.*

*Proof.* Since $P$ is section-minimal, there must be at least 1 vertical bond between neighbouring sections. ∎

**Lemma 7.6.** *A polygon of height $h$ contains no more than $2h-1$ pages.*

*Proof.* Consider figure 7.2. Start from the extreme left of the polygon and move to the extreme right. Initially there must be $(h+1)$ pages, with $h$ section lines between them. Each time a section line from the left is blocked (as we move right) at most one new page is created (*i.e.* two or more pages on the left of the vertical bond(s) become a single page on the right of the bond). Each time a right-hand section line (*i.e.* one that hits the right-hand side of the polygon) emanates from a vertical bond at most two new pages are created (lying on the right of the vertical bond that blocked the section line). Since $h$ section lines from the left and $h$ section lines from the right are blocked, at most $3h$ pages are created — making a total of $(4h+1)$ pages.

On the extreme left of the polygon there must be $(h+1)$ pages; these lie outside the polygon and do not contain any bonds. Similarly on the extreme right there must be $(h+1)$ pages lying outside the polygon devoid of bonds. Hence we are left with at most $2h-1$ pages inside the polygon. ∎

**Corollary 7.7.** *A section-minimal polygon that contains $2V$ vertical bonds contains at most $2V-1$ sections.*

*Proof.* Assume the polygon is of height $h$, and let $M=2V-2h$. The number of vertical bonds within pages is at most $M$. According to lemma 7.6 this polygon contains at most $2h-1$ pages. By lemma 7.5, this polygon contains at most $2h-1+M=2V-1$ sections. ∎

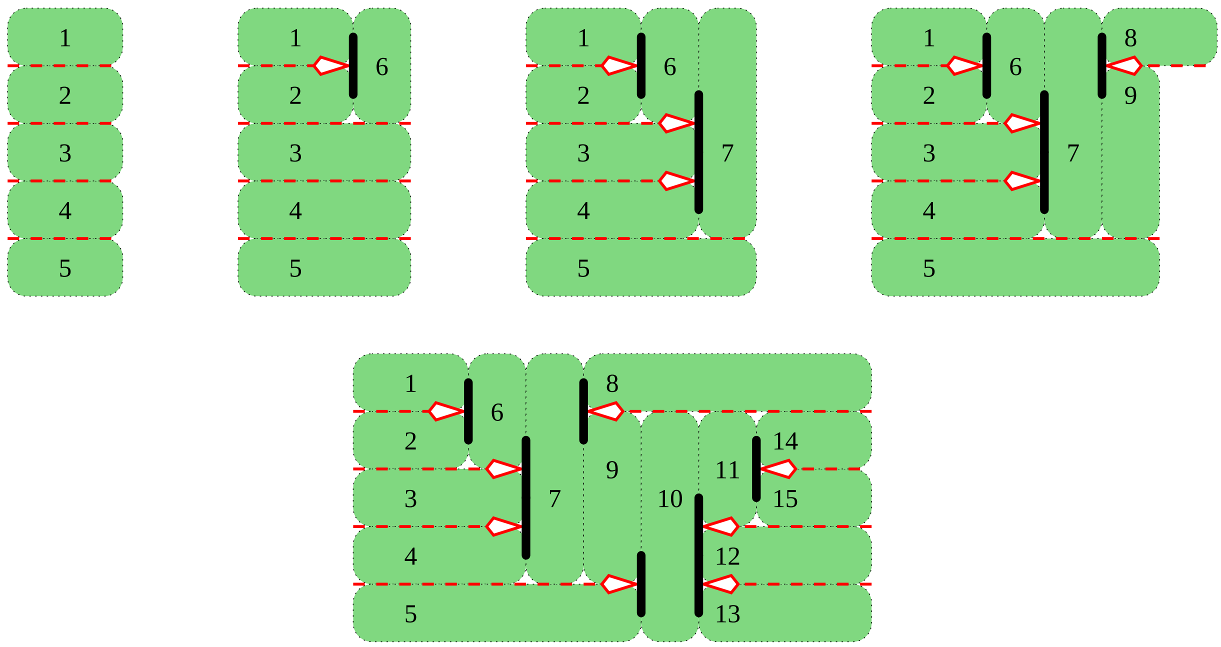



Figure 7.2: A polygon of height $h$. Start from the extreme left and move to the right. Initially there are $(h+1)$ pages (all of which are devoid of bonds and lie outside the polygon). Each time a section line from the left is blocked *at most* one new page is created. Each time a right-hand section line emanates from a vertical bond *at most* two new pages are created. At the extreme right there must be $(h+1)$ pages (again all devoid of bonds and all lying outside the polygon).

**Remark.**
In the work that follows it is useful to assume that every polygon of height $h$ has the potential to have exactly $2h-1$ pages. We will refer to these possible pages as *potential pages*. In particular we will determine how many of these $2h-1$ potential pages may contain $k$-sections and so how many $k$-sections a polygon may contain (by assuming that all these potential pages actually do exist and contain $k$-sections).

◁ ◁ ⋄ ▷ ▷

**Extreme $k$-multi-sections**

Consider a polygon with $2V$ vertical bonds; it can contain at most $2V-1$ sections. A certain number of these $2V-1$ sections must lie outside the outermost (or extreme) $k$-multi-sections and hence cannot be $k$-multi-sections. This is the aim of this subsection. First we must make the idea of an extreme $k$-section more precise.

**Definition 7.3.**

- Two $k$-sections are said to overlap if their projections onto a vertical line have a non-empty intersection.
- If a $k$-section overlaps with a $k$-section on its left and another on its right, then it is *not* an *extreme* $k$-section. On the other hand, if a $k$-section overlaps with another $k$-section on one side and *not* the other, then we say that it is an *extreme* $k$-section.
- If a $k$-section does not overlap with another $k$-section on either its left or its right then it is counted as two extreme $k$-sections (since it is extreme to both its left and its right). See figure 7.3. The reason for double counting such an extreme $k$-section will become clear in lemma 7.8.
- An extreme $k$-multi-section is defined in a similar manner. It is worth noting that an 4-section with a 6-section on either side *can* be an extreme 4-multi-section, but not an extreme 2-multi-section.

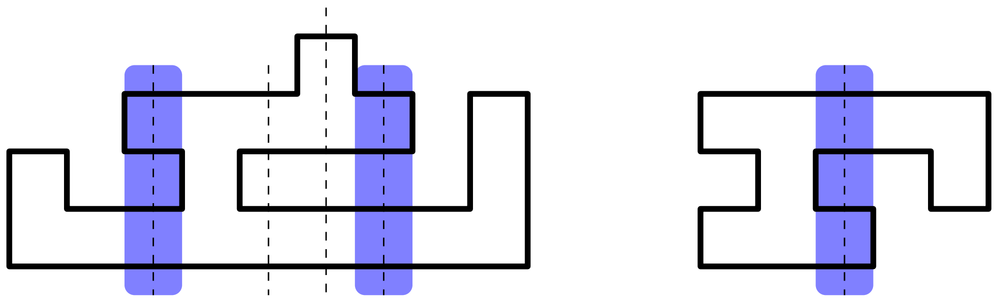


Figure 7.3: The polygon on the left contains four 2-sections, but only 2 of these are extreme. The polygon on the right contains only a single 2-section, but two extreme 2-sections, since there is no overlapping 2-section to either its left or its right.

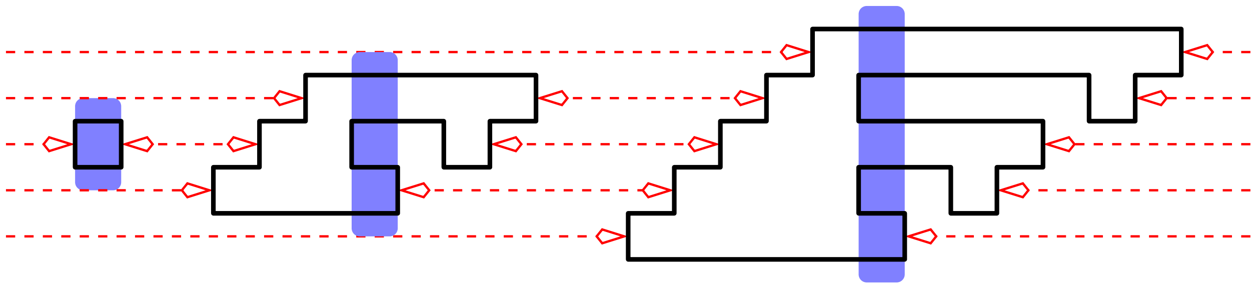


Figure 7.4: Three polygons with a 1, 2 and 3-section respectively (highlighted). In each polygon this section is extreme (in both directions) and hence there are $2(k-1)$ pages and $3k-2$ vertical bonds both to its left and its right.

If we find the extreme left or right $k$-multi-sections, then any section lying outside these cannot be a $k$-multi-section. Hence if we know the number of extreme $k$-multi-sections then we can find the number of sections left over. If we assume that all of these are $k$-multi-sections, then we can obtain an upper bound for the maximum number of $k$-multi-sections.

◁ ◁ ◇ ▷ ▷

**Lemma 7.8.** *To the left or right of an extreme $k$-section, there must be at least $2(k-1)$ potential pages and at least $3k-2$ vertical bonds. More specifically at least $2k-1$ vertical bonds will block section lines and there will be at least $k-1$ not blocking section lines.*

*Proof.* From the proof of theorem 7.2, we see that each $k$-section must have $3k-2$ vertical bonds to its left and another $3k-2$ vertical bonds to its right.

Without loss of generality assume that there is no other $k$-section to the left of the extreme $k$-section. A $k$-section must be of height $2k-1$, and so to the left of an extreme $k$-section, $2k-1$ section lines are blocked. In a similar manner to the proof of lemma 7.6 the blocking of these section lines creates at least $2k-1$ potential pages, the last of which contains the extreme $k$-section, leaving the required number of pages to its left. ∎

### Upper bounds on exponents

We now prove the upper bound on the exponent of $\Psi_k(x)$ by considering a section-minimal polygon that contains $2V$ vertical bonds and is of height $h$. By determining the number of vertical bonds and potential pages that must lie outside the extreme $k$-multi-sections we can determine how many sections are left over. By assuming that all of these remaining sections are $k$-multi-sections we obtain the upper bound.

**Theorem 7.9.**

- *A section-minimal polygon of height $h$, that contains $l$ extreme $k$-multi-sections requires at least $2h + l(k-1)$ vertical bonds.*
- *If a section-minimal polygon of height $h$ has exactly $l$ extreme $k$-multi-sections and $2h + l(k-1) + M$ vertical bonds then it contains at most $(2h-1) + M - 2l(k-1)$ $k$-multi-sections.*
- *Consequently a section-minimal polygon with $2V$ vertical bonds and $l$ extreme $k$-multi-sections, can have at most $S$ $k$-multi-sections, where*

$$S = 2V - 3l(k-1) - 1. \tag{7.1}$$

*Proof.*

- A section-minimal polygon of height $h$ must block $2h$ section lines. This requires $2h$ vertical bonds. By lemma 7.8, to the left (or right) of each extreme $k$-multi-section there will be $\geq 2k-1$ vertical bonds that block section lines (these have already been counted in the $2h$), and at least an additional $\geq k-1$ vertical bonds to its left (or right). Hence the polygon requires an additional $l(k-1)$ vertical bonds, making a total of $\geq 2h + l(k-1)$.

- Consider a section-minimal polygon of height $h$, that contains $l$ extreme $k$-multi-sections, and $2V = 2h + l(k-1) + M$ vertical bonds. By lemma 7.6 this polygon contains $2h-1$ potential pages and at least $2l(k-1)$ of them lie outside the $l$ extreme multi-sections by lemma 7.8. This leaves at most $(2h-1) - 2l(k-1)$ potential pages that can contain $k$-multi-sections.

  Of the $2h + l(k-1) + M$ vertical bonds in the polygon, $2h$ block section lines, and by lemma 7.8 at least $l(k-1)$ lie outside the extreme $k$-multi-sections. This leaves at most $M$ vertical bonds that can lie in the $(2h-1)-2l(k-1)$ potential pages that can contain $k$-multi-sections. By lemma 7.5 there are at most $(2h-1) - 2l(k-1) + M$ sections that are not outside the extreme $k$-multi-sections (and so *can* be $k$-multi-sections).

- Rearranging $2V = 2h + l(k-1) + M$, we have $2h = 2V - l(k-1) - M$, and so (by the arguments of the previous paragraph) there are at most $2V - 3l(k-1) - 1$ sections that are not outside the extreme $k$-multi-sections. If we assume that *all* of these are $k$-multi-sections, then we obtain the upper bound, $S$, on the number of $k$-multi-sections

$$S = 2V - 3l(k-1) - 1.$$

■

**Remark.**
This proof seems to be unnecessarily complicated. If there is only a single $k$-multi-section in each column of the polygon then there are at least 2 extreme $k$-multi-sections, and at least one vertical bond between neighbouring columns. Hence the number of vertical bonds required to construct a section-minimal polygon with $2l+1$ $k$-multi-sections (with only 1 in each column) is at least $6k - 4 + 2l$.

The difficulty occurs when considering the case where there is a column in the polygon that contains more than 1 $k$-multi-section. It would seem to be obvious that the number of vertical bonds required in such a case could not possibly be less, however the author has not yet managed to find a more elegant proof of this.

It is worth noting that when $k = 1$, equation (7.1) is independent of $l$, the number of extreme 1-multi-sections. Indeed, one is able to construct section-minimal polygons with the maximal number of 1-sections (as per theorem 7.9), that contain a column with more than a single 1-section. See figure 7.5. The same is not true when $k > 1$.

◁ ◁ ◇ ▷ ▷

Since we now know how many vertical bonds are required to construct a section-minimal polygon that contains $\alpha$ $k$-multi-sections, we can obtain the required upper bound.

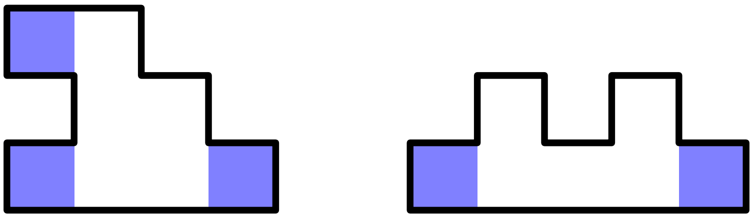


Figure 7.5: To construct a section-minimal polygon with five 1-sections requires 6 vertical bonds. One can construct a section-minimal polygon with five 1-sections but more than 2 extreme 1-sections.

**Corollary 7.10.** *The denominator term $\Psi_k(x)^{2m+1}$ cannot appear in the denominator of $H_n(x)$, for $n < 3k - 2 + m$. Alternatively, the exponent of $\Psi_k(x)$ in the denominator of $H_{3k-2+m}(x)$ cannot be greater than $2m + 1$.*

*Proof.* By theorem 7.9, the maximum number of $k$-multi-sections for a fixed number of vertical bonds is maximised by minimising the number of extreme $k$-multi-sections, $l$. The minimum number of extreme $k$-multi-sections is $l = 2$.

So for fixed $2V$, the upper bound on the number of $k$-sections is $S = 2V - 6k + 5$. Rearranging gives:

$$2V \geq S + 6k - 5.$$

However as we cannot construct a polygon with an odd number of vertical bonds, it is perhaps more accurate to write:

$$V \geq 3k + \left\lceil \frac{S-5}{2} \right\rceil$$

Consequently a section-minimal polygon with $2m+1$ $k$-multi-sections (or indeed $2m$ $k$-multi-sections) cannot be constructed with $< 6k - 4 + 2m$ vertical bonds. The result follows. ∎

**Remark.**
This bound is a tight upper bound, since one can always construct a polygon with $6k-4+2m$ vertical bonds and $2m + 1$ $k$-sections. See figure 7.6.

## 7.3

## *2-4-2* polygons and the exponent of $\Psi_k(x)$ in $H_{3k-2}(x)$

In the previous section we showed that no polygon can contain a $k$-section if it has less than $6k - 4$ vertical bonds. Consequently the factor $\Psi_k(x)$ cannot occur in the denominator of $H_n(x)$ for $n < 3k - 2$. Further, since a section-minimal polygon with more than a single

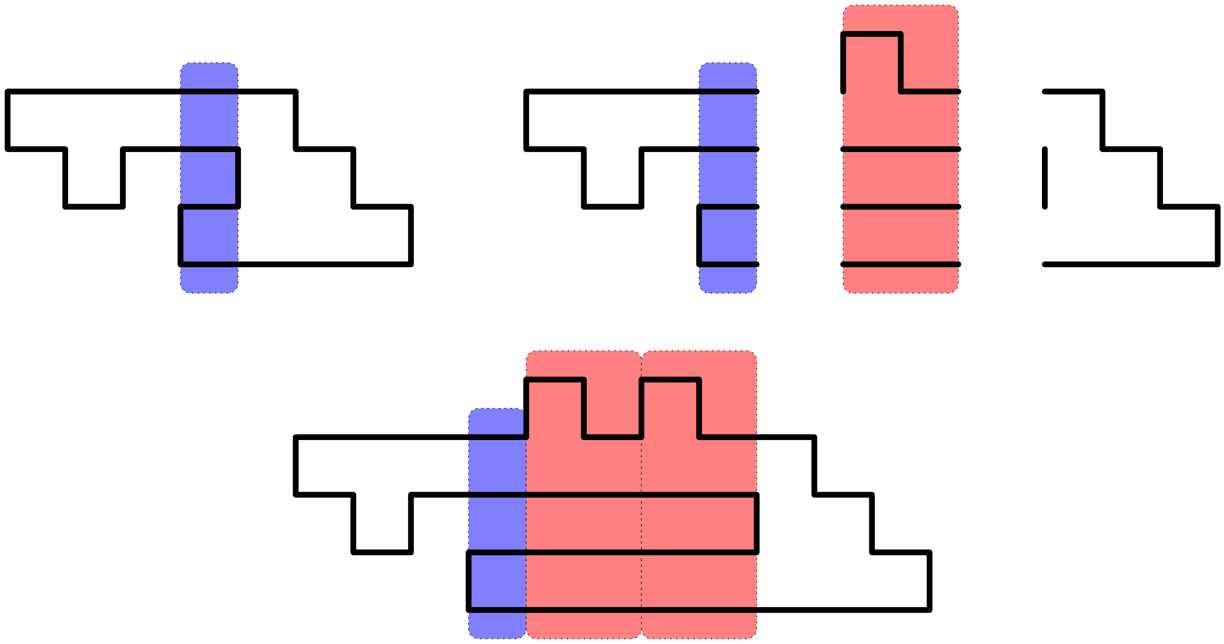


Figure 7.6: To construct a polygon with $2m+1$ $k$-sections and $6k-4+2m$ vertical bonds, start with a polygon with a single $k$-section (highlighted blue) and $6k-4$ vertical bonds as shown (top left). Cut it on the right of the $k$-section. Insert $m$ copies of the pair of $k$-sections (highlighted red) and recombine the polygon. This gives a polygon with $2m+1$ $k$-sections and $6k-4+2m$ vertical bonds.

$k$-multi-section requires more than $6k-4$ vertical bonds, we have also shown that $\Psi_k(x)$ cannot occur in the denominator of $H_{3k-2}(x)$ with exponent greater than 1.

In this section we will characterise those section-minimal polygons that have exactly one $k$-section and exactly $6k-4$ vertical bonds; *i.e.* the polygons that give rise to the first factor of $\Psi_k(x)$. From the characterisation of these section-minimal polygons we can describe a simple superset of their expansions that is dense (in the sense described in chapter 3). The analysis of this superset will show that the factor of $\Psi_k(x)$ in the denominator of $H_{3k-2}(x)$ occurs with exponent *exactly equal to* 1 (except when $k=2$). *i.e.* we will show that the factor of $\Psi_k(x)$ in the denominator of $H_{3k-2}(x)$ does not cancel with the numerator.

### 7.3.1 What are *2-4-2* polygons?

Section-minimal polygons that contain a single $k$-section and $6k-4$ vertical bonds (*i.e.* with the minimum number of vertical bonds) must have 2 vertical bonds in every odd row (first, third ...), and 4 vertical bonds in every even row (second, fourth ...).

By theorem 7.2, a minimal SAP with a single $k$-section requires $\geq 6k-4$ vertical bonds. The number of vertical bonds required is minimised when each gap contains only a single lattice spacing. In this case, each of the $k$ inside gaps requires 2 vertical bonds, and each of the $k-1$ outside gaps require 4 vertical bonds. All the polygons that we wish to study have

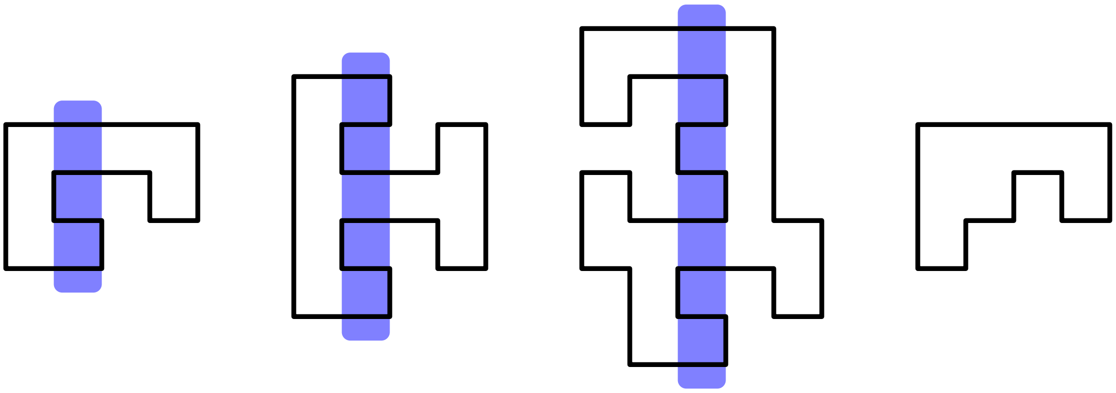


Figure 7.7: Four section-minimal *2-4-2* polygons. The first three polygons contain a 2-section, a 3-section, and a 4-section respectively. The rightmost polygon contains only 1-sections.

this *2-4-2* pattern. Let us give a formal definition of them.

**Definition 7.4.**
We refer to any polygon with this repeating *2-4-2* vertical bond pattern, as *2-4-2 polygons* and write the set of them as $\mathcal{P}_{242}$. We also write $\mathcal{P}_{242}^n = \mathcal{P}_{242} \cap \mathcal{G}_n$.

Note that any polygon that contains exactly $6k-4$ vertical bonds and at least one $k$-section is a *2-4-2* polygon, but not all *2-4-2* polygons that contain $6k-4$ vertical bonds contain a $k$-section (the rightmost polygon in figure 7.7, for example).

◁ ◁ ◇ ▷ ▷

The set of *2-4-2* polygons is too large! If we study these polygons then we are also including polygons that do not contain a $k$-section. This, however, is not a problem. In fact since *2-4-2* polygons have a simple description they are quite easy to study and we can make use of the Hadamard-Temperley techniques described in chapter 5.

Before we can start to examine the generating functions of *2-4-2* polygons we need to prove a little technical result concerning their density.

**Lemma 7.11 (Density of *2-4-2* polygons).** *The expansion of any section-minimal polygon with $6k-4$ vertical bonds and one $k$-section is a subset of $\mathcal{P}_{242}^{3k-2}$. Further $\mathcal{P}_{242}$ is a dense family of polygons.*

*Proof.* Since section deletion and duplication do not alter the number of vertical bonds within each row of an animal, $\mathcal{P}_{242}$ is closed under section duplication and deletion, and hence is dense. Since a section minimal polygon with $6k-4$ vertical bonds and one $k$-section must be an element of $\mathcal{P}_{242}^{3k-2}$, any element of its expansion must also be an element of $\mathcal{P}_{242}^{3k-2}$.

◁ ◁ ◇ ▷ ▷

Due to lemma 7.11 we know that the generating function of $\mathcal{P}_{242}^{3k-2}$ will be a rational function, with cyclotomic factors as given by theorem 3.9. By corollaries 7.3 and 7.4, we expect the highest order cyclotomic factor occurring in the denominator of this generating function to be $\Psi_k(x)$, and that its exponent is at most 1.

We will show that if this factor of $\Psi_k(x)$ does not cancel in the generating function of $\mathcal{P}_{242}^{3k-2}$ then it does not cancel in the denominator of $H_{3k-2}(x)$. However, before we can do so, we will define some notation to make the analysis of the denominators a little easier:

**Notation.**
If $P(x)$ is a polynomial such that

$$P(x) = \prod_{k=1}^{n-1} \Psi_k(x)^{c_k} \qquad \text{with } c_k \in \{0, 1, 2, \dots\},$$

then write $P(x) = \varphi(\Psi_n(x))$. Similarly, if $Q(s, x)$ is a polynomial such that

$$Q(s; x) = (1-s)^{d_0} \prod_{k=1}^{n-1} \Psi_k(x)^{c_k} (1 - sx^k)^{d_k} \qquad \text{with } c_k, d_k \in \{0, 1, 2, \dots\}$$

then write $Q(s; x) = \varphi(1 - sx^n)$.

The following lemma says that if we have a rational function with denominator of the form $\varphi(1 - sx^n)$, then taking its derivative with respect to $s$ does not create new singularities.

**Lemma 7.12.** *If $h(s, x)$ is a rational function of the form $\frac{N(s;x)}{D(s;x)}$ then $\left(\frac{\partial^k h}{\partial s^k}\right)(s; x) = \frac{N_k(s;x)}{D(s;x)^{k+1}}$, for some polynomial $N_k(s; x)$. Hence if the denominator of $h(s, x)$ is $\varphi(1 - sx^n)$, then the denominator of $\left(\frac{\partial^k h}{\partial s^k}\right)(s; x)$ is $\varphi(1 - sx^n)$ as well.*

*Proof.* The quotient rule. ∎

◁ ◁ ◇ ▷ ▷

**Lemma 7.13.** *The factor $\Psi_k(x)$ in the denominator of the generating function of* 2-4-2 *polygons with $6k - 4$ vertical bonds does not cancel with the numerator if and only if the factor of $\Psi_k(x)$ does not cancel in $H_{3k-2}(x)$.*

*Proof.* Since $\mathcal{G}_{3k-2} = \mathcal{P}_{242}^{3k-2} \biguplus (\mathcal{G}_{3k-2} \setminus \mathcal{P}_{242}^{3k-2})$

$$H_{3k-2}(x) = gf(\mathcal{P}_{242}^{3k-2}) + gf(\mathcal{G}_{3k-2} \setminus \mathcal{P}_{242}^{3k-2}).$$

Since the set $\mathcal{G}_{3k-2} \setminus \mathcal{P}_{242}^{3k-2}$ contains no polygon with a $j$-section (with $j \geq k$) we can write $gf(\mathcal{G}_{3k-2} \setminus \mathcal{P}_{242}^{3k-2}) = \frac{N_1(x)}{\varphi(\Psi_k(x))}$. Since each section-minimal polygon in $\mathcal{P}_{242}^{3k-2}$ contains at most a single $k$-section, we can write $gf(\mathcal{P}_{242}^{3k-2}) = \frac{N_2(x)}{\varphi(\Psi_k(x))\Psi_k(x)}$, and so

$$\begin{aligned} H_{3k-2}(x) &= \frac{N_1(x)}{\varphi(\Psi_k(x))} + \frac{N_2(x)}{\varphi(\Psi_k(x))\,\Psi_k(x)} \\ &= \frac{\varphi(\Psi_k(x))\,\Psi_k(x) N_1(x) + \varphi(\Psi_k(x))\,N_2(x)}{\varphi(\Psi_k(x))\,\Psi_k(x)} \end{aligned}$$

Hence $\Psi_k(x)$ does not cancel with $N_2(x)$ if and only if it does not cancel with the numerator of $H_{3k-2}$. ∎

◁ ◁ ◇ ▷ ▷

The above lemma demonstrates why it is perfectly OK to work with *2-4-2* polygons, rather than exactly those polygons containing $6k-4$ vertical bonds and at least one $k$-section. The reason is that lemma 7.13 is valid for any dense superset of these polygons, and we have simply chosen a superset that is easy to describe and analyse.

Since every second row of a *2-4-2* polygon is row-convex, they can be easily studied using the Temperley-Hadamard techniques developed in chapter 5. *2-4-2* polygons that contain at least one $k$-section, it turns out, are harder to describe and harder to analyse.

### 7.3.2 A functional equation for *2-4-2* polygons

We use the Temperley-Hadamard techniques of chapter 5 to obtain a functional equation satisfied by the generating function of *2-4-2* polygons. In particular, let $f(s;x,y)$ be the generating function for *2-4-2* polygons enumerated by bottom row-width, half-horizontal perimeter and half-vertical perimeter ($s, x$ and $y$ respectively).

#### Seeds and building blocks

To find the functional equation of *2-4-2* polygons we need to first find their seed and building blocks. As has been the case for most of the examples to which we have applied the Hadamard-Temperley techniques, the seeds of *2-4-2* polygons are simply rectangles of height 1. The generating function is

$$S(s;x,y) = \frac{sxy}{1-sx}. \tag{7.2}$$

The building blocks are quite complicated and to simplify them a little we have removed their extremities (which we have called “frills”) and illustrated the building blocks (without frills), and the frills separately. The building blocks for *2-4-2* polygons are given in figure 7.8, the “frills” are given in figure 7.9.

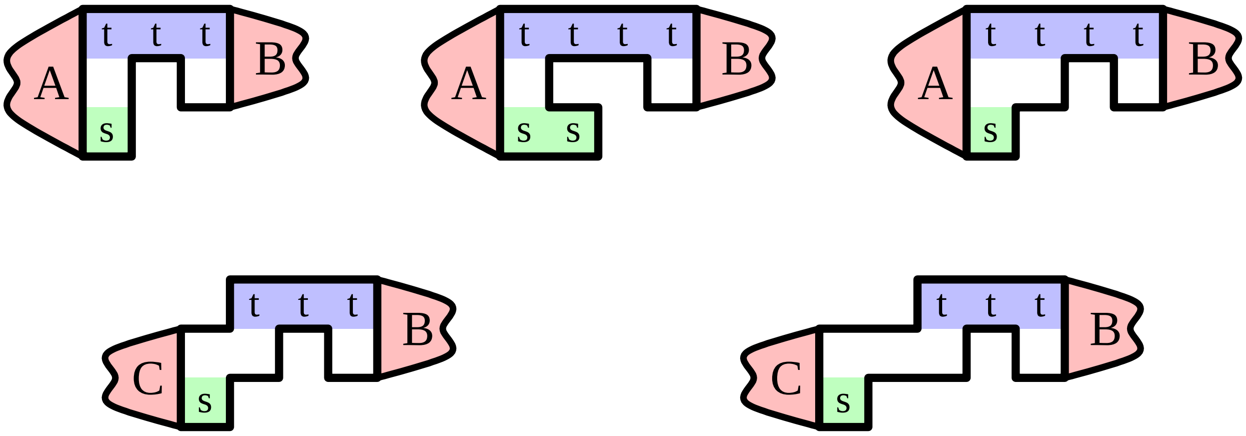



Figure 7.8: *2-4-2* building blocks. The "frills" A,B and C are given in figure 7.9.

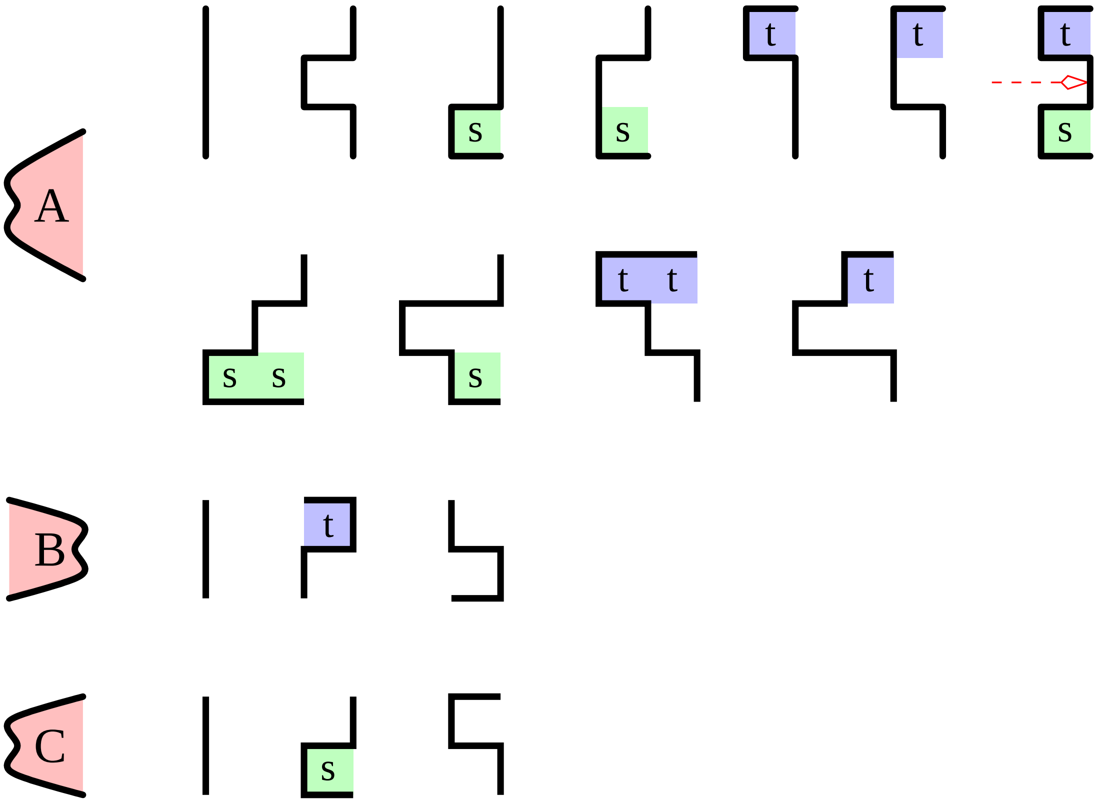



Figure 7.9: The frills of the 2-4-2 building blocks.

The generating functions for the frills are [2]:

$$\begin{aligned} A(s,t;x) &= 1 + [\![ x ]\!] + 2[\![ sx ]\!] + 2[\![ tx ]\!] + [\![ sx ]\!][\![ tx ]\!] + \\ & \quad [\![ sx ]\!]^2 + [\![ sx ]\!][\![ x ]\!] + [\![ tx ]\!]^2 + [\![ tx ]\!][\![ x ]\!] && (7.3) \\ B(s,t;x) &= 1 + [\![ tx ]\!] + [\![ x ]\!] && (7.4) \\ C(s,t;x) &= 1 + [\![ sx ]\!] + [\![ x ]\!] && (7.5) \end{aligned}$$

We can now write down the generating function, $\hat{T}(s,t;x,y)$ for building blocks in *one orientation* (as per figure 7.8) enumerated by half-horizontal perimeter ($x$), half-vertical perimeter ($y$), top and bottom row lengths ($t$ and $s$ respectively):

$$\begin{aligned} \hat{T}(t,s;x,y) &= y^4 \Big( A(s,t;x) \cdot [\![ stx ]\!][\![ tx ]\!]^2 \cdot B(s,t;x) \\ & \quad + A(s,t;x) \cdot [\![ stx ]\!][\![ stx^2 ]\!][\![ tx ]\!]^2 \cdot B(s,t;x) \\ & \quad + A(s,t;x) \cdot [\![ stx ]\!][\![ tx ]\!]^3 \cdot B(s,t;x) \\ & \quad + C(s,t;x) \cdot [\![ sx ]\!][\![ tx ]\!]^3 \cdot B(s,t;x) \\ & \quad + C(s,t;x) \cdot [\![ sx ]\!][\![ x ]\!][\![ tx ]\!]^3 \cdot B(s,t;x) \Big) \qquad (7.6) \end{aligned}$$

This is only the generating function of the building blocks in one of four orientations. The other three orientations can be obtained by reflecting these building blocks across a horizontal and also a vertical line (see figure 7.10).

Reflecting across a vertical line corresponds to multiplying the generating function by 2, while reflecting across a horizontal line corresponds to interchanging $t$ and $s$. Hence the generating function of all the building blocks is:

$$\tilde{T}(t,s;x,y) = 2\left(\hat{T}(t,s;x,y) + \hat{T}(s,t;x,y)\right). \qquad (7.7)$$

When the building blocks are connected to the polygon (using the Hadamard product), the number of horizontal bonds is over-counted by twice the length of the join (since there should be no horizontal bonds *inside* the polygon), and the number of vertical bonds is over-counted by 2. To take account of this over-counting we substitute $t \to t/x$ and divide by $y$. So the final generating function is $T(t,s;x,y) = \tilde{T}(t/x,s;x,y)/y$, which is too big a mess to state here in full.

### Transition function when $s \neq 1$

When $s \neq 1$, we can rewrite $T(t,s;x,y)$ in partial fraction form:

$$T(t,s;x,y) = y^3 \left[ c_0 \cdot t^0 + \sum_{k=1}^{6} c_k \frac{(k-1)!t^{k-1}}{(1-t)^k} + c_7 \frac{1}{1-st} + c_8 \frac{1}{1-stx} \right],$$

[2] $[\![ f ]\!] = \frac{f}{1-f} = \sum_{n \geq 1} f^n$, is the generating function for any non-empty sequence of objects enumerated by $f$.

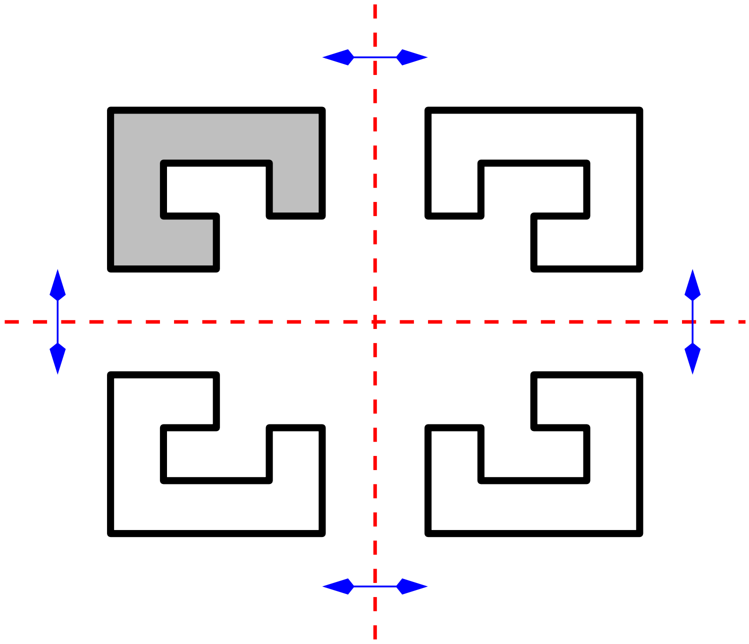


Figure 7.10: The four orientations of *2-4-2* building blocks.

where the $c_i$ are rational functions of $s$, $x$ and $y$. The Hadamard product $f(t;x,y) \odot_t T(t,s;x,y)$ (since there are no rows of zero length, $[t^0]f(t;x,y) = 0$) is then:

$$f(t;x,y) \odot_t T(t,s;x,y) = y^3 \left[ \sum_{k=0}^{5} c_{k+1} \left( \frac{\partial^k f}{\partial t^k} \right) (1;x,y) + c_7 f(s;x,y) + c_8 f(sx;x,y) \right] \tag{7.8}$$

We do not state in full the coefficients, $c_i$, since they are very large and, with the exception of $c_8$, not particularly relevant to the following analysis. We will just state the denominators of all the coefficients, as well as the coefficient $c_8$ in full. If we write the denominator of $c_i$ as $d_i$:

$$\begin{aligned}
d_0 &= (1-x)^3(1-sx)^6(1-s)^6 \\
d_1 &= (1-x)^3(1-sx)^5(1-s)^5 \\
d_2 &= (1-x)^3(1-sx)^3(1-s)^4 \\
d_3 &= (1-x)^3(1-sx)^3(1-s)^3 \\
d_4 &= (1-x)^2(1-sx)^1(1-s)^2 \\
d_5 &= (1-x)^1(1-sx)^1(1-s)^1 \\
d_6 &= (1-s)^1 \\
d_7 &= (1-sx)^6(1-s)^6 \\
c_8 &= -\frac{2sx^2(s^2x^2+sx-s+1)}{(1-sx)^4(1-x)^2}
\end{aligned} \tag{7.9}$$

All of the coefficients are rational functions of the form $\frac{N(s;x)}{\varphi(1-sx^2)}$.

