Multivariate combinatorial exploration with regular strategies

by

Émile Nadeau

Dissertation submitted to the School of Technology,
Department of Computer Science
at Reykjavík University in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

November 2022

Thesis Committee:

Henning Ulfarsson, Supervisor
Assistant Professor, Reykjavík University, Iceland

Christian Bean, Examiner
Postdoctoral Researcher, Reykjavík University, Iceland

Jay Pantone, Examiner
Assistant Professor, Marquette University, United States

Karen Yeats, Examiner
Professor, University of Waterloo, Canada

ISBN 978-9935-9694-6-0 Print version
ISBN 978-9935-9694-7-7 Electronic version
Author ORCID: 0000-0002-3410-5244

The undersigned hereby certify that they recommend to the School of Technology, Department of Computer Science at Reykjavík University for acceptance this dissertation entitled **Multivariate combinatorial exploration with regular strategies** submitted by **Émile Nadeau** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy (Ph.D.) in Computer Science** .

Henning Ulfarsson, Supervisor
Assistant Professor, Reykjavík University, Iceland

Christian Bean, Examiner
Postdoctoral Researcher, Reykjavík University, Iceland

Jay Pantone, Examiner
Assistant Professor, Marquette University, United States

Karen Yeats, Examiner
Professor, University of Waterloo, Canada

30 nov. 2022
....
date

....
Émile Nadeau
Doctor of Philosophy

Multivariate combinatorial exploration with regular strategies

Émile Nadeau

November 2022

Abstract

In this thesis, two methods for automatic enumeration of permutation classes are studied.

The first method extends on the theory of combinatorial exploration. We review how combinatorial exploration works as a two phases process: the first phase creating a universe of rules and the second one searching that universe for a solution to a given enumeration problem. After that, we identify a situation where the search step does not succeed to provide an answer despite the universe "containing" (in some sense) one. We present a way to overcome this shortcoming by developing a new theory that allows to find these "answers". Then, we demonstrate how combinatorial exploration can be extended to apply to problems where some additional statistics are involved. This theory is applied to enumerate permutation classes for which we introduce a new decomposition strategy called fusion. Combining all the tools discussed so far, we provide the first direct enumeration of Av(1342).

For the second method, we study the staircase encoding of permutations, which maps a permutation to a staircase grid with cells filled with permutations. We study the restriction of the staircase encoding to many permutation classes where the encoding becomes a bijection. For each of these classes, we describe the image of the bijections using independent sets of graphs weighted with permutations. We then enumerate the permutation classes by deriving the generating function for the independent sets and then for their weighted counterparts. We use our results to uncover some unbalanced Wilf-equivalences of permutation classes.

Keywords: combinatorics, permutation, enumeration, algorithm, graph

Fléttufræðileg könnun með reglulegum kænskum í mörgum breytistærðum

Émile Nadeau

nóvember 2022

Útdráttur

Í þessari ritgerð er fjallað um tvær sjálfvirkar aðferðir við að telja umraðanaflokka.

Fyrri aðferðin útvíkkar fræðin um fléttufræðilega könnun. Við rifjum upp hvernig fléttufræðileg könnun virkar í tveimur skrefum: fyrsta skrefið býr til heim af reglum, og seinna skrefið leitar í þeim heimi að lausn að talningarvandamáli. Að því loknu finnum við aðstæður þar sem leitin skilar engri niðurstöðu, jafnvel þó heimurinn „innihaldi" (í ákveðnum skilningi) lausn. Við sýnum hvernig megi yfirstíga þetta vandamál með því að þróa nýja fræði sem leifir okkur að finna þessar „lausnir". Síðan sýnum við hvernig nota megi fléttufræðilega könnun megi útvíkka til að takast á við vandamál þar sem margar tölfræðir koma við sögu. Þessari fræði er beitt á umraðanaflokka, og þar skilgreinum við nýja kænsku sem kallast samruni. Með því að nota allar þessar niðurstöður tekst okkur að gefa fyrstu beinu talninguna á $\mathrm{Av}(1342)$.

Seinni aðferðin byggir á stigakóðun umraðana, sem varpar umröðunum í stigalaga kóðun þar sem hvert hólf inniheldur umröðun. Við rannsökum einskorðun þessarar stigakóðunar á umraðanaflokka þar sem kóðunin er gagntæk vörpun. Fyrir þessa flokka lýsum við mynd vörpunarinnar með því að nota óháð mengi í netum vigtuð með umröðunum. Við teljum umraðanaflokkana með því að finna framleiðniföll þessara óháðu mengja, með og án vigtun. Við notum þessar niðurstöður til þess að finna ójöfn Wilf-jafngildi umraðanaflokka.

Efnisorð: fléttufræði, umröðun, upptalning, reiknirit, net

À la mémoire de mon grand-père, Fernand, qui m'a toujours dit qu'il aurait aimé aller à l'école passé l'âge de 12 ans.

In memory of my grandfather, Fernand, who always told me he'd have liked to go to school past the age of 12.

Acknowledgements

I'd like first to thank Henning my supervisor for his support and the continuous stream of new ideas. Your support along the way was essential to the completion of this project. I want to thank Christian for introducing me to all the nitty-gritty details of the codebase as well as the numerous sessions of peer coding. Those definitely made both of us better coders. Thanks to Jay for solving (almost) all my systems of equations and keeping me on my toes with endless feature requests. Thanks to Ásta from the department office for the administrative support throughout the immigration process.

Merci à Myllie qui m'a encouragé à tenter cette expatriation. Sans tes encouragement, jamais, je ne me serais lancé dans cette aventure. Merci à Tristan et Laura, mes premiers amis en Islande. C'est avec vous que j'ai en premier découvert ce nouveau pays. Merci à tout le monde de Reykjavik Accueil de créer ce super réseau pour les nouveaux arrivants francophones.

Takk Yrsa og Kolbeinn fyrir móttaka á landið. Ég verður endalaus þakklát fyrir að hafa komið að sækja mig og allt dótið mitt þegar ég lenti á Íslandi. Takk kærelaga Jóhönnu fyrir stuðningin í siðasta köflunum af þetta verkefni. Þú var alltaf til staðar til að kenna mér eitthvað íslenskt orð eða róa mér þegar stess var aðeins oft mikið.

Finalement, merci à toute ma famille pour votre support constant. Mention spéciale à ma sœur Sandrine avec qui je peux toujours échanger sur les hauts et les bas de la vie académique.

The research in this thesis was supported by Reykjavik University Research fund, the Natural Sciences and Engineering Research Council of Canada and the Fonds de recherche du Québec.

Preface

This dissertation is original work by the author, Émile Nadeau. Portions of the text are used with permission from Bean, Nadeau and Ulfarsson [1] of which I am an author.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Enumerative combinatorics

Enumerative combinatorics is the most classical branch of combinatorics. It focuses on finding the number of certain combinatorial objects. Words, partitions, graphs, permutations, posets and chess piece arrangements are among the structures that are often studied. This variety of objects makes the field strongly connected to other areas of mathematics. At the heart of every problem in enumerative combinatorics is the *combinatorial set* (often called "combinatorial class" in the literature). A combinatorial set is a possibly infinite set of objects associated with a size function such that the number of objects of each size is finite. In Flajolet and Sedgewick [2], a combinatorial set is formally defined as follows:

Definition 1.1 (Definition I.1 in [2])**.** *A* combinatorial set *is a finite or denumerable set on which a size function is defined, satisfying the following conditions:*

1. *the size of an element is a non-negative integer;*
2. *the number of elements of any given size is finite.*

The size of an element a in a combinatorial set $\mathcal{A}$ is denoted as $|a|$. Axiomatically, a combinatorial set is thought of as a pair $(\mathcal{A}, |\cdot|)$ giving the set and the size function. Though in theory the same set could be studied with different size functions, the size function to use is usually clear from context. Therefore, we mostly refer to a combinatorial set by only referring to the set $\mathcal{A}$ itself.

The typical problem of enumerative combinatorics consists of counting the objects of each size in a combinatorial set. The notation $\mathcal{A}_n$ is used to denote the set $\{a \in \mathcal{A} : |a| = n\}$, *i.e.*, the set of objects of $\mathcal{A}$ that are of size n. We use $|\mathcal{A}_n|$ to denote the number of elements in the set $\mathcal{A}_n$.

Definition 1.2 (Definition I.2 in [2])**.** *The* counting sequence *of a combinatorial set is the sequence of integers $(|\mathcal{A}_n|)_{n\geq 0}$.*

The typical problem is therefore, given a combinatorial set, compute the counting sequence. As an example, we consider all words over the alphabet $\{0, 1\}$ as a combinatorial set where the size function $|\cdot|$ gives the number of letters in the word. For this combinatorial set, the counting sequence is $(2^n)_{n\geq 0}$.

In his book Enumerative Combinatorics [3], Stanley gives a short list of standard ways to give the counting sequence of a combinatorial set.

1. Closed form formula: $|\mathcal{A}_n|$ is given as the result of well-known functions and operations.
2. Recurrence relation: $|\mathcal{A}_n|$ is given in terms of previously calculated values of smaller size.
3. An algorithm is given to compute $|\mathcal{A}_n|$.
4. Asymptotic behaviour: We give a function g such that $\lim_{n\to\infty} \frac{|\mathcal{A}_n|}{g(n)} = 1$. That tells us that the counting sequence behaves as g for large enough sizes.
5. Generating function: We give a formal series representation
$$\sum_{n\geq 0} |\mathcal{A}_n| x^n$$
of the counting sequence.

In the example we used above, the generating function for words over the alphabet $\{0, 1\}$ is

$$\sum_{n\geq 0} 2^n x^n = 1 + 2x + (2x)^2 + (2x)^3 + \cdots = \frac{1}{1-2x}.$$

In this work, we focus on automatic methods to obtain the counting sequence using 2. and 5. with an emphasis on applying it to permutation classes that are introduced in the next section.

1.2 Permutation patterns

Despite some earlier appearances, e.g., in the work of MacMahon [4] in 1915, the modern study of permutation patterns is usually dated to its appearance in Exercise 5 of Section 2.1.1 of Knuth's *The Art of Computer Programming* [5]. In this exercise, the reader is asked to show that a sequence of distinct numbers $p_1, \ldots, p_n$ can be sorted using a single stack if and only if there are no indices $i < j < k$ such that $p_k < p_i < p_j$.

The *standardisation* of a sequence s of distinct numbers is the sequence obtained by replacing the i-th smallest entry by i. The result of the standardisation of a sequence of n elements is always a *permutation of size n*, *i.e.*, an arrangement of the numbers $1, 2, \ldots, n$. For example, 43251 is a permutation of size 5. The set of all permutations is $\mathcal{S}$, and the subset of permutations of size n is $\mathcal{S}_n$. We denote by ε, the permutation of size 0 and $|\sigma|$ the size of σ. A permutation σ *contains* a permutation π if π is the standardisation of a (not necessarily consecutive) subsequence

s of σ. In this context, we say that π is a *(classical) pattern* in σ. The subsequence s is called an *occurrence* of π in σ. If σ does not contain the pattern π, we say that σ *avoids* π. We represent the permutation σ on a grid by placing points at coordinates $(i, \sigma(i))$. For example, the permutation $\sigma = 194532678$ is pictured in Figure 1.1. The pattern $\pi = 4321$ is contained in σ since it is the standardisation of the subsequence 9532 of σ. We denote the containment of π in σ by $\pi \leq \sigma$. The permutation σ avoids the pattern 54321 since it does not contain any decreasing subsequence of size 5.

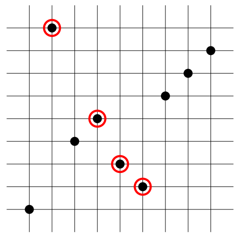


Figure 1.1: The permutation 194532678 with an occurrence of the pattern 4321 (circled in red).

Given a pattern π we define $\mathrm{Av}_n(\pi)$ as the set of permutations of size n that avoid π. For example, $\mathrm{Av}_3(123) = \mathcal{S}_3 \setminus \{123\}$. The set $\mathrm{Av}(\pi)$ is the set of all the permutations that avoid π. Formally,

$$\mathrm{Av}(\pi) = \bigcup_{n \geq 0} \mathrm{Av}_n(\pi).$$

For a set of patterns P, we let $\mathrm{Av}_n(P) = \cap_{\pi \in P} \mathrm{Av}_n(\pi)$, and $\mathrm{Av}(P) = \cup_{n \geq 0} \mathrm{Av}_n(P)$. Hence, $\mathrm{Av}(P)$ is the set of permutations that avoid all the patterns in P. The permutations in $\mathrm{Av}(12)$ are called the *decreasing permutations* and the permutations in $\mathrm{Av}(21)$ are called the *increasing permutations*.

Definition 1.3. *A* permutation class *is any set of permutations defined by the avoidance of a set of (classical) patterns.*

The minimal set of patterns needed to describe a permutation class is called the *basis* of the permutation class. Together with the size function, a permutation class is a combinatorial set as defined in Definition 1.1. The counting sequence of a permutation class $\mathrm{Av}(B)$, is $(|\mathrm{Av}_n(B)|)_{n \geq 0}$ and its generating function is

$$F(x) = \sum_{\sigma \in \mathrm{Av}(B)} x^{|\sigma|}.$$

In the last decades, a lot of research focused on finding the counting sequences of permutation classes. In 2018, Miner and Pantone [6] found the generating function for the counting sequence of the permutation class $\mathrm{Av}(2413, 3412)$ which was the last permutation class avoiding two size 4 patterns for which no generating function or recurrence relation was known. This paper completed an effort from the permutation pattern community that spanned across two decades and numerous articles [6]–[25]. The counting sequences of permutation classes avoiding a single size 4 pattern has been studied by Gessel [26], Bóna [27] and Stankova [28]. At the moment, no polynomial time algorithm is known for the counting sequence of Av(1324) and finding this counting sequence remains one of the main open problems in the field. A few automatic methods have been developed to find the counting sequences of permutation classes. Attention to those methods is given in Section 1.3.

1.3 Automatic methods of enumeration

Many techniques to compute the counting sequence of permutation classes have been developed throughout the years. One of the first automatic methods to appear was the idea of finitely labelled generating trees, used by Chung, Graham, Hoggatt and Kleiman [29] in 1978. Followed, in 1998, by the finite enumeration schemes introduced by Zeilberger [30]. The year 2005 saw the introduction of the insertion encoding by Albert, Linton and Ruškuc [31] and the substitution decomposition by Albert and Atkinson [32]. In 2018, Bean [33] introduced the Tilescope algorithm that was developed further in [34] of which I am an author. The following section takes a deeper look into finite enumeration schemes, the insertion encoding, the substitution decomposition and, the Tilescope algorithm.

1.3.1 Finite enumeration schemes

Finite enumeration schemes were developed by Zeilberger [30]. We give a quick overview using the notation of Vatter [35]. Broadly speaking, the method consists of breaking a permutation class into smaller parts and deriving recurrence relations for those parts.

Definition 1.4. *Given a basis B, $\pi \in \mathcal{S}_k$ and a* gap vector $\mathbf{g} \in \mathbb{N}^{k+1}$, *we define $Z(B;\pi;\mathbf{g})$ as*

$$\{p \in \text{Av}_{k+\|\mathbf{g}\|}(B) : p(g_1+1) = \pi(1), \ldots, p(g_1+\cdots+g_k+k) = \pi(k)\},$$

where $\|\mathbf{g}\|$ is the sum of the components of $\mathbf{g}$.

In words, $Z(B;\pi;\mathbf{g})$ is the set of permutations avoiding B where the pattern formed by the k smallest entries is determined by π and the number of entries between each of those k smallest entries is given by $\mathbf{g}$. In this context, π is referred to as a *downfix*. As an example, we consider $\pi = 132$ and $\mathbf{g} = (2,3,0,2)$ then

$$Z(B;132;(2,3,0,2)) = \{x_1x_21x_3x_4x_532x_6x_7 \in \text{Av}_{10}(B)\}.$$

For a fixed basis B, we define $\mathcal{J}(\pi)$ as the set of entries of $\mathbf{g}$ that must be zero in order for $Z(B;\pi;\mathbf{g})$ to be non-empty. Formally,

$$\mathcal{J}(\pi) = \{j \in \{1,\ldots,|\pi|+1\} : Z(B;\pi;\mathbf{g}) = \emptyset \text{ for all } \mathbf{g} \text{ with } g_j > 0\}.$$

Consider the basis $B = \{1432\}$. The permutations 4132, 1342 and 1324 are respectively in

$$Z(B;132;(1,0,0,0)),\ Z(B;132;(0,0,1,0)) \text{ and } Z(B;132;(0,0,0,1)).$$

Adding any entry between the 1 and the 3 in π would create a 1432 pattern. Hence, $\mathcal{J}(132) = \{2\}$. In general the set $\mathcal{J}(\pi)$ can be computed by looking at all the ways to add a new maximum to π. A gap vector $\mathbf{g}$ that *obeys* $\mathcal{J}(\pi)$ is a vector $\mathbf{g}$ such that $g_j = 0$ for all $j \in \mathcal{J}(\pi)$. If $\mathbf{g}$ does not obey $\mathcal{J}(\pi)$ then $Z(B;\pi;\mathbf{g})$ is guaranteed to be empty.

The *reduction* of the r-th entry of a permutation π is obtained by removing $\pi(r)$ and standardizing. We denote it $d_r(\pi)$. The *reduction* of the r-th entry of a gap vector is obtained by merging two gaps as shown below

$$d_r(\mathbf{g}) = (g_1,\ldots,g_{r-1},g_r+g_{r+1},g_{r+2},\ldots,g_{k+1}).$$

We say that the entry $\pi(r)$ is *enumeration-scheme-reducible for π with respect to B* if

$$|Z(B;\pi;\mathbf{g})| = |Z(B;d_r(\pi);d_r(\mathbf{g}))|,$$

for all gap vectors $\mathbf{g}$ that obey $\mathcal{J}(\pi)$. For conciseness, we just say that the entry $\pi(r)$ is ES-reducible. Zeilberger showed that it can be decided with a finite check:

Proposition 1.5. *The entry $\pi(r)$ of π is ES-reducible if and only if*

$$|Z(B;\pi;\mathbf{g})| = |Z(B;d_r(\pi);d_r(\mathbf{g}))|,$$

for all gap vectors $\mathbf{g}$ that obey $\mathcal{J}(\pi)$ and satisfy $\|\mathbf{g}\| < \|B\|_\infty - 1$, where $\|B\|_\infty$ is the size of the longest permutation of B.

We complete this section with the computation of $|\operatorname{Av}_n(132)|$ from Vatter's paper [35]. We first have that

$$\operatorname{Av}_n(132) = Z(\{132\};\varepsilon;(n))$$

and

$$|Z(\{132\};\varepsilon;(g_1))| = \sum_{i=0}^{g_1-1} |Z(\{132\};1;(i,g_1-i-1))| \qquad \text{for } g_1 > 0,$$

$$\begin{aligned}|Z(\{132\};1;(g_1,g_2))| = &\sum_{i=0}^{g_1-1} |Z(\{132\};21;(i,g_1-i-1,g_2))| \\ &+ \sum_{i=0}^{g_2-1} |Z(\{132\};12;(g_1,i,g_2-i-1))| \qquad \text{for } g_1,g_2 > 0.\end{aligned}$$

One can verify $\mathcal{J}(12) = \{2\}$ and $\mathcal{J}(21) = \emptyset$. Using Proposition 1.5, we obtain that 1 is ES-reducible in 12 and 21. Then for $g_1, g_2 > 0$ we get:

$$\begin{aligned}
|Z(\{132\};1;(g_1,g_2))| &= \sum_{i=0}^{g_1-1} |Z(\{132\};21;(i,g_1-i-1,g_2))| \\
&\quad + |Z(\{132\};12;(g_1,0,g_2-1))| \\
&= \sum_{i=0}^{g_1-1} |Z(\{132\};1;(i,g_1+g_2-i-1))| \\
&\quad + |Z(\{132\};1;(g_1,g_2-1))| \\
&= \sum_{i=0}^{g_1} |Z(\{132\};1;(i,g_1+g_2-i-1))|.
\end{aligned}$$

Those recurrences can be used to compute the values of $|\operatorname{Av}_n(132)|$ for any $n \in \mathbb{N}$. No general theory describes when an enumeration scheme can and cannot be found but a Maple package has been built by Zeilberger to find the schemes. In 2008, Zeilberger's enumeration schemes were extended by Vatter [35] who introduced gap vectors and covered more permutation classes. Recently a new generalisation was introduced by Biers-Ariel [36]. It recovers everything from Vatter's schemes and the regular insertion encoding which will be the topic of the next section.

1.3.2 Insertion encoding

The insertion encoding of permutations is a way to use formal language theory to find the counting sequence of a permutation class. The core idea of the encoding is to describe with a word how a permutation is built by successively adding new maxima. To do so, we introduce the notion of a slot in a permutation. The slots are the places where a new maximum can be added to the permutation. We use a $\diamond$ to mark the slots in a permutation. The notation $123 \diamond 54 \diamond$ represents the permutation 12354 where the $\diamond$ holds the guarantee of a point between the 3 and the 5 and at the end of the permutation. There are four different ways of adding a maximum n in a slot:

- the slot can be filled: we replace $\diamond$ by n,
- the maximum is added on the left of the slot: we replace $\diamond$ by $n\diamond$
- the maximum is added on the right of the slot: we replace $\diamond$ by $\diamond n$
- the maximum is added in the middle of the slot: we replace $\diamond$ by $\diamond n \diamond$

Those operations correspond to the letters f_i, l_i, r_i and m_i where i indicates in which slot the insertion is performed. The insertion encoding of 2647153 is

$$m_1 l_1 r_2 m_1 f_3 f_1 f_1.$$

The sequence of insertion is shown below:

$\diamond \to \diamond 1 \diamond \to 2 \diamond 1 \diamond \to 2 \diamond 1 \diamond 3 \to 2 \diamond 4 \diamond 1 \diamond 3 \to 2 \diamond 4 \diamond 153 \to 264 \diamond 153 \to 2647153$

Given a permutation class, we consider the language of the insertion encodings of the permutations contained in it. Albert, Linton and Ruškuc came up with a simple criterion to determine if the language of a permutation class is regular.

Theorem 1.6 (Albert, Linton and Ruškc [31]). *The insertion encoding of a finitely based permutation class* $\mathrm{Av}(B)$ *forms a regular language if and only if* B *contains at least one permutation from* $\mathrm{Av}(132, 312)$, $\mathrm{Av}(213, 231)$, $\mathrm{Av}(123, 3142, 3412)$ *and* $\mathrm{Av}(321, 2143, 2413)$.

We say that a permutation class is *insertion encodable* if the language of its insertion encoding is regular. For an insertion encodable permutation class, the finite accepting automata can be determined automatically using the method developed by Vatter [9]. Afterwards, tools of regular languages can be used to find the generating function. It is worth noting that the generating function of a regular language is always rational. Hence, permutation classes which are insertion encodable have a rational generating function.

1.3.3 Substitution decomposition

An *interval* of a permutation σ is a set of consecutive indices $i, i+1, \ldots, j$ such that the values $\sigma(i), \sigma(i+1), \ldots, \sigma(j)$ are also consecutive. In the permutation of Figure 1.2, the red region highlight the interval $\{3, 4, 5, 6\}$ where the values are 4532. The blue region highlights the set of consecutive indices $\{2, 3, 4, 5, 6, 7\}$ that is not an interval since $\sigma(7) = 6$, $\sigma(2) = 9$ and but $8 = \sigma(9)$. Every permutation

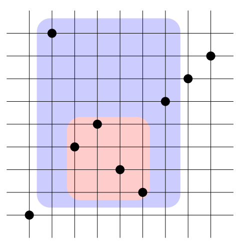


Figure 1.2: Interval in a permutation.

in $\mathcal{S}_n$ has at least one interval of size n containing all of its indices and n singleton intervals containing one index each. A permutation that has no other interval is a *simple permutation*. The first few simple permutations are $1, 12, 21, 2413, 3142, \ldots$.

Let σ be a permutation of size k and $\alpha_1, \alpha_2, \ldots, \alpha_k$ be permutations. We define the *inflation* of σ with the $\alpha_1, \alpha_2, \ldots, \alpha_k$, denoted $\sigma[\alpha_1, \alpha_2, \ldots, \alpha_k]$, as the permutation obtained by replacing π_i in π by an interval that where the value form the permutation α_k when standardised. The permutation $(213)[1, 132, 1324]$ is pictured in Figure 1.3 with the inflated intervals in gray.

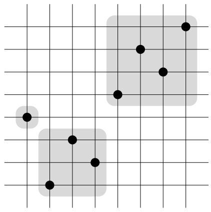


Figure 1.3: Inflation the permutation 213.

Let $\sigma_1, \sigma_2 \in \mathcal{S}$. We define $\sigma_1 \oplus \sigma_2$, the *sum of permutations*, as $12[\sigma_1, \sigma_2]$. The *skew-sum of permutations*, denoted $\sigma_1 \ominus \sigma_2$, is $21[\sigma_1, \sigma_2]$. If a permutation σ cannot be expressed as the sum or the skew-sum of two non-empty permutations, we say that σ is *sum-indecomposable* or *skew-indecomposable*, respectively. Simple permutations play the role of building blocks for permutations. Any permutation can be described as the inflation of a simple permutation.

Proposition 1.7 (Albert and Atkinson [32]). *Let σ be a permutation of size greater than 1. There is a unique simple permutation π of size greater than 1 and a unique sequence $\alpha_1, \alpha_2, \ldots, \alpha_k \in \mathcal{S}$ such that*

$$\sigma = \pi[\alpha_1, \alpha_2, \ldots, \alpha_k].$$

If $\pi \neq 12, 21$ then $\alpha_1, \alpha_2, \ldots, \alpha_k$ is unique. If $\pi = 12$ or 21 then α_1, α_2 are also unique if α_1 is sum-indecomposable or skew-indecomposable respectively.

Hence, any permutation in a permutation class can be described by the inflation of the simple permutations of the permutation class. When a permutation class has finitely many simple permutations, it is possible to look at all the ways for the forbidden patterns to stretch across the blocks of a permutation in the permutation class. This type of argument is used by Albert and Atkinson to prove that:

Theorem 1.8 (Albert and Atkinson [32]). *The generating function of every permutation class that contains finitely many simple permutations is algebraic.*

An algorithm that fully automates those techniques has been developed by Bassino, Bouvel, Pierrot and Rossin [37]. An algorithm to determine if a permutation class has finitely many simple has also been developed by Bassino, Bouvel, Pierrot, Rossin [38] and is implemented in Permuta [39].

1.3.4 Combinatorial exploration and the Tilescope algorithm

In this section we cover the Tilescope algorithm introduced by Albert, Bean, Claesson, Nadeau, Pantone and Ulfarsson [34] and Bean [33]. The Tilescope algorithm is an implementation for permutations of *combinatorial exploration*, a general framework to automatically find counting sequences of combinatorial sets. Combinatorial exploration was also introduced alongside the Tilescope algorithm in [34] and [33]. Combinatorial exploration works by systematically applying a small set of strategies to combinatorial sets to create rules. We call this set of rules the *universe*. In this universe, the algorithm periodically searches for a combinatorial specification (a set of rules where each combinatorial set appears exactly once as a left-hand side) and presents it as a *proof tree*. Combinatorial exploration and its applications to the domain of permutation patterns will be the main focus of Chapters 2 and 3. We will for now avoid any technical definitions and give the reader a feel for combinatorial exploration through an example. For a more precise definition, we will, as needed, refer the reader to the appropriate sections of [34] or defer them until Chapter 2. For our example, we derive a proof tree for Av(132).

To be ready for our example, we first need to introduce some concepts. A *gridded permutation* is a permutation where each point lives in a cell of a grid. We write them as a pair (π, P) where π is a permutation of size n and P is an n-tuple of pairs in $\mathbb{N}^2$. When the gridded permutation is contained in a single cell c we often abbreviate $(\pi, (c, c, \ldots, c))$ to (π, c). Like for permutations, there is a notion of containment for gridded permutations. We say that the gridded permutation (σ, P) contains the gridded permutation (π, Q) if there is a subsequence of (σ, P) such that the subsequence of the permutation standardizes to π and the subsequence of cells is exactly Q. For example the gridded permutation $(2143, ((0,0), (0,0), (0,1), (0,0))$ contains two occurrences of the gridded permutation $((12, ((0,0), (0,1))$. The subsequence of indices corresponding to the occurrences are $1, 3$ and $2, 3$. We write $\mathcal{G}^{(t,u)}$ to represent the set of gridded permutations in the region $[0, t) \times [0, u)$. A more detailed introduction to gridded permutations can be found in Section 6.1 of [34].

A *tiling* is a way to represent a set of gridded permutations. It consists of a finite grid together with a set of *obstructions* and a set of *requirements*. Obstructions are a set of gridded permutations that must be avoided by the permutation in the set represented by the tiling. Requirements are a set of sets of gridded permutations. To be in the set of gridded permutations represented by the tiling, a gridded permutation must contain at least one gridded permutation from each of the sets in the requirements. We always draw obstructions in red and requirements in blue. Figure 1.4 shows examples of two tilings. The one on the left is the empty tiling. The red point is a *point obstruction*. This tiling represents the set of permutations with points in a 1×1 grid such that no point of the gridded permutations are in this cell. In this case, the only gridded permutation is the empty one. For convenience, we picture a cell with a point obstruction as an empty cell. On the right, we find the *point tiling*. We see in red that the cell cannot contain a 12 or a 21 pattern. Hence, the only permutations that can be gridded on this tiling are ε and 1 where the 1 lives in cell $(0, 0)$. However, this tiling also has a requirement. The blue point is the *point requirement* which means that the cell must contain a point. Hence, the only permutation that can be gridded on this tiling is 1. For convenience, we represent such a cell with a black point instead of drawing the two obstructions and the requirement.

Figure 1.5 shows a more complex tiling. We see the 123 obstruction that crosses from cell $(0, 0)$ to cell $(2, 0)$, meaning that a 123 occurrence in a permutation on this tiling cannot have its 1 in the cell $(0, 0)$ and its 2 and 3 in cell $(2, 0)$. A gridded

Figure 1.4: Two simple tilings.

permutation on this tiling must also have exactly one point in the cell $(1,1)$ and an occurrence of 21 in the cell $(0,0)$. The gridded permutation 43512 with cells $(0,0),(0,0),(1,1),(2,0),(2,0)$ is a valid gridded permutation on this tiling. We write $\mathrm{Grid}(\mathcal{T})$ for the set of such gridded permutations on a tiling $\mathcal{T}$. We say that such gridded permutations are *griddable* on the tiling $\mathcal{T}$. Formally, a tiling and $\mathrm{Grid}(\mathcal{T})$ are defined as follows in [34].

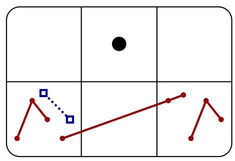


Figure 1.5: Example tiling.

Definition 1.9 (Definition 6.1 in [34]). *A* tiling *is a triple $\mathcal{T} = ((t,u), \mathcal{O}, \mathcal{R})$ where t and u are integers, $\mathcal{O}$ is a set of gridded permutations that we call* obstructions *and $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_k\}$ is a set of sets of gridded permutations that we call* requirements.

A tiling $\mathcal{T}$ represents the combinatorial set of gridded permutations $g \in \mathcal{G}^{(t,u)}$ such that g avoids all gridded permutations in $\mathcal{O}$ and g contains a gridded permutation from each $\mathcal{R}_i$ for $1 \leq i \leq k$. We call this set $\mathrm{Grid}(\mathcal{T})$*. In other words,* $\mathrm{Grid}(\mathcal{T})$ *is the set gridded permutations in the region $[0,t) \times [0,u)$ that avoid* all *of the patterns in $\mathcal{O}$ and contain* at least one *of the patterns in each $\mathcal{R}_i$. We call the individual gridded permutations in each $\mathcal{R}_i$* requirements *and we call each set $\mathcal{R}_i$ a* requirement list.

A more detailed introduction to tilings can also be found in [34] at Section 6.2. We now have all the tools we need to look at a tree describing the structure of the 132-avoiding permutation. This tree is pictured in Figure 1.6. The function F_i represents the generating function for the set $\mathrm{Grid}(\mathcal{T}^{(i)})$.

We first have the tiling labelled $\mathcal{T}^{(1)}$. It represents the permutations avoiding 132 gridded on a single cell. We can split this set based on whether the permutation contains a point or not. Hence, we get

$$\mathrm{Grid}(\mathcal{T}^{(1)}) = \mathrm{Grid}(\mathcal{T}^{(2)}) \sqcup \mathrm{Grid}(\mathcal{T}^{(3)})$$

where $\mathcal{T}^{(2)}$ is obtained from $\mathcal{T}^{(1)}$ by adding a point obstruction and $\mathcal{T}^{(3)}$ is obtained by adding a point requirement. In the terms of the generating functions, this means

$$F_1(x) = F_2(x) + F_3(x).$$

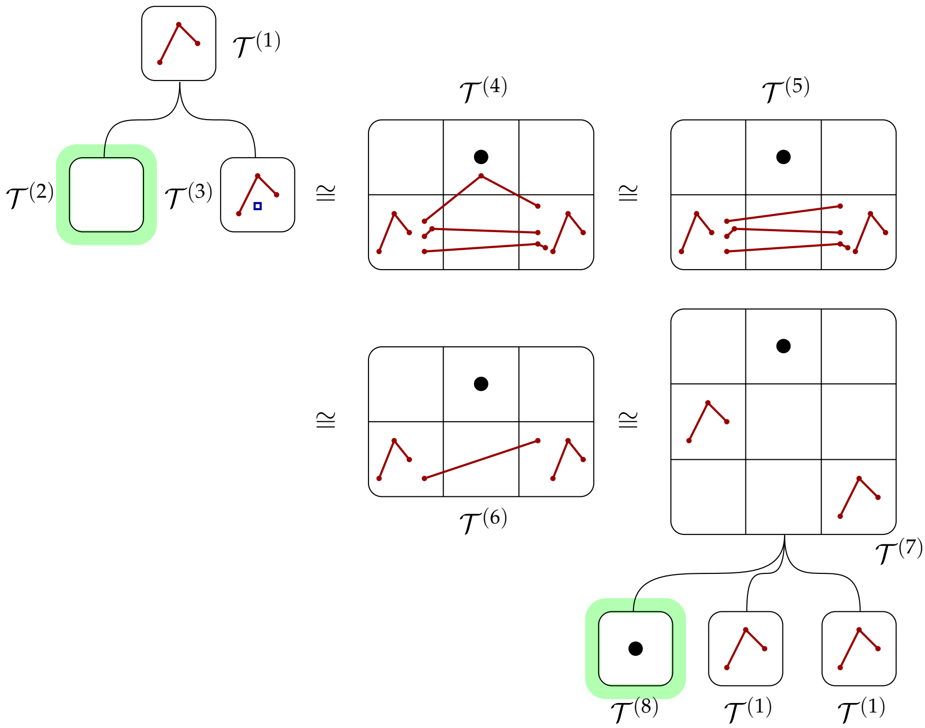


Figure 1.6: A Tilescope proof tree for $\text{Av}(132)$.

This rule is called *cell insertion*.

The only permutation that can be gridded on the tiling labelled $\mathcal{T}^{(2)}$ is ε. Hence,

$$F_2(x) = 1.$$

Since its counting sequence is known we say that $\mathcal{T}^{(2)}$ is verified. Verified tilings are marked in green in the figure.

From $\mathcal{T}^{(3)}$, we use a strategy called *point placement* that consists of forcing the point of a requirement on its own row and column. We assume that the point is the topmost one and place it in its own row and column. We still need to avoid 132, hence we stretch the 132 obstruction in all ways across the new tiling. That gives us the tiling $\mathcal{T}^{(4)}$. The latter is equinumerous to $\mathcal{T}^{(3)}$ in the sense that $\text{Grid}(\mathcal{T}^{(3)})$ and $\text{Grid}(\mathcal{T}^{(4)})$ have the same counting sequence. To get to $\mathcal{T}^{(5)}$, we observe that the 132 obstruction that has a point in cell $(1,1)$ is equivalent to a 12 obstruction going from cell $(0,0)$ to cell $(2,0)$ since a point is guaranteed in cell $(1,1)$. We get to $\mathcal{T}^{(6)}$ by observing that the crossing 12 makes the crossing 132 obstructions redundant. Finally, $\mathcal{T}^{(7)}$ is obtained with the observation that the crossing obstruction forces all the points in cell $(0,0)$ to be above points in cell $(2,0)$. This strategy is called *row separation*. All the steps described above do not change the counting sequence of the sets of griddable permutations on each tiling. Therefore, in terms of the generating functions we have

$$F_3(x) = F_4(x) = F_5(x) = F_6(x) = F_7(x).$$

Finally, we observe that the last tiling consists of three unrelated parts that can be split by a *factor strategy*. We get

$$F_7(x) = F_8(x) \cdot F_1(x)^2.$$

We say that $\mathcal{T}^{(8)}$ is *verified* since we know that the only griddable permutation is the point. Therefore, $F_4(x) = x$.

From these decompositions, we get a system of equations

$$\begin{aligned} F_1(x) &= F_2(x) + F_3(x) \\ F_2(x) &= 1 \\ F_3(x) &= F_7(x) \\ F_7(x) &= F_8(x)F_1(x)^2 \\ F_8(x) &= x \end{aligned}$$

that can be solved. The solution for F_1 is the generating function of the Catalan numbers as expected.

It is also possible to derive counting recurrences from the proof tree presented in Figure 1.6. After some basic simplifications, we obtain the following system of recurrences. As we will do later in the text, we use the notation $\mathcal{T}_n^{(i)}$ to denote $\mathrm{Grid}(\mathcal{T}^{(i)})_n$, *i.e.*, the set of valid gridded permutations of size n on $\mathcal{T}^{(i)}$. With this notation the system is:

$$\begin{aligned} |\mathcal{T}_n^{(1)}| &= |\mathcal{T}_n^{(2)}| + |\mathcal{T}_n^{(3)}| \\ |\mathcal{T}_n^{(2)}| &= \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases} \\ |\mathcal{T}_n^{(3)}| &= |\mathcal{T}_n^{(7)}| \\ |\mathcal{T}_n^{(7)}| &= \sum_{i=1}^{n-1} |\mathcal{T}_i^{(1)}||\mathcal{T}_{n-1-i}^{(1)}|. \end{aligned}$$

For readability, we use the fact that $|\mathcal{T}_n^{(8)}|$ is 1 if $n = 1$ and 0 otherwise to avoid having a double summation in the last equation.

The example above introduced some of the strategies that have been developed for the Tilescope algorithm. All the strategies used in that proof tree are introduced in Section 6.3 of [34]. The beginning of Chapter 2 covers in more detail the general notion of strategies and how combinatorial exploration can find proof trees like the one we just presented.

1.4 Overview

In Chapter 2, we introduce more of the theory underlying combinatorial exploration and the Tilescope algorithm. We also identify some of the limitations of that theory. A new underlying theory for finding specifications is then introduced and we prove its correctness. We conclude by showing how it can be applied to find specifications for permutation classes. Chapter 3 builds on the theory developed in the previous chapter but extends it to combinatorial exploration with catalytic variables. We introduce a new strategy called fusion and explain how combinatorial exploration can be extended to work with combinatorial sets for which we care about the enumeration with respect to some statistics. We conclude by presenting the first direct combinatorial description of Av(1342). This description is found automatically using the techniques developed in Chapters 2 and 3. Chapter 4 presents a different approach for the enumeration of permutation classes using inflation of independent sets of certain types of graphs. We slowly build up from simpler to more complex examples and conclude by using the results derived in the chapter to prove two unbalanced Wilf-equivalences, *i.e.*, we show that two permutation classes with bases of different sizes have the same counting sequence. Chapter 5 concludes the thesis by hinting at different extensions and improvements that could be applied to the method developed in the previous chapters.

Chapter 2

Forests

2.1 Background on combinatorial exploration

2.1.1 Strategies and combinatorial specifications

As mentioned in Section 1.3.4, combinatorial exploration is an automatic framework for the enumeration of combinatorial sets. In practice, the output of combinatorial exploration is a proof tree like the one presented in Figure 1.6. A proof tree is a rooted tree in the sense of graph theory where each vertex represents a combinatorial set. The combinatorial sets represented in the leaves of a proof tree must either have a known counting sequence, in which case we say that they are verified, or be represented as one of the inner vertices of the tree. In the proof tree of Figure 1.6, we know the counting sequence of the leaves that represent $\mathcal{T}^{(2)}$ and $\mathcal{T}^{(8)}$ while the other leaves represent $\mathcal{T}^{(1)}$ which is an internal vertex in the tree (the root). Each of the parent-to-children relationships in the tree gives a structural connection between the parent and the children. In the example of Section 1.3.4, we have shown how this relation translates both in terms of generating function equations and in terms of counting recurrences.

To formalize this notion of structural relationship, Albert et al. [34] introduced the concept of a combinatorial strategy. An m-ary combinatorial strategy S is composed of three components: a decomposition function d_S, a reliance profile function r_S and an infinite sequence of counting functions $\{c_{S,(n)}\}_{n\geq 0}$. The decomposition function takes as input a combinatorial set and returns an m-tuple of combinatorial sets or `DNA` which stands for "do not apply". When $d_S(\mathcal{A}) = \texttt{DNA}$, it means that the strategy S cannot be applied in a meaningful manner to the set $\mathcal{A}$. Otherwise, if $d_S(\mathcal{A}) = (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(m)})$, it means that there is a uniform way of computing the counting sequence $\mathcal{A}$ from the counting sequence of $\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(m)}$. The way to compute the counting sequence is described by the other two components of the combinatorial strategy: the reliance function and the counting functions. It is important to emphasize that those have to be the same no matter what the input $\mathcal{A}$ is. The counting is not dependent on the input of the decomposition function, $\mathcal{A}$.

The reliance profile function is a function from $\mathbb{N}$ to $\mathbb{Z}^m$ that records how much enumerative information from each of the combinatorial sets of the output is required to compute the number of objects of size n in the combinatorial set being decomposed. Finally, the counting function $c_{S,(n)}$ indicates how to compute $|\mathcal{A}_n|$ from the $|\mathcal{B}_j^{(i)}|$. Formally, a combinatorial strategy is defined as followed.

Definition 2.1 (Definition 3.1 in [34])**.** *Let $\mathcal{Z}$ be the collection of all combinatorial sets. An m-ary* combinatorial strategy *S consists of three components.*

1. *A* decomposition function *$d_S : \mathcal{Z} \to \mathcal{Z}^m \cup \{\texttt{DNA}\}$ whose input is a combinatorial set $\mathcal{A}$ (the parent set), and whose output is either an ordered m-tuple of combinatorial sets $(\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(m)})$ (the child sets) or the symbol DNA. When the output is $d_S(\mathcal{A}) = \texttt{DNA}$, short for "does not apply", we say that S* cannot be applied *to the combinatorial set $\mathcal{A}$.*

2. *A* reliance profile function *$r_S : \mathbb{N} \to \mathbb{Z}^m$ whose input is a natural number n and whose output is an ordered m-tuple of integers. We use $r_S^{(i)}(n)$ to denote the i-th component of $r_S(n)$, i.e.,*

$$r_S(n) = (r_S^{(1)}(n), \ldots, r_S^{(m)}(n)).$$

3. *An infinite sequence of* counting functions *$c_{S,(n)}$ indexed by $n \in \mathbb{N}$, each of whose input is m tuples of integers $w^{(1)}, \ldots, w^{(m)}$ and whose output is a natural number. The counting functions must have the property that if $d_S(\mathcal{A}) = (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(m)})$ and $r_S(n) = (r_S^{(1)}(n), \ldots, r_S^{(m)}(n))$, then for input tuples*

$$w^{(i)}(n) = \left(|\mathcal{B}_0^{(i)}|, \ldots, |\mathcal{B}^{(i)}_{r_S^{(i)}(n)}|\right)$$

we have

$$c_{S,(n)}(w^{(1)}(n), \ldots, w^{(m)}(n)) = |\mathcal{A}_n|.$$

To be overly explicit, the domain of $c_{S,(n)}$ is $\mathbb{N}^{D_1} \times \cdots \times \mathbb{N}^{D_m}$, where

$$D_k = \max(0, r_S^{(k)}(n) + 1),$$

while the codomain is simply $\mathbb{N}$.

