Pattern-avoidance in Binary Fillings of Grid Shapes

by

Alexey Spiridonov

A.B., Princeton University, 2004

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

© Alexey Spiridonov, 2009. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Author . . . . . .
Department of Mathematics
May 8, 2009

Certified by . . . . . .
Alexander Postnikov
Associate Professor of Applied Mathematics
Thesis Supervisor

Accepted by . . . . . .
Michel X. Goemans
Chairman, Applied Mathematics Committee

Accepted by . . . . . .
David S. Jerison
Chairman, Department Committee on Graduate Students

Pattern-avoidance in Binary Fillings of Grid Shapes

by

Alexey Spiridonov

Submitted to the Department of Mathematics
on May 8, 2009 in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

A *grid shape* is a set of boxes chosen from a square grid; any Young diagram is an example. We consider a notion of pattern-avoidance for 0-1 fillings of grid shapes, which generalizes permutation pattern-avoidance. A filling avoids a set of patterns if none of its sub-shapes, obtained by removing some rows and columns, equal any of the patterns. We focus on patterns that are *pairs* of 2×2 fillings.

Totally nonnegative Grassmann cells are in bijection with Young shape fillings that avoid particular 2×2 pair, which are, in turn, equinumerous with fillings avoiding another 2×2 pair. The latter ones correspond to acyclic orientations of the shape's bipartite graph. Motivated by this result, due to Postnikov and Williams, we prove a number of such analogs of Wilf-equivalence for these objects — that is, we show that, *in certain classes of shapes, some pattern-avoiding fillings are equinumerous with others*.

The equivalences in this paper follow from two very different bijections, and from a family of recurrences generalizing results of Postnikov and Williams. We used a computer to test each of the described equivalences on a diverse set of shapes. All our results are *nearly* tight, in the sense that we found no natural families of shapes, in which the equivalences hold, but the results' hypotheses do not.

One of these bijections gives rise to some new combinatorics on tilings of skew Young shapes with rectangles, which we name *Popeye diagrams*. In a special case, they are exactly Hugh Thomas's *snug partitions* for $d = 2$. We show that Popeye diagrams are a lattice, and, moreover, each diagram is a sublattice of the Tamari lattice. We also give a simple enumerative result.

Thesis Supervisor: Alexander Postnikov
Title: Associate Professor of Applied Mathematics

Contents

Acknowledgments 5

Chapter 1. Introduction 7
- 1.1. Pattern-avoidance of fillings 7
- 1.2. Motivation 7
- 1.3. Other notions of pattern-avoidance in fillings 8
- 1.4. Related results 9
- 1.5. Contribution 10

Chapter 2. Basic Definitions 13
- 2.1. Shapes and fillings 13
- 2.2. Patterns 14

Chapter 3. Classifying Pattern Pair Equivalences 17
- 3.1. Fundamentals 17
- 3.2. New patterns and new shapes 17
- 3.3. What equivalences hold in a shape? 18
- 3.4. Results 18
- 3.5. Example applications of the results 19

Chapter 4. $\begin{pmatrix} \circ & \bullet \\ \bullet & \circ \end{pmatrix}$ and $\begin{pmatrix} \circ & \bullet \\ \circ & \circ \end{pmatrix}$: ⅃-diagrams and Acyclic Orientations 21
- 4.1. ⅃-diagrams 21
- 4.2. Acyclic orientations 24
- 4.3. Recurrence for $\begin{pmatrix} \circ & \bullet \\ \bullet & \circ \end{pmatrix}$-avoidance 28
- 4.4. Equivalence of $\begin{pmatrix} \circ & \bullet \\ \bullet & \bullet \end{pmatrix}$ and $\begin{pmatrix} \circ & \bullet \\ \circ & \bullet \end{pmatrix}$, and its symmetries 29
- 4.5. $\begin{pmatrix} \circ & \circ \\ \circ & \bullet \end{pmatrix} \leftrightarrow \begin{pmatrix} \circ & \bullet \\ \circ & \bullet \end{pmatrix}$ is stronger with $\begin{pmatrix} \bullet & \circ \\ \circ & \circ \end{pmatrix} \leftrightarrow \begin{pmatrix} \circ & \circ \\ \circ & \bullet \end{pmatrix}$ 30
- 4.6. Same recurrence: $\begin{pmatrix} \circ & \circ \\ \circ & \circ \end{pmatrix}$ and $\begin{pmatrix} \circ & \bullet \\ \bullet & \circ \end{pmatrix}$ 31
- 4.7. Similar recurrence: $\begin{pmatrix} \bullet & \circ \\ \circ & \circ \end{pmatrix}$ and $\begin{pmatrix} \circ & \bullet \\ \bullet & \circ \end{pmatrix}$ 32

Chapter 5. A Bijection between $\begin{pmatrix} \circ & \bullet \\ \circ & \circ \end{pmatrix}$ and $\begin{pmatrix} \circ & \circ \\ \bullet & \circ \end{pmatrix}$ 33
- 5.1. Preliminaries 33
- 5.2. Simple bijection: one upper-left corner 34
- 5.3. Complete bijection: multiple upper-left corners 36

Chapter 6. A Bijection between $\begin{pmatrix} \bullet & \circ \\ \circ & \circ \end{pmatrix}$ and $\begin{pmatrix} \circ & \circ \\ \circ & \bullet \end{pmatrix}$ 59
- 6.1. Preliminaries 59
- 6.2. Bijection 60
- 6.3. Proofs 61

6.4. The same equivalence in a different shape class? 73

Chapter 7. Empirical Observations and Open Problems 75
7.1. When are patterns inequivalent? 75
7.2. Making indirect equivalences direct 76
7.3. A hierarchy of shape classes 76
7.4. How to count the number of pattern-avoiding fillings? 77

Chapter 8. Popeye Diagrams 79
8.1. Definitions 79
8.2. Exchanges and the Popeye lattice 80
8.3. Enumeration and arbitrary skew shapes 85

Bibliography 87

Acknowledgments

I would like to thank my advisor, Alex Postnikov, for suggesting this problem to me, for asking insightful questions, for suggesting new directions, and for keeping me on task. I also thank him for all the mathematics he had taught me, and for the adventures we had together.

I am grateful to Richard Stanley for many helpful discussions, insights and literature suggestions. I am also indebted to him for greatly expanding my view of combinatorics, and for serving on my thesis committee.

I thank Daniel Kleitman for getting me started on giving mathematical talks in his seminar, and for putting up with quite a few! I am also grateful for our conversations, mathematical and not, and for his service on my thesis committee.

Chapter 8 owes its existence to Craig Desjardins. He inspired me to keep investigating Popeye diagrams, and we proved several of the results in collaboration. I thank him for these very enjoyable discussions.

I would like to thank Matthieu Josuat-Vergès for his stimulating letters; they helped shape the exposition and development of several parts of this work. I would also like to thank the reviewers of [**Sp08**] for suggesting several valuable improvements.

I am grateful to Yury Makarychev and Denis Chebikin for proofreading and suggesting ways to rearticulate Chapter 5 — they will make the next iteration of that text far better. I also appreciate Denis, Joel Lewis, Alejandro Morales, and Steven Sam's patience as I presented and refined the argument in Chapter 6.

Last, but not least, I would like to thank my friends and family — too many to list, but you know who you are ☺ — for helping me stay sane and happy through the years!

CHAPTER 1

Introduction

1.1. Pattern-avoidance of fillings

Perhaps the best-known example of pattern avoidance is defined for permutations. Let S_n be the set of permutations of $[n] = \{1, 2, \ldots, n\}$. A permutation $\sigma \in S_n$ *avoids* $\tau \in S_k$ if there is no set of indices $1 \leqslant i_1 \leqslant \cdots < i_k \leqslant n$ such that

$$\pi(i_{\tau^{-1}(1)}) < \pi(i_{\tau^{-1}(2)}) < \cdots < \pi(i_{\tau^{-1}(n)}).$$

In other words, if we take the jth largest value of $\{\pi(i_l)\}$ and replace it by j in $\pi(i_1), \pi(i_2), \ldots, \pi(i_k)$ for all $1 \leqslant j \leqslant k$, we will not get τ's word $\tau(1), \tau(2), \ldots, \tau(k)$ for any index set $\{i_l\}$.

Let $S_n(\sigma)$ be the set of permutations of $[n]$ that avoid σ. Permutations σ and τ are *Wilf-equivalent* if $|S_n(\sigma)| = |S_n(\tau)|$ for all n. For more details on Wilf equivalence, and further references, see [**St07**].

The permutation matrix of σ is a matrix $P_\sigma = (p_{ij})$ with $p_{ij} = 1$ if $\sigma(i) = j$, and $p_{ij} = 0$ otherwise. In terms of these matrices, the permutation $\sigma \in S_n$ avoids $\tau \in S_k$ if no $k \times k$ minor of P_σ equals P_τ. If we draw the permutation matrices with lines separating rows and columns, then permutations are just special cases of $0-1$ fillings of square shapes, or *rook diagrams*, e.g.

$$S_5 \ni 43152 = \begin{array}{|c|c|c|c|c|}\hline \circ & \circ & \bullet & \circ & \circ \\ \hline \circ & \circ & \circ & \circ & \bullet \\ \hline \circ & \bullet & \circ & \circ & \circ \\ \hline \bullet & \circ & \circ & \circ & \circ \\ \hline \circ & \circ & \circ & \bullet & \circ \\ \hline \end{array}.$$

Thus, pattern-avoidance generalizes naturally to fillings of shapes, as follows: a filling F avoids a filling G if no minor of F equals G (*both* the shapes and fillings must agree). Wilf-equivalence also translates to this context — two patterns p_1 and p_2 are *equivalent* if p_1-avoiding fillings are equinumerous with p_2-avoiding fillings.

1.2. Motivation

Permutations are special fillings of special shapes. In Section 1.3, we cite several other notions of pattern-avoiding fillings. All those papers restrict their attention to special shapes (rectangles and Young diagrams). In some, the fillings, too, have extra restrictions (fixed row/column sums).

In contrast, this paper considers arbitrary fillings of arbitrary shapes. However, our patterns are quite special — we require that a filling avoid a

pair of 2×2-fillings. We made this choice because of some results by Alex Postnikov and Lauren Williams [**Po06, Wi05**]. We will restate them in terms of 2×2 pattern pairs:

(1) Fillings of Young diagrams λ that avoid the pattern pair $\left(\begin{smallmatrix}\circ & \bullet \\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet \\ \bullet & \circ\end{smallmatrix}\right)$ (where $\circ = 0$ and $\bullet = 1$) are exactly the $\lrcorner$-diagrams. The latter are in bijection with totally nonnegative Grassmann cells, which are defined in [**Po06**] as elements of a particular cellular decomposition of $Gr_{k,n}^{tnn}$. This $Gr_{k,n}^{tnn}$ denotes, in turn, elements of the Grassmannian with nonnegative Plücker coordinates.

(2) Fillings of Young diagrams λ that avoid $\left(\begin{smallmatrix}\circ & \bullet \\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ \\ \circ & \bullet\end{smallmatrix}\right)$ are acyclic orientations of the diagram's bipartite graph G_λ (rows and columns are vertices, boxes are edges). According to [**Po06**], the following objects associated with λ are equinumerous: acyclic orientations of G_λ, $\lrcorner$-diagrams on λ, totally nonnegative Grassmann cells inside the Schubert cell Ω_λ, and several others.

Thus, $\left(\begin{smallmatrix}\circ & \bullet \\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet \\ \bullet & \circ\end{smallmatrix}\right)$- and $\left(\begin{smallmatrix}\circ & \bullet \\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ \\ \circ & \bullet\end{smallmatrix}\right)$-avoiding fillings are equinumerous in Young shapes. As short-hand, we will instead say "$\left(\begin{smallmatrix}\circ & \bullet \\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet \\ \bullet & \circ\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}\circ & \bullet \\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ \\ \circ & \bullet\end{smallmatrix}\right)$ are equivalent". We show that this result is a part of a large family. We study several other pattern pairs, and show that the above equivalences hold with much weaker conditions on the shapes.

In addition to the $\lrcorner$-diagram motivation, pairs of 2×2s are perhaps the simplest patterns that have not yet been studied extensively. As discussed in Section 1.4, single 2×2 pattern pairs have been described exhaustively in rectangles by Kitaev et al. We conjecture that beyond rectangles, there is not much in the way of single 2×2-pattern equivalences. More specifically, Young shapes are the most regular class after rectangles, and:

Conjecture 1. *The only non-trivial (see Section 3.1)* 2×2*-pattern equivalence that holds in all Young shapes is* $\left(\begin{smallmatrix}\bullet & \circ \\ \circ & \circ\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\bullet & \circ \\ \circ & \bullet\end{smallmatrix}\right)$.

This claim should not be hard to prove using a bijection similar to that of Chapter 6. There is probably not much hope of finding other nice, non-trivial single-2×2-pattern equivalences.

The other possible simplification from 2×2 pairs is to study 2×1 or 1×2 patterns. Avoiding only one kind allows the columns/rows to be independent, and leads to very simple combinatorics — in all cases, there is a "distinguished position" in the row or column, which can be chosen at will. Combinations of 2×1 and 1×2 patterns are a bit more interesting, and could merit a more detailed look.

1.3. Other notions of pattern-avoidance in fillings

Pattern-avoidance in fillings is a recent subject with a lot of variation in the objects of study and definitions. Here, we give an overview of some of these variations.

Marcus and Tardos [**MT04**] generalize permutation-avoidance in a different way. In that paper, a filling *contains* a pattern not only if the some minor equals the pattern — it is enough for the minor to have a 1 wherever the pattern has a 1. This definition leads to nice extremal results in permutation-pattern avoidance, but does not seem related to ours.

Christian Krattenthaler [**Kr06**] discusses the same object — binary fillings of grid shapes, — but with a rather different definition of "pattern". The main objects in his paper are various chains in fillings. For example, an *NE-chain* is a sequence of 1s in the filling, so that each 1 is above and to the right of the previous one. The chain's length is just the number of 1s in it. Many variations on the notion of chain appear in the paper, and the results describe fillings of Ferrers shapes (Young diagrams, reflected) with restricted chain lengths. These include bijections with other objects, and some statistics. In particular, results on non-crossing set partitions and matchings follow from the results on fillings.

Although the problems in Krattenthaler's paper are quite different from ours, there are some interesting commonalities. Our paper, like his, uses $0-1$ fillings to extend previous results. Another benefit is a more uniform approach to a class of problems: both papers can potentially bring out combinatorial connections that would otherwise be obscured. Finally, Krattenthaler shows interest in more general classes of shapes, something that is at the heart of the present paper.

Anna de Mier [**dM07**] uses a definition of avoidance close to that of Marcus and Tardos [**MT04**], but obtains results on noncrossing and nonnesting graphs, related to those in Krattenthaler's paper.

1.4. Related results

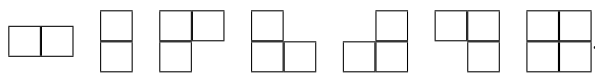
⌟-diagrams have received the most attention of all pattern pairs. Josuat-Vergès [**Jo08**] discovered an intricate bijective proof of the $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \bullet & \circ\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ equivalence. Also, Steingrímsson and Williams [**SW07**] introduced permutation tableaux, a distinguished subset of ⌟-diagrams, and showed that they capture many enumerative properties of permutations.

Stankova [**St07**] does not explicitly mention pattern-avoiding fillings. However, the main relation of that paper is shape-Wilf-ordering $\preceq_s$, which can be rephrased in terms of fillings. Let τ and σ be permutations; then $\sigma \preceq_s \tau$ iff for every Young diagram λ, the number of $0-1$ fillings with exactly one 1 in each row and column, which avoid P_σ, is at most the number of such fillings avoiding P_τ. The notion of ordering patterns by restrictiveness deserves further study in our context. Also, just as in our definition, every cell of the square pattern P_σ must be inside the Young diagram to match.

Jelínek [**Je07**] studies a problem related to Stankova's. He works with $0-1$ fillings of rectangular shapes. However, instead of constraining all row and column sums to be 1 (that would make a permutation, of course), he allows an arbitrary fixed sum for each row and column. In our terms,

his main result is about equivalent patterns (recall — that means "fillings avoiding them are equinumerous"). He shows that the permutations of a fixed order $\leqslant 3$ are all equivalent, when restricted to fillings with a fixed multiset of row and column sums.

The results most closely related to ours are due to Kitaev, Mansour, and Vella [**KMV05**], and Kitaev [**Ki04**]. In both papers, the shapes are rectangles, and the fillings are binary. They consider all nontrivial patterns up to size 2×2 — that is, all $0-1$ fillings of these shapes:

(1.4.1) 

The first paper counts, for each of the 56 described patterns, the number of fillings of an $m \times n$ rectangle, which avoid it. It defines two notions we also use: pattern complement (see Fact 13), and pattern symmetry (Section 3.1).

The second paper finds equivalences between patterns consisting of multiple 3-cell fillings, as in (1.4.1). They forbid 2, 3, and 4 patterns simultaneously, and give equivalence classes of tuples of 3-cell fillings in each case. The main approach, like in [**Ki04**], is to explicitly count the pattern-avoiding fillings of rectangles.

1.5. Contribution

This paper takes a structural approach to the problem of pattern equivalence. The main goal is to classify the equivalences of 2×2 pattern pairs, and the shapes, in which these equivalences hold.

The essential notations and definitions are in Chapter 2. Chapter 3 is a high-level discussion of the problem, our results, and what they mean. In Chapter 4, we extend Postnikov's proof of the $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \bullet & \circ\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ equivalence to far more general shapes, and then apply these ideas to other pattern pairs in Sections 4.6 and 4.7. Chapter 6 briefly describes two nice, and very different bijections, which together give tight descriptions of 6 more (not counting complements) pattern pair equivalences.

This paper does not finish the classification of pattern pairs. However, it makes a substantial step forward. Firstly, it introduces the principal actors – we show that 21 out of 64 (after identifying complements) pattern pairs are involved in interesting equivalences in "nice" shapes. Computational experiments suggest that these patterns are the ones most frequently involved in equivalences. Secondly, we prove several equivalences tying the 21 patterns together. A few of these equivalences seem to be in the broadest terms possible, some have bijective proofs, but many need improvements. However, the broad outlines of the classification are now visible, and the remaining details should be a matter of time.

Computations suggest that the equivalences in this paper, proved and conjectured, are the "major" ones. We have not found any others that occur for large, fairly irregular classes of shapes.

In Chapter 7 especially, and also throughout the paper, we suggest a number of other interesting problems in this framework. Certainly, many more questions are waiting to be asked.

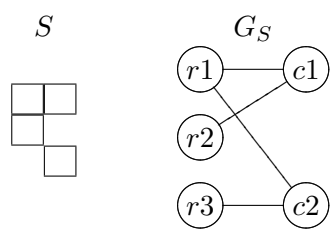
CHAPTER 2

Basic Definitions

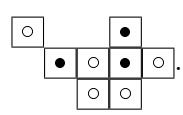
2.1. Shapes and fillings

Definition 2. An $m \times n$-*grid shape* S is a subset of boxes selected from a $m \times n$ 2-dimensional square grid. From now on, we will call these simply *shapes*.

Definition 3. The graph G_S of an $m \times n$-shape S is a bipartite graph on $m + n$ vertices. The first part of G_S, with m vertices, corresponds to the rows of S. The second part, with n vertices, corresponds to the columns. There is an edge in G_S between row i and column j iff S has a box in that position.

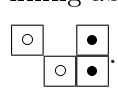


Definition 4. A *filling* F of a shape S places $\circ$ or $\bullet$ (alternative notations: $\circ$ or $\times$, 0 or 1) in every cell of the shape:



Notation 5. Let c be a cell of a shape S or filling F. Then, $S \backslash c$ and $F \backslash c$ denote the shape or filling with the cell removed.

Definition 6. Deleting some rows and columns from a shape or filling makes a *minor*. They are specified with a row set and column set. Here is the $\{1, 2\} \times \{1, 3, 4\}$ minor of the filling above:



The minors of S are exactly the induced subgraphs of G_S.

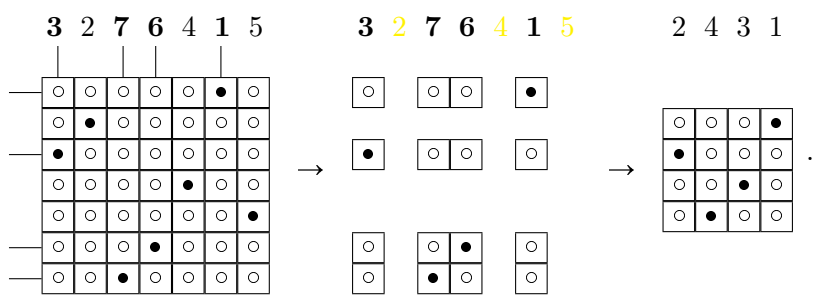
Definition 7. A $k \times k$-*step* in a graph consists of moving from a cell c to another cell d so that both are in the same $k \times k$ minor with all k^2 cells present. A shape is $k \times k$-*connected* if one can get from any cell to any other cell by a sequence of $k \times k$-steps. For the purpose of 2×2 pattern avoidance,

it will be enough to prove theorems for 2×2-connected shapes. Such shapes have a 4-cycle-connected G_S — one can get from any edge to any other edge by walking along adjacent edges of 4-cycles.

DEFINITION 8. A grid shape S is *horizontally connected* if, after removing the shape's empty columns, the cells in every row form a single, unbroken block. *Vertical connectivity* is analogous. A shape is *convex* (or grid-convex, or orthoconvex) if it is both horizontally and vertically connected. This neither implies, nor is implied by $k \times k$-connectivity.

2.2. Patterns

Our definition of patterns for fillings generalizes standard permutation pattern-avoidance. A permutation $\sigma \in S_n$ can be represented as a "rook diagram". We take an $n \times n$ shape and fill it with $\circ$s, except for a $\bullet$ in each cell i, j such that $\sigma(i) = j$. For example, the $3276415 \in S_7$ contains a 2431 instance: 3761 in positions 1346. The pattern corresponds to the $\{1, 3, 6, 7\} \times \{1, 3, 4, 6\}$ minor of the rook diagram:



A permutation is 2431-avoiding if and only if its rook diagram avoids this minor. The generalization of such patterns to general shapes and fillings is simple:

DEFINITION 9. A *pattern* p is a filling of a shape. A filling F contains the pattern p if some minor of F equals p.

The natural question is to characterize the number of fillings of a fixed shape S that avoid (or contain) certain p. We will focus on a specialization of this problem.

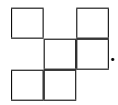
DEFINITION 10. A 2×2-*pattern pair*, further called *pattern pair,* or *pp* for short, is an unordered pair of 2×2 patterns. For example: $\left(\begin{smallmatrix} \circ & \bullet \\ \bullet & \circ \end{smallmatrix} \middle| \begin{smallmatrix} \bullet & \circ \\ \circ & \bullet \end{smallmatrix}\right)$. A filling F avoids a pattern pair $(p_1|p_2)$ if it contains neither p_1 nor p_2. We will call F a P-*avoiding filling*, or a P-*paf* for short.

Fix a shape S and a pattern pair P. In this paper, we will not count the number of fillings of S that avoid P (although that is an interesting question, see [**Ki04, KMV05**]). Instead, we will study the equivalence of pps:

DEFINITION 11. $P_1 = (p_1^1|p_2^1)$ and $P_2 = (p_1^2|p_2^2)$ are *equivalent*, denoted $P_1 \leftrightarrow P_2$, if the fillings that avoid P_1 are equinumerous with P_2-pafs.

Now it is clear why we may prove theorems only for 2×2-connected shapes:

REMARK 12. A 2×2-disconnected shape contains a pattern iff one of its components does. A special case is when the component has exactly one cell. We call such cells *detached*. This shape consists entirely of detached cells:

 .

The first observation about equivalent pps is that we can take a "complement" of any pp, by replacing $\circ$ with $\bullet$, and vice-versa. For example, the complement of $P = \left(\begin{smallmatrix}\circ & \bullet \\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet \\ \bullet & \circ\end{smallmatrix}\right)$ is $\bar{P} = \left(\begin{smallmatrix}\bullet & \circ \\ \bullet & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ \\ \circ & \bullet\end{smallmatrix}\right)$.

FACT 13. *Given a shape S and a pp P, P-pafs are equinumerous with $\bar{P}$-pafs. The bijection is obvious: F is a P-paf iff $\bar{F}$ is a $\bar{P}$-paf.*

Since P and $\bar{P}$ are necessarily equivalent, we will identify every pp with its complement. Now, in order to enumerate all pps, we will number the single patterns. The pattern

$$p = \begin{matrix} a & b \\ c & d \end{matrix}$$

will be assigned number $n(p) = a + 2b + 4c + 8d = dcba_2$, where $\circ$ is 0 and $\bullet$ is 1. Then, $n(\bar{p}) = 15 - p = 1111_2 - dcba_2$ is the number of its complement. For consistency, we will write pps as $(p_1|p_2)$ with $n(p_1) < n(p_2)$.

There are $2^4 = 16$ single patterns, and consequently $\frac{16 \cdot 15}{2} = 120$ pattern pairs. There are 8 self-complementary pps, which have $n(p_1) + n(p_2) = 15$. So, after identifying complements, we are left with $\frac{120-8}{2} + 8 = 64$ classes of pattern pairs.

