UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft
Fakultät für Elektrotechnik, Informatik und Mathematik
Institut für Informatik
33098 Paderborn

On the complexity of computing Kronecker coefficients and deciding positivity of Littlewood-Richardson coefficients

Christian Ikenmeyer

Diplomarbeit

vorgelegt bei

Prof. Dr. Peter Bürgisser
Universität Paderborn

Prof. Dr. Friedrich Eisenbrand
École Polytechnique Fédérale de Lausanne

01. Oktober 2008

Abstract

Littlewood-Richardson coefficients are the multiplicities in the tensor product decomposition of two irreducible representations of the general linear group $\mathrm{GL}_n(\mathbb{C})$. Kronecker coefficients are the multiplicities in the tensor product decomposition of two irreducible representations of the symmetric group S_n. Both coefficients have a wide variety of interpretations in combinatorics, representation theory, geometry and in the theory of symmetric functions.

It is known that the problem of computing Littlewood-Richardson coefficients is hard. More specifically, it is #**P**-complete. This means that the existence of a polynomial time algorithm for this problem is equivalent to the existence of a polynomial time algorithm for evaluating permanents, which is considered unlikely. Our first result shows that the problem of computing Kronecker coefficients is computationally hard as well. More specifically, we prove that this problem is **GapP**-complete.

Quite surprisingly, as first pointed out by Mulmuley and Sohoni, it is possible to decide the positivity of Littlewood-Richardson coefficients in polynomial time. This follows by combining the facts that Knutson and Tao proved the Saturation Conjecture (1999) and that linear optimization is solvable in polynomial time. In the second part of this work, we design an explicit *combinatorial* polynomial time algorithm for deciding the positivity of Littlewood-Richardson coefficients. This algorithm is highly adapted to the problem and uses ideas from the theory of optimizing flows in networks. This algorithm also yields a proof of the Saturation Conjecture and a proof of a conjecture by Fulton, which was proved by Knutson, Tao and Woodward (2004). We further give a polynomial-time algorithm for deciding multiplicity freeness, i.e. whether a Littlewood-Richardson coefficient is exactly 1.

Zusammenfassung

Littlewood-Richardson-Koeffizienten sind die Multiplizitäten in der Tensorproduktzerlegung zweier irreduzibler Darstellungen der allgemeinen linearen Gruppe $\mathrm{GL}_n(\mathbb{C})$. Kronecker-Koeffizienten sind die Multiplizitäten in der Tensorproduktzerlegung zweier irreduzibler Darstellungen der symmetrischen Gruppe S_n. Beide Koeffizienten haben eine Vielzahl von Interpretationen in Kombinatorik, Darstellungstheorie, Geometrie und der Theorie symmetrischer Funktionen.

Es ist bekannt, dass das Problem der Berechnung von Littlewood-Richardson-Koeffizienten schwierig ist, genauer, dass es #**P**-vollständig ist. Dies bedeutet, dass die Existenz eines Polynomialzeitalgorithmus äquivalent ist zur Existenz eines Polynomialzeitalgorithmus zur Berechnung von Permanenten, was als unwahrscheinlich angesehen wird. Unser erstes Ergebnis zeigt, dass das Problem der Berechnung von Kronecker-Koeffizienten auch schwierig ist. Genauer gesagt beweisen wir die **GapP**-Vollständigkeit dieses Problems.

Überraschenderweise konnten Mulmuley und Sohoni aufzeigen, dass es möglich ist, die Positivität von Littlewood-Richardson-Koeffizienten in Polynomialzeit zu entscheiden. Dies ergibt sich aus der Kombination der beiden Tatsachen, dass Knutson und Tao die Saturiertheitsvermutung bewiesen haben (1999) und dass lineare Optimierung in Polynomialzeit lösbar ist. Im zweiten Teil dieser Arbeit konstruieren wir einen expliziten *kombinatorischen* Polynomialzeitalgorithmus, der die Positivität von Littlewood-Richardson-Koeffizienten entscheidet. Er ist stark an das Problem angepasst und benutzt Ideen von Flussoptimierungsalgorithmen. Dieser Algorithmus liefert auch einen Beweis für die Saturiertheitsvermutung und für eine Vermutung von Fulton, die erstmals von Knutson, Tao und Woodward (2004) bewiesen wurde. Außerdem geben wir einen Polynomialzeitalgorithmus zum Überprüfen der Freiheit von Multiplizitäten an, d.h. ob ein Littlewood-Richardson-Koeffizient genau 1 ist.

Eidesstattliche Erklärung Hiermit versichere ich, dass ich die folgende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen als Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Paderborn, den 01.10.2008

Christian Ikenmeyer

Lass aller Menschen Tun gedeihn,
ihr Werk von Dir behütet sein.
Sei jedem nah mit Deiner Kraft,
dass er getreu das Rechte schafft.

Psalteriolum harmonicum sacrarum cantilenarum,
anonymus, Köln 1642

Acknowledgments

I would like to express my gratitude to my teachers who invested much time in giving me valuable assistance during my research and with whom I had very fruitful discussions, namely Prof. Dr. Peter Bürgisser, whose dedication to my support was beyond comparison, and Prof. Dr. Friedrich Eisenbrand, whose suggestions on the last two chapters were of great help. I would also like to thank Martin Niemeier and Jaroslaw Klose for their proofreading and their helpful comments.

Contents

Chapter 1

Introduction

It is well known that the irreducible representations $\mathscr{S}_\lambda$ of the symmetric group S_n on n letters (in characteristic zero) can be indexed by the partitions $\lambda \vdash n$ of n, cf. [Sag01]. For given partitions $\lambda, \mu \vdash n$, the tensor product decomposes into $\mathscr{S}_\lambda \otimes \mathscr{S}_\mu = \bigoplus_{\nu \vdash n} g_{\lambda,\mu,\nu} \mathscr{S}_\nu$, where the multiplicity $g_{\lambda,\mu,\nu}$ is called the *Kronecker coefficient*. Related are the *Littlewood-Richardson coefficients* $c_{\lambda\mu}^{\nu}$. They describe the multiplicities in the tensor product decomposition of irreducible representations of the general linear group $\mathrm{GL}_n(\mathbb{C})$. These problems of computing multiplicities are special cases of *plethysm problems*.

Both coefficients have a wide variety of interpretations in combinatorics, representation theory, geometry, and in the theory of symmetric functions (cf. [Ful97]). However, our interest in the tensor product multiplicities stems from lower bound questions in computational complexity. Early work by Strassen [Str83] pointed out that a good understanding of the Kronecker coefficients could lead to complexity lower bounds for bilinear maps, notably matrix multiplication. The idea is to get information about the irreducible constituents of the vanishing ideal of secant varieties to Segre varieties, for recent results we refer to [LM04].

Kronecker coefficients as well as Littlewood-Richardson coefficients play a crucial role in the geometric complexity theory of Mulmuley and Sohoni (cf. [MS01, MS06]). This is an approach to arithmetic versions of the famous **P** vs. **NP** problem and related questions in computational complexity via geometric representation theory. What has been achieved so far is a series of reductions from orbit closure problems to subgroup restriction problems. The latter involve the problems of deciding in specific situations whether multiplicities $g_{\lambda,\mu,\nu}$ or $c_{\lambda\mu}^{\nu}$ are positive. However, until very recently, no efficient algorithms were known for the general problem of deciding the positivity of such multiplicities.

The well-known Littlewood-Richardson rule gives a combinatorial description of the numbers $c_{\lambda\mu}^{\nu}$ and also leads to algorithms for computing them. All of these algorithms take exponential time in the size of the input partitions (consisting of integers encoded in binary notation). However, quite surprisingly, the positivity of $c_{\lambda\mu}^{\nu}$ can be decided by a polynomial time algorithm! As pointed out by

Mulmuley and Sohoni (cf. [MS05]), this follows from the truth of the Saturation Conjecture, which was proved by Knutson and Tao (cf. [KT99]). On the other hand, Narayanan proved that the computation of $c^{\nu}_{\lambda\mu}$ is a #**P**-complete problem (cf. [Nar06]). Hence there does not exist a polynomial time algorithm for computing $c^{\nu}_{\lambda\mu}$ under the widely believed hypothesis $\mathbf{P} \neq \mathbf{NP}$.

Much less is known about the Kronecker coefficients $g_{\lambda,\mu,\nu}$. Lascoux, Remmel, Whitehead and Rosas (cf. [Las80], [Rem89, Rem92], [RW94] and [Ros01]) gave combinatorial interpretations of the Kronecker coefficients of partitions indexed by two row shapes or hook shapes. Very recently, Ballantine and Orellana managed to describe $g_{\lambda,\mu,\nu}$ in the case where $\mu = (n-p, p)$ has a two row shape and the diagram of λ is not contained inside the $2(p-1) \times 2(p-1)$ square (cf. [BO07]). Except for these special cases, a combinatorial interpretation of the numbers $g_{\lambda,\mu,\nu}$ is still lacking. The existence of such a description is stated as an outstanding open problem by Stanley (cf. [Sta00]).

This thesis has two main results: First we show that the problem of computing the Kronecker coefficients is **GapP**-complete (published in [BI08]), which implies that there does not exist a polynomial time algorithm for computing $g_{\lambda,\mu,\nu}$ under the hypothesis $\mathbf{P} \neq \mathbf{NP}$. As a second result we give a *combinatorial* polynomial-time algorithm for deciding the positivity of Littlewood-Richardson coefficients.

Structure of the thesis This work touches different mathematical areas, namely complexity theory, representation theory and the theory of flows in networks. For each one of these areas there is a preliminary chapter with definitions and facts from this area which are required for this work. Furthermore these chapters introduce notations that will be used in the course of this thesis.

This work presents two independent main results, the first of which is presented in chapter 4 and the second is covered in chapters 6 and 7.

In Chapter 4 we show how the characterization of Ballantine and Orellana can be used to prove that the problem KronCoeff of computing the Kronecker coefficient is **GapP**-complete. It implies that there does not exist a polynomial time algorithm for KronCoeff under the widely believed hypothesis $\mathbf{P} \neq \mathbf{NP}$. Note that we do not know whether KronCoeff is contained in the class #**P**. In fact, the latter would just express that $g_{\lambda,\mu,\nu}$ counts a number of appropriate combinatorial objects (and it can be decided in polynomial time whether a given object is appropriate), which in fact is a combinatorial description of the Kronecker coefficient.

In Chapter 6 and Chapter 7 we design an explicit *combinatorial* polynomial time algorithm for deciding the positivity of Littlewood-Richardson coefficients. This algorithm is highly adapted to the problem and uses ideas from the theory of optimizing flows in networks. It also yields a proof of the Saturation Conjecture. It was conjectured in [MS05] that such an algorithm exists. In the case of three strictly decreasing partitions the algorithm can further be used to check multiplicity freeness in polynomial time, i.e., whether a Littlewood-Richardson coefficient is exactly 1. In this case the analysis of this algorithm gives a direct proof of a conjecture by Fulton, namely that for all $N \in \mathbb{N}$ we have $c_{\lambda\mu}^{\nu} = 1 \Leftrightarrow c_{N\lambda N\mu}^{N\nu} = 1$. This was proved for arbitrary partitions by Knutson, Tao and Woodward (cf. [KTW04]).

In Chapter 6 we introduce the basic version of our algorithm called the LRPA (Littlewood-Richardson Positivity Algorithm), while in Chapter 7 we refine the LRPA with a capacity scaling approach to its polynomial-time counterpart LRP-CSA (Littlewood-Richardson Positivity Capacity Scaling Algorithm).

Chapter 2

Preliminaries – Complexity Theory

In this chapter we recall some definitions and facts from decision complexity theory and the lesser known counting complexity theory. A great introduction to complexity theory is given in [Pap94].

When considering alphabets, let $\Sigma := \{0, 1\}$. Of course all definitions and theorems work for any finite set. Any integers, rational numbers and matrices over the rationals can be encoded in Σ^*, which is the set of finite words over the alphabet Σ. Let $|w|$, $w \in \Sigma^*$ denote the length of the word w. We assume that the reader is familiar with the basic concepts of Turing machines and polynomial running time of algorithms. For details, we refer to [Pap94].

2.1 Decision complexity

Given a language $L \subseteq \Sigma^*$ and $x \in \Sigma^*$, the problem of deciding whether $x \in L$ is called the *decision problem associated with* L. We can identify languages with their decision problems.

Definition 2.1. **P** denotes the class of all languages $L \subseteq \Sigma^*$ that can be decided in polynomial time by a deterministic Turing machine. ■

Definition 2.2. **NP** denotes the class of all languages $L \subseteq \Sigma^*$ that can be decided in polynomial time by a nondeterministic Turing machine. ■

For $L \subseteq \Sigma^*$ we define the characteristic function of L as

$$\chi_L : \Sigma^* \to \{0,1\}, w \mapsto \begin{cases} 1 & \text{if} \quad w \in L \\ 0 & \text{if} \quad w \notin L \end{cases}.$$

Definition 2.3. L' reduces to L, if there is a function $\mathsf{pre} : \Sigma^* \to \Sigma^*$ computable in polynomial time with $\chi_{L'} = \chi_L \circ \mathsf{pre}$.

$L \subseteq \Sigma^*$ is denoted **NP**-*hard*, if each language $L' \in \mathbf{NP}$ reduces to L. If additionally $L \in \mathbf{NP}$, then L is called **NP**-*complete*. ∎

These reductions are often called *many-one reductions* in the literature.

Lemma 2.4. *There is an* **NP**-*complete language in* **P** *iff* $\mathbf{P} = \mathbf{NP}$.

Proof. If $\mathbf{P} = \mathbf{NP}$, then every language in **P** is **NP**-complete. The fact that **P** is nonempty proves the first direction.

It is clear that $\mathbf{P} \subseteq \mathbf{NP}$. Now let $L \in \mathbf{P}$ be **NP**-complete and $L' \in \mathbf{NP}$. Then there is a reduction $\mathsf{pre} : \Sigma^* \to \Sigma^*$ computable in polynomial time with $\chi_{L'} = \chi_L \circ \mathsf{pre}$. As $L \in \mathbf{P}$, χ_L can be computed in polynomial time. Then $\chi_{L'}$ can be computed in polynomial time as well which proves $L' \in \mathbf{P}$. Therefore $\mathbf{NP} \subseteq \mathbf{P}$ which proves the other direction. □

Polyhedra We now recall some important complexity theoretic results from discrete geometry.

Let $\mathbb{N} := \{0, 1, 2, \ldots\}$. Given a matrix $A \in \mathbb{Q}^{n \times m}$ and a vector $b \in \mathbb{Q}^n$, the points in $P(A, b) := \{x \in \mathbb{Q}^m \mid Ax \le b\}$ form a so-called *polyhedron*. Several algorithms exist for checking whether a polyhedron is empty. The ellipsoid method (see [Kha80, Sch98]) and interior point methods (see [Kar84]) are known to solve this problem in polynomial time. Thus we have

$$\mathsf{LP} := \left\{(A, b) \in \mathbb{Q}^{n \times m} \times \mathbb{Q}^n \mid n, m \in \mathbb{N}_{\ge 1}, P(A, b) \ne \emptyset\right\} \in \mathbf{P}$$

A related problem is to decide whether a polyhedron contains any integral points:

$$\mathsf{IP} := \left\{(A, b) \in \mathbb{Q}^{n \times m} \times \mathbb{Q}^n \mid n, m \in \mathbb{N}_{\ge 1}, P(A, b) \cap \mathbb{Z}^m \ne \emptyset\right\}$$

This problem is known to be **NP**-complete (see [Sch98, ch. 18]).