Again, we stress that the reliance profile and the counting function are independent of the input of the decomposition. They have to work for any decomposition produced by the decomposition function.

The concrete application of a strategy to a combinatorial set is called a *combinatorial rule*. We denote it $\mathcal{A} \xleftarrow{S} (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(m)})$ as a way to mean that $d_S(\mathcal{A}) = (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(m)})$. When the information about the strategy is not relevant, we often omit the strategy from the notation and simply write $\mathcal{A} \leftarrow (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(m)})$. Even if, formally, the strategy applies to a combinatorial set, we will slightly abuse the notation to describe the effect on tilings when talking about permutation patterns. In this sense, $d_S(\mathcal{T}^{(3)}) = (\mathcal{T}^{(4)})$ should be read as $d_S(\text{Grid}(\mathcal{T}^{(3)})) = (\text{Grid}(\mathcal{T}^{(4)}))$.

A *combinatorial specification* is a set of rules where each combinatorial set in the specification appears on the left-hand side of exactly one of the rules. Since each combinatorial set appearing in the specification is on the left-hand side of one rule, there is a combinatorial strategy applying to each of the sets in the combinatorial specification. This gives a sequence of counting functions for each of the combinatorial sets. A combinatorial specification is, in fact, another way of representing a proof tree. Each parent-to-child relationship in a proof tree actually corresponds to a rule in the combinatorial specification. The combinatorial specification contains a rule for each leaf of the tree for which the counting sequence is known. These rules are created by a 0-ary strategy, meaning the decomposition function of the strategy returns an empty tuple when it applies. Such strategies are called *verification strategies*. If such a strategy S applies to a combinatorial set $\mathcal{A}$ then $d_S(\mathcal{A}) = ()$. Hence, the counting function doesn't need any input to compute $|\mathcal{A}_n|$. The condition of having the combinatorial sets of the other leaves appear as internal vertices translates into having each combinatorial set on the left-hand side of a rule. The combinatorial specification representation of the proof tree of Figure 1.6 is given below.

$$\begin{aligned}
\mathcal{T}^{(1)} &\leftarrow (\mathcal{T}^{(2)}, \mathcal{T}^{(3)})\\
\mathcal{T}^{(2)} &\leftarrow ()\\
\mathcal{T}^{(3)} &\leftarrow (\mathcal{T}^{(4)})\\
\mathcal{T}^{(4)} &\leftarrow (\mathcal{T}^{(5)})\\
\mathcal{T}^{(5)} &\leftarrow (\mathcal{T}^{(6)})\\
\mathcal{T}^{(6)} &\leftarrow (\mathcal{T}^{(7)})\\
\mathcal{T}^{(7)} &\leftarrow (\mathcal{T}^{(8)}, \mathcal{T}^{(1)}, \mathcal{T}^{(1)})\\
\mathcal{T}^{(8)} &\leftarrow ()
\end{aligned}$$

The definition of combinatorial specification discussed here is similar to the one given by Flajolet and Sedgewick [2], though they did not have the formalism of a strategy. While proof trees refer more to the tree form and combinatorial specifications to the set of rules, we will use these terms interchangeably in the text as they are two perspectives on the same object. For a more detailed introduction to proof trees, strategies and combinatorial specifications, we refer the reader to Sections 3.1 to 3.3 of [34].

2.1.2 Finding the specification

We have so far covered at length the output of combinatorial exploration but not said much about how combinatorial exploration finds those specifications. It is, in fact, a two stage process: the expansion phase and the search phase.

In the expansion phase, a set of strategies is used to decompose combinatorial sets. Starting with the combinatorial set we are interested in, combinatorial exploration applies strategies to decompose it into other sets. Every time a decomposition function applies to a combinatorial set, a rule is created. All the rules are collected together in a collection of rules called the *universe*. As new rules are created, new combinatorial sets appear. Combinatorial exploration then applies the strategies to those new sets, creating more rules. This process can go on until there is no new combinatorial set to apply the strategies to or until we pause it to enter the search phase. Note that, in reality the expansion phase is not as naive as described above. It is, in fact, governed by a complex system of queues with the goal of controlling the combinatorial explosion that often occurs at that stage. An interested reader will find more detail in Section 10 of [34] but the simpler model introduced here is sufficient to understand the content of this thesis.

The expansion phase is periodically paused to proceed to a specification search phase where combinatorial exploration identifies the rules that are in a specification. To do so it uses the prune method introduced in [34].

Algorithm 1 The prune method (Algorithm 1 from [34]).

1: **Input:** A set of combinatorial rules U
2: **Output:** The union of all combinatorial specifications contained in U
3:
4: *changed* $\leftarrow$ **True**
5: **while** *changed* **do**
6:     *changed* $\leftarrow$ **False**
7:     **for** $\mathcal{A} \xleftarrow{S} (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(m)}) \in U$ **do**
8:         **if** any $\mathcal{B}^{(j)}$ is not on the left-hand side of any rule in U **then**
9:             $U \leftarrow U \setminus \{\mathcal{A} \xleftarrow{S} (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(m)})\}$
10:             *changed* $\leftarrow$ **True**
11:         **end if**
12:     **end for**
13: **end while**
14: $V \leftarrow U$
15: **return** V

The algorithm works by continuously removing rules that cannot be in a combinatorial specification. This happens in the for-loop at line 7. For every rule, it checks if all the children appear on the left-hand side of another rule in the universe. If one of the children of a rule does not appear on the left-hand side of any of the remaining rules in U, then this rule cannot be in a specification built from rules in U since any combinatorial set in a specification must appear on the left-hand side of a rule. The prune method therefore removes this rule from the universe (line 9). The process is repeated until nothing changes (line 5). At that point every child of every rule in U is on the left-hand side of another rule in U. As proved in [34], the output is the set of all rules that are in a combinatorial specification.

Theorem 2.2 (Theorem 3.1 in [34])**.** *For any set of combinatorial rules U, the set V returned by Algorithm 1 is equal to the union of all combinatorial specifications that are contained in U.*

More details on how specification are found can be found in Section 3.4 and 3.5 of [34].

2.1.3 Finding a good specification

Sometimes combinatorial specifications do not contain enumerative information. This is actually observed by Albert Bean, Claesson, Nadeau, Pantone and Ulfarsson [34], where the following example of a specification that does not contain enumerative information is given. Consider

$$\begin{aligned}\mathcal{A} &\overset{S_1}{\leftarrow} (\mathcal{B},\mathcal{C})\\ \mathcal{B} &\overset{S_2}{\leftarrow} (\{\varepsilon\},\mathcal{C})\\ \mathcal{C} &\overset{S_3}{\leftarrow} (\{\},\mathcal{B})\end{aligned}$$

where ε is a combinatorial object of size 0. Let S_1 be a strategy identifying that $|\mathcal{A}_n| = |\mathcal{B}_n| + |\mathcal{C}_n|$, S_3 be a strategy identifying that $|\mathcal{C}_n| = |\{\}_n| + |\mathcal{B}_n|$ and S_2 identifies that

$$|\mathcal{B}_n| = \sum_{i=0}^{n} |\{\varepsilon\}_i| \cdot |\mathcal{C}_n - i|.$$

After simplification, we obtain that

$$\begin{aligned}|\mathcal{A}_n| &= |\mathcal{B}_n| + |\mathcal{C}_n|\\ |\mathcal{B}_n| &= |\mathcal{C}_n|\\ |\mathcal{C}_n| &= |\mathcal{B}_n|.\end{aligned}$$

This system of recurrences is of course not sufficient to compute the counting sequences of $\mathcal{A}$, $\mathcal{B}$ or $\mathcal{C}$. If we want to compute how many elements are in $\mathcal{A}_8$, the strategy S_1 tells us to first ask how many objects there are in $\mathcal{B}_8$ and $\mathcal{C}_8$. To compute the number of objects in $\mathcal{B}_8$, S_2 tells us to ask for the number of objects in $\mathcal{C}_8$ while to find the latter we need to first get the number of objects in $\mathcal{B}_8$. This is clearly an infinite cycle. We can therefore not compute those counts. Such a specification is said to be trivial.

The method described above is, however, the general manner a combinatorial specification should be used to compute counts. If we have a rule

$$\mathcal{C} \overset{S}{\leftarrow} (\mathcal{B}^{(1)},\ldots,\mathcal{B}^{(m)})$$

and want to compute $|\mathcal{C}_8|$, the strategy S tells us how much information is needed from each child (via the reliance function) and how to combine this information to obtain $|\mathcal{C}_8|$ (via the counting function). To obtain the needed counts from each child $\mathcal{B}^{(j)}$, it recursively looks at the rule with $\mathcal{B}^{(j)}$ on the left-hand side and uses it to compute all the needed $\mathcal{B}_i^{(j)}$.

Consider a combinatorial specification

$$\begin{aligned} \mathcal{C}^{(1)} &\overset{S_1}{\leftarrow} (\mathcal{C}^{(i_{1,1})}, \dots, \mathcal{C}^{(i_{1,m_1})}) \\ \mathcal{C}^{(2)} &\overset{S_2}{\leftarrow} (\mathcal{C}^{(i_{2,1})}, \dots, \mathcal{C}^{(i_{2,m_2})}) \\ &\vdots \\ \mathcal{C}^{(N)} &\overset{S_N}{\leftarrow} (\mathcal{C}^{(i_{N,1})}, \dots, \mathcal{C}^{(i_{N,m_N})}) \end{aligned}$$

The *reliance graph* of the specification is an infinite directed graph whose vertices are the sets $\mathcal{C}_i^{(j)}$ for $1 \le j \le N$ and $i \in \mathbb{N}$. We draw an edge from a vertex $\mathcal{C}_i^{(j)}$ to $\mathcal{C}_{i'}^{(j')}$ if there is k such that $\mathcal{C}^{(j')} = \mathcal{C}^{(i_{j,k})}$ and $i' \le r_{S_j}^{(k)}(i)$. In other words, there is an edge if $\mathcal{C}^{(j')}$ is on the right-hand side of the rule that has $\mathcal{C}^{(j)}$ on the left-hand side and the reliance function of S_j says that the count of $\mathcal{C}_i^{(j)}$ relies of $\mathcal{C}_{i'}^{(j')}$. The reliance graph perfectly encodes which sizes of which combinatorial sets are needed to compute the counts of each size of each combinatorial set. Therefore, if the reliance graph of a specification has no infinite directed walk the specification can be used to compute the counts as the procedure described above will terminate. In fact, having no infinite directed walk guarantees that the specification uniquely determines the counting sequence of each combinatorial set it contains. This fact is proved by Albert, Bean, Claesson, Nadeau, Pantone, Ulfarsson [34] as Theorem 4.1. Such a specification is called *productive*.

Since not every specification is productive, we have to be careful with the prune method since the output could contain trivial specifications. The way Albert, Bean, Claesson, Nadeau, Pantone and Ulfarsson [34] achieve that is by ensuring that any combinatorial specification contained in the universe is productive. They do so by placing a restriction on the strategies used.

Definition 2.3 (Definition 4.2 in [34]). *We call an m-ary strategy S a* productive strategy *if the following two conditions hold for all combinatorial sets $\mathcal{A}$ with corresponding decomposition $d_S(\mathcal{A}) = (\mathcal{B}^{(1)}, \dots, \mathcal{B}^{(m)})$, and for all $i \in \{1, \dots, m\}$.*

1. *For all $N \in \mathbb{N}$, if $\mathcal{A}_N$ relies on $\mathcal{B}_j^{(i)}$, then $j \le N$.*
2. *If $\mathcal{A}_N$ relies on $\mathcal{B}_N^{(i)}$ for some $N \in \mathbb{N}$, then*
   (a) *$|\mathcal{A}_n| \ge |\mathcal{B}_n^{(i)}|$ for all $n \in \mathbb{N}$, and*
   (b) *$|\mathcal{A}_\ell| > |\mathcal{B}_\ell^{(i)}|$ for some $\ell \in \mathbb{N}$.*

As before, we use the phrase "$\mathcal{A}_n$ relies on $\mathcal{B}_j^{(i)}$" or the diagram $\mathcal{A}_n \to \mathcal{B}_j^{(i)}$ as a simplified way of stating the formal information contained in the reliance profile function of S, namely that $j \le r_S^{(i)}(n)$.

Informally, it means that for any given rule $\mathcal{A} \xleftarrow{S} (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(m)})$ if the count of $\mathcal{B}_j^{(i)}$ is needed to compute the count of $\mathcal{A}_n$ then one of the two following conditions must be satisfied. Either the size of the objects in $\mathcal{B}_j^{(i)}$ are smaller than those in $\mathcal{A}_n$ (*i.e.*, $j < n$) or, if $j = n$ then there are always fewer objects of a given size in $\mathcal{B}^{(i)}$ than in $\mathcal{A}$. This inequality must be strict for at least one size.

Rather surprisingly, this restriction on a strategy that is local to the rule guarantees that any combinatorial specification using these rules will be productive. Albert, Bean, Claesson, Nadeau, Pantone and Ufarsson proved that specification where all the rules are come from productive strategies is productive.

Theorem 2.4 (Theorem 4.2 in [34])**.** *Let P be a proof tree, or the equivalent combinatorial specification, composed entirely of rules derived from productive strategies. Then the reliance graph of P has no infinite directed walks.*

Therefore, in a universe where all the rules are produced by productive strategies, the prune method returns the union of all productive specifications since all specifications are guaranteed to be productive.

It is worth noting that one very useful type of strategy is not considered productive. Those are the unary strategies that identify that a set $\mathcal{A}$ is equinumerous to a set $\mathcal{B}$. For such a strategy the set $\mathcal{A}_n$ relies on $\mathcal{B}_n$ and the number of elements of each size is the same in both sets. Such strategies are called *equivalence strategies*. They do not satisfy part 2b of the definition of a productive strategy. An example of an equivalence strategy can be found in the combinatorial specification of Figure 1.6 when we observe that $\text{Grid}(\mathcal{T}^{(6)})$ is equinumerous to $\text{Grid}(\mathcal{T}^{(7)})$ using the row separation strategy. Equivalence strategies are given special treatment that involves collapsing together into an *equivalence class* combinatorial sets that are discovered to be equinumerous by strategies. The prune method must then be run with respect to rules of those equivalence classes to ensure productivity. More details on how to handle equivalence strategies with the prune method are given in Section 4.3 of [34]. Reliance graphs and productive strategies are covered in greater details in Section 4.1 and 4.2 of the same paper.

The theory of productivity was a great leap forward in the development of combinatorial exploration as it allowed, with a local condition on the strategies, to ensure that any combinatorial specification built from it would be productive. In the next section, we will however study an example of a specification that is productive but where some of the rules could not be built from productive strategies. The rest of the chapter will be dedicated to building the theory to support a new algorithm that is more powerful than the prune method introduced in this section. This algorithm will allow us to work with strategies that do not satisfy the productivity condition.

2.2 Limitation of the prune method

At the root of the material presented in this chapter was an unexpected discovery in some of the universes of rules produced by combinatorial exploration. Often, the searcher builds universes that do not contain a combinatorial specification including the starting combinatorial set. This is easy to verify since, if there is a specification in the universe, then the prune method is guaranteed to find it. We however discovered that, if we converted all the rules of the universe into generating function equations, this system could sometimes be solved to find the generating function of the combinatorial set of interest.

Those examples were discovered early on by building some small universes of rules and feeding the corresponding system of equations to Maple. This method allows us to enumerate combinatorial sets but has two major drawbacks. First, it does not give any structural information on the combinatorial set. When a combinatorial specification is found, it consists of a specific set of decomposition rules describing how to break down your combinatorial set of interest into other combinatorial sets in a tree form. You can then present this decomposition visually as we have done in the previous section. This decomposition often allows us to do more than get the generating function. It describes a polynomial-time algorithm to count, generate the objects in the combinatorial set and also sample the combinatorial set uniformly at random. When you are just solving the universe of rules you do not get such a structural decomposition since all the information is just fed into a solver that cannot explain how it comes up with a solution. Second, this method is limited to small universes. Since this method requires us to solve a system of equations with the same number of equations as the number of rules in the universe it does not scale to bigger universes. While a typical universe can contains hundreds of thousands of rules and can be handle without issue by the prune method, solving the system corresponding to the universe is limited to only a couple of thousand of rules before the system becomes absolutely unmanageable for symbolic computation software.

Nonetheless, using this method on small universes provided indications that there was something more that needed to be investigated in those universes. This sparked our interest in understanding what was going on that allowed us to compute the counting sequences of some combinatorial sets even if they were not in a specification. We proceeded to build a pair of partial combinatorial specifications (in the sense that some of the combinatorial sets were not on the left-hand side of any of the rules). The method was quite ad hoc but it found pairs of partial specifications where the joint systems contained enough information to enumerate the combinatorial set of interest. These pairs were slightly closer to qualifying as a structural description of the combinatorial set but it was still unclear how they could be used for anything else other than solving the joint systems of equations.

One such pair of specifications appears in the conclusion of Bean's doctoral thesis [33] inspired by Claesson [40]. It is a pair of partial specifications for the permutation class

$$\mathrm{Av}(1234, 1243, 1324, 1423, 2134, 2314)$$

for which the combined system of equations solves for the generating function of the permutation class. The pair of specifications appears at Figure 2.1. The reader does not need to pay attention to all the details of the specifications but it is good to observe that the combinatorial set $\mathcal{T}^{(6)}$ appears in both specifications on the right-hand side of a rule but never on the left-hand side of any rule. It is at that point in time that the name forest came into use as that having two trees is obviously enough to qualify as a "forest". However, as we will soon see, the reasoning for this name quickly stopped making sense as the theory developed.

At this point, the main problem we were trying to solve was to find a method that could figure out whether a universe contained sufficient information to enumerate the original combinatorial set even if the universe contained no combinatorial specification for it. In other words, we wanted to obtain the same answer as we get when solving the equations for the entire universe but with a method scalable to much bigger universes. There were many iterations of various theories and code prototypes that we worked on. Until, at some point, we came to a realization that turned out to be central to our theory. We realized that we could sometimes get information from the children of a rule if we knew information about the parent instead of the other way around.

For example, lets take a look at the rule $\mathcal{T}^{(0)} \leftarrow (\mathcal{T}^{(10)}, \mathcal{T}^{(9)})$ from Figure 2.1. The rule is created by a cell insertion strategy where the gridded permutation $(1,(0,0))$ is inserted. The generating function equation for that rule is $T_0(x) = T_{10}(x) + T_9(x)$. Equivalently, we have that $T_9(x) = T_0(x) - T_{10}(x)$. We will formalize the strategy later but we can for now interpret the rearranged equation as the generating function equation for a strategy that would create the rule $\mathcal{T}^{(9)} \leftarrow (\mathcal{T}^{(0)}, \mathcal{T}^{(10)})$.

In a similar fashion, we can consider $T_8(x) = T_5(x) \cdot T_7(x)$, the generating function equation for the rule $\mathcal{T}^{(8)} \leftarrow (\mathcal{T}^{(5)}, \mathcal{T}^{(7)})$. We can rearrange it to get $T_7(x) = \frac{T_8(x)}{T_5(x)}$, that can, as previously, be thought of as the generating function equation of a rule $\mathcal{T}^{(7)} \leftarrow (\mathcal{T}^{(8)}, \mathcal{T}^{(5)})$.

The true power of this way of thinking reveals itself when we use it to rearrange the two partial specification of Figure 2.1 into a single complete specification. Consider the rule $\mathcal{T}^{(7)} \leftarrow (\mathcal{T}^{(6)}, \mathcal{T}^{(11)})$ in the second tree. We can flip it into the rule $\mathcal{T}^{(6)} \leftarrow (\mathcal{T}^{(7)}, \mathcal{T}^{(11)})$ and glue it under $\mathcal{T}^{(6)}$ in the first specification. The first specification is still incomplete but it now misses a rule for $\mathcal{T}^{(7)}$ and $\mathcal{T}^{(11)}$ instead of a rule for $\mathcal{T}^{(6)}$. We can take all the rules under $\mathcal{T}^{(11)}$ in the second tree and glue it under $\mathcal{T}^{(11)}$ in the first tree. Now, the partial specification is only missing a rule for $\mathcal{T}^{(7)}$. We can therefore glue the rule for $\mathcal{T}^{(7)}$ we derived by flipping the rule $\mathcal{T}^{(8)} \leftarrow (\mathcal{T}^{(5)}, \mathcal{T}^{(7)})$. The specification now misses a rule for $\mathcal{T}^{(8)}$. Since from the second tree we know that $\mathcal{T}^{(8)}$ and $\mathcal{T}^{(9)}$ are equivalent, we can add the rule $\mathcal{T}^{(8)} \leftarrow (\mathcal{T}^{(9)})$. Finally, we add the rule $\mathcal{T}^{(9)} \leftarrow (\mathcal{T}^{(0)}, \mathcal{T}^{(10)})$. If we as well transfer all the verification rules, the new tree is now a complete specification in the sense that each combinatorial set in the specification is now the left-hand side of exactly one rule. The full specification can be seen in Figure 2.2. This shows that even though we did not find a specification in the universe, there was actually one if we also considered the flipped versions of the rules in the universe as part of it.

It is, however, a risky business to also consider the flipped version of rules as part of the universe. As discussed earlier, some combinatorial specifications convey no meaningful enumerative information. The method described earlier

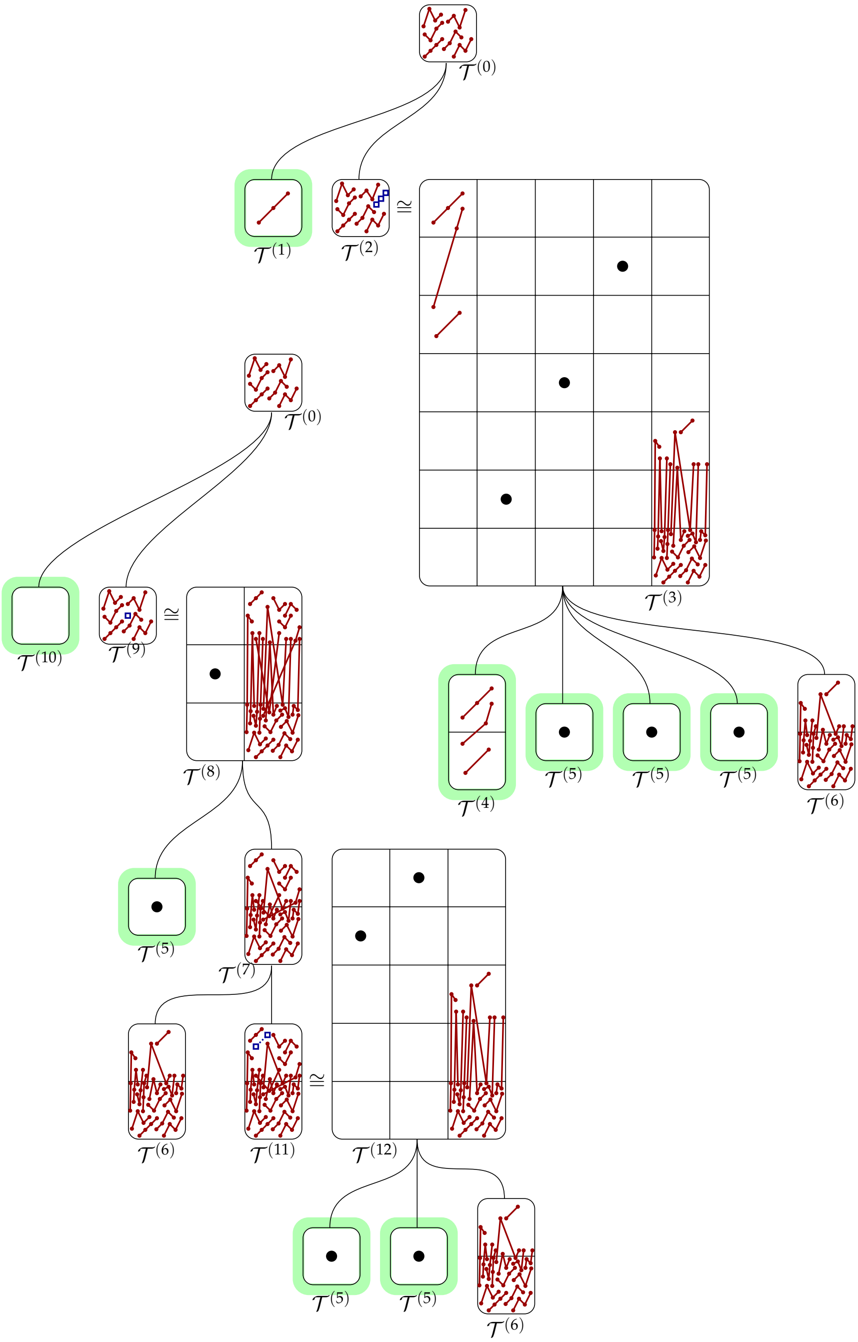


Figure 2.1: Two incomplete combinatorial specifications which together contain sufficient information to compute the counting sequence of the tiling $\mathcal{T}^{(0)}$.

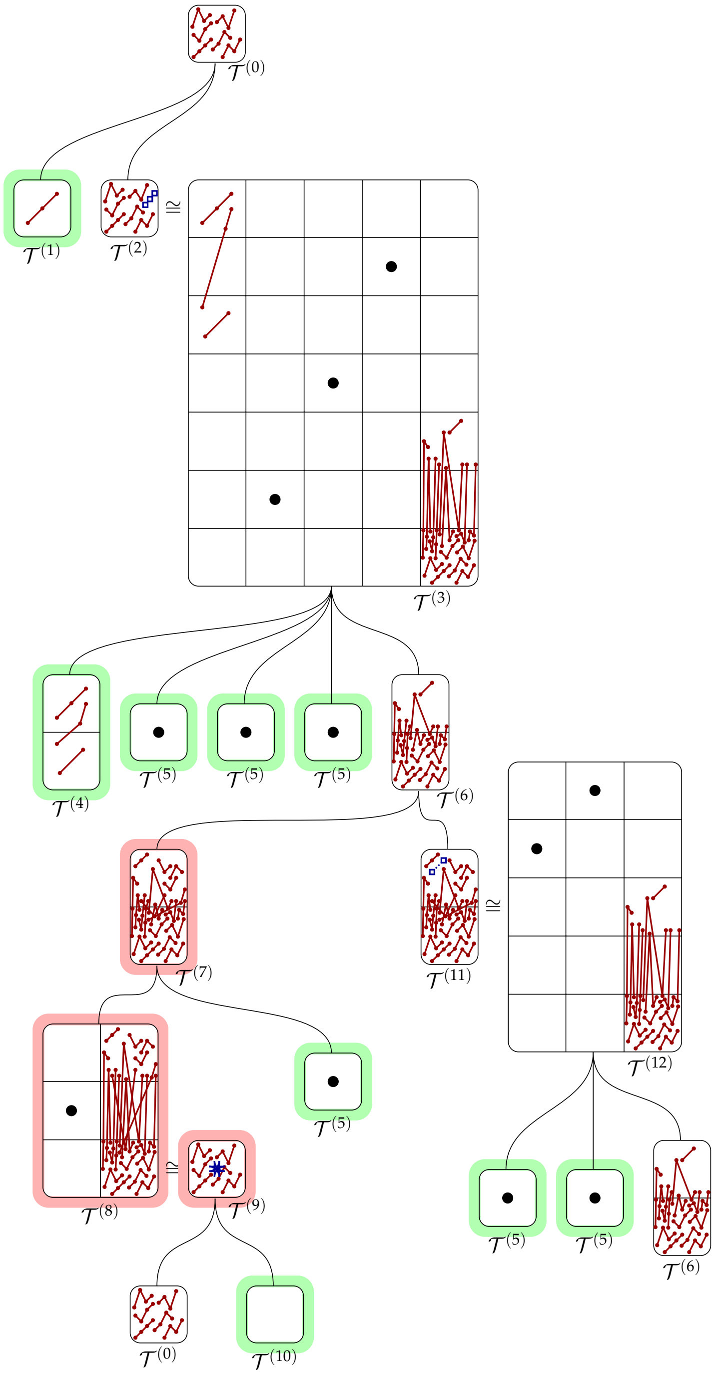


Figure 2.2: A productive combinatorial specification using "flipped" rules. Tilings in green are verified while red is used to highlight flipped rules.

guarantees that you always find a productive specification as long as the strategy used to create the rules are productive. When we allow ourself to rearrange the rules like above, the productivity condition is broken. In fact, we can create a specification for any combinatorial set almost instantly. Start by decomposing the combinatorial set of interest using the rule is-empty-or-not. Then rearrange this rule like we did above for the rule $\mathcal{T}^{(0)} \leftarrow (\mathcal{T}^{(10)}, \mathcal{T}^{(9)})$. Voilà! You have a specification with two rules where the corresponding system of equations is

$$F_1(x) = 1 + F_2(x)$$
$$F_2(x) = F_1(x) - 1.$$

This of course contains no enumerative information.

In the case of the combinatorial specification of Figure 2.2, the fact that the system solves for all the variables can give good hope that the specification actually contains meaningful enumerative information. Let us however convince ourself even further by using Theorem 4.1 from [34] that states that a specification is productive (*i.e.*, we can use it to count the combinatorial sets it contains) if its reliance graph contains no infinite directed walks.

To describe the reliance graph of the specification we need to know the reliance function of each of the strategies. As we have not yet formally defined the strategies used for flipped rules, we will, again stay mostly informal and derive the edges that need to go into the reliance graph by looking at the counting formulas the rules have. All the verification rules have no children and therefore no arrow goes out of the vertices for those combinatorial sets in the reliance graph. Hence, for any verified combinatorial set $\mathcal{T}^{(i)}$ there are no outgoing edges from any of the vertices $\mathcal{T}_j^{(i)}$. Since they have no outgoing edges, such vertices could never be part of an infinite directed walk and can therefore mostly be ignored for the purpose of applying Theorem 4.1. For all the rules coming from disjoint-union type strategies, we know from [34] that each component of the reliance function output for size n is n. For example, rule $\mathcal{T}^{(0)} \leftarrow (\mathcal{T}^{(1)}, \mathcal{T}^{(2)})$ has a reliance function $n \mapsto (n, n)$. Hence, the edges $\mathcal{T}_n^{(0)} \to \mathcal{T}_i^{(1)}$ and $\mathcal{T}_n^{(0)} \to \mathcal{T}_i^{(2)}$ are all in the reliance graph for any $0 \leq i \leq n$.

For the flipped version of disjoint-union type rules like the rule for $\mathcal{T}^{(6)}$ in Figure 2.2, the counting formula is $|\mathcal{T}_n^{(6)}| = |\mathcal{T}_n^{(7)}| - |\mathcal{T}_n^{(11)}|$. Hence, the reliance function for such a rule is also $n \mapsto (n, n)$ and the edges in the reliance graph are $\mathcal{T}_n^{(6)} \to \mathcal{T}_i^{(7)}$ and $\mathcal{T}_n^{(6)} \to \mathcal{T}_i^{(11)}$ for $0 \leq i \leq n$.

Consider the factorization rule $\mathcal{T}^{(3)} \leftarrow (\mathcal{T}^{(4)}, \mathcal{T}^{(5)}, \mathcal{T}^{(5)}, \mathcal{T}^{(5)}, \mathcal{T}^{(6)})$. According to the definition in Section 6.3.5 of Albert et al. [34], the reliance function for the strategy giving that rule is $n \mapsto (n-1, n-1, n-1, n-1, n-1)$ since $\mathcal{T}^{(5)}$ contains no gridded permutation of size 0. The counting formula is

$$|\mathcal{T}_n^{(3)}| = \sum_{\substack{i_1+i_2+i_3+i_4+i_5=n \\ 0 \leq i_1, i_2, i_3, i_4, i_5 < n}} |\mathcal{T}_{i_1}^{(4)}||\mathcal{T}_{i_2}^{(5)}||\mathcal{T}_{i_3}^{(5)}||\mathcal{T}_{i_4}^{(5)}||\mathcal{T}_{i_5}^{(6)}|.$$

However, we can observe that if i_5 is $n-1$ or $n-2$ then at least one of i_2, i_3 or i_4 needs to be zero and therefore the whole term becomes zero. Hence, we can restrict the counting formula to $i_5 \leq n-3$. Similar reasoning shows that we can restrict to $i_1 \leq n-3$ and $i_2, i_3, i_4 \leq n-2$. Hence, the counting formula becomes

$$|\mathcal{T}_n^{(3)}| = \sum_{\substack{i_1+i_2+i_3+i_4+i_5=n \\ 0 \leq i_2, i_3, i_4 \leq n-2 \\ 0 \leq i_1, i_5 \leq n-3}} |\mathcal{T}_{i_1}^{(4)}||\mathcal{T}_{i_2}^{(5)}||\mathcal{T}_{i_3}^{(5)}||\mathcal{T}_{i_4}^{(5)}||\mathcal{T}_{i_5}^{(6)}|$$

and we obtain a slightly more restrictive reliance function

$$n \mapsto (n-3, n-2, n-2, n-2, n-3).$$

This restriction will prove important when we draw the reliance graph and assert the productivity of the specification in Figure 2.2. We will formalize in Section 2.6.2 the definition of this more restrictive reliance function for the factor strategy. For now we will simply add that the reliance function for the $\mathcal{T}^{(12)} \leftarrow (\mathcal{T}^{(5)}, \mathcal{T}^{(5)}, \mathcal{T}^{(6)})$ can be restricted to $n \mapsto (n-1, n-1, n-2)$ using similar reasoning as we did with the other factorization rule.

Finally, let us consider the reliance of the flipped factorization rule $\mathcal{T}^{(7)} \leftarrow (\mathcal{T}^{(8)}, \mathcal{T}^{(5)})$. First, consider the regular factorization rule $\mathcal{T}^{(8)} \leftarrow (\mathcal{T}^{(7)}, \mathcal{T}^{(5)})$. The reliance function is $n \mapsto (n-1, n)$ and the counting formula is

$$|\mathcal{T}_n^{(8)}| = \sum_{i=0}^{n-1} |\mathcal{T}_i^{(7)}||\mathcal{T}_{n-i}^{(5)}|.$$

If we take this formula for $|\mathcal{T}_{n+1}^{(8)}|$, we get

$$|\mathcal{T}_{n+1}^{(8)}| = \sum_{i=0}^{n} |\mathcal{T}_i^{(7)}||\mathcal{T}_{n+1-i}^{(5)}|.$$

We then isolate $|\mathcal{T}_n^{(7)}|$ to obtain

$$|\mathcal{T}_n^{(7)}| = \frac{|\mathcal{T}_{n+1}^{(8)}| - \sum_{i=0}^{n-1} |\mathcal{T}_i^{(7)}||\mathcal{T}_{n+1-i}^{(5)}|}{|\mathcal{T}_1^{(5)}|}.$$

This is the counting formula needed for the flipped rule $\mathcal{T}^{(7)} \leftarrow (\mathcal{T}^{(8)}, \mathcal{T}^{(5)})$. However, as we can see the counting also relies on $\mathcal{T}^{(7)}$ itself but only for shorter sizes then the size we are trying to compute. This dependency on itself did not appear previously when we were only considering the generating function equation. It is in fact more correct to consider the flipped rule as $\mathcal{T}^{(7)} \leftarrow (\mathcal{T}^{(8)}, \mathcal{T}^{(5)}, \mathcal{T}^{(7)})$ where the reliance profile would be $n \mapsto (n+1, n+1, n-1)$.

Finally the reliance function of the three equivalence rules $\mathcal{T}^{(2)} \leftarrow (\mathcal{T}^{(3)})$, $\mathcal{T}^{(8)} \leftarrow (\mathcal{T}^{(9)})$ and $\mathcal{T}^{(11)} \leftarrow (\mathcal{T}^{(12)})$ is $n \mapsto (n)$.

We now have all the information we need to draw the reliance graph of this specification. For readability we will make a few simplifications. First, we do not draw the vertices for verified combinatorial sets. As mentioned previously, those vertices do not have any outgoing edges and are therefore irrelevant in the context of proving productivity using Theorem 4.1. Second, we only draw the edge going to the biggest size for each child. For example the rule $\mathcal{T}^{(0)} \leftarrow (\mathcal{T}^{(1)}, \mathcal{T}^{(2)})$ would imply the edges $(\mathcal{T}_n^{(0)}, \mathcal{T}_0^{(2)}), (\mathcal{T}_n^{(0)}, \mathcal{T}_1^{(2)}), \ldots, (\mathcal{T}_n^{(0)}, \mathcal{T}_n^{(2)})$ but we only draw $(\mathcal{T}_n^{(0)}, \mathcal{T}_n^{(2)})$. The reliance graph is partially pictured in Figure 2.3.

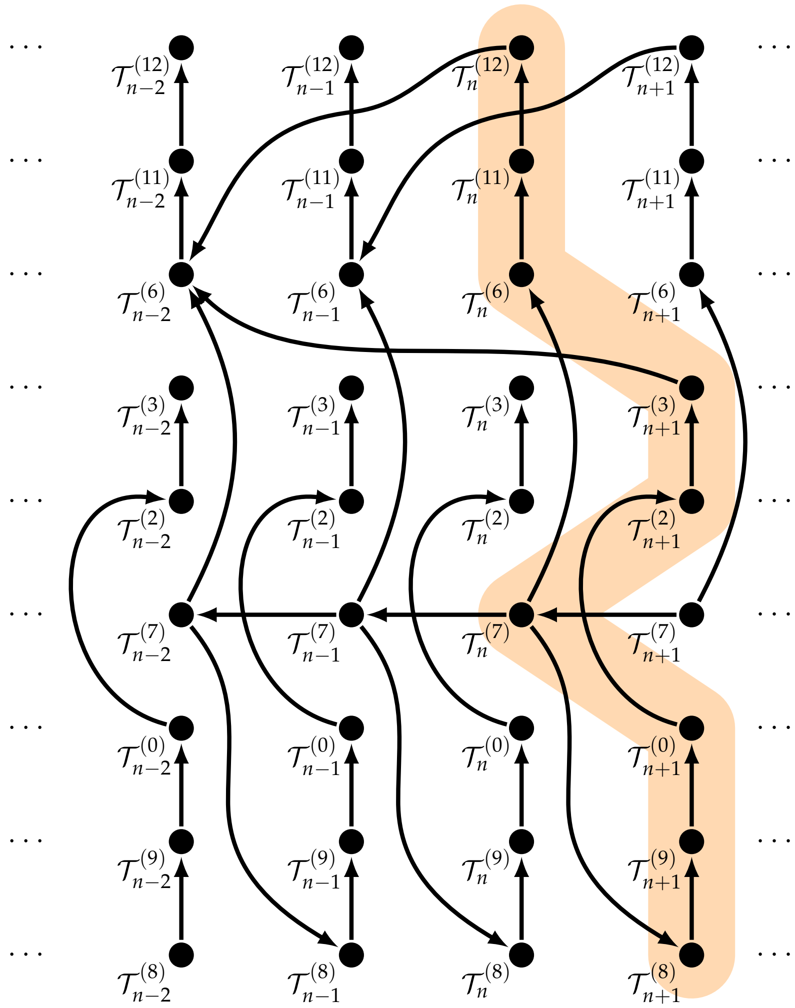


Figure 2.3: Reliance graph of the combinatorial specification in Figure 2.2.

To confirm that the specification is productive we observe that the reliance graph contains no infinite directed walk. To do so, we first partition the vertices of the reliance graph into slices S_n where

$$S_n = \{\mathcal{T}_n^{(12)}, \mathcal{T}_n^{(11)}, \mathcal{T}_n^{(6)}, \mathcal{T}_n^{(7)}, \mathcal{T}_{n+1}^{(3)}, \mathcal{T}_{n+1}^{(2)}, \mathcal{T}_{n+1}^{(0)}, \mathcal{T}_{n+1}^{(9)}, \mathcal{T}_{n+1}^{(8)}\}$$

for $n \in \mathbb{N}$. A slice S_n contains one $\mathcal{T}_j^{(i)}$ for each of the $\mathcal{T}^{(i)}$ in the specification. A slice is highlighted in orange in Figure 2.3. One key characteristic of those slices (and the reason why they were defined like that) is that all edges originating from a vertex in a slice S_n must end in the same slice S_n or in a slice S_i for $i \leq n$. This can be easily checked by looking at the reliance function that we have described or by inspection of Figure 2.3. One can also easily verify that there is no cycle fully contained in a single slice.

We prove by contradiction that the graph cannot contain an infinite directed walk. Suppose an infinite directed walk existed in the reliance graph. Let n be minimal such that an infinite walk starts in S_n and consider an infinite walk starting in S_n. The walk can only have finitely many step in S_n since S_n is finite and contains no cycle. Therefore, the walk eventually uses node in S_m with $m < n$. This means there is an infinite walk starting in S_m contradicting the minimality of n. Therefore, the reliance graph has no infinite directed walk and the specification is productive.

Observing, like we have done in this very long example, that rules can be used in reverse is really the key observation the reader should take from this chapter. It is an observation that may seem like the obvious way *a posteriori* but was really a breakthrough moment in the development of the theory. As we have seen, the new flipped rules can not come from productive strategies so most of the theory of Section 2.1 needs to be rethought. The following sections of this chapter are dedicated to formalizing many of the approximations and hand-waving that took place in this section and developing an alternative to the prune method that can find specifications like the one of Figure 2.2.

2.3 Enumerable subset

What truly determines what can be counted in the universe is the reliance profiles of the strategies. To verify if the term of the counting sequence for a given size and combinatorial set in the universe can be computed, we only need to use the reliance profile function. For a combinatorial set $\mathcal{C}$ in the universe, the term $|\mathcal{C}_n|$ can be computed from the rules in the universe if there is a rule $\mathcal{C} \overset{S}{\leftarrow} (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(k)})$ with a reliance function r_S such that all the terms of the form $|\mathcal{B}_j^{(i)}|$ that are requested by r_S can also be computed from the rules in the universe. Formally, it means that, for all $1 \leq i \leq k$ and $0 \leq j_i \leq r_S^{(i)}(n)$, the term $|\mathcal{B}_{j_i}^{(i)}|$ can be computed from the rules in the universe. Though this recursive way of thinking gives a good intuition, it is unfortunately not very useful in practice as you can quickly fall into infinite recursion. Imagine you have a rule $\mathcal{A} \overset{S_1}{\leftarrow} (\mathcal{B})$ with a reliance function $r_{S_1}(n) = (n)$ and a rule $\mathcal{B} \overset{S_2}{\leftarrow} (\mathcal{A})$ with the same reliance function. Then, if you were trying to verify if the term $|\mathcal{A}_5|$ could be computed from the information in the universe, the procedure outlined above would check if $|\mathcal{B}_5|$ could be computed which would in turn check if $|\mathcal{A}_5|$ could be computed and so on. To resolve this issue we adopt here an approach from the ground up (more akin to dynamic programming).

Let U be a finite universe of combinatorial rules. We say that a combinatorial set $\mathcal{C}$ is *in* U if it is contained in any of its rules. The set of combinatorial sets contained in a universe U is denoted $C(U)$. We define a sequence of functions $\mathfrak{f}_i^{(U)} : C(U) \to \mathbb{N}$. The reader should interpret $\mathfrak{f}_i^{(U)}(\mathcal{C}) = k$ as follows: after i steps, the first k terms of the counting sequence of $\mathcal{C}$ are known, *i.e.*, the terms for size 0 to size $k-1$ of the input combinatorial set have been found. Formally, the sequence $\mathfrak{f}_0^{(U)}, \mathfrak{f}_1^{(U)}, \mathfrak{f}_2^{(U)}, \ldots$ is defined as

- $\mathfrak{f}_0^{(U)}(\mathcal{C}) = 0$ for all $\mathcal{C} \in U$
- $\mathfrak{f}_{i+1}^{(U)}(\mathcal{C}) = \mathfrak{f}_i^{(U)}(\mathcal{C}) + 1$ if the universe U contains a rule $\mathcal{C} \overset{S}{\leftarrow} (\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(k)})$ such that $r_S^{(j)}(\mathfrak{f}_i^{(U)}(\mathcal{C})) < \mathfrak{f}_i^{(U)}(\mathcal{A}^{(j)})$ for $1 \le j \le k$. Otherwise, $\mathfrak{f}_{i+1}^{(U)}(\mathcal{C}) = \mathfrak{f}_i^{(U)}(\mathcal{C})$.