CHAPTER 3

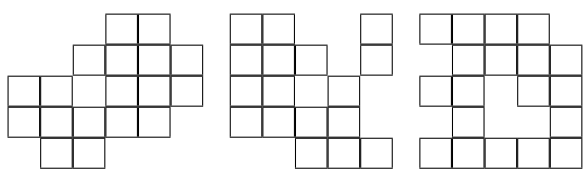
Classifying Pattern Pair Equivalences

3.1. Fundamentals

First, we describe some very simple and universal phenomena in grid shape pattern avoidance. It would be a hassle to mention them for every non-trivial equivalence we prove. So, please keep these in mind:

Complement: Given a a pattern p, taking the complement (exchanging $\circ \leftrightarrow \bullet$) produces a pattern pair $\bar{p}$, which is equivalent to p in all shapes. We are identifying these (see Fact 13). We will normally use as representatives the 2×2 pattern pairs $p = (p_1|p_2)$ with $p_1 + p_2 < 15$.

No universal equivalences: No other equivalences hold in irregular shapes:



.

Careful: the two shapes on the left have single-cell holes. See Section 7.1 for some empirical results on the frequency of such shapes.

The symmetry group of patterns/shapes is generated by row order reversal and transposition (e.g. column reversals are row reversals conjugated by transposition). Note that $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ is invariant under both maps.

Shape symmetry: If a shape is preserved by some transformation T (e.g. rotation by $90°$), then patterns p and q will be equivalent whenever $T(p) = q$.

Such equivalences are trivial to enumerate for any shape, or shape class of interest, so we will not discuss them further.

Equivalent equivalences: If $p \leftrightarrow q$ in shape class C, then for all transformations T, $T(p) \leftrightarrow T(q)$ in shape class $T(C)$.

It is enough to prove just one representative of this class of equivalences, though we will typically also list the equivalences obtained by symmetry.

3.2. New patterns and new shapes

In the Introduction, we mentioned two previously known equivalences. In Young diagrams, $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \circ & \circ\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$, proved recursively in [**Po06**], and bijectively in [**Jo08**]. Also in Young diagrams, $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \circ & \circ\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \circ & \bullet\end{smallmatrix}\right)$, proved bijectively on pages 10-11 of [**Bu07**] and pages 80-81 of [**Po06**].

The main point of this paper is that these results are part of a much bigger picture. The work began with a computer program to count pattern-avoiding fillings of shapes, which found a plethora of other equivalences.

The computer results led to several observations. Firstly, equivalence is not the norm — in the nicest of shapes, the square, there are 15 separate equivalence classes of pps. In many shapes, *all* pps are inequivalent. Secondly, after accounting for symmetry and complements, there are several equivalences unrelated to the ones above, which hold in nice (e.g. convex, Definition 8) shapes. Thirdly, these equivalences, and many more, hold in shapes far less regular than Young diagrams and rectangles.

3.3. What equivalences hold in a shape?

For a shape S, we can enumerate all fillings, count all pafs, and find their equivalence classes. However, counting acyclic orientations even in *planar* bipartite graphs is #P-hard [**VW92**]. On the other hand, testing e.g. whether S is a Young shape is trivial — so, we can check *sufficient* conditions for equivalences to hold.

We will find equivalences by checking *shape properties*, rather than through direct computation. There are two possible approaches:

Hard: For an equivalence, describe *all* shapes, in which it holds.

Easy: For a shape class, describe all equivalences, which hold in it.

The first problem is probably "wild", because, unless we can bound the number of p-pafs and q-pafs away from each other (see e.g. Problem 101), it is hard to rule out sporadic coincidences. We will note whether each of our results appears empirically tight on a limited set of test shapes — that is, whether we found shapes, in which an equivalence holds, but the result's hypotheses do not.

Our proofs will be structured to get the best of both strategies. For each equivalence, we will *try* to give a maximally general description of the shapes in which it holds. Then, we can apply these results to give *complete* descriptions of equivalences for interesting shape classes. Young shapes are described in Corollary 15, and many more classes should be possible — see Section 7.3.

3.4. Results

This paper proves five distinct equivalences, grouped into three chapters. An overview of the equivalences, and their symmetries (read Section 3.1 first!) is given in Figure 3.4.1 on page 20. The figure also includes two bijections due to other authors. Dashed lines correspond to equivalences proved with a recurrence, solid lines indicate bijective proofs. The marks on the arrows indicate the class of shapes, in which each equivalence holds:

凸[1] Convex shapes; every row and column is a single segment.

[1]Chinese for "convex".

⌟, ⌞, ⌜, ⌝	One can erase the whole shape by a recursive procedure, which a removes *special* lower-right, etc. corner cells, computes 2×2 components, and repeats.
⊏, ⊓, ⊐, ⊔	One can erase the whole shape by a similar recursive procedure, except the special cells are leftmost in their row, etc.

In particular, Young shapes are a very special subset of ⌟ shapes.

The best way to understand the implications of these results is to read the next subsection, which gives some example applications.

3.5. Example applications of the results

Our work strengthens both of the previously known equivalences. Theorem 25 proves that the $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \bullet & \circ\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ bijection holds for bottom-right CR-erasable shapes (Definition 16). This class is far larger than Young shapes, or even convex shapes with a single ▥. Additionally, the first bijection of Chapter 6 proves that $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \bullet & \circ\end{smallmatrix}\right)$ is equivalent to $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \bullet\\ \bullet & \bullet\end{smallmatrix}\right)$, its reflection across /, in convex diagrams. Taking the complement, this pp is equivalent to $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$, which is equivalent to $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right)$ in top-left CR-erasable shapes by Theorem **??**. So, we get:

COROLLARY 14. *If a shape is bottom-right CR-erasable, or convex* and *top-left CR-erasable, the pps* $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \bullet & \circ\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ are equivalent. *The second condition simply specifies a convex shape with a single bottom-right corner; this includes all top-left – bottom-right reflections of Young shapes.*

This chain of deductions shows that the equivalence holds for many more shapes than previously known. However, the shape hypotheses are also not too nice, and, empirically, not tight. This illustrates two important points: (i) the results in this paper say more about pp equivalences than is apparent at first glance; (ii) daisy-chaining results as above does not usually give the weakest possible restrictions. It turns out that we can actually integrate the bijection into a proof of a slightly modified recurrence, which is nearly tight empirically (Proposition 4.5).

The case of $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \bullet & \circ\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \bullet & \bullet\end{smallmatrix}\right)$ is similar. Burstein and Postnikov give essentially the same proof for Young shapes on pages 10-11 of [**Bu07**] and pages 80-81 of [**Po06**]. We take the complementary $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right)$, and apply Theorems 25 and 27 to find that the equivalence holds in all bottom-right CR-erasable shapes. Again, this does not seem to be tight empirically, but in this case we lost nothing by daisy-chaining, because the theorems have identical hypotheses.

The previous paragraphs prove equivalences in rather broad classes of shapes. The flip side of the coin is to describe equivalences that hold in a specific class of shapes. For example, take Young shapes. They are top-right, bottom-left, and bottom-right CR-erasable. Hence, Theorems 25, 27 and 29 all apply, and show that several patterns are equivalent to $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$. This

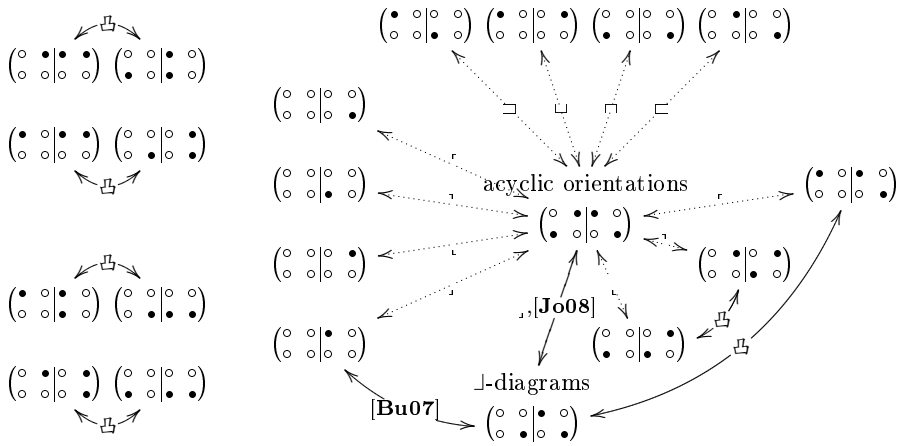


FIGURE 3.4.1. Pattern pair equivalences described in this paper, by section. In order to make the symmetries clearer, the table does not always show the canonical pp of a complementary pair. For example, we write $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ instead of the usual ⊥-diagram $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \bullet & \circ\end{smallmatrix}\right)$.

gives the equivalence class:

$$\left(\begin{smallmatrix}\circ & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right), \left(\begin{smallmatrix}\circ & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \bullet\\ \circ & \circ\end{smallmatrix}\right), \left(\begin{smallmatrix}\circ & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\right), \left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \bullet\\ \circ & \circ\end{smallmatrix}\right), \left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\right), \left(\begin{smallmatrix}\circ & \bullet\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right),$$
$$\left(\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right), \left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right), \left(\begin{smallmatrix}\circ & \bullet\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\right), \left(\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right), \left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right), \left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right).$$

Moreover, the second bijection of Chapter 6 applies, giving four more classes:

$$\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \bullet\\ \circ & \bullet\end{smallmatrix}\right), \left(\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \bullet\\ \circ & \bullet\end{smallmatrix}\right) \quad \left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\right), \left(\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \bullet\\ \circ & \bullet\end{smallmatrix}\right) \quad \left(\begin{smallmatrix}\circ & \bullet\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right), \left(\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \bullet & \bullet\end{smallmatrix}\right) \quad \left(\begin{smallmatrix}\circ & \bullet\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right), \left(\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \bullet & \bullet\end{smallmatrix}\right).$$

Explicitly computing the pp equivalences for the Young diagram $(6, 6, 4, 2, 2, 2)$, we found that these are the *only* equivalence classes.

COROLLARY 15. *The above equivalence classes describe all the equivalences that hold for every Young diagram (also, see the note about symmetry in Section 3.1).*

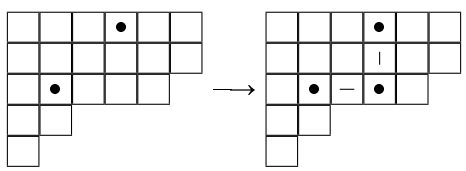
CHAPTER 4

$\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \bullet & \circ\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$: ⅃-diagrams and Acyclic Orientations

This pair of pps started it all. Their equivalence in Young shapes was first proved by Alex Postnikov [**Po06**], using a recurrence for ⅃-diagrams found by Lauren Williams [**Wi05**] and his analogous recurrence for acyclic orientations.

4.1. ⅃-diagrams

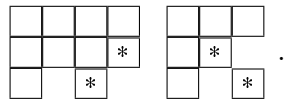
Originally, a ⅃-diagram was defined to be a binary filling of a Young diagram having the ⅃-property: if two cells located at the bottom-left and top-right corners of a rectangle contain $\bullet$, then the cell at the bottom-right corner (making a "⅃" shape) must also contain $\bullet$:



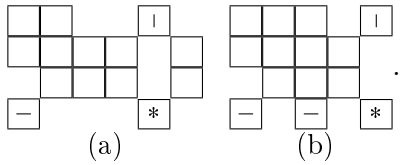
In our terminology, a ⅃-diagram is a $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \bullet & \circ\end{smallmatrix}\right)$-avoiding filling. This definition is valid for all shapes. There is a caveat here: in a Young diagram, one always has the upper-left cell of a 2×2 minor. In general, it is quite possible that it's missing, but our definition requires this cell to be present. Another way to interpret the ⅃-property is that the upper-left cell is irrelevant, and need not even be present. We chose to have a complete 2×2 minor because this naturally preserves the connection between acyclic orientations and ⅃-diagrams — the proof for acyclic orientations *requires* the upper-left cell to be present. Nonetheless, it would be an interesting generalization to permit incomplete minors, though this would only make a difference in skew and other shapes with more than one ▥.

Lauren Williams introduced the polynomial $F_S(q)$, where the coefficient of q^k counts the number of $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \bullet & \circ\end{smallmatrix}\right)$-avoiding fillings of shape S that contain k $\bullet$s. She gave a simple recurrence for $F_S(q)$ in Young diagrams. We will now see that this recurrence generalizes to a much larger class of shapes.

The recurrence starts at a bottom-right corner c of the shape — that is, the cell must be rightmost in its row, and bottommost in its column. In the two shapes below, the cells marked with $*$ are such corners:

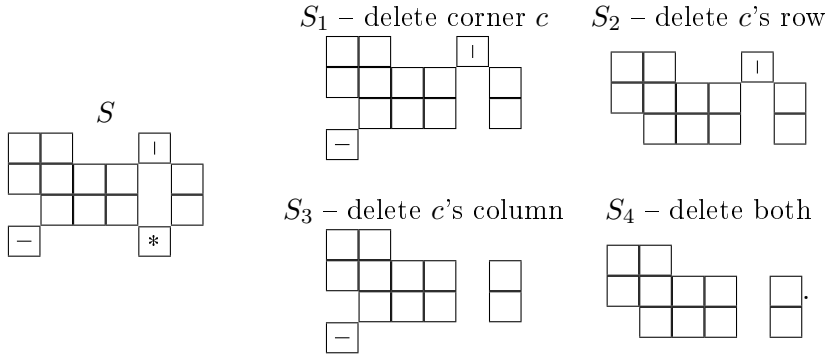


The cells above c and the cells to the left of c, ignoring discontinuities, form the bottom and right edges of a rectangle. In these shapes, the corner c is marked with $*$, while the edges are marked with dashes:



In order for the recurrence to work, all the cells in this rectangle must be present as in (a) and (b). We will call such rectangles *complete*. Because of Remark 12, we will also require the shape to be 2×2-connected. To compute $F_S(q)$, the recurrence requires the values of $F_S(q)$ on four smaller shapes, illustrated on the example (a) above:

(4.1.1)

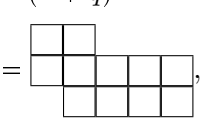


These deletions may render the shape 2×2-disconnected. To compute F_{S_i}, we will split S_i into its 2×2-connected components $S_i^{(j)}$, each a separate shape, and multiply their polynomials:

$$F_{S_i}(q) = \prod_j F_{S_i^{(j)}}(q). \tag{4.1.2}$$

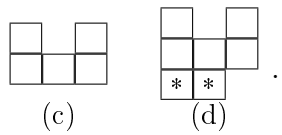
Some of the components will be detached cells, each of which will contribute a factor of $(1+q)$, because it may be filled with either $\circ$ or $\bullet$, independently of any other cell. In our example, the four shapes simplify to one (modulo detached cells, which are responsible for the $(1+q)^i$ factors):

$$S_1^{(1)} = S_2^{(1)} = S_3^{(1)} = S_4^{(1)} =$$



$$\begin{aligned} F_{S_1} &= (1+q)^2 F_{S_1^{(1)}} & F_{S_2} &= (1+q) F_{S_2^{(1)}} \\ F_{S_3} &= (1+q) F_{S_3^{(1)}} & F_{S_4} &= F_{S_4^{(1)}} \end{aligned}.$$

Now, the recurrence (which we will define in Lemma 18) may be used to compute F for every $S_i^{(j)}$. The decomposition of S_i into $S_i^{(j)}$ is necessary to cover a larger class of shapes. Without taking 2×2-components, it would be impossible to apply the recurrence to shapes like (c), and to compute F for shapes like (d) that reduce to (c):



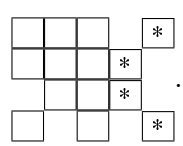
This points out an important problem: for some shapes, we may be unable to repeatedly apply the recurrence all the way down to the empty shape. The worst case is repeated expansion along S_1 (delete corner); in order for it to succeed, this definition must apply:

Definition 16. A shape is *bottom-right complete rectangle-erasable* (brCR-erasable for short) if all of its 2×2-components satisfy the following recursive rule. In each component $S^{(j)}$, considered as a separate shape, we can find a special bottom-right corner c with two properties. Firstly, the corner must have a complete rectangle. Secondly, the shape $S^{(j)} \backslash c$ must be bottom-right CR-erasable. As the base case, the empty shape is brCR-erasable.

Lemma 17. *If $S^{(j)}$ is a bottom-right CR-erasable 2×2-connected component, use the special corner c to obtain S_1, S_2, S_3, and S_4. Then, all the S_i are brCR-erasable.*

Proof. S_1 is brCR-erasable by definition. In S_2, we have deleted a whole row R; nonetheless, it is still CR-erasable. To see this, erase S_1 and S_2 in lockstep. Let d be the cell about to be deleted in S_1 (actually, a sub-sub-...-sub-component of S_1, because the deletion process fragments the shape). If d was in R, then there is nothing to do in S_2. Otherwise, d still has a complete rectangle in the sub-...-component of S_2, because deleting a whole row cannot make a rectangle incomplete. The argument showing that S_3 and S_4 are brCR-erasable is analogous. □

The shape (b) introduced above is not bottom-right CR-erasable. We can eliminate the following cells marked with $*$, and will get stuck with a single corner that has an incomplete rectangle:



The recurrence cannot be computed, and indeed, for this shape there are 5566 $\left(\begin{smallmatrix}\circ & \bullet \\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet \\ \bullet & \circ\end{smallmatrix}\right)$-pafs, but 5476 $\left(\begin{smallmatrix}\circ & \bullet \\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ \\ \circ & \bullet\end{smallmatrix}\right)$-pafs. The reader should check that, in contrast, the recurrence succeeds for the shape (a).

We have now completely described what the recurrence needs to compute $F_S(q)$. It remains to describe how it works:

LEMMA 18. *If the shape S is bottom-right CR-erasable, then the generating polynomial of $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \bullet & \circ\end{smallmatrix}\right)$-avoiding fillings of S with k $\bullet$s can be computed using* only *the recurrence*

$$F_S(q) = \prod_j \left(qF_{S_1^{(j)}}(q) + F_{S_2^{(j)}}(q) + F_{S_3^{(j)}}(q) - F_{S_4^{(j)}}(q) \right), \tag{4.1.3}$$

where $S^{(j)}$ are the 2×2-connected components of S considered as separate shapes, and $S_i^{(j)}$ are copies of $S^{(j)}$ after deleting the special corner, the cells in its row, column, and row plus column, as discussed above. The initial condition is $F_\varnothing(q) = 1$.

PROOF. Most of the proof is done already: we saw that F_S is a product over 2×2-connected components, and we showed that the $S_i^{(j)}$ are brCR-erasable. It remains to explain the expression inside the parentheses.

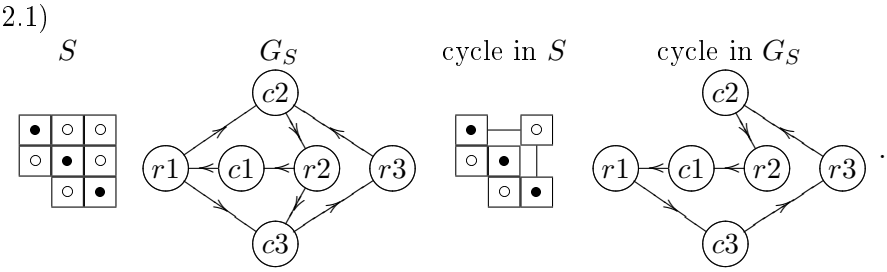
Consider a particular $S^{(j)}$; by Definition 16, it has a special bottom-right corner c with a complete rectangle (the next cell to be deleted). If the corner contains $\bullet$, then no forbidden pattern can involve this corner (because both patterns have $\circ$ in the bottom-right corner). So, the number of such pafs is $F_{S_1^{(j)}}(q)$, and we multiply it by q to account for $\bullet$ in the corner.

If the corner contains $\circ$, this constrains the cells above and to the left of c. Since the shape contains c's complete rectangle, $\bullet$ must not be simultaneously present in both the column above and in the row to the left of c. Either c's row or c's column must consist entirely of $\circ$s. If it is the row, then it cannot participate in a forbidden pattern — both patterns have at least one $\bullet$ in each row. The number of ways to fill the remaining cells is enumerated by $F_{S_2^{(j)}}(q)$. The reasoning for the column case is identical, and that contributes $F_{S_3^{(j)}}(q)$. However, this double-counts the case when both the row and the column are filled with $\circ$s, so we subtract $F_{S_4^{(j)}}(q)$. □

4.2. Acyclic orientations

Recall from Definition 3 that a bipartite graph G_S corresponds to each shape S. A filling of the shape gives an orientation: the edge points from a row to a column if its cell contains $\circ$, and from a column to a row otherwise. A cycle in this graph corresponds to a sequence of cells in the filling alternating between "same column, different row" and "same row, different column", with contents (independently) alternating between $\circ$ and $\bullet$. Here is an example:

(4.2.1)



Note that in this example, there is also a 4-cycle: $c1 - r1 - c2 - r2$, which corresponds to the minor $\{r1, r2\} \times \{c1, c2\}$ in the filling. Every 4-cycle is a 2×2 minor filled with one of the two patterns $\left(\begin{smallmatrix} \circ & \bullet \\ \bullet & \circ \end{smallmatrix}\middle|\begin{smallmatrix} \bullet & \circ \\ \circ & \bullet \end{smallmatrix}\right)$. So, 4-acyclic fillings are exactly the $\left(\begin{smallmatrix} \circ & \bullet \\ \bullet & \circ \end{smallmatrix}\middle|\begin{smallmatrix} \bullet & \circ \\ \circ & \bullet \end{smallmatrix}\right)$-avoiding fillings. We will soon see that 4-cycles are present in every cyclic filling of a large class of shapes. But first, we need to extend the "complete rectangle" that we defined for bottom-right corners of ⌟-diagrams:

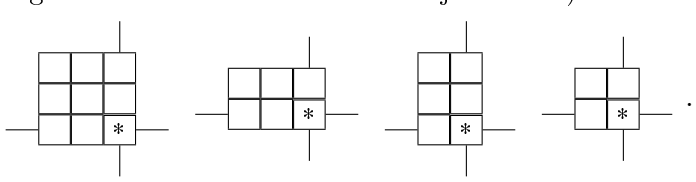
Definition 19. A cell c in shape S has a *complete rectangle (CR)* if for every choice of c_r from c's row, and c_c from c's column, there is a cell c_{rc} in S at the intersection of c_c's row and c_r's column. Here is an example, with c marked by $*$, the c_rs marked by $-$, c_cs marked by $|$, and the c_{rc}s left blank:

			|	
−	−	−	*	−
			|	

This implies that each choice of c, c_r, c_c, c_{rc} is a 2×2 minor, and that the vertices of the edges adjacent to c in G_S induce a complete bipartite subgraph. We will call such cells *CR-cells*.

Lemma 20. *Let S be a 2×2-connected shape with a CR-cell c. Then $S' = S \backslash c$ has at most three 2×2-connected components, all but one of which are detached cells that are leaf edges of $G_{S'}$.*

Proof. Let n_r and n_c be the number of cells (including c) in c's row and column, respectively. Both are at least 2 — otherwise c wouldn't be 2×2-connected. There are 4 cases: $n_r, n_c > 2$; $n_r = 2, n_c > 2$; $n_r > 2, n_c = 2$; $n_r = n_c = 2$. We will use some pictures to illustrate them, and will always place c as the bottom-right-most cell. This is legitimate, because 2×2-connectivity is a property of G_S, and as such is invariant under rearrangements of rows and columns. Here are rectangles of c in each case (> 3 neighbors in either direction works just like 3):



The "$*$" marks c, while "$-$" and "$|$" emphasize that there are no other cells in those rows and columns. In the first case, removing c will leave a 2×2-connected shape. If that were not true, some two cells d and e that are in a 2×2 minor together with c would become disconnected. But, every 2×2 minor involving c belongs to its rectangle. With $n_r > 2, n_c > 2$, the rectangle remains 2×2-connected after deleting c, and no such d and e exist.

The second and third case are symmetric, so we will cover only $n_r = 2, n_c > 2$. The cell c_c in c's column becomes detached — after c is deleted, c_c is the only cell in its column, and thus cannot be in a 2×2 minor (one component). Moreover, c_c's column is a leaf vertex in $G_{S'}$, because c_c is

its only edge. The other cells in the rectangle stay interconnected through minors not involving c. So, the rest of the filling, $S'\backslash c_c$, is 2×2-connected (a second component).

The fourth case is not much different from the second and third. The cells c_r and c_c are left alone in their row and column, respectively, and therefore become detached (two components, two leaf edges). The connections of the remaining cell c_{rc} to the rest of the shape are intact, and $S'\backslash\{c_r, c_c\}$ is therefore the third component.

□

Now, we extend the notion CR-erasability from Definition 16 to allow cells other than bottom-right corners.