**Transition function when** $s = 1$

For $s = 1$ the pole structure of $T(t, s; x, y)$ changes and the pole at $t = 1/s$ coalesces with the pole at $t = 1$. This changes equation (7.8) to:

$$f(t; x, y) \odot_t T(t, 1; x, y) = y^3 \left[ \sum_{k=0}^{6} \hat{c}_{k+1} \left( \frac{\partial^k f}{\partial t^k} \right) (1; x, y) + \hat{c}_8 f(x; x, y) \right] \tag{7.10}$$

The coefficients, $\hat{c}_i$, become somewhat simpler and can be stated here in full:

$$\begin{aligned}
\hat{c}_0 &= -2\frac{x^3(1+x)(2x^2+1)}{(1-x)^6} \\
\hat{c}_1 &= 4\frac{(1+x)(x^2+1)x^3}{(1-x)^6} \\
\hat{c}_2 &= 2\frac{x^2(1+x)(2x^2+x+1)}{(1-x)^5} \\
\hat{c}_3 &= \frac{x^2(1+x)(2x+1)}{(1-x)^4} \\
\hat{c}_4 &= \frac{1}{3}\frac{(1+x)(x^2+x+1)}{(1-x)^3} \\
\hat{c}_5 &= \frac{1}{12}\frac{(x^2+2x+3)}{(1-x)^2} \\
\hat{c}_6 &= \frac{1}{60}\frac{(x+3)}{(1-x)} \\
\hat{c}_7 &= \frac{1}{360} \\
\hat{c}_8 &= -2\frac{x^3(1+x)}{(1-x)^6}
\end{aligned} \tag{7.11}$$

All of the coefficients are rational functions of the form $\frac{N(x)}{\varphi(\Psi_2(x))}$.

### 7.3.3 Analysis of the functional equation

**Lemma 7.14.** *Let $f(s; x, y)$ be the generating function for 2-4-2 polygons enumerated by bottom row-width, half-horizontal perimeter and half-vertical perimeter ($s, x$ and $y$ respectively). $f(s; x, y)$ satisfies the following functional equations:*

$$f(s; x, y) = \frac{sxy}{1-sx} + y^3 \left[ \sum_{k=0}^{5} c_{k+1} \left( \frac{\partial^k f}{\partial s^k} \right) (1; x, y) + c_7 f(s; x, y) + c_8 f(sx; x, y) \right] \tag{7.12}$$

$$f(1; x, y) = \frac{xy}{1-x} + y^3 \left[ \sum_{k=0}^{6} \hat{c}_{k+1} \left( \frac{\partial^k f}{\partial s^k} \right) (1; x, y) + \hat{c}_8 f(x; x, y) \right] \tag{7.13}$$

*with $c_i$ and $\hat{c}_i$ given above.*

*Rewriting the generating function as $f(s;x,y) = \sum_{n\geq 1} f_n(s;x)y^{3n-2}$, where the coefficient $f_n(s;x)$ is the generating function for $\mathcal{P}_{242}^{3n-2}$, allows the above functional equations to be rewritten:*

$$\begin{aligned}
f_1(s;x) &= \frac{sx}{1-sx} \\
f_{n+1}(s;x) &= \sum_{k=0}^{5} c_{k+1}\left(\frac{\partial^k f_n}{\partial s^k}\right)(1;x) + c_7 f_n(s;x) + c_8 f_n(sx;x) \qquad s \neq 1 \\
f_{n+1}(1;x) &= \sum_{k=0}^{6} \hat{c}_{k+1}\left(\frac{\partial^k f_n}{\partial s^k}\right)(1;x) + \hat{c}_8 f_n(x;x)
\end{aligned}$$

*Proof.* Apply lemma 5.2 to the partial fraction form of the transition function for general $s$, and when $s = 1$. ■

**Remark.**
The above functional equations and recurrences could, perhaps, be solved in closed form using the techniques of chapters 5 and 6. Luckily for the purposes of this chapter we do not need a closed form solution; all we need is to know something of the singularities of the solution. This is much easier.

◁ ◁ ◇ ▷ ▷

According to lemma 7.13 if $\Psi_k(x)$ does not cancel in the denominator of the generating function of $\mathcal{P}_{242}^{3k-2}$ then it does not cancel in the denominator of $H_{3k-2}$. The generating function of $\mathcal{P}_{242}^{3k-2}$ is exactly $f_k(1;x)$. In order to determine the existence of a factor of $\Psi_k(x)$ in the denominator of $f_k(1;x)$, we need to know something of the structure of $f_k(s;x)$:

**Lemma 7.15.** *$f_n(s;x)$ is a rational function of the form*

$$f_n(s;x) = \frac{N_n(s;x)}{D_n(s;x)(1-sx^n)},$$

*for some polynomials $N_n(s;x)$ and $D_n(s;x)$, with $D_n(s;x) = \varphi(1-sx^n)$ and $D_n(1;x) \neq 0$.*

*Proof.* We will prove the lemma by induction. For $n = 1$ the lemma holds.

Assume that the lemma holds for $f_n(s;x)$, and substitute $f_n(s;x)$ into the functional equations from lemma 7.14:

$$\begin{aligned}
f_{n+1}(1;x) &= \sum_{k=0}^{6} \hat{c}_{k+1}\left(\frac{\partial^k f_n}{\partial s^k}\right)(1;x) + \hat{c}_8 f_n(x;x) \\
f_{n+1}(s;x) &= \sum_{k=0}^{5} c_{k+1}\left(\frac{\partial^k f_n}{\partial s^k}\right)(1;x) + c_7 f_n(s;x) + c_8 f_n(sx;x)
\end{aligned}$$

Since $D_n(s;x) = \varphi(1-sx^n)$ and is non-zero at $s=1$, lemma 7.12 implies that $\left(\frac{\partial^k f_n}{\partial s^k}\right)(1;x)$ is of the form, $\frac{\text{<mess>}}{\varphi(\Psi_{n+1}(x))}$. The first of the above recurrences then gives

$$f_{n+1}(1;x) = \frac{<\text{mess}>}{\varphi(\Psi_{n+1}(x))\,\Psi_{n+1}(x)},$$

(but does not tell us whether or not the factor of $\Psi_{n+1}(x)$ cancels with the numerator) hence $f_{n+1}(s;x)$ is not singular at $s=1$. The second recurrence then implies that:

$$\begin{aligned} f_{n+1}(s;x) &= \frac{<\text{mess}>}{\varphi(\Psi_{n+1}(x))} + c_7(s;x)\frac{N_n(s;x)}{\varphi(1-sx^{n+1})} + c_8(s;x)\frac{N_n(sx;x)}{\varphi(1-sx^{n+1})\,(1-sx^{n+1})} \\ &= \frac{N_{n+1}(s;x)}{\varphi(1-sx^{n+1})\,(1-sx^{n+1})} = \frac{N_{n+1}(s;x)}{D_{n+1}(s;x)(1-sx^{n+1})} \end{aligned} \tag{7.14}$$

Since $f_{n+1}(s;x)$ is not singular at $s=1$, the factors of $(1-s)$ arising from the denominators of $c_1(s;x),\ldots,c_7(s;x)$ must cancel and so $D_{n+1}(1;x) \neq 0$.

■

◁ ◁ ◇ ▷ ▷

**Remark.**
It should be noted that this lemma does not show that the factor of $(1-sx^n)$ in the denominator does not cancel with the numerator. It could also be the case that if $(1-sx^n)$ did cancel with the numerator of $c_8(s;x)$, then the induction step in the proof of the above lemma would break down. Luckily, all we need from the above lemma is the structure of the denominator of $f_n(s,x)$; it does not matter if there are cancellations.

On the other hand, when we do attempt to show that there are no cancellations we use a very similar induction. So we do need to know if $\Psi_n(x)$ cancels with the numerator of $c_8(s;x)$. More specifically, we need to know if the numerator of $c_8(x^k,x)$ contains any cyclotomic factors. This is the purpose of the following lemma.

**Lemma 7.16.** *Let $P(s;x) = (s^2x^2 + sx - s + 1)$, then $P(x^k,x)$ has only 1 zero on the unit circle $x=-1$, when $k$ is even. And hence $P(x^k,x)$ is only divisible by $\Psi_n(x)$, when $n=2$ and $k$ is even.*

*Proof.* $P(x^k,x) = x^{2k+2} + x^{k+1} - x^k + 1$, hence the zeros must satisfy:

$$\begin{aligned} x^{2k+2}+1 &= x^k - x^{k+1} \\ x^{k+1} + x^{-k-1} &= 1/x - 1 && \text{assume that } x = e^{i\theta} \\ e^{i(k+1)\theta} + e^{-i(k+1)\theta} &= e^{-i\theta} + 1 \\ 2\cos((k+1)\theta) &= e^{-i\theta} + 1 \end{aligned}$$

The left hand-side of the above expression is real, and hence the right hand-side must also be real, and hence $\theta = 0, \pi$ and $x = \pm 1$. Let $x=1$, then $P(1,1) = 1+1-1+1 \neq 0$. Let $x=-1$, then $P((-1)^k, -1) = P(-1, -1) = 4$ if $k$ is odd. If $k$ is even $P((-1)^k, -1) = P(1, -1) = 0$.

$\triangleleft \lhd \diamond \rhd \triangleright$

We now have sufficient lemmas to prove the main theorem:

**Theorem 7.17 (Poles of *2-4-2* polygons.).** *The generating function for* 2-4-2 *polygons with* $6n - 4$ *vertical bonds,* $f_n(1; x)$*, is of the form:*

$$f_n(1; x) = \frac{N_n(1; x)}{\varphi(\Psi_n(x))\, \Psi_n(x)}$$

*where* $\Psi_n(x)$ *does not divide* $N_n(1; x)$*, except when* $n = 2$.

*Proof.* We show by induction on $j$ that $\Psi_k(x)$ does not divide $N_j(x^{k-j}; x)$. Setting $j = k$ completes the proof.

Since $f_1(s; x) = \frac{sx}{1-sx}$, we have $f_1(x^{k-1}; x) = \frac{x^k}{1-x^k}$ and so $\Psi_k(x)$ does not divide $N_1(x^{k-1}; x)$.

Let us assume that $\Psi_k(x) \nmid N_{j-1}(x^{k-j+1}; x)$, and we will show that $\Psi_k(x) \nmid N_j(x^{k-j}; x)$. Starting with the functional equation for $f_{n+1}(s; x)$:

$$\begin{aligned} f_{n+1}(s; x) &= \frac{< \text{mess} >}{\varphi(1 - sx^{n+1})} + c_8(s; x) \frac{N_n(sx; x)}{\varphi(1 - sx^{n+1})\,(1 - sx^{n+1})} \\ &= \frac{(1 - sx^{n+1})\varphi(1 - sx^{n+1}) < \text{mess} > + \varphi(1 - sx^{n+1})\, \eta(s; x) N_n(sx; x)}{\varphi(1 - sx^{n+1})\,(1 - sx^{n+1})}, \end{aligned}$$

where $\eta(s; x) = -2sx^2(s^2x^2 + sx - s + 1)$ is the numerator of $c_8(s; x)$. When $s = 1$ we obtain a similar recurrence:

$$f_{n+1}(1; x) = \frac{\Psi_{n+1}(x)\varphi(\Psi_{n+1}(x)) < \text{mess} > + \varphi(\Psi_{n+1}(x))\, \hat{\eta}(x) N_n(x; x)}{\varphi(\Psi_{n+1}(x))\, \Psi_{n+1}(x)},$$

where $\hat{\eta}(x) = \eta(1; x) = -2x^3(1 + x)$, which is the numerator of $\hat{c}_8(x)$.

Let us substitute $s = x^{k-j}$ and $n = j - 1$ into the above functional equations; this gives (including when $j = k$, $s = 1$):

$$f_j(x^{k-j}; x) = \frac{\Psi_k(x) < \text{mess} > + \varphi(\Psi_k(x))\, \eta(x^{k-j}; x) N_{j-1}(x^{k-j+1}; x)}{\varphi(\Psi_k(x))\, \Psi_k(x)}.$$

We see that for all $k$ and $j$ (excepting when $k = j = 2$), $\Psi_k(x)$ neither divides $N_{j-1}(x^{k-j+1}; x)$ nor $\eta(x^{k-j}; x)$ (by lemma 7.16), and so does not divide $N_j(x^{k-j}; x)$. Setting $j = k$ gives $\Psi_k(x) \nmid N_k(1; x)$ which completes the proof.

In the case that $k = j = 2$ the proof breaks down since $\Psi_2(x) = (1 + x)$ is a factor of $\eta(1; x)$. Consequently $\Psi_2(x)$ does not appear in the denominator of $f_2(1; x)$ and so, by lemma 7.13, does not occur in the denominator of $H_4(x)$. This concurs with observation. ∎

**Corollary 7.18.** *The generating function $H_{3k-2}(x)$, of self-avoiding polygons with $6k-4$ vertical bonds has a denominator of the form $\varphi(\Psi_k(x))\,\Psi_k(x)$, where $\Psi_k(x)$ does not divide the numerator polynomial, except when $k=2$.*

*Proof.* Combine theorem 7.17 and lemma 7.13. ■

◁ ◁ ◇ ▷ ▷

This shows that the coefficients of $y$ in the anisotropic generating function of self-avoiding polygons have an infinite number of singularities, and that these singularities form a set that becomes dense on the unit circle, $|x|=1$. In stark contrast, the coefficients of all solved polygon models have only a finite number of singularities (1 or 2 in fact). Such a nasty analytic property implies that $P(x,y)$ is not at all a simple function. In fact $P(x,y)$ is not expressible in terms of the (arguably) most common functions of mathematics and physics — those that satisfy linear differential equations with polynomial coefficients — differentiably-finite power series.

## 7.4 Classes of formal power series

In this section we give a brief introduction to several families of formal power series. In particular let us consider formal power series in $k$ variables, with coefficients in $\mathbb{C}$.

**Definition 7.5.**
The power series we will consider reside within the following rings and fields.

- $\mathbb{C}[x_1,\ldots,x_k]$ is the ring of polynomials in $x_1,\ldots,x_k$ with coefficients in $\mathbb{C}$, and
- $\mathbb{C}[[x_1,\ldots,x_k]]$ is the ring of formal power series in $x_1,\ldots,x_k$ with coefficients in $\mathbb{C}$.
- $\mathbb{C}(x_1,\ldots,x_k)$ is the field of rational functions in $x_1,\ldots,x_k$ with coefficients in $\mathbb{C}$, and
- $\mathbb{C}((x_1,\ldots,x_k))$ is the field of formal Laurent series in $x_1,\ldots,x_k$ with coefficients in $\mathbb{C}$.

### 7.4.1 Rational

**Definition 7.6.**
A power series $f(t)\in\mathbb{C}[[t]]$ is *rational* if there exist two polynomials, $P(t)$ and $Q(t)$, with coefficients in $\mathbb{C}$, such that

$$f(t)=\frac{P(t)}{Q(t)}. \tag{7.15}$$

Similarly a power series $f(x_1,\ldots,x_k) \in \mathbb{C}[[x_1,\ldots,x_k]]$ is *rational* if there exist two polynomials, $P(x_1,\ldots,x_k)$ and $Q(x_1,\ldots,x_k)$, with coefficients in $\mathbb{C}$ such that

$$f(\mathbf{x}) = \frac{P(\mathbf{x})}{Q(\mathbf{x})}. \tag{7.16}$$

The generating function $f(t)$ of all words in the language $\{a,b,c\}^*$ enumerated by length is given by

$$f(t) = \sum_{n\geq 0} 3^n t^n = \frac{1}{1-3t}. \tag{7.17}$$

The generating function of binomial coefficients is given by

$$f(x,y) = \sum_{n,m\geq 0} \binom{n}{m} x^n y^m = \frac{1}{1-x(1+y)}. \tag{7.18}$$

Both of these generating functions are rational power series.

### 7.4.2 Algebraic

**Definition 7.7.**
A power series $f(t) \in \mathbb{C}[[t]]$ is *algebraic* if there exists a polynomial, $P$, in 2 variables over $\mathbb{C}$ such that

$$P\left(t, f(t)\right) = 0 \tag{7.19}$$

Similarly a power series $f(x_1,\ldots,x_k) \in \mathbb{C}[[(x_1,\ldots,x_k]]$ is *algebraic* if there exists a polynomial, $P$, in $k+1$ variables over $\mathbb{C}$ such that

$$P\left(x_1,\ldots,x_k, f(\mathbf{x})\right) = 0 \tag{7.20}$$

All rational power series are also algebraic.

The generating function of staircase polygons enumerated by their half-perimeter, $f(t) = \frac{1}{2}\left(1-\sqrt{1-4t}\right) - t$, satisfies the algebraic equation

$$f(t) - \left(t + f(t)\right)^2 = 0. \tag{7.21}$$

Similarly the anisotropic generating function of staircase polygons, $f(x,y)$ satisfies

$$f(x,y) - (x + f(x,y))(y + f(x,y)) = 0. \tag{7.22}$$

Hence each of these generating functions is algebraic.

### 7.4.3 Differentiably-finite

The solutions of ordinary differential equations with polynomial coefficients are arguably the most important class of formal power series in mathematics and physics. Such power series are called differentiably-finite power series. The solutions to many combinatorial and physical models belong to this class. They are very easy objects to work with since knowledge of the differential equation satisfied by the power series (even if it cannot be solved in closed form), is sufficient to compute their coefficients very efficiently (in linear time) and also to determine their asymptotic behaviour.

**Definition 7.8.**
A power series $f(t)$ is *differentiably finite* (or D-finite) if it and its derivatives $f'(t), f''(t), \dots,$ span a finite-dimensional subspace of $\mathbb{C}((t))$ regarded as a vector space over $\mathbb{C}(t)$. Equivalently, a power series $f(t) \in \mathbb{C}[[t]]$ is *differentiably finite* if there exist $k+1$ polynomials, $P_0(t), P_1(t), \dots, P_k(t)$ such that

$$P_0(t)f(t) + P_1(t)f'(t) + \cdots + P_k(t)f^{(k)}(t) = 0. \tag{7.23}$$

Equivalently, a power series $f(t) = \sum_{n\geq 0} a_n t^n$ is D-finite, if there exist an integer $k$, polynomials $p_0(n), \dots, p_k(n)$ with coefficients in $\mathbb{C}$, and an integer $n_0$, such that for all $n \geq n_0$, the coefficients $a_n$ satisfy

$$p_0(n)a_n + p_1(n)a_{n-1} + \cdots + p_k(n)a_{n-k} = 0. \tag{7.24}$$

Similarly a power series $f(x_1, \dots, x_k) \in \mathbb{C}[[x_1, \dots, x_k]]$ is *differentiably finite* if it and its partial derivatives span a finite dimensional vector space over $\mathbb{C}(x_1, \dots, x_k)$. Equivalently there exist $k$ differential equations (one for each variable) of the form:

$$\left(A_{in_i}(\mathbf{x})\left(\frac{\partial}{\partial x_i}\right)^{n_i} + A_{in_i-1}(\mathbf{x})\left(\frac{\partial}{\partial x_i}\right)^{n_i-1} + \cdots + A_{i0}(\mathbf{x})\right) f = 0, \tag{7.25}$$

with $A_{ij} \in \mathbb{C}[x_1, \dots, x_k]$.

It can be shown that any algebraic function is D-finite [72].

### 7.4.4 Differentiably algebraic

**Definition 7.9.**
A power series, $f(t) \in \mathbb{C}[[t]]$ is *differentiably algebraic* if there exists some non-zero polynomial $P$ in $n+2$ variables over $\mathbb{C}$ such that

$$P(t, f, f', \dots, f^{(n)}) = 0 \tag{7.26}$$

Differentiably algebraic functions can have very nasty analytic properties, including natural boundaries. The partition generating function

$$P(t) = \prod_{n\geq 1}(1-t^n)^{-1}$$

is known to be differentiably algebraic, and is singular on $|t| = 1$ [17]. Very little is known about them — it is not even known if they are closed under addition!

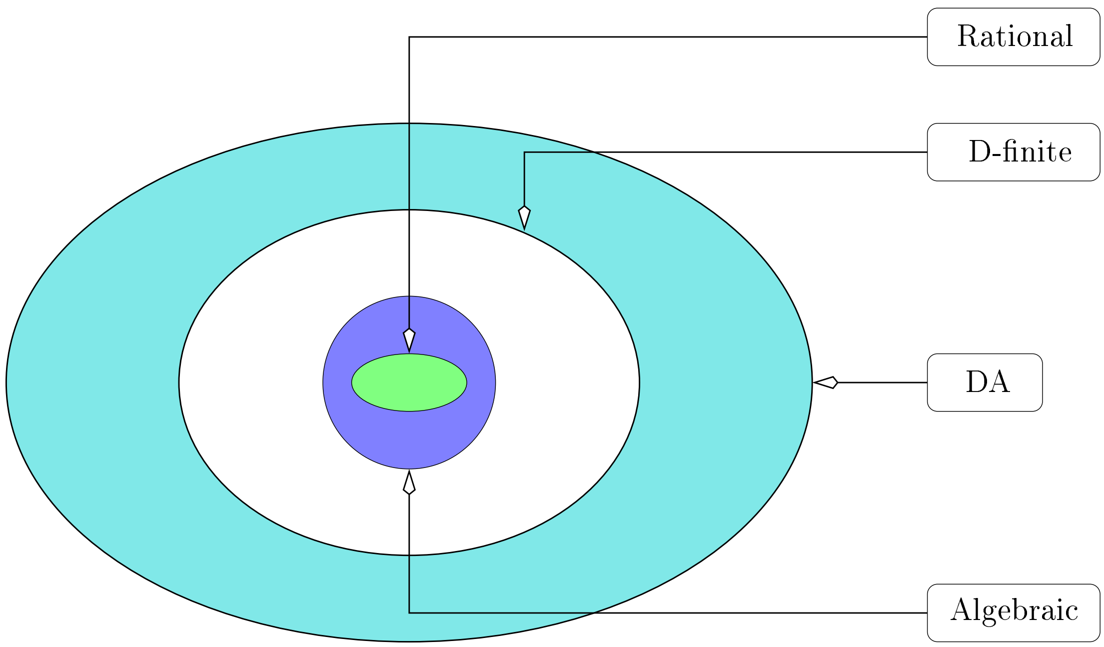



Figure 7.11: A hierarchy of formal power series classes (in one variable). Since $q$-series do not have a clear definition it is difficult to say where they fit into this hierarchy.

### 7.4.5 $q$-series

Members of the ambiguously defined class of power series, $q$-series, appear in many different areas of mathematics and physics[4, 9]. Their definition is far less precise than that of the classes detailed above[3], however, the basic notion of a $q$-series stems from the idea of making *$q$-generalisations* or *$q$-analogues* of basic mathematical objects. For example, $[n]$ denotes the $q$-generalisation of the integer $n$, and is given by

$$[n] = 1 + q + q^2 + \cdots + q^{n-1}. \tag{7.27}$$

In the limit $q \to 1$, the $q$-analogue, $[n]$, returns to $n$. For such simple beginnings, one can make $q$-analogues of many mathematical objects. The $q$-factorial can be defined as

$$[n]! = [1][2]\cdots[n] = 1(1+q)(1+q+q^2)\cdots(1+q+\cdots+q^{n-1}) = \frac{(1-q)(1-q^2)\cdots(1-q^n)}{(1-q)^n}. \tag{7.28}$$

This idea leads to $q$-generalisations of many standard functions. The $q$-Bessel function for example arises in the solution of staircase polygons enumerated by area [108].

Another way in which $q$-series can be "defined", is as the solutions to *$q$-equations*, that is to say equations that involve terms of the form $f(x), f(xq), \ldots, f(xq^k)$. For example, the

[3]When the author first encountered $q$-series, it seemed that their definition was "any series in which one of the variables is denoted by $q$".

partition generating function, $P(x,q) = \prod_{n\geq 1}(1-xq^n)^{-1}$, is the solution to the $q$-linear equation

$$P(x,q) = \frac{1}{1-xq} P(xq,q). \tag{7.29}$$

Due to the imprecise definition of $q$-series it is difficult to compare them with D-finite or differentiably algebraic functions. Since some $q$-series have natural boundaries (such as the partition function), they cannot be D-finite, but it is not clear that every D-finite function is a $q$-series.

## 7.5

# The nature of 2-variable formal power series

How can we determine which class a formal power series falls into, without actually knowing the function? One way is to examine the set of singularities of the series — is it finite, infinite or even dense?

Since all algebraic and rational power series are D-finite power series, we need only consider D-finite power series.

### 7.5.1 Singularities of differentiably-finite power series

The classical theory of linear differential equations implies that a D-finite formal power series in $\mathbb{C}[[x]]$, satisfying a linear ordinary differential equation with polynomial coefficients, has a finite number of singularities. This is a very simple test of "D-finiteness"[4]; if the function has an infinite number of singularities, then it cannot be D-finite. For example, a function such as $f(t) = \frac{1}{\cos(t)}$ is not D-finite.

Alas, the two-variable case is not so simple. One can easily construct power series that are the solutions of differential equations of the form:

$$\left( A_d(x,y)\left(\frac{\partial}{\partial y}\right)^d + A_{d-1}(x,y)\left(\frac{\partial}{\partial y}\right)^{d-1} + \cdots + A_0(x,y)\right) f(x,y) = 0, \tag{7.30}$$

that have an infinite number of singularities, and even have accumulation points.

**Example 7.1.**
Let $f(x,y)$ be a power series in $\mathbb{C}[[x,y]]$, and express $f$ as a power series in $y$, with coefficients in $\mathbb{C}(x)$, *i.e.* $f(x,y) = \sum_{n\geq 0} y^n H_n(x)$.

Consider the example:

$$f(x,y) = \sum_{n\geq 1} \frac{x^n}{1-nx} y^n, \tag{7.31}$$

[4]There must be a better word — "D-finitivity" perhaps?

and so $f(x, y)$ is singular at $x = 1/n$ for all $n \in \mathbb{Z}^+$, however, it is D-finite in $y$; it satisfies the partial-differential equation (in $y$),

$$xy(1 - xy)\left(\frac{\partial f}{\partial y}\right) - (1 - xy)f + xy = 0. \tag{7.32}$$

The coefficients of $y$ (equivalently) satisfy the recurrence:

$$(1 - nx)H_n(x) - x(1 + x - nx)H_{n-1}(x) = 0. \tag{7.33}$$

If we set $y = 1$, then we obtain the series

$$\sum_{n \geq 1} \frac{x^n}{1 - nx}.$$

This defines a formal power series in $x$, that has an infinite number of singularities ($x = 1/n, \forall n \in \mathbb{Z}^+$) and consequently it is not a D-finite power series in $x$.

For $n \geq 0$ let $S_n$ be the set of poles of $H_n(x)$; further let $S = \bigcup_n S_n$. In the above example we have seen that this set $S$ can be an infinite set, and that it can even have accumulation points ($x = 0$ was an accumulation point in $S$); it can be shown that $S$ can only have a finite number of accumulation points.

**Theorem 7.19.** *Let $f(x, y) = \sum_{n \geq 0} y^n H_n(x)$ be a D-finite series in $y$ with rational coefficients, $H_n(x) \in \mathbb{C}(x)$. For $n \geq 0$ let $S_n$ be the set of poles of $H_n(x)$, and let $S = \bigcup_n S_n$. Then $S$ has only a finite number of accumulation points.*

*Proof.* Since $f(x, y)$ is D-finite in $y$ it satisfies an equation of the form of equation (7.30), and by examining powers of $y$ in this equation we see that for $n \geq N_0$ the the coefficients $H_n(x)$ satisfy a recurrence relation with polynomial coefficients (similar to equation (7.24)):

$$P_d(n, x)H_n(x) + P_{d-1}(n, x)H_{n-1}(x) + \cdots + P_0(n, x)H_{n-d}(x) = 0. \tag{7.34}$$

Isolating $H_n(x)$ in this recurrence gives

$$H_n(x) = \frac{-1}{P_d(n, x)}\Big(P_{d-1}(n, x)H_{n-1}(x) + \cdots + P_0(n, x)H_{n-d}(x)\Big), \tag{7.35}$$

where $P_i(n, x)$ is a polynomial in $n$ and $x$. Hence $H_n(x)$ can be written as a rational function with denominator

$$D_n(x) = I(x)\prod_{i=1}^{n} P_d(i, x), \tag{7.36}$$

where $I(x)$ is a polynomial that accounts for the denominators of the $H_n(x)$, $n < N_0$.

Let $a \in \mathbb{C}$ be an accumulation point of $S$. Then for all $\epsilon > 0$, any $\epsilon$-neighbourhood of $a$ must contain a $b \neq a$ such that $\exists n$ for which $D_n(b) = 0$. Consequently there exists a sequence $(n_i, b_i)_{i \geq 0}$, with $n \to \infty$ and $b_i \to a$ such that $P_d(n_i, b_i) = 0$.

Let us expand $P_d(n,x)$ in $n$:

$$P_d(n,x) = \sum_{k=0}^{K} n^k Q_k(x) \qquad \text{with } Q_K(x) \neq 0. \tag{7.37}$$

Then

$$0 = \frac{P_d(n_i,b_i)}{n_i^K} = \sum_{k=0}^{K} n_i^{k-K} Q_k(b_i), \tag{7.38}$$

as $i \to \infty$ the right-hand side becomes $Q_K(a) = 0$.

Hence $a$ must be a zero of the polynomial $Q_K(x)$. This has only finitely many zeros and hence there can only be finitely many accumulation points in $S$. Further we can find these accumulation points by examining the polynomial $P_d(n,x)$. ∎

**Remark.**
We note here that the above theorem is true for any function $f(x,y) = \sum_{n\geq 0} y^n H_n(x)$, where the coefficients $H_n(x)$, with $n \geq N_0$, satisfy a recurrence of the form

$$P(n,x)H_n(x) = Q_n\Big(x, H_0(x), H_1(x), \ldots, H_{n-1}(x)\Big) \tag{7.39}$$

where $P(n,x)$ is a fixed polynomial, and $Q_n(x, H_0(x), \ldots, H_{n-1}(x))$ is a polynomial that may change with $n$. The singularities of $H_n(x)$, are simply those of $H_0(x), \ldots, H_{n-1}(x)$ and the zeros of $P(n,x)$.

### 7.5.2 The nature of the anisotropic SAP generating function.

We are now (finally) ready for the main result.

**Corollary 7.20.** *The generating function $P(x,y)$ of all self-avoiding polygons counted by horizontal and vertical half-perimeters is not D-finite.*

*Proof.* For any $q \in \mathbb{Q}$, there exists $k$ such that $\Psi_k(e^{2i\pi q}) = 0$. Consequently for any $x = e^{2i\pi q}$, with $q \in \mathbb{Q}$, there exists $k$ such the denominator of $H_{3k-2}(x)$ contains $\Psi_k(x)$ and so is singular at $x$. Consequently the singularities of $H_n(x)$ form a set that becomes dense on the unit circle as $n \to \infty$, and so does not have a finite number of accumulation points. Hence the function $P(x,y)$ is not D-finite. ∎

◁ ◁ ◇ ▷ ▷

It should be highlighted that this result does not mean that the isotropic generating function $P(x,x)$ is not D-finite. One can readily construct examples of 2 variable functions that are not D-finite, that become D-finite when $x=y$; for example

$$P(x,y) = \sum_{n\geq 0} \frac{y^n}{(1-x^n)(1-x^{n+1})} \tag{7.40}$$

$$P(x,x) = \frac{x}{(1-x)^2} \tag{7.41}$$

In every solved case, however, we find that "anisotropising" the generating function does not change the (analytic) nature of the generating function. Since we know so little about the nature of the generating functions for unsolved models, we cannot rigorously examine the effect of anisotropisation. There exist non-rigorous Renormalisation Group arguments that imply that anisotropisation should not affect the analytic nature of the generating function [1].

## 7.6

## Conclusions

The fact that self-avoiding polygons enumerated anisotropically are not solvable in terms of D-finite functions, is quite a pessimistic result. It underlines the differences between what we would like to be able to solve, and what we have been able to do.

All the animal models that have been solved to date have had severe topological restrictions, and excluding spiral walks, have solutions that are D-finite functions (*i.e.* the isotropic and anisotropic perimeter generating functions are D-finite). There exists compelling numerical evidence to suggest that the more general animal models that we wish to solve, such as lattice bond animals, are not D-finite [82]. Hence, there exists a marked difference between the models we have been able to solve and the models we wish to solve.

It is also worth noting that common numerical methods used to conjecture solutions from series expansions, such as Differential Approximants and the Maple packages *GFUN* and *MGFUN* [80, 92], search for D-finite solutions, and so will not be able to find solutions to the models we want (at least in the anisotropic case).

In the case of self-avoiding polygons, we have managed to sharpen existing numerical evidence of non D-finiteness into a proof and we hope the techniques that we have developed in previous chapters can be applied to other families of bond animals, in particular general bond animals, lattice trees and directed bond animals.

The application of this method to models of more interest to physicists, such as the susceptibility of the Ising models and the partition function of the $q$-state Potts model, will require much more work, since the partial orders on bond-animals that under-pin all of the above work cannot be applied, at least in their current form.

# CHAPTER 8

# Reciprocity and inversion relations

# 8.1 Introduction

Symmetries are among the most important guiding principles in all of physics and mathematics. It often happens that a problem may be solved by symmetry considerations alone, and even if not, understanding the symmetries of the solution can greatly reduce the amount of work needed to find it.

In this chapter we will examine a type of functional symmetry which is known as "self-reciprocity" to combinatorialists and which is referred to as "inversion relations" in lattice statistical mechanics. In particular we shall demonstrate that one can find examples of functional symmetry in the generating functions of many families of polyominoes.

The inversion relation rose to prominence in statistical mechanics in the early 1980s as the most direct path to the solution of many solvable models [149, 7, 11] and was soon realized to be commonplace in both solved and unsolved models [7, 11, 123]. Let $G(\boldsymbol{x})$ be a generating function or a thermodynamic quantity which depends on a collection of parameters, $\boldsymbol{x}$. An inversion relation is a functional equation

$$G(\boldsymbol{x}) \pm \boldsymbol{x}^{\boldsymbol{\alpha}} G(\boldsymbol{\phi}(\boldsymbol{x})) = \psi(\boldsymbol{x}) \tag{8.1}$$

where $\boldsymbol{\phi}$ and $\psi$ are known functions of $\boldsymbol{x}$. Typically $\boldsymbol{\phi}$ involves taking reciprocals of one or more components of $\boldsymbol{x}$. The inversion relation tightly constrains the function $G$. For some two-dimensional models we also know that $G$ is symmetric under the exchange of horizontal and vertical fugacities, and that it is an analytic function of its arguments. Very often, these three pieces of information taken together uniquely determine the function $G$.

In [145] Stanley presented a general framework for reciprocity results. He established several powerful general conditions under which a generating function will be self-reciprocal. We will use the language and notation of Stanley [145, 147] throughout this chapter.

**Definition 8.1.**
Let $H(y_1, \ldots, y_n)$ be a rational function in the variables $y_i$, with coefficients in $\mathbb{R}$. We say that $H$ is *self-reciprocal* if there exists an $n$-tuple of integers $(\beta_1, \ldots, \beta_n)$ such that

$$H(1/y_1, \ldots, 1/y_n) = \pm y_1^{\beta_1} \ldots y_n^{\beta_n} H(y_1, \ldots, y_n). \tag{8.2}$$

In what follows, we write $\boldsymbol{y}^{\boldsymbol{\beta}} \equiv y_1^{\beta_1} \ldots y_n^{\beta_n}$ and $1/\mathbf{y} \equiv (1/y_1, \ldots, 1/y_n)$. Thus eqn. (8.2) may be concisely expressed as

$$H(1/\boldsymbol{y}) = \pm \boldsymbol{y}^{\boldsymbol{\beta}} H(\boldsymbol{y}).$$

It should be noted that a rational function is self-reciprocal if and only if its numerator and denominator are both self-reciprocal. Also the self-reciprocity of a polynomial amounts to a certain symmetry in its coefficients. Some explicit examples of this are given in subsection 8.3.1.

Let us now demonstrate the relationship between self-reciprocity and inversion relations. Consider the multi-variable generating function

$$\begin{aligned} G(\boldsymbol{x}, \boldsymbol{y}) &= \sum_{\boldsymbol{m},\boldsymbol{n}} C(\boldsymbol{m}, \boldsymbol{n}) x_1^{m_1} \dots x_j^{m_j} y_1^{n_1} \dots y_k^{n_k} \\ &\equiv \sum_{\boldsymbol{m},\boldsymbol{n}} C(\boldsymbol{m}, \boldsymbol{n}) \boldsymbol{x}^{\boldsymbol{m}} \boldsymbol{y}^{\boldsymbol{n}} \end{aligned} \tag{8.3}$$

where $\boldsymbol{m} = (m_1, \dots, m_j)$, $\boldsymbol{x} = (x_1, \dots, x_j)$ and similarly for $\boldsymbol{n}$ and $\boldsymbol{y}$. The summation is over $(j+k)$-tuples of nonnegative integers representing the objects being enumerated. Performing the summation over $\boldsymbol{n}$, we re-express eqn. (8.3) in terms of partial generating functions, $H_{\boldsymbol{m}}(\boldsymbol{y})$ (see also the anisotropic generating functions in chapter 3),

$$G(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{m}} H_{\boldsymbol{m}}(\boldsymbol{y}) \boldsymbol{x}^{\boldsymbol{m}}. \tag{8.4}$$

Now suppose that the partial generating functions are self-reciprocal,

$$H_{\boldsymbol{m}}(1/\boldsymbol{y}) = \pm \boldsymbol{\epsilon}^{\boldsymbol{m}} \boldsymbol{y}^{\boldsymbol{\beta}(\boldsymbol{m})} H_{\boldsymbol{m}}(\boldsymbol{y}), \tag{8.5}$$

where $\boldsymbol{\epsilon}$ is a $j$-tuple of elements in the set $\{-1, 1\}$ which characterises the dependence of the sign on $\boldsymbol{m}$, and where $\boldsymbol{\beta}(\boldsymbol{m})$ depends linearly on $\boldsymbol{m}$:

$$\boldsymbol{\beta}(\boldsymbol{m}) = \mathsf{A}\boldsymbol{m} + \boldsymbol{\alpha}. \tag{8.6}$$

Here, $\mathsf{A} = (a_{\ell,i})$ is a $k \times j$ matrix of integers and $\boldsymbol{\alpha}$ is a $k$-tuple of integers. We can then write

$$G(\boldsymbol{x}, \boldsymbol{y}) \mp \boldsymbol{y}^{-\boldsymbol{\alpha}} G(\boldsymbol{\epsilon x y}^{-\mathsf{A}}, 1/\boldsymbol{y}) = 0, \tag{8.7}$$

where $\boldsymbol{\epsilon x y}^{-\mathsf{A}}$ is the $j$-tuple whose $i^{\text{th}}$ entry is $\epsilon_i x_i \prod_{\ell=1}^{k} y_\ell^{-a_{\ell,i}}$. This is clearly a special case of the inversion relation (8.1). A few comments are in order:

- The right hand side of (8.7) is zero, but in the more general situation some of the partial generating functions, $H_{\boldsymbol{m}}(\boldsymbol{y})$ will fail to be self-reciprocal for certain choices of $\boldsymbol{m}$. If we are fortunate, this will be a small, finite or otherwise controllable set of cases, and we will be able to compute the correction term we need to add to the right hand side explicitly. For many examples in statistical mechanics, this correction term depends on $\boldsymbol{x}$ but not on $\boldsymbol{y}$.

- In all of the cases we shall see below, the denominators of our rational functions will be a product of terms $(1 - \boldsymbol{y}^{\boldsymbol{\alpha}_j})$, which are self-reciprocal. Stanley has proved that this denominator form always holds for certain classes of problems (see Theorem 4.6.11 of ref. [147]). In chapter 3 we have shown that this holds for the denominators of the coefficients of the anisotropic generating function of any dense family of animals.

- It might be asked which of the concepts, inversion or self-reciprocity, is the more general. On one hand, in the derivation of (8.7) the dependence of the exponent $\boldsymbol{\beta}$ on $\boldsymbol{m}$ was assumed to be linear, which may not always hold, implying that reciprocity is more fundamental. On the other hand, the function $\boldsymbol{\phi}$ occurring in (8.1) may in principle be more complicated than $\boldsymbol{x} \to \boldsymbol{\epsilon x y}^{-\mathsf{A}}$, $\boldsymbol{y} \to 1/\boldsymbol{y}$. In this case, the partial generating functions might not be self-reciprocal. An example is provided in section 8.2 by the Potts model, but in the polyomino examples we consider this situation does not arise.

Below is a list (that is certainly not exhaustive) of some methods for finding and proving reciprocity results and inversion relations.

1. If the generating function (or thermodynamic quantity) is known in closed form an inversion relation can be demonstrated directly. As an example, we treat the anisotropic perimeter generating function for directed convex polygons in this manner in section 8.3.

2. For statistical mechanics models which admit a formulation in terms of a family of commuting transfer matrices, a transformation of parameters can often be found which inverts the transfer matrix. The commutativity property then allows the inversion relation to be derived. We review this in detail in section 8.2, with the two-dimensional, zero-field Ising model as primary example.

3. In unsolved models, the transfer matrix will still be invertible and may suggest a possible inversion relation, but the required analyticity property is lacking. Nevertheless, the suggested inversion relation can often be verified by inspection of the partial generating functions up to some finite order in the low-temperature expansion (8.4). The $q > 2$ Potts model inversion relation discussed in section 8.2 was derived this way in ref. [94]. Most of the new results in this chapter were first "discovered" by this method before being re-derived by one of the other methods.

4. The "Temperley methodology" (see chapter 5) can be used to obtain very general reciprocity results for many classes of column-convex polygons. The first step is to derive a functional equation for the generating function using the Hadamard-Temperley method (which can be interpreted as the gluing of an additional column onto the polygon). Step two is to show by induction that gluing an additional column preserves self-reciprocity. This is detailed in section 8.4.