When a rule $\mathcal{C} \leftarrow S(\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(k)})$ satisfies the condition for incrementing in the definition above, *i.e.*, when $r_S^{(j)}(\mathfrak{f}_i^{(U)}(\mathcal{C})) < \mathfrak{f}_i^{(U)}(\mathcal{A}^{(j)})$ for all $1 \le j \le k$, we say the rule satisfies the *increment condition* for $\mathfrak{f}_i^{(U)}$. The construction of the $\mathfrak{f}_i^{(U)}$ should be seen as an iterative process to compute more and more terms of the counting sequence of the combinatorial sets in the universe. The function $\mathfrak{f}_i^{(U)}$ indicating the number of terms known at step i of this process. We start without any information about the terms. Hence, $\mathfrak{f}_0^{(U)}$ is always 0. To compute $\mathfrak{f}_{i+1}^{(U)}$, we look at the information we have about the terms at the previous step, *i.e.*, the function $\mathfrak{f}_i^{(U)}$. If we have enough information from step i to use a rule to compute one extra term for a given combinatorial set then the value of the function $\mathfrak{f}_{i+1}^{(U)}$ for that combinatorial set increases by one from its value for $\mathfrak{f}_i^{(U)}$. Otherwise, if $\mathfrak{f}_i^{(U)}$ does not provide enough information to compute a new term of the counting sequence using any of the rules for the given combinatorial set then the value of $\mathfrak{f}_{i+1}^{(U)}$ stays unchanged from $\mathfrak{f}_i^{(U)}$ for that combinatorial set. Computing the function $\mathfrak{f}_{i+1}^{(U)}$ can be done from $\mathfrak{f}_i^{(U)}$ by making one pass through the rules of the universe where we check whether each rule satisfies the increment condition for $\mathfrak{f}_i^{(U)}$.

Let build an example of the functions $\mathfrak{f}_i^{(U)}$ for a small universe. We consider the universe U consisting of the rules in Table 2.1. The functions $\mathfrak{f}_0^{(U)}$ to $\mathfrak{f}_7^{(U)}$ are given in Table 2.2. The base case gives us that $\mathfrak{f}_0^{(U)}$ is 0 for all the combinatorial sets. Let look in detail how the function $\mathfrak{f}_1^{(U)}$ is derived. The rule $\mathcal{G} \leftarrow (\mathcal{E})$ is the only rule of U with $\mathcal{G}$ as a parent. We have $\mathfrak{f}_0^{(U)}(\mathcal{G}) = 0$ which is mapped to (-2) by the reliance function. Moreover, the rule fulfills the increment condition for $\mathfrak{f}_0^{(U)}$ since $-2 < 0 = \mathfrak{f}_0^{(U)}(\mathcal{E})$. Hence, $\mathfrak{f}_1^{(U)}(\mathcal{G}) = 1$. Let look at the combinatorial set $\mathcal{C}$. The two rules that have $\mathcal{C}$ as a parent have $\mathcal{D}$ as their second child. In both cases the second component of reliance function maps 0 to 0. Hence, since $\mathfrak{f}_0^{(U)}(\mathcal{C}) = 0$

Parent	Rule	Reliance function
$\mathcal{A}$	$\mathcal{A} \leftarrow (\mathcal{B}, \mathcal{C})$	$n \mapsto (n, n)$
$\mathcal{B}$	$\mathcal{B} \leftarrow ()$	$n \mapsto ()$
$\mathcal{C}$	$\mathcal{C} \leftarrow (\mathcal{A}, \mathcal{D})$	$n \mapsto (n-1, n)$
$\mathcal{C}$	$\mathcal{C} \leftarrow (\mathcal{E}, \mathcal{D})$	$n \mapsto (n-1, n)$
$\mathcal{D}$	$\mathcal{D} \leftarrow ()$	$n \mapsto ()$
$\mathcal{F}$	$\mathcal{F} \leftarrow (\mathcal{C}, \mathcal{A})$	$n \mapsto (n, n+1)$
$\mathcal{G}$	$\mathcal{G} \leftarrow (\mathcal{E})$	$n \mapsto (n-2)$

Table 2.1: Example of the rules in a small universe with their reliance functions.

Combinatorial set	$\mathfrak{f}_0^{(U)}$	$\mathfrak{f}_1^{(U)}$	$\mathfrak{f}_2^{(U)}$	$\mathfrak{f}_3^{(U)}$	$\mathfrak{f}_4^{(U)}$	$\mathfrak{f}_5^{(U)}$	$\mathfrak{f}_6^{(U)}$	$\mathfrak{f}_7^{(U)}$
$\mathcal{A}$	0	0	0	1	1	2	2	3
$\mathcal{B}$	0	1	2	3	4	5	6	7
$\mathcal{C}$	0	0	1	1	2	2	3	3
$\mathcal{D}$	0	1	2	3	4	5	6	7
$\mathcal{E}$	0	0	0	0	0	0	0	0
$\mathcal{F}$	0	0	0	0	0	0	1	1
$\mathcal{G}$	0	1	2	2	2	2	2	2

Table 2.2: The functions $\mathfrak{f}_0^{(U)}$ to $\mathfrak{f}_7^{(U)}$ for the universe U of Table 2.1.

and $\mathfrak{f}_0^{(U)}(\mathcal{D}) = 0$ the increment condition is not satisfied by any of those rules for $\mathfrak{f}_0^{(U)}$ and we have $\mathfrak{f}_1^{(U)}(\mathcal{C}) = 0$. A similar reasoning shows that none of the rules for the combinatorial sets $\mathcal{A}$, $\mathcal{E}$ and $\mathcal{F}$ satisfy the increment condition for $\mathfrak{f}_0^{(U)}$ and therefore $\mathfrak{f}_1^{(U)}(\mathcal{A}) = \mathfrak{f}_1^{(U)}(\mathcal{E}) = \mathfrak{f}_1^{(U)}(\mathcal{F}) = 0$. The combinatorial sets $\mathcal{B}$ and $\mathcal{D}$ have a verification rule, *i.e.*, a rule with no children. Hence, the increment condition is trivially satistfied for all $\mathfrak{f}_i^{(U)}$ giving us $\mathfrak{f}_i^{(U)}(\mathcal{B}) = \mathfrak{f}_i^{(U)}(\mathcal{D}) = i$ for any $i \in \mathbb{N}$ and in particular $\mathfrak{f}_1^{(U)}(\mathcal{B}) = \mathfrak{f}_1^{(U)}(\mathcal{D}) = 1$, completing the first column in Table 2.2.

We will not go into details on how each of the subsequent functions in the table is computed as the reasoning is extremely similar. We will however make a few important observations. The first is that since there is no rule with $\mathcal{E}$ as a parent we always have $\mathfrak{f}_i^{(U)}(\mathcal{E}) = 0$ for all $i \in \mathbb{N}$. Second, based on the previous observation and the fact that the only rule for $\mathcal{G}$ depends on $\mathcal{E}$ we can observe that $\mathfrak{f}_i^{(U)}(\mathcal{G}) = 2$ for $i \geq 2$. Finally, we observe that from $\mathfrak{f}_5^{(U)}$ onward, the rules $\mathcal{C} \leftarrow (\mathcal{A}, \mathcal{D})$ and $\mathcal{F} \leftarrow (\mathcal{C}, \mathcal{A})$ satisfy the increment condition for any function with odd index in the sequence while the rule $\mathcal{A} \leftarrow (\mathcal{B}, \mathcal{C})$ satisfies the increment condition for every function with even index in the sequence. Hence, the value of the function for $\mathcal{A}$, $\mathcal{C}$ and $\mathcal{F}$ will keep increasing as we move further in the function sequence. Informally, we can see that as an indication that the universe contains enough information to enumerate $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$ and $\mathcal{F}$ but not enough to enumerate $\mathcal{E}$ and $\mathcal{G}$.

To begin to formalize this distinction between combinatorial sets for which we can compute the counting sequence from the information in the universe and those for which we cannot, we first introduce the limit of the sequence $\{\mathfrak{f}_i^{(U)}\}_{i\geq 0}$. We define the function

$$\begin{aligned}\mathfrak{f}^{(U)}\colon C(U) &\to \mathbb{N}\cup\{\infty\}\\ \mathcal{C} &\mapsto \lim_{i\to\infty}\mathfrak{f}_i^{(U)}(\mathcal{C}).\end{aligned}$$

Since the sequence $\{\mathfrak{f}_i^{(U)}(\mathcal{C})\}_{i\geq 0}$ is weakly increasing for any $\mathcal{C}\in C(U)$, we have that the sequence either converges or diverges to ∞. Hence, the function $\mathfrak{f}^{(U)}$ is well defined. For the universe U in Table 2.1, we have $\mathfrak{f}^{(U)}(\mathcal{E})=0$ and $\mathfrak{f}^{(U)}(\mathcal{G})=2$ while $\mathfrak{f}^{(U)}(\mathcal{A})=\mathfrak{f}^{(U)}(\mathcal{B})=\mathfrak{f}^{(U)}(\mathcal{C})=\mathfrak{f}^{(U)}(\mathcal{D})=\mathfrak{f}^{(U)}(\mathcal{F})=\infty$.

Definition 2.5. *We define $E(U)$, the* enumerable subset *of U, as the preimage of ∞ for $\mathfrak{f}^{(U)}$.*

This set is called the enumerable subset since it is possible to compute the counting sequence of any combinatorial set in it.

Lemma 2.6. *For any combinatorial set $\mathcal{C}\in E(U)$ and any $n\in\mathbb{N}$, we can compute $|\mathcal{C}_n|$.*

Proof. We proceed by induction on i to show that we can compute the counting sequence of $\mathcal{C}$ up to size $\mathfrak{f}_i^{(U)}(\mathcal{C})-1$ for any $\mathcal{C}$ in U. The base case $i=0$, is trivial as $\mathfrak{f}_0^{(U)}$ is always 0. Assume that the statement holds for i, *i.e.*, that we can compute the counting sequence of $\mathcal{C}$ up to size $\mathfrak{f}_i^{(U)}(\mathcal{C})-1$ for any combinatorial set $\mathcal{C}$ in the enumerable subset of U. If $\mathfrak{f}_{i+1}^{(U)}(\mathcal{C}) = \mathfrak{f}_i^{(U)}(\mathcal{C})$, then by the hypothesis we can compute the counting sequence up to size $\mathfrak{f}_{i+1}^{(U)}(\mathcal{C})-1$. Otherwise, $\mathfrak{f}_{i+1}^{(U)}(\mathcal{C}) = \mathfrak{f}_i^{(U)}(\mathcal{C})+1$ and the universe contains a rule $\mathcal{C}\leftarrow S(\mathcal{A}^{(1)},\ldots,\mathcal{A}^{(k)})$ that satisfies the increment condition for $\mathfrak{f}_i^{(U)}$. We have that $r_S^{(j)}(\mathfrak{f}_i^{(U)}(\mathcal{C}))<\mathfrak{f}_i^{(U)}(\mathcal{A}^{(j)})$ for $1\leq j\leq k$. By the induction hypothesis, we can compute the counting sequence of $\mathcal{A}^{(j)}$ up to size $\mathfrak{f}_i^{(U)}(\mathcal{A}^{(j)})-1$ for $1\leq j\leq k$. Hence, we can use the counting function of the strategy S to compute the number of objects of size $\mathfrak{f}_{i+1}^{(U)}(\mathcal{C})-1$ in $\mathcal{C}$. By the induction hypothesis, we can compute the counting sequence of $\mathcal{C}$ up to size $\mathfrak{f}_i^{(U)}(\mathcal{C})-1=\mathfrak{f}_{i+1}^{(U)}(\mathcal{C})-2$. This concludes the induction.

Since $\mathcal{C}$ is in $E(U)$, $\lim_{i\to\infty}\mathfrak{f}_i^{(U)}(\mathcal{C})=\infty$ and we can find i such that $\mathfrak{f}_i^{(U)}(\mathcal{C}) = n+1$ for any $n\in\mathbb{N}$. Hence, it is possible to compute the full counting sequence of any combinatorial set in the enumerable subset. □

The previous lemma shows that any combinatorial set in the enumerable subset is enumerable using the rules of the universe. The natural question that arises from that is whether all the combinatorial sets that are enumerable with the information contained in the universe are in the enumerable subset. Though the answer is not a straightforward yes, it is nonetheless partially answered by the following theorem that gives a sufficient condition for a combinatorial set in a universe U to be in $E(U)$.

Theorem 2.7. *Consider a specification whose reliance graph contains no infinite directed walk. If all the rules of the specification are in a universe U, then all the combinatorial sets in the specification are in the enumerable subset of U.*

Before we can derive the proof of Theorem 2.7, we first need to introduce a new tool: the *augmented reliance graph*. Informally, an edge $\mathcal{C}_n \to \mathcal{B}_m$ in the reliance graph of a specification indicates that, according to the reliance profile of the rule for $\mathcal{C}$, you need to first get the number of objects of size m in $\mathcal{B}$ to be able to compute the number of objects size n in $\mathcal{C}$. To build the augmented reliance graph, we add all the edges of the form $\mathcal{C}_n \to \mathcal{C}_{n-1}$ to the regular reliance graph. Informally, these edges enforce that in order to compute the number of objects of size n for a combinatorial set $\mathcal{C}$ we must first compute the number of objects of smaller size for the same combinatorial set. We first observe that adding the edges to create the augmented version of the reliance graph cannot create infinite directed walks.

Lemma 2.8. *If the reliance graph of a specification has no infinite directed walk then the augmented reliance graph also has no infinite directed walk.*

Proof. We proceed by contradiction. Assume that we have a specification such that its reliance graph has no infinite directed walk but its augmented reliance graph does. We will show that from such walk we can always create another infinite walk with a longer prefix in the reliance graph by cutting and gluing back the walk in a clever way. We write the walk as a sequence of the visited vertices

$$\mathcal{A}^{(0)}_{n_0} \to \mathcal{A}^{(1)}_{n_1} \to \mathcal{A}^{(2)}_{n_2} \to \cdots .$$

Let k be the index of the first vertex in the walk where the outgoing edge of the walk is not the reliance graph. There is such an edge otherwise the infinite walk would be in the reliance graph. By construction, this edge must be of the form $\mathcal{A}^{(k)}_{n_k} \to \mathcal{A}^{(k)}_{n_k-1}$. Let j be the index of the first vertex after $\mathcal{A}^{(k)}_{n_k}$ in the walk where the outgoing edge is in the reliance graph. This vertex exists as if we only used edges not in the reliance graph we end up at $\mathcal{A}^{(k)}_0$ and $\mathcal{A}^{(k)}_0$ has no outgoing edge that are not in the reliance graph. Hence, the infinite path looks like

$$\mathcal{A}^{(0)}_{n_0} \to \cdots \to \mathcal{A}^{(k-1)}_{n_{k-1}} \to \mathcal{A}^{(k)}_{n_k} \to \mathcal{A}^{(k)}_{n_k-1} \to \cdots \to \mathcal{A}^{(k)}_{n_k-j} = \mathcal{A}^{(j)}_{n_j} \to \cdots$$

where $\mathcal{A}^{(k-1)}_{n_{k-1}} \to \mathcal{A}^{(k)}_{n_k}$ is an edge in the reliance graph. Since $n_k - j < n_k$, we have by definition that $\mathcal{A}^{(k-1)}_{n_{k-1}} \to \mathcal{A}^{(k)}_{n_k-j}$ is also an edge in the reliance graph. Hence,

$$\mathcal{A}^{(0)}_{n_0} \to \cdots \to \mathcal{A}^{(k-1)}_{n_{k-1}} \to \mathcal{A}^{(k)}_{n_k-j} = \mathcal{A}^{(j)}_{n_j} \to \cdots$$

is an infinite walk in the augmented reliance graph but with a longer prefix in the reliance graph. Repeating this process again and again on the same walk creates an infinite directed walk with no edge contained only in the augmented reliance graph. This is a contradiction to the reliance graph having no infinite directed walk. Therefore, the augmented reliance graph does not contain any infinite directed walk. □

Armed with this new graph, we define a new function "rank" that maps the vertices of the reliance graph of a specification to $\mathbb{N}$. Precisely, the function maps each vertex to the length of the longest path starting from it in the augmented reliance graph.

Lemma 2.9. *If the reliance graph of the specification has no infinite directed walk then* rank *is well defined.*

Proof. Since the reliance graph of the specification contains no infinite directed walk, the same goes for the augmented reliance graph by Lemma 2.8. Therefore, for rank to be well-defined we only need to have finitely many paths starting from each vertex in the augmented reliance graph. Suppose there are infinitely many walks starting at a vertex v_1. The outdegree of each vertex in the augmented reliance graph is finite. Hence, v_1 must have at least one child where there is also infinitely many walks starting from it. If we keep repeating the argument on the new vertex, we find a path $v_1, v_2, v_3, v_4, \ldots$ that is infinite. This is in contradiction to the augmented reliance graph not containing an infinite directed walk. Therefore rank is well-defined. □

Remark 2.10. *One important observation about the function is that if the augmented reliance graph contains an edge $\mathcal{A}_n \to \mathcal{B}_m$ then $\text{rank}(\mathcal{A}_n) > \text{rank}(\mathcal{B}_m)$. In particular, $\text{rank}(\mathcal{C}_n)$ is always greater than $\text{rank}(\mathcal{C}_{n-1})$ since there is an edge from $\mathcal{C}_n$ to $\mathcal{C}_{n-1}$.*

Recall that the edges of the reliance graph express the reliance for computing the number of objects in the vertices. Hence, the longer the longest path from a vertex is, the more steps are needed to be completed before the number can be computed. In that sense, we can think of $\text{rank}(\mathcal{C}_n)$ as a function giving the number of steps needed to compute the number of objects in $\mathcal{C}_n$. We formalize this intuition in the following lemma.

Lemma 2.11. *Consider a specification whose reliance graph contains no infinite directed walk. If all the rules of the specification are in a universe U, then*

$$\mathfrak{f}^{(U)}_{\text{rank}(\mathcal{C}_n)+1}(\mathcal{C}) \geq n+1$$

for any combinatorial set $\mathcal{C}$ in the specification and any $n \in \mathbb{N}$.

Proof. We proceed by induction on the value of $\text{rank}(\mathcal{C}_n)$.

If $\text{rank}(\mathcal{C}_n) = 0$, there is no edge going out of $\mathcal{C}_n$. In particular, we must have $n = 0$, otherwise there would be an edge $\mathcal{C}_n \to \mathcal{C}_{n-1}$. Consider $\mathcal{C} \overset{S}{\leftarrow} (\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(k)})$, the rule for $\mathcal{C}$ in the specification. Since no edge goes out of $\mathcal{C}_0$, it means $r_S^{(j)}(0) < 0$ for all $1 \leq j \leq k$ since the edges of the reliance graph reflect the reliance function. Hence,

$$\mathfrak{f}^{(U)}_{\text{rank}(\mathcal{C}_n)+1}(\mathcal{C}) = \mathfrak{f}^{(U)}_1(\mathcal{C}) = \mathfrak{f}^{(U)}_0(\mathcal{C}) + 1 = 1.$$

Let $K \in \mathbb{N}$ and assume that

$$\mathfrak{f}^{(U)}_{\text{rank}(\mathcal{C}_n)+1}(\mathcal{C}) \geq n + 1$$

for all $\mathcal{C}_n$ such that $\text{rank}(\mathcal{C}_n) < K$. Let $\mathcal{C}_n$ be such that $\text{rank}(\mathcal{C}_n) = K$. By Remark 2.10, we have that $\text{rank}(\mathcal{C}_{n-1}) < K$. Hence, the induction hypothesis gives

$$n \leq \mathfrak{f}^{(U)}_{\text{rank}(\mathcal{C}_{n-1})+1}(\mathcal{C}) \leq \mathfrak{f}^{(U)}_K(\mathcal{C}).$$

From there, we distinguish two cases. The first one is when $\mathfrak{f}^{(U)}_K(\mathcal{C}) > n$. In that case, we have directly $\mathfrak{f}^{(U)}_{K+1}(\mathcal{C}) \geq n + 1$ as desired. The second is when $\mathfrak{f}^{(U)}_K(\mathcal{C}) = n$. In this case, we use the induction hypothesis to show that the rule for $\mathcal{C}$ in the specification satisfies the increment condition for $\mathfrak{f}^{(U)}_K$. Consider $\mathcal{C} \overset{S}{\leftarrow} (\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(k)})$, the rule for $\mathcal{C}$ in the specification. Let $r_S(n) = (\alpha_1, \ldots, \alpha_k)$. By construction, the augmented reliance graph contains the edges $\mathcal{C}_n \to \mathcal{A}^{(j)}_{\alpha_j}$ for $1 \leq j \leq k$. Therefore, by Remark 2.10, we have that $\text{rank}(\mathcal{A}^{(j)}_{\alpha_j}) < \text{rank}(\mathcal{C}_n) = K$ which allows us to use the induction hypothesis to get that

$$\alpha_j + 1 \leq \mathfrak{f}^{(U)}_{\text{rank}(\mathcal{A}^{(j)}_{\alpha_j})+1}(\mathcal{A}^{(j)}) \leq \mathfrak{f}^{(U)}_K(\mathcal{A}^{(j)}).$$

Consequently, for $1 \leq j \leq k$, we have $r_S^{(j)}(\mathfrak{f}^{(U)}_K(\mathcal{C})) < \mathfrak{f}^{(U)}_K(\mathcal{A}^{(j)})$. In other words, the rule satisfies the increment condition for $\mathfrak{f}^{(U)}_K$ and $\mathfrak{f}^{(U)}_{K+1}(\mathcal{C}) = n + 1$. □

We now have all the tools to prove Theorem 2.7.

Proof of Theorem 2.7. Let $\mathcal{C}$ be a combinatorial set in the specification. By Lemma 2.11 we have

$$\mathfrak{f}^{(U)}_{\text{rank}(\mathcal{C}_n)+1}(\mathcal{C}) \geq n + 1.$$

for any $n \in \mathbb{N}$. Therefore, for each size $n \in \mathbb{N}$, there is a value of i such that $\mathfrak{f}^{(U)}_i(\mathcal{C}) > n$. In other words, since $\{\mathfrak{f}^{(U)}_i(\mathcal{C})\}_{i \geq 0}$ is increasing $\lim_{i \to \infty} \mathfrak{f}^{(U)}_i(\mathcal{C}) = \infty$. Hence, $\mathcal{C}$ is in the enumerable subset. □

As a consequence of this theorem, we have that if all the rules of the specification described in Section 2.2 were in a given universe then all the combinatorial sets from that specification would be in the enumerable subset of that universe.

It is also interesting to note that this theorem applies to all specifications that can be found with the theory developed in [34]. In fact, those specifications contain only rules derived from productive strategies and we know from Theorem 2.4, that the reliance graph of any such specification contains no infinite directed walk. In other words, all the specifications that could be found by the prune method (Algorithm 1) will be in the enumerable subset.

2.4 Regular strategy

We concluded the previous section by showing that any combinatorial set that was in a specification with a reliance graph that has no infinite directed walk would be in the enumerable subset of a universe containing it. We have outlined with Lemma 2.6 that the counting sequence of any combinatorial set in the enumerable subset could be derived from the counting functions of the rules in the universe. Does that mean that any combinatorial set in the enumerable subset is in a specification with a reliance graph containing no infinite directed walk? The answer is unfortunately no as we will show with the following example. Consider the universe consisting of the four rules presented in Table 2.3.

Parent	Rule	Reliance function
$\mathcal{A}$	$\mathcal{A} \leftarrow (\mathcal{B})$	$n \mapsto (n-1)$
$\mathcal{C}$	$\mathcal{C} \leftarrow (\mathcal{B})$	$n \mapsto (n-1)$
$\mathcal{B}$	$\mathcal{B} \leftarrow (\mathcal{A})$	$n \mapsto \begin{cases} n+1 \text{ if } n \text{ is even} \\ n \text{ otherwise} \end{cases}$
$\mathcal{B}$	$\mathcal{B} \leftarrow (\mathcal{C})$	$n \mapsto \begin{cases} n \text{ if } n \text{ is even} \\ n+1 \text{ otherwise} \end{cases}$

Table 2.3: A universe where elements of the enumerable subset are not in any specification.

Table 2.4 shows some of the functions $\mathfrak{f}_i^{(U)}$ for this universe. One important thing to notice is that only the rule $\mathcal{B} \leftarrow (\mathcal{C})$ satisfies the increment condition for $\mathfrak{f}_1^{(U)}$ while only the rule $\mathcal{B} \leftarrow (\mathcal{A})$ satisfies it for $\mathfrak{f}_3^{(U)}$. We observe that for any odd i we can increase the value of $\mathcal{A}$ and $\mathcal{B}$ while for every even i we can increase the value of $\mathcal{B}$ using alternately each of the two rules for that combinatorial set. We can therefore convince ourself that $\mathfrak{f}^{(U)} \equiv \infty$ and $E(U) = \{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$.

Combinatorial set	$\mathfrak{f}_0^{(U)}$	$\mathfrak{f}_1^{(U)}$	$\mathfrak{f}_2^{(U)}$	$\mathfrak{f}_3^{(U)}$	$\mathfrak{f}_4^{(U)}$	$\mathfrak{f}_5^{(U)}$	...	$\mathfrak{f}^{(U)}$
$\mathcal{A}$	0	1	1	2	2	3	...	∞
$\mathcal{B}$	0	0	1	1	2	2	...	∞
$\mathcal{C}$	0	1	1	2	2	3	...	∞

Table 2.4: The function $\mathfrak{f}_0^{(U)}$ to $\mathfrak{f}_5^{(U)}$ as well as $\mathfrak{f}^{(U)}$ for the universe U consisting of the rules in Table 2.3.

By Lemma 2.6 we can, therefore, compute the counting sequence of $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$. It is, however, not possible to build a specification containing the combinatorial set

$\mathcal{B}$ from the rules in U. If we chose to include the rule $\mathcal{B} \leftarrow (\mathcal{A})$ in our specification then we also need a rule for $\mathcal{A}$. The only choice is $\mathcal{A} \leftarrow (\mathcal{B})$. We, therefore, have the edges $\mathcal{B}_2 \rightarrow \mathcal{A}_3$ and $\mathcal{A}_3 \rightarrow \mathcal{B}_2$ which form a directed cycle in the reliance graph of the specification. A similar scenario occurs if we chose instead the rule $\mathcal{B} \leftarrow (\mathcal{C})$. We can, therefore, not find a specification with a reliance graph that has no infinite directed walk containing the combinatorial set $\mathcal{B}$, despite the set being in the enumerable subset.

We will, moving forward, restrict ourselves to universes where all the rules are produced by so-called regular strategies. This restriction will allow us to ensure that all the combinatorial sets in the enumerable subset are in a specification. For those restricted universes, we will show that we can compute $\mathfrak{f}^{(U)}$ from only a finite number of $\mathfrak{f}_i^{(U)}$ which will make this approach computationally feasible. The restriction we impose on strategies for them to be regular is a restriction on their reliance functions. Each component of the reliance function of a regular strategy must be a constant shift from the input.

Definition 2.12. *A k-ary strategy S is* regular *if there are k constants $\mathfrak{C}_S^{(1)}, \ldots, \mathfrak{C}_S^{(k)} \in \mathbb{N}$ such that*

$$r_S(n) = (n - \mathfrak{C}_S^{(1)}, \ldots, n - \mathfrak{C}_S^{(k)})$$

for all $n \in \mathbb{N}$.

For a regular strategy S, if the constant $\mathfrak{C}_S^{(j)}$ is positive, it means that the strategy relies only on smaller sizes of its j-th child in its counting function. If $\mathfrak{C}_S^{(j)}$ is 0, then the counting function of S relies on size n of the j-th child. Similarly, a negative $\mathfrak{C}_S^{(j)}$ means relying on a size bigger than n.

In the example above, the strategies that produced the rules for $\mathcal{B}$ are not regular strategies since the shift in their reliance profile functions is not constant but instead changes with the parity of n.

At first sight, this condition might seem to restrict massively the kind of strategies we can use, especially since it is does not include all productive strategies. In practice, however, the definition of a regular strategy seems to encompass all the useful strategies we use. In particular, we will observe in Section 2.6 that all the strategies for tilings described in [34] are regular strategies. The reliance profile functions described in the big example of Section 2.2 satisfy the conditions for a regular strategy despite some of them not complying with the definition of productive strategies.

Some strategies are regular but not productive, for example all equivalence strategies. As defined in [34], they are unary strategies for which the decomposition function output is another combinatorial set that is equinumerous to the input. The counting function for such a strategy is then the identity while the reliance profile function is $n \mapsto (n)$. As highlighted in Section 4.3 of [34], these strategies are not productive and therefore need to be treated differently when working with the algorithm designed for productive strategies. Equivalence strategies are, however, regular strategies since their reliance function is $n \mapsto (n - 0)$. Hence, we do not need to treat them any differently in our theory.

As we said above, in a universe where all the rules are produced by regular strategies, we can guarantee that for any combinatorial set in the enumerable subset the universe contains a productive specification for that combinatorial set. We prove that in Section 2.4.3 with Theorem 2.20 but we will first need a few intermediate results.

2.4.1 A fully enumerable universe

We first show that if a universe allows us to compute the first term of the counting sequence of all its combinatorial sets, then every combinatorial set in the universe is actually in a productive specification.

Lemma 2.13. *Let U be a universe of rules produced by regular strategies. Assume that there is a $K \in \mathbb{N}$ such that $\mathfrak{f}_K^{(U)}(\mathcal{C}) > 0$ for all $\mathcal{C}$ in $C(U)$. Then, for any combinatorial set $\mathcal{C}$ in $C(U)$, there is a subset of U which is a specification for $\mathcal{C}$ with a reliance graph that contains no infinite directed walk.*

Proof. The proof is structured as follow. First, we extract a specification for $C(U)$ where each combinatorial set of the universe is on the left-hand side of one rule. In other words, we select one rule of the universe for each combinatorial set in the universe. Second, we show that the reliance graph of this specification does not contain an infinite directed walk. To do so we proceed by contradiction. Assuming the graph contains an infinite directed walk, we will first show that this walk needs to be a cycle and then show that it is also impossible to have a cycle in the graph to obtain the contradiction.

We start by selecting the rules of the specification. Let $\mathcal{C}$ be a combinatorial set in the universe. Let $i \in \mathbb{N}$ be maximal such that $\mathfrak{f}_i^{(U)}(\mathcal{C}) < \mathfrak{f}_K^{(U)}(\mathcal{C})$. By the way we chose i we know that, $\mathfrak{f}_{i+1}^{(U)}(\mathcal{C}) = \mathfrak{f}_i^{(U)}(\mathcal{C}) + 1$. Hence, the universe must contain a rule $\mathcal{C} \overset{S_{\mathcal{C}}}{\leftarrow} (\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(k)})$ which satisfies the increment condition for $\mathfrak{f}_i^{(U)}$. This process gives a rule for each combinatorial set of the universe, *i.e.*, a combinatorial specification that contains all the combinatorial sets in the universe. To complete the proof, we need to show that the reliance graph of this specification contains no infinite directed walk.

We first make an observation about the rule selected for each combinatorial set $\mathcal{C}$. Let $S_{\mathcal{C}}$ be the strategy that produced the rule $\mathcal{C} \overset{S_{\mathcal{C}}}{\leftarrow} (\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(k)})$ selected for $\mathcal{C}$ in the specification. Since the rule satisfies the increment condition for $\mathfrak{f}_i^{(U)}$, we have

$$r_{S_{\mathcal{C}}}^{(j)}(\mathfrak{f}_i^{(U)}(\mathcal{C})) < \mathfrak{f}_i^{(U)}(\mathcal{A}^{(j)})$$

for $1 \le j \le k$. Since $\mathfrak{f}_K^{(U)}(\mathcal{C}) = \mathfrak{f}_i^{(U)}(\mathcal{C}) + 1$ and $\mathfrak{f}_i^{(U)} \le \mathfrak{f}_K^{(U)}$, the inequalities can be written in term of $\mathfrak{f}_K^{(U)}$ as

$$r_{S_{\mathcal{C}}}^{(j)}(\mathfrak{f}_K^{(U)}(\mathcal{C}) - 1) < \mathfrak{f}_K^{(U)}(\mathcal{A}^{(j)}).$$

Finally, since $S_{\mathcal{C}}$ is regular, we get that

$$\mathfrak{f}_K^{(U)}(\mathcal{C}) - \mathfrak{C}_{S_{\mathcal{C}}}^{(j)} \leq \mathfrak{f}_K^{(U)}(\mathcal{A}^{(j)}) \tag{2.1}$$

for $1 \leq j \leq k$.

We now show that the reliance graph of the specification contains no infinite directed walk. Suppose that the graph contains an infinite directed walk. We consider the level function denoted lvl from the vertices of the reliance graph of the specification in $\mathbb{N}$ defined as

$$\mathrm{lvl}(\mathcal{C}_n) = n - \mathfrak{f}_K^{(U)}(\mathcal{C}) + M \tag{2.2}$$

where $M = \max_{\mathcal{C} \in C(U)}(\mathfrak{f}_K^{(U)}(\mathcal{C}))$. We select M in this way to ensure that the codomain of the function lvl is $\mathbb{N}$.

Consider an edge $\mathcal{C}_n \to \mathcal{B}_m$ of the reliance graph of the specification. Then, if $\mathcal{C} \overset{S_{\mathcal{C}}}{\leftarrow} (\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(k)})$ is the rule of the specification with $\mathcal{C}$ as the parent, we must have that $\mathcal{B}$ is one of its children. Let $i_{\mathcal{B}}$ be the index such that $\mathcal{A}^{(i_{\mathcal{B}})} = \mathcal{B}$. By construction of the reliance graph, we have

$$m \leq r_{S_{\mathcal{C}}}^{(i_{\mathcal{B}})}(n) = n - \mathfrak{C}_{S_{\mathcal{C}}}^{(i_{\mathcal{B}})}$$

since the graph contains the edge $\mathcal{C}_n \to \mathcal{B}_m$. By adding (2.1) and M on each side, we find

$$m + \mathfrak{f}_K^{(U)}(\mathcal{C}) - \mathfrak{C}_{S_{\mathcal{C}}}^{(i_{\mathcal{B}})} + M \leq n - \mathfrak{C}_{S_{\mathcal{C}}}^{(i_{\mathcal{B}})} + \mathfrak{f}_K^{(U)}(\mathcal{B}) + M.$$

We can add $\mathfrak{C}_{S_{\mathcal{C}}}^{(i_{\mathcal{B}})}$ on both sides to get

$$m - \mathfrak{f}_K^{(U)}(\mathcal{B}) + M \leq n - \mathfrak{f}_K^{(U)}(\mathcal{C}) + M.$$

Using the definition of lvl we can simplify to

$$\mathrm{lvl}(\mathcal{B}_m) \leq \mathrm{lvl}(\mathcal{C}_n).$$

This shows that any edges in the reliance graph can only end at a vertex which has level less than or equal to the level of the vertex it starts at. Since there is a lower bound of 0 on the level of a vertex, the infinite directed walk must have an infinite suffix where all the vertices are of the same level. In particular, the graph must contain an infinite directed walk where all the vertices are of the same level.

Since only a finitely many vertices are of the same level, the graph must contain a cycle. Let

$$\mathcal{A}_{n_1}^{(1)} \to \mathcal{A}_{n_2}^{(2)} \to \cdots \to \mathcal{A}_{n_k}^{(k)} \to \mathcal{A}_{n_{k+1}}^{(k+1)} = \mathcal{A}_{n_1}^{(1)}$$

be a cycle in the reliance graph. Each edge $\mathcal{A}_{n_i}^{(i)} \to \mathcal{A}_{n_{i+1}}^{(i+1)}$ in the cycle gives an inequality of the form

$$n_{i+1} \leq r_{S_{\mathcal{A}_i}}^{(j_i)}(n_i) = n_i - \mathfrak{C}_{S_{\mathcal{A}^{(i)}}}^{(j_i)}$$

where j_i is the index of $\mathcal{A}^{(i+1)}$ among the children of the rule with $\mathcal{A}^{(i)}$ as a parent. If we sum all these inequalities for each edge, the n_i cancel and we get

$$\sum_{i=1}^{k} \mathfrak{C}^{(j_i)}_{S_{\mathcal{A}^{(i)}}} \geq 0. \tag{2.3}$$

To find the contradiction, we now show that this sum must be strictly less than 0. Let K' be the largest integer such that $\mathfrak{f}^{(U)}_{K'}(\mathcal{A}^{(i)}) < \mathfrak{f}^{(U)}_{K}(\mathcal{A}^{(i)})$ for all combinatorial sets $\mathcal{A}^{(i)}$ involved in the cycle. By the maximality of K', there is at least one combinatorial set $\mathcal{A}^{(i)}$ in the cycle such that $\mathfrak{f}^{(U)}_{K'+1}(\mathcal{A}^{(i)}) = \mathfrak{f}^{(U)}_{K}(\mathcal{A}^{(i)})$. Without loss of generality, we can assume that $i = 1$ and

$$\mathfrak{f}^{(U)}_{K'}(\mathcal{A}^{(1)}) + 1 = \mathfrak{f}^{(U)}_{K'+1}(\mathcal{A}^{(1)}) = \mathfrak{f}^{(U)}_{K}(\mathcal{A}^{(1)}).$$

By the choice of K', we have

$$\mathfrak{f}^{(U)}_{K'}(\mathcal{A}^{(2)}) \leq \mathfrak{f}^{(U)}_{K}(\mathcal{A}^{(2)}) - 1.$$

Finally, from the construction of the functions, we derive that

$$\mathfrak{f}^{(U)}_{K'}(\mathcal{A}^{(1)}) - \mathfrak{C}^{(j_1)}_{S_{\mathcal{A}^{(1)}}} = r^{(j_1)}_{S_{\mathcal{A}^{(1)}}}(\mathfrak{f}^{(U)}_{K'}(\mathcal{A}^{(1)})) < \mathfrak{f}^{(U)}_{K'}(\mathcal{A}^{(2)}).$$

Combining the last three equations, we get

$$\mathfrak{f}^{(U)}_{K}(\mathcal{A}^{(1)}) - \mathfrak{C}^{(j_1)}_{S_{\mathcal{A}^{(1)}}} < \mathfrak{f}^{(U)}_{K}(\mathcal{A}^{(2)}).$$

For $2 \leq i \leq k$, we can get a similar inequality (though not strict) from (2.1). Precisely, we have

$$\mathfrak{f}^{(U)}_{K}(\mathcal{A}^{(i)}) - \mathfrak{C}^{(j_i)}_{S_{\mathcal{A}^{(i)}}} \leq \mathfrak{f}^{(U)}_{K}(\mathcal{A}^{(i+1)})$$

for $2 \leq i \leq k$. Summing the inequalities, we get

$$\sum_{i=1}^{k} \mathfrak{C}^{(j_i)}_{S_{\mathcal{A}^{(i)}}} < 0.$$

since all the $\mathfrak{f}^{(U)}_{k}(\mathcal{A}^{(i)})$ terms cancel. This is a contradiction with (2.3). Therefore the reliance graph of the specification does not contain an infinite directed walk. □

One interesting consequence of this lemma is that the enumerable subset of such a universe is always $C(U)$. This follows from Theorem 2.7 since all combinatorial sets of U are in a specification whose reliance graph contains no infinite directed walk.

Corollary 2.14. *Let U be a universe where all rules are produced by regular strategies. Assume there is a $K \in \mathbb{N}$ such that $\mathfrak{f}^{(U)}_{K}(\mathcal{C}) > 0$ for all $\mathcal{C}$ in $C(U)$. Then $C(U) = E(U)$.*

The situation where all the combinatorial sets in the universe are actually in the enumerable subset is however very unlikely to appear in practice. What we expect to see is a universe where some of the combinatorial sets are in the enumerable subset and some can only be enumerated up to some size (or maybe not at all). In those cases, what we would like to do is to identify a subset of the universe that has the same enumerable subset but no extra combinatorial sets that are not in the enumerable subset. If we found such a subset, we could then use Lemma 2.13 to guarantee the existence of a specification for each combinatorial set in the enumerable subset.

2.4.2 A separating function

To be able to identify such a subset, we introduce the notion of a *separating function* of a universe U. For any function $g : C(U) \to \mathbb{N}$, we can always find an interval of size μ such that the preimage of this interval is empty. We call such an interval a *gap* of size μ of g. Once such an interval is fixed, it can always be used to split $C(U)$ into two sets: the combinatorial sets that have their image to the left of the interval on the number line and those to the right of it.

Especially, if the gap is of size $\mu(U)$ where $\mu(U)$ is defined as the maximum the absolute value $\left|\mathfrak{C}_S^{(j)}\right|$ of any strategy S used to produce a rule in U, an interesting property occurs. Any reliance profile function involved in the universe maps numbers to the left of the gap to numbers inside or to the left of the gap. Similarly, any reliance function maps numbers to the right of the gap only to numbers in or to the right of the gap. This means, in particular, that no rule with a child not on the same side of the gap as the parent can satisfy the increment condition for g. We will see later in this section how this property can be used to compute the enumerable subset of the universe.

To be a separating function of U, a function $g : C(U) \to \mathbb{N}$ must satisfy three conditions. First, we must be able to compute at least $g(\mathcal{C})$ terms of the counting sequence of $\mathcal{C}$ using the rules in U. Equivalently, g must be a lower bound for $\mathfrak{f}^{(U)}$. Second, the interval $[K, K + \mu(U) - 1]$ must be a gap of the function. Third, no rule with a parent that is on the left side of the gap of size $\mu(U)$ must satisfy the increment condition for g.

Definition 2.15. *Formally, let U be a universe where the rules are produced by regular strategies and*

$$\mu(U) = \max\left\{\left|\mathfrak{C}_S^{(j)}\right| : S \text{ is an } m\text{-ary strategy used in } U \text{ and } 1 \leq j \leq m\right\}.$$

*Let $g : C(U) \to \mathbb{N}$ be a function and $K \in \mathbb{N}$. Then g is a K-*separating function *for U if*

(i) $g(\mathcal{C}) \leq \mathfrak{f}^{(U)}(\mathcal{C})$ *for all* $\mathcal{C} \in C(U)$,

(ii) $g^{-1}([K, K + \mu - 1]) = \emptyset$

(iii) *for any combinatorial set $\mathcal{C}$ such that $g(\mathcal{C}) < K$ there is no rule with $\mathcal{C}$ as the parent that satisfies the increment condition for g.*

We call $[K, K+\mu(U)-1]$ *the* separating gap *of* g*.*

As mentioned previously, the combinatorial sets in $C(U)$ can be partitioned based on whether g maps them to the left or the right of the gap. If

$$L = \{\mathcal{C} \in C(U) : g(\mathcal{C}) < K\}$$

and

$$R = \{\mathcal{C} \in C(U) : g(\mathcal{C}) \geq K + \mu(U)\}$$

then $C(U)$ is the disjoint union $L \sqcup R$. To understand condition (iii), it is good to think of g as a function describing the number of terms of the counting sequence known for each combinatorial set. This is similar to the way we think of the $\mathfrak{f}_i^{(U)}$ functions. With this point of view, condition (iii) of the definition means that, if, for each combinatorial set, we know the number of terms of the counting sequence given by g, then we cannot compute any more terms of the counting sequence of the combinatorial sets in L using the rules of the universe.

It is relatively easy to check whether a function $\mathfrak{f}_i^{(U)}$ is a K-separating function for U. By construction, all $\mathfrak{f}_i^{(U)}$ are a lower bound of $\mathfrak{f}^{(U)}$. Hence, condition (i) is always satisfied. Condition (ii) is verified if $\mathfrak{f}_i^{(U)}([K, K+\mu(U)-1]) = \emptyset$. Moreover, if $\mathfrak{f}_i^{(U)}(\mathcal{C}) = \mathfrak{f}_{i+1}^{(U)}(\mathcal{C})$ for all combinatorial sets $\mathcal{C}$ on the left of the gap then condition (iii) is also satisfied and $\mathfrak{f}_i^{(U)}$ is a separating function. In fact, for any universe of regular rules, there is always one of the $\mathfrak{f}_i^{(U)}$ that is a separating function.