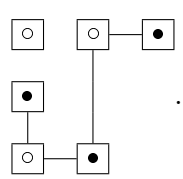
DEFINITION 21. Let P be a cell predicate, such as "bottom-right"; we will omit P to mean "any". A shape is P *complete rectangle-erasable* (P CR-erasable for short) if all of its 2×2-components satisfy the following recursive rule. In each component $S^{(i)}$, considered as a separate shape, we can find a special CR-cell c satisfying P, such that $S^{(i)}\backslash c$ is P CR-erasable. The empty shape is P CR-erasable.

From Lemma 20, we see that if a 2×2-connected shape S is CR-erasable, the deletion procedure is particularly simple. First, we delete some CR-cell from S. Then, we delete the resulting detached cells, and we are once again left with a 2×2-connected CR-erasable shape.

LEMMA 22. *Suppose that S is a 2×2-connected, CR-erasable shape. Then, any cyclic filling of S contains a 4-cycle.*

PROOF. Let F be a filling of S containing a cycle C. Let c be the special CR-cell given by by CR-erasability.

Case 1: Suppose that c belongs to the cycle C. Then, the row and column of c must each contain another cell from the cycle — c_r and c_c, respectively. Without loss of generality, we may assume that c is filled with $\bullet$ (otherwise, take the complementary filling – it will have the same cycles). Thus, c_r and c_c are filled with $\circ$. Now, consider the cell c_{rc} in c_r's column, and c_c's row (it exists because c has a complete rectangle). If c_{rc} is filled with $\bullet$, the four cells c, c_r, c_c, c_{rc} are a 4-cycle, and we are done. So, assume that c_{rc} is filled with $\circ$. Since cycles alternate rows and columns, there must be a further element of the cycle c'_r in c_r's column. Similarly, we get c'_c in c_c's row. Both c'_r and c'_c must be filled with $\bullet$. In this illustration, the bottom-right corner is c, the detached cell is c_{rc}:



Therefore, we can replace $c'_r - c_r - c - c_c - c'_c$ by $c'_r - c_{rc} - c'_c$ in C to obtain a shorter cycle C', which avoids c. This brings us to case 2.

Case 2: If c does not belong to the cycle, we can delete it. This might create a detached cell in c's row or column, or a cell in each, as in the proof of Lemma 20. Suppose c_c is the detached cell from c's column. Then, c_c could not have been in C either, because $c_c \in C$ implies that there is a second cell $d \in C$ in c_c's column. But, the only possibility for d is c, and $c \notin C$. An analogous argument shows that the cycle does not pass through the detached cell in c's row. Thus, the cycle lies entirely in the remaining 2×2-connected component F'. This new filling is strictly smaller, and satisfies our initial assumptions, so we may repeat the argument. After finitely many iterations, the number of cells will become $\leqslant 4$, but the only cyclic filling on $\leqslant 4$ cells is the 4-cycle. $\square$

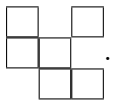
The proof of Lemma 22 can be modified slightly to obtain a result with different assumptions:

Corollary 23. *Suppose that the shape S can be erased by deleting a CR-cell, and repeating the procedure on the resulting shape (without breaking it into 2×2-components). The reader is invited to check that, in particular, this condition holds for convex shapes (see Definition 8). Then, a 4-acyclic filling of S is acyclic.*

Proof. We can use the unmodified "Case 1" from the above proof. If we end up making a shorter cycle that does not pass through c, we delete c. The resulting filling still has a deletion sequence of CR-cells, so we win by induction on the size of the filling. $\square$

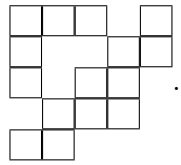
Although the hypotheses of Lemma 22 and Lemma 23 sound similar to Definition 16, they are not more or less general. Specifically, in these results, the cells that we delete do not have to be corners. But, this comes with extra assumptions: 2×2-connectedness, or erasability without decomposing into 2×2-components.

For shapes satisfying one of these conditions, acyclic orientations are the $\left(\begin{smallmatrix} \circ & \bullet \\ \bullet & \circ \end{smallmatrix}\middle|\begin{smallmatrix} \bullet & \circ \\ \circ & \bullet \end{smallmatrix}\right)$-avoiding fillings. As with $\lrcorner$-diagrams, this pattern-avoidance model is not perfect — in this case, if the shape is arbitrary, we cannot express acyclicity in terms of small minors. For example, one *needs* a 3×3 minor to detect a cycle in this shape (see the example (4.2.1)):

(4.2.2)  .

It has 64 $\left(\begin{smallmatrix} \circ & \bullet \\ \bullet & \circ \end{smallmatrix}\middle|\begin{smallmatrix} \bullet & \circ \\ \circ & \bullet \end{smallmatrix}\right)$-avoiding (4-acyclic) fillings, and 62 acyclic ones. It is CR-erasable, but is not 2×2-connected, and cannot be CR-erased without splitting into 2×2-components. Thus, it demonstrates that the extra assumptions in Lemma 22 and Lemma 23 are necessary. The following shape

is 2×2-connected, but is not CR-erasable:



It also has more 4-acyclic fillings – 14894 – than acyclic ones – 13790. It might be an interesting combinatorial problem to describe the graphs of such shapes.

4.3. Recurrence for $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$-avoidance

In Lemma 24.2 of [**Po06**], Alex Postnikov proved a recurrence for the chromatic polynomial $\chi_{G_\lambda}(t)$ of the graph of a Young diagram. He then specialized it to obtain a recurrence for the number of acyclic orientations of G_λ. We will generalize his result to all CR-erasable shapes. However, the chromatic polynomial does not decompose across 2×2-components. If it did, the recurrence for the chromatic polynomial would hold for all CR-erasable graphs, and we would get the same recurrence for $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$-avoiding fillings as for acyclic fillings. But, this is impossible as shown by example (4.2.2).

Therefore, we have to specialize to $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$-avoidance straight away.

LEMMA 24. *Let A_S be the number of $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$-pafs of shape S. If the shape S is CR-erasable, then A_S can be computed using* only *the recurrence*

$$A_S = \prod_j \left(A_{S_1^{(j)}} + A_{S_2^{(j)}} + A_{S_3^{(j)}} - A_{S_4^{(j)}}\right), \tag{4.3.1}$$

where $S^{(j)}$ are the 2×2-connected components of S considered as separate shapes, and $S_i^{(j)}$ are copies of $S^{(j)}$ after deleting the special cell, the cells in its row, column, and row plus column, just like in Lemma 18. The initial condition is $A_\varnothing = 1$.

PROOF. We need to show that for every 2×2-connected shape $S^{(j)}$ with a CR-cell c, the number of $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$-pafs is given by the quantity in the parentheses. The rest comes together just like in Lemma 18.

By Lemma 22, it is enough to compute the number of acyclic fillings of $S^{(j)}$. By Postnikov's Lemma 24.2 [**Po06**], the chromatic polynomial $\chi_{S^{(j)}}$ of $G_{S^{(j)}}$ can be written as

$$\chi_{S^{(j)}}(t) = \chi_{S_1^{(j)}}(t) - \frac{1}{t}\left(\chi_{S_2^{(j)}}(t) + \chi_{S_3^{(j)}}(t) - \chi_{S_4^{(j)}}(t)\right). \tag{4.3.2}$$

Technically, Postnikov's proof was written for a corner of a Young shape, not a CR-cell of a 2×2-connected shape. However, his proof uses only the structure of G_S, and disregards the positions of rows and columns. Therefore, it generalizes without modifications to any shape with a CR-cell. Further following Postnikov, we specialize (4.3.2) to $t = -1$, to obtain a relation in terms of the numbers of acyclic orientations $(-1)^n \chi_{S^{(j)}}(-1)$. The exponent

n is the number of vertices in the graph of $S^{(j)}$, and because we only delete the edges, the graph of $S_i^{(j)}$. So, if ao_S is the number of acyclic fillings of shape S, we get

$$ao_{S^{(j)}} = ao_{S_1^{(j)}} + ao_{S_2^{(j)}} + ao_{S_3^{(j)}} - ao_{S_4^{(j)}}.$$

That is not quite the end — we need to show that acyclic fillings and $\left(\begin{smallmatrix}\circ & \bullet & \circ\\ \bullet & \circ & \bullet\end{smallmatrix}\right)$-pafs are equinumerous in the sub-shapes $S_i^{(j)}$ (for the left-hand side, we know this already). For $S_1^{(j)}$, look back at the proof of Lemma 20. The shape has one large CR-erasable 2×2-connected component, and at most two leaf edges. No cycle can pass through leaf edges, and the big component is okay by Lemma 22.

The proofs for $S_2^{(j)}$, $S_3^{(j)}$, and $S_4^{(j)}$ are slight modifications of the same argument, which we omit due to space limitations. Briefly, the shape remains 2×2-connected after these whole-row or whole-column deletions, and the proof that the new shape is CR-erasable is just like Lemma 17. Thus, Lemma 22 applies. □

4.4. Equivalence of $\left(\begin{smallmatrix}\circ & \bullet & \bullet\\ \bullet & \circ & \circ\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}\circ & \bullet & \circ\\ \bullet & \circ & \bullet\end{smallmatrix}\right)$, and its symmetries

Following [**Po06**], we now specialize (4.1.3) with $q = 1$ to get a recurrence counting the number of ⊥-diagrams of shape S:

$$\perp_S = \prod_j \left(\perp_{S_1^{(j)}} + \perp_{S_2^{(j)}} + \perp_{S_3^{(j)}} - \perp_{S_4^{(j)}} \right). \tag{4.4.1}$$

This holds for all bottom-right CR-erasable shapes. Such shapes are, of course, CR-erasable, and so the $\left(\begin{smallmatrix}\circ & \bullet & \circ\\ \bullet & \circ & \bullet\end{smallmatrix}\right)$ recurrence also applies. The recurrences are identical, and have the same initial conditions, $\perp_\varnothing = A_\varnothing = 1$. That proves $\left(\begin{smallmatrix}\circ & \bullet & \bullet\\ \bullet & \circ & \circ\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \bullet & \circ\\ \bullet & \circ & \bullet\end{smallmatrix}\right)$, though by symmetry, we get more (see Section 3.1):

THEOREM 25. *The following equivalences hold:*

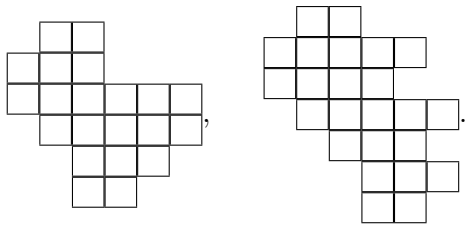
(1) $\left(\begin{smallmatrix}\circ & \bullet & \circ\\ \bullet & \circ & \bullet\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \circ & \circ\\ \circ & \bullet & \bullet\end{smallmatrix}\right)$, *complement of* $\left(\begin{smallmatrix}\circ & \bullet & \bullet\\ \bullet & \circ & \circ\end{smallmatrix}\right)$, *in brCR-erasable shapes.*
(2) $\left(\begin{smallmatrix}\circ & \bullet & \circ\\ \bullet & \circ & \bullet\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \circ & \circ\\ \bullet & \circ & \bullet\end{smallmatrix}\right)$ *in bottom-left CR-erasable shapes.*
(3) $\left(\begin{smallmatrix}\circ & \bullet & \circ\\ \bullet & \circ & \bullet\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \bullet & \circ\\ \circ & \circ & \bullet\end{smallmatrix}\right)$ *in top-right CR-erasable shapes.*
(4) $\left(\begin{smallmatrix}\circ & \bullet & \circ\\ \bullet & \circ & \bullet\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\bullet & \circ & \circ\\ \circ & \circ & \bullet\end{smallmatrix}\right)$ *in top-left CR-erasable shapes.*

□

Empirically, the characterization of these equivalences in terms of CR-erasable shapes is not tight. However, we will improve it substantially in the next section.

4.5. $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ is stronger with $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$

In the previous section we proved directly that $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ is equivalent to $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ in brCR-erasable shapes. Additionally, we can get an indirect equivalence: $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$. At face value, this holds in convex top-left CR-erasable shapes (i.e. convex shapes with a single ▥). That is certainly an improvement — most such shapes are not brCR-erasable. However, many shapes like the following have the $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ equivalence, but are neither brCR-erasable nor convex tlCR-erasable:



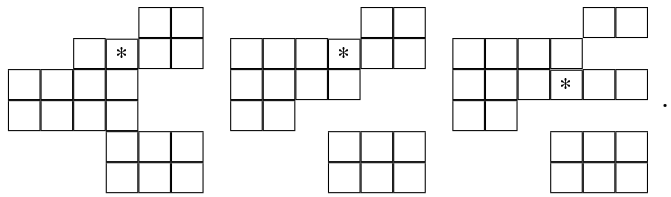
However, we can still explain these! Recall the $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ recurrence ($\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \circ & \circ\end{smallmatrix}\right)$, really, but we will keep talking of the complement). Bottom-right corners were needed, because otherwise, we could not guarantee that we would still have a $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$-paf, once the corner (or its row/column) were added back. Suppose we just deleted a bottom-right CR-cell (and possibly its row/column). Imagine also that one of the resulting components is not brCR-erasable, but is convex tlCR-erasable. That is enough! We can continue the recurrence with top-left CR corners, showing that $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$-pafs of this component are equinumerous with $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$-pafs, and then biject the $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$-pafs to $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$-pafs, and continue with other parts of the $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ recurrence as though nothing happened. Note that in both of the example shapes above, we can delete brCR-cells (and possibly rows/columns) until we get a convex tlCR-erasable shape. Formalizing this idea, we get:

PROPOSITION. $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ *holds in a shape, if its* 2×2*-components can be recursively deleted by the following procedure, which takes a predicate* P *as input. The initial value of* P *is "bottom-right".*

(1) *Delete some* P *CR-cell, and consider the* 2×2 *components separately.*
(2) *Try to apply this recursion to each component, with the same* P*.*
(3) *For the components, which could not be deleted, check if they are convex:*
   No: *The procedure fails.*
   Yes: *Recurse on this component, with the opposite* P *(br* $\leftrightarrow$ *tl).*

Of course, the 3 *symmetric equivalences are strengthened analogously.*

Empirically, this appears close to tight. We found the following family of exceptions:



There are two 2×2-connected components *which share a single edge* $*$. It seems important that the components be convex and tl- or br-CR-erasable. Also, $*$ might have to be a top-left corner of one of the shapes, and on the right edge of another. This is not the most general possible description of such shapes, but a brief summary of our experiments.

4.6. Same recurrence: $\left(\begin{smallmatrix}\circ & \circ \\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ \\ \circ & \circ\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}\circ & \bullet \\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ \\ \circ & \bullet\end{smallmatrix}\right)$

We can get a recurrence for counting $\left(\begin{smallmatrix}\circ & \circ \\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ \\ \circ & \circ\end{smallmatrix}\right)$-pafs by exactly the same method. There is only a small difference in the way the counts are refined by the contents of the fillings:

Lemma 26. *If the shape S is bottom-right CR-erasable, then $O_S(q)$, the generating polynomial of $\left(\begin{smallmatrix}\circ & \circ \\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ \\ \circ & \circ\end{smallmatrix}\right)$-avoiding fillings of S with k $\circ$s can be computed using* only *the recurrence*

$$O_S(q) = \prod_j \left(O_{S_1^{(j)}}(q) + q \left(O_{S_2^{(j)}}(q) + O_{S_3^{(j)}}(q) - O_{S_4^{(j)}}(q) \right) \right), \tag{4.6.1}$$

where $S^{(j)}$ are as in Lemma 18, and the initial condition is $O_\varnothing(q) = 1$.

Proof. The proof is completely analogous to that of Lemma 18. If the corner contains $\bullet$, we are free to delete it — this makes the $S_1^{(j)}$ term. Otherwise, the corner's row or column (or both) consists entirely of $\bullet$s. Such a row or column cannot participate in either pattern of the pp. So, we get the remaining three terms (the deleted $\circ$ corner gives the factor of q). $\square$

We specialize the recurrence with $q = 1$, rotate the pattern pair, and get all the analogous equivalences:

Theorem 27. *Not listing complements, the following pps are equivalent to $\left(\begin{smallmatrix}\circ & \bullet \\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ \\ \circ & \bullet\end{smallmatrix}\right)$: for bottom-right corner CR-erasable shapes — $\left(\begin{smallmatrix}\circ & \circ \\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ \\ \circ & \circ\end{smallmatrix}\right)$, bottom-left — $\left(\begin{smallmatrix}\circ & \circ \\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \bullet \\ \circ & \circ\end{smallmatrix}\right)$, top-left — $\left(\begin{smallmatrix}\circ & \circ \\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ \\ \circ & \bullet\end{smallmatrix}\right)$, top-right — $\left(\begin{smallmatrix}\circ & \circ \\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ \\ \bullet & \circ\end{smallmatrix}\right)$.* $\square$

Empirically, we found no shape with equivalent $\left(\begin{smallmatrix}\circ & \circ \\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ \\ \circ & \circ\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}\circ & \bullet \\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ \\ \circ & \bullet\end{smallmatrix}\right)$, such that the shape is not bottom-right CR-erasable.

4.7. Similar recurrence: $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$

The pair $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\right)$ requires two changes. Firstly, if this pp is present in a shape S, then it also belongs to any S' obtained by permuting rows of S. This is because swapping the rows of the pp does not change it. So, the relevant requirement for a cell c this time a *right complete rectangle* — c is a CR-cell that is rightmost in its row. The recurrence lemma in this case does not give a nice way to count pafs by the number of ∘s or •s they contain.

LEMMA 28. *If the shape S is right CR-erasable, then the number R_S of $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\right)$-pafs of S can be computed using* only *the recurrence*

$$R_S = \prod_j \left(R_{S_1^{(j)}} + R_{S_2^{(j)}} + R_{S_3^{(j)}} - R_{S_4^{(j)}}\right), \tag{4.7.1}$$

where $S^{(j)}$ are as in Lemma 18, and the initial condition is $R_\varnothing = 1$.

PROOF. Again, we need to justify the decomposition into the four subshapes. If c is the special right CR-cell, and it contains •, it is not involved in forbidden patterns and can be deleted to make $S_1^{(j)}$. Now comes the second change from the previous proofs. If c contains ∘, and another cell d in its column contains ∘, then every other cell in c's row must be identical to the corresponding cell in d's row. Indeed if the two cells in one column were different, we would get a pattern from $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\right)$. Thus, there are two cases: either c's column consists of •s, or c's row is fully replicated by another row. In the first case, no cells in the column can participate in forbidden patterns, and we can delete the column to make $S_3^{(j)}$. In the second case, if a forbidden pattern involves c's row, there is a copy of this pattern using d's row instead of c's row. So, the pattern would have to be present in $S_2^{(j)}$ to show up in $S^{(j)}$. Therefore, we can delete the bottom row, and count fillings of $S_2^{(j)}$. Just as before, this double-counts the case where the bottom row is replicated by some other row *and* the column consists of •s; that's $S_4^{(j)}$. □

THEOREM 29. *Omitting complements, the following pps are equivalent to $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$: for right CR-erasable shapes — $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\right)$, left — $\left(\begin{smallmatrix}\circ & \bullet\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right)$, top — $\left(\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right)$, bottom — $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right)$.* □

Empirically, we found no shape such that $\left(\begin{smallmatrix}\circ & \bullet\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ is equivalent to $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$, but the shape is not left CR-erasable.

CHAPTER 5

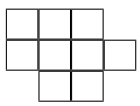
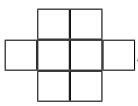
A Bijection between $\left(\begin{smallmatrix}\circ & \bullet\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \circ & \circ\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \bullet & \circ\end{smallmatrix}\right)$

5.1. Preliminaries

We will use the two pattern pairs a great deal, so the following abbreviations will be very useful:

Definition 30. Use $\boxplus$ to denote $\left(\begin{smallmatrix}\circ & \bullet\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \circ & \circ\end{smallmatrix}\right)$, and $\boxplus$ to denote $\left(\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \bullet & \circ\end{smallmatrix}\right)$.

In this section, we will only deal with *convex* shapes (any row or column has a single span of cells). It turns out that the bijection is much simpler when the shape has only one upper-left corner:

 rather than  .

We will first solve this case, and then build on it to prove the more general result.

5.1.1. Common strategy. Broadly speaking, a paf is an non-overlapping arrangement of $\boxed{\circ}$ and $\boxed{\bullet}$ objects on a grid shape. Our bijection will stay within this paradigm: a non-overlapping arrangement of objects on the very same shape. The objects will be horizontal/vertical lines, and rectangles, with somewhat involved rules on where they can go.

The rules for $\boxplus$- and $\boxplus$-pafs are related by reflection across the upper-left – lower-right diagonal. All we need is a way to map pafs to arrangements whose rules are invariant under this reflection. If we have a map for $\boxplus$-pafs, the map for $\boxplus$-pafs would, of course, be symmetric — so, the desired bijection is $\boxplus$-paf $\leftrightarrow$ reflection-invariant arrangement $\leftrightarrow$ $\boxplus$-paf. It is like an involution with a twist.

In the one upper-left corner case, there is a simple bijection from pafs to a line arrangement with symmetric rules.

With multiple upper-left corners, we will need a sequence of simple bijections, from one kind of arrangement to the next. Each time, the rules of the new arrangement will be more invariant under \-reflection. Eventually, we get an arrangement with symmetric rules, and we can interpret it as having come from either a $\boxplus$- or a $\boxplus$-paf.

We will formalize the rules after each step, and identify which of them change under reflection — each time, this suggests a more symmetric representation. Formalizing the rules for new arrangements takes some effort, but it is essential for clarity — otherwise, one would have to keep coming

back to paf rules. With the resulting complexity, one would never be sure that the proofs are correct.

5.1.2. Corner notation. We will be talking about corners of shapes (and subshapes) a lot, so it will help to have a precise, compact notation. Given a grid shape, it can have "out" corners , and "in" corners . The shaded areas are inside the shape, the white — outside.

We will use corners to specify locations — if something is "at" the corner, we mean the corner's point. E.g., the location refers to the midpoint of the following configuration of cells:

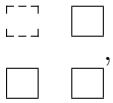
.

We will also refer to a corner's cell. For "out" corners that is the cell *inside* the shape that contains the corner's point. For "in" corners, that is the cell *outside* of the shape, which contains the corner's point.

5.1.3. Empirical tightness. This result seems tight: on our test shapes, the equivalence held iff the shape broke up into convex 2×2-connected components, or was preserved by \-reflections.

5.2. Simple bijection: one upper-left corner

A convex shape with a single upper-left corner is special because for any three cells in the shape arranged in a ⌟,

 ,

the upper-left corner box (shown dashed) will always be present.

This shape constraint lets us reformulate the rule for $\left(\begin{smallmatrix}\circ & \bullet\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \circ & \circ\end{smallmatrix}\right)$-avoidance in a more useful way. Note that the forbidden patterns are identical except for the upper-left corner. Thus, it is enough exclude a single pattern, consisting of the remaining 3 cells:

$p =$.

By the shape restriction, the upper-left corner will always be present, and thus filled with either $\circ$ or $\bullet$, so forbidding this one pattern is equivalent to forbidding the original two.

We will restate this restriction once more. Call a cell b bad if it is the bottom-right cell of an instance of p. A filling avoids p if it has no bad cells. A cell cannot be bad if:

(1) It contains $\bullet$.
(2) It contains $\circ$, but only has $\circ$s in the column above.
(3) It contains $\circ$, but has only $\bullet$s in the row to its left.

These conditions are not just sufficient, but also necessary: a violation immediately gives an instance of p.

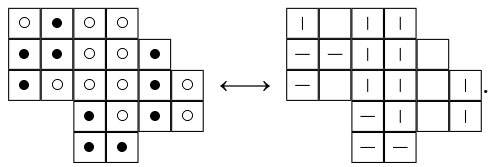
DEFINITION 31. Since a $\circ$-cell is not guaranteed to be good, it must be *justified*, either by a column of $\circ$s above it, or by a row of $\bullet$s to its left.

FACT 32. *A filling avoids* $\left(\begin{smallmatrix}\circ & \bullet\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \circ & \circ\end{smallmatrix}\right)$ *if and only if each of its* $\circ$*s is justified.*

DEFINITION 33. If a $\circ$ is justified by a column of $\circ$s above it, call the union of these cells a *v-line*. On the other hand, if a $\circ$ is justified by a row of $\bullet$s to its left, call the union of the $\bullet$-cells an *h-line*. If a row consists entirely of $\bullet$s, let the whole be involved in the h-line.

Instead of drawing a $\left(\begin{smallmatrix}\circ & \bullet\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \circ & \circ\end{smallmatrix}\right)$-avoiding filling, we can instead draw these lines on the same shape.

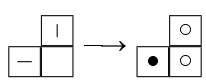
(5.2.1)



This is a bijection between $\boxplus$-pafs, and a set of particular line arrangements.

LEMMA 34. *Every* $\boxplus$*-paf of a single-upper-left-corner shape corresponds to a* non-overlapping *arrangement of v-lines and h-lines with the following properties:*

(1) *Every column contains a v-line, which starts from the top and occupies* 0 *or more contiguous cells (A v-line consists of* $\circ$*s.)*
(2) *Every row contains an h-line, which starts on the left and occupies* 0 *or more contiguous cells. (A h-line consists of* $\bullet$*s. If the cell immediately to the right of the h-line is in the shape, it contains a* $\circ$*.)*
(3) *An h-line may not end in the cell one below, one to the left of the bottom cell of a v-line (because the v-line would then include this cell). I.e., this line configuration is forbidden:*



We call this a line arrangement.