5. If the problem can be rephrased as a system of linear Diophantine equations, whose solutions are subject to certain types of constraints, we may apply self-reciprocity theorems due to Stanley [145]. We have so far succeeded in applying this method only to families of directed polyominoes (section 8.5), but it enables us to treat problems which are impossible, or at least extremely cumbersome, by the method of functional equations (*i.e.* method 4 above).

6. For combinatorial objects with a rational generating function of denominator $\prod_j (1 - \boldsymbol{y}^{\boldsymbol{\alpha}_j})$, one can try to explain self-reciprocity — *i.e.*, the symmetry of the numerator — by interpreting the numerator combinatorially. This has been done by Fédou for a family of objects related to (but distinct from) staircase polygons [60].

In section 8.2 we review the motivation for looking at inversion relations in statistical mechanics and describe the methods used to obtain them. This will be useful for making comparisons with the results obtained later, and for suggesting applications and generalisations of the inversion relations. In section 8.3, we present examples of reciprocity results and inversion relations for polyominoes, and summarise our main new results. The technical heart of the chapter consists of section 8.4 on the Temperley methodology, and section 8.5 on the application of Stanley's results to polyominoes.

## 8.2

## Inversion relations in statistical mechanics

The first use of the inversion relation in statistical mechanics was the solution by Stroganov of certain two dimensional vertex models on the square lattice [149]. Generalisations of Stroganov's models were later solved by the same means by Schultz [141]. Shortly after Stroganov, Baxter used a similar method to solve the hard hexagon model [8] and recognised its broad applicability, giving the eight-vertex and Ising models as examples [7]. Subsequently, a number of authors pointed out that many known solutions to problems in two-dimensional statistical mechanics can be derived easily using the inversion relation method. Among these were Shankar [142], Baxter [10] and Pokrovsky and Bashilov [131].

It is noteworthy that inversion relations are known to hold for models that have *not* been solved. Prominent among such models are the two-dimensional Ising model in a magnetic field whose inversion relation was found by Baxter [7], and the three-dimensional Ising model and non-critical $q$-state Potts model, both of whose inversion relations were found by Jaekel and Maillard [95, 94]. What generally distinguishes solved and unsolved models is the growth rate in the number of poles arising in the partial generating functions in the expansion (8.4), as a function of order. Roughly speaking, a more complicated pole structure implies that the number of parameters needed to specify a given partial generating function is greater, and makes it less likely that an inversion relation can completely determine all of them (in chapter 7 we have shown how such complicated analytic structure implies that the model is not solvable in terms of D-finite functions). Nevertheless, inversion relations are still invaluable in the study of such problems, not least because they provide an independent check on series data.

The fundamental problem of statistical mechanics is to calculate the partition function. Here we consider vertex models defined on a square lattice with each bond coloured with one of $r$ possible colours. Each lattice site makes a contribution to the energy of the system which depends on the colours of the adjacent bonds. This defines an $r^4$-vertex model if all possible colourings are permitted.

Stroganov computed the partition function per site in the thermodynamic limit of several 16- and 81-vertex models. Consider first a finite lattice (on the torus) of $N$ rows and $M$ columns. The partition function can be expressed in terms of the transfer matrix $\mathsf{T}_M$ as

$$Z_{M,N} = \mathrm{Tr}\left[(\mathsf{T}_M)^N\right] \tag{8.8}$$

(see [9, 147]). Here, $\mathsf{T}_M$ is the $r^M \times r^M$ matrix whose $i,j$th entry is the contribution to $Z_{M,N}$ of a single row of $M$ sites connected to the row below by a set of vertical bonds in configuration $i$ and to the row above by a set of vertical bonds in configuration $j$. It depends on the temperature, $T$, and on $r^4$ parameters specifying the vertex energies. In the thermodynamic limit, the partition function per site is given by

$$\kappa = \lim_{M,N\to\infty} (Z_{M,N})^{1/MN} = \lim_{M\to\infty} (\lambda_M)^{1/M} \tag{8.9}$$

where $\lambda_M$ is the largest eigenvalue of $\mathsf{T}_M$ (which is assumed to be non-degenerate).

For simplicity let us consider a family of models whose vertex energies are functions of a single parameter, $b$. The models solved by Stroganov are solvable by virtue of the commutativity of the transfer matrices at different values of this parameter. This implies that the transfer matrix eigenvectors are common to all members of the family, and that the $b$ dependence is only in the eigenvalues. For this reason $b$ is often called the *spectral parameter*. The key observation is that the inverse of the transfer matrix in these models is itself a member of the commuting family, up to a scale factor

$$[\mathsf{T}_M(b)]^{-1} = \psi(b)^{-M}\, \mathsf{T}_M(\phi(b)). \tag{8.10}$$

Acting on the eigenvector corresponding to $\lambda_M(b)$ with both sides of eqn. (8.10) yields the functional equation

$$\kappa(b)\kappa(\phi(b)) = \psi(b). \tag{8.11}$$

It is the commutativity of the transfer matrices for all values of $b$ that allows the analytic continuation of the function $\kappa$ from $b$ to $\phi(b)$. With knowledge of the functions $\psi(b)$ and $\phi(b)$ and using the analyticity of $\kappa(b)$, Stroganov finds a unique solution, thereby reproducing Baxter's results for the symmetric eight vertex and homogeneous ferroelectric models, and obtaining the result for a certain 81-vertex model [149].

As an illustrative example, we review here the derivation by Baxter [7] of Onsager's expression for the partition function of the two-dimensional zero-field Ising model [126]. Let the square lattice be drawn at 45° to the horizontal and let the couplings between nearest neighbours along the two lattice directions be $J$ and $J'$. Define low temperature variables

$$x = e^{-2K}, \quad y = e^{-2K'} \quad \text{with} \quad K = J/k_BT, \quad K' = J'/k_BT, \tag{8.12}$$

where $T$ is the absolute temperature and $k_B$ is Boltzmann's constant. Transfer matrices for different choices of parameters will commute provided they have the same value of $k = (\sinh 2K \sinh 2K')^{-1}$. The transformation which inverts the transfer matrix is

$$K \to K + \frac{i\pi}{2}, \quad K' \to -K', \tag{8.13}$$

and this does not modify the value of $k$. Define the reduced partition function per site by

$$\Lambda(x,y)=\exp(-K-K')\kappa(K,K'). \tag{8.14}$$

Then $\Lambda(x,y)$ obeys the inversion relation

$$\Lambda(x,y)\Lambda(-x,1/y)=1-x^2. \tag{8.15}$$

Note that $\log\Lambda(x,y)-\frac{1}{2}\log(1-x^2)$ has an inversion relation of precisely the form (8.7).

By the horizontal-vertical symmetry of the model, we have

$$\Lambda(x,y)=\Lambda(y,x). \tag{8.16}$$

Inspection of the low temperature expansion leads us to conjecture the form

$$\Lambda(x,y)=1+\sum_{m\geq 1}\frac{P_m(y^2)}{(1-y^2)^{2m-1}}x^{2m}, \tag{8.17}$$

which is very much like the anisotropic generating functions considered in chapter 3. That the coefficient of $x^{2m}$ is a rational function of $y^2$ is apparent from the nature of the low temperature expansion, but that the denominator has such a simple form is not expected on general grounds (such as those described in chapter 3). Presumably it is a consequence of the condition of commuting transfer matrices. Here we take it as a hypothesis. Baxter has shown that the inversion relation (8.15), symmetry (8.16) and the denominator form (8.17) determine $\Lambda(x,y)$ completely. We present his argument in section 8.3.3 where we use it to compute polygon generating functions.

Up till now we have been assuming integrability and in particular we have relied on the property that the transfer matrix and its inverse are both members of some one-parameter commuting family. What about models for which this property doesn't hold? Since analyticity of $\kappa(b)$ breaks down, the step (8.11) in the above derivation is no longer valid. However, it is still possible to obtain an inversion relation by direct analysis of the low-temperature expansion of the partition function to some finite order. As an example, it was shown in ref. [94] that the logarithm of the reduced partition function per site, $G(x,y)=\ln\Lambda(x,y)$, of the $q$-state Potts model satisfies the inversion relation

$$G(x,y)+G\left(-\frac{x}{1+(q-2)x},\frac{1}{y}\right)=\ln\left(\frac{(1-x)(1+(q-1)x)}{1+(q-2)x}\right). \tag{8.18}$$

When $q=2$ this reduces to the Ising model inversion relation (8.15). The inversion relations we will be considering in the remainder of the chapter are derived by analysis of the generating function (analogous to the low temperature expansion) and do not depend on the models being solvable.

An additional new feature is seen in this Potts model example. Neglecting for the moment the nonzero right-hand-side of (8.18), which can be eliminated by a suitable redefinition of $G(x,y)$, we notice that when $q>2$ there is no longer an order-by-order cancellation of the partial generating functions as defined in (8.4), but rather cancellation of combinations of partial generating functions of different orders. However, we may convert to a self-reciprocal form by defining

$$G'(x,y) = G\left(\frac{x}{1-(q-2)x/2}, y\right) \tag{8.19}$$

under which the inversion relation becomes

$$G'(x,y) + G'(-x,1/y) = \ln \frac{1-q^2x^2/4}{1-(q-2)^2x^2/4}. \tag{8.20}$$

In the cases we will examine here, the partial generating functions turn out to be self-reciprocal in the natural variables of the problem. We have not investigated the existence of inversion relations involving more complicated changes of variables.

# 8.3

# Polyomino enumeration and self-reciprocity

## 8.3.1 Self-reciprocity in polyomino enumeration

Before we start, let us define two families of polyominoes that were not defined in chapter 1; we will need them in the work that follows:

**Definition 8.2.** •An ***imperfect staircase polygon*** is a 3-choice polygon that is not a staircase polygon.

•A ***staircase polygon with a staircase hole*** is made up of two staircase polygons one nested inside the other, so that all vertices are of degree 2.

For all of the models of column-convex polygons we have defined the, *anisotropic perimeter and area generating function*,

$$G(x,y,q) = \sum_{m\geq 1}\sum_{n\geq 1}\sum_{a\geq 1} C(m,n,a)x^m y^n q^a \tag{8.21}$$

has been computed exactly (see ref. [23] and references therein). Here $C(m,n,a)$ is the number of polygons of the class with $2m$ horizontal bonds, $2n$ vertical bonds and area $a$. For the classes of convex polygons, the anisotropic perimeter generating function, $G(x,y,1)$ is an algebraic function of the fugacities, $x$ and $y$, whereas the area generating function, $G(1,1,q)$ is a $q$-series. For classes of polygons that are only column-convex, both $G(x,y,1)$ and $G(1,1,q)$ [151] are algebraic, but $G(x,y,q)$ involves $q$-series. A closed-form expression for the three-choice polygon anisotropic perimeter-area generating function is not yet known, but by means of a transfer matrix technique it can be evaluated in polynomial time [42]. The isotropic perimeter generating function, $G(x,x,1)$ is known to have a logarithmic singularity [42], and is therefore not algebraic, but is known to be D-finite. The generating function for staircase polygons with a staircase hole is also not known in closed form. Its properties are expected to be similar in many respects to the generating function for three-choice polygons [91].

We shall be concerned with self-reciprocity properties of the generating functions $H_m(y, q)$ that count polygons of width $m$. We first give two examples.

1. The area generating function for staircase polygons of width 4 is the following rational function [20]:

$$H_4(q) = \frac{q^4(1 + 2q + 4q^2 + 6q^3 + 7q^4 + 6q^5 + 4q^6 + 2q^7 + q^8)}{(1-q)^2(1-q^2)^2(1-q^3)^2(1-q^4)}.$$

   It satisfies

$$H_4(1/q) = -H_4(q),$$

   and is thus self-reciprocal. Observe that the numerator is not only symmetric (due to self-reciprocity), but also unimodal.

2. The (half-)vertical perimeter and area generating function for column-convex polygons of width 3 is the following rational function, which can be derived from the general formula of ref. [23]:

$$\begin{aligned} H_3(y, q) = &\frac{yq^3}{(1-yq)^4(1-yq^2)^2(1-yq^3)} \cdot (y^6q^8 + 4y^5q^7 + 2y^5q^6 + y^4q^6 - y^4q^4 \\ &- 4y^3q^5 - 6y^3q^4 - 4y^3q^3 - y^2q^4 + y^2q^2 + 2yq^2 + 4yq + 1). \end{aligned}$$

   It satisfies

$$H_3(1/y, 1/q) = -\frac{1}{yq^3} H_3(y, q)$$

   and hence is self-reciprocal. Again, observe the symmetry of the coefficients in the numerator.

We shall generalise these results to polygons of any width. Table 8.1 summarises the self-reciprocity properties we have established. Most of them can be proved in various ways. One can, for instance, use a closed form expression of the generating function (Section 8.3.2), or a functional equation that defines it (Section 8.4); one can also encode the polygons by a sequence of numbers constrained by linear Diophantine equations and apply Stanley's general results (Section 8.5). We shall see that the last two methods allow us to introduce many additional parameters and obtain self-reciprocity results that significantly generalise those of Table 8.1.

### 8.3.2 Self-reciprocity via generating functions

When a closed form expression for the generating function of some class of polygons is known, it seems natural to use it to demonstrate an inversion relation. Let us take the example of

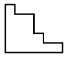
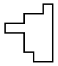
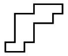
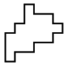
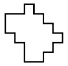
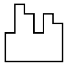
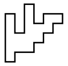
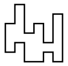
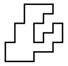
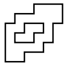
| Class | Picture | Self Reciprocity | Inversion Relation |
|---|---|---|---|
| Ferrers |  | $H_m(1/y,1/q) = (-1)^m y^{m-2} q^{\frac{m^2-3m}{2}} H_m(y,q)$ | $G(x,y) - y^2 G(-x/y, 1/y) = 0$ |
| stack |  | $H_m(1/y,1/q) = -y^{2m-3} q^{m^2-2m} H_m(y,q)$ | $G(x,y) + y^3 G(x/y^2, 1/y) = 0$ |
| staircase |  | $H_m(1/y,1/q) = -y^{m-1} H_m(y,q),\ m \geq 2$ | $G(x,y,q) + yG(x/y, 1/y, 1/q) = -x$ |
| directed convex |  | $H_m(1/y) = -y^{m-2} H_m(y)$ | $G(x,y) + y^2 G(x/y, 1/y) = 0$ |
| convex |  | Not simple | $G(x,y) + y^3 G(x/y, 1/y) = xy - x^3 y \frac{\partial}{\partial x} \frac{1-x+y}{\Delta(x,y)}$ |
| bargraph |  | $H_m(1/y,1/q) = \frac{(-1)^m}{yq^m} H_m(y,q)$ | $G(x,y,q) - yG(-xq, 1/y, 1/q) = 0$ |
| dir. col.-conv. |  | $H_m(1/q) = -\frac{1}{q} H_m(q)$ | $G(x,q) + qG(x, 1/q) = 0$ |
| column-convex |  | $H_m(1/y,1/q) = -\frac{1}{yq^m} H_m(y,q)$ | $G(x,y,q) + yG(xq, 1/y, 1/q) = 0$ |
| three-choice |  | Not simple | $G(x,y,q) + y^2 G(x/y, 1/y, 1/q) = \text{known}$ |
| SC with SC hole |  | Not simple | $G(x,y,q) + y^2 G(x/y, 1/y, 1/q) = \text{known}$ |

Table 8.1: Summary of polyomino inversion relations

the anisotropic perimeter generating function for directed convex polygons, which is known to be [114]

$$G(x, y) = \frac{xy}{\sqrt{\Delta(x, y)}} \tag{8.22}$$

with $\Delta(x, y) = 1-2x-2y-2xy+x^2+y^2 = (1-y)^2\,[1 - x(2 + 2y - x)/(1 - y)^2]$. Expanding this expression in $x$ gives

$$\begin{aligned} G(x, y) &= \sum_{m \geq 1} H_m(y) x^m \\ &= \frac{y}{1-y}x + \frac{y(1+y)}{(1-y)^3}x^2 + \frac{y(1+4y+y^2)}{(1-y)^5}x^3 + \frac{y(1+9y+9y^2+y^3)}{(1-y)^7}x^4 + O(x^5) \end{aligned}$$

which suggests that the partial generating functions, $H_m(y)$ are self-reciprocal, and more precisely, that $H_m(1/y) = -y^{m-2}H_m(y)$. This is equivalent to the inversion relation

$$G(x, y) + y^2 G(x/y, 1/y) = 0, \tag{8.23}$$

which is easily checked from the closed form of the generating function. Note that an explicit expression for $H_m(y)$ is given in [20]. The inversion relations for convex polygons and directed column-convex polygons may also be obtained from the expression for their generating function.

The partial generating functions for directed convex polygons, counted by the area, are not self-reciprocal: for instance, the generating function for width 3 is

$$q^3(1 + 3q + 3q^2 + 2q^3 + q^4)/(1 - q)^2(1 - q^2)^2(1 - q^3).$$

However, many other classes of column-convex polygons have an inversion relation for the full anisotropic perimeter and area generating function. Since these generating functions are also known in closed form they could be derived as above. However more can be shown, namely that there is a self-reciprocity for any parameter which is a linear function of the *vertical heights* in the graph. This very general result will be derived in section 8.4. Likewise, the inversion relations for three-choice polygons and staircase polygons with a staircase hole, given in Table 8.1, are also special cases of more general formulae which will be derived in section 8.5.

### 8.3.3 Using inversion relations to compute generating functions

As in statistical mechanics, the inversion relation and symmetry, and some general assumptions on analyticity of the generating function, are sometimes sufficient to determine the solution completely. In order to have an algorithm for computing a generating function term by term, it is necessary, but not sufficient, to have some property relating terms of different orders. For our purposes this property will always be $x$-$y$ symmetry. Thus we restrict our attention to classes of graphs with $x$-$y$ symmetry, *i.e.*, Ferrers, staircase, directed convex, convex and three-choice polygons, and staircase polygons with a staircase hole. Moreover, we shall only consider the anisotropic perimeter generating function (without area). For the former four classes we will show that the inversion relation provides sufficient additional information to compute the generating function, whereas for the latter two it does not.

The general form of the generating function is

$$G(x,y) = H_1(y)x + H_2(y)x^2 + H_3(y)x^3 + \cdots \tag{8.24}$$

where the partial generating functions, $H_m(y)$ are rational functions

$$H_m(y) = \frac{P_m(y)}{D_m(y)}, \tag{8.25}$$

with $D_m(0) = 1$. The general form of the inversion relation is

$$G(x,y) \pm y^\alpha G(\epsilon x/y, 1/y) = \text{RHS} \tag{8.26}$$

where $\alpha$ is an integer and RHS is zero or some simple function. It is equivalent to a self-reciprocity relation of the form

$$H_m(1/y) \pm \epsilon^m y^{m-\alpha} H_m(y) = \text{RHS}.$$

Whether the inversion relation is sufficient to compute the generating function depends on the value of the exponent $\alpha$ and on the degree of the denominator, $D_m(y)$. The denominator form can often be obtained using the techniques of chapter 3. It can be easily shown that $D_m(y) = (1-y)^m$ for Ferrers graphs, while for other natural families of column-convex polygons, such as staircase, directed convex and convex, $D_m(y) = (1-y)^{2m-1}$.

For the three-choice polygons and staircase polygons with a staircase hole it can be shown that the denominators are

$$D_m(y) = \begin{cases} (1-y)^{2m-1}(1+y)^{2m-7} & m \text{ even} \\ (1-y)^{2m-1}(1+y)^{2m-8} & m \text{ odd}. \end{cases} \tag{8.27}$$

We assume that in general we know the denominator form either empirically or by rigorous proof, and that $D_m(y)$ is of degree $d_m$.

Now we proceed inductively, following Baxter [7]. If we have already computed the coefficient functions $H_1(y), \ldots, H_{m-1}(y)$ in the expansion (8.24) and if $x$-$y$ symmetry holds, we also know the coefficients of $y, y^2, \ldots, y^{m-1}$ in the expansion of $G(x,y)$. In particular, we can compute the coefficients of $y, y^2, \ldots, y^{m-1}$ in the numerator polynomial $P_m(y)$. In order to obtain the unknown coefficients of $P_m(y)$, we must be able to express them in terms of the known ones by means of the inversion relation. Writing $P_m(y) = \sum_k a_k y^k$, and using $D_m(1/y) = \pm y^{-d_m} D_m(y)$, the inversion relation fixes the value of the combinations of coefficients, $a_k \pm a_\ell$, with $k + \ell = \alpha + d_m - m$. Hence the determination of all the coefficients $a_k$ is possible if and only if the following arithmetic condition holds:

$$d_m < 3m - \alpha. \tag{8.28}$$

This condition is seen to hold for all the classes of convex polygons we have looked at, since $d_m \leq 2m-1$, but not for three-choice polygons or staircase polygons with a staircase hole, since $d_m \sim 4m$.

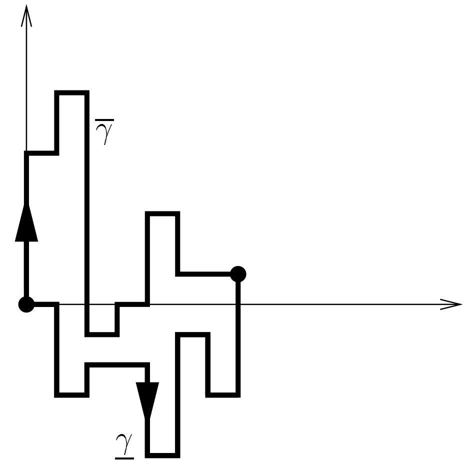



Figure 8.1: A column-convex polygon

## 8.4 Self-reciprocity via Temperley methodology

We consider column-convex polygons as pairs of partially directed paths having the same endpoints, as indicated in Figure 8.1.

Let $P$ be a column-convex polygon of width $m$. For $0 \leq i \leq m$, we denote by $\overline{N}_i$ (resp. $\overline{S}_i$) the number of north (resp. south) steps in the top path $\overline{\gamma}$ at abscissa $i$. For $0 \leq i \leq m$, we denote by $\underline{N}_i$ (resp. $\underline{S}_i$) the number of north (resp. south) steps of the bottom path $\underline{\gamma}$ at abscissa $i$. We choose the end points of the paths in such a way that $\underline{N}_0 = \underline{S}_0 = \overline{N}_m = \overline{S}_m = \overline{S}_0 = \underline{S}_m = 0$. Note that

$$\sum_{k=0}^{m} (\overline{N}_k + \underline{S}_k - \overline{S}_k - \underline{N}_k) = 0.$$

We notice that all standard statistics are *linear* functions of the $\overline{N}_i, \overline{S}_i, \underline{N}_i$ and $\underline{S}_i$. For instance, the vertical perimeter of the polygon is

$$\begin{aligned} 2n &= \sum_{k=0}^{m} (\overline{N}_k + \underline{S}_k + \overline{S}_k + \underline{N}_k) \\ &= 2 \sum_{k=0}^{m} (\overline{N}_k + \underline{S}_k). \end{aligned} \tag{8.29}$$

The height of the $i^{\text{th}}$ column of the polygon is, for $1 \leq i \leq m$,

$$h_i = \sum_{k=0}^{i-1} (\overline{N}_k + \underline{S}_k - \overline{S}_k - \underline{N}_k),$$

and the area of the polygon is

$$a = \sum_{k=0}^{m} (m-k)(\overline{N}_k + \underline{S}_k - \overline{S}_k - \underline{N}_k). \tag{8.30}$$

**Theorem 8.1.** *Let $\mathcal{P}$ be one of the following sets: Ferrers diagrams, stacks (drawn vertically[1]), staircase polygons, bargraphs, column-convex polygons. Let $\mathcal{P}_m$ be the subset of $\mathcal{P}$ containing all polygons of width $m$. Let $F_m$ be the generating function for polygons in the set $\mathcal{P}_m$:*

$$F_m(\overline{\boldsymbol{y}}, \overline{\boldsymbol{z}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}}) = \sum_{P \in \mathcal{P}_m} \overline{\boldsymbol{y}}^{\overline{\mathbf{N}}} \overline{\boldsymbol{z}}^{\overline{\mathbf{S}}} \underline{\boldsymbol{y}}^{\underline{\mathbf{N}}} \underline{\boldsymbol{z}}^{\underline{\mathbf{S}}}.$$

*Then $F_m$ is a rational function, and it is self-reciprocal:*

$$F_m(1/\overline{\boldsymbol{y}}, 1/\overline{\boldsymbol{z}}, 1/\underline{\boldsymbol{y}}, 1/\underline{\boldsymbol{z}}) = C_m F_m(\overline{\boldsymbol{y}}, \overline{\boldsymbol{z}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}}), \tag{8.31}$$

*with*

$$C_m = \begin{cases} \dfrac{(-1)^m \underline{y}_m^{m-2}}{\overline{y}_0} \displaystyle\prod_{i=1}^{m-1} \overline{y}_i & \textit{for Ferrers graphs,} \\[2ex] -\dfrac{\underline{y}_m^{2m-3}}{\overline{y}_0} \displaystyle\prod_{i=1}^{m-1} \overline{y}_i \underline{z}_i & \textit{for stacks,} \\[2ex] -\displaystyle\prod_{i=1}^{m-1} \underline{y}_i \overline{y}_i & \textit{for staircase polygons } (m \geq 2), \\[2ex] \dfrac{(-1)^m}{\overline{y}_0 \underline{y}_m} & \textit{for bargraphs,} \\[2ex] -\dfrac{1}{\overline{y}_0 \underline{y}_m} & \textit{for column-convex polygons.} \end{cases}$$

The proof of the theorem is based on the Temperley method (see chapter 5). Here we provide only the proof for column-convex polygons, since the others are very similar.

We start by showing that the partial generating functions for column-convex polygons, $V_m(\overline{\boldsymbol{y}}, \overline{\boldsymbol{z}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}})$, can be computed recursively using the Hadamard Temperley method.

**Proposition 8.2.** *Let $V_m(\overline{\boldsymbol{y}}, \overline{\boldsymbol{z}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}})$ be the generating function for column-convex polygons of width $m$. Let us denote it, for the sake of simplicity, $V_m(\underline{y}_m)$. Then the series $V_m(\underline{y}_m)$ can be defined inductively by:*

$$V_1(\underline{y}_1) = \frac{\overline{y}_0 \underline{y}_1}{1 - \overline{y}_0 \underline{y}_1}$$

[1]Rotate the stack in figure 2.2 by 90°.

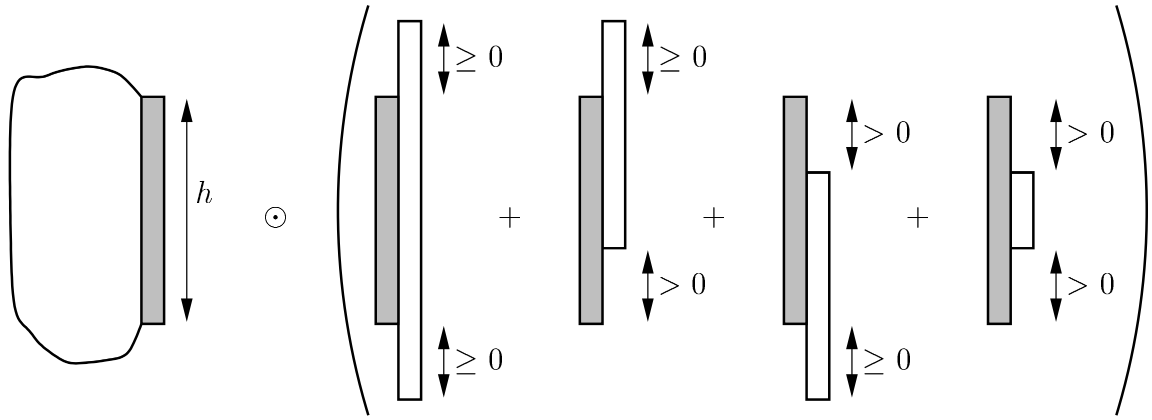



Figure 8.2: Construction of column-convex polygon by Hadamard products.

*and*

$$
\begin{aligned}
V_{m+1}(\underline{y}_{m+1}) = {} & \frac{(1-\underline{y}_m\underline{z}_m)(1-\overline{y}_m\overline{z}_m)V_m(\underline{y}_{m+1})}{(1-\underline{y}_{m+1}\overline{y}_m)(1-\underline{y}_{m+1}\underline{z}_m)(1-\underline{y}_{m+1}^{-1}\underline{y}_m)(1-\underline{y}_{m+1}^{-1}\overline{z}_m)} \\
& + \frac{(\overline{z}_m-\underline{y}_{m+1}\underline{y}_m\underline{z}_m)V_m(\overline{z}_m)}{(1-\underline{y}_{m+1}\underline{z}_m)(1-\underline{y}_{m+1}^{-1}\overline{z}_m)(\underline{y}_m-\overline{z}_m)} + \frac{(\underline{y}_m-\underline{y}_{m+1}\overline{y}_m\overline{z}_m)V_m(\underline{y}_m)}{(1-\underline{y}_{m+1}\overline{y}_m)(1-\underline{y}_{m+1}^{-1}\underline{y}_m)(\overline{z}_m-\underline{y}_m)}.
\end{aligned}
$$

**Proof.** The basic idea is to build a polygon of width $m+1$ block-by-block. The seed blocks are simply single columns, while the building blocks are column-convex polygons of width 2.

The expression for $V_1(\underline{y}_1)$ is obvious (it is the generating function of the seed blocks). We build a column-convex polygon of width $m+1$ as follows: we take a polygon of width $m$ and match its rightmost column with the leftmost column of a column-convex polygon of width 2 (the building blocks). This is illustrated by Figure 8.2, which shows that

$$
V_{m+1}(\underline{y}_{m+1}) = V_m(t)\odot_t R(t), \tag{8.32}
$$

where $R(t)$ is the generating function of the underweighted building blocks:

$$
\begin{aligned}
R(t) = {} & \frac{t\underline{y}_{m+1}}{1-t\underline{y}_{m+1}}\cdot\frac{1}{1-\underline{y}_{m+1}\overline{y}_m}\cdot\frac{1}{1-\underline{y}_{m+1}\underline{z}_m} + \frac{t\underline{y}_{m+1}}{1-t\underline{y}_{m+1}}\cdot\frac{1}{1-\underline{y}_{m+1}\overline{y}_m}\cdot\frac{t\underline{y}_m}{1-t\underline{y}_m} \\
& + \frac{t\underline{y}_{m+1}}{1-t\underline{y}_{m+1}}\cdot\frac{t\overline{z}_m}{1-t\overline{z}_m}\cdot\frac{1}{1-\underline{y}_{m+1}\underline{z}_m} + \frac{t\underline{y}_{m+1}}{1-t\underline{y}_{m+1}}\cdot\frac{t\overline{z}_m}{1-t\overline{z}_m}\cdot\frac{t\underline{y}_m}{1-t\underline{y}_m}.
\end{aligned}
$$

In order to determine the coefficient $r_h$ of $t^h$ in $R(t)$, we expand $R(t)$ in partial fractions of $t$:

$$
\begin{aligned}
R(t) =& \frac{\underline{z}_m \overline{y}_m \underline{y}_{m+1}^2}{(1-\underline{y}_{m+1}\overline{y}_m)(1-\underline{y}_{m+1}\underline{z}_m)} \\
&+ \frac{(1-\underline{y}_m\underline{z}_m)(1-\overline{y}_m\overline{z}_m)}{(1-\underline{y}_{m+1}\overline{y}_m)(1-\underline{y}_{m+1}\underline{z}_m)(1-\underline{y}_{m+1}^{-1}\underline{y}_m)(1-\underline{y}_{m+1}^{-1}\overline{z}_m)} \cdot \frac{1}{1-t\underline{y}_{m+1}} \\
&+ \frac{(\overline{z}_m - \underline{y}_{m+1}\underline{y}_m\underline{z}_m)}{(1-\underline{y}_{m+1}\underline{z}_m)(1-\underline{y}_{m+1}^{-1}\overline{z}_m)(\underline{y}_m-\overline{z}_m)} \cdot \frac{1}{1-t\overline{z}_m} \\
&+ \frac{(\underline{y}_m - \underline{y}_{m+1}\overline{y}_m\overline{z}_m)}{(1-\underline{y}_{m+1}\overline{y}_m)(1-\underline{y}_{m+1}^{-1}\underline{y}_m)(\overline{z}_m-\underline{y}_m)} \cdot \frac{1}{1-t\underline{y}_m}.
\end{aligned}
$$

Note that $V_m(0) = 0$. We now combine eqn. (8.32) with the above expression for $R(t)$ to obtain the announced expression for $V_{m+1}(\underline{y}_{m+1})$. ∎

**Proof of Theorem 8.1.** Induction on $m$ using the functional equation of Proposition 8.2 shows that the partial generating functions for column-convex polygons satisfy

$$V_m(1/\overline{\boldsymbol{y}}, 1/\overline{\boldsymbol{z}}, 1/\underline{\boldsymbol{y}}, 1/\underline{\boldsymbol{z}}) = -\frac{1}{\overline{y}_0\underline{y}_m} V_m(\overline{\boldsymbol{y}}, \overline{\boldsymbol{z}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}}).$$

∎

We proceed similarly for the other families: the functional equation is obtained by setting some of the variables $\overline{y}_i, \underline{y}_i, \overline{z}_i$ and $\underline{z}_i$ to 0. Then, an inductive argument yields the self-reciprocity result.

It would be tempting to write that the self-reciprocity of $V_m$ implies the self-reciprocity of, say, the generating function for staircase polygons, obtained by setting $\underline{z}_i$ and $\overline{z}_i$ to 0 in $V_m$. But replacing a variable by 0 in a self-reciprocal rational function might break the self-reciprocity: for instance, take $P(y_1, y_2) = 1 + y_1 + 2y_1^2 + 2y_2 + y_1y_2 + y_1^2y_2$. Then

$$P(1/y_1, 1/y_2) = \frac{1}{y_1^2y_2} P(y_1, y_2),$$

but $P(y_1, 0) = 1 + y_1 + 2y_1^2$ is *not* self-reciprocal.

However, the following simple lemma gives a useful stability property of self-reciprocal rational functions.

**Lemma 8.3.** *Let $F(y_1, \ldots, y_n)$ be a self-reciprocal rational function. Let $\mathsf{A}$ be an $m \times n$ integer matrix. Let $\boldsymbol{u} = (u_1, \ldots, u_m)$, and define $\boldsymbol{u}^{\mathsf{A}}$ to be the $n$-tuple whose $i^{th}$ coordinate is $\prod_k u_k^{a_{ki}}$. Then the series $G(\boldsymbol{u}) = F(\boldsymbol{u}^{\mathsf{A}})$, if defined, is self-reciprocal in the variables $u_i$. More precisely, if $F(1/\boldsymbol{y}) = \pm\boldsymbol{y}^{\beta}F(\boldsymbol{y})$ then $G(1/\boldsymbol{u}) = \pm\boldsymbol{u}^{\mathsf{A}\beta}G(\boldsymbol{u})$.*

From Theorem 8.1 and Lemma 8.3 we immediately deduce:

**Corollary 8.4.** *For any of the sets $\mathcal{P}$ listed in Theorem 8.1, and any statistics on column-convex polygons that can be expressed as* linear *functions of the quantities $\overline{\mathbf{N}}, \overline{\mathbf{S}}, \underline{\mathbf{N}}, \underline{\mathbf{S}}$, the generating function for polygons in the set $\mathcal{P}_m$ according to these statistics is a self-reciprocal rational function.*

This corollary allows us to complete the top part of Table 8.1. Let us, for instance, derive the inversion relation satisfied by the tri-variate generating function $G(x, y, q)$ for column-convex polygons, taking into account the usual parameters of interest: horizontal and vertical half-perimeters (variables $x$ and $y$), and area (variable $q$).

Eqns. (8.29) and (8.30) express the vertical perimeter and the area in terms of the quantities $\overline{\mathbf{N}}, \underline{\mathbf{N}}, \overline{\mathbf{S}}$ and $\underline{\mathbf{S}}$. They imply that the (half) vertical perimeter and area generating function $H_m(y, q)$ for column-convex polygons of width $m$ is

$$H_m(y, q) = V_m(\overline{\boldsymbol{y}}, \overline{\boldsymbol{z}}, \underline{\boldsymbol{y}}, \underline{\boldsymbol{z}})$$

where $\overline{y}_k = \underline{z}_k = yq^{m-k}$ and $\underline{y}_k = \overline{z}_k = q^{-(m-k)}$. Theorem 8.1 then gives

$$H_m(1/y, 1/q) = -\frac{1}{yq^m} H_m(y, q),$$

which implies

$$G(x, y, q) + yG(xq, 1/y, 1/q) = 0. \tag{8.33}$$

Note that in the first two self-reciprocity relations of Table 1, the exponent of $q$ depends quadratically on the width. For this reason, they only yield an inversion relation for $q = 1$.

8.5

# Self-reciprocity via Stanley's general results

## 8.5.1 Linear homogeneous Diophantine systems

Stanley has analysed the situation where the objects to be counted correspond to integer solutions of a system of linear equations with integer coefficients (linear Diophantine system) subject to a set of constraints. He has established certain conditions under which reciprocity relations will hold between two combinatorics problems defined by the same linear Diophantine system but by different sets of constraints, and also conditions under which the solution to a given problem will be self-reciprocal [145, 147].

Consider the linear homogeneous Diophantine system (LHD-system),

$$\boldsymbol{\Phi}\boldsymbol{\alpha} = \mathbf{0} \tag{8.34}$$

in the vector of unknowns $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_s)$ where $\boldsymbol{\Phi}$ is a matrix of integers having $p$ rows and $s$ columns and $\mathbf{0}$ a $p$-tuple of zeros. The corank $\kappa$ of the system is defined to be $(s - \mathrm{rank}(\boldsymbol{\Phi}))$. For a linearly independent system this simplifies to $\kappa = s - p$. Let $S$ be a set of integer solutions to eqn. (8.34). We define the generating function, $S(\boldsymbol{y})$, as the formal power series

$$S(\boldsymbol{y}) = \sum_{\boldsymbol{\alpha} \in S} \boldsymbol{y}^{\boldsymbol{\alpha}} \tag{8.35}$$

where $\boldsymbol{y} = (y_1, \ldots, y_s)$ is a vector of fugacities associated with the unknowns in eqn. (8.34).

In our applications, we find two types of constraints on the unknowns, $\alpha_j$. Certain of the unknowns, $\alpha_j$, are required to be strictly positive while the rest are required to be non-negative. Conveniently, precisely these kinds of constraints have been treated by Stanley. Let the unknowns be $\boldsymbol{\alpha} = (\boldsymbol{\gamma}, \boldsymbol{\delta}) \equiv \boldsymbol{\gamma} \oplus \boldsymbol{\delta}$ where $\boldsymbol{\gamma}$ is an $n$-tuple and $\boldsymbol{\delta}$ is an $(s-n)$-tuple. Likewise let $\boldsymbol{y} = (\boldsymbol{u}, \boldsymbol{v})$. In what follows we write "$\boldsymbol{\delta} > \mathbf{0}$" to mean that all coordinates of $\boldsymbol{\delta}$ are positive.

**Proposition 8.5.** *Let $E$ be the set of integer solutions, $(\boldsymbol{\gamma}, \boldsymbol{\delta})$, to a linear homogeneous Diophantine system of corank $\kappa$, such that $\boldsymbol{\gamma} \geq \mathbf{0}$ and $\boldsymbol{\delta} > \mathbf{0}$. Let $\overline{E}$ be the set of solutions to the same system with $\boldsymbol{\gamma} > \mathbf{0}$ and $\boldsymbol{\delta} \geq \mathbf{0}$. If the system has an integer solution, $(\boldsymbol{\gamma}, \boldsymbol{\delta})$ such that $\boldsymbol{\gamma} > 0$ and $\boldsymbol{\delta} < 0$, then $E(\boldsymbol{u}, \boldsymbol{v})$ and $\overline{E}(\boldsymbol{u}, \boldsymbol{v})$ are rational functions obeying the reciprocity relation*

$$\overline{E}(\boldsymbol{u}, \boldsymbol{v}) = (-1)^{\kappa} E(1/\boldsymbol{u}, 1/\boldsymbol{v}). \tag{8.36}$$

*Proof.* This is Proposition 8.3 of ref. [145] and the proof is given there. ∎

Proposition 8.5 can be specialised to obtain a self-reciprocity condition, which will be our main tool in the derivations to follow.