Lemma 2.16. *Let U be a universe where the rules are produced by regular strategies. There exists $J, K \in \mathbb{N}$ such that $\mathfrak{f}_J^{(U)}$ is a K-separating function and $E(U)$ is the set of combinatorial sets to the right of the gap.*

Proof. Let $K-1$ be the largest value that $\mathfrak{f}^{(U)}$ reaches that is not infinity, *i.e.*,

$$K = 1 + \max_{\mathcal{C} \in C(U) \setminus E(U)} (\mathfrak{f}^{(U)}(\mathcal{C})).$$

By construction, we can find $J \in \mathbb{N}$ such that $\mathfrak{f}_J^{(U)}(\mathcal{C}) = \mathfrak{f}^{(U)}(\mathcal{C})$ for any combinatorial set $\mathcal{C}$ that is not in the enumerable subset of U and $\mathfrak{f}_J^{(U)}(\mathcal{C}) > K + \mu(U) - 1$ for any combinatorial set $\mathcal{C}$ that is in the enumerable subset. We show that $\mathfrak{f}_J^{(U)}$ is a K-separating function for U. Since $\mathfrak{f}_J^{(U)}$ is a lower bound for $\mathfrak{f}^{(U)}$, condition (i) is satisfied. By the choice of J, condition (ii) is also satisfied. Moreover, since $\mathfrak{f}_J^{(U)}|_{C(U)\setminus E(U)}$ is equal to $\mathfrak{f}^{(U)}|_{C(U)\setminus E(U)}$, it must also be equal to $\mathfrak{f}_{J+1}^{(U)}|_{C(U)\setminus E(U)}$. Hence, condition (iii) is also satisfied. By our choice of J, we also have that $E(U)$ is the set of combinatorial sets to the right of the gap. □

Lemma 2.16 shows that there is always a separating function that allows us to identify the enumerable subset of the universe. Actually, we will, in the following pages, demonstrate that for any separating function the enumerable subset is always the set of combinatorial sets to the right of the gap (see Corollary 2.19). We start by showing with Lemma 2.17 that the combinatorial sets to the left of the gap are never in the enumerable subset. In fact, the value that a separating function can take for the combinatorial sets on the left of the gap are unique. We show that they must be the value of $\mathfrak{f}^{(U)}$.

Lemma 2.17. *Let U be a universe where the rules are produced by regular strategies. Let g be a K-separating function for U. If L is the set of combinatorial sets of U that are to the left of the gap of g, then $\mathfrak{f}^{(U)}|_L = g|_L$.*

Proof. Assume that $\mathfrak{f}^{(U)}|_L \neq g|_L$. Since g is a lower bound for $\mathfrak{f}^{(U)}$, there must be some combinatorial set in L for which $\mathfrak{f}^{(U)}$ exceeds g. Let $i \in \mathbb{N}$ be maximal such that $\mathfrak{f}_i^{(U)}|_L \leq g|_L$. By maximality of i, there exists $\mathcal{C} \in L$ such that $\mathfrak{f}_{i+1}^{(U)}(\mathcal{C}) > g(\mathcal{C})$ and $\mathfrak{f}_i^{(U)}(\mathcal{C}) = g(\mathcal{C})$. By construction, $\mathfrak{f}_{i+1}^{(U)}(\mathcal{C}) = \mathfrak{f}_i^{(U)}(\mathcal{C}) + 1$ and there is a rule $\mathcal{C} \overset{S}{\leftarrow} (\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(m)})$ that satisfies the increment condition for $\mathfrak{f}_i^{(U)}$. Consider $\mathcal{A}^{(j)}$, a child of that rule. If $\mathcal{A}^{(j)} \in L$ then

$$r_S^{(j)}(g(\mathcal{C})) \leq r_S^{(j)}(\mathfrak{f}^{(U)}(\mathcal{C})) < \mathfrak{f}_i^{(U)}(\mathcal{A}^{(j)}) \leq g(\mathcal{A}^{(j)})$$

since $\mathfrak{f}_i^{(U)}$ is a lower bound for g. Otherwise, if $\mathcal{A}^{(j)} \notin L$, then the image of $\mathcal{A}^{(j)}$ under g must be to the right of the gap of g. Consequently,

$$r_S^{(j)}(g(\mathcal{C})) = g(\mathcal{C}) - \mathfrak{c}_S^{(j)} \leq g(\mathcal{C}) + \mu(U) \leq K + \mu(U) - 1 < g(\mathcal{A}^{(j)}).$$

Since one of those cases apply for every child $\mathcal{A}^{(j)}$, the rule satisfies the increment condition for g, which is in contradiction with condition (iii) of Definition 2.15. □

Lemma 2.17 allows us to use a separating function to identify combinatorial sets that are not in the enumerable subset. As we will prove with the next lemma, these combinatorial sets do not have an impact on what is in the enumerable subset. We can remove from U the rules that involve combinatorial sets in L without changing the enumerable subset.

Lemma 2.18. *Let U be a universe where all the rules are produced by regular strategies. Let g be a K-separating function for U. Let R be the set of combinatorial sets of U that are to the right of the gap. If U' is the subset of U consisting of only the rules with all their combinatorial sets in R then*

$$\mathfrak{f}^{(U)}|_R \leq \mathfrak{f}^{(U')} + K + \mu(U) - 1.$$

Proof. We define a sequence of functions $g_i : C(U') \to \mathbb{N}$ by

$$g_i(\mathcal{C}) = \begin{cases} \mathfrak{f}_i^{(U)}(\mathcal{C}) - K - \mu(U) + 1 & \text{if } \mathfrak{f}_i^{(U)}(\mathcal{C}) \geq K + \mu(U) - 1 \\ 0 & \text{otherwise.} \end{cases}$$

and show that $g_i \leq \mathfrak{f}_i^{(U')}$ for all $i \in \mathbb{N}$. We proceed by induction. The base case follows from the fact that g_0 is always 0. Assume that $g_i \leq \mathfrak{f}_i^{(U')}$ and show that $g_{i+1} \leq \mathfrak{f}_{i+1}^{(U')}$. Let $\mathcal{C} \in C(U')$. If $g_i(\mathcal{C}) = g_{i+1}(\mathcal{C})$ or $g_i(\mathcal{C}) < \mathfrak{f}_i^{(U')}(\mathcal{C})$ then $g_{i+1}(\mathcal{C}) \leq \mathfrak{f}_{i+1}^{(U')}(\mathcal{C})$. Otherwise, $g_{i+1}(\mathcal{C}) = g_i(\mathcal{C}) + 1$ and $g_i(\mathcal{C}) = \mathfrak{f}_i^{(U')}(\mathcal{C})$. From the first equality, we deduce that $\mathfrak{f}_{i+1}^{(U)}(\mathcal{C}) = \mathfrak{f}_i^{(U)}(\mathcal{C}) + 1$ and $\mathfrak{f}_i^{(U)}(\mathcal{C}) \geq K + \mu(U) - 1$. By definition of the $\mathfrak{f}_i^{(U)}$, there is a rule $\mathcal{C} \overset{S}{\leftarrow} (\mathcal{A}^{(1)}, \dots, \mathcal{A}^{(m)})$ in U that satisfies the increment condition for $\mathfrak{f}_i^{(U)}$. More formally, the reliance profile function of the strategy S producing the rule satisfies

$$r_S^{(j)}(\mathfrak{f}_i^{(U)}(\mathcal{C})) < \mathfrak{f}_i^{(U)}(\mathcal{A}^{(j)})$$

for $1 \leq j \leq m$. Since the strategy S is regular, the inequalities above can be simplified to

$$\mathfrak{f}_i^{(U)}(\mathcal{C}) - \mathfrak{c}_S^{(j)} < \mathfrak{f}_i^{(U)}(\mathcal{A}^{(j)}). \tag{2.4}$$

Since $\mathfrak{f}_i^{(U)}(\mathcal{C}) \geq K + \mu(U) - 1$, we derive that

$$K + \mu(U) - 1 - \mathfrak{c}_S^{(j)} < \mathfrak{f}_i^{(U)}(\mathcal{A}^{(j)}).$$

By definition, $\mu(U) - \mathfrak{c}_S^{(j)} \geq 0$ so $K \leq \mathfrak{f}_i^{(U)}(\mathcal{A}^{(j)})$. Since $\mathfrak{f}_i^{(U)}$ is a lower bound for $\mathfrak{f}^{(U)}$, we have

$$K \leq \mathfrak{f}^{(U)}(\mathcal{A}^{(j)}). \tag{2.5}$$

By Lemma 2.17, if $g(\mathcal{A}^{(j)}) < K$, then $\mathfrak{f}^{(U)}(\mathcal{A}^{(j)}) = g(\mathcal{A}^{(j)}) < K$ which contradicts Equation (2.5). Therefore, the image of $\mathcal{A}^{(j)}$ under g is on the right of the separating gap. Consequently, the rule $\mathcal{C} \overset{S}{\leftarrow} (\mathcal{A}^{(1)}, \dots, \mathcal{A}^{(m)})$ is a part of U'.

Moreover, from Equation (2.4), we have

$$\mathfrak{f}_i^{(U)}(\mathcal{C}) - (K + \mu(U) - 1) - \mathfrak{c}_S^{(j)} < \mathfrak{f}_i^{(U)}(\mathcal{A}^{(j)}) - (K + \mu(U) - 1).$$

Since $\mathfrak{f}_i^{(U)}(\mathcal{C}) \geq K + \mu(U) - 1$, we have $g_i(\mathcal{C}) = \mathfrak{f}_i^{(U)}(\mathcal{C}) - (K + \mu(U) - 1)$ and by construction $\mathfrak{f}_i^{(U)}(\mathcal{A}^{(j)}) - (K + \mu(U) - 1) \leq g_i(\mathcal{A}^{(j)})$. Hence,

$$g_i(\mathcal{C}) - \mathfrak{c}_S^{(j)} < g_i(\mathcal{A}^{(j)}).$$

Moreover, since $g_i(\mathcal{C}) = \mathfrak{f}_i^{(U')}(\mathcal{C})$ and $g_i \leq \mathfrak{f}_i^{(U')}$, we have

$$\mathfrak{f}_i^{(U')}(\mathcal{C}) - \mathfrak{c}_S^{(j)} < \mathfrak{f}_i^{(U')}(\mathcal{A}^{(j)}).$$

Since S is regular, we have

$$r_S^{(j)}(\mathfrak{f}_i^{(U')}(\mathcal{C})) < \mathfrak{f}_i^{(U')}(\mathcal{A}^{(j)}).$$

Therefore, the rule satisfies the increment condition for $\mathfrak{f}_i^{(U')}$. Since the rule is in U', we have $\mathfrak{f}_{i+1}^{(U')}(\mathcal{C}) = \mathfrak{f}_i^{(U')}(\mathcal{C}) + 1$. So $g_{i+1}(\mathcal{C}) = \mathfrak{f}_{i+1}^{(U')}(\mathcal{C})$ and that concludes the induction.

To conclude the proof, we consider a combinatorial set $\mathcal{C}$ in R. From the induction above, we can infer that

$$\lim_{i\to\infty} g_i(\mathcal{C}) \leq \mathfrak{f}^{(U')}(\mathcal{C}).$$

Moreover, since $\mathcal{C}$ is to the right of the gap, we know that

$$K + \mu(U) \leq g(\mathcal{C}) \leq \mathfrak{f}^{(U)}(\mathcal{C}).$$

Therefore there exists $j \in \mathbb{N}$ such that $\mathfrak{f}_i^{(U)}(\mathcal{C}) \geq K + \mu(U)$ for all $i \geq j$. Hence,

$$\lim_{i\to\infty} g_i(\mathcal{C}) = \lim_{i\to\infty} \mathfrak{f}_i^{(U)}(\mathcal{C}) - K - \mu(U) + 1 = \mathfrak{f}^{(U)}(\mathcal{C}) - K - \mu(U) + 1.$$

Combining with the inequality above, we get

$$\mathfrak{f}^{(U)}(\mathcal{C}) \leq \mathfrak{f}^{(U')}(\mathcal{C}) + K + \mu(U) - 1. \qquad \square$$

A combinatorial set $\mathcal{C}$ in the enumerable subset of U cannot be in the set L by Lemma 2.17. Hence it must be in R. Since it is in the enumerable subset of U, we also have that $\mathfrak{f}^{(U)}(\mathcal{C}) = \infty$. By Lemma 2.18, $\mathfrak{f}^{(U')}(\mathcal{C})$ is also ∞ and $\mathcal{C}$ is in the enumerable subset of U'. Therefore $E(U) \subseteq E(U')$. Since U' is a subset of U and adding rules can only make the enumerable subset bigger, we have that $E(U) = E(U')$. We can therefore think of Lemma 2.18 as allowing us to remove rules from a universe in such a way that the enumerable subset is not affected.

In fact, we can say much more about the enumerable subset when we have a K-separating function for a universe U. Consider the subuniverse U' like the one in Lemma 2.18. From the definition of a K-separating function, we have

$$K + \mu(U) \leq g|_{C(U')} \leq f|_{C(U')}$$

and from Lemma 2.18, we have

$$\mathfrak{f}^{(U)}|_{C(U')} \leq \mathfrak{f}^{(U')} + K + \mu(U) - 1.$$

Combining those two inequalities together, we get

$$1 \leq \mathfrak{f}^{(U')}.$$

Corollary 2.14 gives us that $C(U') = E(U')$ and therefore $E(U) = C(U')$.

Corollary 2.19. *Let U be a universe where all the rules are produced by regular strategies. Let g be a K-separating function for U. Then the enumerable subset of U is*

$$\{\mathcal{C} \in C(U) : g(c) > K\}.$$

2.4.3 Enumerable subset

After all these technical results, time has come to reap the rewards of our work and finally prove that every combinatorial set in the enumerable subset is in a productive specification.

Theorem 2.20. *Let U be a universe of rules produced by regular strategies. The enumerable subset of U is the set of all combinatorial sets of U that are in a specification contained in U and whose reliance graph contains no infinite directed walk.*

Proof. From Theorem 2.7, we already know that any combinatorial set that is in a specification must be in the enumerable subset.

To show that any combinatorial set in the enumerable subset is in a productive specification we will use Lemma 2.13. By Lemma 2.16, we can find a separating function $\mathfrak{f}_K^{(U)}$ for U. We know that the combinatorial sets to the right of the gap form the enumerable subset of U. By Lemma 2.18, the universe U' consisting of the rules of U where all the combinatorial sets are in $E(U)$ has the same enumerable subset as U. This set is also the set of combinatorial sets of U'. In fact, we have $E(U) = E(U') = C(U')$.

Since $E(U') = C(U')$ there is a K' such that $\mathfrak{f}_{K'}^{(U')} > 0$. Therefore, by Lemma 2.13 U' contains a specification whose reliance graph has no infinite directed walk for each combinatorial set in $C(U')$. Since U' is a subset of U, U contains such a specification for any combinatorial set in $C(U') = E(U)$. □

We, therefore, have that the enumerable subset is the set of all combinatorial sets that are in a specification with no infinite directed walk in its reliance graph. This result is in some sense similar to Theorem 3.1 from [34] that states that the output of Algorithm 1 (which we call the prune method here) is the set of all rules contained in a productive specification. The enumerable subset is, in some sense, the analogue of the output of the prune method. The first one giving a set of combinatorial sets that are in a productive specification while the latter gives the set of rules that are in a productive specification. It is also worth noting that in [34] the strategies used must be productive while in the our theory they must be regular. Both approaches, however, serve the same purpose of identifying what can be enumerated in the universe. Having only the combinatorial sets and not the rules that are in a specification will make the extraction of a specification more difficult. This issue will be addressed later but we will first establish how the enumerable subset can be computed by a computer. To do so we will use again the technical lemmas established in this section.

2.5 Computing the enumerable subset

As we have seen in the Section 2.4.2, a separating function allows us to distinguish between combinatorial sets that are in the enumerable subset and those that are not. In particular, Corollary 2.19 tells us that the enumerable subset of U is the set R of combinatorial sets to the right of the gap of any separating function for that universe. This means that, to compute the enumerable subset, we actually only need to find a separating function. We have seen right before Lemma 2.16 how one could easily check whether the function $\mathfrak{f}_i^{(U)}$ is K-separating using $\mathfrak{f}_{i+1}^{(U)}$. This check can, however, only answer the question for one specific K. It could be that our $\mathfrak{f}_i^{(U)}$ is not K-separating but $(K+3)$-separating instead. With the following lemma we however show that it is sufficient to check if a function g is K'-separating for a single well-chosen K' in order to know if there exists a K such the g is K-separating. As in Definition 2.15, $\mu(U)$ is defined as the maximum of all the $\left|\mathfrak{C}_S^{(j)}\right|$ for any strategy S used to produce a rule in U.

Lemma 2.21. *Let U be a universe where the rules are produced by regular strategies. Let g be a K-separating function for U. If*

$$K' = \min\{n \in \mathbb{N} : g^{-1}([n, n + \mu(U) - 1]) = \emptyset\}$$

then g is also K'-separating.

Proof. Let g be a K-separating function for g. We show it is also K'-separating. Since g is K-separating, we have that $g^{-1}([K, K + \mu(U) - 1]) = \emptyset$. Hence, by construction of K', we have that $K' \leq K$. Therefore, condition (iii) of Definition 2.15 is satisfied. By construction, condition (ii) is also satisfied. Condition (i) follows from g being K-separating. □

An algorithm to compute the enumerable subset is described in Algorithm 2. It successively builds the $\mathfrak{f}_i^{(U)}$ (line 4) and checks whether they are a separating function or not (lines 5–7). Lemma 2.21 allows us to check if $\mathfrak{f}_i^{(U)}$ is K-separating for a single K. When one separating function is found, the algorithm returns the set of combinatorial sets to the right of the function gap (line 8). By Corollary 2.19, this is the enumerable subset.

Moreover, from Lemma 2.16 we know that one of the $\mathfrak{f}_i^{(U)}$ is a separating function. The algorithm is, therefore, guaranteed to terminate.

Theorem 2.22. *Algorithm 2 terminates and returns the enumerable subset of the universe.*

A more advanced version of Algorithm 2 has been implemented as part of the `comb_spec_searcher` python package [41]. The implemented version does not compute successively the $\mathfrak{f}_i^{(U)}$ but instead maintains a lower bound for $\mathfrak{f}^{(U)}$ that can be updated as rules are added to the universe. This is useful for combinatorial exploration as we can follow the evolution of the enumerable subset as we create the universe. The implementation also maintains additional data structures to avoid needing to loop over the whole universe of rules. These implementation details allow for a fast computation of the enumerable subset. In practice, computing the enumerable subset takes a negligible amount of time when contrasted with the time needed to create the rules of the universe and compute the reliance functions.

Algorithm 2 Finding the enumerable subset of the universe U.

Input: A universe U of combinatorial rules produced by regular strategies
Output: The enumerable subset of U

$i \leftarrow 0$
$\mathfrak{f}_0^{(U)} : C(U) \to \mathbb{N}$ is always 0
loop
    $\mathfrak{f}_{i+1}^{(U)} \leftarrow$ Make one pass through the rule of U to build from $\mathfrak{f}_i^{(U)}$
    $K \leftarrow \min\{n \in \mathbb{N} : {\mathfrak{f}_i^{(U)}}^{-1}([n, n + \mu(U) - 1]) = \emptyset\}$
    $L \leftarrow \{\mathcal{C} \in U : \mathfrak{f}_i^{(U)}(\mathcal{C}) < K\}$
    if $\mathfrak{f}_i^{(U)}|_L = \mathfrak{f}_{i+1}^{(U)}|_L$ **then**
        return $\{\mathcal{C} \in U : \mathfrak{f}_i^{(U)}(\mathcal{C}) > K\}$
    end if
    $i \leftarrow i + 1$
end loop

2.6 Enumerable subsets for permutation patterns

In [34], the domain of pattern-avoiding permutations is the application of combinatorial exploration that is discussed to the greatest extent. Six different types of combinatorial strategies are discussed at length: requirement insertion, obstruction and requirement simplification, point placement, row separation and column separation, factorization and obstruction inferral.

We will quickly revisit those strategies and show that they are regular. We will also formalize strategies that produce the rules we have obtained by flipping rules in the example of Section 2.2, and show that altough not necessarly productive, they are regular.

2.6.1 The requirement insertion strategy

Section 6.3.1 of [34] introduces the requirement insertion strategy. The idea behind this strategy is to split a set of gridded permutations defined by a tiling into two sets depending on whether they contain one of the gridded permutations in a set of gridded permutations H or not. Formally, a gridded permutation griddable on $\mathcal{T} = ((t, u), \mathcal{O}, \mathcal{R})$ contains one of the gridded permutations in H if it is griddable on $((t, u), \mathcal{O}, \mathcal{R} \cup \{H\})$. On the other hand, it does not contain any of the gridded permutations in H if it is griddable on $((t, u), \mathcal{O} \cup H, \mathcal{R})$. We denote these new tilings respectively, by $\text{ins}_{\mathcal{R}}(H, \mathcal{T})$ and $\text{ins}_{\mathcal{O}}(H, \mathcal{T})$. From the discussion above, it is clear that

$$\text{Grid}(\mathcal{T}) = \text{Grid}(\text{ins}_{\mathcal{O}}(H, \mathcal{T})) \sqcup \text{Grid}(\text{ins}_{\mathcal{R}}(H, \mathcal{T}))$$

where $\sqcup$ denotes the union of two sets.

This leads to the formal definition of the requirement insertion strategy. For any set of gridded permutations H, the strategy $\texttt{ReqIns}_H$ is defined as follows[1]

- If $\mathcal{T}$ is a tiling of dimension $t \times u$ and $H \subseteq \mathcal{G}^{(t,u)}$. Then

$$d_{\texttt{ReqIns}_H}(\mathcal{T}) = (\text{ins}_{\mathcal{O}}(H, \mathcal{T}), \text{ins}_{\mathcal{R}}(H, \mathcal{T})).$$

- The reliance profile function is $r_{\texttt{ReqIns}_H}(n) = (n, n)$.
- The counting functions are

$$c_{\texttt{ReqIns}_H,(n)}((a_0, \ldots, a_n), (b_0, \ldots, b_n)) = a_n + b_n.$$

Consider the tiling $\mathcal{T}^{(7)}$ from the example of Section 2.2 and the gridded permutation $g = (21, (0, 1))$. The tiling $\text{ins}_{\mathcal{O}}(g, \mathcal{T}^{(7)})$ is the tiling $\mathcal{T}^{(6)}$ on in the same example while $\text{ins}_{\mathcal{R}}(g, \mathcal{T}^{(7)})$ is the tiling $\mathcal{T}^{(11)}$. Therefore $d_{\texttt{ReqIns}_{\{g\}}}(\mathcal{T}^{(7)}) = (\mathcal{T}^{(6)}, \mathcal{T}^{(11)})$. Figure 2.4 shows the full decomposition.

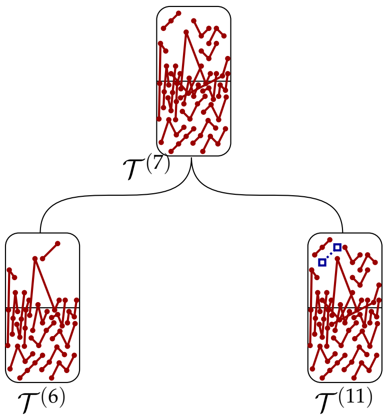


Figure 2.4: An application of the strategy $\texttt{ReqIns}_{\{(12,(0,1))\}}$.

This a typical example of strategy that breaks down a combinatorial set into a disjoint union of combinatorial sets. In general, these are strategies S such that $d_S(\mathcal{A}) = (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(m)})$ implies $\mathcal{A}_n = \mathcal{B}_n^{(1)} \sqcup \cdots \sqcup \mathcal{B}_n^{(m)}$ for all $n \in \mathbb{N}$. These strategies have the reliance function $r_S(n) = n \mapsto (n, \ldots, n)$ and counting functions $c_{S,(n)}$ of the form

$$c_{S,(n)}((b_0^{(1)}, \ldots, b_n^{(1)}), \ldots, (b_0^{(m)}, \ldots, b_n^{(m)})) = \sum_{i=1}^{m} b_n^{(i)}.$$

[1]Like in [34], we describe strategies by their action on tilings even though they actually act on the combinatorial sets that are the sets of griddable permutations.

These are more or less what are called a disjoint-union-type strategies in [34]. However, as we do not need to restrict ourself to having productive strategies, we can be a bit more loose with our definition of disjoint-union-type strategy. In particular, we do not need to restrict the definition to strategies with more than one child nor to strategies that only produce non-empty children.

It follows naturally from the reliance profile functions that such strategies are always regular. The constants $\mathfrak{C}_S^{(j)}$ are, in this case, zero. In particular, the requirement insertion strategy is regular.

Let say that a disjoint-union-type strategy S decomposes a set $\mathcal{A}$ into the sets $\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(m)}$. Then from the counting formulas we know that

$$|\mathcal{A}_n| = |\mathcal{B}_n^{(1)}| + \cdots + |\mathcal{B}_n^{(m)}|.$$

The last equation is naturally equivalent to

$$|\mathcal{B}_n^{(1)}| = |\mathcal{A}_n| - (|\mathcal{B}_n^{(2)}| + \cdots + |\mathcal{B}_n^{(m)}|).$$

Despite being a trivial algebraic manipulation, it shows that we can compute the number of elements of size n in $\mathcal{B}^{(1)}$ if we know the counting sequence of $\mathcal{B}^{(2)}, \ldots, \mathcal{B}^{(m)}$ and $\mathcal{A}$ up to size n. We can therefore have a strategy that decomposes $\mathcal{B}^{(1)}$ into $\mathcal{B}^{(2)}, \ldots, \mathcal{B}^{(m)}$ and $\mathcal{A}$. The child $\mathcal{B}^{(1)}$ does not play any particular role in the reasoning above. Therefore, such a strategy can actually be defined for any child.

- Let S be a disjoint-union type strategy. Let $\mathcal{T}$ be a tiling and i and integer. If $d_S(\mathcal{T}) = (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(m)})$ and $\mathcal{B}^{(i)} = \mathcal{A}$ then

$$d_{\texttt{RevDisjoint}_{S,\mathcal{T},i}}(\mathcal{A}) = (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(i-1)}, \mathcal{B}^{(i+1)}, \ldots, \mathcal{B}^{(m)}, \mathcal{T}).$$

  Otherwise,

$$d_{\texttt{RevDisjoint}_{S,\mathcal{T},i}}(\mathcal{A}) = \texttt{DNA}.$$

- The reliance profile function is

$$r_{\texttt{RevDisjoint}_{S,\mathcal{T},i}} = (n, \ldots, n).$$

- The counting functions are

$$c_{\texttt{RevDisjoint}_{S,\mathcal{T},i},(n)}((a_0^{(1)}, \ldots, a_n^{(1)}), \ldots, (a_0^{(m)}, \ldots, a_n^{(m)})) = a_n^{(m)} - \sum_{j=1}^{m-1} a_n^{(j)}.$$

On its own this strategy is not usable as it is too general but we can use it to define the reverse of some specific disjoint-union-type strategies. Precisely, we use it when a disjoint-union-type strategy creates a rule during a search to create all of the reverse version of the strategy and apply them to the children of the rule. We start with the requirement insertion strategy discussed above and introduce the reverse requirement insertion strategy `RevReqIns`. Formally, we define $\texttt{RevReqIns}_{H,\mathcal{T},i}$ as

$$\texttt{RevReqIns}_{H,\mathcal{T},i} = \texttt{RevDisjoint}_{\texttt{ReqIns}_H,\mathcal{T},i}$$

for any set of gridded permutations H, tiling $\mathcal{T}$ and integer $i \in \{1,2\}$. We can create a reverse version of the rule created above with requirement insertion. Consider the strategy $\texttt{RevReqIns}_{\{g\},\mathcal{T}^{(7)},1}$, since

$$d_{\texttt{ReqIns}_{\{g\}}}(\mathcal{T}^{(7)}) = (\mathcal{T}^{(6)}, \mathcal{T}^{(11)})$$

we have that

$$d_{\texttt{RevReqIns}_{\{g\},\mathcal{T}^{(7)},1}}(\mathcal{T}^{(6)}) = (\mathcal{T}^{(11)}, \mathcal{T}^{(7)})$$

telling us that

$$|\mathcal{T}_n^{(6)}| = |\mathcal{T}_n^{(7)}| - |\mathcal{T}_n^{(11)}|.$$

This rule is pictured in Figure 2.5.

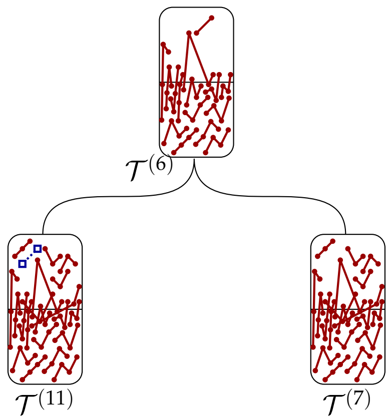


Figure 2.5: An application of reverse requirement insertion.

2.6.2 The factor strategy

As described in Section 6.3.5 of [34] the factor strategy is a strategy that aims to take apart a tiling into simpler parts. To do so, it identifies parts of the tiling that are not interacting and decomposes the tiling into those parts. Two sets of cells of a tiling, let call them S_1 and S_2, are non-interacting if there are no obstructions or requirement lists that interact with both a cell in S_1 and a cell in S_2 and a cell in the first set do not share a row or a column with a cell of the second set. Figure 2.6 shows tiling $\mathcal{T}^{(3)}$ from the example of Section 2.2 that has 5 mutually non-interacting parts. These parts are

$$\{(0,4),(0,6)\}, \quad \{(1,1)\}, \quad \{(2,3)\}, \quad \{(3,5)\} \quad \text{and} \quad \{(4,0),(4,2)\}.$$

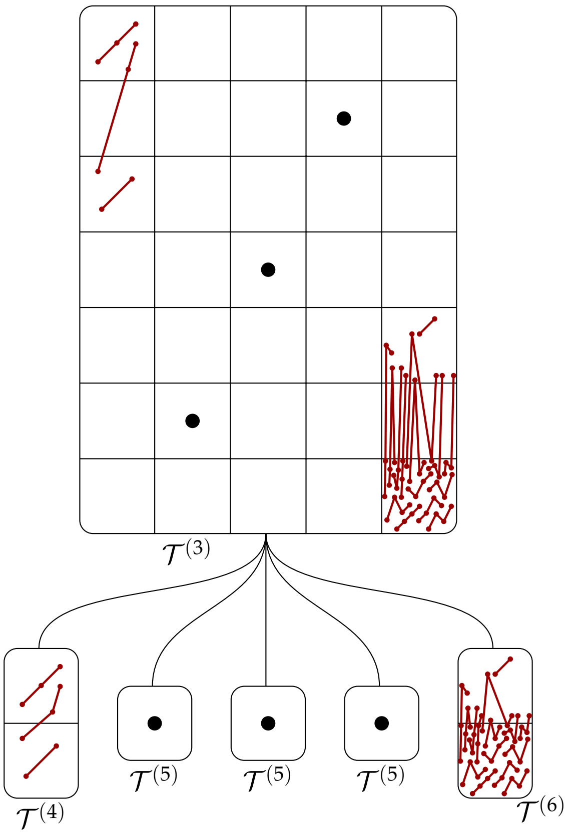


Figure 2.6: Example of a factor decomposition.

Since the parts are non-interacting, we can build uniquely the permutations of size n on the parent tiling by considering all ways of selecting one permutation on each child such that the sum of their sizes is n. To build the permutations on the parent, it is then sufficient to assemble these together according to the factoring partition. As observed in [34], this allows us to derive a counting formula. Assume a tiling $\mathcal{A}$ factors into k tilings $\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(k)}$. Then, we have that

$$|\mathcal{A}_n| = \sum_{i_1+\cdots+i_k=n} |\mathcal{B}^{(1)}_{i_1}| \cdots |\mathcal{B}^{(k)}_{i_k}|.$$

In [34], the authors are quick to point out that we can actually restrict some of the i_j to be strictly less then n and they then define the strategy $\texttt{Factor}_{P,S}$ where S is the set of such j such that i_j could be restricted to be strictly less than n in the counting formula. This leads to having $r^{(j)}_{\texttt{Factor}P,S}(n) = n - 1$ if j is in S and $r^{(j)}_{\texttt{Factor}P,S}(n) = n$ otherwise. This reliance profile function is regular which is practical for our purpose. We will, however, show that we can define the factor strategy slightly differently in order to obtain a "better" reliance function. In this context, better means having bigger constants $\mathfrak{C}^{(j)}_S$ which will make the rule more useful in the context of the enumerable subset as it will need less information from its children.

Consider a vector $M = (m_1, \ldots, m_k)$ where m_i is the size of the smallest gridded permutations in $\mathcal{B}^{(i)}$. Let m be the sum of the m_j. We therefore know that for a summand $|\mathcal{B}^{(1)}_{i_1}| \cdots |\mathcal{B}^{(k)}_{i_k}|$ to be non-zero we must have $m_j \leq i_j$ for $1 \leq j \leq k$ otherwise one of the multiplicand would be zero. If we combine that with the constraint that $i_1 + \cdots + i_k = n$ we obtain

$$i_1 + m_2 + m_3 + \cdots + m_k \leq n$$

or equivalently

$$i_1 \leq n - (m_2 + \cdots + m_k) = n - m + m_1.$$

We get similar constraints on the other i_j using the same reasoning. This allows us to rewrite the counting formula as

$$|\mathcal{A}_n| = \sum_{(i_1,\ldots,i_k)\in I_n} |\mathcal{B}^{(1)}_{i_1}| \cdots |\mathcal{B}^{(k)}_{i_k}|$$

where I_n is the set of size k partitions satisfying these inequalities, *i.e.*,

$$I_n = \{(i_1, \ldots, i_k) \in \{0, \ldots, n\}^k : \sum_{j=1}^{k} i_j = n \text{ and } i_j \leq n - m + m_j \text{ for } 1 \leq j \leq k\}.$$

We formally define the strategy $\texttt{Factor}_{P,M}$. The definition is mostly the same as $\texttt{Factor}_{P,S}$ in [34] with some minor modifications. Like in [34], the first parameter P is a partition of cells but here the second parameter is a tuple of integers and not a set of parts indices as in [34]. This change allows us to refine the reliance profile function as well as the counting functions in the way we described above.

- Let P be a partition of the non-empty cells of $\mathcal{T}$ and for concreteness consider the parts of P to be indexed in increasing order by their lexicographically smallest cell. If the cells of any part of P interact with the cells of any other part, then $\mathcal{T}$ cannot be factored according to this partition of cells. Thus, we assume that P is such that the parts are non-interacting, so that $\mathcal{T}$ will be factored into subtilings $\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(k)}$. Assume that each of these subtilings contains at least one gridded permutation. Let m_i be the size of the smallest gridded permutations in $\mathcal{B}^{(i)}$ and $M = (m_1, \ldots m_k)$. With such P and M we define
$$d_{\texttt{Factor}_{P,M}}(\mathcal{T}) = (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(k)}).$$
In the case where the partition does not match for $\mathcal{T}$, one of the $\mathcal{B}^{(i)}$ is empty or there is a mismatch with one of the m_i then we define
$$d_{\texttt{Factor}_{P,M}}(\mathcal{T}) = \texttt{DNA}.$$

- The reliance profile function of $\texttt{Factor}_{P,M}$ is
$$r_{\texttt{Factor}_{P,M}}(n) = (n - \mathfrak{C}^{(1)}_{\texttt{Factor}_{P,M}}, \ldots, n - \mathfrak{C}^{(k)}_{\texttt{Factor}_{P,M}})$$
where
$$\mathfrak{C}^{(i)}_{\texttt{Factor}_{P,M}} = \sum_{\substack{1 \le j \le k \\ j \neq i}} m_j$$
for $1 \le j \le k$.

- To describe the counting function, we first define vectors of indeterminates
$$b^{(i)} = (b^{(i)}_0, \ldots, b^{(i)}_{n - \mathfrak{C}^{(i)}_{\texttt{Factor}_{P,M}}})$$
for $1 \le i \le k$. The counting functions are
$$c_{\texttt{Factor}_{P,M},(n)}(b^{(1)}, \ldots, b^{(k)}) = \sum_{(i_1, \ldots, i_k) \in I_n} b^{(1)}_{i_1} \cdots b^{(k)}_{i_k},$$
where the sum is over
$$I_n = \{(i_1, \ldots, i_k) \in D_n \,:\, i_1 + \cdots + i_k = n\}$$
and
$$D_n = \{0, \ldots, n - \mathfrak{C}^{(1)}_{\texttt{Factor}_{P,M}}\} \times \ldots \times \{0, \ldots, n - \mathfrak{C}^{(k)}_{\texttt{Factor}_{P,M}}\}.$$

From the definition it is clear that the strategy is regular.

Assume that we have a tiling such that
$$d_{\texttt{Factor}_{P,M}}(\mathcal{T}) = (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(k)}).$$

Then, we know that

$$|\mathcal{T}_n| = \sum_{(i_1,\ldots,i_k)\in I_n} |\mathcal{B}^{(1)}_{i_1}| \cdots |\mathcal{B}^{(k)}_{i_k}|$$

where I_n is defined as above. For conciseness, let $s = \mathfrak{C}^{(1)}_{\mathtt{Factor}_{P,M}}$. One of the valid k-tuple of indices for the summation is $(n-s, m_2, m_3, \ldots, m_k)$. This corresponds to the case when we have as many points as possible in the factor $\mathcal{B}^{(1)}$. We can isolate the summand corresponding to the k-tuple to get

$$|\mathcal{B}^{(1)}_{n-s}| \cdot |\mathcal{B}^{(2)}_{m_2}| \cdots |\mathcal{B}^{(k)}_{m_k}| = |\mathcal{T}_n| - \sum_{(i_1,\ldots,i_k)\in I'_n} |\mathcal{B}^{(1)}_{i_1}| \cdots |\mathcal{B}^{(k)}_{i_k}|.$$

where I'_n is obtained by removing the k-tuple of indices mentioned above from I_n, *i.e.*,

$$I'_n = I_n \backslash \{(n-s, m_2, m_3, \ldots, m_k)\}.$$

By definition of $m_2, \ldots, m_k$, we have that $|\mathcal{B}^{(2)}_{m_2}|, \ldots, |\mathcal{B}^{(k)}_{m_k}$ are non-zero and therefore

$$|\mathcal{B}^{(1)}_{n-s}| = \frac{|\mathcal{T}_n| - \sum_{(i_1,\ldots,i_k)\in I'_n} |\mathcal{B}^{(1)}_{i_1}| \cdots |\mathcal{B}^{(k)}_{i_k}|}{|\mathcal{B}^{(2)}_{m_2}| \cdots |\mathcal{B}^{(k)}_{m_k}|}.$$

Note, the k-tuple $(n-s, m_2, \ldots, m_k)$ is the only one in I_n where the first component is greater than or equal to $n-s$. Hence, I'_n can also be defined as

$$I'_n = \{(i_1, \ldots, i_k) \in D'_n \,:\, i_1 + \cdots + i_k = n\}$$

where

$$D'_n = \{0, \ldots, n-s-1\} \times \{0, \ldots, n - \mathfrak{C}^{(2)}_{\mathtt{Factor}_{P,M}}\} \times \ldots \times \{0, \ldots, n - \mathfrak{C}^{(k)}_{\mathtt{Factor}_{P,M}}\}.$$

The only difference between D_n and D'_n is that, in the latter one, we lower the upper bound from the first set in the product from $n-s$ to $n-s-1$. Therefore, ensuring that the tuple $(n-s, m_2, \ldots, m_k)$ is not in I'_n.

If we shift the indices in the formula above by replacing n by $n+s$, we get

$$|\mathcal{B}^{(1)}_{n}| = \frac{|\mathcal{T}_{n+s}| - \sum_{(i_1,\ldots,i_k)\in I'_{n+s}} |\mathcal{B}^{(1)}_{i_1}| \cdots |\mathcal{B}^{(k)}_{i_k}|}{|\mathcal{B}^{(2)}_{m_2}| \cdots |\mathcal{B}^{(k)}_{m_k}|}.$$

This gives us a formula to compute $|\mathcal{B}^{(1)}_n|$ knowing the counting sequence of $\mathcal{T}$ up to size $n+s$, $\mathcal{B}^{(1)}$ up to size $n-1$, and $\mathcal{B}^{(j)}$ up to size $n + s - \mathfrak{C}^{(j)}_{\mathtt{Factor}_{P,M}}$ for $2 \le j \le k$.

The reasoning above has nothing in particular to do with $\mathcal{B}^{(1)}$ and could be done for any $\mathcal{B}^{(i)}$. We define a strategy called reverse factoring that computes the count of the i-th child of a factoring rule given the count of the parent and the children. We define the strategy $\mathtt{RevFactor}_{P,M,\mathcal{T},i}$ as follows.

- Let P be a partition of cells, M be a list of integers and $\mathcal{T}$ be a tiling such that $d_{\texttt{Factor}_{P,M}}(\mathcal{T}) = (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(k)})$. Then
$$d_{\texttt{RevFactor}_{P,M,\mathcal{T},i}}(\mathcal{A}) = (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(k)}, \mathcal{T})$$
if $\mathcal{A} = \mathcal{B}^{(i)}$. Otherwise,
$$d_{\texttt{RevFactor}_{P,M,\mathcal{T},i}}(\mathcal{A}) = \texttt{DNA}$$
- The reliance profile function of $\texttt{RevFactor}_{P,M,\mathcal{T},i}$ is
$$r_{\texttt{RevFactor}_{P,M,\mathcal{T},i}}(n) = (n - s_1, \ldots, n - s_{k+1})$$
where
$$s_j = \begin{cases} 1 & \text{if } j = i \\ -\mathfrak{C}^{(i)}_{\texttt{Factor}_{P,M}} & \text{if } j = k+1 \\ \mathfrak{C}^{(j)}_{\texttt{Factor}_{P,M}} - \mathfrak{C}^{(i)}_{\texttt{Factor}_{P,M}} & \text{otherwise.} \end{cases}$$
- To describe the counting function, we first define vectors of indeterminate
$$b^{(j)} = (b^{(j)}_0, \ldots, b^{(j)}_{n-s_j}).$$
The counting functions are
$$c_{\texttt{RevFactor}_{P,M,\mathcal{T},i},(n)}(b^{(1)}, \ldots, b^{(k+1)}) = \frac{b^{(k+1)}_{n-s_{k+1}} - \sum_{(i_1,\ldots,i_k)\in I_n} b^{(1)}_{i_1} \cdots b^{(k)}_{i_k},}{b^{(1)}_{m_1} \cdots \widehat{b^{(i)}_{m_i}} \cdots b^{(k)}_{m_k}}$$
where the sum is over
$$I_n = \{(i_1, \ldots, i_k) \in D_n \,:\, i_1 + \cdots + i_k = n - s_{k+1}\}$$
and
$$D_n = \{0, \ldots, n - s_1\} \times \ldots \times \{0, \ldots, n - s_k\}.$$

It is clear from the definition that the strategy is regular and the constant shifts are the s_j. Formally,
$$\mathfrak{C}^{(j)}_{\texttt{RevFactor}_{P,M,\mathcal{T},i}} = s_j$$
for $1 \leq j \leq k+1$.

Also note that, since the decomposition gives
$$d_{\texttt{RevFactor}_{P,M,\mathcal{T},i}}(\mathcal{A}) = (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(k)}, \mathcal{T}),$$
then $\mathcal{A}$ is both the parent and a child of the decomposition. At first sight this might look like it makes the rule is completely pointless. It is, however, not the case and a great example of the importance of the reliance profile function of a strategy. The tiling $\mathcal{A}$ is the i-th tiling in the decomposition and $r^{(i)}_{\texttt{RevFactor}_{P,M,\mathcal{T},i}}(n) = n - 1$. Therefore, it simply means that to compute $|\mathcal{A}_n|$ we first need to know the terms that come before it in the counting sequence, *i.e.*, $|\mathcal{A}_0|, \ldots, |\mathcal{A}_{n-1}|$.

Figure 2.7 shows an example of a reverse factor rule where the child $\mathcal{T}^{(7)}$ of the regular rule becomes the parent of the reverse rule.

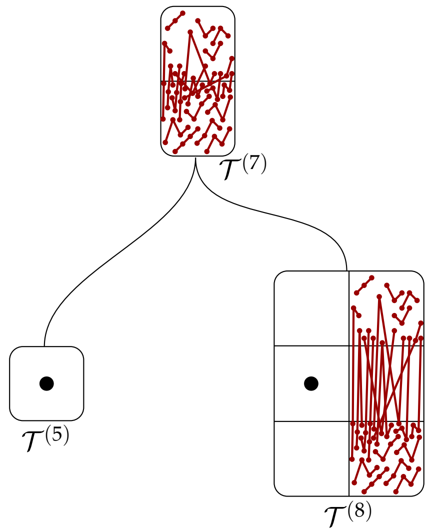


Figure 2.7: An example of reverse factor rule.