PROOF. Using the definitions of v-lines and h-lines, it is very easy to check that going from a $\boxplus$-paf to a line arrangement, all four properties hold.

To prove that this is a bijection, we will show that (i) given a line arrangement satisfying the above properties, we can go back to a filling, and that (ii) the filling corresponds to the same line arrangement. Going back to a filling is very simple: all $|$s and the first empty cell to the right of a line of $-$s become $\circ$s (for $l_r = 0$ imagine a $-$ before the start of every row), while all other cells become $\bullet$s. Lines do not overlap, so there is only one way to fill each cell. The filling will be $\boxplus$-avoiding, since every $\circ$ is necessarily justified by lines. To see that it yields the same line arrangement, it

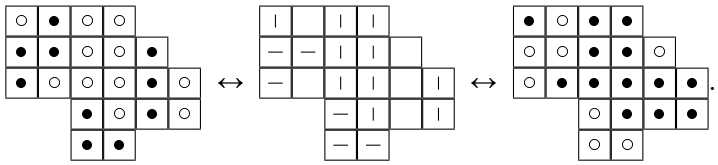
is enough to check that the line lengths in the filling are the same as in the initial arrangement.

Let us first check h-lines. Since lines do not overlap, all cells in the line will be filled with $\bullet$s. The cell to the right of the line may (i) contain a $|$ — which becomes a $\circ$, (ii) be empty — which also becomes a $\circ$, (iii) be outside the shape. In all three cases, the resulting filling generates an h-line of the same length as before.

In v-lines, every cell automatically becomes a $\circ$ in the filling. So, the lines in the filling will be at least as long as the original lines. A v-line could get longer if another $\circ$ appeared immediately below its last cell. But, this $\circ$ would have to be justified by an h-line, and this is forbidden by the rule (3). $\square$

These rules are not changed by a reflection across the upper-left – lower-right diagonal. So, exactly the same rules describe $\boxplus$-pafs, except that h-lines now consist of $\circ$s, and v-lines consist of $\bullet$s (with a $\circ$-cell neighboring the line on the bottom). However, the set of valid line arrangements is unchanged.

Since this bijection maps both $\boxplus$-pafs and $\boxplus$-pafs to valid line arrangements, it is actually a bijection between the two paf types. Here is the whole map, performed on the example from (5.2.1):



5.3. Complete bijection: multiple upper-left corners

5.3.1. The game plan. A line arrangement is not enough to represent fillings of these more general shapes. Some cells will need a new object, *rectangles*, to justify them. That will lead to a [forward] algorithm that represents pafs as line-rectangle arrangements (LRAs). It even has a natural backward algorithm; the hard part is to describe *which* LRAs originate from pafs. We will list some properties, which are obviously satisfied by all such LRAs. We will then prove their *sufficiency* by showing that, on these LRAs, the backward and forward algorithms are inverses. Our proofs will follow this pattern throughout.

So, we get a bijection between pattern-avoiding fillings and $\boxplus$- or $\boxplus$-LRAs. Unfortunately, the rules for $\boxplus$- and $\boxplus$-LRAs are not identical. We can make the two sets more similar by combining rectangles with certain lines adjacent to them into a new kind of object, *extended rectangles*. The rules for these *line-extended rectangle arrangements* (LERAs) are still not-quite identical — they differ in how rectangles may be adjacent to each other. The next step is to forget the boundaries between adjacent rectangles, making a *line-Popeye arrangement* (LPA). Remarkably, from the resulting arrangement, we can reconstruct *either* a $\boxplus$- or a $\boxplus$-LERA. So, the final

bijection will look like this:

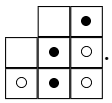
$$\boxplus\text{-paf} \leftrightarrow \boxplus\text{-LRA} \leftrightarrow \boxplus\text{-LERA} \leftrightarrow \text{LPA} \leftrightarrow \boxplus\text{-LERA} \leftrightarrow \boxplus\text{-LRA} \leftrightarrow \boxplus\text{-paf}.$$

5.3.2. Rectangle-justified cells. In the previous subsection, we introduced the notion of justifying ∘-cells — if no ∘-cell is the bottom-right corner of a forbidden pattern, then there are no forbidden patterns at all. We also saw how v- and h-lines can justify ∘-cells. We will relax the constraint that allowed us to forbid this 3-cell pattern instead of two 4-cell patterns:

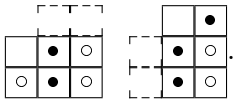
$p =$.

Now, this pattern is forbidden *only* if the dashed box is present in the shape.

A shape with more than one upper-left corner cell has at least one . In this case, asking that every ∘ cell be justified by a line is too strong a constraint. For example, the bottom-right ∘ in the following filling is not justified by a line (fill the blank cells either way):

 .

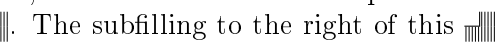
However, the cells above and to the left of this ∘ are not completely unconstrained. Consider two *subfillings*, one obtained by deleting all cells below the , the other — by deleting all cells to its right:

 .

Both the subshapes have a single upper-left corner, so the previous subsection applies, and a ∘ has to be justified by a h- or v-line. However, in the "below" subfillings, the v-line only has to go up to the , and similarly in the "right" subfillings the h-lines only go left until that corner's column. In this way, a ∘ cell can be justified without being justified by lines in the complete filling.

In fact, this example demonstrated the *only* other way to justify cells. The following technical lemma proves that, if a ∘ is not justified by a line, then it must have a *pair* of "shortened" h- and v-lines, extending up to the row and left to the column of some .

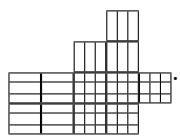
Lemma 35. *Suppose shape of the filling F has at least one* . *A ∘-cell c is justified in F (i.e. is not a bottom-right corner of a forbidden pattern) if and only if for some* , *c is justified by a line* both *in the subfilling below the corner, and in the subfilling to the right of it. If one of the subfillings does not contain c, we consider that subfilling to have justified the cell, vacuously (and at least one of the subfillings always contains c).*

Proof. First, we prove the "only if" direction. Suppose there is a ∘-cell c violating the condition above — there is no corner, for which it is line-justified in both subfillings. Then, we will find a forbidden pattern in the filling. Look at the rightmost . The subfilling to the right of this  has

just one ▥. So, by Section 5.2, if the ∘-cell were not justified by a line inside this subfilling, it would necessarily belong to a forbidden pattern. But by assumption, the ∘-cell is not line-justified in at least one of the subfillings — thus, it is not line-justified in the bottom subfilling.

We will proceed by induction, restricting our attention to this subfilling. But first, we need to make sure that c still violates the lemma's condition in the restricted shape. If there are any ∘s immediately above the ∘-cell, they do not go all the way up to the top of the restriction (otherwise, the cell would be justified in the bottom subfilling, a contradiction). Restricting to a bottom subfilling (or to *its* bottom subfilling, etc.) does not affect h-lines at all. It follows that the lemma's condition is also violated in the restriction, which has one less ▟ than the original. Repeating this argument, we arrive at a subfilling with one ▥, which still violates the lemma's assumptions. But, we have already seen that such a filling *must* contain a forbidden pattern, which completes the proof of "only if".

Now, for the "if" direction. We will assume that the filling contains a forbidden pattern, and show that it must then violate the conditions of the lemma. Consider a ▟ of the shape, and the corresponding two subfillings. The shape breaks up into three pieces, shown horizontally, vertically, and cross-shaded:



The instance of the forbidden pattern cannot simultaneously touch the top piece and the leftmost piece. Thus, either the bottom (▤ or ▦), or the right (▥ or ▦) subfilling will contain the forbidden pattern, and the ∘-cell will not be line-justified. That completes the proof. □

So, we now know a second way to justify cells. It can be called "justification by rectangle", because a single rectangle-justified ∘-cell c dictates a fixed filling for a whole rectangle to its upper-left.

To see this, consider the *rightmost* ▟, for which c is justified in both the bottom and right subfillings. That means that there's a line of ∘-cells going up to the row of the corner from c, and a line of •-cells going left to the column of the corner. These lines do not reach the top or left of the shape, respectively. Moreover,

Claim 36. In the line of ∘-cells going up from c, every cell is justified either by an h-line, or by a rectangle. In both cases, the leftward line of •-cells reaches the corner's column.

Proof. In the first case (•∘-line), it is obvious. In the second case (rectangle), suppose the line of •-cells ended before the corner's column. Then, there would be a ▟ to the right of c's rectangle corner, and c would have to be justified in this corner's subfillings as well, contradicting the assumption that its corner was rightmost with that property. □

So, above c there is a line of $\circ$-cells to the corner's row, and to the left of each of them, there is a line of $\bullet$-cells until the corner's column. All together, they make a rectangle, with the upper-left corner touching the ▟, and the bottom-right corner at C. The rectangle's rightmost column is filled with $\circ$s, and the rest — with $\bullet$s.

In fact, we have a "flag" of rectangles, each justifying the appropriate cell from the vertical stretch of $\circ$-cells. So, for simplicity, we will take the rectangle justifying the most cells, and think of it as justifying its entire right column. Here is the specific procedure we will use:

ALGORITHM 37. *If a $\circ$-cell is not justified by a line in a ⊞-paf, we can find the justifying rectangle as follows:*

(1) *Go down c's column cell-by-cell, for as long as the current cell contains $\circ$.*
(2) *From this bottom cell go up until the first cell that is not justified by an h-line. This is the bottom-right corner of the rectangle.*
(3) *From the bottom-right corner, go up cell-by-cell for as long as the current cell contains $\circ$.*
(4) *Find the leftmost cell in that cell's row.*
(5) *From the leftmost cell, go down to the cell whose upper-left corner is a ▟. This is the rectangle's top-left corner.*

The resulting rectangle will have all the expected properties:

LEMMA 38. *Consider a rectangle identified by Algorithm 37 in a ⊞-paf. Then:*

(1) *The rectangle is filled with $\bullet$-cells, except for a rightmost column of $\circ$-cells.*
(2) *The rectangle justifies all of its $\circ$s.*
(3) *Such rectangles do not overlap.*

PROOF.

(1) The rightmost column consists of $\circ$s by construction, and the rest of the rectangle is $\bullet$s by Claim 36, taking c to be the bottom cell found in step 1 of the algorithm.
(2) For every $\circ$-cell, note that we have a line of $\circ$-cells going up, and a line of $\bullet$-cells going left. Apply Lemma 35, using the rectangle's ▟ and these lines, to get the result.
(3) If two rectangles were to overlap, they would share some of their $\circ$-cells. Taking the union of their rightmost columns, and the right of their two ▟s, we get a valid rectangle — in fact, the one that would be found by the algorithm.

□

5.3.3. Line-rectangle arrangements. It follows that we can represent a ⊞-paf as a line-rectangle arrangement (LRA):

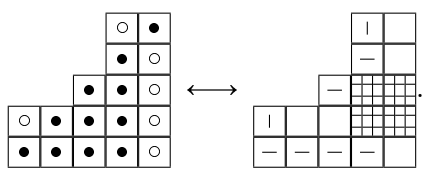
ALGORITHM 39. *Given a filling F,*

(1) *Find all rectangles using Algorithm 37.*
(2) *Remove all rectangle cells from the shape to make F'.*
(3) *In F', compute the line lengths according to Lemma 34 — since the rows and columns might no longer be connected, lines will only occur in the first continuous block.*

CLAIM 40. The lines and rectangles identified by Algorithm 39 justify all $\circ$-cells in the shape.

PROOF. Deleting the rectangle cells may make some rows and columns disconnected. However, in F', only the leftmost/topmost segments of the rows/columns will contain $\circ$s. Suppose not – then some $\circ$ is not justified by a rectangle in F, and is in neither a topmost nor a leftmost segment in F'. At the same time, it must have been justified by a line l in F, by Lemma 35. Thus, l must have passed through a rectangle, and exited on the other side. A rectangle's row is $\bullet \cdots \bullet \circ$, which means l was not an h-line. A v-line could only pass through the rightmost column of a rectangle, but that contradicts Algorithm 37. $\square$

Here is an example:



We have just described the forward map of a bijection between $\boxplus$-pafs and certain LRAs. The reverse map is straightforward:

ALGORITHM 41.

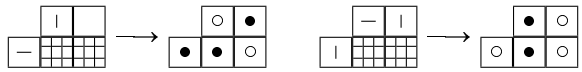
(1) Draw the lines on the original shape as in Section 5.2 — replace $|$s and first empty cells to the right of $-$s by $\circ$s (for $l_r = 0$ imagine a $-$ before the start of every row), and all other cells by $\bullet$s.
(2) Fill each rectangle with $\bullet$s. Then, fill every rectangle's rightmost column with $\circ$s. The first step overwrites the $\circ$s created inside of rectangles by the maximal h-lines; the second makes them rectangles.

We will now check that, starting with a $\boxplus$-paf, then applying the forward map, and the reverse map, we get back the same filling. We will do this by describing LRAs in their own terms, without referring to fillings.

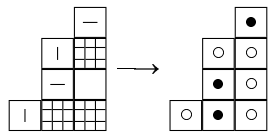
LEMMA 42. *Algorithms 39 and 41 describe a bijection which maps every $\boxplus$-paf to a* non-overlapping *arrangement of lines and rectangles with the following properties:*

(1) *Lines are as in Lemma 34.*
(2) *The top-left corner of every rectangle is at a* *of the shape.*

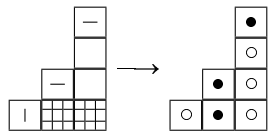
(3) *An h-line may not touch the left side of a rectangle's* ▥*, a v-line may not touch the top of a rectangle's* ▥*. Violations:*



(4) *Two rectangles with the same rightmost column C cannot be above one another so that every row between them has an h-line ending in the column $C-1$. A violation:*



(5) *If some horizontal lines end in the same column $C-1$, are located in adjacent rows, and the bottom line is adjacent to the top row of a rectangle whose right edge is in C, then the lines have the same length — by (2) above, equal to the rectangle's width. Moreover, the row immediately above these lines must extend as far left as the lines (and rectangle) do. A violation:*



We will call such arrangements ⊞-LRA*s.*

The rules for ⊞*-pafs are the same, up to reflection across the upper-left – lower-right diagonal. In particular, rules (1), (2), (3) are preserved by this reflection. We will refer to the rules for* ⊞-LRAs *as (1') – (5').*

We will restate this lemma's rules for extended rectangles in Lemma 49.

Proof. Going from a ⊞-paf to an arrangement, the definitions directly imply rules (1) and (2). Rule (3) is immediate from Algorithm 37. The other two need a comment.

For rule (4), suppose that the forward map produced two such rectangles, separated only by h-lines. First, note that the upper rectangle is necessarily narrower than the lower one, since a higher ▟ must be strictly to the right of a lower one. Now, consider a *combined* rectangle, going from the top ▟ to the bottom-right corner of the lower rectangle. In the filling, it must have contained •s everywhere except for ∘s in the rightmost column. Its bottom row's ∘ is not justified by a •∘-line. Thus, by Lemma 35, this combined rectangle should have been found by Algorithm 37, a contradiction.

Rule (5) ensures that the correct ▟ was selected as the rectangle's upper-left corner. The horizontal lines can be as wide as the rectangle, or shorter. If one of them is shorter, then Algorithm 37 would have chosen the ▟ just below this line. Similarly, if the shape had an ▟ immediately above the top line in this rule, that corner would have been chosen.

To complete a proof of this bijection, we need to show (a) that the reverse map makes a ⊞-paf out of any arrangement satisfying the above properties, and (b) that applying the forward map to this filling yields the original arrangement.

For (a): the argument in Lemma 34 shows that the first step of Algorithm 41 produces a filling — the alteration of the meanings of L_c and L_r makes no difference to the proof, because the algorithm does not consult these values, but only the lengths of the lines. Since rectangles are non-overlapping, the second step is also well-defined — each cell inside a rectangle will be overwritten exactly once. The filling avoids ⊞ because every ∘ in it is justified by construction.

For (b): to check that the forward and reverse maps compose to the identity, we must verify (i) that the same rectangles are identified in the resulting filling, and (ii) that line lengths are preserved. To do this, we first describe the properties of the two varieties of ∘-cells in the filling: line-justified, and rectangle-justified (those that are not line-justified).

Every line, except the L_r-long h-lines, corresponds to one or more ∘-cell in the filling. These ∘-cells lie outside rectangles, by definition of L_r and L_c, and are thus justified only by their incoming lines. Arguing just as in Lemma 34, the corresponding lines will be recognized correctly by Algorithm 39.

The cell to the right of a L_r-long line is either outside the shape, or inside a rectangle. In the former case, the entire row is filled with •s, and is recognized by Algorithm 39. In the latter case, the line creates a ∘-cell which is overwritten by a rectangle. If the rectangle is correctly reconstructed from the filling, then so is the L_r-long line, so it remains to check the rectangles. However, note first that the rectangle actually extends the L_r-long line by adding a $\bullet \cdots \bullet \circ$ sequence to its end. So, the ∘ in that line appears to be justified by an h-line — this will matter for Algorithm 37.

Every ∘-cell not contributed by a line comes from a rectangle — this is clear from Algorithm 41. Such a cell is additionally justified by a line in the filling *if and only if* there is an L_r-long line in its row. We discussed the "if" direction in the previous paragraph. For the "only if" direction — firstly, v-lines cannot justify rectangle ∘-cells by rule (3). Secondly, if the h-line in the row is shorter than L_r, then there already is a ∘ in the row to the left of the rectangle, and the rectangle's ∘-cell is not line-justified.

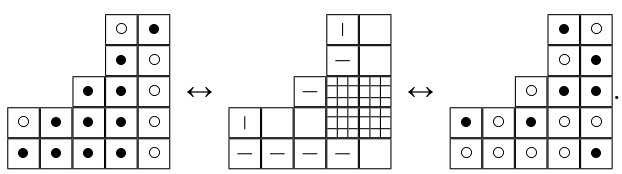
By steps (1) and (2) of the algorithm in Algorithm 37, the bottom-right corners of rectangles are exactly the lowest non-line-justified ∘-cells in column fragments consisting entirely of ∘s. By rule (3), and the preceding ∘-cell classification, the bottom-right corner c of a rectangle will always become a rectangle-justified cell. Additionally, by rule (4), there cannot be another rectangle-justified ∘-cell below c, unless there is a • between them. So, the bottom-right corners of rectangles in the filling will be the same as in the initial arrangement.

Next, we will check that top-left corners match up with the originals. We go to step (3) of Algorithm 37. Starting with some bottom right corner,

consider the height of its ○-cell column fragment. It is at least as tall as the original rectangle — every cell along the right edge of the rectangle is a ○-cell. Going above the rectangle, we might encounter some h-line-justified ○-cells. However, by rule (4), we will not see another rectangle-justified cell unless we pass a •-cell first. Moreover, by rule (5), we know that these lines will all have width equal to that of the original rectangle, which means that steps (4) and (5) of Algorithm 37 will find the original ▟▟▟. This proves that the rectangles found in the filling will be the same as in the original arrangement. □

Now that we have a bijection between ⊞-pafs and ⊞-LRAs, the next step is to note that we have a symmetrical bijection for ⊟-pafs. What remains is to find a bijection between ⊞-LRAs and ⊟-LRAs.

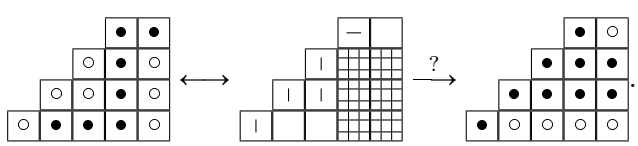
5.3.4. Conflicts between ⊞- and ⊟-LRA rules. In a perfect world, the rules for ⊞-LRAs and ⊟-LRAs would be identical, as with the line arrangements of Section 5.2. In Lemma 42, only rules (1), (2), and (3) remain unchanged between ⊞-LRAs and ⊟-LRAs. In some cases, this is enough — the same line-rectangle arrangement can be successfully interpreted as either kind:



Unfortunately, in many cases rules (4) and (5) can and do conflict with their ⊟ versions.

We will first show a conflict between rules (5) and (5'):

(5.3.1)

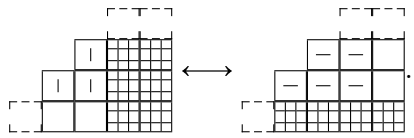


The ⊞-LRA violates (5') (and if we try to apply the ⊟ version of 41 anyway, the resulting filling has a 1×4 rectangle rather than the 3×2 rectangle in the LRA).

In general, a conflict in the ⊞ → ⊟ direction looks like this: (i) there is a rectangle with bottom row R and left column C; (ii) the vertical lines in columns $c \in [C-i, C-1]$ end in row $R-1$; (iii) there is a ▟▟▟ $\mathcal{C}$ whose left side touches $C-i$. Such arrangements obey rule (5), but not (5').

5.3.5. Extended rectangles. The idea is to map the cause-of-conflict to a ⊟-LRA, which would cause an analogous violation of rule (5). The cells involved in this conflict are: the rectangle, the vertical lines in columns $[C-i, C-1]$, and the i cells just below each of these lines. Below, the conflict cells from example 5.3.1 are not dashed. We need to change them to

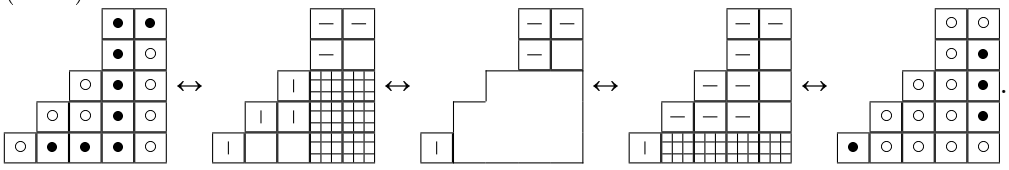
obey the reflected rule (5). The shape isn't symmetric, but we do not need to preserve the shape of rectangles, so this will do:



We can think of this as an (obviously invertible) transformation on the LRAs: for every conflict-causing rectangle, find the $\mathcal{C}$ as above, make it the rectangle's upper-left corner, and appropriate lines in the remaining cause-of-conflict cells.

Recall our strategy of building simple, symmetric bijections from objects with more disparate rules to objects with more similar rules. Here, it suggests that we encapsulate the entire conflict area in a single object, an *extended rectangle*. A first attempt, showing the object as a single tile:

(5.3.2)



The new object is still a rectangle, determined entirely by the positions of its top and bottom rows, left and right columns, but its upper-left part may land outside the shape.

Note that the second row of the ⊞-LRA (second diagram from right) looks like a conflict-causing line. It was left out of the extended rectangle because no touches it from above. For the purposes of reconciling rules (5) and (5'), we do not need this row.

Rule (4) hints that it would be a good idea to include these extra lines anyway:

Definition 43. Take a rectangle in an LRA, which has bottom-right corner (R, C). The corresponding *extended rectangle* will consist of:

(1) All consecutive h-lines adjacent to its top row, such that the lines end in column $C - 1$.
(2) All consecutive v-lines adjacent to its left column, such that the lines end in row $R - 1$.
(3) For each of the above lines, the cell one past the line's end.

The cells above the rectangle are the *top cap*; those to the left are the *left cap*.

Later, when we deal exclusively with extended rectangles, we will refer to them simply as "rectangles". The conversion to extended rectangles is invertible:

Algorithm 44. *To get lines and a plain rectangle from an extended rectangle:*

(1) *We know the original rectangle's bottom-right corner — it remains to find the upper-left. To obtain a* ⊞*-/*⊞*-LRA, use the highest/lowest* ▟ *that touches the extended rectangle.*
(2) *Fill the space above the plain rectangle with h-lines ending in column* $C-1$*. Fill the space to the left with v-lines ending in row* $R-1$*.*

COROLLARY 45. *An extended rectangle is entirely determined by the positions of its top and bottom rows, left and right columns.*

This second, and final, definition of extended rectangles reconciles rules (5) and (5'), as we planned. Now, it is time to see how these new objects interact with lines and with each other.

LEMMA 46. *Starting with two rectangles, independently find their extended rectangles according to Definition 43. The resulting extended rectangles do not overlap.*

5.3.6. Line-extended rectangle arrangements. Example 5.3.2 illustrates a bijection that takes any LRA to an equivalent line-extended rectangle arrangement (LERA). Here are the forward and backward maps:

ALGORITHM 47. ⊞*-LRA* $\longrightarrow$ ⊞*-LERA*

(1) *Identify the extended rectangles using Definition 43.*
(2) *In the top caps' rows, and the left caps' columns, set the line lengths to 0.*

ALGORITHM 48. ⊞*-LERA* $\longrightarrow$ ⊞*-LRA*

(1) *For every extended rectangle, identify the plain rectangle and the cap lines using Algorithm 44.*
(2) *Restore the cap lines to their original lengths.*

Together with Algorithm 47, we have a bijection.