A matrix is called *totally unimodular*, if every square submatrix has determinant 1, -1 or 0. It is known that if A is totally unimodular and b is integral, then $P(A, b)$ has an integral point iff it is not empty and thus

$$\mathsf{IP} \cap \{(A, b) \in \mathbb{Z}^{n \times m} \times \mathbb{Z}^n \mid n, m \in \mathbb{N}_{\ge 1}, A \text{ is totally unimodular}\} \in \mathbf{P}$$

(see [Sch98]). As we will see, there exists a family of polyhedra - the *hive polyhedra* - where the matrix is not totally unimodular but nevertheless the polyhedron is empty iff it has no integral point. So for these polyhedra one can decide in polynomial time as well whether they contain an integral point.

2.2 Counting complexity

If one does not only ask *whether* an integral point in a polyhedron exists, but *how many* integral points exist, this problem lies in the complexity class #**P** as defined in [Val79]:

Definition 2.5. The complexity class $\#\mathbf{P}$ consists of the functions $f\colon \Sigma^* \to \mathbb{N}$ for which there exists a nondeterministic polynomial-time Turing machine M such that for all $w \in \Sigma^*$ we have

$$f(w) = \text{ the number of accepting paths of } M, \text{ when started with input } w.$$

■

For a counting problem $f : \Sigma^* \to \mathbb{N}$, we define the associated decision problem $f_{>0}$ as the following: $f_{>0} = \{w \in \Sigma^* \mid f(w) > 0\}$.

Note that

$$\#\mathsf{IP} := \left\{\mathbb{Q}^{n\times m} \times \mathbb{Q}^n \ni (A, b) \mapsto |\{x \in \mathbb{Z}^m | Ax \leq b\}| \;\middle|\; n, m \in \mathbb{N}_{\geq 1}\right\} \in \#\mathbf{P},$$

because $\mathsf{IP} \in \mathbf{NP}$.

$\#\mathbf{P}$ is closed under addition ($f, g \in \#\mathbf{P} \Rightarrow \mathbf{f} + \mathbf{g} \in \#\mathbf{P}$) and multiplication ($f, g \in \#\mathbf{P} \Rightarrow \mathbf{fg} \in \#\mathbf{P}$). $\#\mathbf{P}$ is also closed under exponential summation in the following sense (cf. [For97]):

Proposition 2.6. *Let $f : \Sigma^* \to \mathbb{N}$ be in $\#\mathbf{P}$, p be a polynomial. Then the function*

$$\Sigma^* \to \mathbb{N}, x \mapsto \sum_{\substack{y\in\Sigma^* \\ |y|\leq p(|x|)}} f(x||y)$$

is in $\#\mathbf{P}$ as well, where $||$ represents the concatenation of words.

$\#\mathbf{P}$ is not closed under subtraction, as $\#\mathbf{P}$ only contains functions that map to $\mathbb{N}$. It is unknown whether $\#\mathbf{P}$ is closed under "safe subtraction" ($f, g \in \#\mathbf{P} \overset{?}{\Rightarrow} \mathbf{x} \mapsto \max\{\mathbf{f}(\mathbf{x}) - \mathbf{g}(\mathbf{x}), \mathbf{0}\} \in \#\mathbf{P}$), but there are some unlikely consequences stated in [OH91], if this were true. To get a class that is closed under subtraction, [FFK91] introduced the following:

Definition 2.7. **GapP** is the class of functions $f : \Sigma^* \to \mathbb{Z}$ where $f = g - h$ with $g, h \in \#\mathbf{P}$. Hence **GapP** is the closure of $\#\mathbf{P}$ under subtraction. ■

[FFK91] showed that $\mathbf{GapP} := \#\mathbf{P} - \#\mathbf{P} = \#\mathbf{P} - \mathbf{FP} = \mathbf{FP} - \#\mathbf{P}$, where the difference of complexity classes is defined via the pointwise function difference and $\mathbf{FP}$ is the class of functions $f : \Sigma^* \to \mathbb{Z}$ that can be computed in polynomial time.

We now describe the definition of reductions and completeness for counting complexity classes.

Definition 2.8. Let $\mathscr{C}$ be a class of functions $\Sigma^* \to \mathbb{Z}$, e.g. $\mathscr{C} = \#\mathbf{P}$ or $\mathscr{C} = \mathbf{GapP}$. We say that $g \in \mathscr{C}$ *reduces to* $f \in \mathscr{C}$, if the following holds: There are functions $\mathsf{pre} : \Sigma^* \to \Sigma^*$ and $\mathsf{post} : \mathbb{Z} \to \mathbb{Z}$, both computable in polynomial time, such that $\mathsf{post} \circ f \circ \mathsf{pre} = g$. If $\mathsf{post} = \mathrm{id}$, we call the reduction *parsimonious*. ■

Definition 2.9. f is denoted *$\mathscr{C}$-hard [under parsimonious reductions]*, if each $g \in \mathscr{C}$ reduces to f [with parsimonious reductions].

f is denoted *$\mathscr{C}$-complete [under parsimonious reductions]*, if it is $\mathscr{C}$-hard [under parsimonious reductions] and additionally $f \in \mathscr{C}$. ■

For example the problem #SAT of counting the satisfying truth assignments of a boolean formula is #**P**-complete under parsimonious reductions (see [Pap94]) and the problem GapSAT of computing the difference between the number of satisfying truth assignments of two boolean formulae is **GapP**-complete under parsimonious reductions.

We now proceed with a few simple observations that will help us classifying the hardness of computing Littlewood-Richardson and Kronecker coefficients.

Lemma 2.10. *Let $\mathscr{C} = \#\mathbf{P}$ or $\mathscr{C} = \mathbf{GapP}$. Let f be $\mathscr{C}$-hard under parsimonious reductions and let $f_{>0}$ be the associated decision problem. Then $f_{>0}$ is* **NP***-hard.*

Proof. Let SAT $:= \#\mathrm{SAT}_{>0}$. The well-known Cook-Levin theorem states that SAT is **NP**-complete. Let f be $\mathscr{C}$-hard under parsimonious reductions and $f_{>0}$ be the associated decision problem. Let $(\mathsf{pre}, \mathrm{id})$ be the parsimonious reduction from #SAT to f, i.e. $f \circ \mathsf{pre} = \#\mathrm{SAT}$. Fix any $w \in \Sigma^*$.
Now $\chi_{\mathrm{SAT}}(w) = 1 \Leftrightarrow \#\mathrm{SAT}(w) \geq 1 \Leftrightarrow (f \circ \mathsf{pre})(w) \geq 1 \Leftrightarrow \chi_{f_{>0}}\big(\mathsf{pre}(w)\big) = 1$.
Therefore $\chi_{\mathrm{SAT}} = \chi_{f_{>0}} \circ \mathsf{pre}$. Thus pre serves as a reduction from SAT to $f_{>0}$. Moreover, $f_{>0}$ is **NP**-hard. □

Corollary 2.11. *Let $\mathscr{C} = \#\mathbf{P}$ or $\mathscr{C} = \mathbf{GapP}$. If $f_{>0} \in \mathbf{P}$ and assuming $\mathbf{P} \neq \mathbf{NP}$, then f is not $\mathscr{C}$-hard under parsimonious reductions.*

Proof. We combine Lemma 2.4 and Lemma 2.10. □

Proposition 2.12. *f is #**P**-hard iff f is **GapP**-hard.*

Note that this is false under parsimonious reductions, as there is no parsimonious reduction from the function $(x \mapsto -1) \in \mathbf{GapP}$ to any function in #**P**.

Proof. As $\#\mathbf{P} \subseteq \mathbf{GapP}$, each **GapP**-hard function is obviously #**P**-hard. Now let f be #**P**-hard, $g_1 - g_2 = g \in \mathbf{GapP}$ with $g_1, g_2 \in \#\mathbf{P}$. As g_1 and g_2 count accepting paths of nondeterministic polynomial-time Turing machines, there exists $k \in \mathbb{N}_{\geq 1}$ such that for all $w \in \Sigma^*$ we have $g_1(w) \leq 2^{|w|^k} < 2^{|w|^k+1}$ and $g_2(w) < 2^{|w|^k+1}$. So we define

$$B : \Sigma^* \to \mathbb{N}, w \mapsto 2^{|w|^k+1}, \quad C : \Sigma^* \to \mathbb{N}, w \mapsto 2^{2|w|^k+2}.$$

We have $B, C \in \mathbf{FP} \subseteq \#\mathbf{P}$. From the closure properties of #**P** it follows that $C + Bg_1 + g_2 \in \#\mathbf{P}$. As f is #**P**-hard, there is a reduction $(\mathsf{post}, \mathsf{pre})$ with $\mathsf{post} \circ f \circ \mathsf{pre} = C + Bg_1 + g_2$. Consider the following function

$$b : \mathbb{N} \to \mathbb{N}, x \mapsto 2^{\left\lfloor \frac{\lfloor \log x \rfloor}{2} \right\rfloor}.$$

If given as input a natural number x that has an odd number n of bits in its bitstring representation, then $b(x) = \frac{n-1}{2}$. Note that $(C + Bg_1 + g_2)(w)$ has an odd number of bits in its bitstring representation for all $w \in \Sigma^*$. Also note that $b(x)$ can be computed in polynomial time, because $\lfloor \log x \rfloor$ can be determined directly from the bitstring representation of x. Now we define

$$\text{decode} : \mathbb{N} \to \mathbb{Z}, x \mapsto (x \text{ div } b(x)) \text{ mod } b(x) - x \text{ mod } b(x),$$

where div and mod basically only cut the bitstring of x, because $b(x)$ is a power of 2. Then $(\text{decode} \circ \mathsf{post}) \circ f \circ \mathsf{pre} = g_1 - g_2 = g$, which proves that g reduces to f. Therefore f is **GapP**-hard. □

Chapter 3

Preliminaries – Representation Theory

In this chapter we describe definitions and facts about representations of the symmetric group S_n and the general linear group $\mathrm{GL}_n(\mathbb{C})$ and about their correspondence to symmetric functions. We will explain where Littlewood-Richardson coefficients and Kronecker coefficients appear in these contexts. See [Sag01, Ful97, FH91, Sta99] for proofs, details and further reading.

3.1 Skew diagrams and tableaux

A *Young diagram* is a collection of boxes, arranged in left justified rows, such that from top to bottom, the number of boxes in a row is monotonically weakly decreasing. For $\lambda := (\lambda_1, \ldots, \lambda_s) \subseteq \mathbb{N}^s$ we define its length as $\ell(\lambda) := \max(\{0\} \cup \{i \mid \lambda_i > 0\})$ and its size as $|\lambda| := \sum_{i=1}^{\ell(\lambda)} \lambda_i$. Moreover we set $\lambda_r := 0$ for all $r > s$. If the λ_i are monotonically weakly decreasing and $|\lambda| = n$, then we call λ a *partition* of n and write $\lambda \vdash n$. In this case, λ specifies a Young diagram consisting of n boxes with λ_i boxes in the ith row for all i (see Figure 3.1(a)). If we know that $m \geq \ell(\lambda)$, we can additionally write $\lambda \vdash_m n$, which means that

(a) The Young diagram of the partition $\lambda = (4,4,2,1,1)$, $\lambda \vdash 12$, $\ell(\lambda) = 5$.

(b) The Young diagram of the conjugate partition $\lambda' = (5,3,2,2)$, $\lambda' \vdash 12$, $\ell(\lambda') = 4$.

(c) A skew diagram with shape $(4,4,2,1,1)/(3,3,2)$.

(d) The same skew diagram also has shape $(4,4,3,1,1)/(3,3,3)$.

Figure 3.1: Young diagrams and skew diagrams.

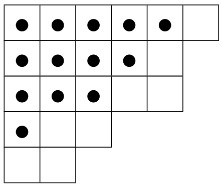

Figure 3.2: The skew diagram of the product $(3, 2)/(1) * (3, 2, 2)/(2, 1)$.

•	•	1	2	3
•	2	2		
2	3	3		
4				

Figure 3.3: A semistandard skew tableau of shape $(5, 3, 3, 1)/(2, 1)$ and type $(1, 4, 3, 1)$. The reverse reading word is $(3, 2, 1, 2, 2, 3, 3, 2, 4)$.

the Young diagram corresponding to λ has at most m rows. To any partition λ there corresponds its *conjugate partition* λ' which is obtained by transposing the Young diagram of λ, that is, reflecting it at the main diagonal (see Figure 3.1(b)). We note that by definition every row in λ corresponds to a column in λ' and vice versa. Moreover, $|\lambda| = |\lambda'|$.

A *skew diagram* is the set of boxes obtained by removing a smaller Young diagram from a larger one (see Figure 3.1(c), removed boxes are marked with dots). If we remove $\alpha \subseteq \lambda$ from λ, then we denote the resulting skew diagram by λ/α and say that it has the *shape* λ/α. Note that for a given skew diagram λ/α, the partitions α and λ are not necessarily uniquely defined (see Figure 3.1(d)). For example, we have $(4, 4, 2, 1, 1)/(3, 3, 2) = (4, 4, 3, 1, 1)/(3, 3, 3)$. Every Young diagram is a skew diagram, as one can choose α to be the empty set of boxes. The *product* $\lambda/\alpha * \tilde{\lambda}/\tilde{\alpha}$ of two skew diagrams λ/α and $\tilde{\lambda}/\tilde{\alpha}$ is defined to be the skew diagram obtained by attaching the upper right corner of λ to the lower left corner of $\tilde{\lambda}$ (see Figure 3.2). A similar definition applies for more than one factor.

A *filling* of a skew diagram λ/α is a numbering of its boxes with (not necessarily distinct) positive integers. A *semistandard skew tableau* T of shape λ/α is defined to be a filling of λ/α such that the entries are weakly increasing from left to right across each row and strictly increasing from top to bottom down each column. If T houses μ_j copies of j, then the tableau T is said to have the *type* $\mu := (\mu_1, \mu_2, \ldots)$ (see Figure 3.3). Note that $|\lambda| - |\alpha| = |\mu|$, but in contrast to λ and α, μ need not be weakly decreasing. A *semistandard Young tableau* of shape λ is defined to be a semistandard skew tableau of shape λ/α, where $\alpha = ()$ is the empty partition. The number of semistandard Young tableaux of shape λ

and type μ is called the *Kostka number* $\mathbf{K}_{\lambda\mu}$. The number of semistandard skew tableaux of shape λ/α and type μ is called the *skew Kostka number* $\mathbf{K}_{\lambda/\alpha;\mu}$.

The *reverse reading word* $w^{\leftarrow}(T)$ of a skew tableau T is the sequence of entries in T obtained by reading the entries from right to left and top to bottom, starting with the first row (see Figure 3.3). The *type* of a word $w \in \mathbb{N}^*_{>0}$ is the type of any tableau T with $w^{\leftarrow}(T) = w$. A *lattice permutation* is a sequence $(a_1, a_2, \cdots, a_n)$ such that in any prefix segment $(a_1, a_2, \ldots, a_p)$, $0 \le p \le n$ the number of i's is at least as large as the number of $(i+1)$'s for all i. For example the word $(3, 2, 1, 2, 2, 3, 3, 2, 4)$ is not a lattice permutation, but the word $(1, 1, 1, 2, 2, 3, 3, 2, 4)$ is a lattice permutation.

3.2 The algebra of symmetric functions

For $m \in \mathbb{N}$, a polynomial $f \in \mathbb{C}[X_1, \ldots, X_m]$ is denoted *symmetric*, if it is invariant under permutation of its variables. For example, let $m = 3$, then

$$f = X_1^2X_2 + X_1^2X_3 + X_2^2X_1 + X_2^2X_3 + X_3^2X_1 + X_3^2X_2 + X_1 + X_2 + X_3$$

is a symmetric polynomial. A *homogeneous polynomial* is a polynomial whose monomials all have the same degree. Let

$$\Lambda_m^n := \{f \in \mathbb{C}[X_1, \ldots, X_m] \mid f \text{ symmetric and homogeneous of degree } n\}$$

denote the vector space of homogeneous symmetric polynomials of degree n in m variables. Then $\Lambda_m := \bigoplus_{n\in\mathbb{N}} \Lambda_m^n$ becomes a graded commutative algebra with the ordinary multiplication of polynomials.