2.6.3 The point placement strategy

The point placement strategy is a strategy that isolates a point of a requirement containing a single gridded permutation and forces it to be the most extreme point of that gridded permutation in one of four directions. The point can either be the topmost, the bottommost, the leftmost or the rightmost one. The strategy is defined rigorously and in full generality in Section 6.3.3 and 6.5.3 of [34]. We will here introduce it via a comprehensive example and observe that the strategy is regular. Then we will discuss the reverse version of the strategy.

Consider the one-by-one tiling with the obstruction $\mathcal{O}_1 = (123, (0,0))$ and the requirement list $\mathcal{R}_1 = \{(12, (0,0))\}$. Let place the leftmost 1 in this 12. To isolate the 1, we subdivide the tiling into a 3×3 tiling and put a point in the middle cell. To ensure that the point is fully isolated in its own row and column, we add point obstructions in cells $(0,1)$, $(1,0)$, $(2,1)$ and $(1,2)$. We also add 12 and 21 obstructions in cell $(1,1)$ to ensure the cell contains only one point. To force this point to be the leftmost 1 in any 12, we add the obstructions where the underlying permutation is 12 and the 1 is in a cell of the left column. Concretely, we add $(12, (0,0))$, $(12, (0,2))$, $(12, (0,0), (0,2))$, $(12, (0,0), (1,1))$, $(12, (0,0), (2,0))$, $(12, (0,0), (2,2))$ and $(12, (0,2), (2,2))$. Finally, we add 123 obstructions in all ways so that any permutation griddable on the tiling still avoids 123. We obtain the second tiling in Figure 2.8. For readability, we have only drawn the non-redundant 123 obstructions. We observe that containing the requirement $\{(12), ((1,1), (2,2))\}$ is equivalent to having a point in cell $(1,1)$ and a point in cell $(2,2)$. Therefore, we can replace the requirements by two point requirements: $\{(1, (1,1))\}$ and $\{(1, (2,2))\}$. This produces the third tiling of Figure 2.8. Finally, since we know that any permutation griddable on the tiling contains a point in cell $(1,1)$, we have that any gridded permutation that avoids $(12, ((0,0),(1,1)))$ actually avoids $(1,(0,0))$ and that any gridded permutation avoiding $(123, ((1,1),(2,2),(2,2)))$ actually avoids $(12,(2,2))$. This is a special case of obstruction inferral presented in Section 6.3.6 of [34]. We add those obstructions and simplify the redundant ones to get the fourth tiling of Figure 2.8.

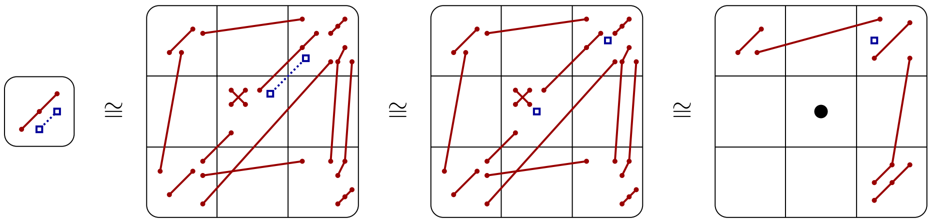


Figure 2.8: Placing the leftmost 1 in a 12 on a 123-avoiding tiling.

The exact strategy that places the point in the way described above is

$$\texttt{PointPl}_{(12,(0,0)),1,\leftarrow}.$$

Meaning that we place the first point of $(12,(0,0))$ in the leftmost direction. In general if a tiling $\mathcal{T}$ contains a requirement $\{h\}$, the index i is the index of a point in h to place and d is one of the directions in $\{\leftarrow,\uparrow,\downarrow,\rightarrow\}$, the function $d_{\texttt{PointPl}_{h,i,d}}$ is defined similarly. As mentioned previously, we skip here the formal definition of the point placement strategy as it is highly technical and not relevant to our aim. However, a curious reader will find all the details in Section 6.5.3 of [34]. In particular, Theorem 6.8 in that paper shows that there is a size-preserving bijection between the griddable permutations on $\mathcal{A}$ and $\mathcal{B}$ if $d_{\texttt{PointPl}_{h,i,d}}(\mathcal{A}) = \mathcal{B}$. This bijection shows that the reliance function is $r_{\texttt{PointPl}_{h,i,d}}(n) = (n)$ and that the counting functions are $c_{\texttt{PointPl}_{h,i,d},(n)}((a_0,\ldots,a_n)) = a_n$

As observed in [34], point placement is not a productive strategy. In fact the decomposition function gives the decomposition $\mathcal{A} \leftarrow (\mathcal{B})$ such that $|\mathcal{A}_n|$ relies on $|\mathcal{B}_n|$ and $|\mathcal{A}_n| = |\mathcal{B}_n|$ for all $n \in \mathbb{N}$. It, therefore, fails condition 2b of the productivity definition (Definition 2.3). Despite not being productive these strategies can still be used with the prune method if they receive the special treatment used for equivalence strategies.

The point placement strategies are however, disjoint-union-type strategies as defined in Section 2.6.1. This strategy is therefore regular and can be used in our framework without any special treatment. We can also introduce the reverse version of that strategy which can be thought of as unplacing the point. Formally,

$$\texttt{RevPointPl}_{h,i,d,\mathcal{T}} = \texttt{RevDisjoint}_{\texttt{PointPl}_{h,i,d},\mathcal{T},1}.$$

Since point placement only has one child, it is not necessary to carry the index of the child in the definition of the reverse point placement strategy.

Figure 2.9 show the results of applying the strategy $\texttt{PointPl}_{(1,(0,0)),1,\leftarrow}$ to the tiling of the left. This strategy is used to decompose the tiling $\mathcal{T}^{(9)}$ into $(\mathcal{T}^{(8)})$.

This decomposition is used in one of the trees of Figure 2.1. The reverse version can also be used to decomposed $\mathcal{T}^{(8)}$ into $(\mathcal{T}^{(9)})$. In this case the strategy used is $\texttt{RevPointPl}_{(1,(0,0)),1,\leftarrow,\mathcal{T}^{(9)}}$. This is the strategy used for the decomposition of $\mathcal{T}^{(8)}$ when we merge the two trees of Figure 2.1 into a single specification in Figure 2.2.

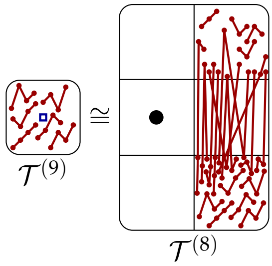


Figure 2.9: Placing the leftmost point in Av(1234, 1243, 1324, 1423, 2134, 2314).

2.6.4 Other strategies

In [34], the authors also discussed a series of other strategies covering obstruction and requirement simplification, row and column separation and obstruction inferral. All those strategies, like point placement, observe that two combinatorial sets are actually equinumerous. They are therefore also disjoint-union-type strategies with a single child. It follows that they are regular and we can derive reverse strategies from them in a similar fashion as we did for `RevPointPl`.

2.6.5 Applying reverse strategies in a search

Throughout Section 2.6, we revisited the strategies from [34] and showed that they were all regular. These strategies, therefore, produce rules that can be processed by Algorithm 2. We also defined strategies that allow us to decompose a child of a rule produced by the aforementioned strategy into the parent and children of that rule. The way we have defined them, these strategies almost never apply. In fact, the decomposition function always returns `DNA` except when the input tiling is a specific child of a specific rule produced by the non-reverse version of that strategy. For example the strategy $\texttt{RevReqIns}_{H,\mathcal{T},i}$ only applies to a tiling $\mathcal{A}$ if $\mathcal{A}$ is the i-th component of the decomposition function $d_{\texttt{ReqIns}_H}$ applied to $\mathcal{T}$. In practice, it is, therefore, highly impractical to try to apply those reverse strategies to arbitrary tilings. What we actually do is run combinatorial exploration as with the prune method. We use the same queueing system and apply only non-reverse strategies. Then, when a strategy applies and creates a rule, we also add immediately all the possible reversed versions into the universe. Consider, as an example, applying the strategy $\texttt{ReqIns}_{(21,(0,1))}$ to tiling $\mathcal{T}^{(7)}$ from Section 2.2. This gives us the rule $\mathcal{T}^{(7)} \leftarrow (\mathcal{T}^{(6)}, \mathcal{T}^{(11)})$ illustrated on Figure 2.4. As soon as this rule is added to the universe, we can immediately add the rule $\mathcal{T}^{(6)} \leftarrow (\mathcal{T}^{(11)}, \mathcal{T}^{(7)})$ obtained by applying the strategy $\texttt{RevReqIns}_{(21,(0,1)),\mathcal{T}^{(7)},1}$ to $\mathcal{T}^{(6)}$ (see Figure 2.5) and the rule $\mathcal{T}^{(11)} \leftarrow (\mathcal{T}^{(6)}, \mathcal{T}^{(7)})$ obtained by applying the strategy $\texttt{RevReqIns}_{(21,(0,1)),\mathcal{T}^{(7)},2}$ to tiling $\mathcal{T}^{(11)}$ (see Figure 2.10).

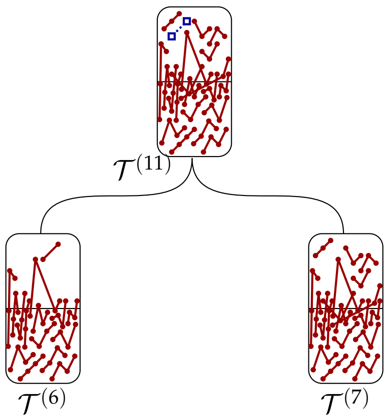


Figure 2.10: A reverse requirement insertion rule where the parent is the second child of the regular rules.

Adding the reverse versions of the rules to the universe allows us to make extra connections between the combinatorial sets and find specifications when one could not be found with only the forward versions of the rules. As we discussed in Section 2.2, the universe consisting of the rules in the two partial specification of Figure 2.1 does not contain a specification. However, when we add the reverse versions of these rules, the universe contains the specification that is pictured in Figure 2.2. There lies the power of the method developed in this chapter. By developing a new specification searcher algorithm that works with regular strategies, we allow ourself to use these reverse strategies that cannot be used with the prune method since they are not productive. This creates more connections in the universe and in the end leads to finding more specifications.

Chapter 3

Combinatorial exploration with catalytic variables

In the previous chapter, the strategies we explored for tilings relied on two counting formulas. Those were the disjoint union and Cartesian product constructors. They gave generating function equations that could be expressed using only sum operators or only product operators, respectively. With these types of equations there is a guarantee that the combinatorial sets in a specification have an algebraic generating function. In this chapter, we introduce a new strategy called *fusion* that has a more complex generating function equation. We see that defining the counting functions for this strategy requires tracking some extra statistics on the gridded permutations. This leads to the introduction of tracked combinatorial sets as well as to a refinement of the definition of a counting sequence. We explore how the definition of a strategy can be extended to accommodate for tracked combinatorial sets and illustrate how the machinery developed in Chapter 2 can still be used in this case. We conclude the chapter with the first direct enumeration of the permutation class Av(1342). The specification found uses both the power of reverse strategies developed in the previous chapter as well as the fusion strategy that we will introduce here. This chapter aims to present the concepts and give the reader intuition on how combinatorial exploration can be extended to work with catalytic variables. It avoids going into technicalities and tedious proofs and focuses on giving illustrative examples.

3.1 The fusion strategy

The fusion strategy is a strategy that merges two adjacent rows or columns of the tiling into a single one under certain conditions. For ease of presentation, we only discuss here fusing two columns but the row case is symmetric.

We start by considering the simplest case of the 2×1 tiling $\mathcal{A}$ presented in Figure 3.1. The key property we find on this tiling is that there is no way a permutation gridded on this tiling can contain a 12 pattern. Suppose a permutation σ contained an occurrence of 12. Once σ is gridded on the two cells the occurrence has either two points in the left cell, two points in the right cell or its 1

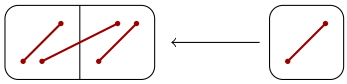


Figure 3.1: Simplest example of a fusion rule.

in the left cell and its 2 in the right cell. Each of these cases contain one of the obstructions on $\mathcal{A}$, respectively the obstruction $12, ((0,0),(0,0))$, $12, ((1,0),(1,0))$ and $(12, ((0,0),(1,0)))$. Hence, we can, in some sense, say that the whole 2×1 tiling avoids 12. Another way to think of it is that all the possible griddings of the permutation 12 are obstructions on the tiling $\mathcal{A}$.

On the other hand, any permutation σ that avoids 12 can be gridded on the tiling as it will never contain any of the obstructions. It can, in fact, be gridded in any way on $\mathcal{A}$. Given a 12-avoiding permutation σ we can "cut" it anywhere, puttin the points to the left of the cut in the left cell and the points to the right of the cut in the right cell, giving a valid gridding of σ on $\mathcal{A}$. If σ is of size n, there are $n+1$ ways to cut it (it is also possible to cut at either end of the permutation). For example, the four possible griddings of 321 are

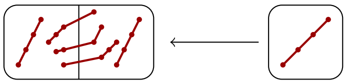
$$(321, ((0,0),(0,0),(0,0))),$$
$$(321, ((0,0),(0,0),(1,0))),$$
$$(321, ((0,0),(1,0),(1,0))), \text{ and}$$
$$(321, ((1,0),(1,0),(1,0))).$$

In the paragraph above, we considered the link between gridded permutations on $\mathcal{A}$ and permutations in $\mathrm{Av}(12)$. For combinatorial exploration and the following exposition, it is however much more convenient to stay in the world of tilings and represent $\mathrm{Av}(12)$ by the tiling $\mathcal{B}$ on the right in Figure 3.1. The correspondence described above, therefore, becomes a correspondence between gridded permutations on $\mathcal{B}$ and gridded permutations on $\mathcal{A}$. Figure 3.1 is our first example of a decomposition produced by a fusion strategy. In the context of such a decomposition, we identify $\mathcal{A}$ as the *unfused tiling*, $\mathcal{B}$ as the *fused tiling* and will say that $\mathcal{A}$ *fuses* to $\mathcal{B}$.

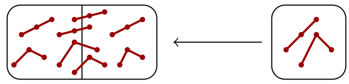
Each permutation of size n on $\mathcal{B}$ corresponds to $n+1$ gridded permutations of size n on $\mathcal{A}$. If a_n and b_n represent respectively the number of gridded permutation of size n griddable on $\mathcal{A}$ and $\mathcal{B}$, then

$$a_n = (n+1)b_n. \tag{3.1}$$

The relation described above between the gridded permutations on the fused and the unfused tiling is not exclusive to having 12 crossing in all ways. It applies, in general, if a longer pattern is crossing in all ways. Such an example can be seen in Figure 3.2 (a). In this case, the longer pattern is 1234 that crosses in all 5 possible ways. Moreover, this tiling fuses to the 1×1 tiling avoiding 1234. It is also possible to have more than one pattern crossing in all ways as shown in the example of Figure 3.2 (b). There, we see an example of having both 132 and 123 permutations gridded in all possible ways on the 2×1 tiling. Since the unfused tiling avoids both 132 and 123 the fused tiling has two obstructions, one for each of the patterns. In all cases, we still have that any permutation of size n on the fused tiling corresponds to $n+1$ gridded permutations on the unfused tiling and the counting sequence of the unfused tiling can therefore be obtained using Equation (3.1).



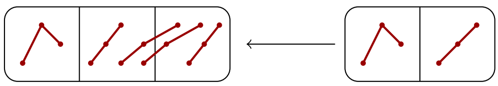
(a)



(b)

Figure 3.2: (a) Fusion with a longer pattern. (b) Fusion with more than one pattern.

Lets now examine what happens if we add an extra column to a 2×1 that could be fused as described above. We consider the tiling $\mathcal{A}$ from Figure 3.3. On their own, the two rightmost cells globally avoid 123. Based on our previous



$\mathcal{A}$ $\mathcal{B}$

Figure 3.3: Two columns of a three column tiling fusing together.

examples, it is sensible to assume that it can be merged into a single cell avoiding 123. This gives us the tiling $\mathcal{B}$ in the same figure. Comparing the terms of the counting sequences of $\mathcal{A}$ and $\mathcal{B}$ it is however clear that the counting formula from Equation 3.1 does not work here. In fact, for size 1 gridded permutations, we find two of them griddable on $\mathcal{B}$ while three are griddable on $\mathcal{A}$. Equation (3.1) however predicts $(1+1) \cdot 2 = 4$ gridded permutations on $\mathcal{A}$. The two gridded permutations of size 1 on $\mathcal{B}$ are $\pi_0 = (1, (0,0))$ and $\pi_1 = (1, (1,0))$ while the three on $\mathcal{A}$ are $\sigma_0 = (1, (0,0))$, $\sigma_1 = (1, (1,0))$ and $\sigma_2 = (1, (2,0))$. Note that π_1 is in the cell that comes from the fusion of two cells while π_0 is in the cell that remains unchanged. Hence, we have two gridded permutations that correspond to π_1, namely σ_1 and σ_2 and one that corresponds to π_0, namely σ_0.

Lets formalize this correspondence between gridded permutations on $\mathcal{A}$ and on $\mathcal{B}$. We define the operator FuseCol_i on gridded permutations to represent fusing together columns i and $i+1$. Applying FuseCol_i to a gridded permutation shifts all the column indices that are greater than i by one to the left. For example, if we fuse column 1 and 2 of

$$(1234, ((0,0), (1,0), (2,0), (3,0))),$$

we obtain

$$(1234, ((0,0), (1,0), (1,0), (2,0))).$$

The reader should think of the FuseCol_i map as a map that merges the columns i and $i+1$ of a gridded permutation into a single one.

Coming back to the three permutations of size 1 griddable on $\mathcal{A}$, we have that

$$\begin{aligned}\text{FuseCol}_1(\sigma_0) &= \pi_0,\\ \text{FuseCol}_1(\sigma_1) &= \pi_1, \text{ and}\\ \text{FuseCol}_1(\sigma_2) &= \pi_1.\end{aligned}$$

Here, we use FuseCol_1 since it is column 1 and 2 of $\mathcal{A}$ that are merging in to a single column to produce $\mathcal{B}$. Since the two columns being fused globally avoid 123, the FuseCol_1 operator, in fact, matches every valid gridded permutation on $\mathcal{A}$ to a valid gridded permutation on $\mathcal{B}$. Table 3.1 presents some of the valid gridded permutations on $\mathcal{A}$ and their images under FuseCol_1. For each of the images in the left column, the right column contains all the preimages griddable on $\mathcal{A}$. One key observation is that we can easily recreate all the preimages of a gridded permutation. Since the image of a gridded permutation is created by merging columns 1 and 2 together into column 1, we can recreate all the preimages by splitting column 1 in all possible ways. Each way of splitting creates a valid gridded permutation on $\mathcal{A}$ since we start with a gridded permutation that avoids 123 in column 1. If a gridded permutation has k points in column 1, there is $k+1$ ways to do the splitting since we can shift between the 0 and k of these points to column 2 while keeping the others in column 1.

π	$\text{FuseCol}_1(\pi)$
$(123,((0,0),(0,0),(0,0)))$	$(123,((0,0),(0,0),(0,0)))$
$(132,((0,0),(0,0),(1,0)))$ $(132,((0,0),(0,0),(2,0)))$	$(132,((0,0),(0,0),(1,0)))$
$(231,((1,0),(1,0),(1,0)))$ $(231,((1,0),(1,0),(2,0)))$ $(231,((1,0),(2,0),(2,0)))$ $(231,((2,0),(2,0),(2,0)))$	$(231,((1,0),(1,0),(1,0)))$

Table 3.1: Correspondence between valid gridded permutations on the tilings $\mathcal{A}$ and $\mathcal{B}$ via the FuseCol_1 map.

The FuseCol_i function gives us a connection where if we know the valid gridded permutations on $\mathcal{B}$, then we can quickly compute the valid gridded permutations on $\mathcal{A}$. Since we are mostly interested in counts, we can however simplify the matter a bit. We have already noticed that what determines the number of preimages of each gridded permutation is the number of points in the column being fused (column 1 in our example). Therefore, if we have $b_{n,k}$ valid gridded permutations on $\mathcal{B}$ that are of size n with k points in column 1 then they give us $(k+1)b_{n,k}$ valid gridded permutations on $\mathcal{A}$. To get the total number of valid gridded permutations on $\mathcal{A}$, we then only have to sum over all possible k. If a_n is the number of valid gridded permutations of size n on $\mathcal{A}$ then

$$a_n = \sum_{k=0}^{n} (k+1) b_{n,k}. \tag{3.2}$$

Note that the formula is a generalisation of Equation (3.1). In fact, in that case the fused tiling only had one column. Therefore, the size of a valid gridded permutation is always the same as the number of points it has in the columns being fused. Hence, $b_{n,k} = 0$ if $k \neq n$ and the formula becomes $a_n = (n+1)b_{n,n}$. The same counting formula can be used even if some obstructions are crossing between the two fusing columns and the rest of the tiling (as long as they act in the same way in both fusing columns). Figure 3.4 shows an example of that generalisation. The key observation is that the three 123 obstructions represent all the ways to have a 1 in the 132 avoiding cell with a 23 in the fusing columns. They are, therefore, equivalent to the single 123 obstruction drawn on the fused tiling $\mathcal{B}$.

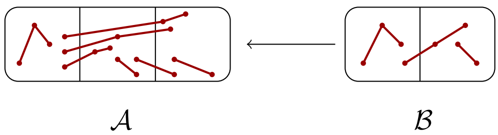


Figure 3.4: A 123 crossing having its 23 in all possible way in the two fusing columns.

One detail we have not covered yet is how to obtain the fused tiling from the original tiling. We use again here the $\mathrm{FuseCol}_i$ operator to define it formally. We first extend the function naturally to sets of gridded permutations such that the image of a set of gridded permutations is the set of the images of the gridded permutations under the map. Mathematically,

$$\mathrm{FuseCol}_i(S) = \{\mathrm{FuseCol}_i(\pi) : \pi \in S\}.$$

We then extend it to tilings by making it fuse all obstructions and requirement sets. Precisely, for a tiling $\mathcal{T} = ((t,u), \mathcal{O}, \{\mathcal{R}_1, \ldots, \mathcal{R}_k\})$ and $0 \leq i < t-1$,

$$\mathrm{FuseCol}_i(\mathcal{T}) = ((t-1,u), \mathrm{FuseCol}_i(\mathcal{O}), \{\mathrm{FuseCol}_i(\mathcal{R}_1), \ldots, \mathrm{FuseCol}_i(\mathcal{R}_k)\}).$$

Figure 3.1 and 3.2 are examples of applying $\mathrm{FuseCol}_0$ while Figure 3.3 and 3.4 display examples of applying $\mathrm{FuseCol}_1$. All of these examples were carefully chosen so that the obstructions were crossing in all ways and, therefore, the number of preimages of $\mathrm{FuseCol}_i$ was always determined by the number of points in the fused column. This is not always the case. Consider the two examples of applying $\mathrm{FuseCol}_1$ in Figure 3.5. In both cases, the counting sequence of the bigger tiling cannot be obtained from the counting sequence of the smaller tiling using Equation (3.2). In the topmost example, we see that the gridded permutation $(123, ((1,0),(2,0),(2,0)))$ is griddable on the left tiling while its image under fuse $\mathrm{FuseCol}_1$ is $(123, ((1,0),(1,0),(1,0)))$ cannot be gridded on the right tiling since

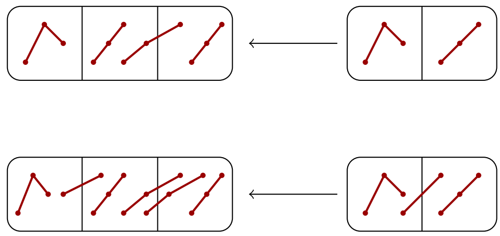


Figure 3.5: Examples of applying the FuseCol_1 map that lead to invalid rules.

it contains one of the obstructions. There is a similar issue with $(12,((0,0),(0,2)))$ in the bottommost example. The general criteria to check if the map is producing a tiling that would be the child of a valid fusion rule is to check if each obstruction on the produced tiling has all of its preimages present on the original tiling.

3.2 Tracked combinatorial sets

One important detail we have not yet addressed is that the counting function of the fusion strategy requires a refinement of the counting of gridded permutations by the number of points in a region. It is not sufficient to only know the number of gridded permutations griddable on the fused tiling, we actually need to know how many gridded permutations of size n have k points in the column that comes from the fusion of the two columns.

To handle that in the combinatorial exploration framework, we need to refine our notion for combinatorial sets to support the counting of extra statistics on the combinatorial objects. As stated in Definition 1.1, a combinatorial set is a set with a size function with the property that there are finitely many objects of each size. It is here useful to think of a combinatorial set axiomatically as a pair $(\mathcal{A}, |\cdot|)$ where $\mathcal{A}$ is a set and $|\cdot|$ is the size function. We define a *tracked combinatorial set* as an extension of a combinatorial set with extra statistic functions $s_1, \ldots, s_m$. The statistic functions have to be functions from the set to non-negative integers. It could, for example, be a function s that given a gridded permutation π returns the number of points of π that are in cell $(0,0)$. In fact, statistic functions are really similar to the size function but do not have the restriction of having a finite number of objects mapping to each value. We think of a tracked combinatorial set as a tuple $(\mathcal{A}, |\cdot|, s_1, \ldots, s_m)$. Formally:

Definition 3.1. *A* tracked combinatorial set $\mathcal{C}$ *is a combinatorial set on which statistic functions are defined. Each statistic function must be a function from the set $\mathcal{C}$ to non-negative integers.*

One very simple example of a tracked combinatorial set is the set of words on the alphabet $\{1,2,3\}$ with the usual size function and two statistic functions tracking the number of 1s and the number of 2s in the word. A slightly more complex (and more relevant for us) example is the set of gridded permutations on a tiling $\mathcal{T}$ with the usual size function and a statistic function that counts the number of points in a given column of a tiling. One could also track different statistics on a tiling, for example the number of inversion. In this text, we will however restrict ourselve to tracking the number of points in regions of a tiling. To formalize the point tracking idea, we introduce a tracked tiling. Tracked tilings are similar to tilings as they have the same first three components, *i.e.*, dimensions, obstructions and requirements. They however have a fourth component which is a tuple of sets of cells that we call the *tracked regions*. Each of these sets represent a region in which we count how many points of each gridded permutation it contains.

Definition 3.2. *A* tracked tiling *is a quadruple* $\mathcal{T} = ((t,u), \mathcal{O}, \mathcal{R}, (G_1, \ldots, G_m))$ *where* (t,u)*,* $\mathcal{O}$*, and* $\mathcal{R}$ *are defined as for tilings and* $G_1, \ldots, G_m$ *are sets of cells of the tiling. Each of the sets* $G_1, \ldots, G_m$ *is called a* tracked region*.*

For a set of cells G, we define s_G as the function that counts the number of points of a gridded permutation (π, P) that are in the cells in the tracked region G. Formally,

$$s_G((\pi,(c_1,\ldots,c_n))) = |\{i : 1 \le i \le n \text{ and } c_i \in G\}|.$$

For any tracked tiling $\mathcal{T} = ((t,u), \mathcal{O}, \mathcal{R}, (G_1, \ldots, G_m))$, we can directly associate a tracked combinatorial set

$$(\mathrm{Grid}(\mathcal{T}), |\cdot|, s_{G_1}, \ldots, s_{G_m}).$$

As with tilings, we will, for convenience, often refer to a tracked tiling as a combinatorial set when it actually means tracked combinatorial as set defined above.

We picture tracked tilings in the same way we picture tilings and use shadings of different colours over each of the regions. Figure 3.6 shows an example of a tiling $\mathcal{T} = ((3,1), \mathcal{O}, \mathcal{R}, (\{(1,1),(2,0)\}))$. This representation is not perfect as it can be hard to represent multiple regions G_i simultaneously (especially if they have some cells in common) and it also does not reflect the order of the sets. This representation is, however, sufficient for our use case since we will only use one tracked region in the upcoming examples.

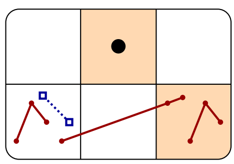


Figure 3.6: Pictorial representation of a tracked tiling.

As for combinatorial sets, we defined $\mathcal{A}_n$ as the set of objects of size n in a tracked combinatorial set $\mathcal{A}$. The counting sequence of the set was then defined as the sequence of integers representing the number of objects of each size in $\mathcal{A}$ (see Definition 1.2). For tracked combinatorial sets, we want the counting sequences to reflect the extra structure brought by the statistic functions. We, therefore, define the n-th term of the counting sequence of a tracked combinatorial set

$(\mathcal{A}, |\cdot|, s_1, \ldots, s_m)$ as

$$\|\mathcal{A}_n\| = \sum_{a \in \mathcal{A}_n} {y_1}^{s_1(a)} \cdots {y_m}^{s_m(a)}.$$

Each term of the counting sequence is, therefore, a polynomial in $y_1, \ldots, y_m$. Formally, we have

Definition 3.3. *The* counting sequence *of a tracked combinatorial set $\mathcal{A}$ is the sequence of polynomials with integer coefficients* $(\|\mathcal{A}_n\|)_{n \geq 0}$.

Let consider the tracked tiling $\mathcal{T}$ of Figure 3.6. The smallest permutation griddable on the tiling is $\pi = (213, ((0,0), (0,0), (1,1)))$. The counting sequence, therefore, starts $0, 0, 0$ as there are no griddable permutations of size 0, 1 or 2 on $\mathcal{T}$. The size 3 gridded permutation π is the only one griddable on $\mathcal{T}$ and it has one point in the tracked region. Therefore, the next term of the counting sequence is y_1. Table 3.2 shows the gridded permutations of size 4 griddable on $\mathcal{T}$ as well as their contribution to the next term of the counting sequence. It shows that the next term is $3{y_1}^2 + 4y_1$. In the case where there is only one statistic, we often replace y_1 by y to simplify notation. In this example it gives us the counting sequence

$$0, 0, 0, y, 3y^2 + 4y, 9y^3 + 16y^2 + 13y, ...$$

Gridded permutation	Contribution to term
$(2130, ((0,0), (0,0), (1,1), (2,0)))$	${y_1}^2$
$(1032, ((0,0), (0,0), (1,1), (2,0)))$	${y_1}^2$
$(1023, ((0,0), (0,0), (0,0), (1,1)))$	${y_1}^1$
$(1203, ((0,0), (0,0), (0,0), (1,1)))$	${y_1}^1$
$(2031, ((0,0), (0,0), (1,1), (2,0)))$	${y_1}^2$
$(2013, ((0,0), (0,0), (0,0), (1,1)))$	${y_1}^1$
$(2103, ((0,0), (0,0), (0,0), (1,1)))$	${y_1}^1$

Table 3.2: Gridded permutations of size 4 on $\mathcal{T}$ and their contributions to the fifth term of the counting sequence.

It is good to note that if a tracked combinatorial set has zero statistic functions then the counting sequence coincides with the definition of (non-tracked) combinatorial sets as the formula for each term just becomes

$$\sum_{a \in \mathcal{A}_n} 1.$$

Moving forward, we will think of all combinatorial sets as being tracked as we can think of (non-tracked) combinatorial sets as tracked combinatorial sets with no statistic functions.

The notion of a strategy introduced in Chapter 2 can easily be extended to accommodate tracked combinatorial sets. In fact, we only need a small change to the notion of a counting function to compute the counting sequences refined by statistics. We simply replace the integers by polynomials while the rest stays the same.

Definition 3.4. *Let $\mathcal{Z}$ be the collection of all tracked combinatorial sets. An m-ary* tracked combinatorial strategy *S consists of three components.*

1. *A* decomposition function $d_S : \mathcal{Z} \to \mathcal{Z}^m \cup \{\texttt{DNA}\}$ *whose input is a tracked combinatorial set $\mathcal{A}$ (the parent set), and whose output is either an ordered m-tuple of tracked combinatorial sets $(\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(m)})$ (the child sets) or the symbol* DNA. *When the output is $d_S(\mathcal{A}) = \texttt{DNA}$, short for "does not apply", we say that S* cannot be applied *to the combinatorial set $\mathcal{A}$.*

2. *A* reliance profile function $r_S : \mathbb{N} \to \mathbb{Z}^m$ *whose input is a natural number n and whose output is an ordered m-tuple of integers. We use $r_S^{(i)}(n)$ to denote the i-th component of $r_S(n)$, i.e.,*

$$r_S(n) = (r_S^{(1)}(n), \ldots, r_S^{(m)}(n)).$$

3. *An infinite sequence of* counting functions $c_{S,(n)}$ *indexed by $n \in \mathbb{N}$, each of whose input is m tuples of polynomials with integer coefficients $w^{(1)}, \ldots, w^{(m)}$ and whose output is a polynomial with integer coefficients. The counting functions must have the property that if $d_S(\mathcal{A}) = (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(m)})$ and $r_S(n) = (r_S^{(1)}(n), \ldots, r_S^{(m)}(n))$, then for input tuples*

$$w^{(i)}(n) = \left(\|\mathcal{B}_0^{(i)}\|, \ldots, \|\mathcal{B}^{(i)}_{r_S^{(i)}(n)}\| \right)$$

we have

$$c_{S,(n)}(w^{(1)}(n), \ldots, w^{(m)}(n)) = \|\mathcal{A}_n\|.$$

To be overly explicit, the domain of $c_{S,(n)}$ is

$$\mathbb{N}[y_1, \ldots, y_{\ell_1}]^{D_1} \times \cdots \times \mathbb{N}[y_1, \ldots, y_{\ell_m}]^{D_m},$$

where ℓ_i is the number of statistic functions of $\mathcal{B}^{(i)}$ and

$$D_k = \max(0, r_S^{(k)}(n) + 1),$$

while the codomain is simply $\mathbb{N}[y_1, \ldots, y_\ell]$, where ℓ is the number of statistic functions of $\mathcal{A}$.

3.3 Back to the fusion

With the formalism of tracked combinatorial sets and strategies for these sets, we can formally define the fusion strategy discussed in Section 3.1. As in Chapter 2, we define the strategy by its effect on (tracked) tilings even though it really is defined on the tracked combinatorial sets of valid gridded permutations.

We concluded Section 3.1 with a general criterion to check when Equation (3.2) can be used to obtain the counting sequence of a tiling $\mathcal{T}$ from the counting sequence of the tiling $\text{FuseCol}_i(\mathcal{T})$. It was a counting argument based on the number of preimages of each obstruction and requirement on the fused tiling. Precisely, for a tiling $\mathcal{T} = ((t,u), \mathcal{O}, \{\mathcal{R}_1, \ldots, \mathcal{R}_m\})$, the counting formula is valid if for each gridded permutation π in $\text{FuseCol}_i(\mathcal{O})$ (resp. $\text{FuseCol}_i(\mathcal{R}_i)$) the number of points of π in the column i is equal to the number of gridded permutations in $\mathcal{O}$ (resp. $\mathcal{R}_i$) that are mapped to π by FuseCol_i. We say that column i and $i+1$ of a tracked tiling $\mathcal{T}$ are *fusable* if that condition is satisfied and $\mathcal{T}$ has at least $i+2$ columns (so that column i and $i+1$ both exist). The strategy $\texttt{ColFusion}_i$ is defined as follow:

- If $\mathcal{T}$ is a tracked tiling with no tracking where column i and $i+1$ are fusable then $d_{\texttt{ColFusion}_i}(\mathcal{T})$ is $\text{FuseCol}_i(\mathcal{T})$ with the addition of tracking over the column i. Otherwise, $d_{\texttt{ColFusion}_i}(\mathcal{T}) = \texttt{DNA}$.
- The reliance profile function is $r_{\texttt{ColFusion}_i}(n) = (n)$.
- The counting functions are

$$c_{\texttt{ColFusion}_i,(n)}((b_0, \ldots, b_n)) = \frac{\partial}{\partial y}(y \cdot b_n)\bigg|_{y=1}.$$

To better understand the counting function of the strategy, it is best, since the function is linear, to think of its effect on a single monomial of the polynomial b_n. In b_n, the monomial y^k represents a gridded permutation of size n with k points in the tracked region of the fused tiling. In this case, the tracked region is the column that is the result of the fusion of two columns. We know, from our discussion in Section 3.1, that this gridded permutation has $k+1$ preimages under the FuseCol_i map. These preimages are all valid gridded permutations on the unfused tiling. If we apply the counting function to it, we effectively get

$$\frac{\partial}{\partial y}(y \cdot y^k)\bigg|_{y=1} = (k+1)y^k\big|_{y=1} = k+1.$$

This also matches Equation (3.2) derived in Section 3.1.

Figure 3.7 shows an application of the column fusion strategy. The first few

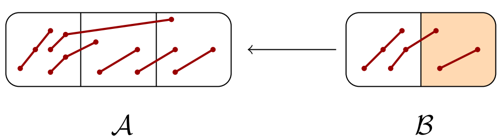


Figure 3.7: Application of the $\texttt{ColFusion}_1$ strategy.

terms of the counting sequence for $\mathcal{A}$ are

$$1, 3, 9, 28, 90, \ldots$$

while the first terms for $\mathcal{B}$ are

$$1, y+1, y^2+2y+2, y^3+3y^2+5y+5, y^4+4y^3+9y^2+14y+14, \ldots$$

If we apply the counting formula to the terms of the counting sequence of $\mathcal{B}$ we recover the counting sequence for $\mathcal{A}$.

3.4 Other tracked strategies

All the strategies covered in Chapter 2 and in [34] are straightforward to adapt to tracked tilings. It is, in all cases, easy to see how the strategy moves the cells of the tiling around and modifies the tracking consequently. It, however, becomes quickly technical when one needs to consider the interactions between different tracked regions on the same tiling as well as the numerous edge cases. To lighten the text, we will not give here any formal definition of a tracked strategy but instead examples of requirement insertion, factor and point placement strategies being applied to a tracked tiling. Without being complicated, these examples should be sufficient for the reader to see how the strategies extend to the tracked context.

Applying the requirement insertion strategy to a tracked tiling acts exactly like on a regular tiling leaving the tracking unchanged. It could, however, be the case that one of the cells being tracked becomes empty. In this case, the tracking is simplified to only track cells that can contain points. Figure 3.8 shows an example of the strategy $\texttt{ReqIns}_{\{(1,(0,1))\}}$ applied to a tracked tiling. On the left child, the tracking disappears since the cell where the tracking was is now empty. The n-th term of the counting sequence of the parent is obtained by summing the n-th term of the counting sequence of both children as for the ReqIns strategy defined for normal tilings in Chapter 2.

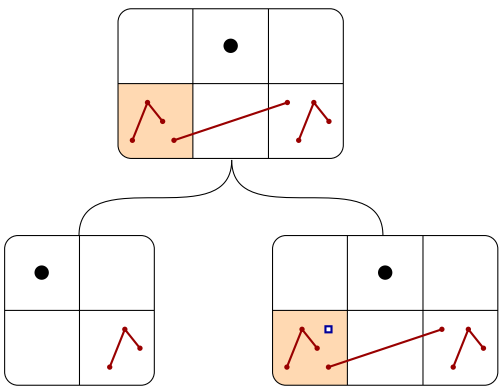


Figure 3.8: Inserting the requirement $(1, (0, 0))$ in a tracked tiling.

When applying the factor strategy to a tracked tiling, the tiling is decomposed as for the case of regular tilings. The cells that were tracked on the parent stay tracked on the children. For example, Figure 3.9 shows an example of factoring a tracked tiling into two parts. On the parent, the 132 and 123-avoiding cells are both tracked so the corresponding cells are also tracked on $\mathcal{B}$ and $\mathcal{C}$. The 312-avoiding cell stays untracked. The counting sequence of $\mathcal{A}$ is obtained by the following formula

$$\|\mathcal{A}_n\| = \sum_{i=1}^{n} \|\mathcal{B}_i\| \cdot \|\mathcal{C}_{n-i}\|.$$

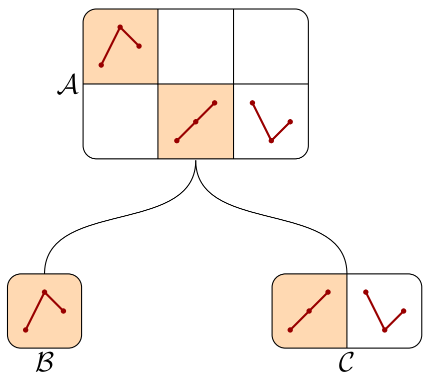


Figure 3.9: Factoring a tracked tiling.

When performing point placement on a tracked tiling, one must be careful to stretch the tracking on the cells that get stretched by the placement. Figure 3.10 shows an example where the bottommost point in cell $(0,0)$ (which is tracked) is placed. The cell $(0,0)$ stretches to the area composed of the first three columns of the child. Therefore, the tracking needs to cover these three columns. However, we only mark the three cells that are not empty as the others have no effect on the associated statistic function. If the cell $(1,0)$ was tracked on the parent tiling, the tracking would need to stretch over cell $(3,0)$ and $(3,2)$ of the child. The counting sequence of the two tiling is the same like for point placement with regular tilings.

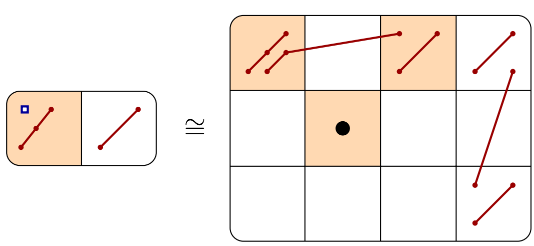


Figure 3.10: Placing the bottommost point in cell $(0,0)$ of a tracked tiling.

The fusion strategy can also be extended to apply to tilings which have some tracked regions. Figure 3.11 shows an example where one of the two columns fusing is tracked. The counting function's output now has to be refined according to

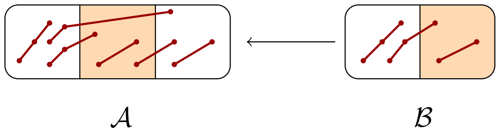


Figure 3.11: Fusion strategy applied to a tiling with tracking.

the number of points in the middle cell of $\mathcal{A}$. In the example of Section 3.3 (see Figure 3.7), a gridded permutation with k points in the tracked cell of the fused tiling contributed $k+1$ permutation to the tiling $\mathcal{A}$. Therefore, the counting function needed to transform y^k into $(k+1)$. In this new example, a gridded permutation with k points still corresponds to $(k+1)$ gridded permutations but each of them has a different number of points in the tracked cell of $\mathcal{A}$. Consequently, we want the operator to map y^k to $1+y+y^2+\cdots+y^k$. This is achieved by the following formula:

$$\|\mathcal{A}_n\| = \frac{\|\mathcal{B}_n\||_{y=1} - y\|\mathcal{B}_n\|}{1-y}.$$

In fact, if we substitute y^k to b_n in this equation, we get

$$\frac{1-y^{k+1}}{1-y} = \frac{(1-y)(1+y+y^2+\cdots+y^k)}{1-y} = 1+y+y^2+\cdots+y^k$$

as expected.

3.5 Using tracked combinatorial sets in combinatorial exploration

In essence, the only changes we did to combinatorial sets and strategies in this chapter is to transform all the counting information from integers to polynomials in order to have more refined counting information. Precisely, the terms for counting sequences are now polynomials while the counting functions for strategies take polynomials as input and output polynomials. Apart from that, the framework stays unchanged. In particular, everything regarding reliance profiles is still the same. The notion of a reliance graph, enumerable subset and regular strategy directly extend to tracked combinatorial sets and tracked strategies. In particular, Theorem 2.20 can be extended to tracked strategies as follow:

Theorem 3.5. *Let U be a universe of rules produced by regular tracked strategies. Then, the enumerable subset of U is the set of all tracked combinatorial sets of U that are in a specification contained in U and whose reliance graph contains no infinite directed walk.*

Moreover, Algorithm 2 can still be used to compute the enumerable subset using the tracked version of the strategies discussed in this thesis.