We are now able to modify Lemma 42 to describe LERAs, and to prove that the above maps do specify a bijection:

LEMMA 49 (LERAs, version 1). *Algorithms 47 and 48 give a bijection that maps any* ⊞*-LRA to a* non-overlapping *arrangement of lines and extended rectangles with the following properties (changes marked in bold):*

(1) *Lines obey Lemma 34.*
(2) ***(modified)** Extended rectangles are still determined by top / bottom / left / right coordinates. But, part of the rectangle may now be outside the shape. The rectangle's* ▟, ▟, ▟ *must all be inside the shape. Also, the point of some* ▟ *of the shape must be inside the rectangle, or touch its edge.*
(3) *No h-line touches the left side of* ▟*, no v-line touches the top of* ▟*. Applied to* 0*-length lines, this rule implies that these two cells must not be on the upper-left boundary of the shape.*
(4) *Extended rectangles with the same rightmost column cannot be adjacent.*

(5) *(**new**) For an extended rectangle with bottom-right corner (R, C): an h-line immediately above it must not end in column $C - 1$, a v-line immediately to its left must not end in row $R - 1$.*

The objects that satisfy these rules are called ⊞-LERAs. *Symmetrically,* ⊞-*LERAs obey the reflected rules (1') – (5').*

This time, rules (1), (2), (3), (5) are preserved by reflection.

Proof. The substance of the lemma is practically identical to Lemma 42, so we only note the differences. Non-overlap: after LRA → LERA — extended rectangles do not overlap lines by Definition 43, each other by Lemma 46; the other direction is obvious.

For the LRA → LERA direction (Algorithm 47), we check Lemma 49 rules:

(1) The cap lines' lengths are set to 0 (they will revert correctly, see below).
(2) This rule follows directly from the construction of extended rectangles. First, we find the rectangles proper (Algorithm 37, Lemma 38). Identifying the caps (Definition 43) moves the top up and the left leftward, but never puts the rectangle's ▥ or ▥ outside the shape. Since the shape's ▟ starts at the ▥ of the rectangle, the growth may move it inside.
(3) Such lines touch the corners of an extended rectangle if and only if they continue to the plain rectangle inside.
(4) In a violation: in an LRA, some lines ending in $C - 1$ may separate the plain rectangles; they correspond to the top cap, meaning the extended rectangles are adjacent in a LERA.
(5) The original rule (5) is implicit in the construction of extended rectangles — we need it to recover the LRA from the LERA. As for the new rule, any line that could violate it would been included in the top cap or left cap.

For the LERA → LRA direction (Algorithm 48), we check Lemma 42 rules. We also verify that the bijection recovers the same LRA:

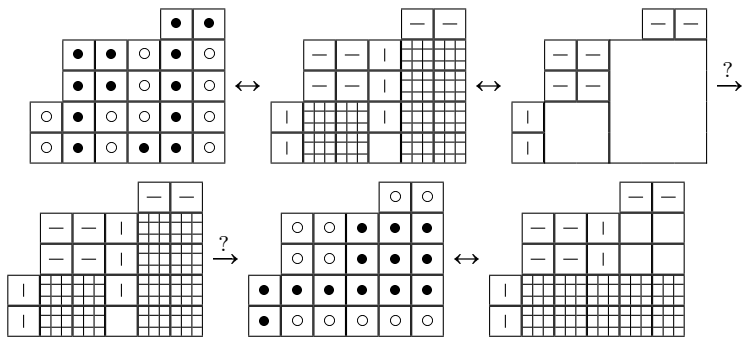
(1) The cap lines are properly reverted, as long as the rectangle is correctly reverted — see the next entry. The other lines are unchanged.
(2) The rectangle's ▥ is unchanged. Its ▥ lands at a ▟ of the shape by construction. The correct ▟ is chosen because Algorithm 44 matches the original rule (5).
(3-4) See the LRA → LERA notes.
(5) Algorithm 44 picks the highest (or lowest) ▟ exactly in order to satisfy this rule.

□

At the moment, we have bijections between pafs and LRAs, LRAs and LERAs. However, rules (4) and (4') still can and do conflict.

5.3.7. Conflicts between ⊞- and ⊞-LERA rules. By going from LRAs to LERAs, we replaced the asymmetric rule (5) of Lemma 42 with a new, symmetric rule in Lemma 49. However, rule (4) remains asymmetric. As a consequence, we cannot treat ⊞-LERAs as ⊞-LERAs — in the following example, two extended rectangles, 2×2 and 4×3, produce a ⊞-filling that maps to a ⊞-LERA with a single 2×5 extended rectangle:

(5.3.3)



We will use the same strategy as in Subsection 5.3.5: going from ⊞- to ⊞-LERAs, transform all situations that would violate rule (4') into violations of (4). First, we give these violations names:

Definition 50. Call a pair of adjacent extended rectangles ⊞-*exchangeable* if they have the same bottom column, or ⊞-*exchangeable* if they share the rightmost row.

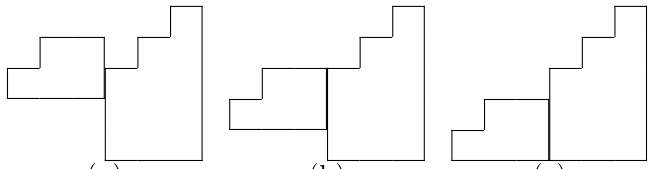
In ⊞- and ⊞-LERAs, such rectangles satisfy rule (4) and rule (4'), respectively, but violate the reflected rules. We will find a way to map both of these to the same, symmetric representation. But, first, we need:

5.3.8. Adjacent rectangles and elbow cells. Here, we discuss rectangles from LERAs.

Definition 51. If rectangles are adjacent along a horizontal line, call them *horizontally adjacent*. Otherwise, they are *vertically adjacent*.

Lemma 52. *Two adjacent extended rectangles make a* ▟ *(located inside the shape, by Corollary 53) at the junction between the top cap of the left (or bottom) rectangle, and the left cap of the right (or top) rectangle.*

Proof. Suppose that the rectangles are vertically adjacent. Consider three cases, showing the two rectangles, but not the shape:



(a) (b) (c)

Case (a), with the left rectangle "above" the right, is impossible, because the right rectangle's left cap starts at the shape's top — meaning that the shape

could not have been convex. Case (b), with both rectangles on the same level, is forbidden because it requires that the left rectangle's upper-left cell be on the shape's border, in violation of (3) of Lemma 49. Case (c), with the left rectangle "below" the right, has the , as desired.

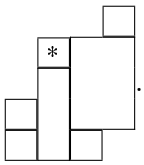
The argument for horizontally adjacent rectangles is symmetric. □

COROLLARY 53. *The cell immediately to the left and above the previous lemma's* *is always part of the shape.*

PROOF. Otherwise, one of the rectangles violates rule (3). □

Such cells turn out to be important, so we name them:

DEFINITION 54. Given two adjacent rectangles, let the *elbow cell* be the cell just to the left of and above the made by the left rectangle's top cap and the right rectangle's left cap. For example, the $\boxed{*}$ is an elbow cell:



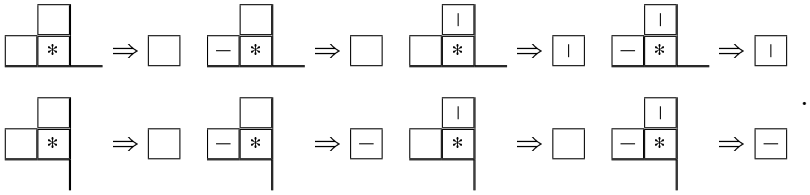
Aside from guaranteed existence, the other crucial result is that there is only one way to fill any elbow cell:

LEMMA 55. *Given the non-elbow cells of a LERA, there is a unique way to fill the elbow cells.*

PROOF. In LERA terms, a cell may contain $\boxed{|}$, $\boxed{-}$ or $\boxed{}$. In practice, there are at most two choices, because the paf has either a $\circ$ or a $\bullet$. Three pieces of context affect the elbow cell's filling:

(1) rectangle adjacency – horizontal or vertical,
(2) the cell above – $\boxed{}$ (equivalent to $\boxed{-}$) or $\boxed{|}$ (equivalent to "cell absent"),
(3) the cell to the right – $\boxed{}$ (equivalent to $\boxed{|}$) or $\boxed{-}$ (equivalent to "absent").

These suffice, because the cells above and to the right cannot be elbow cells. All 8 cases are completely determined by ⊞/⊟-symmetric rules: Lemma 34, and (3), (5) of Lemma 49. So, an elbow cell's filling is unique and independent of LERA type:



□

So, in fact, an elbow cell carries *no* information:

COROLLARY 56. *In a LERA, the positions of elbow cells are determined by the rectangles and the shape. The fillings are determined by the non-elbow cells.*

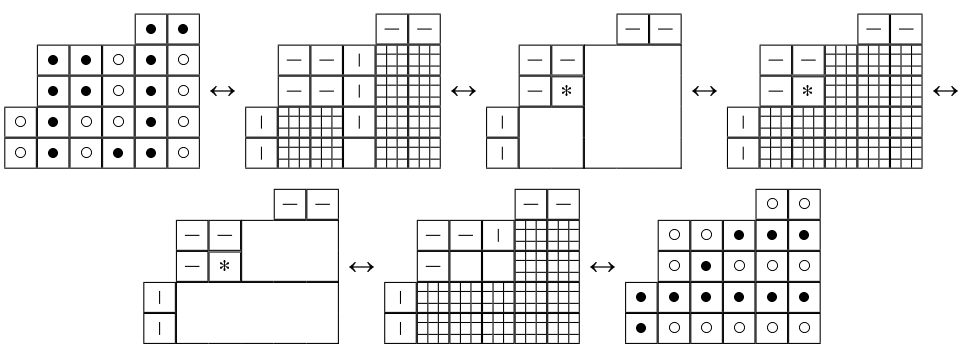
In subsequent drawings of LERAs, we will emphasize this by marking the cells with $\boxed{*}$ instead of $\boxed{}$, $\boxed{-}$ or $\boxed{|}$. As a side effect, such cells will no longer be able to violate rules (3), (5). That suggests an "optimized" variant of Lemma 49:

LEMMA 57 (LERAs, version 2). *A* ⊞*-LERA is a* non-overlapping *arrangement of:*

(1) *Lines as in Lemma 34,*
(2) *Extended rectangles as in rules (2) and (4),*
(3) *Elbow cells, as described by Corollary 56.*

Lines and rectangles must not make configurations described by rules (3) and (5).

5.3.9. Line-Popeye arrangements. Never mind the strange name; the relevant definitions will come later. Nonetheless, we are ready to introduce the final, symmetric representation: Line-Popeye arrangements (LPAs), which link ⊞- and ⊞-LERAs. In a LERA, we think of extended rectangles as disjoint objects, each defined by four parameters (top/bottom/left/right). For the LPA, we will forget that rectangles are disjoint, and only remember the entire set of rectangle cells, ▦. Those are enough to find the elbow cells (Corollary 67), which in turn enable us to reconstruct the rectangle boundaries (Algorithm 68). This forthcoming LPA ↔ LERA map fixes the previously broken example (5.3.3):



Understanding this last bijection will require some preparation.

5.3.10. Exchanging pairs of extended rectangles. Recall that we want to map ⊞- and ⊞-LERAs to a common, symmetrical representation. Before we do that, let us map pairs of exchangeable rectangles to each other:

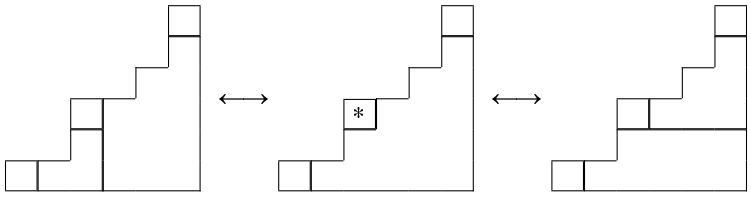
ALGORITHM 58. *The* exchange *operation is a bijection between a pair of* ⊞*- and* ⊞*-exchangeable rectangles:*

(1) *Find the elbow cell* $\boxed{*}$ *for this adjacent pair.*

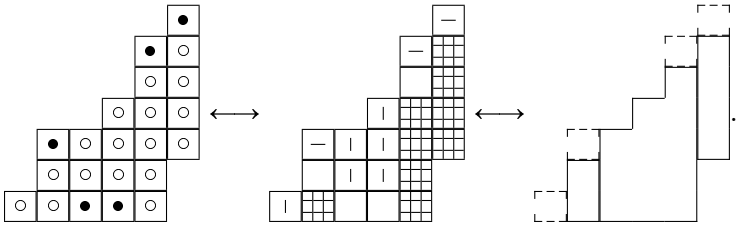
(2) *Consider the union of the rectangles, remembering the cell, and whether the rectangles were separated by a horizontal or vertical line.*

(3) *Break the union into two rectangles again, using a line in the* other *direction, starting from the bottom-right corner of the elbow cell.*

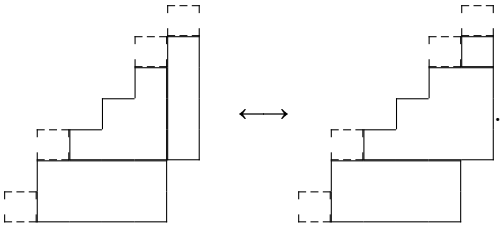
For example:



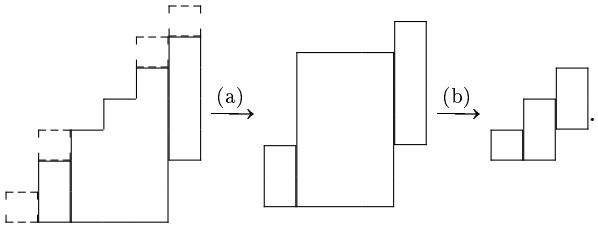
It is quite natural to try to exchange away all ⊞-exchangeable rectangle pairs, and expect to get a ⊞-LERA. That is almost right, but the argument will be much cleaner if we go through LPAs. To see why, consider the extended rectangles of this filling, forgetting about the line cells:



Only one exchange is possible, but it *creates* another ⊞-exchangeable pair:



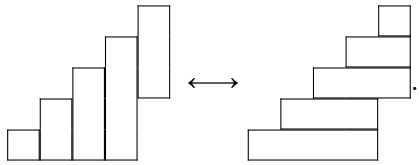
The combinatorics of this process is quite interesting, and depends only on the relative positions of the rectangles. So, we may (a) discard the shape information, and (b) drop redundant rows and columns, remembering which rectangles have shared rows or columns, but forgetting the rectangles' original sizes:



The exchange operation remains intact through these transformations, and operates the same way on the resulting objects. These are called *Popeye*

diagrams. We discuss them in detail in Chapter 8. The following corollary to Proposition 104 is relevant to the present bijection:

COROLLARY 59. *Any maximal sequence of vertical-to-horizontal exchanges will transform a Popeye diagram with vertical internal lines into one with horizontal internal lines, and vice-versa, e.g.*



For the purposes of our LERA $\leftrightarrow$ LPA bijection, this transformation will be atomic, but it is interesting to know that it can be implemented by performing all possible exchanges in any order.

5.3.11. Popeye rectangle arrangements. In Subsection 5.3.9, we said that line-Popeye arrangements do not record rectangle boundaries, but only the set of rectangle cells. Since an LPA is intended to be equivalent to a LERA, the cells must obey certain properties. Most concisely: they must have come from a set of rectangles that obeys the ⊞-LERA rules of Lemma 49 (or, equivalently — as we shall see later, the ⊞-LERA rules).

The rectangle cells of a LERA are determined by (a) the top / bottom / left / right coordinates of its rectangles, and (b) the shape. In this subsection, we will examine the rectangles without shape information, which drastically simplifies the ⊞-LERA rules:

DEFINITION 60. A ⊞-*Popeye rectangle arrangement (PRA)* is a set of rectangles such that:

(1) They do not overlap.
(2) No two upper-left corners share the same row or column. Furthermore, when sorted from left to right, these corners occur at increasing heights.
(3) Two rectangles with the same right column cannot be adjacent.

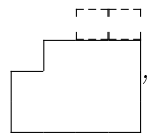
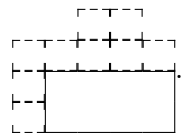
⊞-*Popeye rectangle arrangements* replace the last rule with: "Two rectangles with the same bottom row cannot be adjacent."

Parts (1) and (3) follow trivially from Lemma 49 rules (2) and (4). For part (2), note how Lemma 49 constrains the placement of rectangles in shapes:

COROLLARY 61. *An extended rectangle with (top, bottom) rows* (r, R) *and (left, right) columns* (c, C) *has the corners* (R, c), (R, C), *and* (r, C). *Then,*

(1) *By (3): These 3 corners lie in the shape, away from its upper-left edge — that is, these cells' upper/left edges must not face the outside of the shape.*

(2) *By (2): There must be a* ▟ *of the shape, which* is weakly *to the right of and below* ▟ (r, c).

E.g., , *but not*  , *nor*  .

Since the shape is convex, the shape's ▟s obey Popeye rule (2). By Corollary 61, so do the rectangles' ▟s.

We just verified that the extended rectangles of a LERA form a Popeye rectangle arrangement. And, of course, this PRA's placement on the shape follows Corollary 61.

The reader should check that this description of the rectangles is not only necessary, but also sufficient to replace Lemma 49 rules (2) and (4).

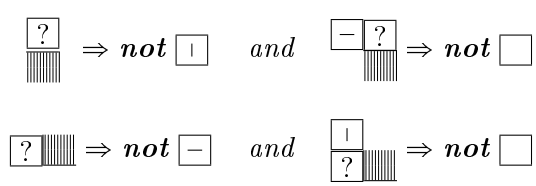
Also, in preparing to discard rectangle boundaries, we will rewrite Lemma 49 rules (3) and (5) in terms of cells. They concern *only* the non-elbow, non-rectangle cells touching a rectangle's ▟ from above, or a ▟ from the left.

These remarks let us update Lemma 57 so that it does not refer to Lemma 49 :

LEMMA 62 (LERAs, version 3). *A* ⊞*-LERA is a* non-overlapping *arrangement of:*

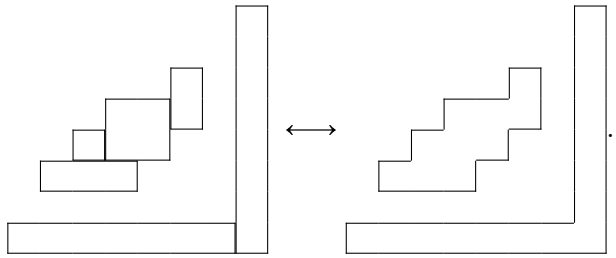
(1) *Lines as in Lemma 34,*
(2) *A* ⊞*-Popeye rectangle arrangement restricted to the shape according to Corollary 61,*
(3) *Elbow cells, as described in Corollary 56.*

A few line-rectangle interactions are forbidden. Namely, if a non-elbow, non-rectangle cell $\boxed{?}$ *is adjacent to a rectangle's* ▟ *or* ▟ *in one of the four ways below, then it* cannot *be filled as indicated:*



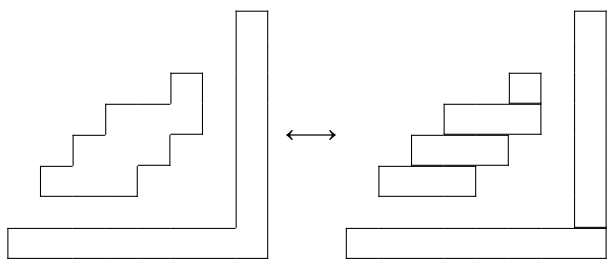
5.3.12. Popeye sets. Now that we have rectangles without shape information, we can go further and forget the rectangle boundaries:

DEFINITION 63. Given a ⊞- and ⊡-Popeye rectangle arrangement, drop all boundaries between adjacent rectangles to obtain a *Popeye set*. For example,

(5.3.4) 

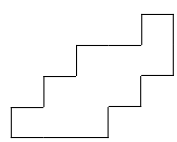
The key result that enables the LERA ↔ LPA bijection is:

PROPOSITION 64. *⊞- and ⊡-Popeye rectangle arrangements are both in bijection with Popeye sets. Any such set can be* uniquely *subdivided into* either a ⊞- *or a ⊡-PRA. The ⊡ variant of example* (5.3.4) *is:*



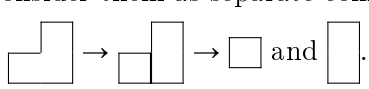
PROOF. The result is completely symmetric for ⊞- and ⊡-PRAs. We do not need to know which kind of arrangement we are making until step (4) of the proof, which makes it clear that either kind is possible.

It is enough to work with connected components of the arrangement, like



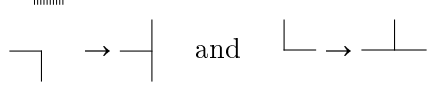
in example (5.3.4).

We will going to infer that certain lines *must* exist. We will draw these *forced* lines on the shape. The line subdivides the component into two, and from then on, we can consider them as separate components, e.g.



Here is how we break up the Popeye set's components:

(1) Every and forces a line:

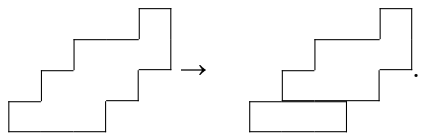


Here is the explanation in detail — the other is symmetric. In the original arrangement (be it ⊞ or ⊡), the other cases are impossible:

This cannot cannot occur in any rectangle arrangement.

This makes a rectangle on the right, whose cannot possibly satisfy the corner-sorting rule of Definition 60.

(2) Every has either a to its right or above it, or both. Indeed, if it has neither, the corner-sorting rule of Definition 60 could not be satisfied. If it has only one of the two, this forces a line in the corresponding direction:



(3) Now, every component has only the following kinds of corners: , , , , , , and both inner corners have s in the same row *and* column. (These are Popeye diagrams, see Subsection 5.3.10 or Chapter 8).

(4) To make a -LRA, look at the top . To avoid having two adjacent rectangles with the same right column (contradicting Definition 60), the line at that corner must be vertical. The right component is a rectangle, the left one satisfies the properties from (3), we proceed inductively.

Similarly, for -LRAs: look at the bottom , the line is forced to be horizontal, induct.

□

LERAs contain Popeye-rectangle arrangements, subject to Corollary 61. A Popeye set occupies the same cells as the equivalent PRA, so LERA rules induce analogous placement rules:

Corollary 65. *A Popeye set, obtained from a Popeye-rectangle arrangement overlaid on a LERA, obeys:*

(1) *All its s, s, and s are inside the shape. Moreover, none of these corners' upper or left edges may face the outside of the shape.*

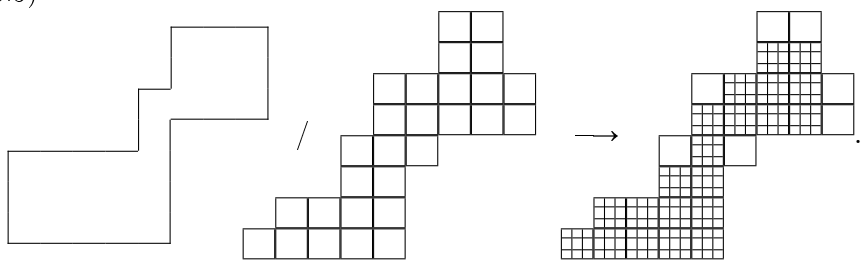
(2) *The cell above/to the left of any (the elbow cell) must be in the shape.*

(3) *To every C of the Popeye set corresponds a of the* shape*, which either coincides with C, or is to the right and below it* and *to the left of any other s right of C.*

5.3.13. Restricting a Popeye set to a shape. We will describe line-Popeye arrangements by restating Lemma 62 in terms of Popeye sets. Given a LERA, one can directly read off its Popeye rectangle arrangement (the restriction to the shape may remove some upper-left cells of the rectangle, but the top/bottom/left/right boundaries are unaffected). In contrast, in an LPA, we know only the set of rectangle cells — the result of restricting a Popeye set to a shape:

DEFINITION 66. Let P be a Popeye set placed on a shape S according to Corollary 65. Its restriction to S, denoted P/S, consists of all cells of P that fall in S.