Definition 3.1. Given $\lambda \vdash_m n$, the Schur polynomial s_λ corresponding to λ is defined as

$$s_\lambda := \sum_{\mu\in\mathbb{N}^m, |\mu|=n} \mathbf{K}_{\lambda\mu} X^\mu \in \mathbb{Z}[X_1, \ldots, X_m],$$

where $X^\mu := X_1^{\mu_1} X_2^{\mu_2} \cdots X_m^{\mu_m}$ ■

It is remarkable that s_λ is symmetric and therefore contained in Λ_m^n. It is further known that $(s_\lambda)_{\lambda\vdash_m n}$ form a $\mathbb{Z}$-basis of Λ_m^n. For most purposes it does not matter how many variables are used, as long as the number of variables is not smaller than the degree of the polynomial, because the projection

$$\Lambda_m^n \to \Lambda_n^n, X_j \mapsto \begin{cases} X_j & \text{if } 1 \le j \le n \\ 0 & \text{otherwise} \end{cases}$$

is an isomorphism for $m \ge n$. Via the inverse of this isomorphism we can map any $f \in \Lambda_n^n$ to $f\!\uparrow^m \in \Lambda_m^n$ as long as $m \ge n$. We define for all $n \in \mathbb{N}_{\ge 1}$:

$\Lambda^n := \Lambda^n_n$, $\Lambda^0 := \mathbb{C}$ and make $\Lambda := \bigoplus_{n\in\mathbb{N}} \Lambda^n$ a graded $\mathbb{C}$-algebra with the following multiplication: Let $f \in \Lambda^n, g \in \Lambda^m$. Then

$$f \cdot g := f\!\uparrow^{n+m} \cdot g\!\uparrow^{n+m} \in \Lambda^{n+m},$$

where the multiplication on the right is the ordinary multiplication in Λ_{n+m}. The s_λ form a basis of Λ, where λ goes over all partitions. Λ is called the *algebra of symmetric functions*.

3.3 The algebra of characters of the symmetric group

A *representation of a group G* is a $\mathbb{C}$-vector space V with a group homomorphism $D : G \to \mathrm{GL}(V)$ from the group G into the general linear group $\mathrm{GL}(V)$ of V where $\dim(V)$ is called the *degree of the representation*. For the sake of simplicity we only consider finite dimensional vector spaces over $\mathbb{C}$. A subspace W of V that is fixed under $D(g)$ for all $g \in G$ is called a *subrepresentation of V*. If V has exactly two subrepresentations, namely the zero-dimensional subspace and V itself, then the representation is called *irreducible*, otherwise it is called *reducible*. Two representations (V_1, D_1) and (V_2, D_2) of G are *isomorphic*, if there exists a vector space isomorphism $\alpha : V_1 \to V_2$ with $\forall g \in G : \alpha \circ D_1(g) \circ \alpha^{-1} = D_2(g)$.

It is well known that there are only finitely many isomorphism classes of irreducible representations of S_n. An explicit list of representatives $\mathscr{S}_\lambda$ called the *Specht modules* can be indexed by the partitions $\lambda \vdash n$ in a natural way (cf. [Sag01]).

A representation $D : \mathrm{GL}_m(\mathbb{C}) \to \mathrm{GL}(V)$ of the general linear group for a finite dimensional vector space V is called *polynomial*, if after choosing bases $\mathrm{GL}_m(\mathbb{C}) \subset \mathbb{C}^{m^2}$ and $\mathrm{GL}(V) \subset \mathbb{C}^{N^2}$ we have that the N^2 coordinate functions of D are polynomial functions of the m^2 variables. An explicit list of representatives E_λ of polynomial irreducible representations of $\mathrm{GL}_m(\mathbb{C})$ called the *Schur modules* can be indexed in a natural way by the partitions $\lambda \vdash_m$ that have at most m rows (cf. [Ful97]). The degree of E_λ is given by $|\lambda|$.

Characters Let G be finite. After choosing bases, the values of D can be interpreted as invertible matrices over $\mathbb{C}$. By taking their trace, one obtains a map $\chi_D : G \to \mathbb{C}$, $g \mapsto \mathrm{tr}(D(g))$ which is called the *character of the representation D*. It is well-defined, because the trace of a matrix is invariant under basis transformations. A fundamental theorem states that two representations are isomorphic iff they have the same character. Characters are always class functions, i.e. $\forall g, h \in G : \chi_D(g) = \chi_D(hgh^{-1})$. Let $R(G)$ denote the $\mathbb{C}$-vector space of class functions $G \to \mathbb{C}$. The characters of the irreducible representations of G form a basis of $R(G)$.

The symmetric group Given a subgroup $H \leq S_n$ and $\varphi \in R(H)$, we define the induced function $\varphi\uparrow_H^{S_n} \in R(S_n)$ as

$$\varphi\uparrow_H^{S_n}(g) := \frac{1}{|H|} \sum_{x \in S_n} \varphi(x^{-1}gx),$$

where $\varphi(x^{-1}gx) := 0$ for $x^{-1}gx \notin H$. Now set $R_0 := \mathbb{C}$ and $R_n := R(S_n)$ for all $n \in \mathbb{N}_{\geq 1}$. Then $R := \bigoplus_{n \in \mathbb{N}} R_n$ becomes a graded commutative $\mathbb{C}$-algebra by defining a multiplication as follows:

$$\varphi \cdot \psi := (\varphi \times \psi)\uparrow_{S_m \times S_n}^{S_{n+m}}$$

where $(\varphi \times \psi) : S_m \times S_n \to \mathbb{C}, (\pi, \sigma) \mapsto \varphi(\pi)\psi(\sigma)$ for $\varphi \in R_m, \psi \in R_n$. Let $\chi^\lambda := \chi_{\mathscr{S}_\lambda}$. The χ^λ form a basis of R, where λ goes over all partitions. R is called the *algebra of characters of the symmetric group.*

Characters of $\mathrm{GL}_m(\mathbb{C})$ Given a polynomial representation $D : \mathrm{GL}_m(\mathbb{C}) \to \mathrm{GL}(V)$, then after choosing bases we define the *character* χ_D *of the representation* D as

$$\chi_D : \mathbb{C}^m \to \mathbb{C}[x_1, \ldots, x_m],\ (x_1, \ldots, x_m) \mapsto \mathrm{tr}\big(D\left(\mathrm{diag}(x_1, \ldots, x_m)\right)\big).$$

The map is well-defined and satisfies $\chi_{E_\lambda}(x_1, \ldots, x_m) = s_\lambda(x_1, \ldots, x_m)$ for all partitions $\lambda \vdash_m$. According to a fundamental theorem we have that two representations are isomorphic iff they have the same character.

3.4 Coefficients in decompositions

We define a linear map $\mathrm{ch} : R \to \Lambda$, $\mathrm{ch}(\chi^\lambda) = s_\lambda$. This is known to be an isomorphism of graded $\mathbb{C}$-algebras. Therefore

$$s_\lambda \cdot s_\mu = \mathrm{ch}(\chi^\lambda \cdot \chi^\mu).$$

Let $*$ be the pointwise product of class functions in R_n. Then ch induces a commutative and associative product on Λ^n by

$$s_\lambda * s_\mu := \mathrm{ch}(\chi^\lambda * \chi^\mu)$$

which is called the *inner product of Schur functions.* We are interested in how these two different products decompose. It is known that in either case the decomposition of a product of two basis elements decomposes into a linear combination of basis elements that has only nonnegative integral coefficients. This gives rise to the following definitions:

Definition 3.2 (Kronecker coefficient)**.** Let $\lambda, \mu \vdash n$,

$$\chi^\lambda * \chi^\mu = \sum_{\nu \vdash n} g_{\lambda,\mu,\nu} \chi^\nu.$$

Then $g_{\lambda,\mu,\nu}$ is denoted the *Kronecker coefficient* of λ, μ and ν.

The problem of computing $g_{\lambda,\mu,\nu}$ for given $\lambda, \mu, \nu \vdash n$ is denoted by KronCoeff with its associated decision problem KronCoeff$_{>0}$. ■

Definition 3.3 (Littlewood-Richardson coefficient)**.** Let $\lambda \vdash m, \mu \vdash n$,

$$s_\lambda \cdot s_\mu = \sum_{\nu \vdash n+m} c^\nu_{\lambda\mu} s_\nu.$$

Then $c^\nu_{\lambda\mu}$ is denoted the *Littlewood-Richardson coefficient* of λ, μ and ν.

The problem of computing $c^\nu_{\lambda\mu}$ for given $\lambda \vdash m, \mu \vdash n, \nu \vdash m+n$ is denoted by LRCoeff with its associated decision problem LRCoeff$_{>0}$. ■

It is well-known that $c^\nu_{\lambda\mu}$ equals the number of semistandard skew tableaux of shape ν/λ and type μ whose reverse reading word is a lattice permutation. For the Kronecker coefficients such a combinatorial description is only known in some special cases.

Symmetries It is clear from the definition that $g_{\lambda,\mu,\nu} = g_{\mu,\lambda,\nu}$ and $c^\nu_{\lambda\mu} = c^\nu_{\mu\lambda}$, because both products are commutative. It is further known that

$$g_{\lambda,\mu,\nu} = \frac{1}{n!} \sum_{g \in S_n} \chi^\lambda(g)\chi^\mu(g)\chi^\nu(g) \tag{3.1}$$

and thus that $g_{\lambda,\mu,\nu}$ is symmetric in λ, μ and ν. Additionally we have $g_{\lambda,\mu,\nu} = g_{\lambda,\mu',\nu'}$ and $c^\nu_{\lambda\mu} = c^{\nu'}_{\lambda'\mu'}$.

Tensor products of representations Maschke's theorem states that representations of finite groups can be decomposed into direct sums of irreducible subrepresentations. This decomposition is unique except for order and isomorphism of its constituents. This is true also for the general linear group $\mathrm{GL}_m(\mathbb{C})$. The number of summands in such a decomposition of a representation V that are isomorphic to a representation W is called the *multiplicity* of W in V. Given two representations $D_1 : G \to \mathrm{GL}(V_1)$ and $D_2 : G \to \mathrm{GL}(V_2)$, then the tensor product $D_1 \otimes D_2 : G \to \mathrm{GL}(V_1 \otimes V_2)$, $g \mapsto D_1(g) \otimes D_2(g)$ is again a representation with character $\chi_{D_1 \otimes D_2}(g) = \chi_{D_1}(g) \cdot \chi_{D_2}(g)$. Littlewood-Richardson coefficients and Kronecker coefficients can be interpreted as multiplicities in decompositions of tensor products as well:

For $\lambda, \mu \vdash n$ we have

$$\mathscr{S}_\lambda \otimes \mathscr{S}_\mu = \bigoplus_{\nu \vdash n} g_{\lambda,\mu,\nu} \mathscr{S}_\nu,$$

which directly follows from Definition 3.2 and for partitions $\lambda \vdash_m, \mu \vdash_m$ we have

$$E_\lambda \otimes E_\mu = \bigoplus_{\nu \vdash |\lambda|+|\mu|} c^\nu_{\lambda\mu} E_\nu,$$

which directly follows from Definition 3.3 and the fact that $\chi_{E_\lambda} = s_\lambda$ for all partitions $\lambda \vdash_m$.

From the interpretation of Littlewood-Richardson coefficients and Kronecker coefficients as multiplicities in tensor product decompositions, we know that both coefficients are always nonnegative integers. According to [BK99], $\mathscr{S}_\lambda \otimes \mathscr{S}_\mu$ is irreducible only in the case of $\mathscr{S}_\lambda$ or $\mathscr{S}_\mu$ being of degree 1.

Chapter 4

The complexity of computing Kronecker coefficients

In this chapter we prove the **GapP**-completeness of computing Kronecker coefficients:

Theorem 4.1. *The problem* KronCoeff *of computing Kronecker coefficients is* **GapP**-*complete.*

We proceed in two steps, first proving in Section 4.1 that the problem is contained in **GapP** and then proving in Section 4.4 that it is **GapP**-hard, which is equivalent by Proposition 2.12 to being #**P**-hard. In contrast to the Littlewood-Richardson coefficients, it is unknown whether the Kronecker coefficient $g_{\lambda,\mu,\nu}$ counts a number of appropriate combinatorial objects. Therefore it is unknown whether KronCoeff $\in$ #**P**. It is further unknown whether KronCoeff$_{>0} \in$ **P** or not.

4.1 Upper bound for KronCoeff

Bürgisser and the author (cf. [BI08]) use a formula of Garsia and Remmel (cf. [Sta99, Ex. 7.84, p. 478]) and the Littlewood-Richardson rule to show the following proposition:

Proposition 4.2. KronCoeff $\in$ **GapP**

Proof. The proof will use ideas and formulas from the literature (cp. [Sta99, Chap. 7]). We fix $n \in \mathbb{N}$, $m \in \mathbb{N}$, $m \geq n$. Let h_k denote the *kth complete symmetric function*:

$$h_k := \sum_{\substack{\mu \in \mathbb{N}^m \\ |\mu| = k}} X_1^{\mu_1} X_2^{\mu_2} \cdots X_m^{\mu_m}.$$

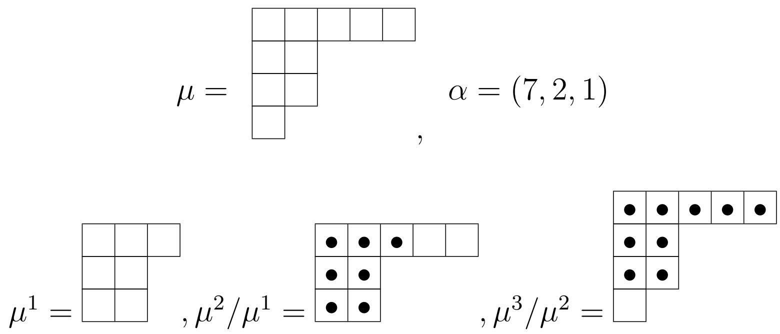

Figure 4.1: A decomposition of shape μ and type α.

For a partition α, we set

$$h_\alpha := h_{\alpha_1} h_{\alpha_2} \cdots h_{\alpha_{\ell(\alpha)}}.$$

The Jacobi-Trudi identity expresses the Schur polynomial s_λ, $\lambda \vdash n$ as the following determinant of a structured matrix, whose entries are the complete symmetric functions:

$$\begin{aligned} s_\lambda &= \det(h_{\lambda_i - i + j})_{1 \le i,j \le n} = \sum_{\pi \in S_n} \prod_{i=1}^{n} h_{\lambda_i - i + \pi(i)} \\ &= \sum_{\pi \in A_n} \prod_{i=1}^{n} h_{\lambda_i - i + \pi(i)} - \sum_{\pi \in S_n \setminus A_n} \prod_{i=1}^{n} h_{\lambda_i - i + \pi(i)} \\ &=: \sum_{\alpha \vdash n} N^+_{\alpha\lambda} h_\alpha - \sum_{\alpha \vdash n} N^-_{\alpha\lambda} h_\alpha. \end{aligned} \tag{4.1}$$

Here, $N^+_{\alpha\lambda}$ counts the even permutations $\pi \in A_n$ such that $\prod_{i=1}^n h_{\lambda_i - i + \pi(i)} = h_\alpha$. Similarly, $N^-_{\alpha\lambda}$ is defined by counting the odd permutations $\pi \in S_n \setminus A_n$. Hence the functions $(\alpha, \lambda) \mapsto N^+_{\alpha\lambda}$ and $(\alpha, \lambda) \mapsto N^-_{\alpha\lambda}$ are contained in the class $\#\mathbf{P}$.

Definition 4.3 (Decomposition)**.** Given partitions $\alpha, \mu \vdash n$. A finite sequence of partitions $D = \left(\mu^0, \ldots, \mu^{\ell(\alpha)}\right)$ with $\emptyset = \mu^0 \subseteq \mu^1 \subseteq \cdots \subseteq \mu^{\ell(\alpha)} = \mu$ and $|\mu^i / \mu^{i-1}| = \alpha_i$ for all i is called a *decomposition of shape* μ *and type* α. The set of decompositions of shape μ and type α is denoted with $D(\mu, \alpha)$. ■

See Figure 4.1 for an illustration.