Piecing all these tools together, we were able to expand tracked universes and find productive combinatorial specifications for many permutation classes. Using fusion without the use of reverse strategies led to a lot of new results. It was however not sufficient to recover all the known results for permutation classes avoiding some sets of size 4 patterns. Here, we consider in particular the results for the size 4 principal permutation classes. These are the permutation classes avoiding a single pattern of size 4. They are generally considered as the most difficult problems avoiding size 4 patterns. Up to symmetry, there are seven different size 4 principal permutation classes. Before the introduction of fusion, combinatorial exploration could not find a combinatorial specification for any of them. Using fusion, but still restricting ourselves to productive strategies with the prune method (Algorithm 1), we could find specifications for Av(1234), Av(1243) and Av(1432). Of the four permutation classes left, we could find specifications that use both fusion and reverse rules using Algorithm 2 for three more principal permutation classes, namely Av(2143), Av(1342) and Av(2413). The specifications we found for those permutation classes all feature both a fusion rule and a rule produced by a reversed strategy that could not be handled without the theory developed in Chapter 2. The final permutation class, Av(1324), is still out of reach of both human mathematicians and the automatic methods of combinatorial exploration despite combining both the power of fusion and reverse strategies.

We conclude this chapter by taking a closer look at one of these specifications that uses both fusion and reverse strategies. We present a specification found for Av(1342) as it was the first specification found using both fusion and reverse strategies but also because its the one with the fewest rules and, therefore, the most likely to fit on one page of this thesis. It is also interesting to note that this specification is the first direct enumeration of this permutation class as previous approaches used bijections to other objects. Figure 3.12 shows the picture of the specification. For a detailed view, we invite the reader to view the proof tree in the PermPal database at `https://permpal.com/tree/12/`. There the reader has the ability to click on the tree to see the strategies applied and all the details about each of the tilings involved.

We highlight here the most interesting rules appearing in the specification.

- The rule $\mathcal{T}^{(5)} \leftarrow (\mathcal{T}^{(6)})$ is produced by the add tracking strategy (that is not discussed in this thesis). It consists of adding the tracking to a specific region of a tiling. The counting sequence of $\mathcal{T}^{(6)}$ is more refined then the one of $\mathcal{T}^{(5)}$ as it accounts for number of points in cell $(0,1)$. We get the n-th term of the counting sequence of $\mathcal{T}^{(5)}$ by substituting $y = 1$ in the n-th term of the counting sequence of $\mathcal{T}^{(6)}$.

- The rule $\mathcal{T}^{(6)} \leftarrow (\mathcal{T}^{(0)}, \mathcal{T}^{(7)})$ is created by a requirement insertion strategy. Precisely, a point is inserted in cell $(0,1)$. There is no tracking on the child $\mathcal{T}^{(0)}$ since the cell that was tracked on the parent $\mathcal{T}^{(6)}$ can no longer contain points, since a point obstruction was added.

- The rule $\mathcal{T}^{(7)} \leftarrow (\mathcal{T}^{(8)})$ is the combination of a requirement placement strategy and a row separation strategy (click on the node in the web version to

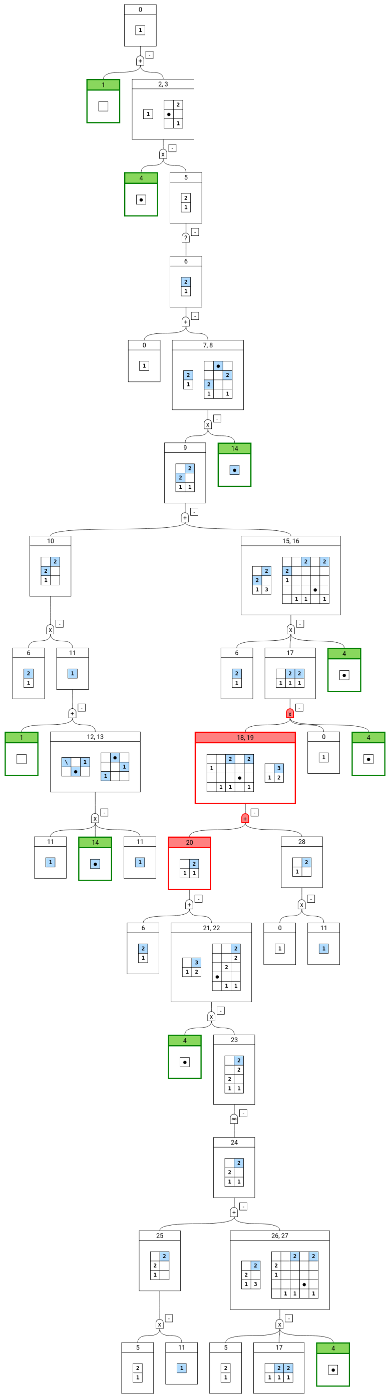


Figure 3.12: Combinatorial specification for Av(1342). The full specification can be viewed at `https://permpal.com/tree/12/`.

see the two strategies applied separately). We see the tracking evolve and span across multiple cells as we place the point and separate the row. Since no point can contribute towards the statistic in the empty cells, only the non-empty ones are marked with the tracking.

- The rule $\mathcal{T}^{(17)} \leftarrow (\mathcal{T}^{(18)}, \mathcal{T}^{(0)}, \mathcal{T}^{(4)})$ is a reversed factor rule which is actually created from the factor rule $\mathcal{T}^{(18)} \leftarrow (\mathcal{T}^{(17)}, \mathcal{T}^{(0)}, \mathcal{T}^{(4)})$.
- The rule $\mathcal{T}^{(19)} \leftarrow (\mathcal{T}^{(20)}, \mathcal{T}^{(28)})$ is a reverse disjoint union rule that comes from the requirement insertion rule $\mathcal{T}^{(20)} \leftarrow (\mathcal{T}^{(28)}, \mathcal{T}^{(18)})$ that consists of inserting a point in cell $(1, 0)$.
- The rule $\mathcal{T}^{(23)} \leftarrow (\mathcal{T}^{(24)})$ is a fusion rule where some tracking on the parent is present akin to the example of Figure 3.11. We are in this case fusing two rows instead of two columns.

The full system of equations for the specification can be found on page 77. It solves to the generating function of Av(1342) which is

$$\frac{32x}{1 + 20x - 8x^2 - (1 - 8x)^{\frac{3}{2}}}$$

as expected.

Table 3.3 gives links to the permpal library to see combinatorial specifications for each of the principal size 4 classes except Av(1324). There the reader can find more complex specifications than the one presented here using both more rules and combinatorial sets with multiple statistics.

Permutation class	Permpal entry
Av(1234)	`https://permpal.com/perms/basis/0123`
Av(1243)	`https://permpal.com/perms/basis/0132`
Av(1342)	`https://permpal.com/perms/basis/0231`
Av(1432)	`https://permpal.com/perms/basis/0321`
Av(2143)	`https://permpal.com/perms/basis/1032`
Av(2413)	`https://permpal.com/perms/basis/1302`

Table 3.3: Link to combinatorial specifications for all principal size 4 permutation classes except Av(1324).

$$
\begin{aligned}
F_0(x) &= F_1(x) + F_2(x)\\
F_1(x) &= 1\\
F_2(x) &= F_3(x)\\
F_3(x) &= F_4(x)\, F_5(x)\\
F_4(x) &= x\\
F_5(x) &= F_6(x,1)\\
F_6(x,y) &= F_0(x) + F_7(x,y)\\
F_7(x,y) &= F_8(x,y)\\
F_8(x,y) &= F_{14}(x,y)\, F_9(x,y)\\
F_9(x,y) &= F_{10}(x,y) + F_{15}(x,y)\\
F_{10}(x,y) &= F_{11}(x,y)\, F_6(x,y)\\
F_{11}(x,y) &= F_1(x) + F_{12}(x,y)\\
F_{12}(x,y) &= F_{13}(x,y)\\
F_{13}(x,y) &= F_{11}(x,y)^2\, F_{14}(x,y)\\
F_{14}(x,y) &= yx\\
F_{15}(x,y) &= F_{16}(x,y)\\
F_{16}(x,y) &= F_{17}(x,y)\, F_4(x)\, F_6(x,y)\\
F_{18}(x,y) &= F_0(x)\, F_{17}(x,y)\, F_4(x)\\
F_{18}(x,y) &= F_{19}(x,y)\\
F_{20}(x,y) &= F_{19}(x,y) + F_{28}(x,y)\\
F_{20}(x,y) &= F_{21}(x,y) + F_6(x,y)\\
F_{21}(x,y) &= F_{22}(x,y)\\
F_{22}(x,y) &= F_{23}(x,y)\, F_4(x)\\
F_{23}(x,y) &= \frac{yF_{24}(x,y) - F_{24}(x,1)}{y-1}\\
F_{24}(x,y) &= F_{25}(x,y) + F_{26}(x,y)\\
F_{25}(x,y) &= F_{11}(x,y)\, F_5(x)\\
F_{26}(x,y) &= F_{27}(x,y)\\
F_{27}(x,y) &= F_{17}(x,y)\, F_4(x)\, F_5(x)\\
F_{28}(x,y) &= F_0(x)\, F_{11}(x,y)
\end{aligned}
$$

System of equations 3.1: Equations from the combinatorial specification for Av(1342) in Figure 3.12.

Chapter 4

Enumeration of permutation classes and weighted labelled independent sets

Bean, Tannock and Ulfarsson [42] introduced the *staircase encoding*, a function which maps a permutation to a staircase grid where cells are filled with non-negative integers. In this context, each integer is the size of the monotone sequence in its cell. In this chapter, we refine the staircase encoding as a function which maps a permutation to a staircase grid where cells are filled with permutations. Using this function, we retrieve the generating function of several permutation classes.

We first recall some permutation background and introduce mesh patterns in Section 4.1. We then recover the results for Av(123) and Av(132) from [42] in Section 4.2, using our more refined encoding and weighted independent sets. Our technique is then extended to describe the structure of Av(2314, 3124) and Av(2413, 3142) in Section 4.3 and 4.4. In Section 4.5, we recall the *updown core graph* introduced by [42] and use it to enumerate Av(2314, 3124, 2413, 3142), before introducing a new core graph in Section 4.6 that is used to give the structure of Av(2314, 3124, 3142). Our notion of weighted independent sets is then generalized to allow labelling. This enables a more refined choice of permutations in our encoding, and is used to enumerate Av(2413, 3142, 3124) in Section 4.7 and Av(2413, 3124) in Section 4.8. By allowing some interleaving between cells in the staircase grid representation of a permutation, we obtain the counting sequences for Av(2413, 2134) and Av(2314, 2134) in Sections 4.9 and 4.10. Finally, in Section 4.11 we use results from previous sections to prove two unbalanced Wilf-equivalences. Our results handle in a unified framework the generating function of the counting sequence of many permutation classes that were first enumerated in [7], [8], [16], [19], [43]–[45]. Moreover, the results also allow one to easily enumerate many subclasses of these classes. To check whether these methods apply to a particular permutation class, we have added routines to the python package *Permuta*. Instruction on how to use it can be found at the end of this chapter in Section 4.12.

4.1 Background

We recall, from Section 1.3.3, the notion of sum and skew-sum of permutations. Remember that the sum of 123 and 21 is $123 \oplus 21 = 12354$ while the skew-sum of these two permutations is $123 \ominus 21 = 34521$. The result of these operations appears in Figure 4.1. The permutation 12435 is said to be sum-decomposable since it can be expressed as $12 \oplus 213$. Similarly, the permutation 43512 is skew-decomposable since it is $213 \ominus 12$. The permutation 34521 is sum-indecomposable since it cannot be expressed as the sum of two non-empty permutations while 12354 is said to be skew-indecomposable since it cannot be expressed as the skew-sum of two non-empty permutations. The sum of a permutation σ and a set of permutations P is the set $\sigma \oplus P = \{\sigma \oplus \alpha : \alpha \in P\}$, and likewise $\sigma \ominus P = \{\sigma \ominus \alpha : \alpha \in P\}$. We also define the sum, and skew-sum, of two sets of permutations in the obvious way.

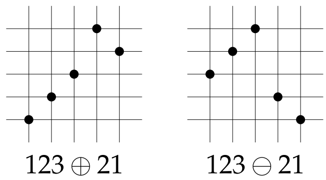
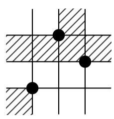


Figure 4.1: The sum and skew-sum of two permutations.

4.1.1 Mesh patterns

We complete this section with a short introduction to mesh patterns which are utilised in some of our proofs. A reader familiar with them might skip directly to Section 4.2. A *mesh pattern* p is a pair $(\pi, \mathcal{R})$ with $\pi \in \mathcal{S}_k$ and $\mathcal{R} \subseteq \{0, 1, \ldots, k\}^2$. Pictorially, we represent a mesh pattern in a similar way as a classical pattern, and we shade, for each $(x, y) \in \mathcal{R}$, the unit square with bottom left corner in (x, y). The mesh pattern $p = (132, \{(0,0), (0,2), (1,2), (2,2), (2,3), (3,2)\})$ is pictured below.



Intuitively, an occurrence of a mesh pattern $p = (\pi, \mathcal{R})$ in a permutation σ is an occurrence of π in σ such that, if we stretch the shading of π onto σ, σ has no point in the shaded region. For example, we consider the permutation 35142 and pick two different occurrences of 132 in it (see Figure 4.2). We stretch the shading of p for both occurrences. The one on the left is an occurrence of p since no points of σ are in the shading, however, the right one is not an occurrence of p since the 3 of σ is in the region corresponding to the box $(0, 2)$ in p.

Formally, the definition of mesh pattern containment is as follows.

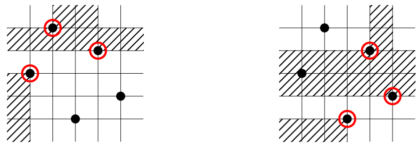


Figure 4.2: Two classical occurrences of 132 in the permutation 35142. On the left the classical occurrence is an occurrence of p whereas the one on the right is not.

Definition 4.1 ([46]). *Let $\pi \in \mathcal{S}_k$ and $\sigma \in \mathcal{S}_n$. An occurrence of the mesh pattern $p = (\pi, \mathcal{R})$ in a permutation σ is a subset ω of the plot of σ, $G(\sigma) = \{(i, \sigma(i)) : i \in \{1, 2, \ldots, n\}\}$ such that there are order preserving injections $\alpha, \beta : \{1, \ldots, k\} \to \{1, \ldots, n\}$ satisfying the following conditions. Firstly, ω is an occurrence of π in the classical sense, i.e.,*

i. $\omega = \{(\alpha(i), \beta(j)) : (i, j) \in G(\sigma)\}$.

Define $R_{ij} = [\alpha(i) + 1, \alpha(i + 1) - 1] \times [\beta(j) + 1, \beta(j + 1) - 1]$ for $i, j \in \{1, \ldots, k\}$ where $\alpha(0) = \beta(0) = 0$ and $\alpha(k + 1) = \beta(k + 1) = n + 1$. Then the second condition is

ii. if $(i, j) \in \mathcal{R}$ then $R_{ij} \cap G(\sigma) = \emptyset$.

If there is an occurrence of p in σ we say that p is *contained in* σ. Otherwise, we say that σ *avoids* the mesh pattern p.

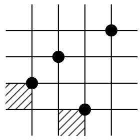
Unlike for classical patterns, it can occur that $\mathrm{Av}(p) = \mathrm{Av}(q)$ for two different mesh patterns, p, q. For instance the mesh patterns $(21, \emptyset)$ and $(21, \{(1,0), (1,1), (1,2)\})$ have the same avoiding permutations, since a permutation has an inversion if and only if it has a descent. Many of these *coincidences* are captured by the Shading Lemma [47, Lemma 11].

Lemma 4.2 (Shading Lemma, Lemma 11 in [47]). *Let $(\pi, \mathcal{R})$ be a mesh pattern of size n such that $\pi(i) = j$ and the box $(i, j) \notin \mathcal{R}$. If all the following conditions are satistified:*

1. *The box $(i - 1, j - 1)$ is not in $\mathcal{R}$;*
2. *At most one of the boxes $(i, j - 1)$ and $(i - 1, j)$ is in $\mathcal{R}$;*
3. *If the box $(\ell, j - 1)$ is in the $\mathcal{R}$ (with $\ell \neq i - 1, i$) then the box (ℓ, j) is also in $\mathcal{R}$;*
4. *If the box $(i - 1, \ell)$ is in $\mathcal{R}$ (with $\ell \neq j - 1, j$) then the box (i, ℓ) is also in $\mathcal{R}$;*

then the patterns $(\pi, \mathcal{R})$ and $(\pi, \mathcal{R} \cup \{(i, j)\})$ are coincident (one cannot appear in a permutation without an occurrence of the other). Analogous conditions determine if other boxes neighboring the point (i, j) to $\mathcal{R}$ while preserving the coincidence of the corresponding patterns.

Throughout this chapter, we will use the Shading Lemma to argue that the occurrence of a classical pattern implies the occurrence of a mesh pattern. For example, in the proof of Lemma 4.13 we argue that an occurrence of 2314 implies an occurrence of

 .

The argument goes as follows. Let σ be a permutation with an occurrence of 2314. We can shade the box $(2,0)$ by replacing the 1 of the occurrence by the bottommost point in that box. The box $(0,1)$ can also be shaded by replacing the 2 of the occurrence with the leftmost point in that cell.

4.2 Encoding permutations on a grid with the staircase encoding

A letter σ_i in a permutation σ is called a *left-to-right minimum* if $\sigma_j > \sigma_i$ for all $j < i$. We denote with $\mathrm{Av}^{(n)}(B)$ the permutations in $\mathrm{Av}(B)$ with exactly n left-to-right minima. For a coarser representation, take a permutation σ in $\mathrm{Av}^{(n)}(B)$ and place the left-to-right minima on the main diagonal of a $n \times n$ grid, and the remaining points into the cells of the grid with respect to their relative positions. We then replace the points in each cell by the permutation they are forming in this cell. This is called the *staircase encoding* of σ and is denoted $\mathrm{SE}(\sigma)$. Figure 4.3 shows the staircase encoding of the permutation 659817432. As permutations contained in cells in the same row or same column can interleave in multiple ways, the staircase encoding is not an injective map. For example, the permutations 659814327 and 659718432 both have the staircase encoding shown in Figure 4.3 (c).

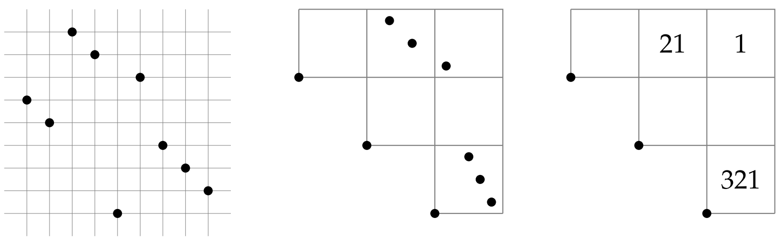


Figure 4.3: (a) The plot of $\sigma = 659817432$. (b) The permutation σ drawn on the staircase grid. (c) The staircase encoding of σ.

By construction, the staircase encoding only uses the cells above and to the right of the left-to-right minima. We define the *staircase grid B_n* as the set of cells of the staircase encoding of a permutation with n left-to-right minima. The cells are indexed using matrix coordinates, *i.e.*, $B_n = \{(i,j) : 1 \leq i \leq n \text{ and } i \leq j \leq n\}$. We say that B_0 is the empty staircase grid which corresponds to the staircase encoding of ε, the empty permutation. Bean, Tannock and Ulfarsson [42] used the staircase grid to enumerate $\mathrm{Av}(123)$ and $\mathrm{Av}(132)$. We briefly review these in terms of our staircase encoding.

A cell in the staircase encoding of a permutation that avoids 123 contains a permutation avoiding 12, since any occurrence of 12 together with one of the left-to-right minima would give an occurrence of 123. Moreover, the presence of a point in a cell forces other cells to be empty. For example, in the encoding of Figure 4.3, we have the staircase encoding of the 123 avoiding permutation 659817432. As the cell $(1,3)$ contains a point, the cell $(2,2)$ must be empty if the encoding is one of a permutation avoiding 123. These constraints are symmetric and can be represented as a graph, where the cells of B_n are the vertices and there is an edge between every pair of cells that cannot both contain a point of the permutation. This graph is called the *up-core* of B_n.

Definition 4.3 (Definition 4.3 in [42]). *Let $n \geq 0$ be an integer. The* up-core *of B_n is the labelled undirected graph* $\mathrm{U}(B_n)$ *with vertex set B_n and edges between (i,j) and (k,ℓ) if $i > k$, $j < \ell$.*

If a permutation avoids 132, we get similar restrictions on the staircase encoding. First, every cell avoids 21 for a similar reason as above. Second, some pairs of cells cannot both contain a point. These restrictions are also described by a graph called the *down-core*.

Definition 4.4 (Definition 4.3 in [42]). *Let $n \geq 0$ be an integer. The* down-core *of B_n is the labelled undirected graph* $\mathrm{D}(B_n)$ *with vertex set B_n and edges between (i,j) and (k,ℓ) if $i < k$, $j < \ell$ and the rectangle $\{i, i+1, \ldots k\} \times \{j, j+1, \ldots, \ell\}$ is a subset of B_n.*

See Figure 4.4 for examples of $\mathrm{U}(B_4)$ and $\mathrm{D}(B_4)$. We say that a cell of the staircase encoding is *active* if it contains a non-empty permutation. From the construction of the graphs, we can see that the set of active cells of the staircase encoding of a permutation in $\mathrm{Av}(123)$ (resp. $\mathrm{Av}(132)$) is an independent set of $\mathrm{U}(B_n)$ (resp. $\mathrm{D}(B_n)$). The image under the staircase encoding of $\mathrm{Av}(123)$ is the set of staircase encodings that are independent sets of $\mathrm{U}(B_n)$, where the permutations in every cell avoid 12.

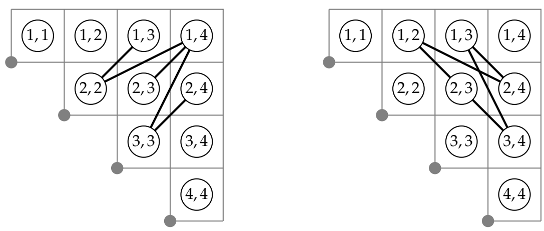


Figure 4.4: The up-core $\mathrm{U}(B_4)$ on the left and the down-core $\mathrm{D}(B_4)$ on the right.

In order to formalize our previous statement, we introduce *weighted independent sets*, an independent set where a weight is given to each of its vertices. In this chapter, the weights will always be permutations.

Definition 4.5. *We denote by* $\mathrm{WI}(G,S)$ *the set of all weighted independents sets of a graph G where the weights are permutations from the set S.*

Since for uniqueness we generally want the weight to not be an empty permutation we introduce the notation $\mathrm{Av}^+(B) = \mathrm{Av}(B)\backslash\{\varepsilon\}$, *i.e.*, $\mathrm{Av}^+(B)$ is the set of non-empty permutations avoiding a given set of patterns. Using this notation, we have that

$$\mathrm{SE}(\mathrm{Av}^{(n)}(123)) \subseteq \mathrm{WI}(\mathrm{U}(B_n), \mathrm{Av}^+(12))$$

and

$$\mathrm{SE}(\mathrm{Av}^{(n)}(132)) \subseteq \mathrm{WI}(\mathrm{D}(B_n), \mathrm{Av}^+(21)).$$

The i-th *row* of a permutation consists of the points with values between the value of the i-th left-to-right minima and the $(i+1)$-st left-to-right minima of the permutation. Avoiding 123 forces rows of the permutation to be *decreasing*. This means that for two cells (i, j) and (i, k) with $j < k$ the points in (i, j) are above the points in (i, k), *i.e.*, larger in value. A decreasing row is pictured in Figure 4.5.

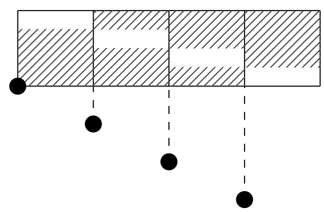


Figure 4.5: A typical decreasing row. There are no points in the shaded regions.

For a 132 avoiding permutation, the rows of the permutation are *increasing*, *i.e.*, for a pair of cells (i, j) and (i, k) with $j < k$ the points in (i, j) are below the points in (i, k), *i.e.*, lower in value.

The j-th *column* of a permutation consists of the points with index between the indices of the j-th and $(j+1)$-st left-to-right minima of a permutation. In a similar manner as above, we say that the j-th column is *increasing* (resp. *decreasing*) if for each pair of cells (i, j) and (k, j) with $i > k$ the points in (i, j) are on the left (resp. right) of the points in (k, j). The columns of a 123 avoiding permutation are decreasing while the columns of a 132 avoiding permutation are increasing.

As mentioned before, the staircase encoding is not an injective map since many permutations can have the same staircase encoding. However, by restricting to the set of permutations with increasing (resp. decreasing) rows and columns the staircase encoding is an injection. The inverse of the staircase encoding restricted to permutations with increasing (resp. decreasing) rows and columns is dperm (resp. uperm).

Definition 4.6. *For a staircase encoding E, we define*

- $\mathrm{uperm}(E)$ *as the permutation σ with decreasing rows and columns such that its staircase encoding* $\mathrm{SE}(\sigma) = E$.

- $\mathrm{dperm}(E)$ *as the permutation σ with increasing rows and columns such that its staircase encoding* $\mathrm{SE}(\sigma) = E$.

Both uperm and dperm are injective maps from the set of staircase encodings to the set of all permutations. Lemma 4.7 follows from the definition.

Lemma 4.7. *The maps* $\mathrm{SE} \circ \mathrm{uperm}$ *and* $\mathrm{SE} \circ \mathrm{dperm}$ *are the identity on the set of all staircase encodings.*

Remark 4.8. *Formally, the staircase encoding is a map from the set of all permutations to the set of staircase grids filled with permutations. However, throughout the chapter we consider the restriction of* SE *to a smaller set such that the restriction is a bijection to its image. Hence, when the context is clear (as in the theorem below),* SE *might refer to a restriction of the staircase encoding.*

Theorem 4.9 (Lemma 2.2 in [42]). *The map* SE *is a bijection between* $\mathrm{Av}^{(n)}(123)$ *and the weighted independent sets* $\mathrm{WI}(\mathrm{U}(B_n), \mathrm{Av}^+(12))$. *It is also a bijection between* $\mathrm{Av}^{(n)}(132)$ *and the weighted independent sets* $\mathrm{WI}(\mathrm{D}(B_n), \mathrm{Av}^+(21))$

By Theorems 2.4 and 3.3 from [42] we know that the number of independent sets of size k in $\mathrm{U}(B_n)$, or $\mathrm{D}(B_n)$, is given by the coefficient of $x^n y^k$ in the generating function $\mathbf{F}(x, y)$ that satisfies

$$\mathbf{F}(x,y) = 1 + x\,\mathbf{F}(x,y) + \frac{xy\,\mathbf{F}\,(x,y)^2}{1 - y(\mathbf{F}(x,y) - 1)}. \tag{4.1}$$

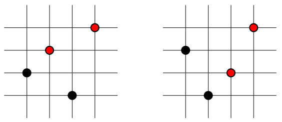
If we substitute y with $\frac{x}{1-x}$ into $\mathbf{F}(x, y)$, we obtain the generating function where the coefficient of x^n is the number of 123 avoiding permutation of size n.

Corollary 4.10. *The generating function for* $\mathrm{Av}(123)$ *and* $\mathrm{Av}(132)$ *is* $\mathbf{F}\left(x, \frac{x}{1-x}\right)$.

4.3 Going from size 3 to size 4 patterns

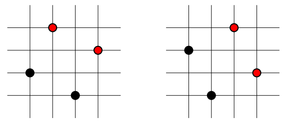
As seen in Section 4.2, avoiding the pattern 123 creates restrictions on which pairs of cells in the staircase grid can contain points of the permutation. These restrictions were encoded by the up-core graph. The same restrictions are enforced on the set of active cells of the staircase encoding for permutations avoiding 2314 or 3124. The two patterns are pictured in Figure 4.6 (a). In this figure, the black points can be thought of as left-to-right minima of the permutation and red points as points in the cells of the staircase grid. Avoiding either of these patterns ensures that two cells connected by up-core edges cannot be active simultaneously. Moreover, the pattern 2314, the *row-up pattern*, denoted r_u, forces the rows to be decreasing. Similarly, the pattern 3124, the *column-up pattern*, denoted c_u, forces the columns to be decreasing.

The down-core restrictions can also be enforced using size 4 patterns. To do so, we introduce the *row-down pattern* 2413, denoted r_d, and the *column-down pattern* 3142, denoted c_d. As for the up-core, thinking of the black points as the left-to-right minima of the permutation and the red points as points in cells (see Figure 4.6 (b)), we can see this results in the same constraints as in the down-core. Moreover, these patterns force rows and columns to be increasing. The above discussion is formalized in the next two lemmas.



$r_u = 2314$ $c_u = 3124$

(a)



$r_d = 2413$ $c_d = 3142$

(b)

Figure 4.6: (a) The row-up pattern on the left and the column-up pattern on the right. (b) The row-down pattern on the left and the column-down pattern on the right.

Lemma 4.11. *Let σ be a permutation. Then*

1. *the rows of σ are decreasing if $\sigma \in \mathrm{Av}(r_u)$,*
2. *the columns of σ are decreasing if $\sigma \in \mathrm{Av}(c_u)$,*
3. *the rows of σ are increasing if $\sigma \in \mathrm{Av}(r_d)$,*
4. *the columns of σ are increasing if $\sigma \in \mathrm{Av}(c_d)$.*

Proof. We only prove 1. since the other cases can be handled similarly. Let σ be a permutation and suppose that one of the rows is not decreasing. This row has at least two active cells, therefore σ has at least two left-to-right minima. Hence, when drawn on a staircase grid, this row contains two cells A and B such that B is to the right of A, and B contains a point higher than a point in A. These two points together with the left-to-right minimum to the left of the columns containing A and B form an occurrence of r_u in σ. □

Lemma 4.12. *Let σ be a permutation with n left-to-right minima and C be the set of active cells of the staircase encoding of σ. Then C is an independent set of*

1. $\mathrm{U}(B_n)$ *if $\sigma \in \mathrm{Av}(r_u) \cup \mathrm{Av}(c_u)$,*
2. $\mathrm{D}(B_n)$ *if $\sigma \in \mathrm{Av}(r_d) \cup \mathrm{Av}(c_d)$.*

Proof. It is sufficient to show 1., which we do by proving the contrapositive: suppose that two active cells of the staircase encoding of a permutation σ are connected by an edge of $\mathrm{U}(B_n)$. Hence, one of the cells is above and to the right of the other. Moreover, since they are in distinct rows and distinct columns of B_n, there exist three left-to-right minima as shown on Figure 4.7.

The red point and the blue point together with the two points in the active cells form a c_u pattern. Replacing the red point by the green one yields an occurrence of the pattern r_u. Hence, σ is not in the union $\mathrm{Av}(r_u) \cup \mathrm{Av}(c_u)$. □

In sections 4.4 to 4.9, we study different combinations of the patterns r_u, c_u, r_d and c_d. This leads to different graphs, weights and constraints on the rows and columns of the staircase grid. With each combination of patterns, we describe a set of patterns P that can be added to the basis while keeping the structural

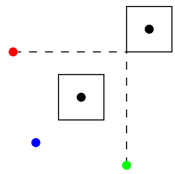


Figure 4.7: Two cells connected by an edge of the up-core. The blue, the red and the green points are distinct left-to-right minima.

properties of the permutation class that we need for enumeration. Even when not specified explicitly, we assume throughout the chapter that the empty permutation is not in the set P. The results are presented in order of increasing complexity, with each section introducing a new tool that is used to build different structural descriptions and generating function arguments. Table 4.1 presents an overview of the results in the upcoming sections. The notation $P^\times$ that appears in the table is introduced in Definition 4.27.

Permutation classes	Conditions on the set P	Enumeration result
$\text{Av}(r_u, c_u, 1 \oplus P)$	P is skew-indecomposable	Corollary 4.15
$\text{Av}(r_d, c_d, 1 \oplus P)$	P is sum-indecomposable	Corollary 4.18
$\text{Av}(r_u, c_u, r_d, c_d, 1 \oplus P)$	No condition on P	Corollary 4.22
$\text{Av}(r_u, c_u, c_d, 1 \oplus P)$	P is skew-indecomposable	Corollary 4.25
$\text{Av}(r_d, c_d, c_u, 1 \oplus P)$	$P^\times$ is sum-indecomposable	Corollary 4.29
$\text{Av}(r_d, c_u, 1 \oplus P)$	P is skew-indecomposable and $P^\times$ is sum-indecomposable	Corollary 4.32
$\text{Av}(r_d, 2134, P)$	P satisfies conditions described in Section 4.9	Corollary 4.37
$\text{Av}(r_u, 2143, P)$	P satisfies conditions described in Section 4.10	Corollary 4.42

Table 4.1: Overview of the permutation classes we cover in the upcoming sections.

4.4 Weighted independent sets of the up-core and the down-core

Lemmas 4.11 and 4.12 say that every permutation in $\text{Av}(r_u, c_u)$ can be constructed by first taking an independent set of the up-core of a staircase grid, and weighting the cells with permutations in $\text{Av}(r_u, c_u)$. We will show how this can be used to enumerate the permutation class $\text{Av}(r_u, c_u)$ and many of its subclasses. We first show an auxiliary result used in the proof of our main results.

Lemma 4.13. *Let P be a set of skew-indecomposable permutations. Then for $n \geq 1$,*

$$\mathrm{uperm}(\mathrm{WI}(\mathrm{U}(B_n), \mathrm{Av}^+(r_u, c_u, P))) \subseteq \mathrm{Av}^{(n)}(r_u, c_u, 1 \oplus P). \tag{4.2}$$

Proof. First, for a skew-indecomposable permutation π, we will show

$$\mathrm{uperm}(\mathrm{WI}(\mathrm{U}(B_n), \mathrm{Av}^+(\pi))) \subseteq \mathrm{Av}^{(n)}(1 \oplus \pi). \tag{4.3}$$

Assume that $\sigma \in \mathrm{uperm}(\mathrm{WI}(\mathrm{U}(B_n), \mathrm{Av}^+(\pi)))$ contains $1 \oplus \pi$. Then, a rectangular region of the staircase grid of σ contains π. As the set of active cells is an independent set of the up-core, the rows and columns are decreasing, and π is skew-indecomposable, π occurs in a single cell. This is a contradiction, since the weights are from $\mathrm{Av}^+(\pi)$.

We will complete the proof by showing that

$$\mathrm{uperm}(\mathrm{WI}(\mathrm{U}(B_n), \mathrm{Av}^+(r_u, c_u))) \subseteq \mathrm{Av}^{(n)}(r_u, c_u).$$

Let σ be a permutation in $\mathrm{uperm}(\mathrm{WI}(\mathrm{U}(B_n), \mathrm{Av}^+(r_u, c_u)))$. If σ contains either r_u or c_u then σ contains one of the mesh patterns m_1 or m_2 in Figure 4.8, by the Shading Lemma. If the cell $(1, 0)$ of m_1 (resp. cell $(0, 1)$ of m_2) contains a point then, by picking the leftmost point in that region, the permutation σ contains an occurrence of m_2 (resp. of m_1) that is below (resp. to the left) of the occurrence we are considering. Repeat this argument on the new occurrence. As σ is finite, we will repeat a finite number of times until we find an occurrence of m_3 or m_4 in Figure 4.8.

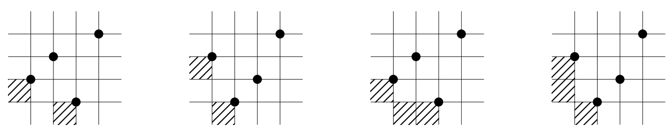


Figure 4.8: A permutation that contains r_u or c_u contains m_1 or m_2, as well as m_3 or m_4.

Assume $\sigma_{i_1}\sigma_{i_2}\sigma_{i_3}\sigma_{i_4}$ is an occurrence of m_3 in σ. Either σ_{i_1} and σ_{i_3} are both left-to-right minima in σ, or both are not left-to-right minima in σ. If they are left-to-right minima of σ then σ_{i_2} and σ_{i_4} are in different columns of the staircase grid, and moreover different rows as rows are decreasing. This implies that two connected cells in $\mathrm{U}(B_n)$ are active, contradicting the fact that an independent set was used. Therefore, σ_{i_1} and σ_{i_3} are not left-to-right minima of σ. There is, therefore, a point (k, σ_k) with $k < i_1$ and $\sigma_k < \sigma_{i_3}$. This new point together with the original occurrence is an occurrence of $1 \oplus r_u$. As r_u is skew-indecomposable, this contradicts Equation (4.3).

Hence, we have shown that σ avoids m_3. A similar argument shows that m_4 is also avoided, and hence r_u and c_u are avoided by σ. □

Theorem 4.14. *Let P be a set of skew-indecomposable permutations. Then* SE *is a bijection between $\mathrm{Av}^{(n)}(r_u, c_u, 1 \oplus P)$ and $\mathrm{WI}(\mathrm{U}(B_n), \mathrm{Av}^+(r_u, c_u, P))$.*

Proof. Let σ be a permutation in $\mathrm{Av}^{(n)}(r_u, c_u, 1 \oplus P)$. By Lemma 4.12, the active cells of the staircase encoding of σ form an independent set of $\mathrm{U}(B_n)$, and the subpermutations in each cell of the staircase encoding are in $\mathrm{Av}(r_u, c_u, P)$. Hence,

$$\mathrm{SE}(\mathrm{Av}^{(n)}(r_u, c_u, 1 \oplus P)) \subseteq \mathrm{WI}(\mathrm{U}(B_n), \mathrm{Av}^{+}(r_u, c_u, P)).$$

By applying SE on both sides of Equation (4.2) in Lemma 4.13, we get by Lemma 4.7

$$\mathrm{WI}(\mathrm{U}(B_n), \mathrm{Av}^{+}(r_u, c_u, P)) \subseteq \mathrm{SE}(\mathrm{Av}^{(n)}(r_u, c_u, 1 \oplus P)).$$

Hence,

$$\mathrm{SE}(\mathrm{Av}^{(n)}(r_u, c_u, 1 \oplus P)) = \mathrm{WI}(\mathrm{U}(B_n), \mathrm{Av}^{+}(r_u, c_u, P)).$$

Since permutations avoiding r_u and c_u have decreasing rows and columns, the map SE is injective when restricted to $\mathrm{Av}^{(n)}(r_u, c_u, 1 \oplus P)$. Therefore, SE is a bijection between $\mathrm{Av}^{(n)}(r_u, c_u, 1 \oplus P)$ and $\mathrm{WI}(\mathrm{U}(B_n), \mathrm{Av}^{+}(r_u, c_u, P))$. □

The following corollary shows how to compute the generating function for any basis covered by the theorem.

Corollary 4.15. *Let P be a set of skew-indecomposable permutations and $A(x)$ be the generating function of* $\mathrm{Av}(r_u, c_u, P)$. *Then* $\mathrm{Av}(r_u, c_u, 1 \oplus P)$ *is enumerated by* $\mathbf{F}(x, A(x) - 1)$, *where* $\mathbf{F}(x, y)$ *is the generating function in Equation* (4.1).

Proof. By Theorem 4.14, $\mathrm{Av}(r_u, c_u, 1 \oplus P)$ is in 1-to-1 correspondence with

$$\bigsqcup_{n \geq 0} \mathrm{WI}(\mathrm{U}(B_n), \mathrm{Av}^{+}(r_u, c_u, P)).$$

Moreover, the size of the permutation obtained is the number of left-to-right minima added to the sizes of the weights of the independent set. This implies $\mathbf{F}(x, A(x) - 1)$ is the generating function for $\mathrm{Av}(r_u, c_u, 1 \oplus P)$. □

Corollary 4.15 can be used to compute the generating function of $\mathrm{Av}(2314, 3124)$, that was first enumerated by [7]. The generating function $A(x)$ for $\mathrm{Av}(2314, 3124)$ satisfies

$$A(x) = \mathbf{F}(x, A(x) - 1). \tag{4.4}$$

Solving gives

$$A(x) = \frac{3 - x - \sqrt{1 - 6x + x^2}}{2}, \tag{4.5}$$

which is the generating function for the large Schröder numbers, which can be found in the Online Encyclopedia of Integer sequences [48] as sequence A006318.

Corollary 4.15 can also be used to enumerate the subclass $\mathrm{Av}(2314, 3124, 1234)$, first enumerated by [44]. In this case, the cells of the independent sets are filled with permutations in Av(2314, 3124, 123) = Av(123). Since the generating function of the latter permutation class is $\frac{1-\sqrt{1-4x}}{2x}$, the generating function of Av(2314, 3124, 1234) is

$$\mathbf{F}\left(x, \frac{1-\sqrt{1-4x}}{2x} - 1\right).$$

There are three different skew-indecomposable permutations of size 3. Those permutations are 123, 132 and 213. Therefore, Theorem 4.14 gives a structural description of Av(2314, 3124) and subclasses obtained by also avoiding any subset of $\{1234, 1243, 1324\}$. This gives 8 permutation classes with bases consisting of only size four patterns. Since, there are 13 skew-indecomposable permutations of size 4, the theorem gives structural description of 2127 bases[1] that contain size 4 and 5 patterns.

Theorem 4.14 extends the number of permutation classes that the up-core describes. A similar method can be used for the down-core to enumerate permutation classes beyond Av(132).

Lemma 4.16. *Let P be a set of sum-indecomposable permutations. Then for $n \geq 1$*

$$\mathrm{dperm}(\mathrm{WI}(\mathrm{D}(B_n), \mathrm{Av}^+(r_d, c_d, P))) \subseteq \mathrm{Av}^{(n)}(r_d, c_d, 1 \oplus P).$$

Theorem 4.17. *Let P be a set of sum-indecomposable permutations. Then* SE *is a bijection between $\mathrm{Av}^{(n)}(r_d, c_d, 1 \oplus P)$ and $\mathrm{WI}(\mathrm{D}(B_n), \mathrm{Av}^+(r_d, c_d, P))$.*

The proofs are left to the reader as they are similar to the proofs of Lemma 4.13 and Theorem 4.14. Corollary 4.18 follows naturally from Theorem 4.17.

Corollary 4.18. *Let P be a set of sum-indecomposable permutations and $A(x)$ be the generating function of $\mathrm{Av}(r_d, c_d, P)$. Then $\mathrm{Av}(r_d, c_d, 1 \oplus P)$ is enumerated by $\mathbf{F}(x, A(x) - 1)$.*

As a consequence of the previous corollary, $A(x)$, the generating function of Av(2413, 3142), first enumerated by [7], also satisfies Equation (4.4) and is given by Equation (4.5). This result can enumerate 8 permutation classes with bases consisting of size four patterns and many more if we consider longer patterns.

It is worth noting that any subclass of Av(2413, 3142) (as well as the class itself) contains finitely many simple permutations and can be enumerated using a more general method called the substitution decomposition, described in Albert and Atkinson [49]. Bassino, Bouvel, Pierrot, Pivoteau, Rossin [50] extended the method to allow for random sampling. We outline briefly in Section 5.3 how the structural description introduced in this chapter can be used to randomly sample in permutation classes, including many with infinitely many simple permutations.

[1]This is the number of bases after removing redundancies.

4.5 Inflating the updown-core

In the previous section, we enumerated Av(2314, 3124) and Av(2413, 3142) and many of their subclasses. However, the intersection of the two permutation classes, namely the subclass Av(2314, 3124, 2413, 3142), cannot be enumerated using the theorems so far. Together these patterns put stricter constraints on the staircase encoding that we have not encountered yet. In this section, we combine different graphs to represent these constraints and, in particular, give the generating function of Av(2314, 3124, 2413, 3142) and many of its subclasses. Again, this permutation class and any subclasses could be enumerated using the substitution decomposition. However, the techniques used in this section are an important stepping stone for the upcoming sections.

To represent the new constraint, we introduce the *column-edges* that connect cells in the same column of a grid and the *row-edges* that connect cells in the same row. More formally:

Definition 4.19.

- *The* column-core *graph* $\mathrm{C}(B_n)$ *is the graph whose vertices are the cells of* B_n *and where there is an edge between cells* (i, j) *and* (k, ℓ) *if* $i \neq k$ *and* $j = \ell$.
- *The* row-core *graph* $\mathrm{R}(B_n)$ *is the graph whose vertices are the cells of* B_n *and where there is an edge between cells* (i, j) *and* (k, ℓ) *if* $i = k$ *and* $j \neq \ell$.