**Corollary 8.6.** *A sufficient condition for the function $E(\boldsymbol{u}, \boldsymbol{v})$ to be self-reciprocal is that the linear homogeneous Diophantine system has the solution $(\boldsymbol{\gamma}, \boldsymbol{\delta}) = (\mathbf{1}, -\mathbf{1})$. In this case*

$$E(1/\boldsymbol{u}, 1/\boldsymbol{v}) = (-1)^{\kappa} \frac{\boldsymbol{u}^{\mathbf{1}}}{\boldsymbol{v}^{\mathbf{1}}} E(\boldsymbol{u}, \boldsymbol{v}). \tag{8.37}$$

*Proof.* Since the solution $(\mathbf{1}, -\mathbf{1})$ satisfies the conditions of Proposition 8.5, the reciprocity result (8.36) holds. The result follows immediately from the shift $(\boldsymbol{\gamma}, \boldsymbol{\delta}) \to (\boldsymbol{\gamma} + \mathbf{1}, \boldsymbol{\delta} - \mathbf{1})$ which establishes a bijection between the sets $E$ and $\overline{E}$. ∎

Since the conditions of the corollary are sufficient but not necessary, it is often possible to find a perfectly valid LHD-system describing a given self-reciprocal generating function, $E(\boldsymbol{u}, \boldsymbol{v})$, which does *not* admit the solution $(\mathbf{1}, -\mathbf{1})$. Hence we are faced with the problem of finding a suitable LHD-system which satisfies the corollary. A useful heuristic is to start with an LHD-system in many unknowns, and selectively eliminate those unknowns whose constraints are not independent of the constraints on the other unknowns. In all the cases we will consider, the resulting system will satisfy the conditions of Corollary 8.6. We do not justify this heuristic here. In a paper subsequent to ref. [145], Stanley [146] develops a more comprehensive theory which overcomes these difficulties, and which additionally gives "correction" terms for systems in which self-reciprocity fails to hold. We have not yet explored the ramifications of this theory.

Before applying the above result to staircase polygons with a staircase hole or to three-choice polygons, we use it to derive the reciprocity relation for ordinary staircase polygons of width three. This will serve to illustrate all the basic ingredients of the method.

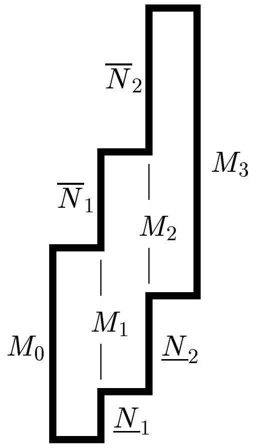



Figure 8.3: Staircase polygon of width three

**Example 8.1.**
Staircase polygons of width three can be characterised by the heights $\overline{N}_1$, $\overline{N}_2$, $\underline{N}_1$, $\underline{N}_2$, $M_0$, $M_1$, $M_2$ and $M_3$, as shown in Figure 8.3. Decomposing the polygon into three columns, and imposing the condition that each column be as high on the left as it is on the right, we obtain the linear homogeneous Diophantine system

$$M_0 - M_1 - \underline{N}_1 = 0 \tag{8.38a}$$

$$M_1 + \overline{N}_1 - M_2 - \underline{N}_2 = 0 \tag{8.38b}$$

$$M_2 + \overline{N}_2 - M_3 = 0. \tag{8.38c}$$

All heights must be nonnegative, but the self-avoidance condition additionally requires that the $M_j$ be positive. The constraints $M_0 > 0$ and $M_3 > 0$ are actually redundant, since they follow from eqns. (8.38a, 8.38c) and the constraints on the remaining unknowns, namely

$$\begin{aligned} \overline{N}_1, \overline{N}_2, \underline{N}_1, \underline{N}_2 &\geq 0 \\ M_1, M_2 &> 0. \end{aligned} \tag{8.39}$$

Since the constraints on $M_0$ and $M_3$ play no role in the solution, we are free to eliminate these unknowns, and it turns out to be necessary to do so in order to apply Corollary 8.6. We are left with the single equation (8.38b) in the six independent unknowns $\boldsymbol{\gamma} = (\overline{N}_1, \overline{N}_2, \underline{N}_1, \underline{N}_2)$ and $\boldsymbol{\delta} = (M_1, M_2)$. Let us associate to the unknown $\overline{N}_i$ (resp. $\underline{N}_i$, $M_i$) the fugacity $\overline{y}_i$ (resp. $\underline{y}_i$, $z_i$).

Let $E'$ be the set of solutions to eqn. (8.38b) subject to the constraints $\boldsymbol{\gamma} \geq 0$ and $\boldsymbol{\delta} > 0$. Since $\boldsymbol{\gamma} = \mathbf{1}$, $\boldsymbol{\delta} = -\mathbf{1}$ is a solution to eqn. (8.38b), Corollary 8.6 tells us that $E'(\overline{\boldsymbol{y}}, \underline{\boldsymbol{y}}, \boldsymbol{z})$ is self-reciprocal,

$$E'(1/\overline{\boldsymbol{y}}, 1/\underline{\boldsymbol{y}}, 1/\boldsymbol{z}) = -\frac{\overline{y}_1 \overline{y}_2 \underline{y}_1 \underline{y}_2}{z_1 z_2} E'(\overline{\boldsymbol{y}}, \underline{\boldsymbol{y}}, \boldsymbol{z}). \tag{8.40}$$

Equations (8.38a, 8.38c) imply that to account for the dependent parameters $M_0$ and $M_3$, we make the substitutions $z_1 \to z_0 z_1$, $\underline{y}_1 \to z_0 \underline{y}_1$, $z_2 \to z_2 z_3$ and $\overline{y}_2 \to z_3 \overline{y}_2$. Applying

Lemma 8.3, we obtain for the set $E$ of nonnegative solutions to (8.38a, 8.38b, 8.38c) such that $\mathbf{M} > \mathbf{0}$:

$$E(1/\overline{\boldsymbol{y}}, 1/\underline{\boldsymbol{y}}, 1/\boldsymbol{z}) = -\frac{\overline{y}_1\overline{y}_2\underline{y}_1\underline{y}_2}{z_1 z_2} E(\overline{\boldsymbol{y}}, \underline{\boldsymbol{y}}, \boldsymbol{z}). \tag{8.41}$$

Notice that reintroducing the dependent unknowns has not changed the constant factor. This feature holds as well in the more complicated models we will look at. The result (8.41) may be verified by inspection of the explicit expression for the generating function

$$E(\overline{\boldsymbol{y}}, \underline{\boldsymbol{y}}, \boldsymbol{z}) = \frac{z_0 z_1 z_2 z_3 (1 - \overline{y}_1 z_0 z_1 z_2 z_3 \underline{y}_2)}{(1 - z_0 \underline{y}_1)(1 - z_0 z_1 \underline{y}_2)(1 - \overline{y}_1 \underline{y}_2)(1 - z_0 z_1 z_2 z_3)(1 - \overline{y}_1 z_2 z_3)(1 - \overline{y}_2 z_3)}.$$

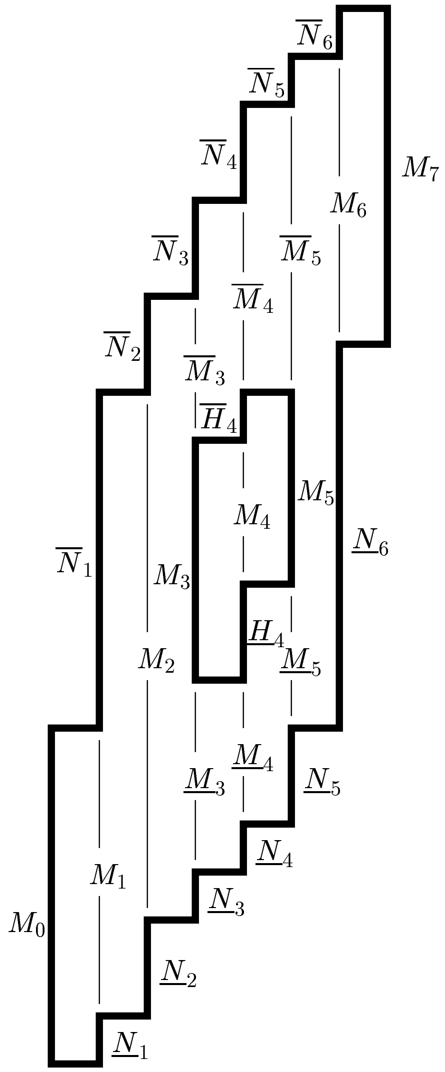
### 8.5.2 Applications

We now apply the methods of section 8.5.1 to staircase polygons with a staircase hole and to three-choice polygons. All the essential steps have already been seen in the derivation of the reciprocity result for staircase polygons of width three. They are

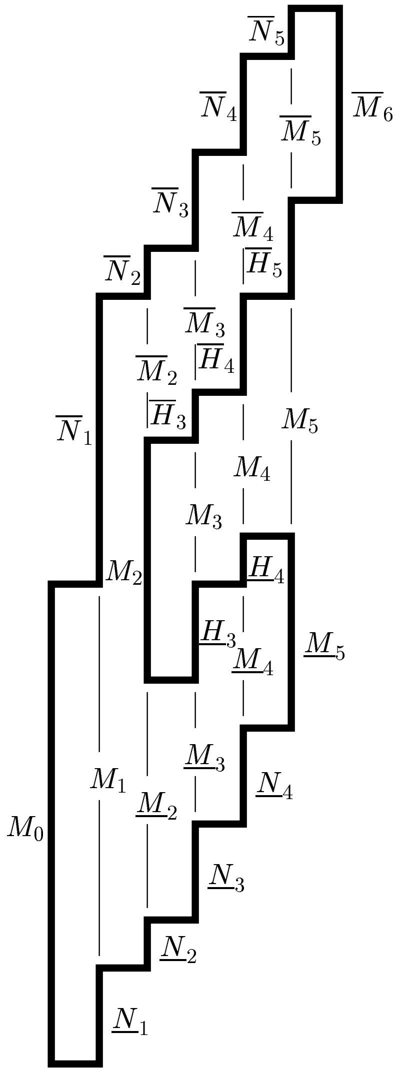
1. Set up a linear homogeneous Diophantine system by decomposing the polyomino into width one rectangles and imposing the condition that the left and right sides of each rectangle have equal height.
2. Sort the unknowns into three classes, $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$ and $\boldsymbol{\tau}$, according to whether they are constrained to be nonnegative, constrained to be positive or constrained by conditions on the other unknowns.
3. Use Gaussian elimination to remove the unknowns in $\boldsymbol{\tau}$.
4. Verify that the resulting system is solved by setting all members of $\boldsymbol{\gamma}$ equal to one and all members of $\boldsymbol{\delta}$ equal to minus one. Apply Corollary 8.6 to obtain the self-reciprocity result for the reduced system.
5. Reintroduce the unknowns in the set $\boldsymbol{\tau}$ by means of Lemma 8.3.

We can define three widths for a staircase polygon with a staircase hole: the distance from the left edge of the figure to the left edge of the hole, $k$, the distance from the left edge of the figure to the right edge of the hole, $\ell$, and the width of the entire figure, $m$. Note that $0 < k < \ell < m$. Recall that for staircase polygons the figures of width one were an exceptional case which did not obey the same reciprocity result as the general case. The staircase polygons with a hole of width one are also an exceptional case, which we must exclude. We thus impose the additional condition $\ell - k > 1$. A figure with given $k$, $\ell$ and $m$ is specified by the following dimensions, as shown in Figure 8.4(a),

1. heights $\underline{N}_j$ and $\overline{N}_j$ of the lower and upper perimeter segments of the polygon, $1 \le j \le m-1$,
2. interior heights $M_j$ to the left and right of, and within, the hole, $0 \le j \le m$,




(a) Staircase polygon with staircase hole, $(k, \ell, m) = (3, 5, 7)$




(b) Three-choice polygon, $(k, \ell, m) = (2, 6, 5)$

Figure 8.4: Labels for polyomino vertical heights

3. heights $\underline{H}_j$ and $\overline{H}_j$ of the lower and upper perimeter segments of the hole, $k+1 \le j \le \ell-1$,

4. interior heights $\underline{M}_j$ and $\overline{M}_j$ below and above the hole, $k \le j \le \ell$.

Three-choice polygons can be regarded as staircase polygons with a hole which doesn't close. The width $k$ has the same meaning as above, $\ell$ denotes the ultimate horizontal extent of the branch of the figure above the hole, and $m$ denotes the ultimate horizontal extent of the branch below the hole. Note that $\ell \ge k$ and $m > k$. Again an exceptional case, $m = k+1$, must be excluded. Hence we impose the restriction $m > k+1$. The labelling of the vertical dimensions follows, with a few obvious modifications, the pattern of staircase polygons with a staircase hole and is shown in Figure 8.4(b). In particular, the heights $M_j$ within the hole are defined only for $j \le \min(\ell, m)$. When $\ell = k$ the unknowns $\overline{M}_j$ and $\overline{H}_j$ do not appear. This special case is treated separately.

As in the case of column-convex polygons, the standard statistics are linear in these heights. The (half-)vertical perimeter for staircase polygons with a staircase hole is given by

$$n = M_0 + \sum_{j=1}^{m-1} \overline{N}_j + M_k + \sum_{j=k+1}^{\ell-1} \overline{H}_j \tag{8.42}$$

and the area is given by

$$a = M_0 + \sum_{j=1}^{k-1}(M_j + \overline{N}_j) + \sum_{j=k+1}^{\ell}(\underline{N}_j + \underline{M}_j) + \sum_{j=k}^{\ell-1}(\overline{M}_j + \overline{N}_j) + \sum_{j=\ell+1}^{m-1}(\underline{N}_j + M_j) + M_m. \tag{8.43}$$

In what follows, we associate to the unknowns $\overline{N}_i$ (resp. $\underline{N}_i$, $\overline{H}_i$, $\underline{H}_i$, $M_i$, $\overline{M}_i$, $\underline{M}_i$) the fugacities $\overline{y}_i$ (resp. $\underline{y}_i, \overline{w}_i, \underline{w}_i, z_i, \overline{z}_i, \underline{z}_i$).

**Proposition 8.7.** *Let $E_{k,\ell,m}(\overline{\boldsymbol{y}}, \underline{\boldsymbol{y}}, \overline{\boldsymbol{w}}, \underline{\boldsymbol{w}}, \boldsymbol{z}, \overline{\boldsymbol{z}}, \underline{\boldsymbol{z}})$ be the generating function for staircase polygons with a staircase hole where $k$, $\ell$ and $m$ are the widths defined above. Then if $\ell - k > 1$, the generating function $E_{k,\ell,m}(\overline{\boldsymbol{y}}, \underline{\boldsymbol{y}}, \overline{\boldsymbol{w}}, \underline{\boldsymbol{w}}, \boldsymbol{z}, \overline{\boldsymbol{z}}, \underline{\boldsymbol{z}})$ is self-reciprocal,*

$$E_{k,\ell,m}(1/\overline{\boldsymbol{y}}, 1/\underline{\boldsymbol{y}}, 1/\overline{\boldsymbol{w}}, 1/\underline{\boldsymbol{w}}, 1/\boldsymbol{z}, 1/\overline{\boldsymbol{z}}, 1/\underline{\boldsymbol{z}}) =$$

$$-\frac{z_k z_\ell \prod_{j=1}^{m-1}(\overline{y}_j \underline{y}_j) \prod_{j=k+1}^{\ell-1}(\overline{w}_j \underline{w}_j)}{\prod_{j=1}^{m-1} z_j \prod_{j=k+1}^{\ell} \underline{z}_j \prod_{j=k}^{\ell-1} \overline{z}_j} E_{k,\ell,m}(\overline{\boldsymbol{y}}, \underline{\boldsymbol{y}}, \overline{\boldsymbol{w}}, \underline{\boldsymbol{w}}, \boldsymbol{z}, \overline{\boldsymbol{z}}, \underline{\boldsymbol{z}}). \tag{8.44}$$

*Proof.* The linear homogeneous Diophantine system is the union of five sets of equations which we label $L_1$–$L_5$. The regions to the left and right of the hole give $L_1$ and $L_2$, the regions below and above the hole give $L_3$ and $L_4$ and the inside of the hole gives $L_5$:

$$L_1 = \begin{cases} M_0 - M_1 - \underline{N}_1 = 0 \\ M_j + \overline{N}_j - M_{j+1} - \underline{N}_{j+1} = 0 \qquad \text{for } 1 \le j \le k-2 \\ M_{k-1} + \overline{N}_{k-1} - \overline{M}_k - M_k - \underline{M}_k - \underline{N}_k = 0 \end{cases}$$

$$L_2 = \begin{cases} \underline{M}_\ell + M_\ell + \overline{M}_\ell + \overline{N}_\ell - M_{\ell+1} - \underline{N}_{\ell+1} = 0 \\ M_j + \overline{N}_j - M_{j+1} - \underline{N}_{j+1} = 0 & \text{for } \ell+1 \le j \le m-2 \\ M_{m-1} + \overline{N}_{m-1} - M_m = 0 \end{cases}$$

$$L_3 = \begin{cases} \underline{M}_k - \underline{M}_{k+1} - \underline{N}_{k+1} = 0 \\ \underline{M}_j + \underline{H}_j - \underline{M}_{j+1} - \underline{N}_{j+1} = 0 & \text{for } k+1 \le j \le \ell-1 \end{cases}$$

$$L_4 = \begin{cases} \overline{M}_j + \overline{N}_j - \overline{M}_{j+1} - \overline{H}_{j+1} = 0 & \text{for } k \le j \le \ell-2 \\ \overline{M}_{\ell-1} + \overline{N}_{\ell-1} - \overline{M}_\ell = 0 \end{cases}$$

$$L_5 = \begin{cases} M_k - M_{k+1} - \underline{H}_{k+1} = 0 \\ M_j + \overline{H}_j - M_{j+1} - \underline{H}_{j+1} = 0 & \text{for } k+1 \le j \le \ell-2 \\ M_{\ell-1} + \overline{H}_{\ell-1} - M_\ell = 0. \end{cases} \tag{8.45}$$

All heights of course are nonnegative. Self-avoidance imposes the additional constraint that the heights denoted $M_j$, $\overline{M}_j$ and $\underline{M}_j$ be positive. The set $\boldsymbol{\tau}$, defined in step 2 above, contains six unknowns whose constraints are not independent which we eliminate as follows: $M_0$ using the first equation of $L_1$, $M_m$ using the last equation of $L_2$, $\underline{M}_k$ using the first equation of $L_3$, $\overline{M}_\ell$ using the last equation of $L_4$, and $M_k$ and $M_\ell$ using the first and last equations of $L_5$. The resulting system is

$$L_1' = \begin{cases} M_j + \overline{N}_j - M_{j+1} - \underline{N}_{j+1} = 0 & \text{for } 1 \le j \le k-2 \\ M_{k-1} + \overline{N}_{k-1} - \overline{M}_k - M_{k+1} - \underline{H}_{k+1} - \underline{M}_{k+1} - \underline{N}_{k+1} - \underline{N}_k = 0 \end{cases}$$

$$L_2' = \begin{cases} \underline{M}_\ell + M_{\ell-1} + \overline{H}_{\ell-1} + \overline{M}_{\ell-1} + \overline{N}_{\ell-1} + \overline{N}_\ell - M_{\ell+1} - \underline{N}_{\ell+1} = 0 \\ M_j + \overline{N}_j - M_{j+1} - \underline{N}_{j+1} = 0 & \text{for } \ell+1 \le j \le m-2 \end{cases}$$

$$L_3' = \Big\{ \underline{M}_j + \underline{H}_j - \underline{M}_{j+1} - \underline{N}_{j+1} = 0 \qquad \text{for } k+1 \le j \le \ell-1$$

$$L_4' = \Big\{ \overline{M}_j + \overline{N}_j - \overline{M}_{j+1} - \overline{H}_{j+1} = 0 \qquad \text{for } k \le j \le \ell-2$$

$$L_5' = \Big\{ M_j + \overline{H}_j - M_{j+1} - \underline{H}_{j+1} = 0 \qquad \text{for } k+1 \le j \le \ell-2. \tag{8.46}$$

The substitutions $\overline{N}_j, \underline{N}_j, \overline{H}_j, \underline{H}_j = 1$ and $M_j, \overline{M}_j, \underline{M}_j = -1$ solve this new system of equations. One should note that when $k = 1$ or $m - \ell = 1$ the system is somewhat modified, but one may check that the solution still holds. Therefore we may apply Corollary 8.6 to obtain a self-reciprocity condition on the generating function for the solutions of $\bigcup_j L_j'$ subject to the positivity constraints on the heights. Making appropriate substitutions to restore the unknowns in set $\boldsymbol{\tau}$, and using Lemma 8.3 we obtain eqn. (8.44). ∎

We now treat three-choice polygons.

**Proposition 8.8.** *Let* $E_{k,\ell,m}(\overline{\boldsymbol{y}}, \underline{\boldsymbol{y}}, \overline{\boldsymbol{w}}, \underline{\boldsymbol{w}}, \boldsymbol{z}, \overline{\boldsymbol{z}}, \underline{\boldsymbol{z}})$ *be the generating function for three-choice polygons where* $k$, $\ell$ *and* $m$ *are the widths defined above. Then if* $m-k>1$, *the generating function* $E_{k,\ell,m}(\overline{\boldsymbol{y}}, \underline{\boldsymbol{y}}, \overline{\boldsymbol{w}}, \underline{\boldsymbol{w}}, \boldsymbol{z}, \overline{\boldsymbol{z}}, \underline{\boldsymbol{z}})$ *satisfies a self-reciprocity condition which, when* $\ell = k$, *takes the form*

$$E_{k,k,m}(1/\overline{\boldsymbol{y}}, 1/\underline{\boldsymbol{y}}, 1/\overline{\boldsymbol{w}}, 1/\underline{\boldsymbol{w}}, 1/\boldsymbol{z}, 1/\overline{\boldsymbol{z}}, 1/\underline{\boldsymbol{z}}) =$$

$$-\frac{\prod_{j=1}^{k-1}\overline{y}_j \prod_{j=1}^{m-1}\underline{y}_j \prod_{j=k+1}^{m-1}\underline{w}_j}{\prod_{j=1}^{k} z_j \prod_{j=k+1}^{m-1}\underline{z}_j} E_{k,k,m}(\overline{\boldsymbol{y}}, \underline{\boldsymbol{y}}, \overline{\boldsymbol{w}}, \underline{\boldsymbol{w}}, \boldsymbol{z}, \overline{\boldsymbol{z}}, \underline{\boldsymbol{z}}), \tag{8.47}$$

*and, when* $\ell > k$, *takes the form*

$$E_{k,\ell,m}(1/\overline{\boldsymbol{y}}, 1/\underline{\boldsymbol{y}}, 1/\overline{\boldsymbol{w}}, 1/\underline{\boldsymbol{w}}, 1/\boldsymbol{z}, 1/\overline{\boldsymbol{z}}, 1/\underline{\boldsymbol{z}}) =$$

$$-\frac{z_k \prod_{j=1}^{\ell-1}\overline{y}_j \prod_{j=1}^{m-1}\underline{y}_j \prod_{j=k+1}^{\ell-1}\overline{w}_j \prod_{j=k+1}^{m-1}\underline{w}_j}{\prod_{j=1}^{\min(\ell,m-1)} z_j \prod_{j=k}^{\ell-1}\overline{z}_j \prod_{j=k+1}^{m-1}\underline{z}_j} E_{k,\ell,m}(\overline{\boldsymbol{y}}, \underline{\boldsymbol{y}}, \overline{\boldsymbol{w}}, \underline{\boldsymbol{w}}, \boldsymbol{z}, \overline{\boldsymbol{z}}, \underline{\boldsymbol{z}}). \tag{8.48}$$

*Proof.* It is simpler to treat the two cases $\ell = k$ and $\ell > k$ separately. The proofs follow very closely that of Proposition 8.7. ∎

As for column-convex polygons, the two propositions above may be extended to other statistics.

**Corollary 8.9.** *Let* $\mathcal{P}$ *be either of the sets of staircase polygons with a staircase hole or three-choice polygons. Let* $\mathcal{P}_{k,\ell,m}$ *be the subset of figures in* $\mathcal{P}$ *with the widths* $k$, $\ell$ *and* $m$ *defined as above. Then the generating function for* $\mathcal{P}_{k,\ell,m}$ *according to any statistics linear in the quantities* $\overline{\mathbf{N}}, \underline{\mathbf{N}}$, $\overline{\mathbf{H}}$, $\underline{\mathbf{H}}$, $\mathbf{M}$, $\overline{\mathbf{M}}$, $\underline{\mathbf{M}}$, *is a self-reciprocal rational function (assuming* $\ell-k>1$ *for staircase polygons with a staircase hole and* $m-k>1$ *for three-choice polygons).*

The half-horizontal perimeter for either of the sets $\mathcal{P}$ is given by $m+\ell-k$. Using this in combination with Corollary 8.9, (8.42) and (8.43), we obtain the inversion relations specialised to horizontal and vertical perimeter, and area, which are listed in Table 8.1. The exceptional cases ($\ell - k = 1$ and $m - k = 1$ respectively) can be computed explicitly by the methods of [23].

## 8.6 Discussion

Each of the methods we have discussed for obtaining reciprocity or inversion relations has its own particular uses. For example, the method of Stroganov is suitable for lattice models in statistical mechanics which are characterised by a family of commuting transfer matrices. The Temperley method is mainly applicable to families of polygons that are column-convex or nearly so. Stanley's method for obtaining reciprocity results applies to any problem defined by a system of linear homogeneous Diophantine (LHD) equations, but the solutions to this system must be constrained by a system of simple inequalities of a certain form.

It is probable that for many lattice models in statistical mechanics the low temperature expansion can be framed as an LHD-system. However, most are likely to require more general types of constraints than the simple inequalities of the directed polyomino problems we have considered. Likewise, the non-directed polygon problems that we have successfully treated using the Temperley method can be recast as LHD-systems with more complex constraints. How to handle such constraints is a worthy problem for future investigation.

In recent work [33] this statistical mechanical language has been adapted for the enumeration of lattice paths, and may apply to polyomino problems as well. It is intriguing to speculate that the inversion relations found here may be connected with this approach.

We have not searched for inversion relations for any polyomino problem in variables other than the natural variables for the problem. Yet the example of the Potts model demonstrates that such inversion relations may exist. It is also possible that symmetries in addition to the ones presented here can be found for some problems. It is our hope that such additional symmetries might lead to the solution of currently intractable problems.

For the moment, we remark that the search for inversion and symmetry relations appears to provide a new method to tackle certain combinatorial problems. The degree of applicability of this method is still unclear.

# Part III

# Numerical techniques

# CHAPTER 9

# Two step restricted walks

# 9.1

# Introduction

The *self-avoiding walk* (SAW) (see for example [121] and references therein) and its derivatives have been a major source of models describing the thermodynamic, geometric and topological properties of different types of long chain polymers in solution. A large number of modifications, such as the addition of various interactions (*e.g.* surface or intra-polymer — see chapter 6) or particular restrictions (*e.g.* directedness), have been made to the basic model to mimic either various physical situations or to allow for easier analysis (such as exact solution). Some of these changes in the basic model modify the scaling behaviour (*i.e.* the asymptotic behaviour) of system properties, and hence change the universality class. For example, it is well known that restricting SAW on the square or cubic lattices by only allowing steps in the positive axial directions, thus producing so-called directed (or rather fully directed) walks (see [140, 136]), changes the way that the geometric size of the polymer scales with its length. Geometric restrictions, such as directedness or spirality, are one particular type of modification whose effect on the scaling behaviour of walks is of considerable interest. Non-directed but restricted walks were first introduced by Grassberger [77] and the models he examined were found to be in the same universality class as SAW. However, spiral self-avoiding walks (SSAW) [90, 100, 157, 15] on the square lattice were subsequently found to be in a different universality class to unrestricted SAW (and, of course, directed walks). Another novel universality class, studied subsequently [124, 158, 88, 86, 37], is that of anisotropic spiral self-avoiding walks (ASSAW), also defined on the square lattice. This class has proven difficult to analyse [37] but appears to be distinct from the other classes.

The self-avoiding walk models previously analysed that give rise to the various universality classes of walks with geometric constraints can be described by the title 'two-step-restricted walks' (TSRW) since these models are specified by the directions in which subsequent steps are allowed after steps in each of the four (square lattice) lattice directions are made. For example, one might specify that after either positive $x$ or $y$ axis steps only positive $x$ or $y$ axis steps can be made and that negative steps are disallowed — this gives fully directed walks. Such rules might model oriented polymers in complicated external fields. We note here that these two-step-restricted walk models are by their nature oriented. In two dimensions a wide-ranging study [86] of the universality classes, as determined by the scaling of the "size" of two-step-restricted SAW on the square lattice (as measured by the mean end-to-end distance), without interactions, has been made. Apart from cataloguing the universality classes in two dimensions, and analysing the ASSAW class further, this study investigated the relationship between the symmetries of the lattice models and their universality classes. The study found that symmetry was a major factor in deciding into which universality class a particular TSRW model would fall.

The theoretical understanding of the effect of geometric restrictions on the scaling behaviour of three-dimensional self-avoiding walks is not as well developed. Guttmann and Wallace [87] introduced two walk models on the simple cubic lattice, which they called

Model S and Model A. Model A was argued to be in the same universality class as three-dimensional SAW, while the Model S, a three-dimensional equivalent of the spiral walk, appeared to be a member of a distinct class.

In this work we examine a subset of the possible two-step restriction rule models on the simple cubic lattice in a manner similar to that of Guttmann *et al.* [86]. Our purpose in doing this is twofold: Firstly to determine the possible universality classes for such rules in three dimensions and secondly to attempt to find a similar relationship between the symmetry of the microscopic rule and the macroscopic scaling behaviour of the ensemble of walk configurations, as in the work of Guttmann *et. al.* [86]. We focus on rules that are likely to produce configurations that are not simply directed, or zero-, one- or two-dimensional. Exact enumeration and subsequent series analysis has been the basis of our studies here. Our first analyses on the radius of gyration[1] series show the possibility of novel universality classes in three dimensions. However, no correspondence can be made between the symmetry of the rules and these apparent universality classes. Also, the difference in the exponents between different universality classes is relatively small. More detailed analysis paints a different picture: that there are no 'real' equivalents of the spiral or anisotropic spiral classes among three-dimensional two-step rules — that is, there are no novel universality classes — and so we deduce that the unique topology of two dimensions must be an important factor in determining the number of different universality classes there.

We begin our discussion in the next section with the definition of a TSRW model. The space of two-step-restricted walk models is much larger in three dimensions than in two. To understand the factors important in our choice of models to study we first review the results of earlier square lattice studies with a view to extracting the salient features. We provide a complete pictorial classification of the interesting square lattice TSRW models in figure 9.3. This motivates the cubic lattice models we have studied, which we describe in section 9.2 also. We then describe the generation of our exact enumeration data and its various analyses in section 9.3. Finally, in section 9.4 we provide a discussion of our numerical results, summarising the different reasonable theoretical scenarios, and cautiously pointing out the most likely conclusions.

[1]The radius of gyration of an $n$ step walk configuration, $\varphi_n$, is defined to be

$$R_g^2(\varphi_n) = \frac{1}{2(n+1)^2} \sum_{i,j=0}^{n} (\vec{r}_i - \vec{r}_j)^2,$$

where $\{\vec{r}_i \mid i = 0, \ldots, n\}$ are the coordinates of the $(n+1)$ sites within $\varphi_n$. Like the square end-to-end distance, the radius of gyration is a measure of the geometric size of a walk.

# 9.2

# Two-step-restricted walk models

## 9.2.1 Definition

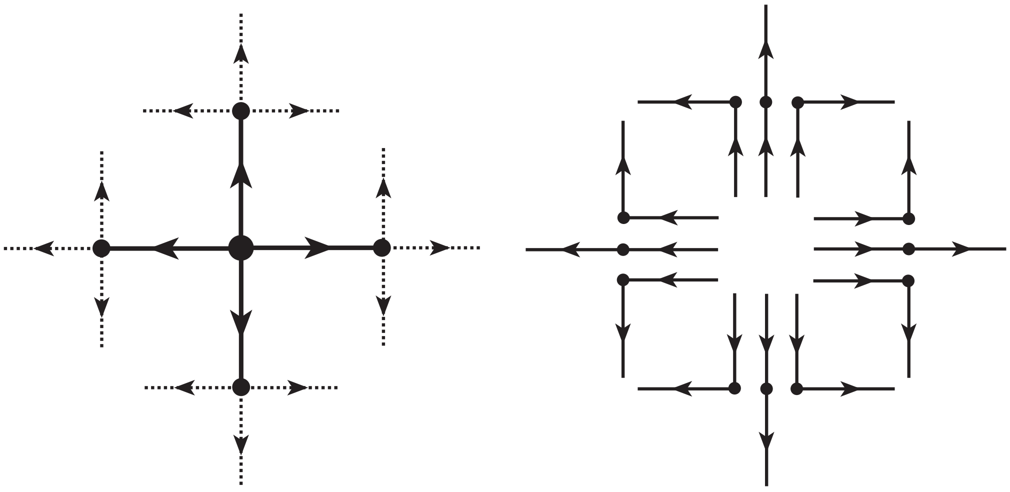
In this chapter we consider two-step-restricted self-avoiding walk (TSRW) models on the square and simple cubic lattices. However, this type of model can be defined on any lattice with a finite number of types of vertices. To be specific, and for the sake of simplicity, let us first consider the square lattice problem. One begins by considering oriented self-avoiding walks on this lattice (starting from some fixed origin). To specify the model one does two things. The first is to generate a set of allowed two-step configurations. To do this one considers a vertex of the lattice and, in turn, each of the bonds emanating from that site. On the square lattice there are 4 bonds emanating from each site. Assuming a step of the walk is on the bond (one of the four) under consideration one then specifies the bonds that are allowed for the next step of the walk. From each of the four first-step bonds there are 3 possible continuing bonds (considering self-avoidance). This means there are 12 possible two-step configurations for an oriented SAW on the square lattice. Figure 9.1 illustrates this construction with the possible 12 two-step configurations explicitly given. To specify a TSRW model one must say which of the 12 two-step configurations are allowed and which are not allowed. There are hence $2^{12} = 4096$ possible rules, and so 4096 walk models. Figure 9.2 illustrates one TSRW rule with the associated allowed two-step configurations also shown. The second part in obtaining a model is to take all oriented self-avoiding walks on the lattice where each two-step segment of each walk is one of the allowed two-step configurations. One can then ignore the orientation and this leaves a set of SAW which then defines a two-step-restricted rule model.

Now, as mentioned above, this construction can be made on any regular lattice. On a $d$-dimensional hypercubic lattice the cardinality of the rule space is $2^{2d(2d-1)}$, so while in two dimensions the cardinality of the rule space is $2^{12} = 4096$, in three dimensions it is $2^{30} = 1073741824$. The sheer size of the three-dimensional rule space does not make it at all easy to investigate, and so before we can start we must re-examine the two-dimensional rule space more carefully.

## 9.2.2 Two dimensions

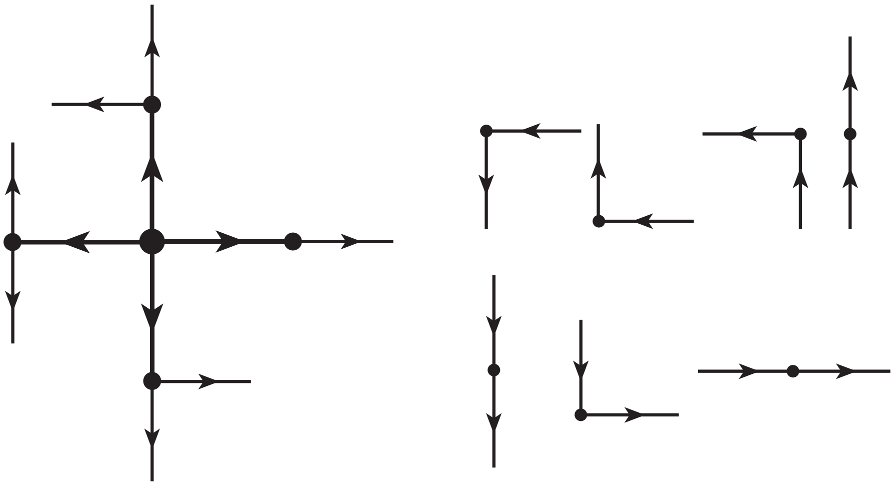
### Summary of two-dimensional results

On the square lattice Guttmann *et. al.* [86] have catalogued the classes of two-step-restricted self-avoiding walks, and examined the relationship between the scaling of the size of the objects, as measured by the various components of the end-to-end distance, and the symmetries of the walk rule. Some SAW models on other two-dimensional lattices have also been considered previously. Most notably the unrestricted SAW on the triangular and honeycomb lattices [58, 81, 89] appear, within error calculations, to be in the same universality class



(a) (b)

Figure 9.1: The construction of the allowed two-step configurations in a TSRW starts in (a) with the consideration of each of the possible bonds (full lines) of a vertex. One specifies which of the next steps (dashed lines) is allowed. Shown here also in (b) are the 12 square lattice two-step configurations.



(a) (b)

Figure 9.2: A particular TSRW rule is illustrated in (a) with the associated allowed two-step configurations shown in (b).

as square lattice SAW. Variants of the spiral SAW on the triangular lattice have also been studied [101, 118, 150]. At least with respect to the scaling of the size of the walks, as measured by the radius of gyration for example, the triangular and square lattice SAW show similar scaling behaviour.

In Guttmann *et. al.* [86] the universality class was conjectured mainly by considering the mean square end-to-end distance $\langle R_e^2 \rangle_n$ scaling of these walk models with walk length $n$; this being a measure of the size of the model polymer similar in behaviour to the radius of gyration, $\langle R_g^2 \rangle_n$. The exponent associated with any[2] measure of the average size of configurations, $\langle R^2 \rangle_n$, is denoted $\nu$ and one usually expects the dominant asymptotic form to be

$$\langle R^2 \rangle_n \sim A\, n^{2\nu} \quad \text{as} \quad n \to \infty\,, \tag{9.1}$$

where $\nu$ is the *critical exponent.* For SAW in two dimensions without restriction we expect that $\nu = 3/4$ [125], on any regular lattice. To be most general, one needs to define scaling exponents in the maximal and minimal scaling directions, that is $\nu_{\|}$ and $\nu_{\perp}$ respectively; this takes into account the fact that some walk configurations may be cigar-shaped (much longer in one direction than the other). The work of Guttmann *et. al.* [86] concluded that there are 7 universality classes of two-step restriction models on the square lattice differentiated by their 'size' scaling. Three of these are such that neither of the exponents take on the values 1 or 0 (these values imply some kind of one- or zero-dimensional behaviour respectively). These three classes are: unrestricted self-avoiding walks (SAW) with $\nu_{\|} = \nu_{\perp} = 3/4$; spiral self-avoiding walks (SSAW) where $\nu_{\|} = \nu_{\perp} = 1/2$ [101, 15] with confluent multiplicative logarithmic factors in the asymptotic form; and anisotropic spiral self-avoiding walks (ASSAW) where the latest Monte Carlo evidence [37] suggests $\nu_{\|} = 2\nu_{\perp} = 0.95(2)$. To be precise, the scaling form [101, 15] for SSAW geometric size has been derived exactly as

$$\langle R^2 \rangle_n \sim A_{Sp}\, n\, (\log n)^2 \quad \text{as} \quad n \to \infty\,. \tag{9.2}$$

The ASSAW model has certainly been the hardest to characterise and this may also be because of the existence of confluent logarithms in the scaling form for $\langle R^2 \rangle_n$ for that class [37]. There is no apparent exact solution for any ASSAW rule as there is for the SSAW class. A point worth noting here is that the corrections to the dominant scaling form (or corrections-to-scaling) of the radius of gyration seem to be smaller than those affecting the mean square end-to-end distance [37], and so exponent estimates obtained from the radius of gyration converge more quickly to a stable asymptotic value.

The classification of rules *not* falling into one of the three classes mentioned above is given by the following: some rules do not produce any long walks so $\nu_{\|} = 0$ and $\nu_{\perp} = 0$, that is, they are trivial or zero-dimensional rules; there are one-dimensional rules with $\nu_{\|} = 1$ and $\nu_{\perp} = 0$ where configurations are essentially made up of a single one-dimensional walk; there are rules that produce configurations made up of different one-dimensional walks, perhaps concatenated together a bounded number of times — they have $\nu_{\|} = 1$ and $\nu_{\perp} = 1$; finally some rules give walks that fall into the universality class of directed walks with $\nu_{\|} = 1$ and

[2] It is a standard scaling assumption that changing the way in which "size" is measured does not change the way in which the size scales.

$\nu_\perp = 1/2$ (these are often described as (1+1)-dimensional or even '3/2'-dimensional). See table 9.2 for a list of size scaling exponents for the different two-dimensional universality classes.