We can define *skew Schur polynomials* $s_{\lambda/\alpha}$ similarly to the Schur polynomials:

$$s_{\lambda/\alpha} := \sum_{\mu \in \mathbb{N}^m, |\mu| = n} \mathbf{K}_{\lambda/\alpha;\mu} X^\mu \in \mathbb{Z}[X_1, \ldots, X_m].$$

A formula of Garsia and Remmel (see also [Sta99, Ex. 7.84, p. 478]) states that for $\alpha, \mu \vdash n$ we have

$$h_\alpha * s_\mu = \sum_{\substack{D=\left(\mu^0,\ldots,\mu^{\ell(\alpha)}\right)\\ \in D(\mu,\alpha)}} \prod_{i=1}^{\ell(\alpha)} s_{\mu^i/\mu^{i-1}}. \tag{4.2}$$

If for any decomposition $D = \left(\mu^0, \ldots, \mu^{\ell(\alpha)}\right)$ of shape μ and type α, we denote by $\pi(D)/\varrho(D)$ the skew diagram $\mu^1 * (\mu^2/\mu^1) * \cdots * (\mu^{\ell(\alpha)}/\mu^{\ell(\alpha)-1})$, then $s_{\pi(D)/\varrho(D)} = \prod_i s_{\mu^i/\mu^{i-1}}$. Therefore we can restate (4.2) as

$$h_\alpha * s_\mu = \sum_{D\in D(\mu,\alpha)} s_{\pi(D)/\varrho(D)}. \tag{4.3}$$

It is well known that the Littlewood-Richardson coefficients appear in the expansion of skew Schur polynomials as

$$s_{\pi(D)/\varrho(D)} = \sum_{\nu\vdash n} c^{\pi(D)}_{\varrho(D)\nu}\ s_\nu. \tag{4.4}$$

We set

$$M^\nu_{\alpha\mu} := \sum_{D\in D(\mu,\alpha)} c^{\pi(D)}_{\varrho(D)\nu},$$

which according to (4.3) and (4.4) results in

$$h_\alpha * s_\mu = \sum_{D\in D(\mu,\alpha)} \sum_{\nu\vdash n} c^{\pi(D)}_{\varrho(D)\nu}\ s_\nu = \sum_{\nu\vdash n} M^\nu_{\alpha\mu} s_\nu. \tag{4.5}$$

The Littlewood-Richardson rule implies that the map $(\pi, \rho, \nu) \mapsto c^\pi_{\varrho\nu}$ is in the class $\#\mathbf{P}$ (compare [Nar06]). Since $\#\mathbf{P}$ is closed under exponential summation (see Proposition 2.6), the map $(\alpha, \mu, \nu) \mapsto M^\nu_{\alpha\mu}$ is contained in $\#\mathbf{P}$ as well. Combining (4.1) and (4.5), we have

$$\begin{aligned}
s_\lambda * s_\mu &\overset{(4.1)}{=} \left(\sum_{\alpha\vdash n} N^+_{\alpha\lambda} h_\alpha \ - \ \sum_{\alpha\vdash n} N^-_{\alpha\lambda} h_\alpha\right) * s_\mu \\
&= \sum_{\alpha\vdash n} N^+_{\alpha\lambda}(h_\alpha * s_\mu) \ - \ \sum_{\alpha\vdash n} N^-_{\alpha\lambda}(h_\alpha * s_\mu) \\
&\overset{(4.5)}{=} \sum_{\alpha\vdash n} N^+_{\alpha\lambda} \sum_{\nu\vdash n} M^\nu_{\alpha\mu} s_\nu \ - \ \sum_{\alpha\vdash n} N^-_{\alpha\lambda} \sum_{\nu\vdash n} M^\nu_{\alpha\mu} s_\nu \\
&= \sum_{\nu\vdash n} \left(\underbrace{\sum_{\alpha\vdash n} N^+_{\alpha\lambda} M^\nu_{\alpha\mu} \ - \ \sum_{\alpha\vdash n} N^-_{\alpha\lambda} M^\nu_{\alpha\mu}}_{\overset{\text{Def. 3.2}}{=} g_{\lambda,\mu,\nu}}\right) s_\nu
\end{aligned}$$

Hence the expression in the parenthesis equals $g_{\lambda,\mu,\nu}$. Proposition 2.6 implies that the map $(\lambda, \mu, \nu) \mapsto \sum_{\alpha \vdash n} N^{+}_{\alpha\lambda} M^{\nu}_{\alpha\mu}$ is in $\#\mathbf{P}$. Similarly, $(\lambda, \mu, \nu) \mapsto \sum_{\alpha \vdash n} N^{-}_{\alpha\lambda} M^{\nu}_{\alpha\mu}$ is in $\#\mathbf{P}$. Therefore we have written $(\lambda, \mu, \nu) \mapsto g_{\lambda,\mu,\nu}$ as the difference of two functions in $\#\mathbf{P}$, which means that it is contained in **GapP**. $\square$

4.2 Special cases of Kronecker coefficients

There are many special cases in which the calculation of Kronecker coefficients can be done in polynomial time. These situations are obviously not suited to show the hardness of KronCoeff. Rosas (cf. [Ros01]) summarizes and gives new proofs for several cases, at first discovered in [Rem89, Rem92, RW94], where explicit formulas exist that compute $g_{\lambda,\mu,\nu}$ in polynomial time. We briefly discuss these results.

A one-row partition If $\lambda = (n)$ is a one-row partition, then $g_{\lambda,\mu,\nu} = \begin{cases} 1 & \text{if } \mu = \nu \\ 0 & \text{otherwise} \end{cases}$.
The proof follows directly from (3.1), $\chi^{(n)} = 1$ and the *orthogonal relations* stating that for any two irreducible representations D_1 and D_2 of a finite group G we have $\frac{1}{n!} \sum_{g \in S_n} \chi_{D_1}(g) \chi_{D_2}(g) = \begin{cases} 1 & \text{if } D_1 \text{ is isomorphic to } D_2 \\ 0 & \text{otherwise} \end{cases}$.

Two two-row partitions ([RW94, Ros01]) If $\mu = (\mu_1, \mu_2) \vdash n, \nu = (\nu_1, \nu_2) \vdash n, \lambda \vdash n$ and $\ell(\lambda) > 4$, then $g_{\lambda,\mu,\nu} = 0$. If $\ell(\lambda) \leq 4$, then $g_{\lambda,\mu,\nu}$ can be described as the number of paths through certain rectangles (cf. [Ros01, Thm. 39]). Explicit formulas are also given for these. From these [Ros01] concludes that the set of $g_{\lambda,\mu,\nu}$, where μ and ν are two-row partitions, is unbounded, i.e. multiplicities can become arbitrarily large.

Two hook partitions ([Rem89, Ros01]) For $\mu = (\mu_1, \underbrace{1, \ldots, 1}_{n-\mu_1 \text{ times}}) \vdash n, \nu = (\nu_1, \underbrace{1, \ldots, 1}_{n-\nu_1 \text{ times}}) \vdash n, \lambda \vdash n$, the formula for $g_{\lambda,\mu,\nu}$ gets rather complicated, but is still computable in polynomial time. In this case we have $g_{\lambda,\mu,\nu} \in \{0, 1, 2\}$.

A hook partition and a two-row partition ([Rem92, Ros01]) If $\mu = (\mu_1, \underbrace{1, \ldots, 1}_{n-\mu_1 \text{ times}}), \nu = (\nu_1, n - \nu_1), \lambda \vdash n$, the formula for $g_{\lambda,\mu,\nu}$ also is rather complicated, but is still computable in polynomial time. In this case we have $g_{\lambda,\mu,\nu} \in \{0, 1, 2, 3\}$.

Certain two-row partitions ([BO07]) In the case of $\mu = (n-p, p), \lambda \vdash n, \nu \vdash n$ such that $n \geq 2p$ and $\lambda_1 \geq 2p-1$, Ballantine and Orellana (cf. [BO07]) give a combinatorial interpretation of $g_{\lambda,\mu,\nu}$. In Section 4.4 we will see that this description is the key result that enables us to prove the #**P**-hardness of KronCoeff.

4.3 Ballantine and Orellana's description

To understand the description of the Kronecker coefficients from Ballantine and Orellana, we recall the definitions from [BO07].

Definition 4.4 (α-lattice permutation)**.** Let $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)$ be a partition. A sequence $a = (a_1, a_2, \ldots, a_n)$ is called an *α-lattice permutation*, if the concatenation $(\underbrace{1,\ldots,1}_{\alpha_1 \text{ times}}, \underbrace{2,\ldots,2}_{\alpha_2 \text{ times}}, \ldots, \underbrace{n,\ldots,n}_{\alpha_n \text{ times}})||a$ is a lattice permutation. ■

For example, the word $w = (4,4,3,2,1,1,1,4,3,2)$ is not a lattice permutation, but an α-lattice permutation for $\alpha = (4,3,2)$. As the concatenation of two lattice permutations is a lattice permutation, the concatenation $a||b$ of an α-lattice permutation a and a lattice permutation b is an α-lattice permutation.

Definition 4.5 ((λ, ν, α)-Kronecker-Tableau)**.** Let the λ, α, ν be partitions such that $\alpha \subseteq \lambda \cap \nu$. A semistandard skew tableau T is called a *(λ, ν, α)-Kronecker-tableau*, if it has shape λ/α and type $\nu - \alpha$, its reverse reading word is an α-lattice permutation and additionally one of the following three conditions is satisfied:

- $\alpha_1 = \alpha_2$
- $\alpha_1 > \alpha_2$ and the number of 1's in the second row of T is exactly $\alpha_1 - \alpha_2$,
- $\alpha_1 > \alpha_2$ and the number of 2's in the first row of T is exactly $\alpha_1 - \alpha_2$.

We denote by $k^{\alpha}_{\lambda;\nu}$ the number of (λ, ν, α)-Kronecker-tableaux. ■

The reader may forgive that we did not use the same sub- and superscript order as in [BO07]. In our notation the outer shape and the type are always in the subscript, as in the case of the Kostka numbers as well. See Figure 4.2 for an example of a (λ, ν, α)-Kronecker-tableau.

The following theorem gives the desired combinatorial interpretation:

Theorem 4.6 (Key theorem from [BO07])**.** *Suppose $\mu = (n-p, p), \lambda \vdash n, \nu \vdash n$ such that $n \geq 2p$ and $\lambda_1 \geq 2p-1$. Then we have*

$$g_{\lambda,\mu,\nu} = g_{\lambda,(n-p,p),\nu} = \sum_{\substack{\beta \vdash p \\ \beta \subseteq \lambda \cap \nu}} k^{\beta}_{\lambda;\nu}.$$

$$\begin{array}{|c|c|c|c|c|}\hline \bullet & \bullet & \bullet & 3 & 3 \\ \hline \bullet & \bullet & \bullet & 4 & \multicolumn{1}{c}{} \\ \cline{1-4}\end{array}$$

Figure 4.2: A (λ, ν, α)-Kronecker-tableau T of shape λ/α and type $\nu - \alpha$ for $\lambda = (5,4)$, $\nu = (3,3,2,1)$ and $\alpha = (3,3)$. $w^{\leftarrow}(T) = (3,3,4)$ is an α-lattice permutation.

4.4 Lower bound for KronCoeff

Definition 4.7 (The problem KostkaSub)**.** Given a two-row partition $x = (x_1, x_2) \vdash m$ and $y = (y_1, \ldots, y_\ell)$ with $|y| = m$, the problem of computing the Kostka number $\mathbf{K}_{xy}$ is denoted by KostkaSub. ■

Narayanan proved that KostkaSub is #**P**-complete (cf. [Nar06]). In this section we will see that Ballantine and Orellana's description is the key result that enables us to reduce the #**P**-complete problem KostkaSub to KronCoeff, which results in KronCoeff being #**P**-hard and therefore **GapP**-complete (see Proposition 2.12 and Proposition 4.2). This proves Theorem 4.1. Although not needed for the hardness result, our reduction will be parsimonious.

Proposition 4.8. *The problem* KronCoeff *of computing Kronecker coefficients is* **GapP***-hard.*

Given a two-row partition $x = (x_1, x_2) \vdash m$ and a type $y = (y_1, \ldots, y_\ell)$ with $|y| = m$, we have to find $n, p \in \mathbb{N}$, $\lambda, \nu \vdash n$ computable in polynomial time with $\mathbf{K}_{xy} = g_{\lambda,(n-p,p),\nu}$. This will be obtained step by step by the construction of several bijections between classes of semistandard tableaux.

The rest of this section will be devoted to the proof of Proposition 4.8. For the entire proof we fix a two-row partition $x = (x_1, x_2) \vdash m$ and a type $y = (y_1, \ldots, y_\ell)$ with $|y| = m$.

For any skew shape λ and any type ν we denote by $\mathscr{T}_{\lambda;\nu}$ the set of all semistandard skew tableaux of shape λ and type ν. So $\mathbf{K}_{\lambda\nu} = |\mathscr{T}_{\lambda;\nu}|$.

Definition 4.9. Given a skew shape λ and a type ν. We call the tuple $(\lambda;\nu)$ *α-nice*, if α is a partition and for all skew tableaux $T \in \mathscr{T}_{\lambda;\nu}$ the reverse reading word $w^{\leftarrow}(T)$ is an α-lattice permutation. ■

In a first step, we try to find $n, p \in \mathbb{N}$, $\lambda, \nu \vdash n, \alpha \vdash p$ such that we get a bijection between

$$\mathscr{T}_{x;y} \leftrightarrow \{(\lambda, \nu, \alpha)\text{-Kronecker tableaux}\}. \tag{4.6}$$

The idea is to find λ, ν and α such that $(\lambda/\alpha; \nu - \alpha)$ is α-nice, which will help us to set up the bijection. From this, we will go on and find $n, p \in \mathbb{N}$, $\lambda, \nu \vdash n$ with

1	1	1	2	3	4	4
2	3	4				

Figure 4.3: A semistandard Young tableau T with shape $x = (7, 3)$ and type $y = (3, 2, 2, 3)$. $\varrho = (2 + 2 + 3 = 7, 2 + 3 = 5, 3) = (7, 5, 3)$. $w^{\leftarrow}(T) = (4, 4, 3, 2, 1, 1, 1, 4, 3, 2)$ is not a lattice-permutation, but a ϱ-lattice permutation.

$\mathbf{K}_{xy} = g_{\lambda,(n-p,p),\nu}$. This will be done by showing that $g_{\lambda,(n-p,p),\nu}$ counts a number of (λ, ν, α)-Kronecker tableaux, where the parameter α will be implicitly given by λ, ν, n and p.

We will construct the bijection (4.6) step by step. To see the main difficulty, we have a look at the trivial construction: $\lambda := x, \mu := y$ and $\alpha := ()$ is the empty partition. We get an equality between $\mathscr{T}_{x;y}$ and $\mathscr{T}_{\lambda;\nu}$, but in general $(\lambda; \nu)$ is not $()$-nice, i.e. it is not true that every semistandard skew tableau of shape $\lambda/\alpha = \lambda$ and type $\nu - \alpha = \nu$ has a lattice permutation as its reverse reading word. The following lemma can be used to overcome this problem:

Lemma 4.10 (ϱ-lattice permutation)**.** *Given a word w of type $y = (y_1, \ldots, y_\ell)$. Then w is a ϱ-lattice permutation for $\varrho = (\sum_{i>1} y_i, \sum_{i>2} y_i, \ldots, y_\ell)$.*

We define for a word w, $i \in \mathbb{N}_{\geq 1}$, $k \in \mathbb{N}$:

$$\#(i, k, w) := \text{the number of entries } i \text{ up to } k \text{ in the word } w.$$

For notational convenience, we define for a skew tableau T, $i \in \mathbb{N}_{\geq 1}$, $k \in \mathbb{N}$:

$$\#(i, k, T) := \#(i, k, w^{\leftarrow}(T)).$$

Proof of Lemma 4.10. The entries in y are nonnegative and thus ϱ is a partition. Let $1 \leq k \leq |w|$ be a position in w. For every entry $i \geq 1$ we have

$$\#(i, k, w) + \varrho_i \geq \varrho_i = \varrho_{i+1} + y_{i+1} \geq \#(i + 1, k, w) + \varrho_{i+1}.$$

Therefore w is a ϱ-lattice permutation, which proves the claim. □

Let $\varrho := (\sum_{i>1} y_i, \sum_{i>2} y_i, \ldots, y_\ell)$. Then Lemma 4.10 shows that the reverse reading word $w^{\leftarrow}(T)$ of each skew tableau T of shape x and type y is a ϱ-lattice permutation (see Figure 4.3). Therefore $(x; y)$ is ϱ-nice.