We combine the edges of the four graphs $\mathrm{U}(B_n)$, $\mathrm{D}(B_n)$, $\mathrm{C}(B_n)$ and $\mathrm{R}(B_n)$ in the natural way. For example, the graph $\mathrm{UDC}(B_n)$ has the cells of B_n as vertices and the edges of $\mathrm{U}(B_n)$, $\mathrm{D}(B_n)$ and $\mathrm{C}(B_n)$. Figure 4.9 shows $\mathrm{UDC}(B_4)$.

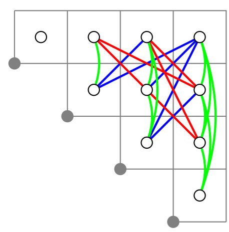


Figure 4.9: The graph $\mathrm{UDC}(B_4)$. The up-edges are blue, the down-edges are red and the column-edges are green.

As we did for the up-core and the down-core with Lemma 4.12, we describe sufficient conditions for the staircase encoding of a permutation to respect the constraints enforced by the column-core and the row-core.

Lemma 4.20. *Let* σ *be a permutation with* n *left-to-right minima and* C *be the set of active cells of the staircase encoding of* σ. *Then* C *is an independent set of*

1. $\mathrm{R}(B_n)$ *if* $\sigma \in \mathrm{Av}(r_u, r_d)$,
2. $\mathrm{C}(B_n)$ *if* $\sigma \in \mathrm{Av}(c_u, c_d)$.

Proof. By Lemma 4.11 if $\sigma \in \mathrm{Av}(r_u, r_d)$, the rows of σ are both increasing and decreasing. Therefore, there is at most one active cell in each row, and so the active cells correspond to an independent set in $\mathrm{R}(B_n)$. The proof of 2. is similar. □

Theorem 4.21. *Let P be a set of permutations. Then the staircase encoding* SE *is a bijection between* $\mathrm{Av}^{(n)}(r_d, c_d, r_u, c_u, 1 \oplus P)$ *and* $\mathrm{WI}(\mathrm{UDRC}(B_n), \mathrm{Av}^+(r_d, c_d, r_u, c_u, P))$.

Proof. By Lemmas 4.12 and 4.20 we have that

$$\mathrm{SE}(\mathrm{Av}(r_d, c_d, r_u, c_u, 1 \oplus P)) \subseteq \mathrm{WI}(\mathrm{UDRC}(B_n), \mathrm{Av}^+(r_d, c_d, r_u, c_u, P)).$$

Let σ be a permutation in $\mathrm{uperm}(\mathrm{WI}(\mathrm{UDRC}(B_n), \mathrm{Av}^+(r_d, c_d, r_u, c_u, P)))$. As any independent set of $\mathrm{UDRC}(B_n)$ is an independent set of $\mathrm{U}(B_n)$ and

$$\mathrm{Av}(r_d, c_d, r_u, c_u, P) \subseteq \mathrm{Av}(r_u, c_u)$$

we have that

$$\mathrm{uperm}(\mathrm{WI}(\mathrm{UDRC}(B_n), \mathrm{Av}^+(r_d, c_d, r_u, c_u, P))) \subseteq \mathrm{uperm}(\mathrm{WI}(\mathrm{U}(B_n), \mathrm{Av}^+(r_u, c_u)))$$

Therefore, by Lemma 4.13 it follows that $\sigma \in \mathrm{Av}(r_u, c_u)$. By observing that uperm and dperm are equivalent when building from an independent set of $\mathrm{RC}(B_n)$, it follows from a symmetric argument and Lemma 4.16 that $\sigma \in \mathrm{Av}(r_d, c_d)$.

Assume that σ contains a pattern $1 \oplus \pi$, where $\pi \in P$. Without loss of generality, we can assume that the 1 in the occurrence is a left-to-right minimum in σ. Then π occurs in the rectangular region of cells north-east of the minimum. By the constraints of $\mathrm{UDRC}(B_n)$, there can be at most one active cell in this region, and so π is contained in this cell. This contradicts the fact that active cells are filled with permutations avoiding π. Therefore, σ avoids $1 \oplus P$, and moreover $\sigma \in \mathrm{Av}^{(n)}(r_d, c_d, r_u, c_u, 1 \oplus P)$. We get

$$\mathrm{uperm}(\mathrm{WI}(\mathrm{UDRC}(B_n), \mathrm{Av}^+(r_d, c_d, r_u, c_u, P))) \subseteq \mathrm{Av}^{(n)}(r_d, c_d, r_u, c_u, 1 \oplus P)$$

or equivalently by applying SE on both sides

$$\mathrm{WI}(\mathrm{UDRC}(B_n), \mathrm{Av}^+(r_d, c_d, r_u, c_u, P)) \subseteq \mathrm{SE}(\mathrm{Av}^{(n)}(r_d, c_d, r_u, c_u, 1 \oplus P)).$$

Since permutations avoiding r_u and c_u have decreasing rows and columns, the map SE is injective when restricted to $\mathrm{Av}^{(n)}(r_u, c_u, r_d, c_d, 1 \oplus P)$. Therefore, SE is a bijection between $\mathrm{Av}^{(n)}(r_u, c_u, r_d, c_d, 1 \oplus P)$ and the image of that set which is $\mathrm{WI}(\mathrm{UDRC}(B_n), \mathrm{Av}^+(r_u, c_u, r_d, c_d, P))$. □

Since the graph $\mathrm{UDRC}(B_n)$ is the same graph as the updown-core of B_n defined in [42], Lemma 4.13 of the same paper gives that the number of independent sets of size k in $\mathrm{UDRC}(B_n)$ is given by the coefficient of $x^n y^k$ in the generating function

$$\mathbf{Y}(x, y) = \frac{1 - x}{x^2 - xy - 2x + 1}.$$

As in Section 4.4, we get a generating function result for the permutation classes $\mathrm{Av}(r_d, c_d, r_u, c_u, 1 \oplus P)$.

Corollary 4.22. *Let P be a set of permutations and $A(x)$ be the generating function of* $\mathrm{Av}(r_d, c_d, r_u, c_u, 1 \oplus P)$. *Then $A(x)$ satisfies*

$$A(x) = \mathbf{Y}(x, B(x) - 1)$$

where $B(x)$ is the generating function of $\mathrm{Av}(r_d, c_d, r_u, c_u, P)$.

Solving the equation $A(x) = \mathbf{Y}(x, A(x) - 1)$ gives

$$A(x) = \frac{x^2 - x - \sqrt{x^4 - 2x^3 + 7x^2 - 6x + 1} + 1}{2x}$$

which is the generating function for $\mathrm{Av}(2413, 3142, 2314, 3124)$. This permutation class was first enumerated by [43] and the sequence appears on OEIS as A078482.

4.6 New cores

To this point we have considered permutation classes that can be described by filling the independent sets of the graphs $\mathrm{U}(B_n)$, $\mathrm{D}(B_n)$ and $\mathrm{UDRC}(B_n)$, which were first used by [42] to enumerate permutation classes avoiding size 3 patterns. In this section, we begin to consider new graphs that were not motivated by permutation classes avoiding size 3 patterns.

We first consider $\mathrm{UDC}(B_n)$. The active cells in the staircase encoding of a permutation σ avoiding c_u and c_d are an independent set of $\mathrm{C}(B_n)$. By Lemma 4.12, they are also an independent set of $\mathrm{U}(B_n)$ and $\mathrm{D}(B_n)$. In order to make the filling of independent sets unique we need that the rows are either increasing or decreasing, *i.e.*, σ avoids r_u or r_d. In this section, we consider additionally avoiding r_u, and delay the discussion of avoiding r_d to Section 4.7.

Theorem 4.23. *Let P be a set of skew-indecomposable permutations. Then* SE *is a bijection between* $\mathrm{Av}^{(n)}(r_u, c_u, c_d, 1 \oplus P)$ *and* $\mathrm{WI}(\mathrm{UDC}(B_n), \mathrm{Av}^+(r_u, c_u, c_d, P))$.

Proof. Let $\sigma \in \mathrm{uperm}(\mathrm{WI}(\mathrm{UDC}(B_n), \mathrm{Av}^+(r_u, c_u, c_d, P)))$. As $\mathrm{UDC}(B_n)$ contains the edges of $\mathrm{U}(B_n)$ we have

$$\mathrm{uperm}(\mathrm{WI}(\mathrm{UDC}(B_n), \mathrm{Av}^+(r_u, c_u, c_d, P))) \subseteq \mathrm{uperm}(\mathrm{WI}(\mathrm{U}(B_n), \mathrm{Av}^+(r_u, c_u, c_d, P)))$$

and so by Lemma 4.13, σ avoids r_u, c_u, $1 \oplus c_d$ and $1 \oplus P$. Suppose that σ contains an occurrence of c_d, then by the Shading Lemma it also has an occurrence of the mesh pattern with the same underlying pattern and cells $(0,2)$ and $(1,0)$ shaded. Further, the avoidance of r_u and $1 \oplus c_d$ imply that there is an occurrence with the cells $(0,1)$ and $(0,0)$ also shaded. Let $\sigma_{i_1}\sigma_{i_2}\sigma_{i_3}\sigma_{i_4}$ be an occurrence of this mesh pattern, shown in Figure 4.10. Both σ_{i_1} and σ_{i_2} are left-to-right minima of σ. Therefore, σ_{i_3} and σ_{i_4} are in two separate active cells, contradicting the fact that the active cells are an independent set of $\mathrm{D}(B_n)$ and $\mathrm{C}(B_n)$. Hence, $\sigma \in \mathrm{Av}(r_u, c_u, c_d, 1 \oplus P)$ and

$$\mathrm{uperm}(\mathrm{WI}(\mathrm{UDC}(B_n), \mathrm{Av}^+(r_u, c_u, c_d, P))) \subseteq \mathrm{Av}^{(n)}(r_u, c_u, c_d, 1 \oplus P).$$

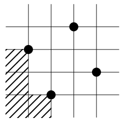


Figure 4.10: The mesh pattern that is contained in the permutation σ if it contains an occurrence of c_d.

By applying SE to both sides we get

$$\mathrm{WI}(\mathrm{UDC}(B_n), \mathrm{Av}^+(r_u, c_u, c_d, P)) \subseteq \mathrm{SE}(\mathrm{Av}^{(n)}(r_u, c_u, c_d, 1 \oplus P)).$$

By Lemmas 4.12 and 4.20,

$$\mathrm{SE}(\mathrm{Av}^{(n)}(r_u, c_u, c_d, 1 \oplus P)) \subseteq \mathrm{WI}(\mathrm{UDC}(B_n), \mathrm{Av}^+(r_u, c_u, c_d, P)).$$

Moreover, by Lemma 4.11 the rows and columns are decreasing, therefore, restricted to $\mathrm{Av}^{(n)}(r_u, c_u, c_d, 1 \oplus P)$, SE is injective and a bijection to its image which is

$$\mathrm{WI}(\mathrm{UDC}(B_n), \mathrm{Av}(r_u, c_u, c_d, P)). \qquad \square$$

In order to use the theorem above for enumerative purposes, we need to find the generating function where the coefficient of $x^n y^k$ is the number of independent sets of size k in $\mathrm{UDC}(B_n)$. We prove a slightly more general statement that tracks the number of rows occupied by the independent set. Although, not required for the permutation classes discussed in this section it will be necessary for the results in Section 4.7.

Proposition 4.24. *The number of independent sets of size k occupying ℓ rows in* $\mathrm{UDC}(B_n)$ *is given by the coefficient of $x^n y^k z^\ell$ in the generating function*

$$\mathbf{G}(x, y, z) = \frac{1 - x - xy}{x^2 y - xyz + x^2 - xy - 2x + 1}.$$

Proof. An independent set in $\mathrm{UDC}(B_n)$ can contain an arbitrary number of vertices in the topmost row, *i.e.*, vertices of the form $(1, j)$. The number of such vertices is called the *degree*. If the degree is 0, then the subgraph induced by the remaining vertices is isomorphic to a smaller core $\mathrm{UDC}(B_{n-1})$. If the degree is not 0, let

$$k = \max\{j : (1, j) \text{ is a vertex of the independent set}\},$$

i.e., k is the column of the rightmost vertex in the topmost row. The independent set cannot contain a vertex (ℓ, m) if $1 < \ell \leq k$ or if $\ell = 1$ and $m \geq k$. This corresponds to the region shaded in gray in Figure 4.11.

Moreover, the vertices $\{(1, j) : 1 \leq j < k\}$ share no edges. We can, therefore, chose independently if they are in the independent set.

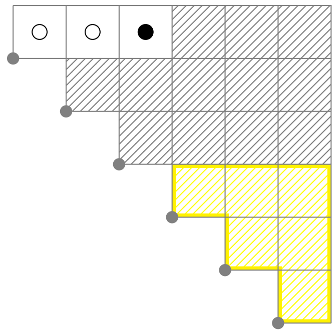


Figure 4.11: A staircase grid with an active cell marked by a black point. The shaded cells are the cells that cannot be added to make an independent set. Cells marked with a circle are disconnected from the graph induced by removing the shaded cells.

The graph induced by the remaining vertices (ℓ, m) with $\ell > k$ and $m > k$, form an instance of the graph $\mathrm{UDC}(B_{n-k})$. In Figure 4.11 this is the yellow region. Hence, $\mathbf{G}(x, y, z)$ satisfies

$$\begin{aligned}\mathbf{G}(x,y,z) &= 1 + x\,\mathbf{G}(x,y,z) + xyz\,\mathbf{G}(x,y,z) + \cdots + x^i y(y+1)^{i-1} z\,\mathbf{G}(x,y,z) + \cdots \\ &= 1 + x\,\mathbf{G}(x,y) + \frac{xyz\,\mathbf{G}(x,y,z)}{1 - x(y+1)}.\end{aligned}$$

Solving this equation gives the closed form claimed in the proposition. □

As the proof of Theorem 4.23 gives a unique encoding of the permutation classes $\mathrm{Av}(r_u, c_u, c_d, 1 \oplus P)$ we derive the following corollary to give their counting sequences.

Corollary 4.25. *Let P be a set of skew-indecomposable permutations and $A(x)$ be the generating function of $\mathrm{Av}(r_u, c_u, c_d, 1 \oplus P)$. Then $A(x)$ satisfies*

$$A(x) = \mathbf{G}(x, B(x) - 1, 1)$$

where $B(x)$ is the generating function of $\mathrm{Av}(r_u, c_u, c_d, P)$.

4.7 Generalizing the fillings

As Section 4.6 considers the basis $\{r_u, c_u, c_d\}$, one could hope that we can handle $\{r_d, c_d, c_u\}$ similarly. Unfortunately, the proof of Theorem 4.23 relies heavily on the fact that c_d is skew-indecomposable. To repeat the argument for $\{r_d, c_d, c_u\}$, one would need c_u to be sum-indecomposable which is not the case. However, tracking an additional statistic on the independent set allows us to enumerate these permutation classes. Even if the permutation classes considered in the section could be enumerated using the substitution decomposition, the tracking we are about to introduce will also be used in sections 4.8, 4.9 and 4.10 for many permutation classes that cannot be enumerated with the substitution decomposition.

If we consider the staircase encoding of $\sigma \in \mathrm{Av}^{(n)}(r_d, c_d, c_u)$, we have from Lemmas 4.12 and 4.20 that the set of active cells form an independent set of $\mathrm{UDC}(B_n)$ and that the rows are increasing. Hence, if one cell in a row contains an occurrence of 312 and is not the rightmost non-empty cell of the row, an occurrence of $c_u = 3124$ is created (see Figure 4.12). If an active cell contains 312 but is the rightmost cell in the row then no c_u pattern is created since the cells above and to the right are empty. Hence, the staircase encoding of a permutation avoiding the pattern c_u avoids 312 in its cells except the rightmost active cell of each row.

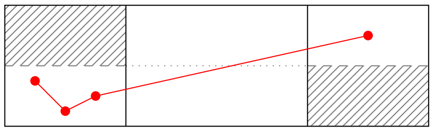


Figure 4.12: An occurrence of $c_u = 3124$ spanning across two cells.

In order to describe the set of staircase encodings of permutations in the class $\mathrm{Av}(r_d, c_d, c_u)$ we enrich our definition of a weighted independent set to a *weighted labelled independent set*. We first define a labelling function r on an independent set. This function maps a vertex v of an independent set I to a label in the set $\{y, z\}$.

$$\mathrm{r}(v, I) = \begin{cases} y & \text{if } v \text{ is not the rightmost cell of } I \text{ in its row,} \\ z & \text{otherwise.} \end{cases}$$

Definition 4.26. *For a graph G on the staircase grid, we define* $\mathrm{WI_r}(G, Y, Z)$ *as the set of weighted independent sets of G where the weight of a vertex v in an independent set I is an element of Y if* $\mathrm{r}(v, I) = y$ *or an element of Z if* $\mathrm{r}(v, I) = z$.

We introduce an operation that removes the last value of a permutation if this value is the maximum of the permutation.

Definition 4.27. *Let* σ *be a permutation. We define the permutation* $\sigma^{\times}$ *as*

$$\sigma^{\times} = \begin{cases} \alpha & \text{if } \sigma = \alpha \oplus 1 \text{ for a permutation } \alpha, \\ \sigma & \text{otherwise.} \end{cases}$$

For example, $3124^{\times} = 312$ and $1432^{\times} = 1432$. For a set of patterns P, let $P^{\times} = \{\pi^{\times} : \pi \in P\}$.

Theorem 4.28. *Let P be a set of permutations such that* $P^{\times}$ *contains only sum-indecomposable permutations. Then* SE *is a bijection between* $\mathrm{Av}^{(n)}(r_d, c_d, c_u, 1 \oplus P)$ *and*

$$\mathrm{WI_r}(\mathrm{UDC}(B_n), \mathrm{Av}^{+}(312, P^{\times}), \mathrm{Av}^{+}(r_d, c_d, c_u, P)).$$

Proof. From Lemmas 4.12, and 4.20 and the discussion above, we have that

$$\mathrm{SE}(\mathrm{Av}^{(n)}(r_d, c_d, c_u, 1 \oplus P)) \subseteq \mathrm{WI_r}(\mathrm{UDC}(B_n), \mathrm{Av}^{+}(312, P^{\times}), \mathrm{Av}^{+}(r_d, c_d, c_u, P)).$$

To show the reverse inclusion, we partition P into two sets depending on whether the permutation ends with its maximum or not. We set

$$P_1 = \{\pi \in P : \pi^\times = \pi\} \text{ and } P_2 = \{\pi \in P : \pi^\times \neq \pi\}.$$

Both $\mathrm{Av}(312, P^\times)$ and $\mathrm{Av}(r_d, c_d, c_u, P)$ are subclasses of $\mathrm{Av}(r_d, c_d, P_1)$, so we get

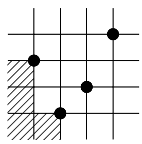
$$\begin{aligned}&\mathrm{dperm}(\mathrm{WI_r}(\mathrm{UDC}(B_n), \mathrm{Av}^+(312, P^\times), \mathrm{Av}^+(r_d, c_d, c_u, P)))\\ &\subseteq \mathrm{dperm}(\mathrm{WI}(\mathrm{UDC}(B_n), \mathrm{Av}^+(r_d, c_d, P_1)))\\ &\subseteq \mathrm{dperm}(\mathrm{WI}(\mathrm{D}(B_n), \mathrm{Av}^+(r_d, c_d, P_1))).\end{aligned}$$

As P_1 contains only sum-indecomposable permutations, by Lemma 4.16, we have that

$$\mathrm{dperm}(\mathrm{WI_r}(\mathrm{UDC}(B_n), \mathrm{Av}^+(312, P^\times), \mathrm{Av}^+(r_d, c_d, c_u, P))) \subseteq \mathrm{Av}^{(n)}(r_d, c_d, 1 \oplus P_1).$$

We also need to show that c_u and $1 \oplus P_2$ are avoided. Let σ be a permutation in $\mathrm{dperm}(\mathrm{WI_r}(\mathrm{UDC}(B_n), \mathrm{Av}^+(312, P^\times), \mathrm{Av}^+(r_d, c_d, c_u, P)))$. We first show that for $\pi \in P_2 \cup \{c_u\}$, $1 \oplus \pi$ does not occur. By the hypothesis, we know that $\pi = \alpha \oplus 1$, with α sum-indecomposable. If $1 \oplus \pi$ is contained in σ, then π is fully contained in a rectangular region of the grid. In such a region, the active cells of the encoding are in the same row. Hence, π is contained in a single row. Since the rows are increasing and α is sum-indecomposable, the only way to split the occurrence is to have an occurrence of α in a cell and an occurrence of 1 in a cell to the right. This is not allowed by the way the vertices can be weighted. Hence, it is contained in a single cell which is also forbidden. Therefore, by contradiction, $1 \oplus \pi$ is avoided for any π in $P_2 \cup \{c_u\}$.

In particular $1 \oplus c_u$ is avoided. Using the Shading Lemma to shade the cells $(0, 2)$ and $(1, 0)$, the avoidance of r_d to shade $(0, 1)$ and the avoidance of $1 \oplus c_u$ to shade $(0, 0)$, we see that if c_u is contained in σ then σ contains an occurrence of the mesh pattern


.

An occurrence of this mesh pattern violates either the column-edges constraint or the up-edges constraints. Thus c_u is avoided, and we have

$$\mathrm{dperm}(\mathrm{WI_r}(\mathrm{UDC}(B_n), \mathrm{Av}^+(312, P^\times), \mathrm{Av}^+(r_d, c_d, c_u, P))) \subseteq \mathrm{Av}^{(n)}(r_d, c_d, c_u, 1 \oplus P),$$

and, by Lemma 4.7,

$$\mathrm{WI_r}(\mathrm{UDC}(B_n), \mathrm{Av}^+(312, P^\times), \mathrm{Av}^+(r_d, c_d, c_u, P)) \subseteq \mathrm{SE}(\mathrm{Av}^{(n)}(r_d, c_d, c_u, 1 \oplus P)).$$

By Lemma 4.11, the rows and columns of a permutation in $\mathrm{Av}^{(n)}(r_d, c_d, c_u, 1 \oplus P)$ are decreasing and therefore, restricted to this set, the map SE is injective. Hence, SE is a bijection between $\mathrm{Av}^{(n)}(r_d, c_d, c_u, 1 \oplus P)$ and

$$\mathrm{SE}(\mathrm{Av}^{(n)}(r_d, c_d, c_u, 1 \oplus P)) = \mathrm{WI_r}(\mathrm{UDC}(B_n), \mathrm{Av}^+(312, P^\times), \mathrm{Av}^+(r_d, c_d, c_u, P)).$$

□

Recall that $\mathbf{G}(x, y, z)$ is the generating function of independent sets of $\mathrm{UDC}(B_n)$ where x tracks the size of the grid, y the size of the independent set, and z the number of rows of the independent set. Therefore, $\mathbf{G}(x, y, \frac{z}{y})$ is the generating function where y tracks the number of cells labelled y by the labelling function r and z tracks the number of cells labelled z. Let C_1 and C_2 be two permutation classes enumerated by $A(x)$ and $B(x)$. Let $F(x)$ be the generating function for the number of weighted independent sets where the cells labelled y are weighted with a non-empty permutation from C_1 and the cells labelled z are weighted with a non-empty permutation from C_2. Then

$$F(x) = \mathbf{G}\left(x, A(x) - 1, \frac{B(x) - 1}{A(x) - 1}\right).$$

This leads to the following enumeration result:

Corollary 4.29. *Let P be a set of permutations such that $P^\times$ contains only permutations that are sum-indecomposable, and $A(x)$ be the generating function of* $\mathrm{Av}(r_d, c_d, c_u, 1 \oplus P)$*. Then $A(x)$ satisfies*

$$A(x) = \mathbf{G}\left(x, C(x) - 1, \frac{B(x) - 1}{C(x) - 1}\right)$$

where $B(x)$ is the generating function of $\mathrm{Av}(r_d, c_d, c_u, P)$ *and $C(x)$ is the generating function of* $\mathrm{Av}(312, P^\times)$*.*

As an example, we derive $A(x)$, the generating function of $\mathrm{Av}(2413, 3142, 3124)$ which was first derived by [44] and appears in the OEIS as A033321. Since the basis is $\{r_d, c_d, c_u\}$, $B(x) = A(x)$. Moreover, $\mathrm{Av}(312)$ is enumerated by the Catalan numbers and $C(x) = \frac{1-\sqrt{1-4x}}{2x}$. Hence, $A(x)$ satisfies

$$A(x) = \mathbf{G}\left(x, A(x) - 1, \frac{C(x) - 1}{A(x) - 1}\right).$$

The equation can be solved to get the explicit form of the generating function.

4.8 Avoiding the row-down and column-up patterns

In this section, we consider permutation classes described by weighted independent sets of the graphs $\mathrm{UD}(B_n)$. This corresponds to removing one of the three patterns from the results in sections 4.6 and 4.7.

Theorem 4.30. *Let P be a set of skew-indecomposable permutations such that all permutations of $P^\times$ are sum-indecomposable. Then* SE *is a bijection between* $\mathrm{Av}^{(n)}(r_d, c_u, 1 \oplus P)$ *and* $\mathrm{WI_r}(\mathrm{UD}(B_n), \mathrm{Av}^+(312, P^\times), \mathrm{Av}^+(r_d, c_u, P))$*.*

Proof. By Lemma 4.11, the map SE is injective when restricted to $\mathrm{Av}^{(n)}(r_d, c_u, 1 \oplus P)$ as each permutation in this set has decreasing columns and increasing rows. Therefore, to show that SE is the claimed bijection, it is sufficient to show that

$$\mathrm{SE}(\mathrm{Av}^{(n)}(r_d, c_u, 1 \oplus P)) = \mathrm{WI_r}(\mathrm{UD}(B_n), \mathrm{Av}^+(312, P^\times), \mathrm{Av}^+(r_d, c_u, P)).$$

By Lemma 4.12 any encoding in $\mathrm{SE}(\mathrm{Av}^{(n)}(r_d, c_u, 1 \oplus P))$ is an independent set of $\mathrm{UD}(B_n)$. Moreover, since the rows are increasing, all active cells but the rightmost of each row avoid 312 and $P^{\times}$ as discussed at the beginning of Section 4.7. This implies

$$\mathrm{SE}(\mathrm{Av}^{(n)}(r_d, c_u, 1 \oplus P)) \subseteq \mathrm{WI_r}(\mathrm{UD}(B_n), \mathrm{Av}^{+}(312, P^{\times}), \mathrm{Av}^{+}(r_d, c_u, P)).$$

Take I in $\mathrm{WI_r}(\mathrm{UD}(B_n), \mathrm{Av}^{+}(312, P^{\times}), \mathrm{Av}^{+}(r_d, c_u, P))$. We consider σ, the permutation obtained from I by building the permutation with decreasing columns and increasing rows. We show that σ is in $\mathrm{Av}^{(n)}(r_d, c_u, 1 \oplus P)$. We start by showing that σ avoids $1 \oplus r_d$, $1 \oplus c_u$ and $1 \oplus P$. In an occurrence of any of these patterns in σ, we can assume that the 1 is a left-to-right minimum. Hence, we have to show that r_d, c_u and P are avoided in the square formed by the set of cells that are north and east of a left-to-right minimum. Let π be any pattern in $\{r_d, c_u\} \cup P$. We know that π is skew-indecomposable and that $\pi^{\times}$ is sum-indecomposable.

Assume that π occurs in a square region of the cells that are north and east of a left-to-right minimum. We consider the rightmost column in the region that contains a point of the occurrence of π. In this column, we consider the topmost cell that contains such a point. In Figure 4.13, this cell is colored blue. By construction, the gray region is empty. Moreover, there is no point of the permutation in the

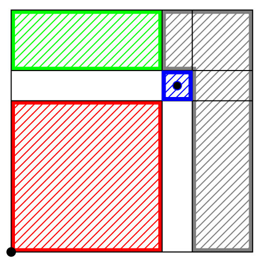
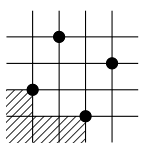
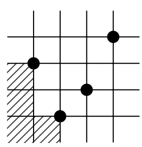


Figure 4.13: Decomposition of an occurrence of π in a square region.

green region (resp. the red region) because the set of active cells is an independent set of $\mathrm{D}(B_n)$ (resp. $\mathrm{U}(B_n)$). Since the columns are decreasing, if the occurrence contains any point in a cell below the blue one, π is skew-decomposable. Hence, the only active cell in the column is the blue one. Finally, since the row is increasing and $\pi^{\times}$ is sum-indecomposable, π is either fully contained in the blue cell, which is forbidden, or $\pi^{\times}$ is fully contained in a cell that is not the rightmost active one in the row, which is also forbidden. Consequently, $1 \oplus r_d$, $1 \oplus c_u$ and $1 \oplus P$ are avoided. Using shading arguments as done in previous proofs, we can show that if σ contains an occurrence of r_d or c_u then σ contains an occurrence of either of the mesh patterns

 or  .

An occurrence of either of these pattern violates the edge, increasing rows, or decreasing column constraints. Hence, σ is in $\mathrm{Av}^{(n)}(r_d, c_u, 1 \oplus P)$ and

$$\mathrm{SE}(\mathrm{Av}^{(n)}(r_d, c_u, 1 \oplus P)) \supseteq \mathrm{WI_r}(\mathrm{UD}(B_n), \mathrm{Av}^{+}(312, P^{\times}), \mathrm{Av}^{+}(r_d, c_u, P)).$$

This proves that the image of $\mathrm{Av}^{(n)}(r_d, c_u, 1 \oplus P)$ under SE is the set claimed in the theorem. □

In order to enumerate these permutation classes, we first enumerate the independent sets of $\mathrm{UD}(B_n)$ while keeping track of the number of rows.

Proposition 4.31. *The number of independents set of size k in $\mathrm{UD}(B_n)$ occupying ℓ rows is given by the coefficient of $x^n y^k z^\ell$ in the generating function that satisfies*

$$\mathbf{W}(x,y,z) = 1 + x\,\mathbf{W}(x,y,z) + D(x,y,z)\,\mathbf{W}(x,y,z),$$

where

$$D(x,y,z) = \frac{xyz(xy^2z - x + 1)}{(xyz + x - 1)(xy + x - 1)}.$$

Proof. We consider the topmost row of B_n. If it contains no vertex of the independent set, then we are looking at the independent set in a smaller core graph and it contributes $x\,\mathbf{W}(x,y,z)$ to $\mathbf{W}(x,y,z)$.

If the topmost row contains vertices of the independent set, we consider its rightmost vertex. The vertex is highlighted in blue in Figure 4.14. The cells in the yellow region do not contain any vertices of the independent set because they are connected to the blue cell by an edge. The vertices of the independent set are, therefore, in the hook formed by the white and blue cells, and the pink region. The pink region is completely disconnected from the hook and hence the vertices of the independent set in this region correspond to an independent set of a smaller core.

To find $\mathbf{W}$, we need to enumerate the independent set of the hook that contains the corner cell of the hook. We say that the *leg length* of the hook is the number of cells in the horizontal strip. If the coefficient of $x^n y^k z^\ell$ in $D(x,y,z)$ is the number of such sets of k vertices using ℓ rows in the hook of leg length n, then

$$\mathbf{W}(x,y,z) = 1 + x\,\mathbf{W}(x,y,z) + D(x,y,z)\,\mathbf{W}(x,y,z).$$

To find the generating function $D(x,y,z)$ we first notice that any vertices that we take in the vertical leg add to the row count of the independent set while vertices in the horizontal leg do not change the row count. First, the case where the hook is a single cell contributes xyz to $D(x,y,z)$.

Second, if the leg length of the hook is greater than 1, the cells at the end of each leg are not connected to the hook by any edges of the graph. Hence, we have complete freedom to put them in the independent set. Therefore, the second case is of the form $(xyz)(xyz + x)(y + 1)(\cdots)$. As they are accounted for, we completely ignore the corner cell and the two cells at the end of the leg and focus on enumerating the part of the independent set in the remaining cells.

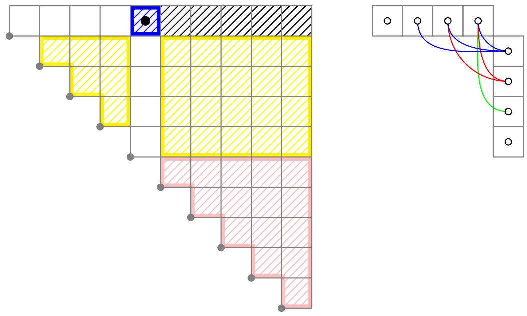


Figure 4.14: Decomposition of an independent set according to the rightmost vertex in the top row. The picture on the left shows the whole staircase. The picture on the right shows the induced subgraph of the hook with a corner in the blue cell.

If no cell of the vertical leg is in the independent set, then any cell of the horizontal leg can be in it. Otherwise, if there are i cells above the topmost active cell of the vertical leg then the i leftmost cells of the horizontal leg are the only cells from that leg that can be in the independent set (see Figure 4.15).

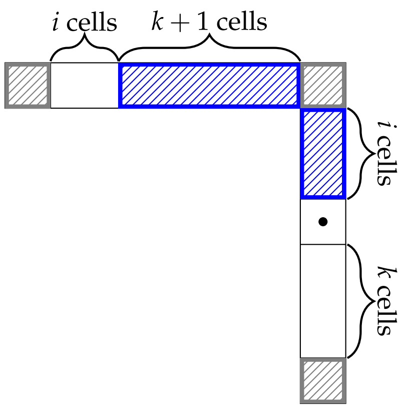


Figure 4.15: The structure of an independent set in the hook. The only active cells are in the regions that are not shaded.

Hence,

$$\begin{aligned} D(x,y,z) &= xyz + (xyz)(xyz+x)(y+1) \\ &\quad \cdot \left(\frac{1}{1-x(y+1)} + \frac{xyz}{(1-x(y+1))(1-x(1+yz))} \right) \\ &= \frac{xyz(xy^2z - x + 1)}{(xyz+x-1)(xy+x-1)} \end{aligned}$$

which completes the proof. □

Using the same reasoning as in Section 4.7 we derive the following enumeration result:

Corollary 4.32. *Let P be a set of skew-indecomposable permutations such that all permutations in $P^{\times}$ are sum-indecomposable and $A(x)$ be the generating of* $\mathrm{Av}(r_d, c_u, 1 \oplus P)$. *Then $A(x)$ satisfies*

$$A(x) = \mathbf{W}\left(x, C(x) - 1, \frac{B(x) - 1}{C(x) - 1}\right)$$

where $B(x)$ is the generating function of $\mathrm{Av}(r_d, c_u, P)$ *and $C(x)$ is the generating function of* $\mathrm{Av}(312, P^{\times})$.

4.9 Avoiding r_d and 2134

In this section, we consider the pattern 2134 that has not been considered yet. As when we analysed r_u, c_u, r_d and c_d, we consider the two black points of Figure 4.16 as left-to-right minima of the permutation. Then, we study the effect of the two red points on the staircase encoding of the permutation, its rows and its columns. An occurrence of 2134 where the two black points are left-to-right minima cannot have a point in the cell of the leading diagonal of the grid. Hence, the pattern does not enforce any restrictions on those cells. However, in the remaining cells, the pattern 2134 has the same effect as 123 has on the grid B_{n-1} since there are always two left-to-right minima to the left and below those cells.

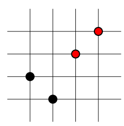


Figure 4.16: The pattern 2134.

Let $G(B_{n-1})$ be a graph with cells of B_{n-1} as vertices. Let S be a subset of B_n. For the remainder of the chapter, we will make a small abuse of the definition and say that S is an independent set of $G(B_{n-1})$ if the set

$$\{(x, y-1) : (x, y) \in S \text{ and } x \neq y\}$$

is an independent set of $G(B_{n-1})$. In other words, a subset of B_n is an independent set of a graph on B_{n-1} if it is an independent set of the graph obtained by overlaying $G(B_{n-1})$ on B_n as in Figure 4.17.

Using similar arguments as we did for Lemmas 4.11, 4.12 and 4.20, we can prove the following lemmas.

Lemma 4.33. *Let σ be a permutation in* $\mathrm{Av}^{(n)}(2134)$. *We consider the set C of active cells of the staircase encoding of σ that are not in the main diagonal of the grid. Then*

- *C is an independent set of $U(B_{n-1})$*
- *cells in C contain decreasing permutations*

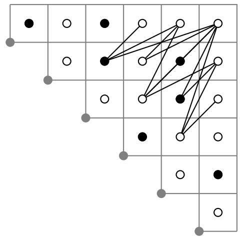
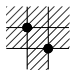


Figure 4.17: The graph $\mathrm{U}(B_5)$ on the staircase grid B_6. The black vertices form an independent set of $\mathrm{U}(B_5)$ for the grid B_6.

Moreover, if we remove the cells of the leading diagonal from σ, the rows and columns are decreasing.

Lemma 4.34. *Let σ be a permutation in $\mathrm{Av}^{(n)}(2134, r_d)$. The set of active cells of the staircase encoding of σ is and independent set of $\mathrm{R}(B_{n-1})$.*

For the rest of this section, we let P be set of patterns, such that for all π in P:

- π avoids  , and
- $\pi \notin \mathcal{S} \oplus (\mathrm{Av}^+(12) \backslash \{1\})$.

Note, r_d and 2134 satisfy these conditions. As further examples, the permutations 312 and 1423 also satisfy the conditions.

For two graphs on staircase grids of different sizes, we define the *merge* of those graphs by gluing them by the top right corner cell. Figure 4.18 shows an example of a merge. The merge of two graphs A and B is denoted $A \vee B$.

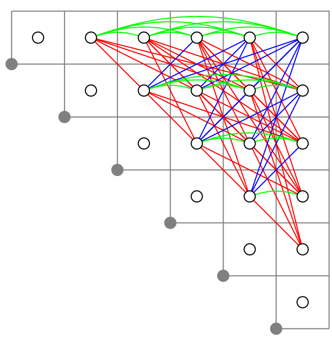


Figure 4.18: The merge of $\mathrm{U}(B_5)$, $\mathrm{R}(B_5)$ and $\mathrm{D}(B_6)$. The green edges come from $\mathrm{R}(B_5)$, the blue ones from $\mathrm{U}(B_5)$, and the red ones from $\mathrm{D}(B_6)$.

In order to describe the structure of the staircase encodings of the permutations in $\mathrm{Av}(r_d, 2134, 1 \oplus P)$, we define a labelling ϕ where the set of labels is $\{y, z, s, t\}$.

For a subset I of the staircase grid and vertex v of this set, we let

$$\phi(v,I)=\begin{cases} y & \text{if } v \text{ is not in the leading diagonal,}\\ z & \text{if there is a } v'\in I \text{ in the same column,}\\ s & \text{if there is a } v'\in I \text{ that is north east of } v,\\ t & \text{otherwise.}\end{cases}$$

Figure 4.19 shows an independent set labelled with ϕ.

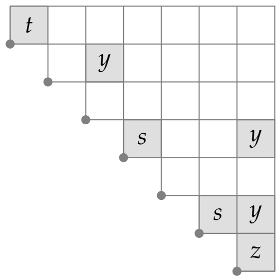


Figure 4.19: The labelling of an independent set with ϕ.

As we did for $\mathrm{WI_r}$, we define $\mathrm{WI}_\phi(G,Y,Z,S,T)$, as the set of weighted independent sets of G such that for a vertex v in the independent set I, the weights of v is an element of

- Y if $\phi(v,I)=y$,
- Z if $\phi(v,I)=z$,
- S if $\phi(v,I)=s$, and
- T if $\phi(v,I)=t$.

During the rest of the section we consider the set

$$\mathrm{WI}_\phi(\,\mathrm{D}(B_n)\vee\mathrm{UR}(B_{n-1}),\mathrm{Av}^+(12),\mathrm{Av}^+(r_d,2134,P)\backslash\{1\},\\ \mathrm{Av}^+(213,P^\times),\mathrm{Av}^+(r_d,2134,P)).$$

For sake of brevity, we name it $\mathcal{I}_n$ in this section.

Theorem 4.35. *There is a bijection between* $\mathrm{Av}^{(n)}(r_d,2134,1\oplus P)$ *and* $\mathcal{I}_n$.

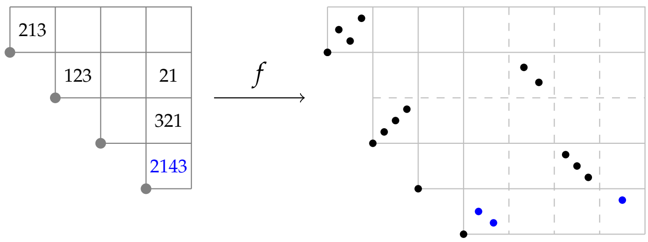
Proof. Let I be a weighted independent set in $\mathcal{I}_n$. Let E be the staircase encoding such that cell v contains the corresponding permutation, except when $\phi(v,I)=z$. In this case, we write the weight as $\alpha m\beta$ where m is the maximum, and in this cell of E, we add $\alpha\beta$.

Define f to be a map which maps I to the permutation $f(I)$ with staircase encoding E such that

- the rows of $f(I)$ are increasing,

- excluding points in the leading diagonal, the columns are decreasing,
- in an active cell v labelled z, with weight $\alpha m \beta$, α is to the left and β is to the right of the points in the column.

Figure 4.20 shows the map f applied to an independent set.



(15)(17)(16)(18)9(10)(11)(12)5132(14)(13)8764

Figure 4.20: The map f from $\mathcal{I}_n$ to $\mathcal{S}$.

We will show that f is the bijection desired. Assume that $\sigma = f(I)$ contains $1 \oplus \pi$ for some $\pi \in P$. If σ contains $1 \oplus \pi$, then it contains an occurrence where the 1 in the occurrence is a left-to-right minimum in σ. Therefore, σ contains an occurrence π in a rectangular region of the staircase grid. This region is pictured in Figure 4.21 (a) with the cell in the leading diagonal in blue.

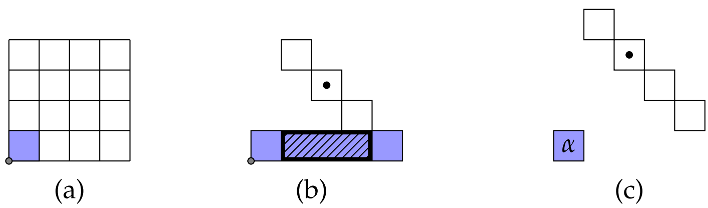
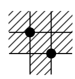
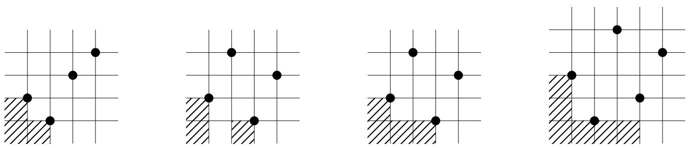


Figure 4.21: Decomposition of the rectangular region containing the occurrences of π.

In this region, without loss of generality we can assume the cell in the lower left corner is in the leading diagonal. The permutations that fill the active cells avoid P, therefore, the occurrence has points in at least two cells. In this region, consider the leftmost cell not in the leading diagonal containing a point of the occurrence of π. If this cell is in the first column, then by the definition of I the columns to the right are not active in $f(I)$. The column consists of a decreasing permutation in the cells not in the leading diagonal (see Figure 4.21 (b)). As π avoids , the occurrence can use exactly one point in these cells, say k. Therefore, an occurrence of π is of the form $\alpha k \beta$, where the α and β are in the cell in leading diagonal, which contradicts the definition of f.

Otherwise, the leftmost cell not in the leading diagonal is not in the first column. By the definition of I, the columns to the right and left are empty if they are not in the diagonal. Moreover, this column is decreasing as shown in Figure 4.21 (c). Therefore, since the blue cell avoids $\pi^{\times}$, it follows that $\pi \in \mathcal{S} \oplus (\text{Av}^{+}(12)\backslash\{1\})$, contradicting the second condition of P. Hence, we have shown that $f(I)$ avoids $1 \oplus \pi$.

If σ contains an occurrence of 2134 then it either contains an occurrence of $1 \oplus 2134$ or an occurrence of $(2134, \{(0,0)\})$. By the Shading Lemma, the latter implies an occurrence of m_1, see Figure 4.22. If m_1 occurs then the 2 and the 1 of the occurrence are left-to-right minima of the permutation. The 3 and 4 of the occurrence violate either a decreasing cell constraint if they are in the same cell or an up-edge or decreasing column constraint if they are not. Hence, if σ contains an occurrence of 2134 it also contains an occurrence of $1 \oplus 2134$. As 2134 satisfies the conditions of P this implies σ avoids 2134.



m_1 m_2 m_3 m_4

Figure 4.22: Mesh patterns that do not occur in σ.