The other property that has been commonly used to classify the behaviour of SAW model is the scaling, or asymptotic form, of the number of configurations , $c_n$, of length $n$. For SAW it is usually expected that the dominant asymptotic form of $c_n$ is given by

$$c_n \sim B\mu^n n^{\gamma-1} \quad \text{as} \quad n \to \infty\,, \tag{9.3}$$

where $\mu$ is the 'connective' constant and $\gamma$ is the universal entropic critical exponent. However, the scaling forms of $c_n$ for SSAW and ASSAW have been shown, exactly in the case of SSAW [90, 100, 157, 15, 101], and predicted numerically in the case of ASSAW [88], to have different forms. The two-dimensional results for $c_n$ are summarised in the table 9.1. Only in

| Walk model | Scaling form for the number of walks | Exponent and constant values or estimates |
|---|---|---|
| SAW | $c_n \sim B\mu^n n^{\gamma-1}$ | $\gamma = 43/32$ [125] |
| Square Spiral | $c_n \sim Be^{\frac{2\pi}{\sqrt{3}}\sqrt{n}} n^\beta$ | $\beta = -7/4$ [90, 100, 157, 15] |
| Triangular Spiral I | $c_n \sim Be^{\pi\sqrt{\frac{2}{3}}\sqrt{n}} n^\beta$ | $\beta = -5/4$ [101] |
| Triangular Spiral II and III | $c_n \sim Be^{2\pi\sqrt{n}} \log(\frac{n}{12}) n^\beta$ | $\beta = -13/4$ [150] |
| ASSAW | $c_n \sim B\mu^n e^{a\sqrt{n}} n^\beta$ | $\beta \approx 0.9$, $a \approx 0.14$ [88] |
| DW | $c_n \sim B\mu^n$ | $\mu > 1$ and $\gamma = 1$ [69] |

Table 9.1: The scaling of the number of configurations, $c_n$, for several two-dimensional walk models.

the case of unrestricted SAW and directed walks (DW) is it possible to interpret the power of the algebraic factor as a critical exponent (only in these cases does the associated generating function have a dominant algebraic singularity). The models labelled Triangular Spiral II and III have a multiplicative logarithmic confluent factor in their scaling form relative to the square lattice spirals. The geometric size scaling for these models remains undetermined so it is unclear how to interpret the triangular lattice results. It is likely that the geometric size scaling is less sensitive to minor variations in the model and that the exponent $\nu$ for these models is 1/2 (log) as in the case of square lattice spiral and 'Triangular Spiral I' walks.

### Discussion of square lattice rules

We now concentrate on the classification of the square lattice TSRW models via their geometric scaling form. As described above, Guttmann *et. al.* [86] found that there are 7 universality classes. They also attempted to ascertain the microscopic constraints on the rules that determine the geometric scaling form of the walks. They concluded that three factors were important in determining the universality class. The first was the (somewhat imprecise) idea that there must be enough of the 12 two-step configurations allowed to give a non-trivial or non-one-dimensional rule. Secondly, they quoted 'balance' as a criterion: rules that do not have equal numbers of continuing steps in the positive and negative components of each axis are either directed, one-dimensional or trivial. This criterion was tested for several rules and seems to be well borne out by the exact enumeration studies. One can also argue that if the rule is unbalanced then the random walk generated with the rule will be directed. Furthermore one can plausibly conjecture that adding self-avoidance should not affect this directedness. Hence unbalancedness is a good indication that the self-avoiding walk generated with this rule is directed also. Note, however, that the converse is not true: the 'balance' condition is satisfied by some rules that give directed and one-dimensional walks. The final determining factor in the classification of the non-directed and non-one-dimensional (and trivial) rules was argued to be that of symmetry. The symmetries involved are single rotations and reflections: on the square lattice rotations by $\pi$ and $\pi/2$, and reflections about the lines $\theta = 0, \pi/4, \pi/2, 3\pi/4$. In table 9.2 the 7 (geometric scaling) universality classes are listed, along with the symmetries obeyed by the various rules in that class.

| Rule | $\nu_{\parallel}$ | $\nu_{\perp}$ | Rotation by 90° | Rotation by 180° | Reflection |
|---|---|---|---|---|---|
| SAW (S) | 3/4 | 3/4 | e | y | y |
| Spiral (P) | 1/2(log) | 1/2(log) | y | y | n |
| Anisotropic Spiral (A) | 0.95(2) | 0.47(1) | n | y | n |
| Directed (D) | 1 | 1/2 | n | e | e |
| Pseudo-1d (U) | 1 | 1 | e | e | e |
| 1d (O) | 1 | 0 | n | e | e |
| Trivial (T) | 0 | 0 | e | e | e |

Table 9.2: Length scale exponents and symmetries for the 7 known two-dimensional universality classes of two-step-restricted rule SAW. We define a one letter code for each class. The letter 'y' stands for 'yes', 'n' for 'no', and 'e' for 'either' (*i.e.* when some members of the class possess this symmetry and some do not).

So if one excludes directed, one- or zero-dimensional TSRW rules then:

- All *SAW*-like rules have rotation-by-$\pi$ symmetry and some reflection symmetry.
- All *Spiral*-like rules have rotation-by-$\pi/2$ (and hence rotation-by-$\pi$) symmetry but no reflection symmetry.
- All *ASSAW*-like rules have rotation-by-$\pi$ symmetry but no rotation-by-$\pi/2$ or reflection symmetries.

To be able to tackle the three-dimensional TSRW models let us first consider this classification in a little more detail. The balance condition can be expanded. Since one can always obtain the same walk configurations of a particular rule by considering the 'reverse' rule, which is obtained by reversing the orientation on the set of two-step configurations (and reversing the origin of the walks), it is sensible to also enforce a balance condition on the reverse of any rule one requires to be non-directed or non-one-dimensional. We denote this the 'reverse-balance' condition. Note that there are rules that are balanced but not reverse-balanced and so produce walks in the directed or one-dimensional universality classes (see for example rule (k) in [86]). As mentioned previously there are 4096 TSRW models on the square lattice. There are however only 80 such rules that obey the balance and reverse-balance conditions, which we shall refer to as the *symmetric-balance* condition from now on.

Starting with the unrestricted SAW rule in two dimensions and deleting possible moves, whilst imposing the symmetric-balance condition, one arrives at all the interesting rules first: being SAW (rule 1) and other SAW-like rules (rules 2 and 3), *spiral* (rule 7), *three-choice* (rule 5), *two-choice* (rule 6(a)) and *reverse-two-choice* (rule 6(b)). That is, these walks have the most continuing steps, as one might expect — all these rules have at most 4 missing continuing steps. We provide in figure 9.3 a complete list of all the distinct (up to rotations, reflections and reversals) symmetric-balanced rules starting with the rules with the most continuing steps.

In constructing this list we have found a walk-rule that is symmetric-balanced that was not examined by Guttmann *et. al.*. These rules display the interesting property that "spiral" configurations cannot occur, and so we have called them *anti-spiral* walks. In [139] we enumerate all anti-spiral walks up to length 50; differential approximant analysis of this series is badly behaved but seems to indicate that anti-spirals fall into a novel universality class. In an attempt to verify this observation we are currently conducting Monte Carlo simulations of anti-spiral walks using a modified version of the PERM algorithm (see [78] for a discussion of the PERM algorithm). Initial analysis of walks of lengths up to 8192, seems to indicate that anti-spiral walks are affected by very strong corrections-to-scaling, and that they probably fall into the directed walk universality class. It is also interesting to note that the "directedness" of the anti-spiral rule is due to self-avoidance; if self-avoidance is "turned off" then the walk becomes a random walk centred about the origin. No other directed walk displays this property.

A summary of the universality classes of the contents of the figure 9.3 is given in table 9.3.

| Universality Class | $\nu_\parallel$ | $\nu_\perp$ | Number of Rules | Distinct rules |
|---|---|---|---|---|
| SAW (S) | 3/4 | 3/4 | 4 | 3 |
| Anti-Spiral (I) | $\approx 0.8$ | $\approx 0.6$ | 4 | 1 |
| ASSAW (A) | 0.955(2) | 0.4775(10) | 12 | 2 |
| Spiral (P) | 1/2(log) | 1/2(log) | 2 | 1 |
| Directed Walk (D) | 1 | 1/2 | 6 | 2 |
| Pseudo-1d (U) | 1 | 1 | 19 | 5 |
| 1d (O) | 1 | 0 | 24 | 6 |
| Trivial (T) | 0 | 0 | 9 | 4 |

Table 9.3: The 80 two-dimensional balanced and reverse-balanced walk rules form a total of 7 (possibly 8) universality classes. The anti-spiral (rule 4 in figure 9.3) may be either in a novel universality class (I) or more likely is a slowly converged element of the directed walk (D) class. The anti-spiral exponent estimates are from our initial series analysis and should be treated with caution.

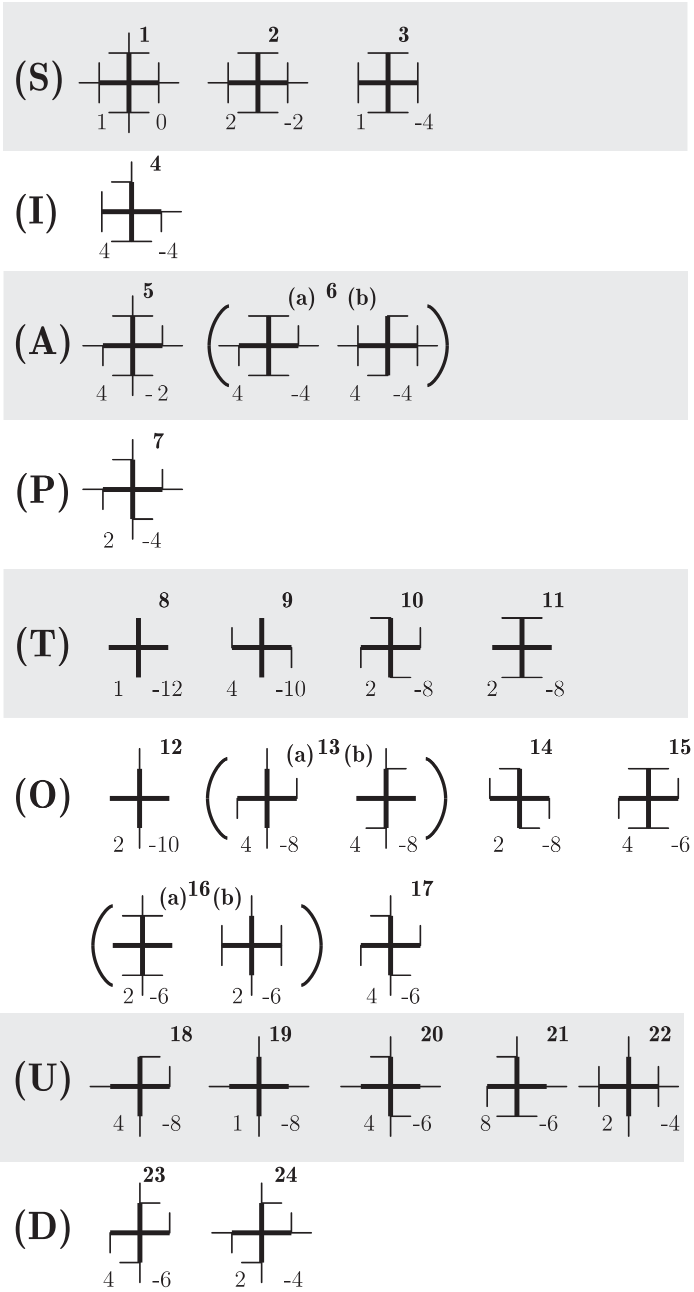



Figure 9.3: Diagrams of all 24 distinct (up to rotations, reflections and reversals) two-step-restricted walk rules on the square lattice. The rules are classified according to their apparent universality class — since we are yet to determine the universality class of the anti-spiral rule (rule number 4) we have placed it separately and denoted its class by "I" for indeterminate. Each pair of rules 6 (a) and 6 (b), rules 13 (a) and 13(b), and rules 16 (a) and 16(b) are simply reversals (traversing the rule backwards) of each other respectively. We have assigned each distinct rule a number label (in bold) above each pictogram, and indicated below each rule both the degeneracy of the rule on the left and on the right the number of continuing steps that have to be removed from the unrestricted SAW rule to give that rule (*e.g.* $-4$ means 4 steps need to be removed).

There is clearly one further criterion missing to distinguish the 'interesting' rules. One can see in hindsight that none of the rules that are symmetric-balanced and fall into the D,U,O or T classes obey the following condition: that from each of the four directions one can by a sequence of allowed steps end up in any direction (including itself again) while obeying self-avoidance. We call this the *mixing* condition: it is, of course, related to the fact that there are 'enough' continuing steps in each direction.

So following on from Guttmann *et. al.* [86] one can write down a simple set of rules to determine if a rule produces walks in one of the S, P or A classes. Starting with all TSRW rules:

1. All rules that are not symmetric-balanced are directed, one-dimensional or trivial (that is in the D,U,O or T classes).
2. All symmetric-balanced rules that are not 'mixing' rules are also in the D,U,O or T classes.
3. All symmetric-balanced and mixing rules that have reflection symmetry about any axis are of the SAW (S) universality class.
4. All symmetric-balanced and mixing rules that do not have a reflection symmetry but are symmetric with respect to a rotation by $\frac{\pi}{2}$ are in the SSAW (P) universality class.
5. All symmetric-balanced and mixing rules that are neither reflection symmetric nor symmetric under a $\frac{\pi}{2}$ rotation are in the ASSAW (A) class.

Note that all rules in the S, P and A classes (in fact of the original 4096 there are only 18 rules — 6 distinct rules — in these classes) are symmetric with respect to rotations of $\pi$.

**Remark.**
Since we do not yet have sufficient data to draw a conclusion on the universality class of anti-spiral walks, we cannot yet *definitively* alter this classification to include them. In any case a few remarks are in order.

The *anti-spiral* walk rule displays the interesting property that after the walk (with rule as shown in figure 9.3) has reached a given point (let us call it $(0,0) \in \mathbb{Z}^2$), it is not possible for the walk to reach (after any number of steps) any point $(x, y)$ with $x, y \in \mathbb{Z}^+$ — we call this "region-exclusion". It is worth noting that the symmetric balanced rules in the T, O, U and D universality classes display this region-exclusion property.

As noted above, we are currently investigating this rule, both by series analysis and Monte Carlo simulation, and it appears that it is probably in the directed-walk universality class. If this is the case, then the above classification can be made to work by modifying the "mixing" condition to disallow region-exclusion.

If it transpires that the anti-spiral walk is in fact in a novel universality class then the above classification would simply be expanded to distinguish the symmetric-balanced and mixing rules that have a reflection symmetry but no rotation symmetry at all.

It is also interesting to note that one can distinguish the non-one-dimensional and non-directed walks by examination of their turning numbers. We define the turning number for a two-dimensional walk rule as the square of the difference of the number of two-step configurations that make a left turn to the number of two-step configurations that make a right turn. Those two-step configurations that proceed straight ahead make no contribution to the turning number. Using the turning number we could replace the last three steps in the above classification scheme by:

3. (S) rule walks have turning number 0,
4. (P) rule walks have turning number 16 and
5. (A) rule walks have turning number 4.

So, in summary, the TSRW rules giving walks in the 'interesting' classes of S, P and A are distinguished from other rules by the symmetric-balance and mixing conditions, while they are distinguished from each other by the consideration of the symmetry of the rules. We note that it may be possible to distinguish rules in the D,U,O or T classes from each other but this is of less interest.

### 9.2.3 Three dimensions: delineating properties of the two-step rule space

In two dimensions the cardinality of the TSRW rule space is $2^{12} = 4096$, while in three dimensions it is $2^{30} = 1073741824$. In this section we shall follow the lessons learned in the discussion of two-dimensional models described above. We do this by only considering rules that are symmetric-balanced and mixing. Let us call the set of TSRW rules that obey the symmetric-balanced and mixing conditions the symmetric-mixing rules. So let us consider the implications of these conditions for the space of three-dimensional TSRW rules.

#### Characterising symmetric-balanced rules

On the simple cubic lattice there are 30 two-step configurations, so we could encode a particular rule by a 30 bit binary number. Alternatively we can encode the rules using a $6 \times 6$ square matrix of zeros and ones in the following way (6 because vertices are 6 fold coordinated on the simple cubic lattice). We label the lattice axes in the usual way with $x$, $y$ and $z$, and steps in the $\pm x$ direction as **r** and **l** respectively, steps in the $\pm y$ direction as **f** and **b** respectively, and steps in the $\pm z$ direction as **u** and **d** respectively. Hence a two-step configuration made up of an 'up' step in the positive $z$ direction followed by a 'backward' step in the negative $y$ direction is labelled as **ub**. We define the matrix, $\mathbf{M}$, as

$$\mathbf{M} = \begin{bmatrix} \mathbf{rr} & 0 & \mathbf{rf} & \mathbf{rb} & \mathbf{ru} & \mathbf{rd} \\ 0 & \mathbf{ll} & \mathbf{lf} & \mathbf{lb} & \mathbf{lu} & \mathbf{ld} \\ \mathbf{fr} & \mathbf{fl} & \mathbf{ff} & 0 & \mathbf{fu} & \mathbf{fd} \\ \mathbf{br} & \mathbf{bl} & 0 & \mathbf{bb} & \mathbf{bu} & \mathbf{bd} \\ \mathbf{ur} & \mathbf{ul} & \mathbf{uf} & \mathbf{ub} & \mathbf{uu} & 0 \\ \mathbf{dr} & \mathbf{dl} & \mathbf{df} & \mathbf{db} & 0 & \mathbf{dd} \end{bmatrix} \tag{9.4}$$

where the elements are 1 or 0 depending on whether the corresponding two-step configurations occur in the rule space or not, respectively. That is, if the two-step configuration **ub** occurs in our rule then the position $(5, 4)$, labelled by **ub** in equation (9.4) above, will contain a 1, otherwise it contains 0. For general dimensional hypercubic lattices the matrix $\mathbf{M}$ has binary elements with fixed zero elements for positions $(2k-1, 2k)$ and $(2k, 2k-1)$ for all $k \in \{1, \ldots, d\}$.

We can write the various balance restrictions for the hypercubic lattice simply as follows: the balance condition requires that the sum of elements of $\mathbf{M}$ in successive columns taken in pairs is the same, that is

$$\sum_i \mathbf{M}_{i,(2k-1)} = \sum_i \mathbf{M}_{i,2k} \quad \text{for } k \in \{1, \ldots, d\} \tag{9.5}$$

while the reverse-balance condition requires that the sum of elements of $\mathbf{M}$ in successive rows taken in pairs is the same, that is

$$\sum_j \mathbf{M}_{(2k-1),j} = \sum_j \mathbf{M}_{2k,j} \quad \text{for } k \in \{1, \ldots, d\}\,. \tag{9.6}$$

The number of such rules can be calculated, and further, the detailed numbers of such rules made from a fixed number of two-step configurations can be calculated (see appendix 9.5) by constructing a generating function that sums over all allowed matrices $\mathbf{M}$ subject to the constraints (9.5) and (9.6) above.

### Symmetries of TSRW rules in three dimensions

By restricting the consideration to only distinct symmetric-balanced rules the number of rules on the square lattice falls from 80 to 24. If we further add mixing to the constraints imposed then this number falls to 7. So we now consider the symmetries of cubic lattice TSRW. This serves a dual purpose. The symmetries allow us to focus our attention on the distinct rules (rules that give distinct ensembles) and also provide a possible set of conditions that may be used to distinguish any new universality classes discovered, as in two dimensions. We also consider the turning number as a property that may also delineate such classes.

We now list all single rotation and reflection symmetries of the rules, as well as defining what we call the turning number of the rule. We also define a symmetry we call flip-symmetry: this was considered since its existence is a quick way to ensure that the rule is symmetric-balanced. The properties we have used in three dimensions to attempt to classify TSRW rules are

1. Rotational symmetries about the coordinate axes
   - $\pm\frac{\pi}{2}$
   - $\pi$.
2. Reflection symmetries
   - reflection in coordinate planes. The normals of these planes are given by: $\vec{n} = \hat{e}_x, \hat{e}_y, \hat{e}_z$; the unit vectors in each axial direction.
   - reflection in diagonal planes[3]. The normals of these planes are given by: $\vec{n} = (\hat{e}_x \pm \hat{e}_y), (\hat{e}_x \pm \hat{e}_z), (\hat{e}_y \pm \hat{e}_z)$.
3. Turning number
   - The turning number for a three-dimensional walk rule is the sum of the turning numbers of the planar walk rules in each co-ordinate plane when viewed from the positive side of the co-ordinate axis normal to that plane.
4. Flip symmetry
   - send each co-ordinate to its negative. In two dimensions this is equivalent to rotation by $\pi$. If a walk has the flip symmetry, then it is balanced and reverse-balanced. The reverse is not true.

It should be noted that it can be easily shown that TSRW rules in three dimensions with a $\frac{\pi}{2}$ rotational symmetry have a plane of reflection; the normal of the plane of reflection is the axis of the rotation. The converse is not true.

So in later discussion we attempt to classify the rules studied into universality classes according to which *single* symmetry operations acting on the rules leave the rules unchanged. All the rules (bar one) examined had the flip symmetry, and as noted above this implies all the rules were symmetric-balanced.

### 9.2.4 Construction of symmetric-mixing cubic lattice TSRW rules

Without actually going through all 432096, or even the smaller number of distinct iso-symmetric sets, of the symmetric-balanced TSRW rules on the cubic lattice we wanted to be able to choose a set of three-dimensional rules that adequately explored this set of models (by encompassing the different symmetries, etc). We also wanted to ensure the mixing condition

[3]The operator to reflect a vector in the plane with normal vector $\vec{n} = (\hat{e}_x + \hat{e}_y + \hat{e}_z)$, is given by the matrix:

$$R_{(\hat{e}_x+\hat{e}_y+\hat{e}_z)} = \frac{1}{3}\begin{bmatrix} 1 & -2 & -2 \\ -2 & 1 & -2 \\ -2 & -2 & 1 \end{bmatrix}$$

From this matrix we see that the vector $\hat{e}_x$ is mapped to $\frac{1}{3}(\hat{e}_x - 2\hat{e}_y - 2\hat{e}_z)$. That is, this operator is not closed on the vector space $\mathbb{Z}^3$. Hence a walk rule cannot have symmetry under this reflection, nor reflections in planes with normals given by $(\hat{e}_x \pm \hat{e}_y \pm \hat{e}_z)$.

held. To do this we chose to consider walk rules in three dimensions such that each of the three planes of the rule was one of a small number of planar walks. We chose to construct rules using representatives from most of the two-dimensional universality classes. We chose from the following rules (see the catalogue in figure 9.3 for the rule numbers):

*S* Rule 1 — Unrestricted planar SAW: in class (S), turning number 0;

*3* Rule 5 — Three-choice walks: in class (A), turning number 4;

*2* Rule 6(a) — Two-choice walks: in class (A), turning number 4;

*P* Rule 7 — Planar spiral walks: in class (P), turning number 16;

*D* Rule 24 — Directed walks (with reverse): in class (D), turning number 0;

*C* Rule 22 — Walks that are concatenations of 1d walks: in class (U), turning number 0;

*O* Rule 19 — Walks that are one-dimensional along either axis: in class (U), turning number 0;

*R* Rule 17 — Walks that form incomplete one-dimensional rectangles: in class (O) turning number 4.

Each three-dimensional rule we considered can be given a 3 letter code such as (*P*-*O*-*3*), which means the plane with a normal along the $z$-axis was given the square lattice rule ($P$) while the plane with a normal along the $y$-axis was given the square lattice rule ($O$) and the plane with a normal along the $x$-axis was given the square lattice rule (*3*). The three-dimensional self-avoiding walk is denoted by (*S*-*S*-*S*), Guttmann and Wallace's spiral walk [87] (model S) is (*P*-*P*-*P*) while Guttmann and Wallace's 'anisotropic spiral' 3d-equivalent (model A) is (*S*-*C*-*3*). Note that not every combination of these types is possible; for example one cannot construct a (*S*-*3*-*2*) walk. This is because each quadratic lattice walk is defined by 12 choices of two-step configurations that are made, and since there are 3 axial planes we have a total of 36 choices to make. However, in three dimensions one really has only 30 possible choices to make, so some combinations will be inconsistent (and so impossible). The five models that we have analysed most extensively are illustrated in figures 9.4, 9.5, 9.6, 9.7, and 9.8. In figure 9.4 the rule (*P*-*P*-*P*) is shown, in figure 9.5 the rule (*P*-*2*-*2*) is shown, in figure 9.6 the rule (*P*-*O*-*3*) is shown, in figure 9.7 the rule (*S*-*C*-*P*) is shown, while in figure 9.8 the rule (*P*-*R*-*2*) is shown. By definition the turning number of the three-dimensional rule is simply the sum of the turning numbers of the three two-dimensional rules that make up the rule.

In fact we have examined by exact enumeration studies all the walk rules constructed plane-by-plane with the added restriction of not having more than one plane with a one-dimensional or directed rule (*D, C, O* or *R*). This restriction was made to ensure we ended up with three-dimensional rules that were mixing. We considered the 38 such rules initially. All these rules had flip-symmetry and so were symmetric-balanced, and therefore symmetric-mixing. This was still too many rules to consider in depth and because of the

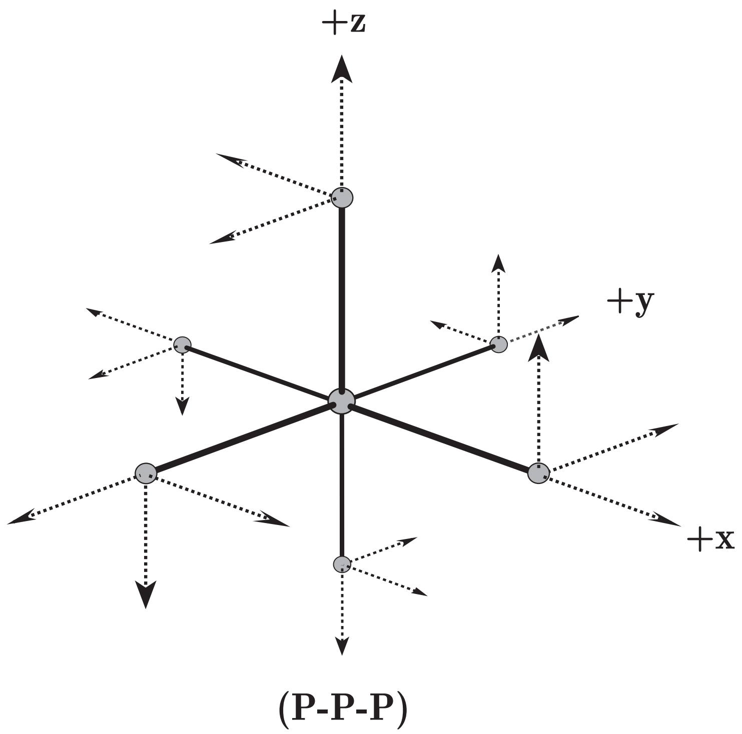



Figure 9.4: An illustration of the (*P*-*P*-*P*) rule.

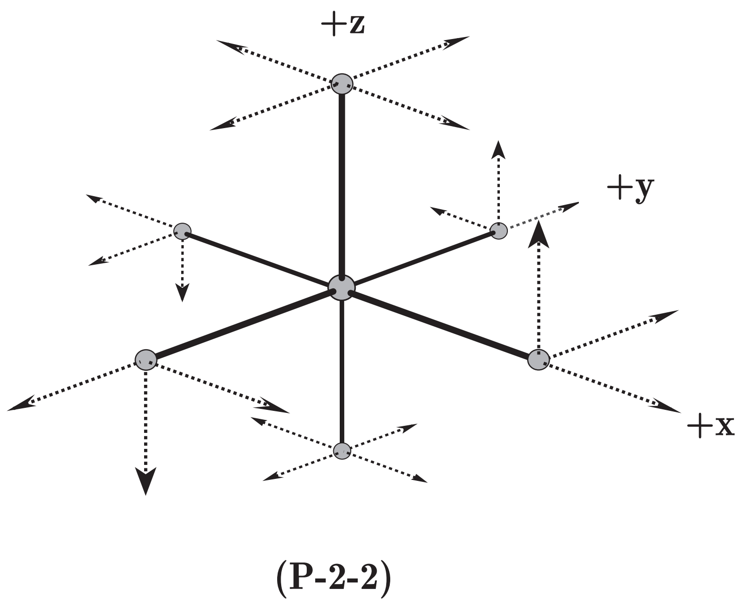



Figure 9.5: An illustration of the (*P*-*2*-*2*) rule.

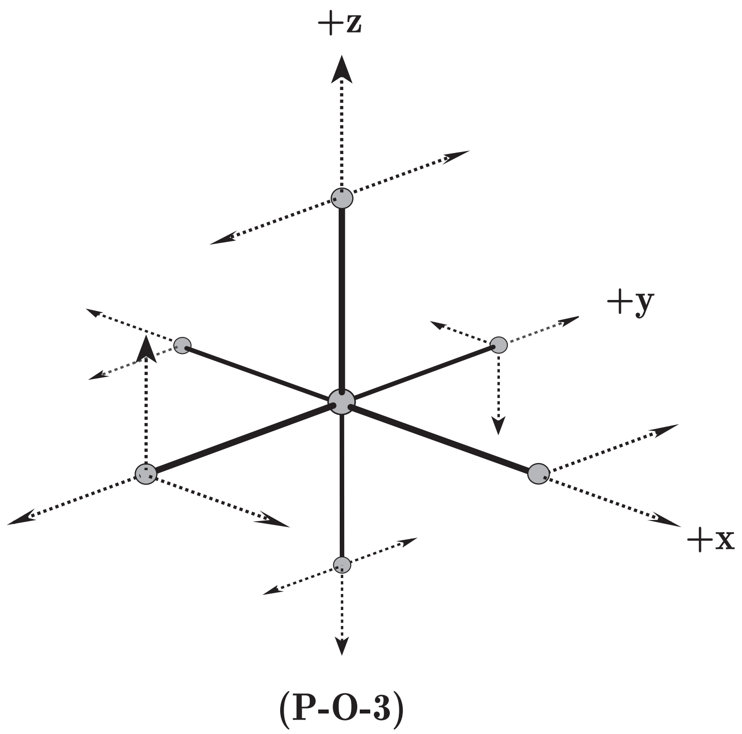



Figure 9.6: An illustration of the (*P-O-3*) rule.

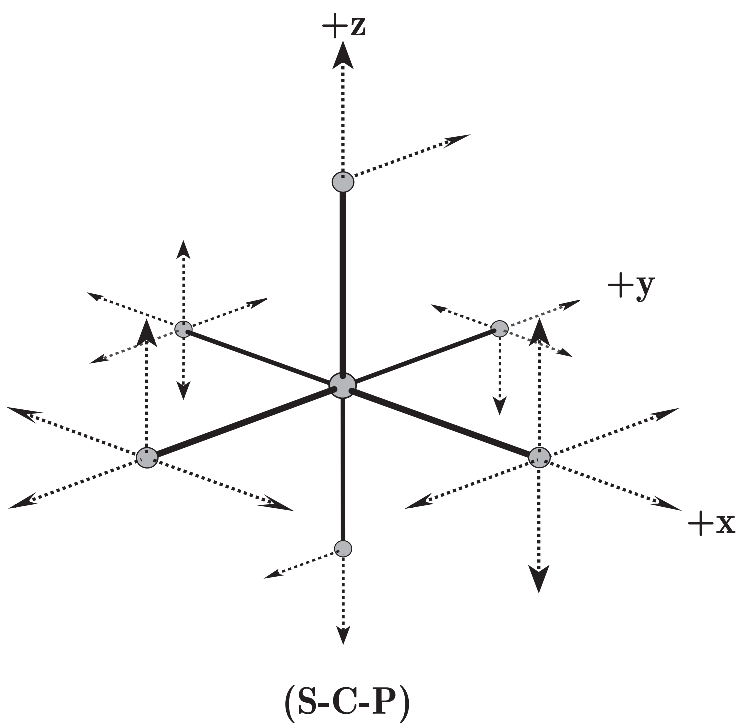



Figure 9.7: An illustration of the (*S-C-P*) rule.

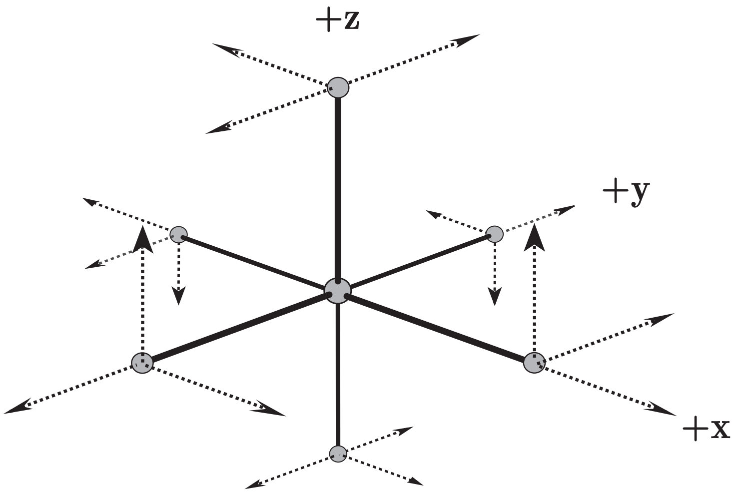



Figure 9.8: An illustration of the (*P-R-2*) rule.

larger connective constants in three dimensions than in two the series enumerated were relatively shorter. We chose the 9 most promising rules from a numerical point of view that covered many of the symmetry combinations plus 3 other rules (one not constructed in the above manner) so as to include all possible symmetry combinations. These rules included both the rules considered by Guttmann and Wallace previously [87].

We especially constructed a rule outside the gamut of the procedure described above so as to produce a rule that was symmetric under rotations by $\pi$ but not under rotations by $\pi/2$ or any reflection. We call this rule *Rot-*$\pi$ and it is a symmetric-balanced rule. The $\mathbf{M}$ matrix for this rule is

$$\mathbf{M}_{Rot\text{-}\pi} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} . \tag{9.7}$$

## 9.3 Exact enumeration results and analysis

We now describe the enumeration and analyses of the 11 TSRW models constructed plane-by-plane, in the manner described in the section 9.2.4. The rules were (*3-3-C*), (*S-C-3*), (*P-2-2*), (*P-R-2*), (*P-3-3*), (*P-O-3*), (*P-P-P*), (*S-P-3*), (*P-P-3*), (*P-P-D*) and (*S-C-P*). These were chosen from the original 38 models on which we performed short enumerations, not detailed here. We also considered the (*Rot-*$\pi$) rule, described above, to ensure the different possible symmetry combinations are covered. After the analysis of the 12 models we concentrated on the 5 models that were numerically best behaved and representative of the numerical behaviour found in the 12 models.

We began by enumerating the numbers of walks, $c_n$, and the total radius of gyration, $r_n$, where $\langle R_g^2 \rangle_n = r_n / c_n$, for each of the 12 models listed above using a recursive back-tracking algorithm up to various maximum lengths, $n \leq N$, that depended on the model. (Later we also calculated the full moment of inertia for 5 of the models.) We chose to calculate and analyse the radius of gyration rather than the end-to-end distance, since in two-dimensional studies the asymptotic analysis of the radius of gyration [37] has proven less affected by corrections-to-scaling[4], as discussed in section 2. The lengths of the enumerations depended on the effective connective constants of the models, and our enumerations ranged in length, $N$, from 18 to 29. The initial enumerations of the 12 models are given in appendix 9.6.1. In particular, we have increased the length of the enumerations for the ($P$-$P$-$P$) model from 23 [87] to 29 and the ($S$-$C$-$3$) model from 18 [87] to 23 steps. As an example, the ($P$-$P$-$P$) enumerations up to length 28 took approximately 150 CPU hours on a Digital Alphastation 500/266.

### 9.3.1 Review of previous three-dimensional work

Unrestricted SAW, ($S$-$S$-$S$), on the cubic lattice have been studied by both exact enumeration [81] and Monte Carlo techniques [111, 40]. For unrestricted SAW, scaling theory [121] predicts that the number of walks scales as

$$c_n \sim B\,\mu^n\, n^{\gamma-1} \quad \text{as } n \to \infty \tag{9.8}$$

$$\text{so} \quad \sum_{n=0}^{\infty} c_n x^n \sim B'\,(1-\mu x)^{-\gamma} \quad \text{as } x \to 1/\mu^-\,, \tag{9.9}$$

and that the total radius of gyration scales as

$$r_n^2 \sim D\,\mu^n n^{2\nu+\gamma-1} \quad \text{as } n \to \infty \tag{9.10}$$

$$\text{so} \quad \sum_{n=0}^{\infty} r_n^2 x^n \sim D'\,(1-\mu x)^{-(2\nu+\gamma)} \quad \text{as } x \to 1/\mu^-\,, \tag{9.11}$$

while the average radius of gyration scales as

$$\langle R_g^2 \rangle_n \sim A\, n^{2\nu} \quad \text{as } n \to \infty \tag{9.12}$$

$$\text{so} \quad \sum_{n=0}^{\infty} \langle R_g^2 \rangle_n x^n \sim A'\,(1-x)^{-(2\nu+1)} \quad \text{as } x \to 1^-\,. \tag{9.13}$$

[4] There is no reason to believe that this is a general feature, however it is not an unreasonable assumption.

An exact enumeration[5] study [81] on the simple cubic lattice, of SAW up to length 21 found

$$
\begin{aligned}
1/\mu &= 0.213496(4)\,, & (9.14)\\
\gamma &= 1.161(1)\,, & (9.15)\\
\text{and} \quad 2\nu &= 1.184(6)\,. & (9.16)
\end{aligned}
$$

Various high precision Monte Carlo studies using walks up to lengths $N = 40000$ and $N = 80000$ respectively have found $\gamma = 1.1575(6)$ [40] and $2\nu = 1.1754(12)$ [111]. To make a fair comparison with our series analysis of the 12 TSRW we shall only use the exact enumeration results quoted above for the unrestricted SAW model.

Guttmann and Wallace [87] studied both the ($P$-$P$-$P$) and ($S$-$C$-$3$) model by exact enumeration, with walk lengths up to 23 and 18 respectively. They calculated the end-to-end distance rather than the radius of gyration. They used a differential approximant analysis, and also various ratio analyses to study the models. The differential approximant analysis concluded that for the ($P$-$P$-$P$) model

$$
\begin{aligned}
1/\mu &= 0.3765(2)\,, & (9.17)\\
\gamma &= 1.24(20)\,, & (9.18)\\
\text{and} \quad 2\nu &= 1.3(4)\,, & (9.19)
\end{aligned}
$$

while for the ($S$-$C$-$3$) model

$$
\begin{aligned}
1/\mu &= 0.2883(2)\,, & (9.20)\\
\gamma &= 1.16(2)\,, & (9.21)\\
\text{and} \quad 2\nu &= 1.19(5)\,. & (9.22)
\end{aligned}
$$

The ratio methods gave more precise answers for the geometric-size exponents, $2\nu = 1.29(3)$ for the ($P$-$P$-$P$) model, and $2\nu = 1.18(1)$ for the ($S$-$C$-$3$) model. Note that, despite having similar numbers of terms as the unrestricted SAW enumeration, the analyses for these models give far less precise exponent estimates, and the series are far less well-behaved under differential approximant analysis. Despite the large error bars on the differential approximant analyses, the further analyses via the ratio method led the authors to conjecture that the ($S$-$C$-$3$) model is a member of the unrestricted SAW universality class while the ($P$-$P$-$P$) model is part of a novel class, being a three-dimensional counterpart to the 2d-spiral class (P). They noted that both ($S$-$C$-$3$) rule and unrestricted SAW have a plane of reflection symmetry while the ($P$-$P$-$P$) rule does not. They then concluded that this may be the microscopic criterion for the difference in the universality class, as it is in two dimensions. We

[5] A very recent study [119] enumerates all self-avoiding walks and their mean square end-to-end distance on the cubic lattice up to length 26. Updated estimates of $\mu, \gamma$ and $\nu$ are:

$$
\begin{aligned}
1/\mu &= 0.213492(4)\,,\\
\gamma &= 1.1575(6)\,,\\
\text{and} \quad 2\nu &= 1.1751(11)\,.
\end{aligned}
$$

This data is also used to find an improved upper bound on the connective constant, $\mu < 4.7114$.

note in passing that, while (*S-C-3*) possesses a rotation-by-$\pi$ symmetry, the (*P-P-P*) rule has no rotation symmetry (in contradiction to the claim made in [87]), so that could equally well be the microscopic criterion.

### 9.3.2 Differential approximant analysis of $c_n$ and $\langle \boldsymbol{R}_g^2 \rangle_n$ for 12 TSRW models

We first performed differential approximant analyses [80] on the number of walks ($c_n$) and the radius of gyration ($\langle R_g^2 \rangle_n$) for the 12 TSRW models. We used second-order inhomogeneous approximants that utilised all the available coefficients, and we varied the range of the order of the polynomials to suit the lengths of the series. We checked some of the analyses with first and third order approximants. In general the first order approximants were not as well behaved and the third order approximants gave similar results to the second order approximants, though they could not be used effectively due to the short length of many of the series.

Since we had no *a priori* estimates for the critical points of the various $c_n$ series, we arrived at our $1/\mu$ and $\gamma$ estimates by using unbiased approximants, and obtained error estimates from the spread (standard deviations) of the approximants. The estimates calculated were the mean values and the errors were two standard deviations.

We then considered the radius of gyration series, $\langle R_g^2 \rangle_n$. Since the generating function of $\langle R_g^2 \rangle_n$ has a critical point equal to unity we used this as one measure of the convergence of that series. Many of the $\langle R_g^2 \rangle_n$ series were poorly converged compared to the unrestricted SAW series. Most also had a relatively wide spread of approximants. We found that using 'biased approximants', as per [80], yielded very poor results. Instead we arrived at a biased exponent estimate from the unbiased approximants in the following way: we took a large range of (unbiased) approximants, made a linear fit to the central section of the approximants and then extrapolated back to a critical point of $x = 1$. The various estimates and associated errors we calculated were taken from this linear fit. Firstly, we calculated the mean critical point and exponent of the approximants on which the linear fit was taken, which we denote $\bar{x}_c$ and $2\bar{\nu}$ respectively (errors are two standard deviations). Secondly, we calculated the estimate from linear biasing itself, $2\nu_{LB}$ (error quoted is the error from the linear regression). The final estimate, $2\nu_{final}(\Delta_{stat})(\Delta_{sys})$ is the same as the biased value, $2\nu_{LB}$, but quoted with two errors: $\Delta_{stat}$ is the statistical error from the linear regression; while $\Delta_{sys}$ is a measure of the systematic error in biasing the approximants back to unity — it is equal to the difference between the mean exponent value and biased value, $|2\nu_{LB} - 2\bar{\nu}|$. Using $\Delta_{sys}$ for our estimate of systematic error may be considered rather conservative, but this estimate has proved to be a useful measure of systematic error in other SAW problems, such as polymer adsorption [6]. A typical differential approximant spread for the $\langle R_g^2 \rangle_n$ series is shown in figure 9.9, with illustrations of the various estimates and associated errors we calculated given.