Lemma 4.11 (Type shifting)**.** *Let $k \in \mathbb{N}$. Then there is a bijection between $\mathscr{T}_{x;y}$ and $\mathscr{T}_{x;(\underbrace{0,\ldots,0}_{k \text{ times}},y_1,y_2,\ldots,y_\ell)}$. Moreover, $\left(x; (\underbrace{0, \ldots, 0}_{k \text{ times}}, y_1, y_2, \ldots, y_\ell)\right)$ is $(\underbrace{m, \ldots, m}_{k \text{ times}})||\varrho$-nice.*

See Figure 4.4 for an illustration.

$$\begin{array}{|c|c|c|c|c|c|c|}\hline 1&1&1&2&3&4&4\\ \hline 2&3&4\\ \cline{1-3}\end{array} \quad \xleftrightarrow{\text{bij.}} \quad \begin{array}{|c|c|c|c|c|c|c|}\hline 4&4&4&5&6&7&7\\ \hline 5&6&7\\ \cline{1-3}\end{array}$$

Figure 4.4: Illustration of the bijection between $\mathscr{T}_{x;y}$ and $\mathscr{T}_{x;(0,0,0,y_1,\ldots,y_\ell)}$.

Proof. Let $x^* := x$, $y^* := (\underbrace{0,\ldots,0}_{k\text{ times}}, y_1, y_2, \ldots, y_\ell)$, $\varrho^* := (\underbrace{m,\ldots,m}_{k\text{ times}})||\varrho$. Consider $\eta : \mathscr{T}_{x;y} \to \mathscr{T}_{x^*;y^*}$ which sends each box entry e to $e+k$. This is clearly a well-defined bijection, because the preimage is semistandard iff the image is semistandard.

We know that ϱ^* is a partition, because ϱ is a partition and $m \geq \varrho_1$. We have to show that $(x^*;y^*)$ is ϱ^*-nice.

Let $\eta(T) \in \mathscr{T}_{x^*;y^*}$ with reverse reading word $w^{\leftarrow}(\eta(T)) = (w_1+k, w_2+k, \ldots, w_n+k)$. As $(x;y)$ is ϱ-nice, we have that $w^{\leftarrow}(T) = (w_1, w_2, \ldots, w_n)$ is a ϱ-lattice permutation.

Let $1 \leq j \leq |w^{\leftarrow}(\eta(T))|$ be a position in $w^{\leftarrow}(\eta(T))$.

For the first entries $1 \leq i \leq k-1$ we have that

$$\#(i,j,\eta(T)) + \varrho^*_i = m = \#(i+1,j,\eta(T)) + \varrho^*_{i+1}.$$

For the kth entry we have

$$\begin{aligned} \overbrace{\#(k,j,\eta(T))}^{=0} + \varrho^*_k \quad &= \quad m \quad = \quad \varrho_1 + y_1 \\ &\geq \quad \#(k+1,j,\eta(T)) + \varrho_1 \\ &= \quad \#(k+1,j,\eta(T)) + \varrho^*_{k+1}. \end{aligned}$$

As $w^{\leftarrow}(T)$ is an ϱ-lattice permutation, we have for the other entries $i > k$ that

$$\begin{aligned} \#(i,j,\eta(T)) + \varrho^*_i \quad &= \quad \#(i-k,j,T) + \varrho_{i-k} \\ &\geq \quad \#(i-k+1,j,T) + \varrho_{i-k+1} \\ &= \quad \#(i+1,j,\eta(T)) + \varrho^*_{i+1}. \end{aligned}$$

□

Now let $x^* := x, y^* := (0,0,0,y_1,y_2,\ldots,y_\ell), \varrho^* := (m,m,m)||\varrho$. Then, according to Lemma 4.11, there is a bijection between $\mathscr{T}_{x;y}$ and $\mathscr{T}_{x^*;y^*}$. Moreover, $\big(x^*;y^*\big)$ is ϱ^*-nice.

Lemma 4.12 (Adding 1s)**.** *Given $M \in \mathbb{N}, M \geq x^*_1$. Then there is a bijection between $\mathscr{T}_{x^*;y^*}$ and $\mathscr{T}_{(M)||x^*;(M,0,0,\ldots)+y^*}$. Moreover, $\big((M)||x^*;(M,0,0,\ldots)+y^*\big)$ is ϱ^*-nice as well.*

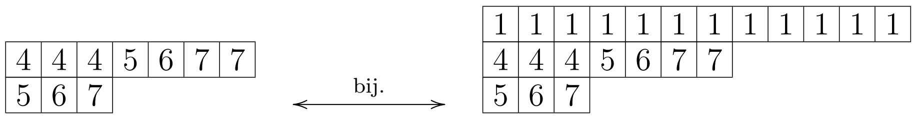

Figure 4.5: Illustration of the bijection between $\mathscr{T}_{x^*;y^*}$ and $\mathscr{T}_{(M)||x^*;(M,0,0,\dots)+y^*}$ with $M = 12$.

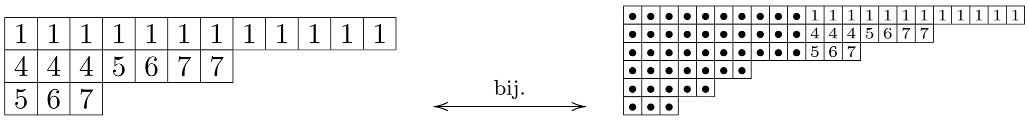

Figure 4.6: Illustration of the bijection between $\mathscr{T}_{x^{**};y^{**}}$ and $\mathscr{T}_{(x^{**}+\varrho^{**})/\varrho^{**};y^{**}}$ with $M = 12$.

See Figure 4.5 for an illustration.

Proof. $(M)||x^*$ is a partition, because $M \geq x_1^*$. Consider $\eta : \mathscr{T}_{x^*;y^*} \to \mathscr{T}_{(M)||x^*;(M,0,0,\dots)+y^*}$ that adds a top row that is filled with M 1s. The map η is well-defined, i.e. the image is semistandard, because $y_1^* = 0$ ensures column strictness. The map η is a bijection, because the column strictness of semistandard tableaux forces 1s to be in the first row. It remains to show that $\big((M)||x^*; (M, 0, 0, \dots) + y^*\big)$ is ϱ^*-nice. Given $T \in \mathscr{T}_{x^*;y^*}$, then $w^{\leftarrow}(\eta(T)) = (\underbrace{1, \dots, 1}_{M \text{ times}})||w^{\leftarrow}(T)$, which is a ϱ^*-lattice permutation, because $w^{\leftarrow}(T)$ is a ϱ^*-lattice permutation. □

We set $x^{**} := (M)||x^*$, $y^{**} := (M, 0, 0, \dots) + y^*$, $\varrho^{**} := \varrho^*$. According to Lemma 4.11 and Lemma 4.12 we obtain the bijections

$$\mathscr{T}_{x;y} \leftrightarrow \mathscr{T}_{x^*;y^*} \leftrightarrow \mathscr{T}_{x^{**};y^{**}}.$$

Moreover, $(x^{**}; y^{**})$ is ϱ^{**}-nice. Note that x^{**} and y^{**} are dependent of M.

Remark 4.13. As $\ell(x^{**}) \leq 3$ and $\varrho_1^* = \varrho_2^* = \varrho_3^*$, there is an obvious bijection (see Figure 4.6) between $\mathscr{T}_{x^{**};y^{**}}$ and $\mathscr{T}_{(x^{**}+\varrho^{**})/\varrho^{**};y^{**}}$. Moreover, $\big((x^{**} + \varrho^{**})/\varrho^{**}; y^{**}\big)$ is ϱ^{**}-nice as well.

Lemma 4.14. *$\mathscr{T}_{(x^{**}+\varrho^{**})/\varrho^{**};y^{**}}$ equals the set of (λ, ν, α)-Kronecker tableaux where $\lambda = x^{**} + \varrho^{**}$, $\nu = y^{**} + \varrho^{**}$ and $\alpha = \varrho^{**}$.*

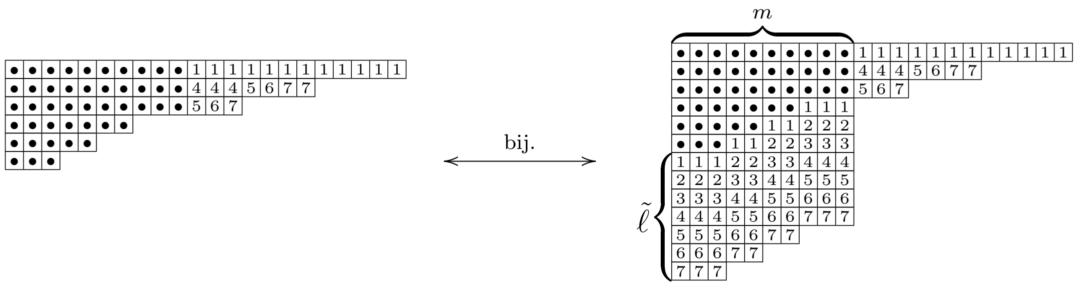

Figure 4.7: Illustration of the bijection between $\mathscr{T}_{\lambda/\alpha;\nu-\alpha}$ and $\mathscr{T}_{\tilde{\lambda}/\alpha;\tilde{\nu}-\alpha}$ with $m = 10, \tilde{\ell} = 7$.

Proof. We know that $\alpha = \varrho^{**}$ is a partition. λ is the sum of two partitions and therefore a partition. We have $\nu = (M+m, m, m, y_1 + \varrho_1, y_2 + \varrho_2, \dots, y_\ell) = (M+m, m, m, m, \varrho_1, \dots, \varrho_{\ell-1})$, which is a partition, because $\varrho_1 \leq |y| = m$. As $\alpha_1 = \alpha_2$ and and $\big(\lambda/\alpha, \nu - \alpha\big)$ is α-nice, the set of (λ, ν, α)-Kronecker tableaux equals $\mathscr{T}_{\lambda/\alpha;\nu-\alpha} = \mathscr{T}_{(\varrho^{**}+x^{**})/\varrho^{**};y^{**}}$. $\square$

With Lemma 4.14 we established a bijection between

$$\mathscr{T}_{x;y} \leftrightarrow \mathscr{T}_{\lambda/\alpha;\nu-\alpha} = \{(\lambda, \nu, \alpha)\text{-Kronecker tableaux}\}.$$

For the rest of this section, we fix λ, ν, α as in Lemma 4.14.

Now we want to connect this result with the Kronecker coefficients. With Theorem 4.6 we have $g_{\tilde{\lambda},(n-p,p),\tilde{\nu}} = \sum_{\substack{\beta \vdash p \\ \beta \subseteq \tilde{\lambda} \cap \tilde{\nu}}} k^{\beta}_{\tilde{\lambda};\tilde{\nu}}$ for $\tilde{\lambda} \vdash n, \tilde{\nu} \vdash n$ if $n \geq 2p$ and $\tilde{\lambda}_1 \geq 2p - 1$. The next crucial lemma gives the desired connection:

Lemma 4.15. *Let $\tilde{\ell} := \ell(|\nu|) = \ell + 3$. Let $\tilde{\lambda} := (\lambda' + \big(\underbrace{\tilde{\ell}, \dots, \tilde{\ell}}_{m \text{ times}}\big))'$ result from λ by adding $\tilde{\ell}$ additional boxes in each of the first m columns (see Figure 4.7 for an illustration). Let $\tilde{\nu} := \nu + (\underbrace{m, \dots, m}_{\tilde{\ell} \text{ times}})$. Then $\tilde{\lambda}$ and $\tilde{\nu}$ are partitions and there is a bijection between $\mathscr{T}_{\lambda/\alpha;\nu-\alpha}$ and $\mathscr{T}_{\tilde{\lambda}/\alpha;\tilde{\nu}-\alpha}$. Moreover, $\big(\tilde{\lambda}/\alpha; \tilde{\nu} - \alpha\big)$ is α-nice and $\mathscr{T}_{\tilde{\lambda}/\alpha;\tilde{\nu}-\alpha} = \{(\tilde{\lambda}, \tilde{\nu}, \alpha)$-Kronecker tableaux$\}$.*

Additionally, $\mathscr{T}_{\tilde{\lambda}/\beta;\tilde{\nu}-\beta} = \emptyset$ for all $\beta \vdash |\alpha|$ that satisfy $\beta \subseteq \tilde{\lambda} \cap \tilde{\nu}$ and $\beta \neq \alpha$.

Before proving Lemma 4.15, we present its implications. Recall that

$$\begin{aligned} \lambda &= x^{**} + \varrho^{**} = (M, x_1, x_2) + \varrho^* \\ &= (M+m, x_1+m, x_2+m, \varrho_1, \dots, \varrho_{\ell-1}), \end{aligned}$$

$$\begin{aligned}\nu &= y^{**} + \varrho^{**} = (M, 0, 0, y_1, y_2, \ldots, y_\ell) + \varrho^* \\ &= (M + m, m, m, y_1 + \varrho_1, y_2 + \varrho_2, \ldots, y_\ell) = (M + m, m, m, m, \varrho_1, \ldots, \varrho_{\ell-1})\end{aligned}$$

and $\alpha = (m, m, m, \varrho_1, \ldots, \varrho_{\ell-1})$. Therefore

$$\tilde{\lambda} = (M + m, x_1 + m, x_2 + m, \underbrace{m, \ldots, m}_{\tilde{\ell} \text{ times}}, \varrho_1, \ldots, \varrho_{\ell-1}), \tag{4.7}$$

and

$$\tilde{\nu} = (M + 2m, 2m, 2m, 2m, m + \varrho_1, \ldots, m + \varrho_{\ell-1}) \tag{4.8}$$

and we can set $n := |\tilde{\lambda}| = M + (\tilde{\ell} + 3)m + |x| + |\varrho|$ and $p := |\alpha| = 3m + |\varrho|$. If we choose M to be large enough (e.g. $M := 2p - 1 - m$), we have $n \geq 2p$ and $\tilde{\lambda}_1 = 2p - 1$ and therefore all technical restrictions are satisfied to conclude with Theorem 4.6 that $g_{\tilde{\lambda},(n-p,p),\tilde{\nu}} = \sum_{\substack{\beta \vdash p \\ \beta \subseteq \tilde{\lambda} \cap \tilde{\nu}}} k^{\beta}_{\tilde{\lambda};\tilde{\nu}}$. Moreover, $\tilde{\lambda}, \tilde{\mu}, p$ and n can be calculated in polynomial time. As $p = |\alpha|$, we get from Lemma 4.15 the following two equalities:

$$\sum_{\substack{\beta \vdash p \\ \beta \subseteq \tilde{\lambda} \cap \tilde{\nu}}} k^{\beta}_{\tilde{\lambda};\tilde{\nu}} = k^{\alpha}_{\tilde{\lambda};\tilde{\nu}} = k^{\alpha}_{\lambda;\nu}.$$

Hence, applying the proved bijections, we get

$$g_{\tilde{\lambda},(n-p,p),\tilde{\nu}} = \sum_{\substack{\beta \vdash p \\ \beta \subseteq \tilde{\lambda} \cap \tilde{\nu}}} k^{\beta}_{\tilde{\lambda};\tilde{\nu}} = k^{\alpha}_{\lambda;\nu} = |\mathscr{T}_{x;y}| = \mathbf{K}_{xy}.$$

This proves the $\#\mathbf{P}$-hardness of KronCoeff (Proposition 4.8).