If σ contains an occurrence of r_d then by a similar argument as above, it either contains an occurrence of $1 \oplus r_d$ or m_2. Moreover, if σ contains an occurrence of m_2, we can find an occurrence of m_3 or m_4. An occurrence of m_3 would violate the row increasing constraint or a down-edge constraint. The 5 and the 2 in an occurrence of m_4 are in the same column of the staircase grid as they cannot have any left-to-right minima between them. The 5 and 4 in an occurrence of m_4 are also in the same column, otherwise they violate a down-edge or a row-edge constraint. Hence, since the 2 in an occurrence is in a different row than the 5 and 4, we have a violation of the way we build the column. Therefore, m_4 does not occur in σ and σ avoids m_2. We conclude that if σ contains r_d then it contains $1 \oplus r_d$. As r_d satisfies the conditions of P this implies σ avoids r_d.

The injectivity of f follows from the uniqueness of the map. For surjectivity, we consider a permutation σ in $\text{Av}^{(n)}(r_d, 2134, 1 \oplus P)$. The rows of σ are increasing by Lemma 4.11. By Lemma 4.33, the cells labelled y in σ contain decreasing permutations. Cells labelled s avoid 213 and $P^{\times}$ since there is a guaranteed point of the permutation to the north east of the points in those cells. The cells labelled z and t avoid r_d, 2134 and P. Moreover, the active cells of σ form an independent set of $\text{D}(B_n) \vee \text{UR}(B_{n-1})$ by Lemma 4.33 and 4.34.

We study how cells in the same column interact. We consider a column of the staircase grid. By Lemma 4.33, except for the bottommost cell, the column is decreasing and each cell contains a decreasing sequence. If there is a point in the bottom cell with index between two points of the decreasing sequence, then it creates an occurrence of r_d. Hence, the bottommost cell can only have points on both sides of the decreasing sequences. Figure 4.23 shows a typical column. In the bottommost cell, any point in the gray region would create an r_d pattern. Moreover, the content of this cell cannot create one of the forbidden patterns with one of the points above. Hence, it can only split in a place where a new maximum could be added without creating a pattern in $\{r_d, 2134\} \cup P$. The content of this cell comes from a permutation $\alpha m \beta \in \mathrm{Av}(r_d, 2134, P)$ where m is the maximum, α is placed on the left of the decreasing sequence and β on the right.

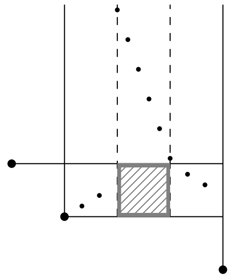


Figure 4.23: A typical column for a permutation avoiding 2143 and r_d.

This shows that σ can be obtained from an element of $\mathcal{I}_n$ by applying f. □

To compute the generating function of $\mathrm{Av}(2134, r_d, 1 \oplus P)$, we need to compute the generating function for the independent set of $\mathrm{D}(B_n) \vee \mathrm{UR}(B_{n-1})$ for $n \in \mathbb{N}$. For these sets we track the number of vertices with each label by a different variable.

Proposition 4.36. *Let* $\mathbf{H}(x,y,z,s,t)$ *be the generating function of independent sets of* $\mathrm{D}(B_n) \vee \mathrm{UR}(B_{n-1})$ *such that the variable* y, z, s, t *track the number of vertices with labels* y, z, s, t *in the set. Then* $\mathbf{H}(x,y,z,s,t)$ *satisfies*

$$\mathbf{H}(x,y,z,s,t) = 1 + x(1+t)\,\mathbf{H}(x,y,z,s,t) + \frac{x^2 y(s+1)(z+1)}{1 - x(s+1)(y+1)}\,\mathbf{H}(x,y,z,s,t).$$

Proof. We observe that the vertices in the leading diagonal of $\mathrm{D}(B_n) \vee \mathrm{UR}(B_{n-1})$ are disconnected from the graph. Hence, they can be freely added or removed from any independent set.

Because of the row constraint on B_{n-1}, the topmost row can contain at most one vertex that is not in the leading diagonal. First, if the independent set does not contain such a vertex then it contributes $x(1+t)\,\mathbf{H}(x,y,z,s,t)$ to $\mathbf{H}$.

Otherwise, the graph decomposes as shown on Figure 4.24 and we get a contribution of

$$\frac{x^2 y(s+1)(z+1)}{1 - x(s+1)(y+1)}\,\mathbf{H}(x,y,z,s,t).$$

Hence, $\mathbf{H}(x,y,z,s,t)$ satisfies

$$\mathbf{H}(x,y,z,s,t) = 1 + x(1+t)\,\mathbf{H}(x,y,z,s,t) + \frac{x^2 y(s+1)(z+1)}{1 - x(s+1)(y+1)}\,\mathbf{H}(x,y,z,s,t)$$

as claimed in the proposition. □

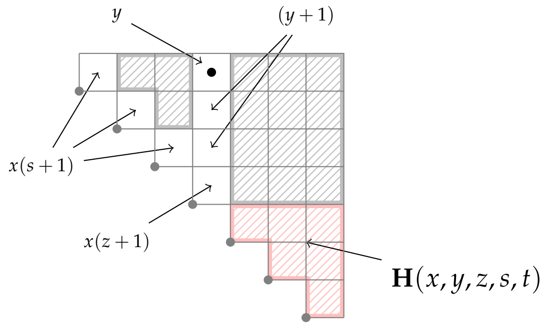


Figure 4.24: The decomposition of an independent set of $\mathrm{D}(B_n) \vee \mathrm{UR}(B_{n-1})$ when a cell in the topmost row is active.

From Theorem 4.35 and Proposition 4.36, we get the counting sequence of $\mathrm{Av}(r_d, 2134, 1 \oplus P)$.

Corollary 4.37. *The generating function of* $\mathrm{Av}(2134, 2413, 1 \oplus P)$ *is*

$$\mathbf{H}\left(x, \frac{x}{1-x}, \frac{B(x)-(1+x)}{x}, C(x)-1, B(x)-1\right)$$

where

- $B(x)$ *is the generating function of* $\mathrm{Av}(2134, 2413, P)$*,*
- $C(x)$ *is the generating function of* $\mathrm{Av}(213, P^{\times})$*.*

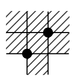
Using the corollary above, we can compute $A(x)$, the generating function of $\mathrm{Av}(2134, 2413)$ that was first enumerated by [19]. In this example, P is empty. Hence, $B(x) = A(x)$ and $C(x)$ is the generating function for the Catalan numbers. We get that the generating function $A(x)$ satisfies

$$A(x) = \mathbf{H}\left(x, \frac{x}{1-x}, \frac{A(x)-(1+x)}{x}, C(x)-1, A(x)-1\right).$$

This equation can be solved explicitly to find the counting sequence that appears in OEIS as A165538.

4.10 Avoiding r_u and 2143

Using similar techniques as in the previous section we enumerate permutation classes of the form $\mathrm{Av}(r_u, 2143, 1 \oplus P)$ where each pattern π in P satisfies

- π avoids  , and
- $\pi \notin \mathcal{S} \ominus \mathrm{Av}^{+}(21)$.

For the entire section, we let P be such a set.

We look at the pattern 2143 in Figure 4.25 as we did for 2134. If we consider the two black points as left-to-right minima of a permutation, then the two red points will enforce a down-core structure on the staircase grid except for the leading diagonal. The following lemmas follow from similar arguments as in Lemmas 4.11, 4.12 and 4.20.

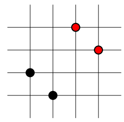


Figure 4.25: The pattern 2143.

Lemma 4.38. *Let σ be a permutation in $\mathrm{Av}^{(n)}(2143)$. We consider the set C of active cells of the staircase encoding of σ that are not in the main diagonal of the grid. Then*

- *C is an independent set of $D(B_{n-1})$,*
- *cells in C contain increasing permutations.*

Moreover, if we remove the cells of the leading diagonal from σ, the rows and columns are increasing.

Lemma 4.39. *Let σ be a permutation in $\mathrm{Av}^{(n)}(2143, r_u)$. The set of active cells of the staircase encoding of σ is an independent set of $\mathrm{R}(B_{n-1})$.*

From the two previous lemmas and Lemma 4.12, we know that for σ in the set $\mathrm{Av}^{(n)}(r_u, 2143)$ the active cells of $\mathrm{SE}(\sigma)$ are an independent set of $\mathrm{U}(B_n) \vee \mathrm{DR}(B_{n-1})$. To describe the structure of the staircase encoding of the permutations in $\mathrm{Av}(r_u, 2143, 1 \oplus P)$ we introduce new labelled weighted independent sets. First, for a subset I of the staircase grid and a vertex v of I, we set

$$\psi(v, I) = \begin{cases} y & \text{if } v \text{ is not in the leading diagonal,} \\ z & \text{if there is a } v' \in I \text{ in the same column,} \\ s & \text{otherwise.} \end{cases}$$

Figure 4.26 shows of a independent set labelled with ψ.

We define $\mathrm{WI}_\psi(G, Y, Z, S)$ as the set of weighted independent sets of G such that for a vertex v in the independent set I the weight of v is an element of

- Y if $\psi(v, I) = y$,
- Z if $\psi(v, I) = z$, and
- S if $\psi(v, I) = s$.

Theorem 4.40. *There is a bijection between $\mathrm{Av}^{(n)}(r_u, 2143, 1 \oplus P)$ and*

$$\mathrm{WI}_\psi(\mathrm{U}(B_n) \vee \mathrm{DR}(B_{n-1}), \mathrm{Av}^+(21), \mathrm{Av}^+(r_u, 2143, P)\backslash\{1\}, \mathrm{Av}^+(r_u, 2143, P)).$$

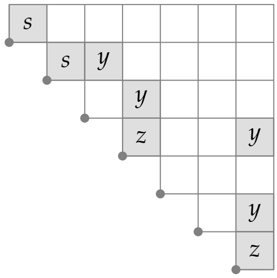


Figure 4.26: The labelling with ψ of an independent set.

Proof. For conciseness, we denote with $\mathcal{I}_n$ the set of weighted independent sets stated in the theorem. Let I be a weighted independent set in $\mathcal{I}_n$.

Let E be the staircase encoding such that a cell v contains the same permutation as in I, except when $\phi(v, I) = z$. In this case, we write the weight as $\alpha m \beta$ where m is the maximum, and add $\alpha\beta$ to cell v.

Define f to be a map which maps I to the permutation $f(I)$ with staircase encoding E such that

- the rows of $f(I)$ are decreasing,
- excluding points in the leading diagonal, the columns are increasing,
- in an active cell v labelled z, with weight $\alpha m \beta$, α is to the left and β is to the right of the points in the column.

Figure 4.27 shows the map f applied to an independent set.

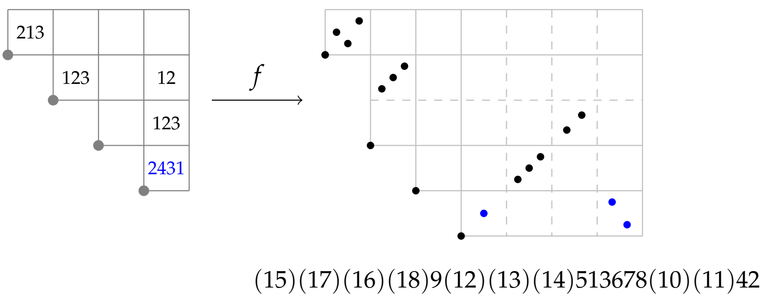
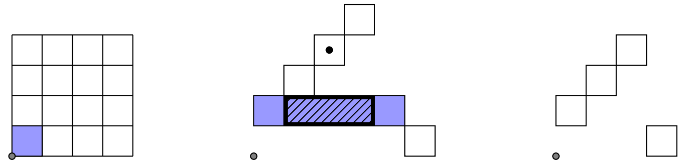


Figure 4.27: The map f from $\mathcal{I}_n$ to $\mathcal{S}$.

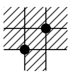
We show that f is the desired bijection. Assume that $\sigma = f(I)$ contains $1 \oplus \pi$ for some $\pi \in P$. If σ contains $1 \oplus \pi$, then it contains an occurrence where the 1 in the occurrence is a left-to-right minimum in σ. Therefore, σ contains an occurrence π in a rectangular region of the staircase grid. This region is pictured in Figure 4.28 (a) with the cell in the leading diagonal colored blue.

In this region, without loss of generality, we can assume the cell in the lower left corner is in the leading diagonal. The permutations contained in each active



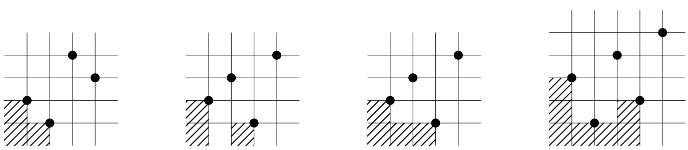
(a) (b) (c)

Figure 4.28: Decomposition of the rectangular region containing the occurrences of π.

cell avoid P, therefore, the occurrence has points in at least two cells. In this region, consider the leftmost cell not in the leading diagonal containing a point of the occurrence of π. If this cell is in the first column then, by the definition of I, only cells in the leftmost column and the bottommost row can be active. Moreover, the row-edges imply that only one cell in the bottom row that is not in the leading diagonal can be active. The column consists of an increasing permutation in the cells not in the leading diagonal and the rows are decreasing (see Figure 4.28 (b)). As π is not in $\mathcal{S} \ominus \mathrm{Av}^+(21)$, the bottommost cell in this figure is empty. Also, as π avoids  , the occurrence can use exactly one point in the remaining white cells, say k. Therefore, the occurrence of π is of the form $\alpha k \beta$, where the α and β are in the cell in leading diagonal, which contradicts the definition of f.

Otherwise, the leftmost cell not in the leading diagonal is not in the first column. By the definition of I, only the column of this cell and the bottommost row can be active. Moreover, this column is increasing and the bottom row is decreasing. Only one white cell of the bottom row can be active because of the row-edges. This is shown in Figure 4.28 (c). Therefore, it follows, since $\pi \notin \mathcal{S} \ominus \mathrm{Av}^+(21)$, that the bottommost cell is empty. Hence π is an increasing permutation, which contradicts the first condition on P. Hence, we have shown that $f(I)$ avoids $1 \oplus \pi$.

If σ contains an occurrence of 2143 then it either contains an occurrence of $1 \oplus 2143$ or an occurrence of $(2143, \{(0,0)\})$. By the Shading Lemma, the latter implies an occurrence of m_1 (see Figure 4.29). If m_1 occurs then the 2 and the 1 of the occurrence are left-to-right minima of the permutation. The 3 and 4 of the occurrence violate either an increasing cell constraint if they are in the same cell or a down-edge constraint if they are not. Hence, if σ contains an occurrence of 2143 it also contains an occurrence of $1 \oplus 2143$. As 2143 satisfies the conditions of P this implies σ avoids 2143.



m_1 m_2 m_3 m_4

Figure 4.29: Mesh patterns that do not occur in σ.

If σ contains an occurrence of r_u then by a similar argument as above, it either contains an occurrence of $1 \oplus r_u$ or m_2. Moreover, if σ contains an occurrence of m_2, we can find an occurrence of m_3 or m_4. An occurrence of m_3 would violate the row decreasing constraints or the up-edges constraints. The 4 and the 2 in an occurrence of m_4 are in the same column of the staircase grid since they cannot have any left-to-right minima between them. The 5 and 4 in an occurrence of m_4 are also in the same column otherwise they violate an up-edge constraint. Hence, since the point at index 4 is in a different row than the ones at index 3 and 5, we have a violation of the way we build the column. Therefore, m_4 does not occur in σ and σ avoids m_2. We conclude that if σ contains r_u then it contains $1 \oplus r_u$. As r_u satisfies the conditions of P, this implies σ avoids r_u.

The injectivity of f follows from the uniqueness of the map. For surjectivity, we consider a permutation σ in $\mathrm{Av}^{(n)}(r_u, 2143, 1 \oplus P)$. The rows of σ are decreasing by Lemma 4.11. By Lemma 4.38, the cells labelled y in σ contain increasing permutations. The cells labelled z and s avoid r_u, 2143 and P. Moreover, the active cells of σ form an independent set of $\mathrm{U}(B_n) \vee \mathrm{DR}(B_{n-1})$ by Lemmas 4.38 and 4.39.

Finally, we need to consider the points in a column of the staircase grid. By Lemma 4.38, except for the bottommost cell, the column is increasing and each cell contains an increasing permutation. If there is a point in the bottom cell with index between two points of the increasing sequence, then it creates an occurrence of r_u. Hence, the bottommost cell can only have points on either side of the increasing sequence. Figure 4.30 shows a typical column. In the bottommost cell, any point

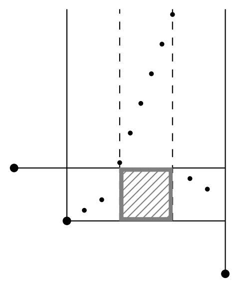


Figure 4.30: A typical column for a permutation avoiding 2143 and r_u.

in the gray region would create an r_u pattern. Moreover, the points in this cell together with another point in this column cannot create one of the forbidden patterns. Hence, it can only be split in a place where a new maximum could be added without creating a pattern in $\{r_u, 2143\} \cup P$. Therefore, the points in the cell are of the form $\alpha\beta$ where α is to the left of the other points in the column, β is to the right of the other points in the column, and $\alpha m \beta \in \mathrm{Av}(r_u, 2143, P)$.

Therefore we have shown that σ can be obtained from an element of $\mathcal{I}_n$ by applying f. $\square$

To compute the generating function of $\mathrm{Av}(r_u, 2143, 1 \oplus P)$, we need to compute the generating function for the independent sets of $\mathrm{U}(B_n) \vee \mathrm{DR}(B_{n-1})$ for any natural number n. For these sets we track the number of vertices and their labels with different variables.

Proposition 4.41. *Let* $\mathbf{J}(x,y,z,s)$ *be the generating function of independent sets of the graph* $\mathrm{D}(B_n) \vee \mathrm{UR}(B_{n-1})$ *such that the variables* y, z, s *track the number of vertices with label* y, z, s *in the set. Then* $\mathbf{J}(x,y,z,s)$ *satisfies*

$$\mathbf{J}(x,y,z,s) = 1 + x(s+1)\,\mathbf{J}(x,y,z,s) + \frac{xy(z+1)(\mathbf{J}(x,y,z,s)-1)}{1-x(y+1)}.$$

Proof. We consider the top row of the graph. The leftmost cell is completely disconnected from the graph, hence, it can be freely added to the independent set. The row constraint on B_{n-1} implies the topmost row contains at most one vertex that is not in the leading diagonal. First, if the independent set does not contain such a vertex then it contributes $x(s+1)\,\mathbf{J}(x,y,z,s)$ to $\mathbf{J}$.

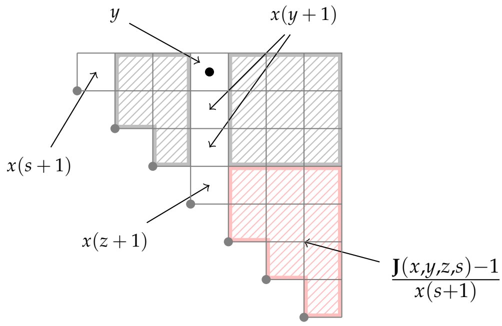


Figure 4.31: The decomposition of an independent set of $\mathrm{U}(B_n) \vee \mathrm{DR}(B_{n-1})$ when a cell in the topmost row is active.

Otherwise, the graph decomposes as shown on Figure 4.31 and we get a contribution of

$$\frac{x(s+1)y}{1-x(y+1)}x(z+1)\frac{\mathbf{J}(x,y,z,s)-1}{x(s+1)}$$

where the $\frac{\mathbf{J}(x,y,z,s)}{x(s+1)}$ term is due to the pink region which is precisely a smaller core, where a single disconnected cell has been removed. Hence, $\mathbf{J}(x,y,z,s,t)$ satisfies

$$\mathbf{J}(x,y,z,s) = 1 + x(s+1)\,\mathbf{J}(x,y,z,s) + \frac{xy(s+1)}{1-x(y+1)}x(z+1)\frac{\mathbf{J}(x,y,z,s)-1}{x(s+1)}$$

as claimed in the proposition. □

From Theorem 4.40 and Proposition 4.41, we get the generating function for the counting sequence of $\mathrm{Av}(r_u, 2143, 1 \oplus P)$.

Corollary 4.42. *The generating function of* $\mathrm{Av}(2134, 2413, 1 \oplus P)$ *is*

$$\mathbf{J}\left(x, \frac{x}{1-x}, \frac{B(x)-(1+x)}{x}, B(x)-1\right)$$

where $B(x)$ is the generating function of $\mathrm{Av}(2314, 2143, P)$.

Using the corollary above, we can compute $A(x)$, the generating function of $\mathrm{Av}(2314, 2143)$ that was first enumerated by [16]. In this example, P is empty. Hence, $B(x) = A(x)$. We get that the generating function $A(x)$ satisfies

$$A(x) = \mathbf{J}\left(x, \frac{x}{1-x}, \frac{A(x)-(1+x)}{x}, A(x)-1\right).$$

This equation can be solved explicitly to find the generating function

$$\frac{1-\sqrt{1-8x+16x^2-8x^3}}{4(x-x^2)}.$$

This generating function gives the counting sequence that appears in OEIS as A109033.

4.11 Unbalanced Wilf-equivalence

A combination of many of our results can be used to prove that some permutation classes have the same counting sequence. When two permutation classes have the same counting sequence, we say that they are *Wilf-equivalent*. This equivalence is said to be *unbalanced* [51] if one of the bases has a pattern of size k, but the other basis does not, like in the next theorem.

Theorem 4.43. *The permutation classes* $\mathrm{Av}(2413, 2134, 1234)$ *and* $\mathrm{Av}(2413, 2134, 1324, 12534)$ *are Wilf-equivalent.*

Proof. The first permutation class in our notation is $\mathrm{Av}(r_d, 2134, 1234)$. Hence, by Corollary 4.37, its generating function $A_1(x)$ is

$$\mathbf{H}\left(x, \frac{x}{1-x}, \frac{B_1(x)-(1+x)}{x}, C_1(x)-1, B_1(x)-1\right)$$

where

- $B_1(x)$ is the generating function of $\mathrm{Av}(r_d, 2134, 123) = \mathrm{Av}(r_d, 123)$.
- $C_1(x)$ is the generating function of $\mathrm{Av}(213, 12) = \mathrm{Av}(12)$.

The second permutation class is $\mathrm{Av}(r_d, 2134, 1324, 12534)$. Again, by Corollary 4.37, its generating function $A_2(x)$ is

$$\mathbf{H}\left(x, \frac{x}{1-x}, \frac{B_2(x)-(1+x)}{x}, C_2(x)-1, B_2(x)-1\right)$$

where

- $B_2(x)$ is the generating function of $\mathrm{Av}(r_d, 2134, 213, 1423) = \mathrm{Av}(213, 1423)$.

- $C_2(x)$ is the generating function of $\mathrm{Av}(213, 21, 1423) = \mathrm{Av}(21)$.

The permutation class $\mathrm{Av}(r_d, 123)$ is the same permutation class as $\mathrm{Av}(r_u, c_u, r_d, 123)$ since 123 is contained in r_u and c_u. Moreover, the last permutation class is a symmetry of $\mathrm{Av}(r_u, c_u, c_d, 123)$. Hence, by Corollary 4.25,

$$B_1(x) = \mathbf{G}\left(x, \frac{x}{1-x}, 1\right).$$

We also have that $\mathrm{Av}(213, 1423)$ is a symmetry of $\mathrm{Av}(132, c_u)$ which is the same permutation class as $\mathrm{Av}(r_d, c_d, c_u, 132)$. Hence, by Corollary 4.29,

$$B_2(x) = \mathbf{G}\left(x, \frac{x}{1-x}, \frac{\frac{x}{1-x}}{\frac{x}{1-x}}\right) = \mathbf{G}\left(x, \frac{x}{1-x}, 1\right).$$

We showed that $B_1(x) = B_2(x)$ and we know that $C_1(x) = C_2(x) = \frac{1}{1-x}$. Therefore, we have that $A_1(x) = A_2(x)$. □

Theorem 4.44. *The permutation classe* $\mathrm{Av}(2134, 2413)$ *is Wilf-equivalent to the permutation class* $\mathrm{Av}(2314, 3124, 13524, 12435)$.

Proof. In our notation, the two permutation classes are $\mathrm{Av}(r_d, 2134)$ and $\mathrm{Av}(r_u, c_u, 13524, 12435)$. Let $A_1(x)$ be the generating function of $\mathrm{Av}(2134, r_d)$ as computed in Section 4.9. Let $A_2(x)$ be the generating function of $\mathrm{Av}(r_u, c_u, 13524, 12435)$. By Corollary 4.15, $A_2(x) = \mathbf{F}(x, B(x) - 1)$ where $B(x)$ is the generating function of $\mathrm{Av}(r_u, c_u, r_d, 1324)$. As this permutation class is a symmetry of $\mathrm{Av}(r_u, c_u, c_d, 1324)$, by Corollary 4.25, $B(x) = \mathbf{G}(x, C(x) - 1)$ where $C(x)$ is the generating function for $\mathrm{Av}(r_u, c_u, r_d, 213)$. The previous permutation class is in fact $\mathrm{Av}(213)$. Hence, $C(x)$ is the generating function for the Catalan numbers. Rewinding the previous step, we can compute $A_2(x)$ explicitly. A simple verification then shows that $A_1(x) = A_2(x)$. □

4.12 Implementation

The techniques of enumeration studied in this chapter have been implemented in the python package *Permuta* [39]. To test if any of the theorems in this chapter apply to a basis one can use the code snippet below. It will print a reference to any of the results in this chapter that apply to any symmetry of the basis of interest.

```
from permuta import Perm
from permuta.enumeration_strategies import find_strategies

basis = [Perm((1,2,0,3)), Perm((2,0,1,3)), Perm((0,1,2,3))]
for strat in find_strategies(basis):
    print(strat.reference())
```

Chapter 5

Future directions

To conclude the thesis we present some open directions both for combinatorial exploration and the staircase encoding. We explore how we can improve the extraction of combinatorial specifications from the stable subset, detail how both combinatorial specifications and the staircase encoding could be used to sample at random from a permutation class and point towards ways that the staircase encoding method could be extended to enumerate more permutation classes.

5.1 Extraction of combinatorial specifications

When searching for a combinatorial specification using the method developed in [34], the prune method is used to obtain a set of rules that are guaranteed to be in a productive combinatorial specification. Algorithm 2 from [34] can then be used to quickly extract a random combinatorial specification. This is a very quick process that is mostly linear in the number of rules that end up in the specification. We can take advantage of this speed and sample many random specifications from a universe in order to find for example a small one.

When we use Algorithm 2 from Chapter 2 to compute the enumerable subset, we obtain the set of combinatorial sets that are in a combinatorial specification contained in U. To extract a productive combinatorial specification, we start with all the rules that contain only combinatorial sets in the enumerable subset. We then attempt to remove a rule, proceed to compute the enumerable subset again (using Algorithm 2) and look whether the combinatorial set of interest is still in the enumerable subset. If it is the case then the removed rule was not necessary and we repeat with the smaller subset of rules. If the combinatorial set is no longer in the enumerable subset we know this rule must be in the specification so we keep it and try reducing the set by removing another rule. The process ends when the rules left form a specification, *i.e.*, each combinatorial set is on the left-hand side of exactly one rule. This process basically requires us to compute the enumerable subset as many times as there are combinatorial rules. This can become really expensive as the enumerable subset can sometimes contain hundreds of thousands of combinatorial sets. Using some heuristics and optimizations, we are able to massively reduce the number of times we need to compute the enumerable subset which made this approach feasible in practice though still time-intensive. It would however be better to find an approach that does not rely on computing repeatedly the enumerable subset.

5.2 Random sampling

The language of combinatorial specifications opens the door to many existing tools, including, but certainly not limited, to generating the objects in a combinatorial set, or uniformly sampling objects at random in a combinatorial set (see [52], [53]).

Random sampling from forests Though the details are not published yet the ability to randomly sample and generate permutations is already implemented for combinatorial specifications made with productive rules. It is for example possible to sample uniformly at random from all permutation classes avoiding two size 4 patterns as well as from the three size 4 principal permutation classes for which we have a combinatorial specification that does not use reverse strategies. For more details and pictures of heat maps of those permutation classes see Section 5.3 of [34]. The approach is based on implementing random sampling for each strategy. We can then sample at random a parent of a rule assuming we can sample at random from the children. Consider the example of a disjoint-union type rule $\mathcal{A} = \mathcal{B} \sqcup \mathcal{C}$. If $|\mathcal{B}_n| = b_n$ and $|\mathcal{C}_n| = c_n$ then $a_n = b_n + c_n$. We can sample from $\mathcal{A}_n$ by sampling from either $\mathcal{B}_n$ or $\mathcal{C}_n$. To ensure uniformity we sample from $\mathcal{B}_n$ with probability $\frac{b_n}{a_n}$ and from $\mathcal{C}_n$ with probability $\frac{c_n}{a_n}$. It is however still unclear how one could sample from a reverse version of that rule where the parent would be $\mathcal{C}_n$. A similar problem arises with Cartesian product. Solving those issues would allow us to compute heat maps for the last three principal size 4 permutation classes for which the counting sequence is known.

Random sampling using the core structure It is also not too hard to convert the argument used in [42] to find $\mathbf{F}(x, y)$, the generating function for independent sets of the down-cores, to a specification. From Corollary 4.18, we know that $A(x)$, the generating function of $\text{Av}(r_d, c_d)$, satisfies $A(x) = \mathbf{F}(x, A(x) - 1)$. If we work with the q, t-analog of $A(x)$ where q tracks the number of left-to-right minima of the permutations and t tracks the number of active cells, that is

$$A(x, q, t) = \mathbf{F}(qx, tA(x) - t),$$

then by looking at this distribution we can pick randomly the number of minima, and active cells. We can then use the specification for $\mathbf{F}(x, y)$, and standard techniques to uniformly sample an independent set from the down-core of appropriate size. Finally, by recursively choosing the permutations to fill the corresponding active cells in the same manner, we will be able to sample uniformly from $\text{Av}(r_d, c_d)$.

This approach could be applied to the methods within Chapter 4, perhaps requiring some extra 'bookkeeping' along the way.

5.3 Extensions of the core method

Tracking more bases with the same cores Let π be a skew-indecomposable permutation. In Theorem 4.14, we described the structure of bases of the form $\{r_u, c_u, 1 \oplus \pi\}$. It seems possible to enumerate permutation classes whose bases are of the form $\{r_u, c_u, 21 \oplus \pi\}$. In this case, the staircase encoding would contain permutations avoiding $\{r_u, c_u, \pi\}$ in the cells that are not in the leading diagonal and permutations avoiding $\{r_u, c_u, 21 \oplus \pi\}$ in the cells in the leading diagonal. Hence, tracking the vertices of the independent set that are in the leading diagonal would be sufficient to enumerate this permutation class. It is likely that this reasoning can be extended to replace 21 with an arbitrary decreasing sequence.

Increasing the size of the patterns. In Section 4.3, we go from size 3 to size 4 patterns. To do so, we gave a set of patterns of size 4 that put the same constraints on the staircase grid as 123 did. This idea is not limited to size 4 patterns. We can easily see that the nine patterns of size 5 in Figure 5.1 enforce the same constraints as 123 on the staircase grid. Therefore, one can expect the results to generalize to greater size. However, to do so, two major issues need to be overcome. First, one needs to be sure that if a pattern occurs in a permutation, then there needs to be an occurrence of the pattern using the left-to-right minima of the permutation. This can be done with the addition of other patterns to the basis. Computation shows that for the size 5 cases, 7 patterns is the smallest number of patterns that can be added to do so. Secondly, the technique will not give information about the permutations with exactly two left-to-right minima.

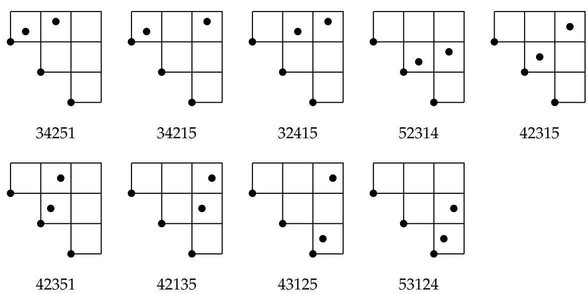


Figure 5.1: Patterns of size 5 that enforce the up-core constraint.

Independent sets on boundary grids. The letter π_i in a permutation π is a right-to-left maximum if $\pi_j < \pi_i$ of all $j > i$. Building a skew-shaped grid from the left-to-right minima and the right-to-left maxima we get the *boundary encoding* of the permutation. Figure 5.2 shows an example of a boundary encoding. One might be able to use the boundary encoding to generalize the method of the staircase encoding, but in this new case, any permutation avoiding 123 could potentially be a boundary.

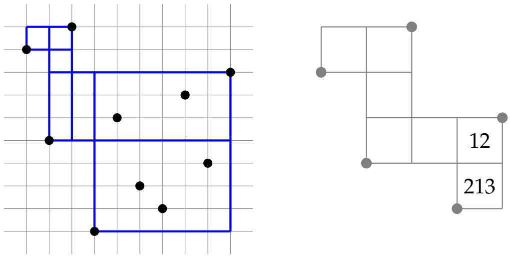


Figure 5.2: Boundary encoding of the permutation 95(10)1632748.

Wilf-equivalence and bijective proof. In Section 4.11, we uncovered two Wilf-equivalences by computing the generating functions with our results. The proof of Theorem 4.43 nicely highlights a structural argument for the Wilf-equivalence as both permutation classes are built from the same core. However, it is not the case in the proof of Theorem 4.44. It would be interesting to establish a bijection between the two permutation classes using the core structure.

Bibliography

[1] C. Bean, É. Nadeau, and H. Ulfarsson, "Enumeration of Permutation Classes and Weighted Labelled Independent Sets", *Discrete Mathematics & Theoretical Computer Science*, vol. 22 no. 2, Permutation Patterns 2019, Mar. 2021.

[2] P. Flajolet and R. Sedgewick, *Analytic combinatorics*. Cambridge University Press, 2009.

[3] R. P. Stanley, "What is enumerative combinatorics?", in *Enumerative combinatorics*, Springer, 1986, pp. 1–63.

[4] P. A. McMahon, *Combinatory analysis*. Cambridge UP, 1915.

[5] D. E. Knuth, *The art of computer programming*. Pearson Education, 1997.

[6] S. Miner and J. Pantone. (2018). Completing the structural analysis of the 2x4 permutation classes. arXiv: `1802.00483 [math.CO]`.

[7] D. Kremer, "Permutations with forbidden subsequences and a generalized Schröder number", *Discrete Math.*, vol. 218, no. 1-3, pp. 121–130, 2000.

[8] ——, "Postscript: "Permutations with forbidden subsequences and a generalized Schröder number" [Discrete Math. **218** (2000), no. 1-3, 121–130; MR1754331 (2001a:05005)]", *Discrete Math.*, vol. 270, no. 1-3, pp. 333–334, 2003.

[9] V. Vatter, "Finding regular insertion encodings for permutation classes", *J. Symbolic Comput.*, vol. 47, no. 3, pp. 259–265, 2012.

[10] M. H. Albert, M. D. Atkinson, and R. Brignall, "The enumeration of three pattern classes using monotone grid classes", *Electron. J. Combin.*, vol. 19, no. 3, Paper 20, 34, 2012.

[11] ——, "The enumeration of permutations avoiding 2143 and 4231", *Pure Math. Appl. (PU.M.A.)*, vol. 22, no. 2, pp. 87–98, 2011.

[12] M. H. Albert, M. D. Atkinson, and V. Vatter, "Counting 1324,4231-avoiding permutations", *Electron. J. Combin.*, pp. 1–9, 2009.

[13] M. D. Atkinson, "Permutations which are the union of an increasing and a decreasing subsequence", *Electron. J. Combin.*, vol. 5, Research paper 6, 13, 1998.

[14] M. D. Atkinson, B. E. Sagan, and V. Vatter, "Counting (3+1)-avoiding permutations", *European J. Combin.*, vol. 33, no. 1, pp. 49–61, 2012.

[15] S. Miner. (2016). Enumeration of several two-by-four classes. arXiv: `1610.01908` [math.CO].

[16] I. Le, "Wilf classes of pairs of permutations of length 4", *Electron. J. Combin.*, vol. 12, Research Paper 25, 26, 2005.

[17] D. Callan. (2013). The number of 1243, 2134-avoiding permutations. arXiv: `1303.3857` [math.CO].

[18] J. Pantone, "The enumeration of permutations avoiding 3124 and 4312", *Ann. Comb.*, vol. 21, no. 2, pp. 293–315, 2017.

[19] M. H. Albert, M. D. Atkinson, and V. Vatter, "Inflations of geometric grid classes: Three case studies", *Australas. J. Combin.*, vol. 58, pp. 24–47, 2014.

[20] M. Bóna, "The permutation classes equinumerous to the smooth class", *Electron. J. Combin.*, vol. 5, Research Paper 31, 12, 1998.

[21] D. Bevan, "The permutation classes $\mathrm{Av}(1234, 2341)$ and $\mathrm{Av}(1243, 2314)$", English, *Australas. J. Comb.*, vol. 64, pp. 3–20, 2016.

[22] ——, "The permutation class Av(4213,2143)", English, *Discrete Math. Theor. Comput. Sci.*, vol. 18, no. 2, p. 14, 2016.

[23] M. H. Albert, C. Homberger, J. Pantone, N. Shar, and V. Vatter, "Generating permutations with restricted containers", *J. Combin. Theory Ser. A*, vol. 157, pp. 205–232, 2018.

[24] D. Callan. (2013). Permutations avoiding 4321 and 3241 have an algebraic generating function. arXiv: `1306.3193` [math.CO].

[25] J. Bloom and V. Vatter, "Two vignettes on full rook placements", English, *Australas. J. Comb.*, vol. 64, pp. 77–87, 2016.

[26] I. M. Gessel, "Symmetric functions and P-recursiveness", *J. Combin. Theory Ser. A*, vol. 53, no. 2, pp. 257–285, 1990.

[27] M. Bóna, "Exact enumeration of 1342-avoiding permutations: A close link with labeled trees and planar maps", *J. Combin. Theory Ser. A*, vol. 80, no. 2, pp. 257–272, 1997.

[28] Z. Stankova, "Classification of forbidden subsequences of length 4", *European J. Combin.*, vol. 17, no. 5, pp. 501–517, 1996.

[29] F. R. K. Chung, R. L. Graham, V. E. Hoggatt Jr., and M. Kleiman, "The number of Baxter permutations", *J. Combin. Theory Ser. A*, vol. 24, no. 3, pp. 382–394, 1978.

[30] D. Zeilberger, "Enumeration schemes and, more importantly, their automatic generation", *Ann. Comb.*, vol. 2, no. 2, pp. 185–195, 1998.

[31] M. H. Albert, S. Linton, and N. Ruškuc, "The insertion encoding of permutations", *Electron. J. Combin.*, vol. 12, Research Paper 47, 31, 2005.

[32] M. H. Albert and M. D. Atkinson, "Simple permutations and pattern restricted permutations", *Discrete Math.*, vol. 300, no. 1-3, pp. 1–15, 2005.

[33] C. Bean, "Finding structure in permutation sets", PhD thesis, Reykjavík University, 2018.

[34] M. H. Albert, C. Bean, A. Claesson, É. Nadeau, J. Pantone, and H. Ulfarsson, *Combinatorial Exploration: An algorithmic framework for enumeration*, 2022. arXiv: `2202.07715 [math.CO]`.

[35] V. Vatter, "Enumeration schemes for restricted permutations", *Combin. Probab. Comput.*, vol. 17, no. 1, pp. 137–159, 2008.

[36] Y. Biers-Ariel, "Flexible Schemes for Pattern-Avoiding Permutations", *arXiv.org*, 2019. arXiv: `1905.07283 [math.CO]`.

[37] F. Bassino, M. Bouvel, A. Pierrot, C. Pivoteau, and D. Rossin, "An algorithm computing combinatorial specifications of permutation classes", *Discrete Appl. Math.*, vol. 224, pp. 16–44, 2017.

[38] F. Bassino, M. Bouvel, A. Pierrot, and D. Rossin, "An algorithm for deciding the finiteness of the number of simple permutations in permutation classes", *Adv. in Appl. Math.*, vol. 64, pp. 124–200, 2015.

[39] R. P. Ardal, A. B. Arnarson, C. Bean, A. B. Bjarnason, J. S. Eliasson, B. A. Gudmundsson, S. I. Jónsson, B. J. Kristinsson, É. Nadeau, J. Pantone, T. K. Shimomura-Magnusson, M. Tannock, and H. Ulfarsson, *Permutatriangle/permuta*, version v2.1.0, Jun. 2021.

[40] A. Claesson, "Counting segmented permutations using bicoloured Dyck paths", *Electron. J. Combin.*, vol. 12, Research Paper 39, 18, 2005.

[41] M. Albert, R. Ardal, A. B. Arnarson, C. Bean, A. Claesson, J. S. Elíasson, U. F. Erlendsson, B. J. Kristinsson, É. Nadeau, J. Pantone, J. Robb, T. K. Shimomura-Magnusson, and H. Ulfarsson, *Permutatriangle/comb_spec_searcher*, version v3.0.0, Jun. 2021.

[42] C. Bean, M. Tannock, and H. Ulfarsson, "Pattern avoiding permutations and independent sets in graphs", *J. Comb.*, vol. 11, no. 4, pp. 705–732, 2020.

[43] M. D. Atkinson and T. Stitt, "Restricted permutations and the wreath product", *Discrete Math.*, vol. 259, no. 1-3, pp. 19–36, 2002.

[44] D. Callan, T. Mansour, and M. Shattuck, "Wilf classification of triples of 4-letter patterns II", *Discrete Math. Theor. Comput. Sci.*, vol. 19, no. 1, Paper No. 6, 44, 2017.

[45] T. Mansour and M. Shattuck, "Nine classes of permutations enumerated by binomial transform of Fine's sequence", *Discrete Appl. Math.*, vol. 226, pp. 94–105, 2017.

[46] P. Brändén and A. Claesson, "Mesh patterns and the expansion of permutation statistics as sums of permutation patterns", *Electron. J. Combin.*, vol. 18, no. 2, Paper 5, 14, 2011.

[47] Í. Hilmarsson, I. Jónsdóttir, S. Sigurðardóttir, L. Viðarsdóttir, and H. Ulfarsson, "Wilf-classification of mesh patterns of short length", *Electron. J. Combin.*, vol. 22, no. 4, Paper 4.13, 27, 2015.

[48] OEIS, *OEIS Foundation Inc. (2021), The Online Encyclopedia of Integer Sequences*, `https://oeis.org/`, 2021.

[49] M. H. Albert and M. D. Atkinson, “Simple permutations and pattern restricted permutations”, *Discrete Math.*, vol. 300, no. 1-3, pp. 1–15, 2005.

[50] F. Bassino, M. Bouvel, A. Pierrot, C. Pivoteau, and D. Rossin, “An algorithm computing combinatorial specifications of permutation classes”, *Discrete Appl. Math.*, vol. 224, pp. 16–44, 2017.

[51] A. Burstein and J. Pantone, “Two examples of unbalanced Wilf-equivalence”, *J. Comb.*, vol. 6, no. 1-2, pp. 55–67, 2015.

[52] P. Flajolet, P. Zimmerman, and B. Van Cutsem, “A calculus for the random generation of labelled combinatorial structures”, *Theoret. Comput. Sci.*, vol. 132, no. 1-2, pp. 1–35, 1994.

[53] P. Duchon, P. Flajolet, G. Louchard, and G. Schaeffer, “Boltzmann samplers for the random generation of combinatorial structures”, *Combin. Probab. Comput.*, vol. 13, no. 4-5, pp. 577–625, 2004.

School of Technology, Department of Computer Science
Reykjavík University
Menntavegur 1
102 Reykjavík, Iceland
Tel. +354 599 6200
www.ru.is
ISBN 978-9935-9694-6-0 Print version
ISBN 978-9935-9694-7-7 Electronic version