(5.3.5)



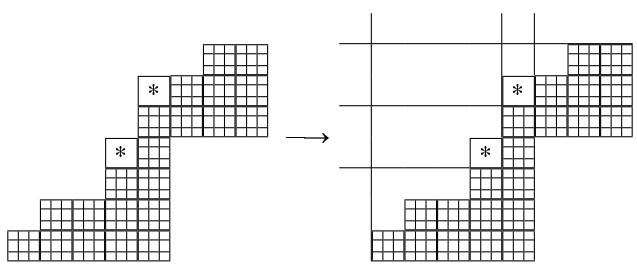
Surprisingly, no information is lost, which will let us discuss the Popeye set of an LPA. Here is how to recover the original set P from such a restriction. First, a simple application of Corollary 65 will identify P's s (elbow cells):

COROLLARY 67. *Any cell of the shape, which is just above and to the left of a of P/S, is an elbow cell.*

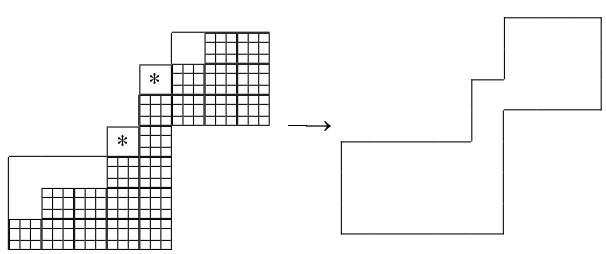
Now, we can reconstruct the original set:

ALGORITHM 68. *Take P/S as above, and find its elbow cells. To compute P:*

From every in P/S, draw a ray going up. From every , draw a ray heading left. From the bottom-right corner of each elbow cell, draw both upward and leftward rays. Example (5.3.5) *becomes:*



Next, turn each ray into a segment with an endpoint where it first *intersects a perpendicular ray. These segments form -corners, which become the s of the Popeye set P, made by adding the cells enclosed by the segments to P/S:*



5.3.14. Defining line-Popeye arrangements. In the preceding subsections, we collected all the pieces of the LERA $\leftrightarrow$ LPA bijection. We will define LPAs to be the image of all LERAs under the forward map:

ALGORITHM 69. *LERA (as in Lemma 62) $\to$ LPA (as in Lemma 70)*

(1) *Convert the LERA's Popeye rectangle arrangement into a Popeye set (Definition 63).*
(2) *Restrict the Popeye set to the shape (Definition 66).*

We can also rewrite Lemma 62 in terms of Popeye sets, giving an explicit description of LPAs:

LEMMA 70. *A* $\boxplus$*-LPA is a* non-overlapping *arrangement of:*

(1) *Lines as in Lemma 34,*
(2) *A Popeye set restricted to the shape according to Corollary 65,*
(3) *Elbow cells, as described in Corollary 56.*

Neighbors of the Popeye set's ▥*s and* ▥*s obey the restrictions from Lemma 62.*

The moment of truth: we can take an LPA, and reconstruct a $\boxplus$- *or* a $\boxplus$-LERA:

ALGORITHM 71. *LPA* $\to$ $\boxplus$- or $\boxplus$-*LERA*

(1) *Recover the Popeye set from the rectangle cells of the LPA (Algorithm 68).*
(2) *Map the Popeye set to a* $\boxplus$*- or a* $\boxplus$*-Popeye rectangle arrangement (Proposition 64).*

5.3.15. The bijection between $\boxplus$- and $\boxplus$-pafs. That's it, the bijection chain is complete, and we have a correspondence between the two kinds of pattern-avoiding fillings. We summarize the algorithm:

ALGORITHM 72. *paf $\to$ LPA*

(1) *Identify the rectangles (Algorithm 37).*
(2) *Identify the lines (Definition 33) in the non-rectangle cells, get a LRA.*
(3) *Identify the extended rectangles (Definition 43).*
(4) *Mark the elbow cells (Definition 54), get a LERA.*
(5) *Merge extended rectangles into a set of cells (Definition 63), get an LPA.*

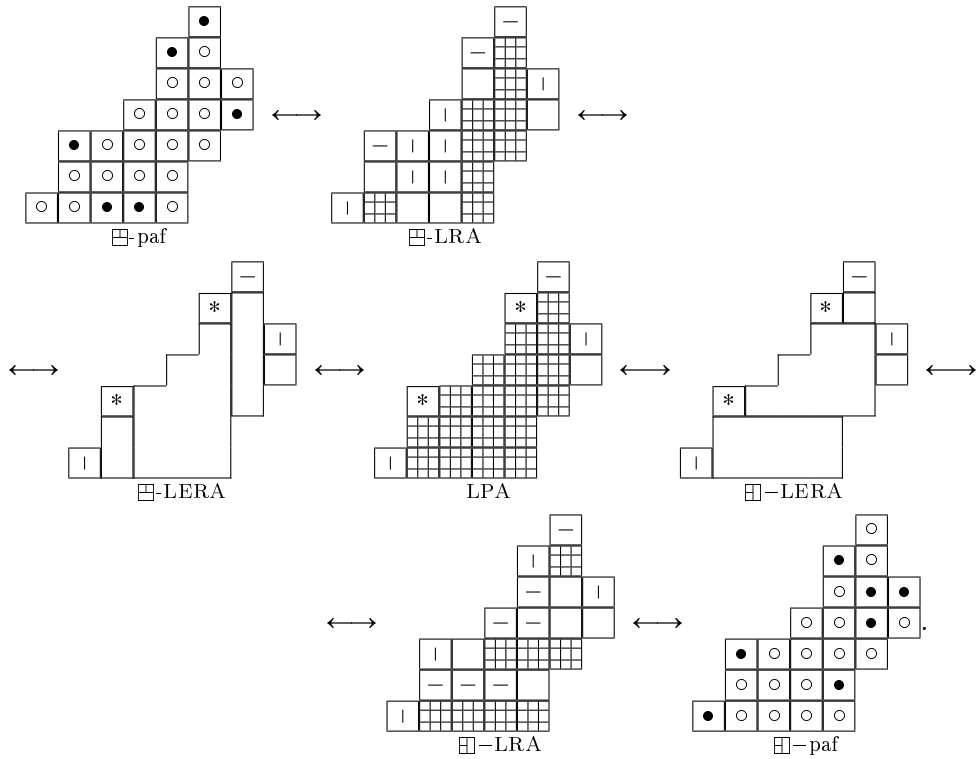
ALGORITHM 73. *LPA* $\to$ $\boxplus$- *or* $\boxplus$- *paf*

(1) *Recover the Popeye set from the rectangle cells (Algorithm 68).*
(2) *Map the Popeye set to a* $\boxplus$*- or* $\boxplus$*- Popeye-rectangle arrangement (Proposition 64), get a LERA.*
(3) *Fill in the elbow cells as specified in Lemma 55.*
(4) *Extract the lines and plain rectangles from the extended rectangles (Algorithm 44), get a LRA.*

(5) *Fill in the lines (Definition 33).*
(6) *Fill in the rectangles (Lemma 38), overwriting ∘s of some maximal lines.*

THEOREM 74. *Algorithms 72 and 73 give a bijection between ⊞-pafs, ⊞-pafs, and line-Popeye arrangements.*

For example,



CHAPTER 6

A Bijection between $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$

6.1. Preliminaries

This section will focus on convex shapes (of course, by Remark 12, convexity is only required of the shape's 2×2-connected components).

Given a $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$, we would like to find instances of $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ and replace each with a $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right)$, until there are no more left. Any such procedure terminates, because each move takes some $\bullet$ and moves it up and to the right by at least one — the shape is finite, so eventually, none of the $\bullet$s would have any space to move. The resulting shape, of course, avoids $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right)$, though it need not avoid $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$:

$$\begin{array}{|c|c|}\hline \bullet & \circ\\ \hline \bullet & \circ\\ \hline \circ & \circ\\ \hline\end{array} \xrightarrow[(1,1)\times(3,2)]{\text{swap}} \begin{array}{|c|c|}\hline \circ & \circ\\ \hline \bullet & \circ\\ \hline \circ & \bullet\\ \hline\end{array}.$$

The outcome also depends on the order, in which we do swaps — the shape above could also evolve like this:

$$\begin{array}{|c|c|}\hline \bullet & \circ\\ \hline \bullet & \circ\\ \hline \circ & \circ\\ \hline\end{array} \xrightarrow[(2,1)\times(3,2)]{\text{swap}} \begin{array}{|c|c|}\hline \bullet & \circ\\ \hline \circ & \circ\\ \hline \circ & \bullet\\ \hline\end{array} \xrightarrow[(1,1)\times(2,2)]{\text{swap}} \begin{array}{|c|c|}\hline \circ & \circ\\ \hline \circ & \bullet\\ \hline \circ & \bullet\\ \hline\end{array}.$$

In order to go from this idea to a bijection, we will have to avoid creating $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$s, to produce a well-defined output, and to make sure the map is invertible. It turns out that we can achieve all of those goals by picking a "good" swap order.

Any bijection built in this way will be quite nice:

Lemma 75. *Any* $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ *bijection based on making (intelligently chosen) swaps until no more can be made will:*

(1) *Preserve the number of* $\circ$*s and* $\bullet$*s in the filling.*
(2) *Leave intact all fillings that avoid* all *of:* $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right)$, $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right)$, *and* $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$.

There are several other desiderata:

Map symmetry: $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ pafs are different sets, so we cannot get an involution, but they are related by symmetry, so we can ask for the next best thing: if $F(f)$ is the forward map of a filling, and $B(f)$ is the backward map, and $R(f)$ is the entire filling reflected across / , then:

$$B(f) = R(F(R(f)).$$

Pattern symmetry: Reflection across $\backslash$ R' preserves both pattern pairs. So, if the filling's shape is symmetric under R', we would like

$$F(f) = R'(F(R'(f)).$$

Shape symmetry: One pattern pair can be transformed into the other by a reflection across /. If our shape is preserved by this reflection, it is desirable that the bijection naturally flips the filling.

Locality: Take a $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$- or $\left(\begin{smallmatrix}\circ&\circ\\\circ&\bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$-paf, and add to it some more filled cells (row/column insertions permitted) so that the result is the same kind of paf. Suppose that the new cells are such that, regardless of the content of the old, they cannot participate in a forbidden pattern (any of $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\right)$, $\left(\begin{smallmatrix}\circ&\circ\\\circ&\bullet\end{smallmatrix}\right)$, $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$) — for example, a row of $\bullet$s. Then, it would be desirable if the bijection, restricted to the old shape, were unchanged.

Shape symmetry and locality are incompatible because, for example, this addition makes the shape symmetric:

$$\begin{array}{|c|c|}\hline \bullet&\circ\\\hline \bullet&\circ\\\hline \circ&\circ\\\hline\end{array}\ \xrightarrow{\text{add column of }\bullet\text{s}}\ \begin{array}{|c|c|c|}\hline \bullet&\circ&\bullet\\\hline \bullet&\circ&\bullet\\\hline \circ&\circ&\bullet\\\hline\end{array},$$

so a local bijection would leave the new column intact, whereas a reflection would make it into a row. Our algorithm will be local, with map symmetry and pattern symmetry.

6.1.1. Empirical tightness. This result seems tight: on our test shapes, the equivalence held iff the shape broke up into convex 2×2-connected components, or was preserved by /-reflections.

6.2. Bijection

It turns out that we do not need any specific order of operations. The key is to act on a specific occurrences of patterns:

Definition 76. An instance of a forbidden pattern is called *minimal* if its entire rectangle (which is in the shape by convexity), not counting the 4 corners, is filled with $\bullet$s. E.g.:

$$\begin{array}{|c|c|c|c|c|c|}\hline \bullet&\bullet&\bullet&\bullet&\bullet&\circ\\\hline \bullet&\bullet&\bullet&\bullet&\bullet&\bullet\\\hline \circ&\bullet&\bullet&\bullet&\bullet&\circ\\\hline\end{array}\qquad \begin{array}{|c|c|}\hline \circ&\circ\\\hline \bullet&\bullet\\\hline \bullet&\bullet\\\hline \circ&\bullet\\\hline\end{array}\qquad \begin{array}{|c|c|c|}\hline \bullet&\bullet&\circ\\\hline \circ&\bullet&\bullet\\\hline\end{array}\ .$$

a minimal $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\right)$ a minimal $\left(\begin{smallmatrix}\circ&\circ\\\circ&\bullet\end{smallmatrix}\right)$ a minimal $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$

Algorithm 77.

→ *Start with a* $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$*-paf, and swap minimal instances of* $\left(\begin{smallmatrix}\circ&\circ\\\circ&\bullet\end{smallmatrix}\right)$ *for* $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\right)$ *in any order, until no more can be found.*

← *Start with a* $\left(\begin{smallmatrix}\circ&\circ\\\circ&\bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$*-paf, and swap minimal instances of* $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\right)$ *for* $\left(\begin{smallmatrix}\circ&\circ\\\circ&\bullet\end{smallmatrix}\right)$ *in any order, until no more can be found.*

To see that this is the desired bijection, we will need these results:

LEMMA 78. *At every step of Algorithm 77,*

(1) *The filling contains no* minimal *instance of* $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$.
(2) *If the filling contains an instance of* $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right)$ *or* $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ *it contains a corresponding minimal instance.*

LEMMA 79. *The outcome of Algorithm 77 does not depend on which minimal instances are swapped, or on the swap order.*

In fact, it will be easier to prove a stronger result in which the $\circ$s are labeled. Here is just the $\leftarrow$ version to simplify the writing:

PROPOSITION 80. *Given a* $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$*, assign a unique label to each of the* $\circ$*s. Apply Algorithm 77 to the paf, using the following swap method (labels afford several possibilities, but only some of them work):*

$$\begin{matrix}\bullet & a_\circ\\ c_\circ & b_\circ\end{matrix} \leftrightarrow \begin{matrix}a_\circ & b_\circ\\ c_\circ & \bullet\end{matrix}\,. \tag{6.2.1}$$

Then, whichever minimal $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right)$*s are swapped, in whatever order, the* labeled *result will always be the same.*

The proofs come later, but first, the prize:

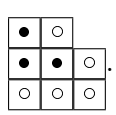
THEOREM 81. *Algorithm 77 is a bijection. By Section 6.1, it has the properties from Lemma 75, as well as locality, pattern symmetry, and map symmetry.*

PROOF. It is enough to check only $\rightarrow$. Consider some particular sequence of swaps performed by Algorithm 77. By Lemma 78, the end result is a $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$-paf. Performing these swaps in reverse will recover the original $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$-paf. So, this reverse sequence is a a *maximal* swap sequence — one of many possible executions of $\leftarrow$. By Lemma 79, all maximal swap sequences yield the same result, so this is indeed a bijection.

Map symmetry and locality are obvious, Lemma 75 needs no further proof. Pattern symmetry holds because the swap operation is symmetric under reflections across $\backslash$. $\square$

CONJECTURE 82. *A* $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ *with labeled* $\circ$*s generates a lattice, which has minimal* $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right) \rightarrow \left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ *swaps as the cover relations.*

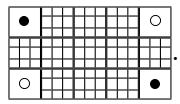
The lattice would not be modular: consider, e.g. all swap sequences in:



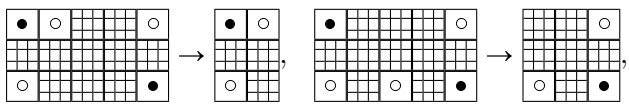
6.3. Proofs

Unfortunately, the proofs of these simple results are rather technical. There must be a cleaner and more insightful way to get them!

Proof of Lemma 78-(1). "No step of Algorithm 77 makes minimal $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$s." It is enough to treat the $\leftarrow$ direction, with the algorithm swapping $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\right)$s for $\left(\begin{smallmatrix}\circ&\circ\\\circ&\bullet\end{smallmatrix}\right)$s. Consider the *first* minimal $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$ instance, with ⊞ denoting "internal" $\bullet$s:

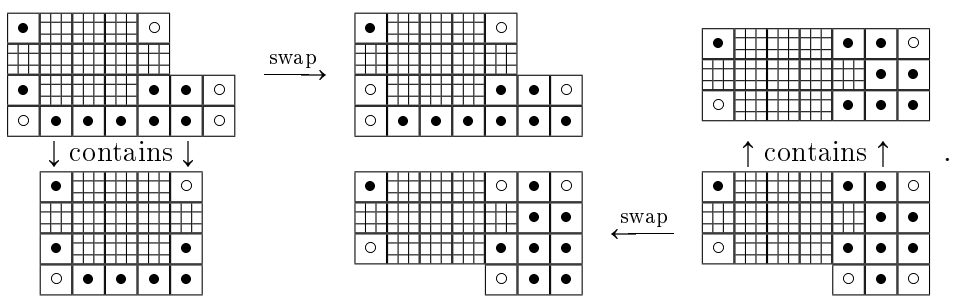


The swap that just occurred must have changed the filling to create this $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$. It could have changed just one cell above. Suppose that one of the ⊞ cells was formerly a $\circ$ — then, the pre-swap filling also contained a minimal $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$, a contradiction:



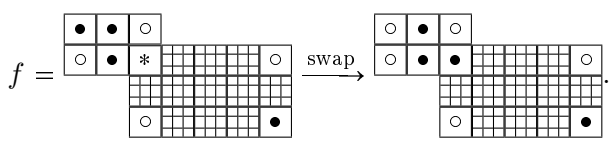
The bottom-right $\bullet$ could not have been $\circ$ — if it were, the swap that changed it to $\bullet$ would have also changed the top-left $\bullet$ to $\circ$.

Suppose that one of the $\circ$s was previously a $\bullet$, which implies one of the following pre-swap pictures:



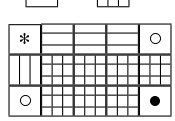
Again, both of them contain a minimal $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$.

Lastly, the top-left $\bullet$ might have been a $\circ$ (marked $*$) before the swap:



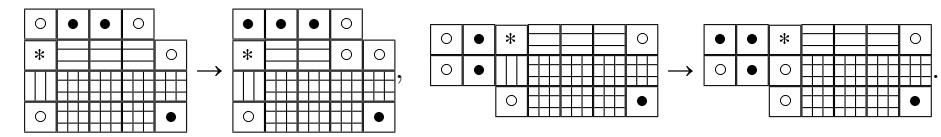
Note that if we perform another "backwards" swap on the minimal $\left(\begin{smallmatrix}\circ&\circ\\\circ&\bullet\end{smallmatrix}\right)$ in f, the upper-left $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\right)$ turns into a minimal $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$. To get back to the original shape, we do need to eliminate this $\left(\begin{smallmatrix}\circ&\circ\\\circ&\bullet\end{smallmatrix}\right)$ — but maybe there is a way to do it without introducing a minimal $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$? It will take some work to get a contradiction.

First, let us see what backwards swaps can do to the $\left(\begin{smallmatrix}\circ&\circ\\\circ&\bullet\end{smallmatrix}\right)$ — here it is, marked up in more detail (⊞ = ⊟ = ⊞ = $\bullet$):



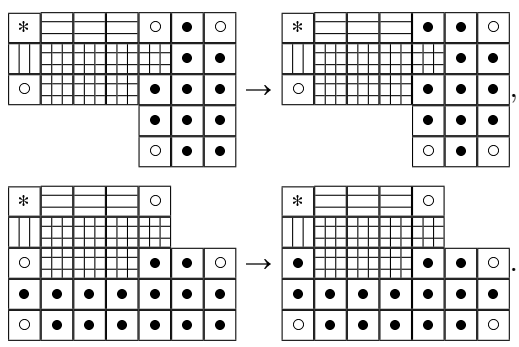
The only way to touch the $*$ is to swap this very $\begin{pmatrix}\circ & \circ\\ \circ & \bullet\end{pmatrix}$. That is also the only way to touch the bottom-right $\bullet$.

The ▦ cells cannot become $\circ$s: that requires a $\circ$ to the left of, and a $\circ$ above the target cell, and only $\bullet$s in the resulting rectangle — but, for every ▦ cell, $*$ would have to be inside any possible rectangle. Any of the ▤ and ▥ cells can become $\circ$s:



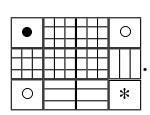
But, these swaps just shrink the minimal $\begin{pmatrix}\circ & \circ\\ \circ & \bullet\end{pmatrix}$, without changing f's structure.

This leaves only the two $\circ$s — there is just one way to replace either with $\bullet$:



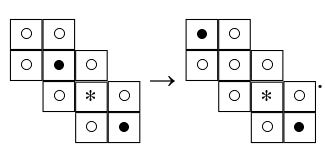
Either kind of swap simply grows the minimal $\begin{pmatrix}\circ & \circ\\ \circ & \bullet\end{pmatrix}$, without changing f's structure.

So, we are stuck with the minimal $\begin{pmatrix}\circ & \circ\\ \circ & \bullet\end{pmatrix}$ in the bottom part of f. Let us see if we can instead fix up the $\begin{pmatrix}\bullet & \circ\\ \circ & \circ\end{pmatrix}$ to prevent it from becoming a $\begin{pmatrix}\bullet & \circ\\ \circ & \bullet\end{pmatrix}$. Here it is, with more detailed markings (▦ = ▤ = ▥ = $\bullet$):



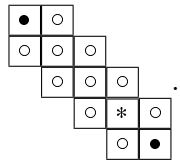
Clearly, $* \to \bullet$ does not work. Much as before, ▦ $\to \circ$ creates a minimal $\begin{pmatrix}\bullet & \circ\\ \circ & \bullet\end{pmatrix}$. Finally, ▤ $\to \circ$ or ▥ $\to \circ$ simply shrinks this minimal $\begin{pmatrix}\bullet & \circ\\ \circ & \circ\end{pmatrix}$ without changing the structure. That leaves the top-right $\bullet$, which can become $\circ$ if it is the bottom-right corner of *another* $\begin{pmatrix}\circ & \circ\\ \circ & \bullet\end{pmatrix}$ (internal $\bullet$s hidden to save space):

(6.3.1)

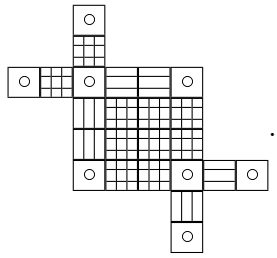


Going backwards from our first minimal $\begin{pmatrix}\bullet & \circ\\ \circ & \bullet\end{pmatrix}$, we ruled out every swap except the one that produced the $\begin{pmatrix}\bullet & \circ\\ \circ & \circ\end{pmatrix}$-$\begin{pmatrix}\circ & \circ\\ \circ & \bullet\end{pmatrix}$ structure. Going back further, we ruled

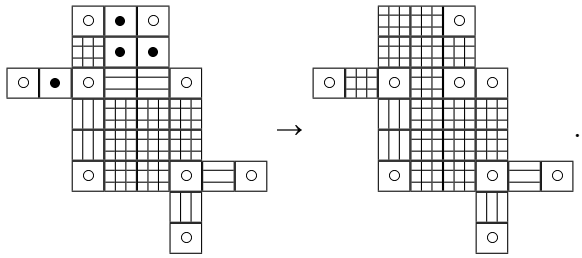
out every swap but (6.3.1). Obviously, we can repeat this, and move the $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right)$ further away from the $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right)$, building a *tower*:



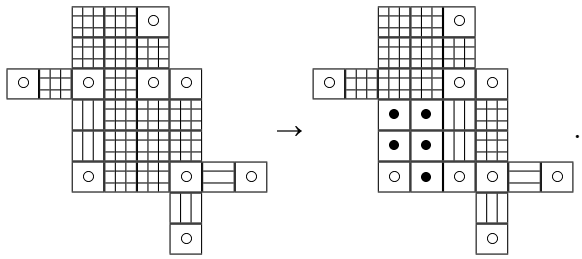
The previous results still hold in this structure. Briefly: swaps affecting the top-left $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right)$ can shrink the pattern, or add another level to the tower. Similarly, swaps touching the bottom-right $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ can shrink or grow the pattern, or move the $*$ up, which deletes a level. It remains to see how swaps touching the intermediate levels affect the tower structure — here is an internal block, marked up in detail (= = = $\bullet$):



The block's top-left $\circ$ and the cells cannot be changed by a swap. If the bottom-right $\circ$ becomes $\bullet$, this becomes the lowest level of the tower. A bottom-left or top-right $\circ$ can become a $\bullet$, growing the current block, and simultaneously inserting a $\circ$ in a or cell of the block below, e.g.



We will keep track of these inserted $\circ$s as part of the tower structure — it is possible for several to accumulate in a row or column. Conversely, swaps inside this expanded tower structure can eliminate them, shifting block boundaries:



In the above examples, markup with , , serves to identify cells that belong to the tower, and also to distinguish whether or not a swap can make them $\circ$.

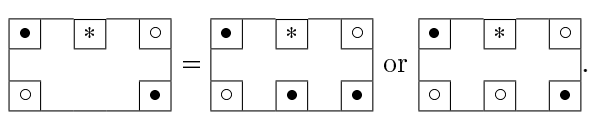
The point of this discussion is that with a slightly expanded definition of a tower (the reader may wish to formalize it), all possible swaps preserve its structure. Eventually, all possible swaps have to be made — including the potential swaps within the tower. As noted above, those either eliminate the "inserted" $\circ$s, or reduce the number of levels in the tower. Eventually, the tower returns to a $\begin{pmatrix}\bullet & \circ\\ \circ & \circ\end{pmatrix}$-$\begin{pmatrix}\circ & \circ\\ \circ & \bullet\end{pmatrix}$ pair, and the only possible swap creates a $\begin{pmatrix}\bullet & \circ\\ \circ & \bullet\end{pmatrix}$.

To sum up: we imagined that Algorithm 77 made some "forward" $\begin{pmatrix}\bullet & \circ\\ \circ & \circ\end{pmatrix} \to \begin{pmatrix}\circ & \circ\\ \circ & \bullet\end{pmatrix}$ swaps to create a *first* minimal $\begin{pmatrix}\bullet & \circ\\ \circ & \bullet\end{pmatrix}$. We worked back from that assumption, making $\begin{pmatrix}\circ & \circ\\ \circ & \bullet\end{pmatrix} \to \begin{pmatrix}\bullet & \circ\\ \circ & \circ\end{pmatrix}$ swaps, and taking care not to make any more minimal $\begin{pmatrix}\bullet & \circ\\ \circ & \bullet\end{pmatrix}$s. This led us to observe a "tower" structure, which contains the pre-history of the first $\begin{pmatrix}\bullet & \circ\\ \circ & \bullet\end{pmatrix}$. We observed that the structure was preserved under all possible swaps, and, at the very beginning, when no $\begin{pmatrix}\circ & \circ\\ \circ & \bullet\end{pmatrix}$s existed, had to have been a minimal $\begin{pmatrix}\bullet & \circ\\ \circ & \bullet\end{pmatrix}$ — a contradiction. $\square$

Though its proof was involved, part (1) is a real workhorse — we will need it for everything that follows. The next lemma is used both in part (2) and in Proposition 80.