The results from the differential approximant analysis are given in table 9.4. First considering the entropic exponent $\gamma$ estimates in table 9.4, we find that the SAW value of 1.161 falls within the respective errors estimated for all the models. This would suggest one reasonable conclusion to be that all the models are in the *same* universality class as unrestricted SAW.

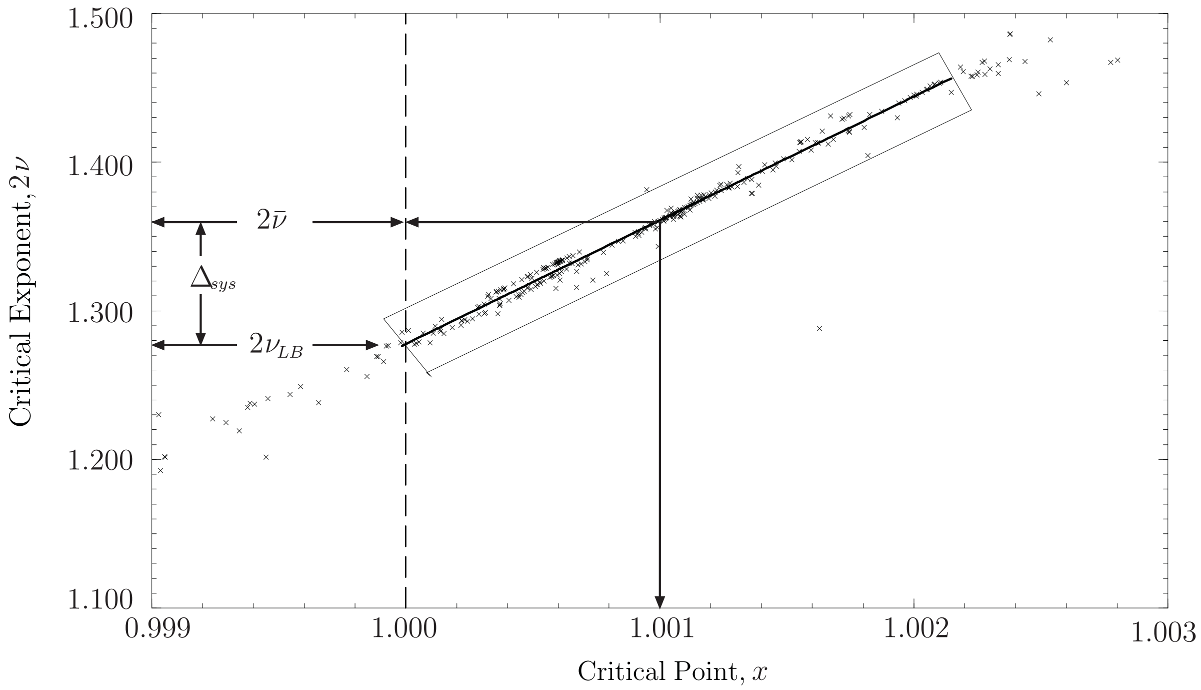



Figure 9.9: Shown is the differential approximant spread for the radius of gyration exponent, $2\nu$, for the *(P-P-P)* model. The box indicates the area over which an average was taken. A linear fit is shown which gives our 'biased' estimate, $2\nu_{LB}$, from the point where the fit has an intercept with the dashed $x = 1$ vertical line. The arrows indicate the means of the boxed region's critical point, $\bar{x}_c$, and exponent, $2\bar{\nu}$. The difference between the boxed mean, $2\bar{\nu}$, and the biased estimate, $2\nu_{LB}$, of the critical exponent gives us an estimate of systematic error, $\Delta_{sys}$: in other walk problems this has usually proved to be conservative, though not always.

The $\langle R_g^2 \rangle_n$ approximants, by contrast, seem to indicate that the models fall into possibly three different universality classes:

- *SAW*-like: (*S-S-S*), (*S-C-3*), (*S-C-P*), (*3-3-C*), (*Rot-*$\pi$) and (*S-P-3*);
- New: (*P-3-3*), (*P-O-3*) and (*P-2-2*);
- 3d-spiral: (*P-P-P*) and (*P-R-2*).

We were unable to classify the (*P-P-3*) rule because of the size of the associated error bars on the exponent estimates, however it seems to lie in either the apparent 3d-spiral class or the new class, though it may, of course, form another class again. Though the results from the analysis of the data are too poor to draw any reasonable conclusions we have included the (*P-P-D*) data because of the symmetries of the walk rule and its winding number.

The spread of $\langle R_g^2 \rangle_n$ approximants for several representative rules is shown in figure 9.10. This figure clearly shows what appears to be 3 separate bands of approximants representing 3 separate universality classes. Our results in this section generally concur with those of

| Rule | $1/\mu$ | $\gamma$ | $\bar{x}_c$ | $2\bar{\nu}$ | $2\nu_{LB}$ | $2\nu_{final}$ |
|---|---|---|---|---|---|---|
| *(S-C-3)* | 0.2884(1) | 1.16(2) | 1.0002(4) | 1.20(2) | 1.19(1) | 1.19(1)(1) |
| *(3-3-C)* | 0.2904(2) | 1.16(2) | 1.0005(6) | 1.22(3) | 1.19(1) | 1.19(1)(3) |
| *(S-C-P)* | 0.3046(2) | 1.17(2) | 1.0002(7) | 1.21(3) | 1.20(1) | 1.20(1)(1) |
| *(S-P-3)* | 0.26985(4) | 1.169(6) | 1.0003(3) | 1.22(2) | 1.21(1) | 1.21(1)(1) |
| *(Rot-$\pi$)* | 0.3575(4) | 1.17(4) | 0.9997(14) | 1.19(7) | 1.20(1) | 1.20(1)(1) |
| *(P-O-3)* | 0.4078(2) | 1.18(3) | 0.9999(4) | 1.22(2) | 1.226(6) | 1.23(1)(1) |
| *(P-3-3)* | 0.30558(8) | 1.173(14) | 1.0002(2) | 1.23(1) | 1.220(5) | 1.22(1)(1) |
| *(P-2-2)* | 0.3442(1) | 1.18(1) | 1.0003(3) | 1.25(2) | 1.23(1) | 1.23(1)(2) |
| *(P-P-3)* | 0.33167(7) | 1.176(8) | 1.0008(2) | 1.30(2) | 1.25(2) | 1.25(2)(5) |
| *(P-P-D)* | 0.3586(4) | 1.17(4) | 1.004(2) | 1.50(8) | 1.33(11) | 1.33(11)(17) |
| *(P-R-2)* | 0.3739(2) | 1.19(2) | 1.0011(6) | 1.36(3) | 1.30(2) | 1.30(2)(6) |
| *(P-P-P)* 2nd | 0.3757(1) | 1.18(2) | 1.001(1) | 1.37(6) | 1.29(2) | 1.29(2)(8) |
| *(P-P-P)* 3rd | 0.3757(1) | 1.17(4) | 1.001(1) | 1.36(10) | 1.27(2) | 1.27(2)(9) |
| SAW - here | 0.213497(10) | 1.162(2) | 1.0002(2) | 1.20(1) | 1.192(5) | 1.19(1)(1) |
| SAW - there | 0.213496(4) | 1.161(2) | | ○ | | 1.184(6) |

Table 9.4: Exponent estimates from differential approximant analysis. There are two sets of results for the $(P\text{-}P\text{-}P)$ model, corresponding to second order and third order approximants. We include, for completeness, two sets of SAW $(S\text{-}S\text{-}S)$ values, both obtained from exact enumeration data, one using our analysis method and one quoted from previous work.

Guttmann and Wallace [87] for the $(S\text{-}C\text{-}3)$ and $(P\text{-}P\text{-}P)$ models. On the other hand, it is difficult to make conclusions on the basis of these results due to the relatively large systematic errors in the $\nu$ estimates that arise from the biasing procedure. It is possible that the conclusion that there is more than one universality class is a manifestation of corrections-to-scaling: the series being too short for the method of differential approximants to work effectively.

Assuming for the moment that the $\langle R_g^2 \rangle_n$ analysis does imply the existence of 3 separate universality classes then our next task was to attempt to establish the microscopic criteria that classify rules according to these apparent universality classes. We attempted this using the criteria that have proved useful in two dimensions. Given that all the rules are symmetric-balanced, the various rule symmetries and the turning numbers were the candidates considered. The symmetries and turning numbers of these rules along with the maximum exact enumeration lengths are shown in table 9.5. All rules except $(Rot\text{-}\pi)$ were flip symmetric. Note that the unrestricted SAW model, $(S\text{-}S\text{-}S)$, has all the symmetries listed in table 9.5. Also, any rule that is symmetric under rotations by $\pi/2$ must be symmetric under rotations by $\pi$ and also, as mentioned in section 9.2.3, must possess a reflection symmetry. Hence, the unrestricted SAW rule $(S\text{-}S\text{-}S)$ is such a rule.

One can see immediately that all the rules in the new class, in the 3d-spiral class, and some in the SAW-like class do not have any rotation or reflection symmetries. Hence, these symmetries cannot be used to classify models into the apparent classes. In particular, reflection symmetry does not delineate the SAW class from the others. Hence, a classification

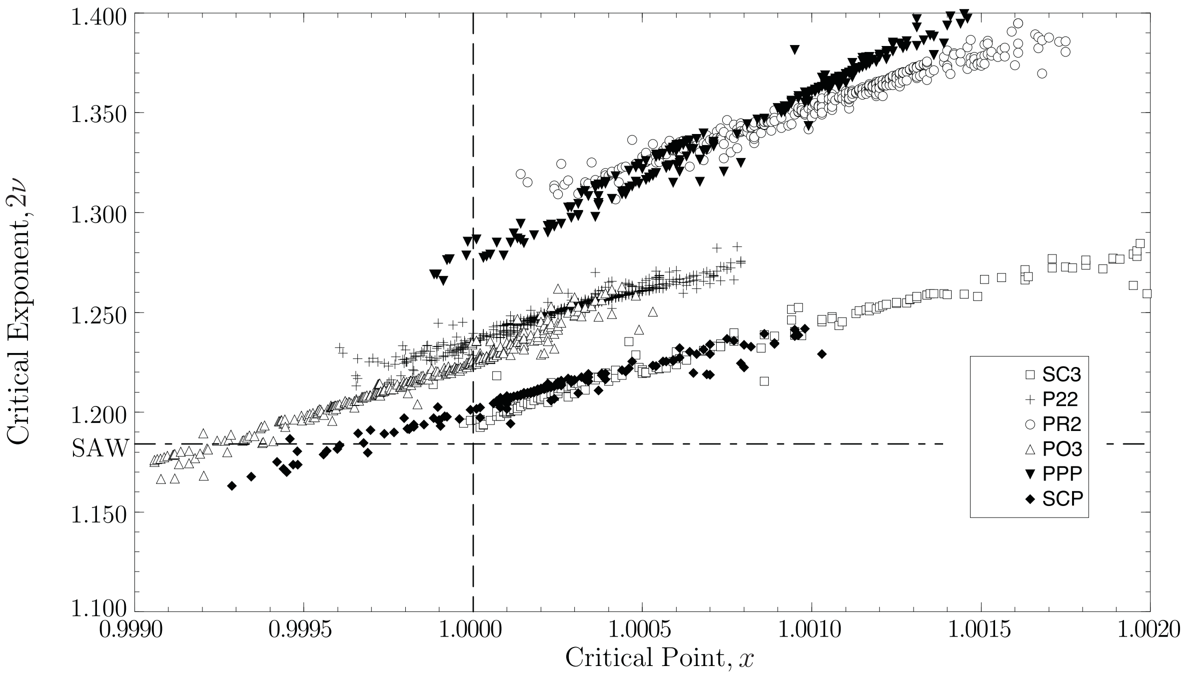



Figure 9.10: A plot of $\langle R_g^2 \rangle_n$ differential approximants for the exponent $2\nu$ for several representative models. Notice the 3 distinct bands of approximants, each one made up from the approximants for several models. These bands seem to indicate that there are 3 universality classes.

| Class | Rule | $N$ | Rotation by $\frac{\pi}{2}$ | Rotation by $\pi$ | Any Reflection | Turning Number |
|---|---|---|---|---|---|---|
| *SAW-like* | (*S-S-S*) | 21 | y | y | y | 0 |
| | (*S-C-3*) | 23 | n | y | y | 4 |
| | (*3-3-C*) | 20 | n | n | n | 8 |
| | (*S-C-P*) | 22 | n | y | y | 16 |
| | (*Rot-π*) | 22 | n | y | n | 16 |
| | (*S-P-3*) | 18 | n | n | n | 20 |
| New | (*P-O-3*) | 28 | n | n | n | 20 |
| | (*P-3-3*) | 24 | n | n | n | 24 |
| | (*P-2-2*) | 24 | n | n | n | 24 |
| Undetermined | (*P-P-D*) | 22 | n | n | y | 32 |
| | (*P-P-3*) | 25 | n | n | n | 36 |
| 3d-spiral | (*P-R-2*) | 26 | n | n | n | 36 |
| | (*P-P-P*) | 29 | n | n | n | 48 |

Table 9.5: Symmetries of the TSRW models examined. They are grouped according to the initial classification made from differential approximant analysis of the exact enumerations of the radius of gyration. The maximum length of the enumerations $N$ is also given.

according to symmetries of the rules is not forthcoming. A classification according to turning number would seem to be more successful (increasing turning number giving new classes at particular values, 20 and 36 perhaps). Here also there are problems: both the (*S-P-3*) and (*P-O-3*) rules have turning number 20 but seem to be in different universality classes and it is unclear in which class the model (*P-P-3*) lies despite having the same turning number as the model (*P-R-2*). These difficulties with an attempted microscopic classification and the consistency of the values of $\gamma$ with the SAW value across all the rule models led us to examine the scaling behaviour of the models both using different quantities and with different analysis techniques in an attempt to find a consistent answer to this conflicting information.

We chose to examine 5 models representative of the apparent universality classes: (*P-P-P*) and (*P-R-2*) for the 3d-spiral class, (*P-2-2*) and (*P-O-3*) for the new class, and (*S-C-P*) representing the SAW class. We first considered further analysis of the $c_n$ series that attempted to take account of non-standard scaling forms such as occurs in the two-dimensional spiral class (P), and secondly we analysed the mean moment of inertia tensor, $\langle \mathbf{I} \rangle_n$, to search for exponent anisotropy as occurs in the two-dimensional anisotropic spiral class (A).

### 9.3.3 Further analysis of $c_n$ for 5 TSRW models

As stated above, we first expected that the number of walks, $c_n$, behaved according to equation (9.8) for each of the rule models: the differential approximant analysis described above fits to this form (with implicit corrections). However, different scaling forms have been found in some two-dimensional walk models. Two-dimensional spiral walks [90] and walks of the ASSAW class [88] have scaling forms that include $e^{\sqrt{n}}$ factors (see table 9.1). This additional factor present in the spiral walks' partition function scaling is mathematically related to the scaling of partitions of integers [90]; in three dimensions it may be possible that there will be plane-partition-like terms which scale as [4, 14]

$$p_n \sim C\, n^{-25/36} e^{a_{PP} n^{2/3}} \qquad \text{as } n \to \infty\,. \tag{9.23}$$

We examined the possibility of corrections to the scaling form of the sort found in two-dimensional spirals and also those found in plane partitions. Differential approximant analysis (at least without significant modification) is unsuited to the study of such scaling forms, since the new factors imply that the generating function has an essential singularity. Of course, if such factors are present then the previous differential approximant analysis would have been inappropriate. With this in mind we attempted to make a direct fit of the data to the following forms

$$\begin{aligned} c_n &\sim C\, \mu^n n^{\gamma-1} \quad \text{as } n \to \infty, &(9.24)\\ c_n &\sim C\, \mu^n n^{\gamma-1} \exp(\alpha\sqrt{n}) \quad \text{as } n \to \infty\,, &(9.25)\\ \text{and} \quad c_n &\sim C\, \mu^n n^{\gamma-1} \exp(\alpha n^{2/3}) \quad \text{as } n \to \infty\,. &(9.26)\end{aligned}$$

More precisely, we made successive fits using the series terms $n$, $n-2$, $n-4$ and $n-6$, as necessary, to each of the forms (which (linearly) contain 3 or 4 constants to ascertain)

$$\log c_n = a_1 + n \log \mu + (\gamma - 1) \log n, \tag{9.27}$$
$$\log c_n = a_1 + n \log \mu + (\gamma - 1) \log n + a_2 \sqrt{n}, \quad \text{and} \tag{9.28}$$
$$\log c_n = a_1 + n \log \mu + (\gamma - 1) \log n + a_2 n^{2/3}, \tag{9.29}$$

exactly. We made these fits with either $\gamma$ free, or fixed at the SAW value (in the second case we used one less term at each stage). We did not attempt to consider multiplicative logarithmic corrections, as well as those above, since such forms are too difficult to fit with all but the longest series. The type of analysis described above is more refined than ratio analysis and has been used to great effect in SAW problems when considering corrections-to-scaling [45].

We illustrate the results of this fitting procedure in detail for the ($P$-$P$-$P$) walks in appendix 9.7.1. To summarise, after examining tables 9.14 and 9.15, one can see that if corrections of the form $e^{\sqrt{n}}$ or $e^{n^{2/3}}$ are introduced, then they have coefficients very close to zero. This implies that any effect of these terms is quite negligible, and indeed that they are probably not present. We have found similar results for the other four walk models considered. This gives us confidence that the differential approximant analysis of $c_n$ is giving reliable information. Hence, given that $e^{\sqrt{n}}$ or $e^{n^{2/3}}$ corrections seem not to be present in the scaling form for $c_n$, our original conclusion from the differential approximant analysis then stands: namely, that all the rules, including ($S$-$S$-$S$), have the same value of the $\gamma$ exponent within error, the value being that of the SAW universality class (around 1.16). If the walks scale with subtly different exponents or multiplicative logs, the series at hand are too short to allow an investigation of these possibilities.

### 9.3.4 Analysis of the inertia tensor for 5 TSRW models

In our initial enumerations of the 12 models (as well as our very initial enumerations — not described here — of the 38 models of section 9.2.4) we enumerated the radius of gyration so as to measure the scaling of the average size of walks in the various models. To gain a finer view of the scaling of the geometric size we calculated the full moment of inertia tensor for the 5 models we designated for more intense study. This allowed us to look for any scaling anisotropy and to consider the eigenvalues of this matrix in addition to the radius of gyration.

The eigenvectors of the moment of inertia matrix correspond to the *natural* coordinate axes in which the ('average') walks scale, and the eigenvalues to the radius of gyration in those directions. For example, in two dimensions the three-choice walk model [86] has a mean moment of inertia tensor with eigenvectors $\{[1,1],[1,-1]\}$, which correspond to the preferred and transverse directions. The eigenvalues for any three-dimensional TSRW model's moment of inertia matrix are expected to scale as

$$\lambda_j(n) \sim A_j n^{2\nu_j} \qquad j = 1, 2, 3 \quad \text{as } n \to \infty \tag{9.30}$$

where the values of $\nu_j$ and $A_j$ may or may not be independent of direction $j$. Hence, there are two types of anisotropy: scaling anisotropy where different eigenvalues scale with different exponents ($\nu_i \neq \nu_j$), and a milder anisotropy where only the constants $A_j$ differ. The moment of inertia tensor, $\mathbf{I}$, for a walk configuration $\varphi_n$ of $n$ steps, defined by the positions of its $n+1$ monomers (sites) as the set $\{\vec{r}_i = (x_i, y_i, z_i); i = 0, \ldots, n\}$, is given by

$$\mathbf{I}(\varphi_n) = \frac{1}{n+1}\sum_{i=0}^{n}(r_i^2\mathbf{1} - \vec{r}_i\vec{r}_i) \tag{9.31}$$

where we have used dyadic notation, and $\mathbf{1}$ is the identity tensor. Explicitly this gives

$$\begin{aligned}\mathbf{I}(\varphi_n) &= \frac{1}{n+1}\sum_{i=0}^{n}\left(\begin{bmatrix} r_i^2 & 0 & 0 \\ 0 & r_i^2 & 0 \\ 0 & 0 & r_i^2 \end{bmatrix} - \begin{bmatrix} x_ix_i & x_iy_i & x_iz_i \\ y_ix_i & y_iy_i & y_iz_i \\ z_ix_i & z_iy_i & z_iz_i \end{bmatrix}\right) \\ &= \frac{1}{n+1}\sum_{i=0}^{n}\begin{bmatrix} y_i^2+z_i^2 & -x_iy_i & -x_iz_i \\ -x_iy_i & x_i^2+z_i^2 & -y_iz_i \\ -x_iz_i & -y_iz_i & x_i^2+y_i^2 \end{bmatrix}. \end{aligned} \tag{9.32}$$

In our enumerations we computed the expectation of the moment of inertia tensor averaging over all walks $\varphi_n$ of length $n$, that is,

$$\langle \mathbf{I} \rangle_n = \frac{1}{c_n}\sum_{\varphi_n}\mathbf{I}(\varphi_n), \tag{9.33}$$

where the sum is over all walks, $\varphi_n$, of length $n$. The value of the moment of inertia depends on the origin of the coordinate system. Now, the trace of the moment of inertia tensor yields

$$\mathrm{Tr}(\langle \mathbf{I} \rangle_n) = \frac{1}{c_n}\sum_{\varphi_n}\left(\frac{2}{n+1}\sum_{i=0}^{n}\left(x_i^2+y_i^2+z_i^2\right)\right), \tag{9.34}$$

which is equal to twice the mean square distance of a monomer to the end-point if $\mathbf{I}$ is computed about one end-point of the walks, or $2\langle R_g^2\rangle_n$ if $\mathbf{I}$ is computed about the centre of mass. So, to expand on our radius of gyration enumerations, we enumerated the centre of mass $\frac{1}{n+1}\sum_{j=0}^{n}\vec{r}_j$ for each walk and then the components of the moment of inertia matrix about the centre of mass. The enumerations of the mean moment of inertia tensor computed about the centre of mass for our 5 models are given in appendix 9.6.2.

In a similar manner to the analysis of the $\langle R_g^2\rangle_n$ data, we used second-order inhomogeneous differential approximants to analyse the scaling of the eigenvalues of the mean moment of inertia tensor computed about the centre of mass, which we denote as $\lambda_1(n)$, $\lambda_2(n)$ and $\lambda_3(n)$ in ascending order of size. Note that since the trace of a matrix is the sum of its eigenvalues the scaling of the mean square monomer-to-end distance and the radius of gyration are dominated by the scaling of the largest eigenvalue. The results of this analysis are given in table 9.6.

These results for each of the 5 models indicate that the smallest eigenvalue, $\lambda_1(n)$, apparently scales with an exponent that is smaller than the exponent associated with the two larger eigenvalues. These two larger eigenvalues seem to scale with approximately the same

| Rule | $\lambda_1$ | | $\lambda_2$ | | $\lambda_3$ | |
|---|---|---|---|---|---|---|
| | $x_1$ | $2\nu_1$ | $x_2$ | $2\nu_2$ | $x_3$ | $2\nu_3$ |
| SAW | 1 | 1.184(6) | $\circ$ | | $\circ$ | |
| *(S-C-P)* | 0.9995(10) | 1.18(2)(2) | 1.0004(6) | 1.21(1)(1) | 1.0007(7) | 1.21(2)(3) |
| *(P-O-3)* | 0.9997(4) | 1.20(1)(2) | 0.9999(8) | 1.23(2)(1) | 1.0002(5) | 1.23(1)(2) |
| *(P-2-2)* | 1.0002(4) | 1.22(1)(1) | 1.0003(5) | 1.24(1)(1) | 1.0002(4) | 1.24(1)(2) |
| *(P-R-2)* | 1.0005(14) | 1.24(1)(4) | 1.0015(7) | 1.33(4)(7) | 1.0011(6) | 1.31(3)(5) |
| *(P-P-P)* 2nd | 0.9997(16) | 1.22(3)(2) | 1.0058(20) | 1.26(12)(45) | $\circ$ | |
| *(P-P-P)* 3rd | 0.9991(24) | 1.22(3)(7) | 1.0021(26) | 1.29(2)(16) | $\circ$ | |

Table 9.6: Estimates of $2\nu$ from the differential approximant analysis of the eigenvalues of the mean moment of inertia tensor computed about the centre of mass. Estimates of the critical points $x_i$ of the associated generating functions are included to show quality of convergence. (*P-P-P*) has two eigenvalues, the larger having multiplicity 2. SAW has only one eigenvalue and the estimate here comes from [81]. We also note here that for the (*P-R-2*) model although $\lambda_3 \geq \lambda_2$ we estimate $\nu_2 \geq \nu_3$. This implies that corrections to scaling are masking that $\nu_2 = \nu_3$.

exponent, that is $\nu_2 = \nu_3$. The value obtained for this exponent is the same as the one obtained from the analysis of the radius of gyration. This anisotropic scaling implies that the typical walk looks like a flattened ball, and that the ball becomes flatter as the walk length increases. It is interesting to note that the degree of anisotropy in the exponents seems to be of the order of 5%, in contrast to 50% found in the two-dimensional ASSAW (and DW) models. If it were true that this anisotropy were real it would indeed be curiously subtle, and unusual, for three-dimensional critical phenomena, as far as we are aware. Intriguingly, on closer examination of the results we find that the smallest eigenvalue is also typically the best converged, and it is (with the exception of the (*P-R-2*) model), well converged to a SAW-like value close to 1.20 with errors that encompass the best series estimate of the unrestricted SAW model of 1.184(6). So the best converged (the biased shift in the exponent is about the same size as the statistical spread of approximants and is about 0.01) differential approximant analysis is for the smallest eigenvalue: one might expect naively that the smallest eigenvalue is affected most by corrections-to-scaling and so behaves the worst under scaling analysis. Moreover, the analyses of the largest eigenvalues have such large systematic errors that the estimate-ranges often encompass (sometimes just so) the SAW value of 1.184(6). It is then advantageous to attempt to analyse these eigenvalue scalings again with other techniques.

Because of these unusual results we then analysed the series data again, making allowances for the existence of analytic and non-analytic corrections-to-scaling by fitting to an assumed scaling form in much the same manner as [45], and as we did for the further scaling of $c_n$ in section 9.3.3. In particular, we examined the cases of analytic corrections (to order $1/n^2$) and non-analytic corrections of the form $1/\sqrt{n}$ as reasonable guesses, assuming some compatibility with previous SAW work: that is, we considered

$$\lambda(n) \sim An^{2\nu}(1 + c_1/n + c_2/n^2 + O(1/n^3)) \quad \text{as } n \to \infty \tag{9.35}$$

and

$$\lambda(n) \sim An^{2\nu}(1 + c_2/\sqrt{n} + c_1/n + O(1/n^{3/2})) \quad \text{as } n \to \infty\,. \tag{9.36}$$

We did not fit directly to the above scaling form (since we did not have conjectured values of $2\nu$), rather we fitted to the term-by-term logarithm of the series

$$\log(\lambda(n)) = a_1 + 2\nu\log(n) + a_2/n + a_3/n^2 + O(1/n^3) \tag{9.37}$$

and

$$\log(\lambda(n)) = a_1 + 2\nu\log(n) + a_3/\sqrt{n} + a_2/n + O(1/n^{3/2})\,. \tag{9.38}$$

The (linear) fits were then examined on the basis of the stability of the coefficients. For each of these two forms above the exponent $2\nu$ was either allowed to be free, fixed at a SAW-like value of 1.19, or fixed to the apparent differential approximant calculated estimate obtained previously (1.29 for the 3d-spiral models and 1.23 for the new class).

We illustrate this analysis in more detail for ($P$-$P$-$P$) walks.

### Exact fitting to corrections-to-scaling for ($P$-$P$-$P$) model

For the ($P$-$P$-$P$) model the mean inertia tensor takes the form

$$\langle \mathbf{I} \rangle_{(PPP)} = \begin{bmatrix} a & b & -b \\ b & a & b \\ -b & b & a \end{bmatrix} \tag{9.39}$$

and such a matrix has the following eigenvalues, $\lambda_j$, and eigenvectors, $\vec{\phi}_j$:

$$\lambda_1 = \quad a - 2b \quad \text{and} \quad \{\vec{\phi}_1 = [1, -1, 1]\}, \tag{9.40}$$

$$\text{and} \quad \lambda_2 = \quad a + b \quad \text{and} \quad \{\vec{\phi}_2 = [-1, 0, 1], \vec{\phi}_3 = [1, 1, 0]\}\,. \tag{9.41}$$

The first interesting feature to notice is that this rule is not automatically isotropic, unlike the 2d-spiral (P) class model.

Differential approximant analysis on the two eigenvalues of the ($P$-$P$-$P$) model yielded (see table 9.6 and figure 9.11):

$$\lambda_1 \sim \quad A_1\, n^{2\nu_1} \quad \text{as } n \to \infty \quad \text{with} \quad 2\nu_1 \approx 1.22(5) \tag{9.42}$$

$$\lambda_2 = \lambda_3 \sim \quad A_3\, n^{2\nu_3} \quad \text{as } n \to \infty \quad \text{with} \quad 2\nu_3 \approx 1.29(18)\,, \tag{9.43}$$

where the errors quoted here are simply the sum of the statistical and systematic errors. The central estimates of the exponents imply that ($P$-$P$-$P$) walks scale anisotropically, and that the typical walk is shaped like a flattened ball, shorter in the $[1, -1, 1]$ direction, with a preferred plane normal to this. Since one eigenvalue apparently scales with a smaller exponent the ball becomes flatter as the walks become longer. Since the radius of gyration is the sum of the eigenvalues (up to a constant), the scaling of $\langle R_g^2 \rangle_n$ is dominated by the scaling of the largest and less well-converged eigenvalues; this translates into the relatively poor convergence of the $\langle R_g^2 \rangle_n$ series data for this model. On the other hand both estimates' ranges include the SAW value of 1.19. So we might conclude that it is simply the case that the larger eigenvalues are afflicted with large corrections-to-scaling.

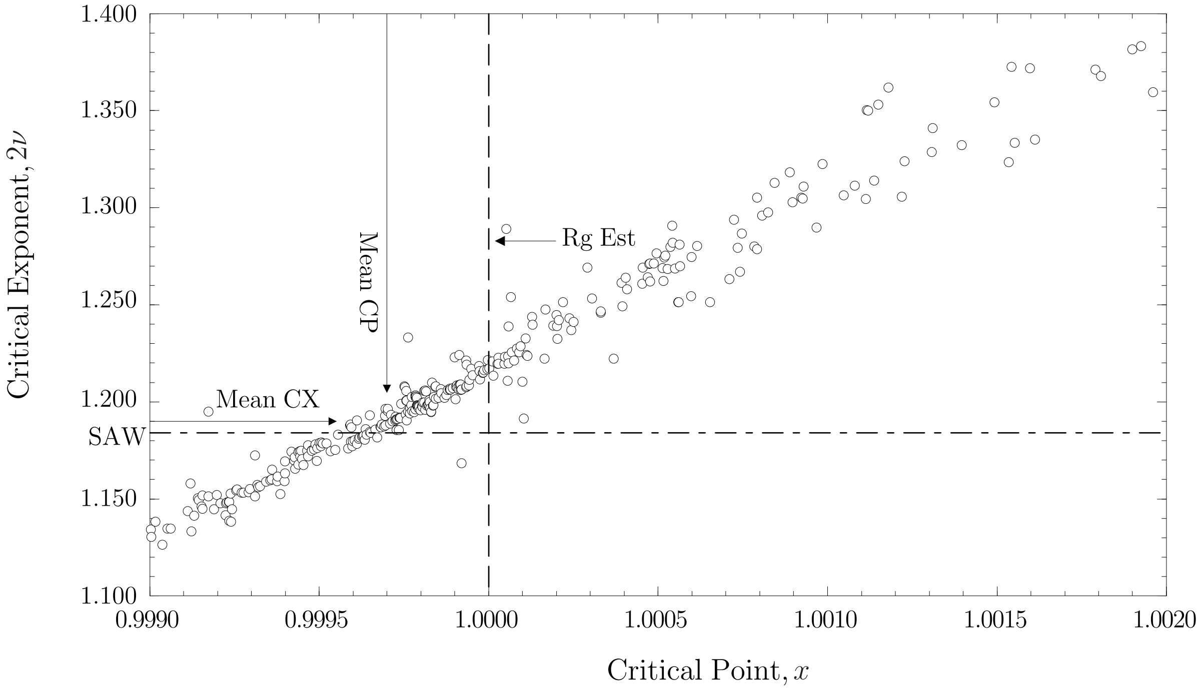



Figure 9.11: A plot of the $\lambda_1$ (smallest eigenvalue of the moment of inertia tensor) differential approximants for the $(P\text{-}P\text{-}P)$, which estimate the exponent $2\nu$. The mean value of the critical points and exponents of these approximants is indicated. The vertical dashed line is simply the $x = 1$ (correct) critical point, while the horizontal dashed line marks the SAW value of the exponent $2\nu$. The estimate of the exponent from the radius of gyration analysis, **Rg Est**, is also marked.

As mentioned above, it would be an unusual result if the walks scale as three-dimensional SAW in one direction, and differently in the other two. To explore this result further we tried to fit the eigenvalues directly to the scaling forms given by equations (9.35) and (9.36), in much the same way as [45], to examine the possible corrections-to-scaling. The results of these fits are given in appendix 9.7.2. In summary, the smallest eigenvalue is fitted best with standard $1/n$ corrections with a dominant exponent close to the unrestricted SAW value, in agreement with the differential approximant analysis. However, the larger eigenvalue can also be fitted with an exponent of $2\nu = 1.19$, which is close to the unrestricted SAW value, if $1/\sqrt{n}$ corrections (as in (9.36)) are added, and moreover, this is arguably the best fit of 6 attempted for the larger eigenvalue. Hence one can argue sensibly from these fits that a SAW-like exponent is not only consistent with but rather the best value (of those tried) for all the eigenvalues of the (*P-P-P*) model. See appendix 9.7.2 for further details.

Similar results have been obtained for the other four models: (*P-O-3*), (*P-R-2*), (*S-C-P*) and (*P-2-2*). In particular we see that the fitted values for $\nu$ drop with increasing $n$ with either $1/n$ or $1/\sqrt{n}$ corrections, suggesting that the differential approximant estimates (particularly for the larger eigenvalues) are too high.

We therefore can conclude from the above analyses that strong corrections-to-scaling occur in the scaling of largest eigenvalues of the moment of inertia (and hence in the radius of gyration) of the models in the apparent new universality class and also those in the 3d-spiral class. Furthermore, by including such corrections into the analysis the enumeration data is consistent with there being only one scaling exponent, namely that of the unrestricted SAW.

## 9.4

## Discussion

We now review the various analyses and conflicting conclusions made from those above analyses to come to some global conclusions. In the two-step restricted walk models we have examined, the radius of gyration data confirms an earlier suggestion [87] of a three-dimensional spiral universality class. Our data also supports the new suggestion that there is at least one other novel class. Poor convergence of the series analyses, based upon differential approximants, makes this conclusion contentious however. On the other hand, examination of the number of walks, $c_n$, again using differential approximants, seems to indicate that all the models considered have approximately the same value (within error) of their entropic exponent $\gamma$, and further that the scaling of $c_n$ does not contain $e^{\sqrt{n}}$ or $e^{n^{2/3}}$ factors that may invalidate the differential approximant analyses. Moreover, the symmetries of the walk rules offer little insight into the differences in their $\langle R_g^2 \rangle_n$ scaling, in distinction to two dimensions. The turning number of the rule, which we have defined here, does seem to be a better candidate as a microscopic criterion to differentiate the scaling behaviours of the TSRW models though this too is problematic. Our examination of the scaling of the eigenvalues of the inertia tensor for 5 of the models indicates that the smallest eigenvalue is the best behaved numerically and further it scales with an exponent quite close to that of the SAW universality class. While the analyses of the larger eigenvalues seem to be hampered by relatively larger corrections-to-scaling, they reflect the analysis of the radius of gyration which gives an exponent central-estimate larger than the SAW value. However, further analyses of the larger eigenvalues are *also consistent* with lower SAW-like values of the exponents.

The above facts leave us with a couple of alternate and reasonable conclusions. Of course, there are other possibilities given the moderate size of some of the errors on our analyses but the following are the most likely, consistent, and simplest conclusions. Either

1. all the TSRW models in three dimensions we examined (and probably all symmetric-mixing rules) are members of the unrestricted SAW universality class with different

corrections-to-scaling (categorised maybe by the turning number) or

2. the walk rules split into at least 3 universality classes with different values of the exponent $\nu$. The non-SAW-like classes have anisotropic exponents which are only slightly anisotropic (of the order of 5%). Moreover, all classes have one exponent (the smallest) that takes on a value very close to the 3d-SAW one.

Also we conclude that the exponent $\gamma$ for any of the TSRW models examined (and presumably all symmetric-mixing models) probably takes on the SAW value (around 1.16) or a value very close to it.

Now, given our further analysis of the moment of inertia eigenvalues, the weight of evidence favours the first conclusion we believe. In any event it would be advantageous indeed to have confirmation of some of these results using Monte Carlo simulations.

# 9.5 Appendix A: Number of symmetric-balanced rules

## 9.5.1 Calculating the total number of balanced rules

Balanced rules must have the same number of continuing steps in the positive component as in the negative component of each axial direction. On the $d$-dimensional hypercubic lattice there are a total of $2d-1$ different possible continuing steps in each of the positive and negative components of each axial direction.

To obtain a balanced rule with $k$ continuing steps in some positive axial direction we hence require that there be $k$ continuing steps in the negative axial direction. There are $\binom{2d-1}{k}$ ways of choosing the former set of $k$ steps, and the same number of ways of choosing the latter set of $k$ steps. Hence

$$\text{the number of choices of } k \text{ pairs of steps} = \binom{2d-1}{k}^2. \tag{9.44}$$

We can sum this over $k$ to get the number of rules that are balanced with respect to that axis

$$\sum_{k=0}^{2d-1} \binom{2d-1}{k}^2 = \binom{4d-2}{2d-1}, \tag{9.45}$$

by a well-known binomial coefficient result. Since there are $d$ directions on the hypercubic lattice we obtain the following result:

$$\text{the number of balanced rules on the hypercubic lattice} = \binom{4d-2}{2d-1}^d. \tag{9.46}$$

However, there are rules that are balanced that are not reverse balanced, and hence we wish to calculate the cardinality of the symmetric-balanced rule space.

### 9.5.2 Calculating the number of symmetric-balanced rules sorted according to the number of two-step configurations

In this section we do two things simultaneously: we count the number of symmetric-balanced rules and further we sort them according to the number of two-step configurations the rule contains. To do this we will introduce suitable generating functions. To accomplish the first task we make use of MacMahon's $\underset{=}{\overset{\lambda}{\Omega}}$ operator [120], which is a 'constant term' operator (see below).

We show here in detail the analysis for square lattice TSRW and then just state the result for higher-dimensional hyper-cubic lattices. We first write the $\mathbf{M}$ matrix for two dimensions (in the manner described in section 2) as

$$\mathbf{M} = \begin{bmatrix} \mathbf{rr} & 0 & \mathbf{rf} & \mathbf{rb} \\ 0 & \mathbf{ll} & \mathbf{lf} & \mathbf{lb} \\ \mathbf{fr} & \mathbf{fl} & \mathbf{ff} & 0 \\ \mathbf{br} & \mathbf{bl} & 0 & \mathbf{bb} \end{bmatrix} \tag{9.47}$$

with each variable entry taking the value of 0 or 1. That is, the TSRW rules are in bijection with matrices with binary entries constrained by the condition that the elements in the positions $(1,2)$, $(2,1)$, $(3,4)$ and $(4,3)$ are fixed to be 0. Let us call the set of all such matrices $\mathcal{M}$. The symmetric-balance conditions for the square lattice are given by

$$\sum_i M_{i,2j-1} = \sum_i M_{i,2j} \qquad j = 1,2 \tag{9.48}$$

and

$$\sum_i M_{2j-1,i} = \sum_i M_{2j,i} \qquad j = 1,2\,. \tag{9.49}$$

That is, we require the sum of the first two rows (columns) to be equal and the sum of the second two rows (columns) to be equal.