Proof of Lemma 4.15. From (4.7) and (4.8) it follows that $\tilde{\lambda}$ and $\tilde{\nu}$ are both partitions. We have $\tilde{\lambda}' = \lambda' + (\underbrace{\tilde{\ell}, \ldots, \tilde{\ell}}_{m \text{ times}})$, which means that for $1 \leq i \leq m$, the ith column of $\tilde{\lambda}$ has $\tilde{\ell}$ more boxes than the ith column of λ. We have $\tilde{\nu} - \nu = (\underbrace{m, \ldots, m}_{\tilde{\ell} \text{ times}})$, which means that in comparison to ν we have additional m copies of each number from 1 to $\tilde{\ell}$ in $\tilde{\nu}$.

Consider $\eta : \mathscr{T}_{\lambda/\alpha;\nu-\alpha} \to \mathscr{T}_{\tilde{\lambda}/\alpha;\tilde{\nu}-\alpha}$, which fills the additional boxes in the first m columns with the numbers from 1 to $\tilde{\ell}$ respecting column strictness (see Figure 4.7 for an illustration). As α is a partition, this results in a semistandard tableau: We have column strictness, because no box is filled in the first m columns in the preimage tableau. We have row monotonicity, because α is a partition and the rows of the new entries cannot overlap with the rows of entries in the preimage tableau. So η is well-defined. It is clearly injective. To show that it is surjective, we have to show that our filling of the first m columns is the only possible semistandard filling of these boxes. This is true, because as $\tilde{\ell} \geq \ell(\nu - \alpha)$, we only have the numbers from 1 up to $\tilde{\ell}$ to fill any boxes and we have exactly $\tilde{\ell}$

boxes to fill in each of the first m columns. So η is surjective because of column strictness.

We now show that $\big(\tilde{\lambda}/\alpha; \tilde{\nu} - \alpha\big)$ is α-nice. Let $T \in \mathscr{T}_{\tilde{\lambda}/\alpha;\tilde{\nu}-\alpha}$. Let $T_{\leq 3}$ be the restriction of T to the first 3 rows and let $T_{\geq 4}$ be the restriction of T to the remaining rows. By assumption $w^{\leftarrow}(T_{\leq 3}) = w^{\leftarrow}(\eta^{-1}(T))$ is an α-lattice permutation. $w^{\leftarrow}(T_{\geq 4})$ is a lattice permutation, which follows from the observation that for each entry $i > 1$ in $T_{\geq 4}$ there is an entry $i-1$ in the same column right above. As $w^{\leftarrow}(T) = w^{\leftarrow}(T_{\leq 3})||w^{\leftarrow}(T_{\geq 4})$ is the concatenation of an α-lattice permutation and a lattice permutation, we conclude that $w^{\leftarrow}(T)$ is an α-lattice permutation. Therefore $\big(\tilde{\lambda}/\alpha; \tilde{\nu} - \alpha\big)$ is α-nice.

We have $\mathscr{T}_{\tilde{\lambda}/\alpha;\tilde{\nu}-\alpha} = \{(\tilde{\lambda}, \tilde{\nu}, \alpha)\text{-Kronecker tableaux}\}$, because $\alpha_1 = \alpha_2$ and $\big(\tilde{\lambda}/\alpha; \tilde{\nu} - \alpha\big)$ is α-nice.

Now we additionally prove that $\mathscr{T}_{\tilde{\lambda}/\beta;\tilde{\nu}-\beta} = \emptyset$ for all $\beta \vdash |\alpha|, \beta \subseteq \tilde{\lambda} \cap \tilde{\nu}, \beta \neq \alpha$. Let $\beta \vdash |\alpha|, \beta \subseteq \tilde{\lambda} \cap \tilde{\nu}$. Assume that we have $T \in \mathscr{T}_{\tilde{\lambda}/\beta;\tilde{\nu}-\beta}$. Then T can only be filled with elements from the set $\{1, 2, \ldots, \tilde{\ell}\}$. Hence, because of T's column strictness property, each of its columns can contain at most $\tilde{\ell}$ boxes. In the ith column of $\tilde{\lambda}$, $1 \leq i \leq m$, there are exactly $\tilde{\ell} + \alpha'_i$ boxes. Since the ith column of T can contain at most $\tilde{\ell}$ boxes, the top α'_i boxes must belong to β, which means $\beta'_i \geq \alpha'_i$ for all $1 \leq i \leq m$. So in the first m columns, this results in at least $\sum_{i=1}^{m} \alpha'_i = |\alpha|$ boxes belonging to β. But $\beta \vdash |\alpha|$, therefore $\beta'_i = \alpha'_i$ for all $1 \leq i \leq m$ and $\beta'_i = 0$ for $i > m$. Hence $\beta' = \alpha'$, which means $\beta = \alpha$ and proves the claim. $\square$

It is easy to see that the proofs in this section are independent of the number of rows in x, but it suffices here to consider two-row partitions.

Chapter 5

Preliminaries – Flows in networks

In this chapter we introduce basic terminology and facts about flows and augmenting-path algorithms (cf. [AMO93, Jun04]). These will be used to describe the algorithms in Chapter 6 and 7. At the end of this chapter we will have a look at the well-known Ford-Fulkerson algorithm and its polynomial-time capacity scaling variant. This capacity scaling approach will be used in Chapter 7 to refine the LRPA (Littlewood-Richardson Positivity Algorithm) into the polynomial-time algorithm LRP-CSA (Littlewood-Richardson Positivity - Capacity Scaling Algorithm).

5.1 Graphs

A *graph* $G = (V, E)$ consists of a finite set V of vertices and a finite set $E \subseteq \binom{V}{2}$ of edges whose elements are unordered pairs of distinct vertices. We say that the edge $\{v, w\} \in E$ *connects* v and w. Since in our case edges are pairs of distinct vertices, our graphs have no loops, which are edges that connect a vertex with itself. We call two vertices v and w *adjacent*, if $\{v, w\} \in E$. We call a vertex v and an edge e *incident*, if $v \in e$. A *face* is a region bounded by edges, including the outer, infinitely-large region.

5.2 Flows on digraphs

Given a graph $G = (V, E)$ we can assign an edge direction to each edge in E by adding to G an *orientation function* $o : E \to V$ which puts the vertices in order by mapping each edge to one of its vertices. This makes G a *directed graph (digraph)*. An edge $\{v, w\}$ can either be *directed away from v and towards w* ($o(\{v, w\}) = v$) or *directed away from w and towards v* ($o(\{v, w\}) = w$). The incident edges of each vertex $v \in V$ can then be divided into $\delta_{\text{in}}(v)$ (the edges that are directed towards v) and $\delta_{\text{out}}(v)$ (the edges that are directed away from v). Now for a mapping $f : E \to \mathbb{R}$ we define

$$\delta_{\text{in}}(v, f) := \sum_{e \in \delta_{\text{in}}(v)} f(e)$$

and

$$\delta_{\text{out}}(v, f) := \sum_{e \in \delta_{\text{out}}(v)} f(e).$$

As a vertex can be contained in the vertex set of several digraphs, it is not always clear from the context which underlying digraph is meant. In these situations we add an additional superscript as for example in $\delta^G_{\text{in}}(v, f)$ or $\delta^G_{\text{out}}(v, f)$ to avoid confusion.

Definition 5.1 (Flow)**.** A *flow* f on a digraph $G = (V, E)$ is a mapping $f : E \to \mathbb{R}$ which satisfies the following flow constraints:

$$\forall v \in V : \delta_{\text{in}}(v, f) = \delta_{\text{out}}(v, f) \tag{5.1}$$

■

Flows are also called *circulations* in the literature.

Flow vector space We note that negative flows on edges are allowed and that therefore the flows on a digraph $G = (V, E, o)$ form a real vector space $F(G)$, which is a subspace of the vector space of mappings $E \to \mathbb{R}$. The next lemma shows that the choice of the specific orientation function o is not essential.

Lemma 5.2. *Let $G = (V, E, o)$, $G' = (V, E, o')$ be two digraphs that share the vertex and edge set but have different orientation functions. Then there is the following natural isomorphism of vector spaces:*

$$\iota_o^{o'} : F(G) \to F(G'), \forall e \in E : \iota_o^{o'}(f)(e) = \begin{cases} f(e) & \textit{if } o(e) = o'(e) \\ -f(e) & \textit{otherwise} \end{cases}.$$

Proof. We need to show that $\iota_o^{o'}$ is well-defined, i.e. for all $f \in F(G)$ we have $\iota_o^{o'} \in F(G')$. It is sufficient to prove the claim for two orientations o and o' that differ only on one edge $\{v, w\}$. Let $\{v, w\}$ be directed from v to w w.r.t. o and from w to v w.r.t. o'. Let f be a flow on G. Trivially, the flow constraints are satisfied for G' in every node of $V \setminus \{v, w\}$.
Since $\delta^G_{\text{in}}(v, f) = \delta^G_{\text{out}}(v, f)$, we have
$\delta^{G'}_{\text{in}}(v, f) = \delta^G_{\text{in}}(v, f) + (-f(e)) = \delta^G_{\text{out}}(v, f) - f(e) = \delta^{G'}_{\text{out}}(v, f)$.
And since $\delta^G_{\text{in}}(w, f) = \delta^G_{\text{out}}(w, f)$, we have
$\delta^{G'}_{\text{in}}(w, f) = \delta^G_{\text{in}}(w, f) - f(e) = \delta^G_{\text{out}}(w, f) + (-f(e)) = \delta^{G'}_{\text{out}}(w, f)$.
Thus $\iota_o^{o'}(f) \in F(G')$ and thus $\iota_o^{o'}$ is well-defined.

Clearly, $\iota_o^{o'}$ is a linear map. It is bijective, because $\iota_{o'}^{o}$ is inverse to $\iota_o^{o'}$. Therefore $\iota_o^{o'}$ is an isomorphism of vector spaces. □

We now analyze the dimension of the vector space of flows on a digraph.

Definition 5.3 (Path, connected vertices)**.** Given a graph $G = (V, E)$ and two vertices $v_1, v_2 \in V$. A *path between* v_1 *and* v_2 *in* G is a finite sequence of distinct nodes $v_1 = v^1, \dots, v^m = v_2$ such that $\{v^i, v^{i+1}\} \in E$ for all $1 \leq i < m$.

Two vertices $v_1, v_2 \in V$ are called *connected*, if there exists a path between v_1 and v_2. ■

It is easy to show that being connected is an equivalence relation on V. Each equivalence class is called a *connected component*. If a graph $G = (V, E)$ has only 1 connected component or if $V = \emptyset$, then G is called *connected*.

Lemma 5.4. *The flows on a digraph* $G = (V, E, o)$ *form a real vector space* $F(G)$ *with dimension* $\dim F(G) = |E| - |V| +$ *#connected components of* G*.*

Proof. Let $C_1, \dots, C_m$ be the connected components of G. For each connected component C_i and for each mapping $f : E \to \mathbb{R}$ we have

$$\sum_{v \in C_i} \delta_{\text{in}}(v, f) = \sum_{v \in C_i} \delta_{\text{out}}(v, f), \tag{5.2}$$

because each edge contributes exactly once to the left sum and exactly once to the right sum. Choose an arbitrary vertex $v_i \in C_i$ and let $f : E \to \mathbb{R}$ be a map such that $\delta_{\text{in}}(v, f) = \delta_{\text{out}}(v, f)$ for all $v \in C_i \setminus v_i$. Then we can deduce $\delta_{\text{in}}(v_i, f) = \delta_{\text{out}}(v_i, f)$ as follows:

$$\begin{aligned} & \forall v \in C_i \setminus \{v_i\} : \delta_{\text{in}}(v, f) = \delta_{\text{out}}(v, f) \\ \Rightarrow \quad & \sum_{v \in C_i \setminus \{v_i\}} \delta_{\text{in}}(v, f) = \sum_{v \in C_i \setminus \{v_i\}} \delta_{\text{out}}(v, f) \\ \overset{(5.2)}{\Rightarrow} \quad & \delta_{\text{in}}(v_i, f) = \delta_{\text{out}}(v_i, f). \end{aligned}$$

Hence in each connected component one flow constraint equality can be left out and thus

$$\dim F(G) \geq |E| - |V| + \#\text{connected components}.$$

Now omit 2 restrictions in a connected component C: $\delta_{\text{in}}(v_1, f) = \delta_{\text{out}}(v_1, f)$ and $\delta_{\text{in}}(v_2, f) = \delta_{\text{out}}(v_2, f)$ with $v_1, v_2 \in C$. Given a flow $f \in F(G)$. As C is a connected component, there exists a path between v_1 and v_2. Then by sending 1 unit along the path (that means increasing/decreasing the flow on the path's edges while respecting all flow constraints but the ones in v_1 and v_2), f can be transformed into a mapping $f' : E \to \mathbb{R}$ where all flow constraints except the flow constraints in v_1 and v_2 are satisfied. This shows that omitting 2 or more restrictions in the same connected component strictly extends the vector space $F(G)$ beyond consisting of only flows. Therefore we have

$$\dim F(G) = |E| - |V| + \#\text{connected components}.$$

□

5.3 Flow decomposition

We want to decompose flows into smaller parts called cycles which are easier to handle. Therefore in this section we describe the fairly standard idea of flow decomposition.

Definition 5.5 (Cycle)**.** A *cycle* $c = (v_1, \ldots, v_\ell, v_{\ell+1} = v_1)$ on a graph $G = (V, E)$ is a finite sequence of vertices in V with the following properties:

- $\ell \geq 3$
- $\forall 1 \leq i, j \leq \ell, i \neq j : v_i \neq v_j$
- $\forall 1 \leq i \leq \ell : \{v_i, v_{i+1}\} \in E$

We can see the edges $\{v_i, v_{i+1}\}, 1 \leq i \leq \ell$ as part of the cycle and write $\{v_i, v_{i+1}\} \in c$. The length $\ell(c)$ is defined as the number of edges in c. ■

Given a digraph $G = (V, E, o)$, we assign a *cycle flow* f_c to each cycle $c = (v_1, \ldots, v_\ell, v_{\ell+1} = v_1)$ on G by setting for all $1 \leq i \leq \ell$:

$$f_c(\{v_i, v_{i+1}\}) := \begin{cases} 1 & \text{if } \{v_i, v_{i+1}\} \text{ is directed from } v_i \text{ towards } v_{i+1} \\ -1 & \text{if } \{v_i, v_{i+1}\} \text{ is directed from } v_{i+1} \text{ towards } v_i \end{cases}$$

and $f_c(e) := 0$ for all $e \in E \setminus c$.

To simplify the notation, we identify c with its cycle flow f_c. Note that changing the underlying orientation from o to o' changes a cycle's flow c to $\iota_o^{o'}(c)$ (cf. Lemma 5.2).

We define the *support* of a flow $f \in F(G)$ as $\mathrm{supp}(f) := \{e \in E \mid f(e) \neq 0\}$.

Lemma 5.6 (Flow decomposition)**.** *Given a digraph $G = (V, E, o)$ and a flow f on G. Then there is $m \leq |\mathrm{supp}(f)|$ and cycles $c_1, \ldots, c_m$ on G and $\alpha_1, \ldots, \alpha_m \in \mathbb{R}_{>0}$ with $\sum_{i=1}^m \alpha_i c_i = f$ such that for all $1 \leq i \leq m$ and for all edges $e \in c_i$ we have $\mathrm{sgn}(c_i(e)) = \mathrm{sgn}(f(e))$. We call α_i the* multiplicity *of the cycle c_i in the decomposition.*

We will prove a stronger variant of this lemma later (cf. Lemma 5.11).