LEMMA 83. *The rectangle of an instance of* $\begin{pmatrix}\bullet & \circ\\ \circ & \bullet\end{pmatrix}$ *contains* both $\begin{pmatrix}\bullet & \circ\\ \circ & \circ\end{pmatrix}$ *and* $\begin{pmatrix}\circ & \circ\\ \circ & \bullet\end{pmatrix}$.

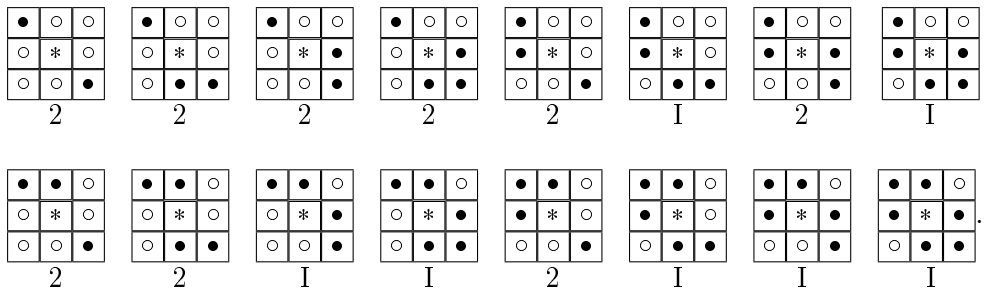
PROOF. No instance of $\begin{pmatrix}\bullet & \circ\\ \circ & \bullet\end{pmatrix}$ is minimal, so there is always some $\circ$ inside (possibly on the boundary). We will label the special $\circ$ as $*$. Suppose that it is on the boundary of the shape:



If the lowest cell below $*$ contains a $\circ$, the requested $\begin{pmatrix}\bullet & \circ\\ \circ & \circ\end{pmatrix}$ and $\begin{pmatrix}\circ & \circ\\ \circ & \bullet\end{pmatrix}$ are in evidence. Otherwise, look at the rectangle defined by $*$ and the bottom-left $\circ$. It is a smaller $\begin{pmatrix}\bullet & \circ\\ \circ & \bullet\end{pmatrix}$, which can never be minimal, so we win by induction.

If $*$ is on the interior of the shape, there are 16 possibilities for the boundary cells above, below, to the left, and to the right of it. We mark each with "2" for "both $\begin{pmatrix}\bullet & \circ\\ \circ & \circ\end{pmatrix}$ and $\begin{pmatrix}\circ & \circ\\ \circ & \bullet\end{pmatrix}$ are visible" or "I" for "induction on smaller pattern $\begin{pmatrix}\bullet & \circ\\ \circ & \bullet\end{pmatrix}$". In these compact drawings, we omit the non-border,

non-$*$ rows and columns:

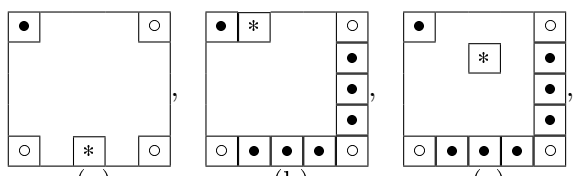


□

The easier part (2) shows that Algorithm 77 does in fact produce pafs:

Proof of Lemma 78-(2). "During Algorithm 77 every $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\right)$ or $\left(\begin{smallmatrix}\circ&\circ\\\circ&\bullet\end{smallmatrix}\right)$ contains a matching minimal instance."

We will try to make a $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\right)$ that does not contain a smaller instance of $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\right)$ inside. We will progress through some cases, checking to see if a $\circ$ can be placed in a particular spot. This "test" $\circ$ will be shown as $*$ to distinguish it.



(a) (b) (c)

In (a), we try to put the $*$ in the bottom row, which creates a smaller instance of $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\right)$. A $\circ$ in the rightmost column would have the same effect. So, from (b) onwards those cells contain $\bullet$s . In (b) and (c), we try to put the $*$ in any other cell of the shape. In all cases, we get an instance of $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$, which gives an instance of $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\right)$ by Lemma 83. So, our $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\right)$'s rectangle contains only $\bullet$ s, except for the three $\circ$ corners.

The same assertion for $\left(\begin{smallmatrix}\circ&\circ\\\circ&\bullet\end{smallmatrix}\right)$ follows by symmetry. □

Proof of Proposition 80. It is enough to prove the claim for the $\leftarrow$ direction of Algorithm 77, swapping $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\right)$s for $\left(\begin{smallmatrix}\circ&\circ\\\circ&\bullet\end{smallmatrix}\right)$s.

We will start with a fairly technical discussion, which describes an "Area of Interest" for every $\circ$ cell (Algorithm 85), and shows that swaps preserve all of its key properties (Definition 87). Assuming that there are two swap sequences with different outcomes, we will take a labeled $\circ$ from a particular location in which they differ, and use the invariants to prove that the $\circ$ with this label *must* end up in the same place.

By definition of the swap in (6.2.1):

$$\begin{matrix}\bullet & a_\circ\\ c_\circ & b_\circ\end{matrix} \leftrightarrow \begin{matrix}a_\circ & b_\circ\\ c_\circ & \bullet\end{matrix},$$

a $\circ$ can only move left, or up.

LEMMA 84. *If a* labeled $\circ$ *is leftmost in its row, then it stays leftmost in the same row. Likewise, the lowest* $\circ$ *in a column remains lowest in the same column.*

PROOF. It is enough to prove the first claim. Given a leftmost $\circ$:

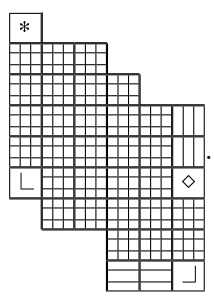
(1) If it is the upper-right/lower-left corner of a $\begin{pmatrix}\bullet & \circ\\ \circ & \circ\end{pmatrix}$, or the upper-left/lower-left corner of a $\begin{pmatrix}\circ & \circ\\ \circ & \bullet\end{pmatrix}$ during a swap, it remains leftmost *and* stays in the same row.
(2) It cannot be the lower-right corner of a $\begin{pmatrix}\bullet & \circ\\ \circ & \circ\end{pmatrix}$, nor the upper-right corner of a $\begin{pmatrix}\circ & \circ\\ \circ & \bullet\end{pmatrix}$.
(3) A $\circ$ moving up or down cannot land to the left of the leftmost $\circ$, because cells moving vertically always land to the left of another.

$\square$

Every $\circ$ in the shape has a rectangular *"Area of Interest"* (further, AoI) — an almost-rectangular region with an almost-specified filling.

ALGORITHM 85. *Given a* $\circ$*, further marked* $\diamond$*, we can compute its AoI thus:*

(1) *Find the first* $\circ$ *to the left of* $\diamond$ *and label it* $\llcorner$*. If* $\diamond$ *is the leftmost* $\circ$ *, imagine a virtual* $\llcorner$ *just to the left of its row.*
(2) *Find the first* $\circ$ *above the* $\llcorner$ *and label it* $*$*. If* $\llcorner$ *is virtual, imagine a virtual* $*$ *just above* $\llcorner$ *(that cell may be in the shape, but that is unimportant). If* $\llcorner$ *is the top* $\circ$*, create a virtual* $*$ *just above its column.*
(3) *Find the first* $\circ$ *below* $\diamond$ *and label it* $\lrcorner$*. If* $\diamond$ *is the bottom* $\circ$*, imagine a virtual* $\lrcorner$ *just below its column.*
(4) *The AoI is the rectangle with top-left corner* $*$ *and bottom-right corner* $\lrcorner$*. Then* $\diamond$ *is in its right column, and* $\llcorner$ *is in its left column. Also, we exclude the cells to the right of* $*$*, and those below* $\llcorner$*. Some top-right and bottom-left cells may be outside the shape. E.g.,*

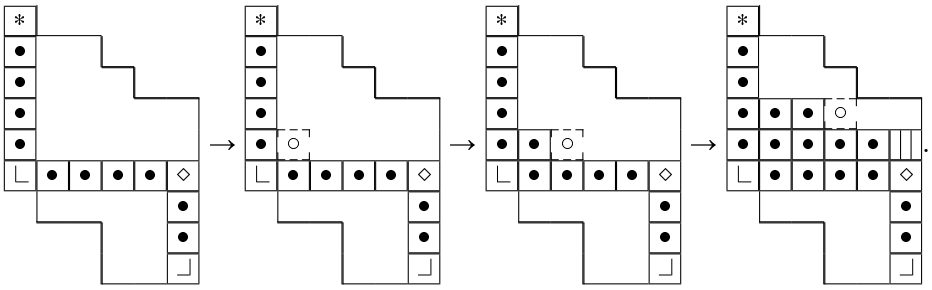
(6.3.2) 

.

The reader should verify that the picture makes sense with virtual $\llcorner$*,* $*$*, or* $\lrcorner$*.*

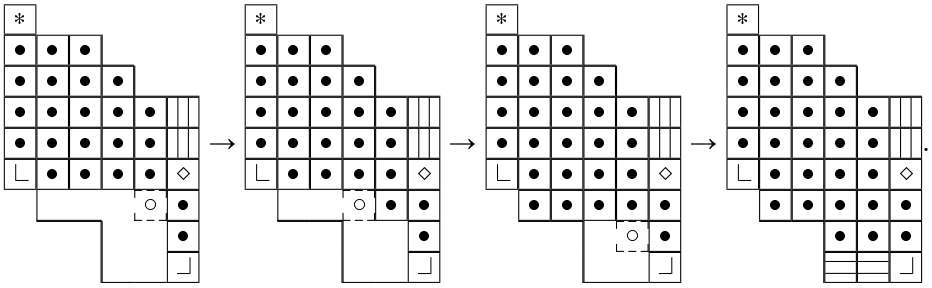
LEMMA 86. *At any step of Algorithm 77, consider the AoI of* $\diamond$ *and mark it as in* (6.3.2)*: cells above* $\diamond$ *with* ▥*, those to the left of* $\lrcorner$ *with* ▤*, and all remaining unmarked cells with* ▦*. Then, the* ▦ *cells* must *contain only* $\bullet$*s, while the* ▤ *and* ▥ *cells may contain either.*

PROOF. Cells between ∟ and ⋄, between ⌟ and ⋄, ∗ and ∟ are all •s by definition (the slightly thicker lines show the edge of the shape, not just of the AoI).



We start from the cell just above and to the right of ∟, and walk increasing rows, left-to-right. We try to place a ∘ in each cell, see that this would create a minimal $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$ involving ∟, and fill the cell with •. On the right edge, the $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\bullet\end{smallmatrix}\right)$ is prevented by the presence of ⋄ below, so we mark those cells ▥ to emphasize the ambiguity.

We then apply the same method to the lower part of the AoI:



□

We have described the AoI completely. $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\right) \to \left(\begin{smallmatrix}\circ&\circ\\\circ&\bullet\end{smallmatrix}\right)$ swaps can change the coordinates of all the special cells, and most of their identities. However, we will see that much about the AoI is not changed by swaps:

DEFINITION 87. The *AoI invariants* are:

(1) The description of the AoI's shape and filling given in Algorithm 85 and Lemma 86.
(2) The label of the ∗ (requires proof), and the label of the ⋄ (by construction).

Swaps, which affect the AoI, come in two varieties:

DEFINITION 88. Given an AoI, an *internal* swap replaces a minimal $\left(\begin{smallmatrix}\bullet&\circ\\\circ&\circ\end{smallmatrix}\right)$, which is entirely contained in the AoI. Any other swap is *external*.

LEMMA 89. *In an AoI at any step of Algorithm 77, an external swap cannot:*

(1) *Move ⋄ (internal swap required).*
(2) *Change ⌟ to a • (internal swap required).*

(3) *Change any cells that are not in $\diamond$'s column, and which Lemma 86 marked with* $\boxplus$.

Similarly, an internal swap cannot affect $*$ or $\llcorner$.

Proof. Moving $\diamond$ means having it as (i) a bottom-right corner — but then $\llcorner$ is the bottom-left, and some $\boxed{|||}$ is the top-right corner, or (ii) as a top-right corner — meaning that $\lrcorner$ is the bottom-right, and some $\boxed{\equiv}$ is the bottom-left. Both are internal swaps.

Similarly, to replace the current $\lrcorner$ with $\bullet$, it has to be the bottom-right corner, while $\diamond$ is the top-right, and some $\boxed{\equiv}$ is the bottom-left. That's an internal swap from the previous paragraph.

Changing a non-rightmost $\bullet$-filled cell to a $\circ$ requires that it be a swap's top-left corner. Then, the top-right corner is a $\diamond$, some $\boxed{|||}$, or a cell even farther right. Similarly, the bottom-left corner is some $\boxed{\equiv}$, or a cell farther down. If both top-right and bottom-left are in the cell, then bottom-right is $\lrcorner$, and the swap is internal. Otherwise, such a swap cannot be minimal, because $\lrcorner$ lands in its interior. $\square$

It is not hard to see that external swaps can change $\boxed{\equiv}$ and $\boxed{|||}$ cells both to $\circ$s and to $\bullet$s — however, such changes do not change the AoI's invariants. Now, we tackle the remaining cells:

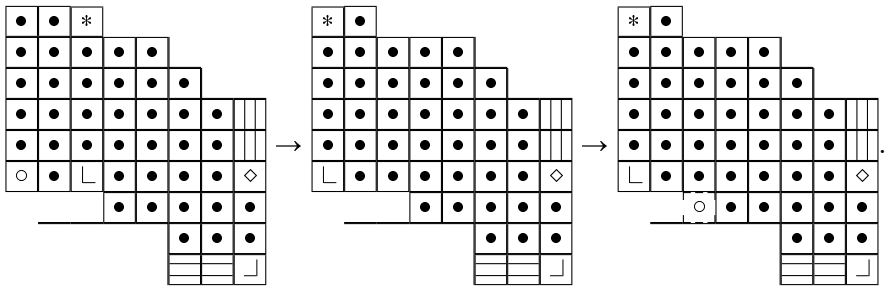
Lemma 90. *Consider an AoI at some step of Algorithm 77. Below, we describe how external swaps affect the special cells $\lrcorner$, $\llcorner$, and $*$. They all preserve the AoI's invariants.*

Proof. Here are all ways in which external swaps can affect $\lrcorner$, $\llcorner$, and $*$:

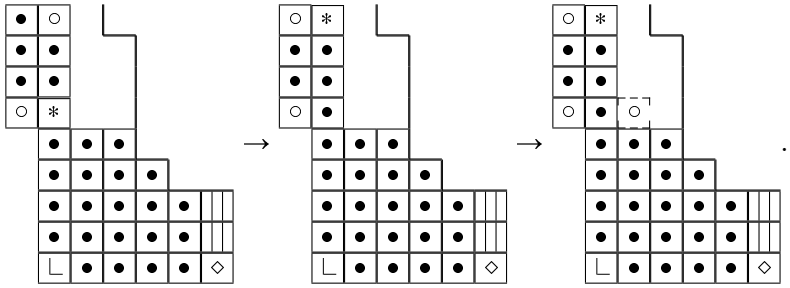
(1) A cell between $\lrcorner$ and $\diamond$ can change to a $\circ$ (via a swap of a $\begin{pmatrix}\bullet & \circ\\ \circ & \circ\end{pmatrix}$ that has $\lrcorner$ as the bottom-right corner). The cell becomes the new $\lrcorner$ (so, $\lrcorner$'s label is not constant), shortening the AoI — but its structure is unchanged.

(2) $\llcorner$ can become a $\bullet$ only via a swap that has it as the bottom-right corner, and $*$ as the top-right corner. Then, there is a $\circ$ to its left, which becomes the new $\llcorner$. The $*$ moves to the same column — so, $*$'s label is unchanged. As a consequence, the AoI is widened. The newly added cells are all $\bullet$s because they were part of a minimal instance of a $\begin{pmatrix}\bullet & \circ\\ \circ & \circ\end{pmatrix}$. In the widened AoI, we also need to check that the part *below* the $\llcorner$-$\diamond$ line is extended with $\bullet$s only. The argument of Lemma 86 works: try to put a $\circ$, find a minimal $\begin{pmatrix}\bullet & \circ\\ \circ & \bullet\end{pmatrix}$, conclude that the cell must have a $\bullet$, move on to the next cell. Thus, the AoI structure is preserved. Here is the swap, and of the first minimal

$\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$, which involves $\diamond$ — the thick line marks the shape's boundary:



(3) $*$ can become a $\bullet$ only via a swap that has it as the bottom-right corner, which means that the $*$ moves up, while its cell becomes filled with $\bullet$. The result still has AoI structure — we only need to check that the newly added cells are filled with $\bullet$s. Just as with the $\llcorner$-moving swap, the minimal $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ argument of Lemma 86 works. Here is the upper part of an AoI, illustrating the swap and the first minimal $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$, which involves $\llcorner$ (the line shows the shape's boundary):



Note that $*$ does keep the original label in this AoI.

$\square$

In Lemmas 89 and 90, we covered the effect of external swaps on ▦ cells in $\diamond$'s column, the remaining ▦ cells, discussed the special cells $\diamond$, $\lrcorner$, $\llcorner$, and $*$. We discounted ▤ and ▥ cells since their fillings are not mandated. Those are *all* the ways in which an external swap can change an AoI.

Corollary 91. *External swaps preserve AoI invariants.*

Lemma 92. *Internal swaps preserve AoI invariants.*

Proof. The following internal swaps are possible:

(1) $\lrcorner$ can change to a $\bullet$ if it is the bottom-right corner of a $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right)$. The swap has $\diamond$ as the top-right corner, and the rightmost ▤ cell containing $\circ$ as the bottom-left corner. The bottom-left corner *must* be to the left of $*$'s column — otherwise the swap could not be minimal because of $\llcorner$.

The swap moves $\diamond$ to that $\circ$'s column. The $\circ$ becomes the new $\lrcorner$. The rectangle narrows, but the AoI invariants are preserved.

(2) $\diamond$ can move up via a swap that has $\llcorner$ as the bottom-left corner, and the lowest $\boxed{|||}$ cell containing $\circ$ as the top-right corner. The top-right corner has to be below $*$'s row — otherwise the swap would not be minimal.

In addition to moving $\diamond$ up, the swap also moves the $\boxed{|||}$ $\circ$-cell into $*$'s column, making it the new $\llcorner$. The rectangle gets shorter, but the AoI invariants are preserved.

There are no other internal swaps. Indeed, they have to have a $\circ$-filled bottom-right cell. The candidates are $*$,$\lrcorner$, some $\boxed{|||}$ cells, some $\boxed{\equiv}$ cells, $\llcorner$ (covered), $\diamond$ (covered). Any swap using $*$, $\lrcorner$, and $\boxed{\equiv}$ would lack a bottom-left cell, while those with $\boxed{\equiv}$ would lack a top-right cell.

$\square$

Corollary 93. *All* $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right) \to \left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ *swaps preserve the AoI invariants from Definition 87.*

Remark 94. In discussing AoI invariants, we did not address the possibility that some of its special cells are "virtual". Nonetheless, all the claims and arguments are valid, because in an AoI with virtual cells, some of the swaps are simply unavailable — and cannot change the invariants. Some comments are needed:

(1) If $*$ is virtual, "keeping its label" translates into "it remains virtual". Indeed, without $*$, $\llcorner$ cannot be replaced by $\bullet$, which means that the AoI's right edge stays in the same column, and no $\circ$ can appear above $\llcorner$.
(2) Virtual $\lrcorner$: $\diamond$ is the lowest $\circ$ in the column, so by Lemma 84 $\lrcorner$ remains forever virtual.
(3) Virtual $\llcorner$: $\diamond$ is leftmost in the row, and by Lemma 84 it stays so.

In short, a virtual cell is forever virtual. A non-virtual cell likewise remains forever non-virtual. The careful reader is invited to re-read the arguments while keeping virtual cells in mind.

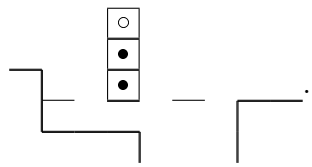
For the next few paragraphs, we will assume that the special cells are not virtual.

We will pick now a very particular $\diamond$ for further analysis. Suppose that, in violation of this proposition, there exists a $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$-paf with labeled $\circ$s, which is mapped to two different fillings a and b by different Algorithm 77 $\leftarrow$ swap sequences. Find $a_{i,j} \neq b_{i,j}$, first minimizing j, then maximizing i (choose the leftmost column with a difference, in it pick the lowest difference). We may assume that $a_{i,j}$ is a $\circ$. Since the it is labeled, we will examine this particular $\circ$ in more detail — and refer to it as $\diamond$ from here on.

By assumption, a and b agree in every cell to the left of column j. But, $\diamond$ must have different locations — and in b, it cannot be to the left of j. In both fillings, $\diamond$ has the same $*$ (that is an AoI invariant), and in the same location, since it is to the left of j in a. Then, both a's and b's $\llcorner$ s are to the left of j, which means they *must* coincide — if not, one $\llcorner$ is above another,

but all cells between $\llcorner$ and $*$ must be filled with $\bullet$s. Hence, $\diamond$ is in the same row in both a and b. Then, $\diamond$ in b is at i, k with column $k > j$.

CLAIM 95. Consider a $\circ$ in a filling at some step of Algorithm 77 $\leftarrow$, and a horizontal line (dashed) below it (thick lines show the shape's boundary):



Suppose also that, in this column, this is the lowest $\circ$ above the dashed line. If, after some sequence of "backwards" $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right) \to \left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right)$ swaps, $\circ$ is above the line, then it remains lowest above the line in its column.

PROOF. If our $\circ$ is the top-right corner of a swapped $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right)$, it either moves below the line, or remains lowest above the line. If it is the top-left corner, then the bottom-left corner *must* be below the line, and our $\circ$ moves left, but remains lowest above the line. Finally, a $\bullet$ between $\circ$ and the line can only be the bottom-right corner of a swapped $\left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ if our $\circ$ is the top-right corner — so, no other $\circ$ can be inserted below it. $\square$

By Lemma 84, since $\diamond$ changed columns, it is not lowest in its column. In a, $\diamond$ was the lowest point where a and b disagree — thus, they agree on a's $\diamond$'s $\lrcorner$ in row l (though in b it is not a *special* $\circ$). Furthermore, b must have a $\bullet$ at i, j because i, j is between $\diamond$ and $\llcorner$. Looking at rows i and l, columns j and k, we have:

If $\ulcorner\!\lrcorner$ is in the shape, this is either a $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right)$ or a $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$, which contains $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right)$ by Lemma 83. The presence of $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\right)$ proves that Algorithm 77 $\leftarrow$ could not have ended in b. So, $\ulcorner\!\lrcorner$ must be outside the shape. We know that $\diamond$ must have started in column k or to the right, where the bottom of the shape is no lower than in column k. So, apply Claim 95 to $\diamond$ in a with the line passing through the very bottom of column k. It follows that $\diamond$ must have been the lowest cell in its column at the start of the algorithm — again, a contradiction by Lemma 84, since it can change columns. Thus, all maximal sequences of minimal swaps produce the same result.

The argument above did rely on the special cells being non-virtual. However, it is not hard to fix it when any of them are virtual:

(1) If $*$ is virtual, it is still true that $\diamond$'s $\llcorner$ is in the same column in a and b — see Remark 94.
(2) If $\llcorner$ is virtual, $\diamond$ still ends up in the same row in a and b. Indeed, without $\llcorner$, there is no internal swap that moves $\diamond$ up.
(3) If $\lrcorner$ is virtual, the $\diamond$ can never move left. It also remains true that both fillings have it $\diamond$ the same row.

Regardless of which set of cells is virtual, we always find that a and b cannot have $\diamond$ in different locations. $\square$

6.4. The same equivalence in a different shape class?

If a shape is both top-left *and* bottom-right CR-erasable, the same equivalence follows by combining $\left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \circ\\ \circ & \bullet\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$. The resulting proof uses a recurrence, and is thus less informative than our bijection. Moreover, it seems plausible that the the recursive result is strictly weaker:

Conjecture 96. *Any shape which is both top-left and bottom-right CR-erasable consists of convex* 2×2*-connected components.*

CHAPTER 7

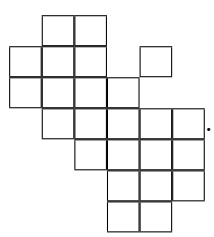
Empirical Observations and Open Problems

7.1. When are patterns inequivalent?

In the theorems above, we state whether the result appears to be empirically tight. That is, whether we found any shapes that do not satisfy the assumptions of the theorem, but have an equivalence between the corresponding pattern pairs.