To be able to sort rules according to the number of two-step configurations (TSC) we introduce a generating function whose variable $t$ counts the number of two-step configurations. As an example, we can easily write down the TSC generating function $g_{\text{all}}(t)$ for all rules. Let us assign a variable, $t$, to each two-step configuration in the rule (which is equivalent to assigning a variable to each non-zero element of $\mathbf{M}$). The generating function is obtained by summing over all the possible allowed matrices $\mathbf{M}$, that is $\mathbf{M} \in \mathcal{M}$:

$$\begin{aligned} g_{\text{all}}(t) &= \sum_{\mathbf{M}\in\mathcal{M}} \prod_{(i,j)} t^{M_{i,j}} \\ &= \prod_{(\text{allowed } i,j)} \sum_{M_{i,j}=0,1} t^{M_{i,j}} = (1+t)^{(\#\text{ allowed } i,j)} \\ &= (1+t)^{12} = \sum_{k=0}^{12} \binom{12}{k} t^k\,. \end{aligned} \tag{9.50}$$

We can impose the symmetric-balance conditions by associating an additional weight with each element of **M** and then taking the constant term of the resultant polynomials with respect to these variables. To illustrate the method in detail, we first consider the number of balanced rules sorted according to the number of two-step configurations. This generalises the result of the previous section. The TSC generating function $g_{\text{bal}}(t)$ we require is then

$$g_{\text{bal}}(t) = \sum_{\mathbf{M}\in\mathcal{M}\left|\left(\sum_i M_{i,1}=\sum_i M_{i,2} \text{ and } \sum_i M_{i,3}=\sum_i M_{i,4}\right)\right.} \prod_{(i,j)} t^{M_{i,j}} . \tag{9.51}$$

We now show that this generating function $g_{\text{bal}}(t)$ can be written as the constant term of a generalised generating function $P(\mu_x, \mu_y; t)$ with respect to the new variables $\mu_x$ and $\mu_y$. Taking the constant term of such a series is described by the action of the operators $\underset{=}{\overset{\mu_x}{\Omega}}$ and $\underset{=}{\overset{\mu_y}{\Omega}}$ defined by MacMahon [120]. The operator $\underset{=}{\overset{\lambda}{\Omega}}$ acts on a series $f(\lambda)$ so as to remove any term in which $\lambda$ appears; *i.e.* finds the constant term (with respect to $\lambda$) of the expression upon which it acts. For example,

$$\underset{=}{\overset{\lambda}{\Omega}}\left\{(1+x\lambda)^2(1+x/\lambda)^2\right\} = 1+4x^2+x^4 . \tag{9.52}$$

In our work all the series we consider are simply polynomials.

We begin by rewriting the balance constraint on the sum over the rules as a Kronecker $\delta$ factor acting on an unconstrained sum. Hence we have

$$g_{\text{bal}}(t) = \sum_{\mathbf{M}\in\mathcal{M}} \left[\prod_{i,j} t^{M_{i,j}}\, \delta\left(\sum_i M_{i,1} - \sum_i M_{i,2}\right) \delta\left(\sum_i M_{i,3} - \sum_i M_{i,4}\right)\right] . \tag{9.53}$$

The Kronecker $\delta$ picks out of the unconstrained (finite) sum the terms satisfying the conditions. Another way to evaluate a finite sum of the form $\sum_j a_j \delta\left(f(j)\right)$ is to introduce a variable that 'counts' deviations from this constraint and takes the constant term of the resulting function. That is,

$$\sum_j a_j \delta\left(f(j)\right) = \text{Constant Term of } \lambda \text{ in} \sum_j a_j \lambda^{f(j)} . \tag{9.54}$$

Introducing variables $\mu_x$ and $\mu_y$ to take account of the two constraints we obtain

$$\begin{aligned} g_{\text{bal}}(t) &= \underset{=\,=}{\overset{\mu_x \mu_y}{\Omega\Omega}} \left\{\sum_{\mathbf{M}\in\mathcal{M}} \left(\prod_{i,j} t^{M_{i,j}} \mu_x^{\sum_i M_{i,1}-\sum_i M_{i,2}} \mu_y^{\sum_i M_{i,3}-\sum_i M_{i,4}}\right)\right\} \\ &= \underset{=\,=}{\overset{\mu_x \mu_y}{\Omega\Omega}} \left\{P(\mu_x, \mu_y; t)\right\} . \end{aligned} \tag{9.55}$$

which also provides the definition of the function $P(\mu_x, \mu_y; t)$. Note that $P(\mu_x, \mu_y; t)$ can be constructed by forming a generating function from the set $\mathcal{M}$ by weighting the (non-zero) elements in column 1 by $\mu_x$ and those in column 2 by $1/\mu_x$ and weighting the (non-zero) elements in column 3 by $\mu_y$ and those in column 4 by $1/\mu_y$, as well as weighting any (non-zero) element by $t$. Now we can easily evaluate $P(\mu_x, \mu_y; t)$ as a product in the same way we previously found $g_{\text{all}}(t)$. Hence, examining the operand $P(\mu_x, \mu_y; t)$ in more detail gives

$$\begin{aligned} P(\mu_x, \mu_y; t) &= \sum_{\mathbf{M}\in\mathcal{M}} \left( \prod_i (\mu_x t)^{M_{i,1}} (t/\mu_x)^{M_{i,2}} (\mu_y t)^{M_{i,3}} (t/\mu_y)^{M_{i,4}} \right) \\ &= (1+\mu_x t)^3 (1+t/\mu_x)^3 (1+\mu_y t)^3 (1+t/\mu_y)^3 \,. \end{aligned} \tag{9.56}$$

In this case one can explicitly evaluate the constant term as

$$\begin{aligned} g_{\text{bal}}(t) &= \underset{=}{\overset{\mu_x}{\Omega}} \left\{ \left( \sum_{k=0}^{3} \binom{3}{k} (\mu_x t)^k \right) \left( \sum_{k=0}^{3} \binom{3}{k} (t/\mu_x)^k \right) \right\} \\ &\quad \times \underset{=}{\overset{\mu_y}{\Omega}} \left\{ \left( \sum_{k=0}^{3} \binom{3}{k} (\mu_y t)^k \right) \left( \sum_{k=0}^{3} \binom{3}{k} (t/\mu_y)^k \right) \right\} \\ &= \left( \sum_{k=0}^{3} t^{2k} \binom{3}{k}^2 \right)^2 . \end{aligned} \tag{9.57}$$

The coefficient of $t^n$ is the number of balanced configurations of $\mathcal{M}$ with $n$ two-step configurations. Note that putting $t = 1$ recovers the result (9.46) in the case $d = 2$.

We now consider the generating function of rules that are symmetric-balanced in two dimensions. We define a generating function that counts such rules according to the number of two-step configurations as

$$g_{2,\text{sym-bal}}(t) = \sum_{\mathbf{M}\in\mathcal{M}}^{*} \prod_{i,j} t^{M_{i,j}} \tag{9.58}$$

where the $*$ on the sum means that the sum is constrained by both conditions (9.48) and (9.49). Again this can be written as a multiple constant term expression,

$$g_{2,\text{sym-bal}}(t) = \underset{=}{\overset{\mu_x}{\Omega}}\underset{=}{\overset{\mu_y}{\Omega}}\underset{=}{\overset{\lambda_x}{\Omega}}\underset{=}{\overset{\lambda_y}{\Omega}} \left\{ P_{2,\text{sym-bal}}(\mu_x, \mu_y, \lambda_x, \lambda_y; t) \right\} \tag{9.59}$$

where

$$\begin{aligned} P_{2,\text{sym-bal}}(\mu_x, \mu_y, \lambda_x, \lambda_y; t) &= (1+t\mu_x\lambda_x)(1+t\mu_y\lambda_x)(1+t\lambda_x/\mu_y) \times \\ &\quad (1+t/\mu_x\lambda_x)(1+t\mu_y/\lambda_x)(1+t/\mu_y\lambda_x) \times \\ &\quad (1+t\mu_x\lambda_y)(1+t\lambda_y/\mu_x)(1+t\mu_y\lambda_y) \times \\ &\quad (1+t\mu_x/\lambda_y)(1+t/\mu_x\lambda_y)(1+t/\mu_y\lambda_y) \,. \end{aligned} \tag{9.60}$$

Using the Maple™ [6] to expand the polynomial and take the constant term, we find that

$$g_{2,\text{sym-bal}}(t) = 1 + 6t^2 + 19t^4 + 28t^6 + 19t^8 + 6t^{10} + t^{12} \,. \tag{9.61}$$

[6] Maple™ is a registered trademark of Waterloo Maple Software.

As constructed, the coefficient of $t^n$ represents the number of symmetric-balanced TSRW rules made up of $n$ two-step configurations. This can be explicitly seen, after some counting, from the catalogue in figure 9.3. By a similar process as that outlined above we obtain the following results for three dimensions

$$\begin{aligned} g_{3,\text{sym-bal}}(t) &= \underset{=}{\overset{\mu_x,\mu_y,\mu_z,\lambda_x,\lambda_y,\lambda_z}{\Omega}} \{P_{3,\text{sym-bal}}(\mu_x, \mu_y, \mu_z, \lambda_x, \lambda_y, \lambda_z; t)\} \\ &= t^{30} + 15t^{28} + 177t^{26} + 1519t^{24} + 8457t^{22} + 30183t^{20} + \\ &\quad 69829t^{18} + 105867t^{16} + 105867t^{14} + 69829t^{12} + \\ &\quad 30183t^{10} + 8457t^8 + 1519t^6 + 177t^4 + 15t^2 + 1 \end{aligned} \tag{9.62}$$

and four dimensions

$$\begin{aligned} g_{4,\text{sym-bal}}(t) &= \underset{=}{\overset{\mu_i,\lambda_i;\{i=1\ldots4\}}{\Omega}} \{P_{4,\text{sym-bal}}(\mu_i, \lambda_i; t)\} \\ &= t^{56} + 28t^{54} + 738t^{52} + 16268t^{50} + 274907t^{48} + \\ &\quad \cdots + 94001836824t^{28} + \cdots \\ &\quad +274907t^8 + 16268t^6 + 738t^4 + 28t^2 + 1\,. \end{aligned} \tag{9.63}$$

Substituting $t = 1$ into the generating functions gives the total number of symmetric-balanced rules. Hence we find[7] that there are

- 80 TSRW rules in two dimensions,
- 432096 TSRW rules in three dimensions and
- 478340593664 TSRW rules in four dimensions.

This subset of symmetric-balanced two-step rules is manageable in two dimensions (we have in fact catalogued them completely in this chapter). However, in three dimensions it is far too large to be examined in its entirety.

[7] This construction can be applied in any dimension, however the computation of similar results for higher dimensions seems to require more (a lot more!) computer time and memory than was at hand.

# 9.6

# Appendix B: Exact enumeration data

## 9.6.1 Radius of gyration and number of walk configuration tables

| | $(Rot\text{-}\pi)$ | | $(P\text{-}P\text{-}3)$ | | $(P\text{-}P\text{-}D)$ | |
|---|---|---|---|---|---|---|
| $n$ | $c_n$ | $\langle R_g^2\rangle_n$ | $c_n$ | $\langle R_g^2\rangle_n$ | $c_n$ | $\langle R_g^2\rangle_n$ |
| 1 | 6 | .2500000000 | 6 | 0.250000 | 6 | .2500000000 |
| 2 | 18 | .5185185185 | 20 | 0.511111 | 18 | .5185185185 |
| 3 | 54 | .7870370370 | 66 | 0.772727 | 54 | .7962962963 |
| 4 | 158 | 1.062278481 | 206 | 1.060971 | 154 | 1.112727273 |
| 5 | 466 | 1.330591321 | 652 | 1.352079 | 446 | 1.444319880 |
| 6 | 1338 | 1.616180104 | 2012 | 1.667018 | 1266 | 1.810233098 |
| 7 | 3886 | 1.894042717 | 6264 | 1.984924 | 3626 | 2.189878654 |
| 8 | 11082 | 2.189255850 | 19254 | 2.321767 | 10282 | 2.598303160 |
| 9 | 31842 | 2.479427800 | 59442 | 2.662321 | 29262 | 3.020260406 |
| 10 | 90542 | 2.782749105 | 182148 | 3.018780 | 82790 | 3.467095379 |
| 11 | 258466 | 3.083967846 | 559568 | 3.378845 | 234674 | 3.925950941 |
| 12 | 733190 | 3.395616965 | 1710476 | 3.752893 | 662518 | 4.406914942 |
| 13 | 2084726 | 3.706346879 | 5237592 | 4.130143 | 1872906 | 4.897985755 |
| 14 | 5902350 | 4.026035244 | 15980914 | 4.519812 | 5279098 | 5.408667159 |
| 15 | 16738270 | 4.345208279 | 48822216 | 4.912359 | 14895186 | 5.927984092 |
| 16 | 47319166 | 4.672362374 | 148757054 | 5.316068 | 41934586 | 6.464774052 |
| 17 | 133935010 | 4.999214463 | 453683704 | 5.722368 | 118150286 | 7.009094165 |
| 18 | 378189902 | 5.333289679 | 1380795336 | 6.138830 | 332311930 | 7.569169389 |
| 19 | 1068895606 | 5.667227224 | 4205655986 | 6.557627 | 935234466 | 8.135866302 |
| 20 | 3015366794 | 6.007756322 | 12788431474 | 6.985764 | 2628389278 | 8.716905437 |
| 21 | 8512718274 | 6.348276986 | 38910558974 | 7.416014 | 7390474750 | 9.303800493 |
| 22 | 23995774102 | 6.694862106 | 118229769376 | 7.854911 | 20756622062 | 9.903846043 |
| 23 | | | 359424248858 | 8.295731 | | |
| 24 | | | 1091427480250 | 8.744604 | | |
| 25 | | | 3315651798324 | 9.195238 | | |

Table 9.7: The enumeration data, $c_n$ and $\langle R_g^2\rangle_n$, for the rules $(Rot\text{-}\pi)$, $(P\text{-}P\text{-}3)$ and $(P\text{-}P\text{-}D)$.

| | (*S-C-3*) | | (*S-C-P*) | | (*3-3-C*) | |
|---|---|---|---|---|---|---|
| $n$ | $c_n$ | $\langle R_g^2\rangle_n$ | $c_n$ | $\langle R_g^2\rangle_n$ | $c_n$ | $\langle R_g^2\rangle_n$ |
| 1 | 6 | 0.250000 | 6 | .2500000000 | 6 | 0.250000 |
| 2 | 24 | 0.500000 | 22 | .5050505050 | 22 | 0.505051 |
| 3 | 90 | 0.766667 | 78 | .7692307690 | 80 | 0.771875 |
| 4 | 324 | 1.059259 | 266 | 1.056842106 | 284 | 1.060845 |
| 5 | 1166 | 1.366447 | 910 | 1.354273505 | 1014 | 1.359358 |
| 6 | 4138 | 1.694716 | 3054 | 1.672266549 | 3564 | 1.678142 |
| 7 | 14730 | 2.031848 | 10310 | 1.993913676 | 12588 | 2.002155 |
| 8 | 51992 | 2.386510 | 34446 | 2.331570689 | 44098 | 2.341145 |
| 9 | 183898 | 2.747547 | 115450 | 2.672526808 | 154832 | 2.684732 |
| 10 | 646980 | 3.122930 | 384530 | 3.026504010 | 540770 | 3.040744 |
| 11 | 2279702 | 3.502933 | 1283462 | 3.382982338 | 1891584 | 3.400228 |
| 12 | 8002976 | 3.895122 | 4265822 | 3.750604402 | 6592486 | 3.770323 |
| 13 | 28127418 | 4.290869 | 14199618 | 4.120239122 | 23001542 | 4.143167 |
| 14 | 98585096 | 4.697309 | 47120838 | 4.499743779 | 80037948 | 4.525304 |
| 15 | 345848306 | 5.106591 | 156545474 | 4.880926243 | 278740232 | 4.909733 |
| 16 | 1210704274 | 5.525411 | 518858122 | 5.270979045 | 968743336 | 5.302460 |
| 17 | 4241348770 | 5.946576 | 1721232166 | 5.662463560 | 3369017390 | 5.697144 |
| 18 | 14833284544 | 6.376375 | 5699369614 | 6.062020580 | 11697449542 | 6.099341 |
| 19 | 51907058582 | 6.808156 | 18885164058 | 6.462824345 | 40635868918 | 6.503236 |
| 20 | 181392476966 | 7.247841 | 62483445082 | 6.871045435 | 140979332596 | 6.914013 |
| 21 | 634197818374 | 7.689232 | 206852302966 | 7.280367595 | | |
| 22 | 2214804822718 | 8.137924 | 683942288222 | 7.696558450 | | |
| 23 | 7737946227490 | 8.588107 | | | | |

| | (*S-P-3*) | | (*S-P-3*)-cont | | |
|---|---|---|---|---|---|
| $n$ | $c_n$ | $\langle R_g^2\rangle_n$ | $n$ | $c_n$ | $\langle R_g^2\rangle_n$ |
| 1 | 6 | 0.250000 | 10 | 1105052 | 2.894842 |
| 2 | 24 | 0.500000 | 11 | 4165768 | 3.232301 |
| 3 | 96 | 0.752604 | 12 | 15635564 | 3.580525 |
| 4 | 368 | 1.029130 | 13 | 58773288 | 3.930671 |
| 5 | 1422 | 1.311260 | 14 | 220229536 | 4.290298 |
| 6 | 5392 | 1.612548 | 15 | 826135272 | 4.651580 |
| 7 | 20562 | 1.917156 | 16 | 3091645402 | 5.021351 |
| 8 | 77590 | 2.237052 | 17 | 11579713514 | 5.392576 |
| 9 | 293760 | 2.559677 | 18 | 43290642466 | 5.771503 |

Table 9.8: The enumeration data, $c_n$ and $\langle R_g^2\rangle_n$, for the rules (*S-C-3*), (*S-C-P*), (*3-3-C*) and (*S-P-3*).

| | (*P-O-3*) | | (*P-3-3*) | | (*P-2-2*) | |
|---|---|---|---|---|---|---|
| $n$ | $c_n$ | $\langle R_g^2\rangle_n$ | $c_n$ | $\langle R_g^2\rangle_n$ | $c_n$ | $\langle R_g^2\rangle_n$ |
| 1 | 6 | .2500000000 | 6 | 0.250000 | 3 | 0.250000 |
| 2 | 16 | .5277777780 | 22 | 0.505051 | 10 | 0.488889 |
| 3 | 44 | .8125000000 | 78 | 0.769231 | 32 | 0.734375 |
| 4 | 112 | 1.147142857 | 266 | 1.056842 | 98 | 1.002449 |
| 5 | 290 | 1.489080460 | 914 | 1.351872 | 302 | 1.275754 |
| 6 | 730 | 1.862230920 | 3072 | 1.670095 | 905 | 1.572804 |
| 7 | 1858 | 2.238125673 | 10388 | 1.992612 | 2731 | 1.872414 |
| 8 | 4644 | 2.641571230 | 34696 | 2.333659 | 8121 | 2.190149 |
| 9 | 11692 | 3.045703045 | 116326 | 2.677870 | 24254 | 2.510089 |
| 10 | 29128 | 3.471307038 | 387094 | 3.037176 | 71801 | 2.844642 |
| 11 | 72866 | 3.898065933 | 1291232 | 3.399126 | 213168 | 3.181094 |
| 12 | 181092 | 4.342164570 | 4285502 | 3.773997 | 629044 | 3.530483 |
| 13 | 451246 | 4.787541630 | 14247504 | 4.151048 | 1860191 | 3.881363 |
| 14 | 1119492 | 5.247478820 | 47195288 | 4.539472 | 5476975 | 4.243746 |
| 15 | 2782326 | 5.708806865 | 156532896 | 4.929770 | 16151357 | 4.607443 |
| 16 | 6893472 | 6.182730795 | 517758628 | 5.330238 | 47474808 | 4.981465 |
| 17 | 17101294 | 6.658119150 | 1714230392 | 5.732355 | 139716218 | 5.356659 |
| 18 | 42325616 | 7.144718770 | 5663536142 | 6.143698 | 410143001 | 5.741247 |
| 19 | 104857966 | 7.632773830 | 18725774596 | 6.556509 | 1205156331 | 6.126905 |
| 20 | 259302146 | 8.131019980 | 61809095034 | 6.977778 | 3534113551 | 6.521186 |
| 21 | 641714298 | 8.630686255 | 204143781018 | 7.400373 | 10371867145 | 6.916460 |
| 22 | 1585767184 | 9.139738495 | 673309667636 | 7.830787 | 30389668612 | 7.319704 |
| 23 | 3921055588 | 9.650150405 | 2221867592366 | 8.262410 | 89099312437 | 7.723878 |
| 24 | 9683676170 | 10.16930800 | 7323479170494 | 8.701308 | 260877974233 | 8.135467 |
| 25 | 23927486656 | 10.68974044 | | | | |
| 26 | 59062356252 | 11.21838694 | | | | |
| 27 | 145850988968 | 11.74822225 | | | | |
| 28 | 359856328600 | 12.28581177 | | | | |

Table 9.9: The enumeration data, $c_n$ and $\langle R_g^2\rangle_n$, for the rules (*P-O-3*), (*P-3-3*) and (*P-2-2*).

| | (*P-P-3*) | | (*P-R-2*) | | (*P-P-P*) | |
|---|---|---|---|---|---|---|
| $n$ | $c_n$ | $\langle R_g^2 \rangle_n$ | $c_n$ | $\langle R_g^2 \rangle_n$ | $c_n$ | $\langle R_g^2 \rangle_n$ |
| 1 | 6 | 0.250000 | 6 | .2500000000 | 6 | 0.250000 |
| 2 | 20 | 0.511111 | 18 | .4938271605 | 18 | 0.518519 |
| 3 | 66 | 0.772727 | 54 | .7361111110 | 54 | 0.777778 |
| 4 | 206 | 1.060971 | 150 | 1.009066667 | 150 | 1.059200 |
| 5 | 652 | 1.352079 | 426 | 1.282603026 | 426 | 1.325117 |
| 6 | 2012 | 1.667018 | 1170 | 1.582766440 | 1158 | 1.613620 |
| 7 | 6264 | 1.984924 | 3250 | 1.884923077 | 3204 | 1.891912 |
| 8 | 19254 | 2.321767 | 8890 | 2.206907470 | 8682 | 2.187736 |
| 9 | 59442 | 2.662321 | 24444 | 2.532434954 | 23724 | 2.481578 |
| 10 | 182148 | 3.018780 | 66598 | 2.875134116 | 64194 | 2.788945 |
| 11 | 559568 | 3.378845 | 182044 | 3.221091696 | 174378 | 3.097380 |
| 12 | 1710476 | 3.752893 | 494462 | 3.582785943 | 470856 | 3.418129 |
| 13 | 5237592 | 4.130143 | 1346212 | 3.947356701 | 1274430 | 3.740403 |
| 14 | 15980914 | 4.519812 | 3648594 | 4.326093949 | 3434826 | 4.074125 |
| 15 | 48822216 | 4.912359 | 9905610 | 4.707547609 | 9272346 | 4.409709 |
| 16 | 148757054 | 5.316068 | 26803048 | 5.101762050 | 24953004 | 4.755764 |
| 17 | 453683704 | 5.722368 | 72618674 | 5.498518505 | 67230288 | 5.103799 |
| 18 | 1380795336 | 6.138830 | 196243194 | 5.906916350 | 180705126 | 5.461477 |
| 19 | 4205655986 | 6.557627 | 530861042 | 6.317670205 | 486152604 | 5.821086 |
| 20 | 12788431474 | 6.985764 | 1433106984 | 6.739154200 | 1305430884 | 6.189624 |
| 21 | 38910558974 | 7.416014 | 3871966934 | 7.162813220 | 3507947838 | 6.560016 |
| 22 | 118229769376 | 7.854911 | 10443886572 | 7.596427065 | 9412114986 | 6.938709 |
| 23 | 359424248858 | 8.295731 | 28189364062 | 8.032055660 | 25268587338 | 7.319174 |
| 24 | 1091427480250 | 8.744604 | 75981934450 | 8.476964825 | 67752451146 | 7.707392 |
| 25 | 3315651798324 | 9.195238 | 204918079282 | 8.923748270 | 181754458194 | 8.097294 |
| 26 | | | 552010606124 | 9.379224335 | 487060621596 | 8.494472 |
| 27 | | | | | 1305761069730 | 8.893243 |
| 28 | | | | | 3497441209182 | 9.298868 |
| 29 | | | | | 9371171057352 | 9.705999 |

Table 9.10: The enumeration data, $c_n$ and $\langle R_g^2 \rangle_n$, for the rules (*P-P-3*), (*P-R-2*) and (*P-P-P*).

### 9.6.2 Moment of inertia eigenvalue enumeration tables

| | (*S-C-P*) | | | (*P-O-3*) | | |
|---|---|---|---|---|---|---|
| $n$ | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ |
| 1 | .1666666667 | .1666666667 | .1666666667 | .166666666667 | .166666666667 | .166666666667 |
| 2 | .3232323232 | .3232323232 | .3636363636 | .302276833646 | .364389833021 | .388888888889 |
| 3 | .4839743590 | .5836984453 | .4707887341 | .426824397156 | .569876200940 | .628299401917 |
| 4 | .6556390977 | .8281265757 | .6299185371 | .566948065248 | .814084994660 | .913252654398 |
| 5 | .8321123321 | 1.083682865 | .7927518113 | .698817526155 | 1.07055653765 | 1.20878685577 |
| 6 | 1.018670730 | 1.358246075 | .9676162926 | .834240876666 | 1.35502001927 | 1.53520094359 |
| 7 | 1.206237876 | 1.636945906 | 1.144643570 | .965014568266 | 1.64539790396 | 1.86583887337 |
| 8 | 1.401436351 | 1.930158848 | 1.331546178 | 1.09943234575 | 1.95967965142 | 2.22403046199 |
| 9 | 1.597762148 | 2.226718876 | 1.520572592 | 1.23112509025 | 2.27586578203 | 2.58441521738 |
| 10 | 1.800378395 | 2.535047728 | 1.717581896 | 1.36639773324 | 2.60995149821 | 2.96626484468 |
| 11 | 2.003823794 | 2.845912484 | 1.916228398 | 1.50016239182 | 2.94527512399 | 3.35069434886 |
| 12 | 2.212733910 | 3.166808488 | 2.121666404 | 1.63751026726 | 3.29451156412 | 3.75230730827 |
| 13 | 2.422322095 | 3.489717948 | 2.328438200 | 1.77386437145 | 3.64475300486 | 4.15646588292 |
| 14 | 2.636818108 | 3.821491853 | 2.541177597 | 1.91361659394 | 4.00636209865 | 4.57497894901 |
| 15 | 2.851890731 | 4.154931390 | 2.755030364 | 2.05268553596 | 4.36899823418 | 4.99592995707 |
| 16 | 3.071423508 | 4.496325628 | 2.974208952 | 2.19487704734 | 4.74131512204 | 5.42926942078 |
| 17 | 3.291456083 | 4.839133151 | 3.194337891 | 2.33659198558 | 5.11473588958 | 5.86491042151 |
| 18 | 3.515583085 | 5.189171582 | 3.419286494 | 2.48117103722 | 5.49671470589 | 6.31155179334 |
| 19 | 3.740151811 | 5.540434799 | 3.645062085 | 2.62540633802 | 5.87982499605 | 6.76031632378 |
| 20 | 3.968508782 | 5.898336079 | 3.875246007 | 2.77228050171 | 6.27070584816 | 7.21905361238 |
| 21 | 4.197260774 | 6.257314900 | 4.106159520 | 2.91890872388 | 6.66271834526 | 7.67974543712 |
| 22 | 4.429540412 | 6.622436345 | 4.341140145 | 3.06797945586 | 7.06190067532 | 8.14959685536 |
| 23 | | | | 3.21687407350 | 7.46218518484 | 8.62124155310 |
| 24 | | | | 3.36804906946 | 7.86917066793 | 9.10139626525 |
| 25 | | | | 3.51909521920 | 8.27720172517 | 9.58318392935 |
| 26 | | | | 3.67228451329 | 8.69154694518 | 10.0729424218 |
| 27 | | | | 3.82538054292 | 9.10687438372 | 10.5641895689 |
| 28 | | | | 3.98050021705 | 9.52818050950 | 11.0629428101 |

Table 9.11: Eigenvalues of the moment of inertia tensor computed about the centre of mass for the (*S-C-P*) and (*P-O-3*) rules.

| | $(P\text{-}2\text{-}2)$ | | | $(P\text{-}R\text{-}2)$ | | |
|---|---|---|---|---|---|---|
| $n$ | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ |
| 1 | .1666666667 | .1666666667 | .1666666667 | .1666666667 | .1666666667 | .1666666667 |
| 2 | .3061167806 | .3111111111 | .3605498861 | .2942928684 | .3209876543 | .3723737985 |
| 3 | .4380603326 | .4570312500 | .5736584178 | .4116361451 | .4769444645 | .5836416133 |
| 4 | .5768549300 | .6171428575 | .8109001732 | .5335997647 | .6539704927 | .8305630769 |
| 5 | .7147827913 | .7801507133 | 1.056574958 | .6515748546 | .8336572627 | 1.079973934 |
| 6 | .8619902703 | .9586341077 | 1.324983923 | .7745233588 | 1.034225881 | 1.356783641 |
| 7 | 1.008795944 | 1.138534983 | 1.597496983 | .8959786064 | 1.237000753 | 1.636866796 |
| 8 | 1.162874755 | 1.330319099 | 1.887103536 | 1.021468006 | 1.455531410 | 1.936815524 |
| 9 | 1.317037920 | 1.523442032 | 2.179698169 | 1.146692636 | 1.677217902 | 2.240959373 |
| 10 | 1.477216020 | 1.725865771 | 2.486202909 | 1.276102492 | 1.912047787 | 2.562117957 |
| 11 | 1.637644958 | 1.929534679 | 2.795008619 | 1.405388728 | 2.149897142 | 2.886897521 |
| 12 | 1.803541690 | 2.141291486 | 3.116133542 | 1.538733190 | 2.399671785 | 3.227166913 |
| 13 | 1.969670042 | 2.354060265 | 3.438996579 | 1.672079918 | 2.652093350 | 3.570540133 |
| 14 | 2.140712992 | 2.573986173 | 3.772793127 | 1.809124263 | 2.915259538 | 3.927804101 |
| 15 | 2.312008455 | 2.794817983 | 4.108059365 | 1.946291614 | 3.180868289 | 4.287935315 |
| 16 | 2.487741851 | 3.022057624 | 4.453129695 | 2.086834798 | 3.456143376 | 4.660545924 |
| 17 | 2.663728951 | 3.250112163 | 4.799476692 | 2.227556192 | 3.733680281 | 5.035800539 |
| 18 | 2.843771098 | 3.483990690 | 5.154731719 | 2.371400424 | 4.020017670 | 5.422414610 |
| 19 | 3.024065642 | 3.718614760 | 5.511129042 | 2.515443402 | 4.308440799 | 5.811456215 |
| 20 | 3.208093348 | 3.958584700 | 5.875693123 | 2.662393290 | 4.604966665 | 6.210948447 |
| 21 | 3.392371559 | 4.199246804 | 6.241301327 | 2.809548671 | 4.903410105 | 6.612667668 |
| 22 | 3.580107306 | 4.444850763 | 6.614449240 | 2.959418683 | 5.209367525 | 7.024067925 |
| 23 | 3.768090188 | 4.691102500 | 6.988563927 | 3.109495958 | 5.517093898 | 7.437521480 |
| 24 | 3.959295468 | 4.941953028 | 7.369686415 | 3.262117597 | 5.831822352 | 7.859989694 |
| 25 | | | | 3.414944831 | 6.148190665 | 8.284361061 |
| 26 | | | | 3.570166948 | 6.471113948 | 8.717167785 |

Table 9.12: Eigenvalues of the moment of inertia tensor computed about the centre of mass for the $(P\text{-}2\text{-}2)$ and $(P\text{-}R\text{-}2)$ rules.

| | ($P$-$P$-$P$) | |
|---|---|---|
| $n$ | $\lambda_1$ | $\lambda_2 = \lambda_3$ |
| 1 | .1666666667 | .1666666667 |
| 2 | .3456790123 | .3456790123 |
| 3 | .5092592595 | .5231481481 |
| 4 | .6741333341 | .7221333334 |
| 5 | .8182055301 | .9160146058 |
| 6 | .9626731535 | 1.132282965 |
| 7 | 1.094159643 | 1.344832631 |
| 8 | 1.225643519 | 1.574914466 |
| 9 | 1.352100826 | 1.805527734 |
| 10 | 1.479217568 | 2.049336503 |
| 11 | 1.604029273 | 2.295365288 |
| 12 | 1.730382624 | 2.552937955 |
| 13 | 1.855199371 | 2.812803563 |
| 14 | 1.981875267 | 3.083187469 |
| 15 | 2.107554467 | 3.355931295 |
| 16 | 2.235127971 | 3.638200261 |
| 17 | 2.362027099 | 3.922785165 |
| 18 | 2.490795277 | 4.216079012 |
| 19 | 2.619076033 | 4.511547856 |
| 20 | 2.749167507 | 4.815040036 |
| 21 | 2.878906185 | 5.120562709 |
| 22 | 3.010385583 | 5.433515799 |
| 23 | 3.141610116 | 5.748368769 |
| 24 | 3.274505289 | 6.070139649 |
| 25 | 3.407213837 | 6.393687511 |
| 26 | 3.541525172 | 6.723709674 |
| 27 | 3.675698752 | 7.055393692 |
| 28 | 3.811409808 | 7.393163447 |

Table 9.13: Eigenvalues of the moment of inertia tensor computed about the centre of mass for the ($P$-$P$-$P$) rule. Note that the largest eigenvalue of the ($P$-$P$-$P$) model has degeneracy of 2.

9.7

# Appendix C: Tables of fits to scaling forms

In this section we include the tables of coefficients from the exact local fittings of various scaling forms for both the numbers of walk configurations and the eigenvalues of the moment of inertia tensor.

## 9.7.1 Exact fitting analysis of the number of walk configurations

In table 9.14 there are the results of 3 fits to the numbers of walks, $c_n$ for the ($P$-$P$-$P$) model. The top third of the table gives the coefficients of a linear fit using $n$, $n-2$ and $n-4$ for $n = 20, \ldots, 28$ to the form (9.27), which is the canonical scaling form for self-avoiding walks. The middle of the table and the bottom give similar fits to the forms (9.28) and (9.29) respectively, using one more term, $n-6$, per fit. In each case the value of the exponent $\gamma$ is allowed to vary. In table 9.15 the same three fitting forms are used but this time the value of the exponent $\gamma$ is fixed to be the (exact enumeration) SAW estimate of 1.161 at each $n$ (and hence one less term of the series is needed for the fit at each length). In both cases the addition of $1/\sqrt{n}$ or $1/n^{2/3}$ corrections only worsen the stability of the other coefficients (slightly). More to the point, the coefficients of these corrections are small in all cases and though quite unstable, they appear to be decreasing in magnitude as $n$ increases (but not monotonically). Given their apparent size, it is likely that these corrections are spurious; we have included both fits for completeness. Whether a value of $\gamma$ closer to 1.16 or 1.19 produces a better fit is a matter for debate.

| $n$ | $a_1$ | $\log\mu$ | $\gamma - 1$ | corrections | $a_2$ |
|---|---|---|---|---|---|
| 20 | .8324756081 | .9783192165 | .1972603078 | | |
| 21 | .8455488938 | .9785440966 | .1915972483 | | |
| 22 | .8373399541 | .9784511613 | .1947556593 | | |
| 23 | .8483239152 | .9786137500 | .1902053258 | | |
| 24 | .8409529395 | .9785354534 | .1929868688 | | |
| 25 | .8503887379 | .9786587025 | .1892170502 | | |
| 26 | .8439695129 | .9785969601 | .1915731899 | | |
| 27 | .8520667855 | .9786908435 | .1884461032 | | |
| 28 | .8466883266 | .9786460255 | .1903471629 | | |
| 20 | .8386930908 | .9813251606 | .2511716395 | $\sqrt{n}$ | $-0.0509466361$ |
| 21 | .8470176349 | .9798194475 | .2157468764 | $\sqrt{n}$ | $-0.0222091673$ |
| 22 | .8377676328 | .9808777278 | .2431470492 | $\sqrt{n}$ | $-0.0433633175$ |
| 23 | .8471827158 | .9799644881 | .2184954545 | $\sqrt{n}$ | $-0.0247359243$ |
| 24 | .8376594009 | .9802556927 | .2307406980 | $\sqrt{n}$ | $-0.0322466960$ |
| 25 | .8474684875 | .9796235445 | .2113569120 | $\sqrt{n}$ | $-0.0184932533$ |
| 26 | .8381955848 | .9799756433 | .2245944906 | $\sqrt{n}$ | $-0.0269970353$ |
| 27 | .8480120233 | .9794441350 | .2072396293 | $\sqrt{n}$ | $-0.0150543115$ |
| 28 | .8387374396 | .9798462623 | .2214956491 | $\sqrt{n}$ | $-0.0244635323$ |
| 20 | .8167669269 | .9828759076 | .2381152320 | $n^{\frac{2}{3}}$ | $-0.0268477019$ |
| 21 | .8375366772 | .9804710583 | .2098411804 | $n^{\frac{2}{3}}$ | $-0.0115609483$ |
| 22 | .8193514045 | .9821262157 | .2313935557 | $n^{\frac{2}{3}}$ | $-0.0224305704$ |
| 23 | .8367633784 | .9806570383 | .2116019897 | $n^{\frac{2}{3}}$ | $-0.0126767502$ |
| 24 | .8241918436 | .9811373146 | .2215348590 | $n^{\frac{2}{3}}$ | $-0.0163948459$ |
| 25 | .8398157465 | .9801174498 | .2059519375 | $n^{\frac{2}{3}}$ | $-0.0093291643$ |
| 26 | .8271044829 | .9806847143 | .2165707673 | $n^{\frac{2}{3}}$ | $-0.0135432894$ |
| 27 | .8419078052 | .9798288918 | .2026412139 | $n^{\frac{2}{3}}$ | $-0.0074836755$ |
| 28 | .8289702197 | .9804538038 | .2138057326 | $n^{\frac{2}{3}}$ | $-0.0120453271$ |

Table 9.14: Linear fit of the enumerations, $c_n$, for the rule ($P$-$P$-$P$), to various scaling forms, with and without corrections.

| $n$ | $a_1$ | $\log\mu$ | $\gamma-1$ | corrections | $a_2$ |
|---|---|---|---|---|---|
| 20 | .9028978209 | .9802294140 | 0.161 | | |
| 21 | .9065489941 | .9800752350 | 0.161 | | |
| 22 | .9062902192 | .9800597940 | 0.161 | | |
| 23 | .9093432958 | .9799421730 | 0.161 | | |
| 24 | .9092103892 | .9799270590 | 0.161 | | |
| 25 | .9118060176 | .9798350981 | 0.161 | | |
| 26 | .9117668293 | .9798205410 | 0.161 | | |
| 27 | .9140085907 | .9797469950 | 0.161 | | |
| 28 | .9140311693 | .9797334510 | 0.161 | | |
| 20 | .8282929603 | .9762973926 | 0.161 | $\sqrt{n}$ | 0.0342666881 |
| 21 | .8436960392 | .9769286659 | 0.161 | $\sqrt{n}$ | 0.0281350308 |
| 22 | .8370434575 | .9767585843 | 0.161 | $\sqrt{n}$ | 0.0302475055 |
| 23 | .8494932420 | .9772188950 | 0.161 | $\sqrt{n}$ | 0.0255399818 |
| 24 | .8437391859 | .9770777906 | 0.161 | $\sqrt{n}$ | 0.0273227611 |
| 25 | .8541199161 | .9774294158 | 0.161 | $\sqrt{n}$ | 0.0235656305 |
| 26 | .8493225760 | .9773207765 | 0.161 | $\sqrt{n}$ | 0.0249926735 |
| 27 | .8579899424 | .9775908142 | 0.161 | $\sqrt{n}$ | 0.0219846353 |
| 28 | .8541789041 | .9775151871 | 0.161 | $\sqrt{n}$ | 0.0230489648 |
| 20 | .8464170165 | .9742748997 | 0.161 | $n^{\frac{2}{3}}$ | 0.0238286629 |
| 21 | .8589863846 | .9753123551 | 0.161 | $n^{\frac{2}{3}}$ | 0.0193890739 |
| 22 | .8539157434 | .9750654537 | 0.161 | $n^{\frac{2}{3}}$ | 0.0206650354 |
| 23 | .8640994381 | .9758243720 | 0.161 | $n^{\frac{2}{3}}$ | 0.0173047187 |
| 24 | .8597299761 | .9756198890 | 0.161 | $n^{\frac{2}{3}}$ | 0.0183709537 |
| 25 | .8682239511 | .9761997433 | 0.161 | $n^{\frac{2}{3}}$ | 0.0157272307 |
| 26 | .8646044552 | .9760442026 | 0.161 | $n^{\frac{2}{3}}$ | 0.0165611703 |
| 27 | .8717111764 | .9764906343 | 0.161 | $n^{\frac{2}{3}}$ | 0.0144687926 |
| 28 | .8688510450 | .9763842349 | 0.161 | $n^{\frac{2}{3}}$ | 0.0150699597 |

Table 9.15: Linear fit of the enumerations, $c_n$, for the rule $(P\text{-}P\text{-}P)$, to various scaling forms, with and without corrections, and with $\gamma$ fixed at the $SAW$ value.