5.4 Capacities

We can assign capacities to a digraph $G = (V, E, o)$ by defining two functions $u : E \to \mathbb{R} \cup \{\infty\}$ and $l : E \to \mathbb{R} \cup \{-\infty\}$ which we call the *upper bound* and *lower bound* respectively. We use the subscript notation $u_e := u(e), l_e := l(e)$. A digraph with capacities is sometimes called a *network* in the literature. Throughout this work, we will restrict ourselves to the simple case where

$$\forall e \in E : l_e \leq 0, u_e \geq 0.$$

This is a general assumption whenever speaking about capacities. An edge e with $l_e = -\infty$ and $u_e = \infty$ is called *uncapacitated*. All other edges are called *capacitated*.

Definition 5.7 (Feasible flow)**.** Let $G = (V, E, o)$ be a digraph with capacities u and l. A flow f on G is denoted *feasible with respect to u and l*, if $l_e \leq f(e) \leq u_e$ on each edge $e \in E$. The set $P_{\text{feas}}(G) \subseteq F(G)$ of feasible flows on G is called the *polyhedron of feasible flows on G*. ■

We now prove that the specific orientation of the edges is not essential for feasible flows as well.

Lemma 5.8. *Given two digraphs $G = (V, E, o)$ and $G' = (V, E, o')$ that share the vertex and edge set but have different orientation functions. Given upper bounds $u : E \to \mathbb{R} \cup \{\infty\}$ and lower bounds $l : E \to \mathbb{R} \cup \{-\infty\}$ on the digraph G and a flow f on G. Define the natural bijective map $\tilde{\iota}_o^{o'}$:*

$$\tilde{\iota}_o^{o'} : F(G) \times (\mathbb{R} \cup \{\infty\})^E \times (\mathbb{R} \cup \{-\infty\})^E \to F(G') \times (\mathbb{R} \cup \{\infty\})^E \times (\mathbb{R} \cup \{-\infty\})^E,$$

$$(f, u, l) \mapsto (\iota_o^{o'}(f), u', l')$$

with $\forall e \in E$

$$u'_e := \begin{cases} u_e & \text{if } o(e) = o'(e) \\ -l_e & \text{otherwise} \end{cases}, \quad l'_e := \begin{cases} l_e & \text{if } o(e) = o'(e) \\ -u_e & \text{otherwise} \end{cases}.$$

Then f is feasible w.r.t. u and l iff $\iota_o^{o'}(f)$ is feasible w.r.t. u' and l'. Thus feasible flows are invariant under $\tilde{\iota}_o^{o'}$.

Proof. Given a flow f on G. Then for each edge $e \in E$ with $o(e) = o'(e)$ we have

$$l_e \leq f(e) \leq u_e \Leftrightarrow l'_e = l_e \leq f(e) = \iota_o^{o'}(f)(e) \leq u_e = u'_e.$$

And for each edge $e \in E$ with $o(e) \neq o'(e)$ we have

$$l_e \leq f(e) \leq u_e \Leftrightarrow -u'_e \leq f(e) \leq -l'_e \Leftrightarrow l'_e \leq \iota_o^{o'}(f)(e) \leq u'_e.$$

□

We define the *directed capacity function* $\vec{u} : V \times V \to \mathbb{R}_{\geq 0}$ of $G = (V, E, o)$ as follows:

$$\vec{u}(v, w) := \begin{cases} 0 & \text{if } \{v, w\} \notin E \\ u_{\{v,w\}} & \text{if } \{v, w\} \in E \text{ and } \{v, w\} \text{ is directed from } v \text{ towards } w \\ -l_{\{v,w\}} & \text{if } \{v, w\} \in E \text{ and } \{v, w\} \text{ is directed from } w \text{ towards } v \end{cases}$$

From the definition we have $\vec{u}(v, w) \geq 0$ for all $v, w \in V$. Note that $\vec{u}(v, w)$ is preserved under any $\tilde{\iota}_o^{o'}$. If it is not clear of which digraph G the capacity functions are meant, we write $\vec{u}^G(v, w)$.

Definition 5.9 (Well-directed cycle)**.** A cycle $c = (v_1, \dots, v_\ell, v_{\ell+1} = v_1)$ is denoted *well-directed*, if for all $1 \leq i \leq \ell$ it holds $\vec{u}(v_i, v_{i+1}) > 0$. ■

Lemma 5.10. *A cycle c is well-directed iff there is an $\varepsilon > 0$ such that the flow εc is a feasible flow.*

Proof. Let $c = (v_1, \dots, v_\ell, v_{\ell+1} = v_1)$ be well-directed. We set

$$\varepsilon := \min_{1 \leq i \leq \ell} \{\vec{u}(v_i, v_{i+1})\}$$

and note that $\varepsilon > 0$. Consider εc. Let $e = \{v_i, v_{i+1}\}$ for any $1 \leq i \leq \ell$. If e is directed from v_i to v_{i+1}, then $l_e \leq 0 \leq \varepsilon c(e) = \varepsilon \leq \vec{u}(v_i, v_{i+1}) = u_e$. If e is directed from v_{i+1} to v_i, then $u_e \geq 0 \geq \varepsilon c(e) = -\varepsilon \geq -\vec{u}(v_i, v_{i+1}) = l_e$. Therefore εc is a feasible flow.

Now let $\varepsilon > 0$ such that εc is a feasible flow. Let $e = \{v_i, v_{i+1}\}$ for any $1 \leq i \leq \ell$. If e is directed from v_i to v_{i+1}, then $l_e \leq \varepsilon c(e) \leq u_e \Rightarrow 0 < \varepsilon \leq u_e$ and therefore $u_e = \vec{u}(v_i, v_{i+1}) > 0$. If e is directed from v_{i+1} to v_i, then $l_e \leq \varepsilon c(e) \leq u_e \Rightarrow 0 > -\varepsilon \geq l_e$ and therefore $-l_e = \vec{u}(v_i, v_{i+1}) > 0$. Therefore c is a well-directed cycle. □

There is a flow decomposition lemma for feasible flows as well (cf. Lemma 5.6):

Lemma 5.11 (Feasible flow decomposition)**.** *Given a digraph $G = (V, E, o)$ and a feasible flow f on G. Then there is $m \leq |\mathrm{supp}(f)|$ and well-directed cycles $c_1, \dots, c_m$ on G and $\alpha_1, \dots, \alpha_m \in \mathbb{R}_{>0}$ with $\sum_{i=1}^m \alpha_i c_i = f$ such that for all $1 \leq i \leq m$ and for all edges $e \in c_i$ we have $\mathrm{sgn}(c_i(e)) = \mathrm{sgn}(f(e))$. We call α_i the* multiplicity *of the cycle c_i in the decomposition.*

Proof. We do induction by the size of the support of f. For the base case assume that $|\mathrm{supp}(f)| = 0$. Thus $f(e) = 0$ for all $e \in E$ and we can choose $m = 0$ to show the induction basis.

Now let the assumption be true for all flows on G whose support contains at most N edges. Let f be a feasible flow on G with $|\mathrm{supp}(f)| = N + 1$. We create a well-directed cycle c as follows: Start at a vertex v that is incident to an edge e_1 with $f(e_1) \neq 0$. Add v to c. Now choose an edge e that is either directed away from v with $f(e) > 0$ or that is directed towards v with $f(e) < 0$. Such an edge must exist because of the flow constraints. Now consider the other vertex incident to e. This is the next vertex in c. Continue this process until you have to add a vertex w to c which you have already added. Then a cycle is found starting at w. Just dismiss the first preceding vertices. Note that with this construction we have for all edges $e \in c : \mathrm{sgn}(c(e)) = \mathrm{sgn}(f(e))$. Lemma 5.10 shows that c is a well-directed cycle. Now set α to be the maximum value such that $|\alpha c| \leq |f|$. By construction α is positive and there is an edge e with $\alpha c(e) = f(e)$. Thus $|\mathrm{supp}(f - \alpha c)| \leq N$. By induction hypothesis, there are well-directed cycles $c_1, \dots, c_m$ and $\alpha_1, \dots, \alpha_m$, $m \leq N$ with $f - \alpha c = \sum_{i=1}^m \alpha_i c_i$ such that for all

$1 \le i \le m$ and for all edges $e \in c_i$ we have $\operatorname{sgn}(c_i(e)) = \operatorname{sgn}(f(e))$. Hence by setting $\alpha_{m+1} := \alpha, c_{m+1} := c$, we get $f = \sum_{i=1}^{m+1} \alpha_i c_i$, $m+1 \le N+1$ and we have for all $1 \le i \le m+1$ and for all edges $e \in c_i : \operatorname{sgn}(c(e)) = \operatorname{sgn}(f(e))$. □

Note that this decomposition is not necessarily unique. Also note that we can prove Lemma 5.6 by setting the capacities on each edge e to $l_e := -\infty, u_e := \infty$ and using Lemma 5.11.

5.5 The Ford-Fulkerson algorithm

The LRPA has a close relationship to the Ford-Fulkerson algorithm (denoted FFA here). See [AMO93, ch. 6] where the FFA is called the "labeling algorithm". We prove the correctness and running time of this well-known algorithm here in a slightly different way as usual and then in Chapter 6 transfer the results to our situation and explain the LRPA. In Section 5.6 we describe the polynomial-time version of the FFA, which we call FF-CSA, which stands for Ford Fulkerson Capacity Scaling Algorithm. The capacity scaling approach is used in Chapter 7 to convert the LRPA into its polynomial-time counterpart LRP-CSA.

We can restate the traditional maximum flow problem ([FF62, AMO93]) in the following (slightly different) way:

Definition 5.12 (Maximum flow problem)**.** Given a digraph $G = (V, E, o)$ with integral capacities $u_e \in \mathbb{Z}_{\ge 0}, l_e \in \mathbb{Z}_{\le 0}$ on each edge e with one special edge $\{t, s\}$ directed from t towards s, the *maximum flow problem* is the problem of computing a feasible flow f on G with maximum $f\big(\{t, s\}\big)$. ■

There are some minor differences to [FF62], but both formulations are easily seen to be equivalent. Although this description of the maximum flow problem is more complicated then the traditional description, it is suitable to illustrate the ideas that are used in the construction of the LRPA.

Recall that $P_{\text{feas}}(G) \subseteq F(G)$ denotes the polytope of feasible flows on G:

$$P_{\text{feas}}(G) = \{f \in F(G) \mid \forall e \in E : l_e \le f(e) \le u_e\}$$

We define the linear function δ, which is to be maximized, as $\delta : F(G) \to \mathbb{R}, f \mapsto f(\{t, s\})$. Note that for all $f \in P_{\text{feas}}(G)$ we have $\delta(f) \le u_{\{t,s\}} < \infty$.

The residual network To state the FFA, we need the construction $\mathsf{RES}(f)$ called the *residual network with respect to f*. $\mathsf{RES}(f)$ has the same underlying digraph as G, only the capacities are different: Each edge's bounds l_e, u_e in G are adjusted to new bounds $l'_e := l_e - f(e), u'_e := u_e - f(e)$. Recall that

$$P_{\text{feas}}(\mathsf{RES}(f)) = \{d \in F(\mathsf{RES}(f)) \mid \forall e \in E : l'_e \le d(e) \le u'_e\}.$$

The following lemma shows a crucial property of the residual network:

Lemma 5.13 (Residual Correspondence Lemma for Maximum Flow)**.** *Given a digraph $G = (V, E, o)$ and a feasible flow $f \in P_{\text{feas}}(G)$. Then for all $d \in F(G)$:*

$$d \in P_{\text{feas}}(\mathsf{RES}(f)) \Longleftrightarrow f + d \in P_{\text{feas}}(G).$$

Proof.

$$f + d \in P_{\text{feas}}(G) \Leftrightarrow \forall e \in E : l_e \leq (f + d)(e) \leq u_e$$
$$\Leftrightarrow \forall e \in E : l_e - f(e) \leq d(e) \leq u_e - f(e) \Leftrightarrow d \in P_{\text{feas}}(\mathsf{RES}(f)).$$

□

As $P_{\text{feas}}(\mathsf{RES}(f)) = P_{\text{feas}}(G) - f := \{d \in F(G) \mid d + f \in P_{\text{feas}}(G)\}$ we have that $P_{\text{feas}}(\mathsf{RES}(f))$ and $P_{\text{feas}}(G)$ are the same polyhedra up to a translation. We will see that the situation of the LRPA is more complicated and that we will not have such a strong Residual Correspondence Lemma. In some cases we will not be able to construct a residual network at all. We will be able to construct $\mathsf{RES}(f)$ only for so-called shattered flows.

The following lemmas lead to the construction of the FFA:

Lemma 5.14. *Given a digraph G with integral capacities u, l and let $f \in P_{\text{feas}}(G)$ be an integral feasible flow on G. Let c be a well-directed cycle on $\mathsf{RES}(f)$. Then $f + c \in P_{\text{feas}}(G)$.*

Proof. We have $c \in P_{\text{feas}}(\mathsf{RES}(f))$, because c is well-directed and the capacities on $\mathsf{RES}(f)$ are integral. Lemma 5.13 shows that $f + c \in P_{\text{feas}}(G)$. □

Lemma 5.15. *Given a digraph $G = (V, E, o)$ with capacities, a feasible flow $f \in P_{\text{feas}}(G)$ and any linear function $\delta : F(G) \to \mathbb{R}$. If there is no well-directed cycle c on $\mathsf{RES}(f)$ with $\delta(c) > 0$, then f maximizes δ in $P_{\text{feas}}(G)$.*

Proof. Let $f \in P_{\text{feas}}(G)$ such that f does not maximize δ in $P_{\text{feas}}(G)$. Then there is $g \in P_{\text{feas}}(G)$ with $\delta(g) > \delta(f)$. Define $d := g - f$. As $f + d \in P_{\text{feas}}(G)$ according to Lemma 5.13 we have $d \in P_{\text{feas}}(\mathsf{RES}(f))$. With Lemma 5.11 we can decompose d into well-directed cycles $c_1, \ldots, c_m$ on $\mathsf{RES}(f)$ with

$$d = \sum_{i=1}^{m} \alpha_i c_i$$

where $\alpha_i > 0$ for all $1 \leq i \leq m$. We have $\delta(d) > 0$, because $\delta(g) > \delta(f)$. As δ is linear there exists $i \in \{1, \ldots, m\}$ with $\delta(c_i) > 0$. This proves the lemma. □

We can now describe the Ford-Fulkerson algorithm, which is Algorithm 1, and prove its correctness.

Remark 5.16. Note that well-directed cycles c with $\delta(c) > 0$ are exactly those cycles, which contain t and s and have $c(\{t, s\}) > 0$. So breadth-first-search or any pathfinding algorithm from s to t will suffice to find a well-directed cycle with that property (line 6). These algorithms run in polynomial time.

Algorithm 1 Ford-Fulkerson algorithm (FFA)

Input: A digraph $G = (V, E, o)$ with integral capacities l and u and one special edge $\{t, s\}$ directed from t towards s.
Output: A feasible flow f on G with maximal flow on $\{t, s\}$.

1: $f \leftarrow 0$.
2: // We have $f \in P_{\text{feas}}(G)$ and f is integral.
3: `done` $\leftarrow$ **false**.
4: **while** not `done` **do**
5:    Construct $\mathsf{RES}(f)$.
6:    **if** there is a well-directed cycle c in $\mathsf{RES}(f)$ with $\delta(c) > 0$ **then**
7:       Augment 1 unit over c: $f \leftarrow f + c$.
8:       // Lemma 5.14 ensures that $f \in P_{\text{feas}}(G)$. Moreover, f is integral.
9:    **else**
10:       `done` $\leftarrow$ **true**.
11:    **end if**
12: **end while**
13: // There are no well-directed cycles c on $\mathsf{RES}(f)$ with $\delta(c) > 0$. Lemma 5.15 ensures that f maximizes δ in $P_{\text{feas}}(G)$.
14: **return** f.