Each theorem was tested on a set of about 160 shapes. The data set includes all 2×2-connected 3×3 examples, some Young diagrams, rectangles, skew shapes, other shapes that are convex, a number of shapes with no apparent regularities, and many shapes that were made up as tests of hypotheses, or counterexamples. Nonetheless, it is a small and unrepresentative set, and claims of empirical tightness should be taken with a grain of salt. We intend to check them more systematically on all shapes up to 5×4, and in particular to see how frequently sporadic coincidences occur. However, we do not have high hopes for results that say when two pps are *not* equivalent.

Simple experiments show that random (each cell is in the shape with probability $\frac{1}{2}$) shapes have no equivalences with probability $\approx \frac{1}{55}$ when drawn on a 5×5 grid, $\approx \frac{1}{12}$ on a 6×5 grid, $\approx \frac{1}{5}$ on a 6×6 grid. These numbers do not even account for many of these shapes not being 2×2-connected. So, it is extremely easy to make large shapes irregular enough to have no equivalences. For instance, this simple modification of a convex shape has none:



It would be interesting to precisely characterize the frequency of such shapes — as a subset of all shapes, and as a subset of all 2×2 connected shapes (most large shapes should be). We conjecture that a large random shape with a sufficiently high density of cells will almost always have this property.

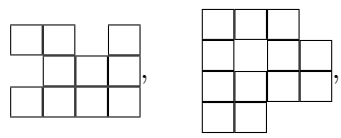
Problem 97. Find, as a function of $0 \leqslant \lambda \leqslant 1$, the fraction of $n \times n$ shapes with λn^2 cells, which have no equivalent pps.

7.2. Making indirect equivalences direct

Of the $\frac{64\cdot 63}{2} = 2016$ possible equivalences, we have (partially) described $160 = 13 \times 12 + 4$ — there are 12 pattern pairs, all connected through $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$, plus 4 Chapter 5 equivalences. Many of the descriptions, especially the indirect ones, are not tight. Here is an example of such a connection:

$$\left(\begin{smallmatrix}\circ & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \circ\\ \bullet & \circ\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\bullet & \circ\\ \circ & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\circ & \bullet\\ \circ & \circ\end{smallmatrix}\right). \tag{7.2.1}$$

In order for it to work, the shape must be top-right CR-erasable *and* bottom CR-erasable. Requiring both pps to be equivalent to $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ gives excessively strong conditions for the relation between the two. For (7.2.1), this is true empirically — the equivalence holds in many shapes where $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ differs from both:



etc.

As another example, the bijection in Chapter 6 also has an indirect proof, which is conjecturally weaker than the bijection — e.g. the bijection works in skew shapes, which cannot be both BR and TL CR-erasable. It is also conceivable that these results have a common generalization.

A common generalization is exactly what we got for $\left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \bullet\\ \bullet & \circ\end{smallmatrix}\right) \leftrightarrow \left(\begin{smallmatrix}\circ & \bullet\\ \bullet & \circ\end{smallmatrix}\middle|\begin{smallmatrix}\bullet & \circ\\ \circ & \bullet\end{smallmatrix}\right)$ in Proposition 4.5 by combining BR-erasing steps, Chapter 6 bijections through convex components, and TL-erasing steps. That took the equivalence from being far too weak to being *almost* empirically tight.

So, one could certainly tighten many of the equivalences between the already described pattern pairs. Furthermore, of the 51 untouched pattern pairs, many appear to be involved in sporadic equivalences. With further experimentation, one might be able to identify the additional shape properties, which enable them.

7.3. A hierarchy of shape classes

In Chapter 3, we gave all the equivalences which hold in every Young shape. There are many other potentially interesting classes of shapes, which are more and less general: rectangles (4-corner shapes), rotations of Young shapes (3-corner shapes), skew, flipped-skew, and half-rectangle shapes (2-corner shapes), 1-corner shapes, general convex shapes — "moon" shapes, left-leaning shapes, and right-leaning shapes. It would be great to explicitly describe *all* the equivalences which hold uniformly in each of these convex shape classes, and to depict out how equivalences disappear as the shape properties are relaxed. The results in this paper should suffice to cover most of these classes, but even so, presenting this classification of equivalences in convex shapes would be an interesting expository challenge.

There is plenty to say about non-convex shapes, too, but our results only offer the recursive and opaque description of shapes in terms of complete-rectangle erasability. Perhaps it is possible to reformulate some of the hypotheses in terms of forbidden subshapes? Moving away from this recursive shape description might lead to tighter indirect equivalences, too.

7.4. How to count the number of pattern-avoiding fillings?

It may be desirable to know more than just the equivalences between pairs of pps. The recurrences in this paper can be used to count the number of pafs in some circumstances, and *slowly*. Thus, many questions remain:

PROBLEM 98. Given a set of pattern pairs, is it possible to compute the number of the corresponding pafs in a shape (satisfying hypotheses, or arbitrary) in polynomial time (in the number of cells in the shape)?

The paper [**AS04**] by Alon and Shapira is a very good start.

PROBLEM 99. Are there explicit formulas counting the number of specific pafs? In specific shapes? (Kitaev et al. answer this for rectangular shapes in [**KMV05, Ki04**])

PROBLEM 100. Is there a good description of shapes, in which the number of certain pafs is strictly less (greater) than that of another kind of pafs? That is, a theory of pattern-ordering and not just of pattern-equivalence.

PROBLEM 101. Look at "nice enough" shapes (e.g. dense enough, with a height to width ratio not far from 1, perhaps convex, perhaps even Young, rectangular, or square) as the number of cells n grows. The total number of fillings is 2^n. Can we compute the asymptotic numbers of various pafs as a function of n? Empirically, it seems to be exponential, with rates between approximately $\Theta(2^{0.4n})$ to $\Theta(2^{0.6n})$. The exponent depends on the pattern pair, of course. So, it might be possible to asymptotically order the patterns, and perhaps even refine the ordering along the lines of [**St07**].

CHAPTER 8

Popeye Diagrams

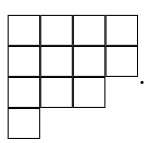
Popeye diagrams[1] are nice bit of combinatorics that appeared in the complicated bijection of Chapter 5. Though their relevance to pattern-avoiding fillings is a bit indirect, they have some intriguing connections to other areas of combinatorics, so it seemed appropriate to include a brief overview.

Popeye diagrams on the shape "the $n \times n$ square minus the $(n-1)$-side staircase" are exactly Hugh Thomas's [**Th02**] snug partitions for $d = 2$ — these are counted by the Catalan numbers, as mentioned in Stanley's Catalan Addendum [**St09**]. We will refer to them as *maximal Popeye diagrams.*

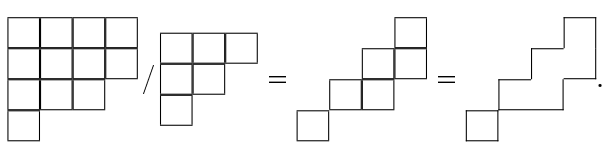
Taking the exchange (Section 8.2) as a cover relation, these maximal Popeye diagrams describe the Tamari lattice. Furthermore, "smaller" Popeye diagrams, which use a partition fitting inside the $n \times n$ square, give sublattices of the Tamari lattice (Remark 107). Finally, any lattice order on Dyck paths arranges these sublattices into a lattice. So, we get a lattice of sublattices of the Tamari lattice $\overset{\shortparallel}{\smile}$.

8.1. Definitions

We will draw Popeye diagrams on special skew-shapes. A skew-shape λ/μ is a *spinach shape* if λ and μ are of a particular form. λ is a Young shape obtained from a $n \times n$ square by thinking of its right and bottom edges as a Dyck path $(U \ldots UD \ldots D)$, and by substituting any $2n$-step Dyck path instead. E.g., the path $UUDUDDUD$ gives:

$\lambda =$  .

Given such a λ, μ is just the staircase shape with $n-1$ rows. For example, this is a spinach shape:

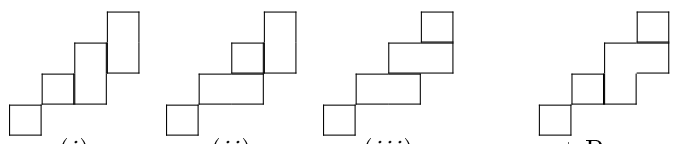
$\lambda/\mu =$  .

[1]When I first introduced these objects to several combinatorics students at MIT, they suggested many intriguing ideas for what they might be equivalent to. None of them seemed to work, so I started saying "these things are what they are". Hence the name.

To draw Popeye diagrams, we will keep only the outline of the shape, as above.

We will write $|\lambda/\mu| = n$ to denote the spinach diagram's size.

Subtracting the staircase shape creates $n-1$ $\sqcup$-corners on the upper-left edge of the skew shape. Call the corner points *branch points.* To get a *Popeye diagram* on a shape λ/μ, draw either a horizontal line going left or a vertical line going down starting from every branch point. The lines can be drawn in any order, and each line continues until it hits another line or the shape's edge. The only other restriction is that the result has to be a partition of λ/μ *into rectangles.* Here are all 3 Popeye diagrams for our example shape, and one counterexample:

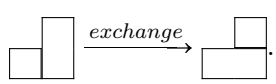


(i) (ii) (iii) not Popeye

FACT 102. *In any Popeye diagram, a* $\sqcap$*-corner on the boundary of the shape must be touched by at least one horizontal or vertical line emitted from a branch point.*

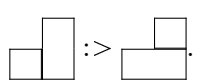
8.2. Exchanges and the Popeye lattice

We can transform one Popeye diagram D into another via an *exchange* operation. Two rectangles in D are *exchangeable* if they are vertically adjacent with a common bottom edge, or horizontally adjacent with a common right edge. The exchange operation toggles the line separating them between vertical and horizontal. For example, in diagram (i) above, there is one pair of exchangeable rectangles:



Notice that a single exchange operation takes (i) to (ii), and another exchange takes (ii) to (iii).

DEFINITION 103. Given a spinach shape s, the *Popeye lattice* $P(s)$ is the poset of all Popeye diagrams on s, with exchange taken as the cover relation (denoted $<$:). Vertical is greater than horizontal, e.g.



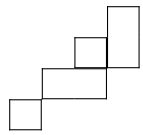
PROPOSITION 104. *For any spinach shape s, $P(s)$ is a lattice.*

PROOF. By symmetry, it is enough to show that $P(s)$ is a meet-semilattice.

Every Popeye diagram with branch points, labeled $1, \ldots, n-1$ from left to right, has a unique representation as:

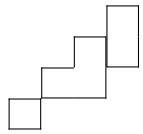
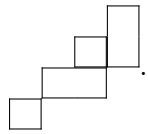
$$(l_i | i = 1, \ldots, n),$$

where l_i is the height of the vertical line at branch point i (or 0 if i's line is horizontal). E.g.



$\longrightarrow (0, 0, 1)$.

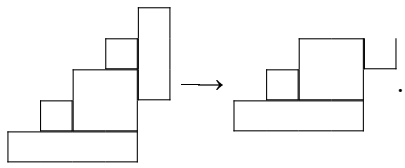
To see this is an injection, let us reconstruct the Popeye diagram from its line-height tuple. On the empty spinach shape, draw all the vertical lines. Next, from each branch point without a vertical line, draw a horizontal line that goes until it touches a vertical line or the shape's border:

$(0, 0, 1) \longrightarrow$  $\longrightarrow$  .

Notation. By abuse of notation, l_i will also stand for the whole line, from the branch point to the bottom point. If l_i is continued by the shape's border, then $\mathrm{ext}(l_i)$ will stand for the line plus the continuation. E.g., above, $\mathrm{ext}(l_2)$ has length 2.

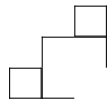
Claim 105. A line-height sequence (l_i) corresponds to a Popeye diagram on a shape s if and only if:

(1) The heights fit inside s. That is, if you draw the vertical lines in s, only the final point of each line may contact the boundary of s.
(2) For any vertical line l_i, let $j > i$ be the leftmost branch point with $\mathrm{ext}(l_j)$ ending below the branch point $i+1$. Then, $\mathrm{ext}(l_j)$ must end below l_i. In the following example, take $i = 2$ and look below $i + 1$ to get:



Here, $j = 4$, and l_i ends 2 units below $i + 1$, whereas $\mathrm{ext}(i_j)$ ends 3 units below $i + 1$.

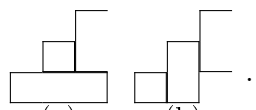
Proof of Claim. *Only if:* (1) is obvious. If (2) is violated, then look at the ⌈-corner above the problematic branch point. It bounds (below and to the right) a region of the shape. This must be a rectangle, but by assumption, the left vertical edge is strictly taller than the right vertical edge:



Thus, it cannot be a rectangle, and (2) is necessary.

If: It follows from the reconstruction algorithm and (1) that each branch point will emit either a horizontal or a vertical line, which will continue until

it touches another line. It remains to check that all parts of the diagram are rectangles. Consider a point in some region. Moving up and to the left as far as possible in this region leads to a unique ⌈-point on the upper-left boundary of the shape. This is because (i) a region must have a ⌈-corner; (ii) this corner cannot be on the inside of the shape, since lines do not cross, and only one line exits from each branch point. One step below this ⌈-corner is a branch point. If it emits a horizontal line, the region is clearly a 1-tall rectangle. Otherwise, consider the branch point one step to the right of the ⌈-corner. If it emits a vertical line, the region is a 1-wide rectangle. So, we may assume that the ⌈-corner is continued down by a vertical line, and to the right by a horizontal line. The former comes to an end either (a) at a horizontal line from some other branch point, or (b) at a ⌈-corner of the lower-right boundary of the shape:

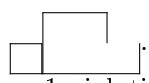


(a) (b)

In either case, this is a ⌊-corner of the region, with a horizontal line going right. By a similar argument, the horizontal line from the ⌈ comes to a ⌉-corner c with a vertical line going down.

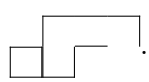
Going right from the ⌊-corner, the line necessarily encounters a ⌋-corner. Let d be the first (leftmost one) one. The relative positions of c and d can lead to two distinct violations of (2):

(1) The ⌉-corner c cannot be to the left of d — if it were, c's vertical line would have to stop before reaching the horizontal line coming into d:



type 1 violation

(2) Similarly, c cannot be to the right of d — if it were, d's vertical line would not go up so far as to touch c's horizontal. That would mean that d is part of the shape's lower-left boundary, and that this boundary is strictly above d, wherever c's vertical hits it:



type 2 violation

We are assuming that (2) holds, so c and d have to be exactly above each other. Moreover, their vertical lines have to merge. Thus, the region is a rectangle. □

Popeye diagrams have a very simple meet operation in terms of line-height sequences (subject to (1) and (2)).

LEMMA 106. *Given two Popeye line-height sequences* (a_i) *and* (b_i) *for diagrams* a *and* b*, their meet is* $(c_i = \min(a_i, b_i))$*, another Popeye line-height-sequence. The resulting meet-semilattice is equal to* $P(s)$*.*

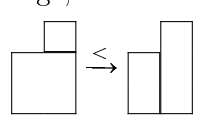
PROOF OF LEMMA. First, we check that (c_i) is Popeye. It obviously satisfies rule (1).

If (2) is violated, there is a special pair of vertical lines at branch points $i_1 < i_2$. Firstly, i_2 must be the leftmost branch point $> i_1$ such that $\mathrm{ext}(c_{i_2})$ goes below the $i_1 + 1$ branch point. Secondly, to violate (2), $\mathrm{ext}(c_{i_2})$ must also end above c_{i_1}. We may assume without loss of generality that $a_{i_2} < b_{i_2}$, so that $\mathrm{ext}(a_{i_2})$ does not go as low as $a_{i_1} \geqslant c_{i_1}$. Since a is Popeye, there is a horizontal line passing through the bottom end of $\mathrm{ext}(a_{i_2})$. This line could not have originated from a branch point, because it would have then crossed a_{i_1}. Therefore, the horizontal line is the bottom border of the shape, and $a_{i_2} = b_{i_2}$. Moving to the left along this horizontal line, we must reach a $\ulcorner$-corner, because the shape's bottom border is lower at i_1. Because of the vertical line at i_1, a horizontal line cannot reach this corner, so by Fact 102, both a_j and b_j touch this corner. Hence, c_j does too. But $j > i_1$ is to the left of i_2, and $\mathrm{ext}(c_j)$ obviously goes below the $i_1 + 1$ branch point — contradicting that i_2 is leftmost with this property. Thus, (2) cannot be violated in (c_i).

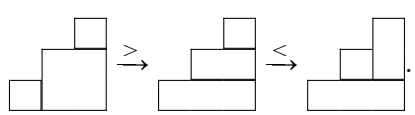
Our meet operation is clearly associative, commutative, and idempotent. So, it defines a valid meet-semilattice on Popeye diagrams.

Furthermore, this line-height meet defines a poset equivalent to that made by the exchange cover relation of Definition 103. We can define $a <_{lh} b$ in terms of the line-height meet as $a \wedge b = a$; then, $a <\!:_{lh} b$ means that $a <_{lh} b$ and no c exists with $a <_{lh} c <_{lh} b$. Correspondingly, $a <\!:_{exch} b$ means that a horizontal-to-vertical exchange takes a to b. We will show that $a <\!:_{lh} b$ if and only if $a <\!:_{exch} b$.

A horizontal-to-vertical exchange increases exactly one height value, leaving the rest fixed. Moreover, the increase is by the smallest possible amount. To see why, consider this exchange, which adds 2 units of height:



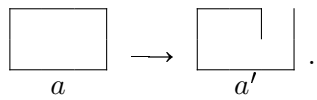
In order to achieve a smaller increase in height, the bottom-left rectangle would need to be shorter to start with, which implies that another line must be horizontal, rather than vertical:



Thus, a smaller increase in height necessarily can only happen in an incomparable diagram, proving that $a <\!:_{exch} b \Rightarrow a <\!:_{lh} b$.

To see the reverse, we need to characterize the cover relation in the line-height poset. First, if $a <\!:_{lh} b$, then $a_i = b_i$ for all but one i. Suppose

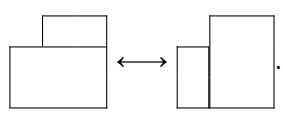
otherwise, and let j be the largest index such that $a_j < b_j$. Set $a'_i = a_i$ for $i \neq j$. and $a'_j = b_j$. Then, a' is a Popeye line-height sequence. It trivially satisfies condition (1). Neither a type 1 nor a type 2 violation of (2) is possible in a'. A type 2 violation would involve a too-short vertical line to the right of j. By construction, a and b are the same to the right of j, so if some line was too short for b_j in a', it would have to be too short in b too — but b is Popeye. A type 1 violation means that lengthening the line at j made it protrude into a previously untouched rectangle:



This is impossible, because this violation cannot be repaired by lengthening any other vertical lines, which is all that differentiates a' from the Popeye b.

Since $a_i = b_i$ for all i except $i = j$, it follows that no c with $a_j < c < b_j$ makes a Popeye diagram. Condition (1) is not violated, so these intermediate values must be prohibited by (2). We reason as in the previous paragraph: since b_j is valid, there can be no "too short" vertical line to the right of j. Thus, intermediate c values make a line that protrudes into the middle of a rectangle, and b_j closes off a new, narrower rectangle:

Since b_j gives a Popeye diagram, the region to the right of j must also be a rectangle, which shows that going between a_j and b_j is necessarily an exchange:



In other words, $a <:_{lh} b \Rightarrow a <:_{exch} b$, which completes the lemma's proof. □

To sum up: we introduced a line-height sequence encoding of Popeye diagrams, showed that the sequences have to obey just two rules in Claim 105, defined a meet on line-height sequences, and showed that the resulting meet semilattice is equivalent to the original poset $P(s)$. By symmetry, $P(s)$ is indeed a lattice. □

Remark 107. Given a non-maximal Popeye diagram, map it to a maximal one by (i) expanding the bottom-right border appropriately, (ii) growing the horizontal rays as far to the right as possible, (iii) growing the remaining vertical rays as far down as possible. This is not a canonical map (e.g. switch horizontal and vertical growth), but it is injective. It also preserves the lattice ordering.

8.3. Enumeration and arbitrary skew shapes

We include the following results without proof — the details will appear in a joint paper with Craig Desjardins.

PROPOSITION 108. *For a fixed spinach shape s, there are ρ_s Popeye diagrams:*

$$\rho_s = \sum_{\lambda \in \Lambda} (-1)^{l(\lambda)+k+1} \prod_{i \in \lambda} C_i,$$

where C_i is the ith Catalan number, or 0 for i negative, and Λ is a set of $3^{|s|}$ special tuples $\lambda = (i_1, i_2, \dots)$, such that $\sum_j i_j = |s|$, and $l(\lambda)$ denotes the number of elements of λ. The special tuples are obtained by taking *s of the shape, and casting either a horizontal ray to the left, a vertical ray up, or both, to partition the staircase shape. The partitions are read-off left-to-right, and when one overlaps a previously partitioned area, the corresponding i_j is negative.*

PROPOSITION 109. *Fixing a "generalized spinach shape" (λ/μ with μ arbitrary), the set of Popeye diagrams on this shape is a product of smaller Popeye lattices.*

PROPOSITION 110. *Among the "generalized spinach shapes" (λ/μ with μ arbitrary), ρ_s is highest for a fixed λ exactly when μ is a staircase.*

Bibliography

[AS04] N. Alon, A. Shapira, *Testing subgraphs in directed graphs*, J. Comput. Syst. Sci., **69** (2004), No. 3, pp. 354-382

[Bu07] A. Burstein: *On some properties of permutation tableaux*, Annals of Combinatorics **11** (2007), pp. 355–368, arXiv:math/0703004v1

[Je07] V. Jelínek: *Pattern-avoiding fillings of rectangular shapes*, Formal Power Series and Algebraic Combinatorics Nankai University, Tianjin, China, 2007, available at `http://www.fpsac.cn/PDF-Proceedings/Posters/47.pdf`

[Jo08] M. Josuat-Vergès: *Bijections between pattern-avoiding binary fillings of Young diagrams*, arXiv:0801.4928

[Ki04] S. Kitaev: *On multi-avoidance of right angled numbered polyomino patterns*, INTEGERS: Electronic Journal of Combinatorial Number Theory **4** (2004), A21

[KMV05] S. Kitaev, T. Mansour, A. Vella: *Pattern avoidance in matrices*, Journal of Integer Sequences **8** (2005), Article 05.2.2

[Kr06] C. Krattenthaler: *Growth diagrams, and increasing and decreasing chains in fillings of Ferrers shapes*, Adv. Appl. Math. **37** (2006), pp. 404-431, available at `http://www.mat.univie.ac.at/~kratt/artikel/growth.html`.

[MT04] A. Marcus, G. Tardos: *Excluded permutation matrices and the Stanley-Wilf conjecture*, Journal of Combinatorial Theory, Series A **107** (2004), No. 1, pp. 153-160, available at `http://www.renyi.hu/~tardos/submatrix.ps`.

[dM07] A. de Mier: k-noncrossing and k-nonnesting graphs and fillings of Ferrers diagrams, Electronic Notes in Discrete Mathematics **28** (2007), pp. 3-10. arXiv:math/0602195v2.

[Po06] A. Postnikov: *Total positivity, Grassmannians, and networks*, accessed at `http://math.mit.edu/~apost/papers/tpgrass.pdf` on November 14, 2007; arXiv:math/0609764 may be outdated

[Sp08] A.N. Spiridonov: *Pattern-avoidance in binary fillings of grid shapes* (extended abstract), 20th Annual International Conference on Formal Power Series and Algebraic Combinatorics, FPSAC 2008, DMTCS Proceedings, to appear. Available at: `http://math.mit.edu/~lesha/papers/pattern-pair-avoidance-extended-abstract.pdf`, poster at: `http://math.mit.edu/~lesha/papers/fpsac-2008-poster.pdf`

[St02] Z. Stankova: *A new class of Wilf-equivalent permutations*, J. Alg. Combin. **15** (2002), No. 3, pp. 271–290

[St07] Z. Stankova: *Shape-Wilf-ordering on permutations of length 3*, Electronic J. Combin. **14** (2007), No. 1, R56

[St09] R. Stanley, *Catalan Addendum,* available at `www-math.mit.edu/~rstan/ec/catadd.pdf`

[SW07] E. Steingrímsson, L. Williams: *Permutation tableaux and permutation patterns*, Journal of Combinatorial Theory, Series A **114** (2007), No. 2, pp. 211–234

[TeDB] B. Tenner: *Database of permutation pattern avoidance*, available at `http://math.depaul.edu/~bridget/patterns.html`

[Th02] H. Thomas: *New combinatorial descriptions of the triangulations of cyclic polytopes and the second higher Stasheff–Tamari posets*, Order **4** (2002), No. 4, pp. 327–342

[VW92] D.L. Vertigan, D.J.A. Welsh, *The computational complexity of the Tutte plane: the bipartite case*, Combin. Probab. Comput. **1** (1992), pp. 181-187, available at `http://journals.cambridge.org/production/action/cjoGetFulltext?fulltextid=2199312`

[Wi05] L. Williams: *Enumeration of totally positive Grassmann cells*, Advances in Mathematics **190** (2005), No. 2, pp. 319–342. math.CO/0307271