### 9.7.2 Exact fitting analysis of the moment of inertia eigenvalues

In tables 9.16 and 9.17 are the fits to the smallest eigenvalue, $\lambda_1(n)$, of the moment of inertia matrix for the ($P$-$P$-$P$) model using the forms (9.37) and (9.38) respectively. The form (9.37) includes $1/n$ type corrections while the form (9.38) includes $1/\sqrt{n}$ type corrections. The number of terms of the series used varied according to the number of free coefficients: *e.g.* if there were 3 free coefficients then the three terms $n$, $n-2$ and $n-4$ were utilised. In both cases the top of the table gives coefficients for fits where the value of $2\nu_1$ is free, while the middle and bottom sections give coefficients that have $2\nu_1$ fixed. We tried two fixed values: the SAW estimate of 1.19 and the best $\langle R_g^2 \rangle_n$ differential approximant estimate for the ($P$-$P$-$P$) model of 1.29. Coefficients from fits for $n = 20$ to $n = 28$ are included. The standard correction fit in table 9.16, with $2\nu_1$ free, gives values of $2\nu_1$ close to the SAW value, if slightly increasing. The coefficient, $a_2$, of the $1/n$ term is less stable if the value of $2\nu_1$ is fixed at 1.29 than if it is fixed at 1.19 or allowed to be free. Note that the value of $a_2$ is only affected a little by the change from free to fixed at 1.19. All the fits in table 9.17 that use the $1/\sqrt{n}$ type corrections are less stable than those in table 9.16 that use the $1/n$ corrections. In particular the coefficient $a_2$ is far less stable. This implies that the best fit concurs with the differential approximant analysis with a value of $2\nu_1$ around 1.19, and uses the relatively benign $1/n$ corrections. This second point concurs with the observation that the differential approximants of the smallest eigenvalue, $\lambda_1(n)$, were relatively well behaved.

In tables 9.18 and 9.19 are the fits to the largest eigenvalue, $\lambda_3(n)$, of the moment of inertia matrix for the ($P$-$P$-$P$) model using the forms (9.37) and (9.38) respectively. Once again the top of the tables gives coefficients for the fits where the value of $2\nu_3$ is free while the middle and bottom sections have $2\nu_3$ fixed, as for $\lambda_1(n)$. The fit with $1/n$ corrections and $2\nu_3$ free (table 9.18) is clearly unstable with $2\nu_3$ decreasing and the coefficient $a_2$ changing rapidly. Furthermore by fixing the value of $2\nu_3$ to 1.29, which is the same as the free estimate of around 1.29, a big change is produced in the coefficient $a_2$. Better behaved are some of the fits in the table 9.19, with the fit using $2\nu_3$ free and that fixing it at 1.19 giving similar results, although not as good as in the $\lambda_1(n)$ analysis. So we can tentatively conclude that, *opposed* to the differential approximant analysis for $\lambda_3(n)$, an estimate of $2\nu_3$ around the SAW value of 1.19, with strong $1/\sqrt{n}$ corrections, is best supported by the data fits. While somewhat debatable, it is certainly *consistent* with the data fits.

| $n$ | $a_1$ | $2\nu_1$ | $a_2$ | $a_3$ |
|---|---|---|---|---|
| 20 | −2.724815224 | 1.163388477 | 5.666275943 | −12.96033492 |
| 21 | −2.749265119 | 1.169014693 | 5.858321147 | −13.84016515 |
| 22 | −2.774577794 | 1.174725610 | 6.073152086 | −14.77803577 |
| 23 | −2.793298510 | 1.178924259 | 6.233843907 | −15.61234080 |
| 24 | −2.814179170 | 1.183535931 | 6.424678029 | −16.52535092 |
| 25 | −2.826827330 | 1.186303333 | 6.543063127 | −17.22713264 |
| 26 | −2.836927800 | 1.188491651 | 6.642284471 | −17.71643162 |
| 27 | −2.846262248 | 1.190496482 | 6.735594254 | −18.32935342 |
| 28 | −2.852661690 | 1.191855454 | 6.803456388 | −18.67947825 |
| 20 | −2.841622425 | 1.19 | 6.621334854 | −17.22703337 |
| 21 | −2.842513608 | 1.19 | 6.653550396 | −17.59298974 |
| 22 | −2.843234486 | 1.19 | 6.682593176 | −17.80737537 |
| 23 | −2.843624178 | 1.19 | 6.697973178 | −18.03610699 |
| 24 | −2.843851650 | 1.19 | 6.708514087 | −18.07892776 |
| 25 | −2.843960818 | 1.19 | 6.712785354 | −18.19870428 |
| 26 | −2.843982848 | 1.19 | 6.714549189 | −18.14820023 |
| 27 | −2.843920033 | 1.19 | 6.710827665 | −18.17525276 |
| 28 | −2.843834255 | 1.19 | 6.707119559 | −18.05547848 |
| 20 | −3.280556421 | 1.29 | 10.21020381 | −33.26010493 |
| 21 | −3.286864991 | 1.29 | 10.44300945 | −35.47611016 |
| 22 | −3.292724237 | 1.29 | 10.67258083 | −37.64051879 |
| 23 | −3.298000706 | 1.29 | 10.88843806 | −39.91926059 |
| 24 | −3.302886931 | 1.29 | 11.09941400 | −42.11210440 |
| 25 | −3.307447054 | 1.29 | 11.30407740 | −44.48184698 |
| 26 | −3.311730277 | 1.29 | 11.50620791 | −46.78139096 |
| 27 | −3.315754269 | 1.29 | 11.70282370 | −49.25849539 |
| 28 | −3.319594507 | 1.29 | 11.89941941 | −51.68867047 |

Table 9.16: The coefficients of 3 fits to the smallest eigenvalue, $\lambda_1(n)$, of the moment of inertia for the $(P\text{-}P\text{-}P)$ model. The top fit allows the value of $\nu$ to be free while the bottom two fits fix this exponent. These fits use $1/n$ corrections: see equation (9.37).

| $n$ | $a_1$ | $2\nu_1$ | $a_2$ | $a_3$ |
|---|---|---|---|---|
| 20 | $-4.463970915$ | 1.438199488 | $-3.606651092$ | 6.024597655 |
| 21 | $-4.417089672$ | 1.430174649 | $-3.480093584$ | 5.893284132 |
| 22 | $-4.383389048$ | 1.424513214 | $-3.361443569$ | 5.792753946 |
| 23 | $-4.336970494$ | 1.416691312 | $-3.224987032$ | 5.658579334 |
| 24 | $-4.305279471$ | 1.411481234 | $-3.101773019$ | 5.559966820 |
| 25 | $-4.251393955$ | 1.402529166 | $-2.928596939$ | 5.399327527 |
| 26 | $-4.185092547$ | 1.391743755 | $-2.668915949$ | 5.190114130 |
| 27 | $-4.134393805$ | 1.383454870 | $-2.491134144$ | 5.033823177 |
| 28 | $-4.068857947$ | 1.372925337 | $-2.218436807$ | 4.821717645 |
| 20 | $-3.002625739$ | 1.19 | .8401583251 | 1.820134994 |
| 21 | $-2.989943090$ | 1.19 | 1.064189417 | 1.712490790 |
| 22 | $-2.977800616$ | 1.19 | 1.311181915 | 1.603789591 |
| 23 | $-2.967160395$ | 1.19 | 1.519273404 | 1.508779897 |
| 24 | $-2.956610992$ | 1.19 | 1.755659212 | 1.409638571 |
| 25 | $-2.947756459$ | 1.19 | 1.945718751 | 1.326801891 |
| 26 | $-2.939000159$ | 1.19 | 2.160325380 | 1.240761320 |
| 27 | $-2.931582457$ | 1.19 | 2.333557705 | 1.168364089 |
| 28 | $-2.924319651$ | 1.19 | 2.527044334 | 1.093985623 |
| 20 | $-3.591404322$ | 1.29 | $-.9514706910$ | 3.514121096 |
| 21 | $-3.584155248$ | 1.29 | $-.8278880799$ | 3.453222689 |
| 22 | $-3.577164741$ | 1.29 | $-.6812936357$ | 3.390026328 |
| 23 | $-3.571423036$ | 1.29 | $-.5735623368$ | 3.339377936 |
| 24 | $-3.565542637$ | 1.29 | $-.4375063145$ | 3.283538195 |
| 25 | $-3.561147627$ | 1.29 | $-.3477369847$ | 3.243010963 |
| 26 | $-3.556660462$ | 1.29 | $-.2334094556$ | 3.198363502 |
| 27 | $-3.553336603$ | 1.29 | $-.1604352409$ | 3.166495500 |
| 28 | $-3.550005621$ | 1.29 | $-.0671715826$ | 3.131828410 |

Table 9.17: The coefficients of 3 fits to the smallest eigenvalue, $\lambda_1(n)$, of the moment of inertia for the ($P$-$P$-$P$) model. The top fit allows the value of $\nu$ to be free while the bottom two fits fix this exponent. These fits use $1/\sqrt{n}$ corrections: see equation (9.38).

| $n$ | $a_1$ | $2\nu_3$ | $a_2$ | $a_3$ |
|---|---|---|---|---|
| 20 | −2.474887010 | 1.330133182 | 1.169158567 | 1.380214814 |
| 21 | −2.451727355 | 1.324819993 | .9845802007 | 2.056262136 |
| 22 | −2.424625717 | 1.318682413 | .7582064092 | 3.216116413 |
| 23 | −2.402331687 | 1.313703614 | .5633338950 | 4.044177082 |
| 24 | −2.384502562 | 1.309756026 | .4020464689 | 4.986481513 |
| 25 | −2.363446512 | 1.305145685 | .204718825 | 5.916901080 |
| 26 | −2.345549804 | 1.301270237 | .029444054 | 7.025885609 |
| 27 | −2.327992403 | 1.297496245 | −.146489591 | 7.927417445 |
| 28 | −2.310098163 | 1.293691011 | −.333726665 | 9.196060404 |
| 20 | −1.859794757 | 1.19 | −3.860040364 | 23.84789160 |
| 21 | −1.852652904 | 1.19 | −4.124366246 | 26.16626450 |
| 22 | −1.846211559 | 1.19 | −4.376201889 | 28.73784289 |
| 23 | −1.840251622 | 1.19 | −4.620417516 | 31.11437593 |
| 24 | −1.834780204 | 1.19 | −4.856318771 | 33.76763879 |
| 25 | −1.829762026 | 1.19 | −5.081959793 | 36.18085137 |
| 26 | −1.825086009 | 1.19 | −5.302251764 | 38.88617400 |
| 27 | −1.820788301 | 1.19 | −5.512698552 | 41.34074272 |
| 28 | −1.816777611 | 1.19 | −5.717671690 | 44.07061471 |
| 20 | −2.298728754 | 1.29 | −.2711713545 | 7.814819509 |
| 21 | −2.297004286 | 1.29 | −.3349072110 | 8.283144238 |
| 22 | −2.295701309 | 1.29 | −.3862142636 | 8.904699700 |
| 23 | −2.294628149 | 1.29 | −.4299526474 | 9.231222466 |
| 24 | −2.293815488 | 1.29 | −.4654187863 | 9.734461374 |
| 25 | −2.293248258 | 1.29 | −.4906678487 | 9.897709787 |
| 26 | −2.292833438 | 1.29 | −.5105930268 | 10.25298309 |
| 27 | −2.292622533 | 1.29 | −.5207026654 | 10.25750186 |
| 28 | −2.292537860 | 1.29 | −.5253719254 | 10.43742375 |

Table 9.18: The coefficients of 3 fits to the larger eigenvalue, $\lambda_3(n) = \lambda_2(n)$, of the moment of inertia for the $(P\text{-}P\text{-}P)$ model. The top fit allows the value of $\nu$ to be free while the bottom two fits fix this exponent. These fits use $1/n$ corrections: see equation (9.37).

| $n$ | $a_1$ | $2\nu_3$ | $a_2$ | $a_3$ |
|---|---|---|---|---|
| 20 | −2.289709609 | 1.300873021 | 2.156581910 | −.641494007 |
| 21 | −2.203913210 | 1.286015179 | 2.372074485 | −.875640625 |
| 22 | −2.074442935 | 1.264311462 | 2.811630202 | −1.260842112 |
| 23 | −2.002427148 | 1.252107081 | 3.013642266 | −1.465889862 |
| 24 | −1.934589619 | 1.240977908 | 3.276552782 | −1.677637868 |
| 25 | −1.874144773 | 1.230877504 | 3.457941812 | −1.854517812 |
| 26 | −1.810893738 | 1.220664223 | 3.722052864 | −2.058290520 |
| 27 | −1.770887298 | 1.214043641 | 3.844019958 | −2.177090231 |
| 28 | −1.711412712 | 1.204558139 | 4.107592804 | −2.373584861 |
| 20 | −1.636913110 | 1.19 | 4.143013210 | −2.519666678 |
| 21 | −1.633379274 | 1.19 | 4.188757262 | −2.547008047 |
| 22 | −1.629046469 | 1.19 | 4.292272220 | −2.588222694 |
| 23 | −1.627137319 | 1.19 | 4.313440217 | −2.602820328 |
| 24 | −1.624169435 | 1.19 | 4.394573742 | −2.632908767 |
| 25 | −1.623405873 | 1.19 | 4.395447180 | −2.637815348 |
| 26 | −1.621492277 | 1.19 | 4.456090152 | −2.658581094 |
| 27 | −1.621394660 | 1.19 | 4.443674362 | −2.657516847 |
| 28 | −1.620325282 | 1.19 | 4.485241679 | −2.670248710 |
| 20 | −2.225691696 | 1.29 | 2.351384151 | −.8256805566 |
| 21 | −2.227591431 | 1.29 | 2.296679775 | −.8062761517 |
| 22 | −2.228410599 | 1.29 | 2.299796592 | −.8019859233 |
| 23 | −2.231399964 | 1.29 | 2.220604371 | −.7722222431 |
| 24 | −2.233101073 | 1.29 | 2.201408356 | −.7590092032 |
| 25 | −2.236797036 | 1.29 | 2.101991533 | −.7216063134 |
| 26 | −2.239152583 | 1.29 | 2.062355258 | −.7009788882 |
| 27 | −2.243148807 | 1.29 | 1.949681393 | −.6593854256 |
| 28 | −2.246011252 | 1.29 | 1.891025753 | −.6324059189 |

Table 9.19: The coefficients of 3 fits to the largest eigenvalue, $\lambda_3(n) = \lambda_2(n)$, of the moment of inertia for the ($P$-$P$-$P$) model. The top fit allows the value of $\nu$ to be free while the bottom two fits fix this exponent. These fits use $1/\sqrt{n}$ corrections: see equation (9.38).

# Bibliography


[1] A. Aharony and M.E. Fisher. Universality in analytic corrections to scaling for planar Ising models. *Phys. Rev. Lett.*, 45:679–682, 1980.

[2] J.-P. Allouche. Sur la transcendence de la série formelle $\pi$. *Sém. Théor. Nombres Bordeaux, Série* 2, 1:103–117, 1990.

[3] J.-P. Allouche. Personal communication, 2000.

[4] G. E. Andrews. In G.-C. Rota, editor, *The Theory of Partitions*, volume 2 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley, Reading, Massachusetts, 1976.

[5] F. C. Auluck. On some new types of partitions associated with generalized Ferrers graphs. *Proc. Cam. Phil. Soc.*, 47:679–686, 1951.

[6] M. T. Batchelor, D. Bennett-Wood, and A. L. Owczarek. Two-dimensional polymer networks at a mixed boundary: Surface and wedge exponents. *Eur. Phys. J. B Condens. Matter Phys.*, 5:139–142, 1998.

[7] R. J. Baxter. Exactly solved models. In G.D. Cohen, editor, *Fundamental Problems in Statistical Mechanics V; Proceedings of the 1980 Enschede Summer School*, pages 109–141, North Holland, Amsterdam, 1980.

[8] R. J. Baxter. Hard hexagons: exact solution. *J. Phys. A: Math. Gen.*, 13:L61–L70, 1980.

[9] R. J. Baxter. *Exactly Solved Models in Statistical Mechanics.* Academic Press, London, 1982.

[10] R.J. Baxter. The inversion relation method for some two-dimensional exactly solved models in lattice statistics. *J. Statist. Phys.*, 28:1–41, 1982.

[11] R.J. Baxter. Two-dimensional models in statistical mechanics. In V.V. Bazhanov and C.J. Burden, editors, *Statistical Mechanics and Field Theory; Proceedings of the Seventh Physics Summer School*, pages 129–167, ANU, 1995. World Scientific.

[12] J. Bétréma and J.-G. Penaud. Animaux et arbres guingois. *Theoret. Comput. Sci.*, 117:67–89, 1993.

[13] J. Bétréma and J.-G. Penaud. Modèles avec particules dures, animaux dirigés et séries en variables partiellement commutatives. Technical Report 93-18, LaBRI, Université Bordeaux I, 1993.

[14] D. P. Bhatia, M. A. Prasad, and D. Arora. Asymptotic results for the number of multidimensional partitions of an integer and directed compact lattice animals. *J. Phys. A: Math. Gen.*, 30:2281–2285, 1997.

[15] H. W. J. Blöte and H. J. Hilhorst. Spiralling self-avoiding walks: an exact solution. *J. Phys. A: Math. Gen.*, 17:L111–L115, 1984.

[16] M. Bousquet-Mélou. Denominators of anisotropic 3-choice polygons. Personal communication with author.

[17] M. Bousquet-Mélou. The partition function is (under a simple transformation) a modular form and consequently differentiably algebraic. Personal communication with author.

[18] M. Bousquet-Mélou. *$q$-Énumération de polyominos convexes.* PhD thesis, University of Bordeaux I, 1991.

[19] M. Bousquet-Mélou. Convex polyominoes and algebraic languages. *J. Phys. A: Math. Gen.*, 25:1935–1944, 1992.

[20] M. Bousquet-Mélou. Convex polyominoes and heaps of segments. *J. Phys. A: Math. Gen.*, 25:1925–1934, 1992.

[21] M. Bousquet-Mélou. $q$-Énumération de polyominos convexes. *J. Combin. Theory Ser. A*, 64:265–288, 1993.

[22] M. Bousquet-Mélou. Codage des polyominos convexes et équations pour l'énumération suivant l'aire. *Discrete Appl. Math.*, 48:21–43, 1994.

[23] M. Bousquet-Mélou. A method for the enumeration of various classes of column-convex polygons. *Discrete Math.*, 154:1–25, 1996.

[24] M. Bousquet-Mélou. Rapport d'habilitation, 1996. LaBRI, Université Bordeaux I.

[25] M. Bousquet-Mélou. New enumerative results on two-dimensional directed animals. *Discrete Math.*, 180:73–106, 1998.

[26] M. Bousquet-Mélou and J.-M. Fédou. The generating function of convex polyominoes: the resolution of a $q$-differential system. *Discrete Math.*, 137:53–75, 1995.

[27] M. Bousquet-Mélou and A. J. Guttmann. Enumeration of three-dimensional convex polygons. *Annal Comb.*, 1:27–53, 1997.

[28] M. Bousquet-Mélou, A. J. Guttmann, W. P. Orrick, and A. Rechnitzer. Inversion relations, reciprocity and polyominoes. *Annal Comb.*, 3:223–249, 1999.

[29] M. Bousquet-Mélou and A. Rechnitzer. Lattice animals and heaps of dimers. In preparation.

[30] M. Bousquet-Mélou and A. Rechnitzer. The site-perimeter of bargraphs. In preparation.

[31] M. Bousquet-Mélou and A. Rechnitzer. Connected heaps are nice animals. In C. Martínez, M. Noy, and O. Serra, editors, *Proceedings of the 11th Conference on Formal Power Series and Algebraic Combinatorics*, pages 84–95, Barcelona, 1999.

[32] M. Bousquet-Mélou and X. G. Viennot. Empilements de segments et $q$-énumération de polyominos convexes dirigés. *J. Combin. Theory Ser. A*, 60:196–224, 1992.

[33] R. Brak, J.W. Essam, and A.L. Owczarek. From the Bethe Ansatz to the Gessel-Viennot theorem. *Annal Comb.*, 3:251–265, 2000.

[34] R. Brak and A. J. Guttmann. Exact solution of the staircase and row-convex polygon perimeter and area generating function. *J. Phys. A: Math. Gen.*, 23:4581–4588, 1990.

[35] R. Brak, A. J. Guttmann, and I. G. Enting. Exact solution of the row-convex polygon perimeter generating function. *J. Phys. A: Math. Gen.*, 23:2319–2326, 1990.

[36] R. Brak, A. J. Guttmann, and S. G. Whittington. A collapse transition in a directed walk model. *J. Phys. A: Math. Gen.*, 25:2437–2446, 1992.

[37] R. Brak, A. L. Owczarek, and C. Soteros. On anisotropic spiral self-avoiding walks. *J. Phys. A: Math. Gen.*, 31:4851–4869, 1998.

[38] S. R. Broadbent and J. M. Hammersley. Percolation processes I. Crystals and mazes. *Proc. Cam. Phil. Soc.*, 53:629–641, 1957.

[39] L. Brown, editor. *The New Shorter Oxford English Dictionary*, volume 1. Clarendon Press, Oxford, 1993.

[40] S. Caracciolo, M. S. Causo, and A. Pelissetto. High-precision determination of the critical exponent gamma for self avoiding walks. *Phys. Rev. E*, 57:R1215–R1218, 1998.

[41] F. Carlson. Über Potenzreihen mit ganzzahligen Koeffizienten. *Math. Zeitschrift*, 9:1–13, 1921.

[42] A. R. Conway, M. Delest, and A. J. Guttmann. On the number of three choice polygons. *Mathematical and Computer Modelling*, 26:51–58, 1997.

[43] A. R. Conway, I. G. Enting, and A. J. Guttmann. Algebraic techniques for enumerating self-avoiding walks on the square lattice. *J. Phys. A: Math. Gen.*, 26:1519–1534, 1993.

[44] A. R. Conway and A. J. Guttmann. On two-dimensional percolation. *J. Phys. A: Math. Gen.*, 28:891–904, 1995.

[45] A. R. Conway and A. J. Guttmann. Square lattice self-avoiding walks and corrections to scaling. *Phys. Rev. Lett.*, 77(26):5284–5287, 1996.

[46] A. R. Conway and A. J. Guttmann. Hexagonal lattice directed site animals. In M. T. Batchelor and L. T. Wille, editors, *Statistical Physics on the Eve of the 21st Century*, volume 14 of *Advances in Statistical Mechanics*, pages 491–504. World Scientific, Singapore, 1999.

[47] P.-G. de Gennes. Exponents for the excluded volume problem as derived by the Wilson method. *Phys. Lett.*, 38A:339–340, 1972.

[48] M. Delest. Polyominoes and animals: some recent results. *J. Math. Chem.*, 8:3–18, 1991.

[49] M. Delest and S. Dulucq. Enumeration of directed column-convex animals with given perimeter and area. *Croat. Chem. Acta*, 66:59–80, 1993.

[50] M. P. Delest. Generating functions for column-convex polyominoes. *J. Combin. Theory Ser. A*, 48:12–31, 1988.

[51] M. P. Delest and J. M. Fédou. Exact formulas for fully diagonal compact animals. Rapport LaBRI 89-06, Université de Bordeaux I, 1989.

[52] M. P. Delest, D. Gouyou-Beauchamps, and B. Vauquelin. Enumeration of parallelogram polyominoes with given bond and site perimeter. *Graphs Combin.*, 3:325–339, 1987.

[53] M. P. Delest and G. Viennot. Algebraic languages and polyomino enumeration. *Theoret. Comput. Sci.*, 34:169–2068, 1984.

[54] D. Dhar. Equivalence of the two-dimensional directed-site animal problem to Baxter's hard-square lattice-gas model. *Phys. Rev. Lett*, 49:959–962, 1982.

[55] D. Dhar. Exact solution of a directed-site animals-enumeration problem in three dimensions. *Phys. Rev. Lett.*, 51:853–856, 1983.

[56] J. P. Dubernard and I. Dutour. Enumération de polyominos convexes dirigés. *Discrete Math.*, 157:79–90, 1996.

[57] I. G. Enting. Generating functions for enumerating self-avoiding rings on the square lattice. *J. Phys. A: Math. Gen.*, 13:3713–3722, 1980.

[58] I. G. Enting and A. J. Guttmann. Polygons on the honeycomb lattice. *J. Phys. A: Math. Gen.*, 22:1371–1384, 1989.

[59] L. Euler. *Introductio in analysin infinitorum.* Marcum-Michaelem Bousquet, Lausannae, 1748. French translation: *Introduction à l'analyse infinitésimale.* ACL éditions, 1987.

[60] J. M. Fédou. Fonctions de Bessel, empilements et tresses. In P. Leroux and C. Reutenauer, editors, *Proceedings of the 4th Conference on Formal Power Series and Algebraic Combinatorics*, pages 189–202, Université du Québec à Montréal, 1992. Publications du LaCIM.

[61] J. M. Fédou and N. Rouillon. Polyominos et $q$-analogues des fonctions de Bessel, une preuve combinatoire. In B. Leclerc and J.-Y. Thibon, editors, *Proceedings of the* $7^{th}$ *Conference on Formal Power Series and Algebraic Combinatorics*, pages 203–220, Marne-la-Vallée, June 1995.

[62] S. Feretić. A new coding for column-convex directed animals. *Croat. Chem. Acta*, 66:81–90, 1993.

[63] S. Feretić. The area generating function for the column-convex polyominoes on the checkerboard lattice. *Croat. Chem. Acta*, 68:75–90, 1995.

[64] S. Feretić. The column-convex polyominoes perimeter generating function for everybody. *Croat. Chem. Acta*, 69:741–756, 1996.

[65] S. Feretić. A new way of counting the column-convex polyominoes by perimeter. *Discrete Math.*, 180:173–184, 1998.

[66] S. Feretić. A $q$-enumeration of directed diagonally convex polyominoes. In C. Martínez, M. Noy, and O. Serra, editors, *Proceedings of the 11th Conference on Formal Power Series and Algebraic Combinatorics*, pages 195–206, Barcelona, 1999.

[67] S. Feretić. An alternative method for $q$-counting directed column-convex polyominoes. *Discrete Mathematics*, 210:55–70, 2000.

[68] S. Feretić and D. Svrtan. On the number of column-convex polyominoes with given perimeter and number of columns. In A. Barlotti, M. Delest, and R. Pinzani, editors, *Proceedings of the* $5^{th}$ *Conference on Formal Power Series and Algebraic Combinatorics*, pages 201–214, Florence, June 1993.

[69] M. E. Fisher and M. F. Sykes. Excluded-volume problem and the Ising model of ferromagnetism. *Phys. Rev.*, 114:45–58, 1959.

[70] P. Flajolet and A. Odlyzko. Singularity analysis of generating functions. *SIAM J. Discrete Math.*, 3(2):216–240, 1990.

[71] G. Forgacs and V. Privman. Directed compact lattice animals: exact results. *J. Statist. Phys.*, 49:1165–1180, 1987.

[72] A. R. Forsyth. *Theory of Differential Equations*, volume 4. Cambridge University Press, Cambridge, 1902. Part 3.

[73] I. M. Gessel. On the number of convex polyominoes. Manuscript, 1990.

[74] S. W. Golomb. Checkerboards and polyominoes. *Amer. Math. Monthly*, 61:675–682, 1954.

[75] S. W. Golomb. *Polyominoes: Puzzles, Patterns, Problems, and Packings*. Princeton University Press, second edition, 1996.

[76] D. Gouyou-Beauchamps and G. Viennot. Equivalence of the two-dimensional directed animal problem to a one-dimensional path problem. *Adv. in Appl. Math.*, 9:334–357, 1988.

[77] P. Grassberger. The critical behaviour of two-dimensional self-avoiding random walks. *Z. Phys. B*, 48:255–260, 1982.

[78] P. Grassberger. Pruned-enriched rosenbluth method: Simulations of $\theta$ polymers of chain length up to 1000000. *Phys. Rev. E*, 56(3):3682–3693, 1997.

[79] G. Grimmett. *Percolation*. Springer-Verlag, New York, 1989.

[80] A. J. Guttmann. Asymptotic analysis of coefficients. In C. Domb and J. Lebowitz, editors, *Phase Transitions and Critical Phenomena*, volume 13, pages 1–234. Academic Press, London, 1989. Programs available from `http://www.ms.unimelb.edu.au/~tonyg`.

[81] A. J. Guttmann. On the critical behaviour of self-avoiding walks: II. *J. Phys. A: Math. Gen.*, 22:2807–2813, 1989.

[82] A. J. Guttmann. Indicators of solvability for lattice models. *Discrete Math.*, 217:167–189, 2000.

[83] A. J. Guttmann and I. G. Enting. Solvability of some statistical mechanical systems. *Phys. Rev. Lett.*, 76:344–347, 1996.

[84] A. J. Guttmann and D. S. Gaunt. On the asymptotic number of lattice animals in bond and site percolation. *J. Phys. A: Math. Gen.*, 11:949–953, 1978.

[85] A. J. Guttmann and T. Prellberg. Staircase polygons, elliptic integrals, Heun functions and lattice Green functions. *Phys. Rev. E.*, 47:R2233–2236, 1993.

[86] A. J. Guttmann, T. Prellberg, and A. L. Owczarek. On the symmetry classes of planar self-avoiding walks. *J. Phys. A: Math. Gen.*, 26:6615–6623, 1993.

[87] A. J. Guttmann and K. J. Wallace. On three-dimensional spiral anisotropic self-avoiding walks. *J. Phys. A: Math. Gen.*, 18:L1049–L1054, 1985.

[88] A. J. Guttmann and K. J. Wallace. Two-dimensional spiral self-avoiding walks. *J. Phys. A: Math. Gen.*, 19:1645–1653, 1986.

[89] A. J. Guttmann and J. Wang. The extension of self-avoiding random walk series in two dimensions. *J. Phys. A: Math. Gen.*, 24:3107–3109, 1991.

[90] A. J. Guttmann and N. Wormald. On the number of spiral self-avoiding walks. *J. Phys. A: Math. Gen.*, 17:L271–L274, 1984.

[91] A.J. Guttmann, I. Jensen, L. H. Wong, and I.G. Enting. Punctured polygons and polyominoes on the square lattice. *J. Phys. A: Math. Gen.*, 33:1735–1764, 2000.

[92] Programs developed by B. Salvy, P. Zimmermann, F. Chyzak and colleagues at INRIA, France. Available from http://pauillac.inria.fr/algo.

[93] E. Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeit. Phys.*, 31:253–258, 1925.

[94] M.T. Jaekel and J.-M. Maillard. Inverse functional relation on the Potts model. *J. Phys. A: Math. Gen.*, 15:2241–2257, 1982.

[95] M.T. Jaekel and J.-M. Maillard. Symmetry relations in exactly soluble models. *J. Phys. A: Math. Gen.*, 15:1309–1325, 1982.

[96] E. J. Janse van Rensburg and N. Madras. Metropolis Monte Carlo simulation of lattice animals. *J. Phys. A: Math. Gen.*, 30:8035–8066, 1997.

[97] I. Jensen. Anisotropic series for bond animals, directed bond animals and lattice trees. Personal communication with author.

[98] I. Jensen and A. J. Guttmann. Anisotropic series for self-avoiding polygons. Personal communication with author.

[99] I. Jensen and A. J. Guttmann. Statistics of lattice animals (polyominoes) and polygons. *J. Phys. A: Math. Gen.*, 33:L257–L263, 2000.

[100] G. S. Joyce. An exact formula for the number of spiral self-avoiding walks. *J. Phys. A: Math. Gen.*, 17:L463–L467, 1984.

[101] G. S. Joyce and R. Brak. An exact solution for a spiral self-avoiding walk model on the triangular lattice. *J. Phys. A: Math. Gen.*, 18:L293–L298, 1985.

[102] G. S. Joyce and A. J. Guttmann. Exact results for the generating function of directed column-convex animals on the square lattice. *J. Phys. A: Math. Gen.*, 27:4359–4367, 1994.

[103] D. Kim. The number of convex polyominos with given perimeter. *Discrete Math.*, 70:47–51, 1988.

[104] D. A. Klarner. Some results concerning polyominoes. *Fibonacci Quart.*, 3:9–20, 1965.

[105] D. A. Klarner. Cell growth problems. *Canad. J. Math.*, 19:851–863, 1967.

[106] D. A. Klarner. A combinatorial formula involving the Fredholm integral equation. *J. Combin. Theory*, 5:59–74, 1968.

[107] D. A. Klarner and R. L. Rivest. A procedure for improving the upper bound for the number of $n$-ominoes. *Canad. J. Math.*, 25:585–602, 1973.

[108] D. A. Klarner and R. L. Rivest. Asymptotic bounds for the number of convex $n$-ominoes. *Discrete Math.*, 8:31–40, 1974.

[109] G. Kreweras. Sur les éventails de segments. *Cahiers du BURO*, 15:3–41, 1970.

[110] J. Levine. Note on the number of pairs of non-intersecting routes. *Scripta Math.*, 24:335–338, 1959.

[111] B. Li, N. Madras, and A. D. Sokal. Critical exponents, hyperscaling and universal amplitude ratios for two- and three-dimensional self-avoiding walks. *J. Statist. Phys.*, 80:661–754, 1995.

[112] K. Y. Lin. Perimeter generating function for row-convex polygons on the rectangular lattice. *J. Phys. A: Math. Gen.*, 23:4703–4705, 1990.

[113] K. Y. Lin. Exact solution of the convex polygon perimeter and area generating function. *J. Phys. A: Math. Gen.*, 24:2411–2417, 1991.

[114] K. Y. Lin and S. J. Chang. Rigorous results for the number of convex polygons on the square and honeycomb lattices. *J. Phys. A: Math. Gen.*, 21:2635–2642, 1988.

[115] K. Y. Lin and W. J. Tzeng. Perimeter and area generating functions of the staircase and row-convex polygons on the rectangular lattice. *Internat. J. Modern Phys. B*, 5:1913–1925, 1991.

[116] K. Y. Lin and F. Y. Wu. Unidirectional-convex polygons on the honeycomb lattice. *J. Phys. A: Math. Gen.*, 23:5003–5010, 1990.

[117] L. Lipshitz. D-finite power series. *J. Algebra*, 122:353–373, 1989.

[118] K. C. Liu and K. Y. Lin. Spiral self-avoiding walks on a triangular lattice: end-to-end distance. *J. Phys. A: Math. Gen.*, 18:L647–L650, 1985.

[119] D. MacDonald, S. Joseph, D. L. Hunter, L. L. Moseley, N. Jan, and A. J. Guttmann. Self-avoiding walks on the simple cubic lattice. *J. Phys. A: Math. Gen.*, 33:1–11, 2000.

[120] P. A. MacMahon. *Combinatory Analysis*, volume 2. Chelsea, New York, 1960. Section VIII.

[121] N. Madras and G. Slade. *The Self-Avoiding Walk.* Birkhäuser, Boston, 1993.

[122] A. Magid. *Enumeration of convex polyominoes. A generalisation of the Robinson-Schensted correspondence and the dimer problem.* PhD thesis, Brandeis University, 1992.

[123] J.-M. Maillard. The inversion relation. *J. Physique*, 46:329–341, 1985.

[124] S. S. Manna. Critical behaviour of anisotropic spiral self avoiding walks. *J. Phys. A: Math. Gen.*, 17:L899–L903, 1984.

[125] B. Nienhuis. Exact critical exponents of the $o(n)$ models in two dimensions. *Phys. Rev. Lett.*, 49:1062–1065, 1982.

[126] L. Onsager. Crystal statistics I. A two-dimensional model with an order-disorder transition. *Phys. Rev.*, 65:117–149, 1944.

[127] W. J. C. Orr. Statistical treatment of polymer solutions at infinite dilution. *Trans. Faraday Soc.*, 43:12–27, 1947.

[128] A. L. Owczarek and T. Prellberg. Exact solution of the discrete $(1+1)$-dimensional SOS model with field and surface interactions. *J. Statist. Phys.*, 70:1175–1194, 1993.

[129] J.-G. Penaud. Une nouvelle bijection pour les animaux dirigés. In *Actes du 22ème Séminaire Lotharingien de Combinatoire*, Université de Strasbourg, France, 1989.

[130] J.-G. Penaud. Animaux dirigés diagonalement convexes et arbres ternaires. Technical Report 90-62, LaBRI, Université Bordeaux I, 1990.

[131] S.V. Pokrovsky and Yu.A. Bashilov. Star-triangle relations in the exactly solvable statistical models. *Comm. Math. Phys.*, 84:103–132, 1982.

[132] G. Pólya. On the number of certain lattice polygons. *J. Combin. Theory*, 6:102–105, 1969.

[133] T. Prellberg and R. Brak. Critical exponents from nonlinear functional equations for partially directed cluster models. *J. Statist. Phys.*, 78:701–730, 1995.

[134] T. Prellberg and A. L. Owczarek. Stacking models of vesicles and compact clusters. *J. Statist. Phys.*, 80:755–779, 1995.

[135] V. Privman and N. M. Švrakić. Exact generating function for fully directed compact lattice animals. *Phys. Rev. Lett.*, 60:1107–1109, 1988.

[136] V. Privman and N. M. Švrakić. *Directed Models of Polymers, Interfaces, and Clusters: Scaling and Finite–Size Properties*, volume 338 of *Lecture Notes in Physics*. Springer–Verlag, Berlin, 1989.

[137] A. Rechnitzer. Haruspicy and anisotropic generating functions. In preparation.

[138] A. Rechnitzer and A. L. Owczarek. On three-dimensional self-avoiding walk symmetry classes. *J. Phys. A: Math. Gen.*, 33:2685–2723, 2000.

[139] A. Rechnitzer, A. L. Owczarek, and L. H. Wong. The anti-spiral self-avoiding walk. In preparation.

[140] S. Redner and I. Majid. Critical properties of directed self-avoiding walks. *J. Phys. A: Math. Gen.*, 16:L307–L310, 1983.

[141] C.L. Schultz. Solvable $q$-state models in lattice statistics and quantum field theory. *Phys. Rev. Lett.*, 46:629–632, 1981.

[142] R. Shankar. Simple derivation of the Baxter-model free energy. *Phys. Rev. Lett.*, 47:1177–1180, 1981.

[143] L. W. Shapiro. Directed animals and Motzkin paths. Preprint, 1999.

[144] R. P. Stanley. Differentiably finite power series. *European J. Combin.*, 1:175–188, 1980.

[145] R.P. Stanley. Combinatorial reciprocity theorems. *Adv. Math.*, 14:194–253, 1974.

[146] R.P. Stanley. Linear Diophantine equations and local cohomology. *Invent. Math.*, 68:175–193, 1982.

[147] R.P. Stanley. *Enumerative Combinatorics*, volume I. Wadsworth and Brooks/Cole, Monterey, California, 1986.

[148] D. Stauffer and A. Aharony. *An Introduction to Percolation Theory.* Taylor and Francis, London, 2nd edition, 1992.

[149] Yu.G. Stroganov. A new calculation method for partition functions in some lattice models. *Phys. Lett.*, 74A:116–118, 1979.

[150] G. Szekeres and A. J. Guttmann. Spiral self-avoiding walks on the triangular lattice. *J. Phys. A: Math. Gen.*, 20:481–493, 1987.

[151] H. N. V. Temperley. Combinatorial problems suggested by the statistical mechanics of domains and of rubber-like molecules. *Phys. Rev.*, 103:1–16, 1956.

[152] W. J. Tzeng and K. Y. Lin. Exact solution of the row-convex polygon generating function for the checkerboard lattice. *Internat. J. Modern Phys. B*, 5:2551–2562, 1991.

[153] B. L. van der Waerden. Die lange Reichweite der regelmässigen Atomanordnung in Mischkristallen. *Zeit. Phys.*, 118:473–479, 1941.

[154] A. R. Veal, J. M. Yeomans, and G. Jug. The effect of attractive monomer-monomer interactions on the adsorption of a polymer chain. *J. Phys. A: Math. Gen.*, 24:827–849, 1991.

[155] G. Viennot. Problèmes combinatoires posés par la physique statistique. *Astérisque*, 121–122:225–246, 1985. Soc. Math. France.

[156] G. X. Viennot. Heaps of pieces 1: basic definitions and combinatorial lemmas. In G. Labelle and P. Leroux, editors, *Combinatoire énumérative*, volume 1234 of *Lecture Notes in Mathematics*. Springer-Verlag, New York, 1985.

[157] S. G. Whittington. The asymptotic form for the number of spiral self-avoiding walks. *J. Phys. A: Math. Gen.*, 17:L117–L119, 1984.

[158] S. G. Whittington. Anisotropic spiral self-avoiding walks. *J. Phys. A: Math. Gen.*, 18:L67–L69, 1985.

[159] S. G. Whittington and C. E. Soteros. Lattice animals: rigorous results and wild guesses. In G. R. Grimmett and D. J. A. Welsh, editors, *Disorder in physical systems: a volume in honour of J. M. Hammersley*. Clarendon Press, Oxford, 1990.

[160] S. G. Whittington, G. M. Torrie, and D. S. Gaunt. Restricted valence site animals on the triangular lattice. *J. Phys. A: Math. Gen.*, 12:L119–L123, 1979.

[161] D. Zeilberger. Symbol-crunching with the transfer-matrix method in order to count skinny physical creatures. Preprint, 2000.