Proposition 5.17. *The FFA terminates on any input* $(G, l, u, \{t, s\})$.

Proof. We have to ensure that the while-loop in line 4 always terminates. Each iteration of the while-loop increases $\delta(f)$ by 1. For the initial solution 0 we have$\delta(0) = 0$. We know that $\delta(f)$ is bounded by $u_{\{t,s\}}$. So the while-loop always terminates. □

Thus we have the following proposition:

Proposition 5.18. *The FFA terminates on any input* $(G, l, u, \{t, s\})$ *and returns a feasible flow* $f \in P_{\text{feas}}(G)$ *which optimizes* δ *in* $P_{\text{feas}}(G)$.

Proof. Combine Proposition 5.17 and Lemma 5.15. □

5.6 The Ford-Fulkerson Capacity Scaling Algorithm

We will use a *capacity scaling* approach in chapter 7 to convert the LRPA into its polynomial-time counterpart LRP-CSA. We now illustrate this scaling approach by showing a scaled version of the Ford-Fulkerson algorithm: The Ford-Fulkerson Capacity Scaling Algorithm, denoted FF-CSA here. See [AMO93, ch. 6, ch. 7.3] where the FF-CSA is called the "labeling algorithm". We prove the correctness and running time of this algorithm in this section in a slightly different way as usual and then in Chapter 7 transfer the results to the situation of the LRPA and explain the LRP-CSA.

Recall that

$$P_{\text{feas}}(G) = \{f \in F(G) \mid \forall e \in E : l_e \leq f(e) \leq u_e\}$$

and that $\delta(f) = f(\{t,s\})$.

The residual network We will use a slightly different residual network $\mathsf{RES}_{2^k}(f)$ defined as follows: We first construct $\mathsf{RES}(f)$ with capacities u', l' as in the FFA. For $k \in \mathbb{N}$ we obtain $\mathsf{RES}_{2^k}(f)$ by defining new capacities:

$$u''_e := \begin{cases} u'_e & \text{if } u'_e \geq 2^k \\ 0 & \text{otherwise} \end{cases}, \quad l''_e := \begin{cases} l'_e & \text{if } l'_e \leq -2^k \\ 0 & \text{otherwise} \end{cases}.$$

Note that $P_{\text{feas}}(\mathsf{RES}_{2^k}(f)) \subseteq P_{\text{feas}}(\mathsf{RES}(f))$.

Lemma 5.19. *Given a digraph $G = (V, E, o)$ with integral capacities u, l and a feasible flow $f \in P_{\text{feas}}(G)$. For each well-directed cycle c on $\mathsf{RES}_{2^k}(f)$ we have $f + 2^k c \in P_{\text{feas}}(G)$.*

Proof. By construction of $\mathsf{RES}_{2^k}(f)$ we have for all $v, w \in V$ that $\vec{u}^{\mathsf{RES}_{2^k}(f)}(v,w) = 0$ or $\vec{u}^{\mathsf{RES}_{2^k}(f)}(v,w) \geq 2^k$. Therefore $2^k c \in P_{\text{feas}}(\mathsf{RES}_{2^k}(f)) \subseteq P_{\text{feas}}(\mathsf{RES}(f))$. Lemma 5.13 shows that $f + 2^k c \in P_{\text{feas}}(G)$. □

The FF-CSA is listed as Algorithm 2. The following lemmas prove its correctness:

Lemma 5.20. *When the FF-CSA terminates on an input $(G, l, u, \{t, s\})$, it returns a feasible flow $f \in P_{\text{feas}}(G)$ that maximizes δ in $P_{\text{feas}}(G)$.*

Proof. When the FF-CSA terminates, there are no well-directed cycles c on $\mathsf{RES}_1(f)$ with $\delta(c) > 0$. The graph G has integral capacities and the flow f stays integral throughout the FF-CSA. Therefore the capacities on all residual networks that appear during a run of the FF-CSA are integral. As in particular the capacities of $\mathsf{RES}_1(f)$ are integral, it follows that we have $\mathsf{RES}(f) = \mathsf{RES}_1(f)$ at line 15. Then from Lemma 5.15 we know that f maximizes δ in $P_{\text{feas}}(G)$. □

It remains to show that the FF-CSA runs in polynomial time:

Definition 5.21. Given a digraph $G = (V, E, o)$ and a cycle $c = (v_1, \ldots, v_\ell, v_{\ell+1} = v_1)$ on $\mathsf{RES}(f)$. A tuple $(v_i, v_{i+1}), 1 \leq i \leq \ell$ is called *critical*, if $\vec{u}^{\mathsf{RES}_{2^k}(f)}(v_i, v_{i+1}) = 0$. ■

Note that well-directed cycles on $\mathsf{RES}_{2^k}(f)$ are exactly those that have no critical tuple. Also note that $\vec{u}^{\mathsf{RES}_{2^k}(f)}(v_i, v_{i+1}) = 0 \Leftrightarrow \vec{u}^{\mathsf{RES}(f)}(v_i, v_{i+1}) < 2^k$.

Algorithm 2 Ford-Fulkerson capacity scaling algorithm (FF-CSA)

Input: A digraph $G = (V, E, o)$ with integral capacities l and u and one special edge $\{t, s\}$ directed from t to s.
Output: A feasible flow f on G with maximal flow on $\{t, s\}$.

$f \leftarrow 0$.
// We have $f \in P_{\text{feas}}(G)$.
$U \leftarrow \max\{\vec{u}(v, w) \mid v, w \in V\}$.
for $k = \lceil \log(U) \rceil$ down to 0 **do**
    `rounddone` $\leftarrow$ **false**.
    while not `rounddone` **do**
        Construct $\mathsf{RES}_{2^k}(f)$.
        if there is a well-directed cycle c in $\mathsf{RES}_{2^k}(f)$ with $\delta(c) > 0$ **then**
            Augment 2^k units over c: $f \leftarrow f + 2^k c$.
            // Lemma 5.19 ensures that $f \in P_{\text{feas}}(G)$. Moreover, f is 2^k-integral.
        else
            `rounddone` $\leftarrow$ **true**.
        end if
    end while
end for
// Lemma 5.20 ensures that f maximizes δ in $P_{\text{feas}}(G)$.
return f.

Lemma 5.22. *On any input $(G, l, u, \{t, s\})$ the FF-CSA uses only polynomial time.*

Proof. There are subalgorithms for finding shortest well-directed cycles c in $\mathsf{RES}_{2^k}(f)$ with $\delta(c) > 0$ in polynomial time. We call each iteration of the for-loop in line 4 a *round*. As the edge capacities are encoded in the input, the number of rounds is linear in the input size. We show that the while-loop in line 6 runs at most $2|E| + 1$ times each round:

Each iteration of the while-loop except the last one (where `rounddone` is set to true) increases $\delta(f)$ by 2^k.

For the initial solution $f = 0$ we have $\delta(f) = 0$. We have $\delta(f) \leq u_{\{t,s\}}$ for all $f \in P_{\text{feas}}(G)$. But $u_{\{t,s\}} \leq U$. So in the first round ($k = \lceil \log(U) \rceil$), there is at most 1 iteration of the while-loop.

Let $f_{\max} \in P_{\text{feas}}(G)$ such that $f_{\max}$ maximizes δ in $P_{\text{feas}}(G)$. Let f be a flow after finishing the while-loop with a fixed k. Let $d := f_{\max} - f$. We want to show that

$$\delta(f_{\max}) - \delta(f) = \delta(d) < 2^k |E|.$$

After finishing the while-loop, there are no well-directed cycles c on $\mathsf{RES}_{2^k}(f)$ with $\delta(c) > 0$. Therefore each cycle $c = (v_1, \dots, v_\ell, v_{\ell+1} = v_1)$ on $\mathsf{RES}_{2^k}(f)$ with $\delta(c) > 0$ must use a critical tuple. According to Lemma 5.11 the flow d can be decomposed into at most $|E|$ well-directed cycles on $\mathsf{RES}(f)$:

$$\sum_{i=1}^{|E|} \alpha_i c_i = d,$$

where $c_1, \dots, c_{|E|}$ are well-directed cycles on $\mathsf{RES}(f)$ and $\alpha_1, \dots, \alpha_m \in \mathbb{R}_{>0}$ with $\sum_{i=1}^{m} \alpha_i c_i = d$ such that for all $1 \leq i \leq |E|$ and for all edges $e \in c_i$ we have $\operatorname{sgn}(c_i(e)) = \operatorname{sgn}(d(e))$. Each one of these cycles c_i has $\delta(c_i) \leq 0$ or uses a critical tuple (v, w), i.e. $\vec{u}^{\mathsf{RES}(f)}(v, w) < 2^k$. Since d is a feasible flow on $\mathsf{RES}(f)$, we have $\alpha_i < 2^k$ for all $1 \leq i \leq |E|$ that satisfy $\delta(c_i) > 0$. Therefore $\delta(d) < 2^k |E|$.

In the next round in each iteration of the while-loop besides the last one, $\delta(f)$ is augmented by 2^{k-1}. Thus the while-loop only runs for at most $2|E| + 1$ iterations in each round. $\square$

Chapter 6

Deciding positivity of LR-coefficients

In this chapter, we will design a combinatorial algorithm to decide the positivity of Littlewood-Richardson coefficients. These coefficients have several different combinatorial interpretations. Valuable work has been done by Pak and Vallejo (cf. [PV05]) by describing three major approaches and analyzing their correlation. The most widely known interpretation can be given with the so-called Littlewood-Richardson rule in terms of Littlewood-Richardson tableaux. The other two interpretations are the Berenstein-Zelevinsky triangles (cf. [BZ92]) and the Knutson-Tao hives (cf. [KT99]). [PV05] give explicit bijections between them.

The idea of this chapter is to use the language of hives and transform the problem of deciding positivity of Littlewood-Richardson coefficients into an optimization problem and solve it with a Ford-Fulkerson-like algorithm. We will see that for our problem we can design a residual network in which the so-called hive inequalities are transformed into capacity constraints. On this residual network *shortest* well-directed cycles can be used for augmenting the flow by an integral amount.

We start with a motivation in Section 6.1 and continue with ideas and definitions in Section 6.2, then we introduce the basic algorithm in Section 6.4. In Section 6.5 we discuss an algorithm that decides whether a Littlewood-Richardson coefficient is exactly 1 and we give a proof of a conjecture by Fulton. In Chapter 7, we will refine the LRPA to become a polynomial-time algorithm.

6.1 Saturation Conjecture and hive polytopes

Additionally to proving the #**P**-completeness of computing Kostka numbers, Narayanan proved that the computation of Littlewood-Richardson coefficients LRCoeff is #**P**-complete (cf. [Nar06]). This is interesting, because the associ-

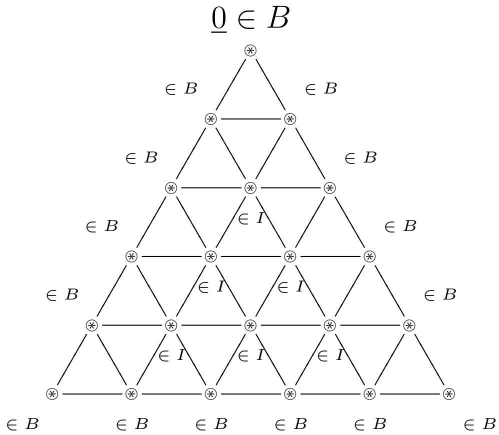

Figure 6.1: The big triangle graph Δ.

ated decision problem LRCoeff$_{>0}$ is decidable in polynomial time, which was first pointed out by Mulmuley and Sohoni (cf. [MS05]). We remark that assuming $\mathbf{P} \neq \mathbf{NP}$, LRCoeff is not $\#\mathbf{P}$-complete under parsimonious reductions (see Corollary 2.11). There are several ways to prove that LRCoeff$_{>0} \in \mathbf{P}$, each using linear optimization algorithms and the following so-called Saturation Conjecture, which was proved by Knutson and Tao (cf. [KT99]):

Theorem 6.1 (Saturation Conjecture)**.** *Let λ, μ, ν be partitions, $N \in \mathbb{N}_{\geq 1}$. Then*

$$c_{\lambda\mu}^{\nu} > 0 \Longleftrightarrow c_{N\lambda N\mu}^{N\nu} > 0.$$

Buch gives a proof based solely on the hive model (cf. [Buc00]). We do not use the Saturation Conjecture for deciding LRCoeff$_{>0}$. Instead we do it the other way round: We will use the hive model to give a *combinatorial* algorithm for deciding LRCoeff$_{>0}$. As a byproduct we obtain a proof of the Saturation Conjecture.

For our approach we now introduce notations that lead to the definition of the hive polytope. Given partitions λ, μ, ν such that $|\nu| = |\lambda| + |\mu|$, it is easy to see that for $\ell(\nu) < \max\{\ell(\mu), \ell(\lambda)\}$ we have $c_{\lambda\mu}^{\nu} = 0$, because $c_{\lambda\mu}^{\nu}$ equals the number of semistandard Young tableaux with shape ν/λ and type μ whose reverse reading word is a lattice permutation. So we can assume that $\max\{\ell(\lambda), \ell(\mu), \ell(\nu)\} = \ell(\nu)$. Let $n := \ell(\nu)$.

We start with a triangular array of vertices, $n+1$ on each side, as seen in Figure 6.1. This graph is called the *big triangle graph* Δ with vertex set H. To avoid confusion with vertices in other graphs that will be introduced later, vertices in Δ are denoted by underlined capital letters ($\underline{A}$, $\underline{B}$, etc.). The vertices on the border of the big triangle graph form the set B. The inner vertices form

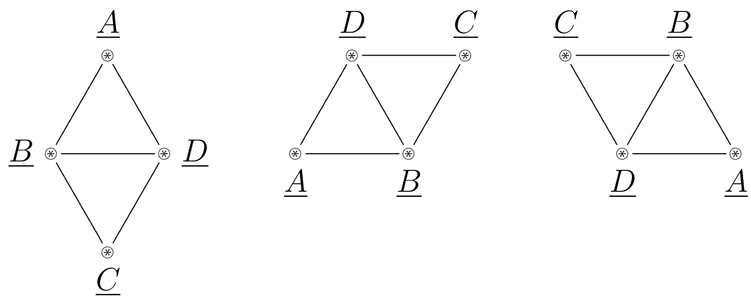

Figure 6.2: Rhombus labelings in all possible ways.

the set $I := H \setminus B$. Denote with $\underline{0}$ the top vertex of H and set $H' := H \setminus \{\underline{0}\}$. The graph Δ is subdivided into $(n(n+1))/2 + (n(n-1))/2 = n^2$ small triangles whose corners are graph vertices. We call a triangle in Δ an *upright* triangle, if it is of the form '$\triangle$'. Otherwise ('$\bigtriangledown$') we call the triangle an *upside down* triangle. By a *rhombus* $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ with $\underline{A}, \underline{B}, \underline{C}, \underline{D} \in H$ we mean the union of two small triangles next to each other, where $\underline{A}$ is the acute vertex of the upright triangle and $\underline{B}$, $\underline{C}$ and $\underline{D}$ are the other vertices in counterclockwise direction (see Figure 6.2). If we do not want to assign a name to a vertex of the rhombus, we use a syntax like $\Diamond(\underline{A}, \underline{B}, ., \underline{D})$. Two rhombi are called *overlapping*, if they share exactly one triangle.

Each rhombus induces a so-called *hive inequality* on the vector space of real vertex labelings $\mathbb{R}^H$: The sum of the labels at the obtuse vertices must be greater than or equal to the sum of the labels at the acute vertices. So for a rhombus $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ and a vertex labeling $h \in \mathbb{R}^H$ we require

$$h(\underline{B}) + h(\underline{D}) \geq h(\underline{A}) + h(\underline{C}). \tag{6.1}$$

We call such a rhombus h-flat, if

$$h(\underline{B}) + h(\underline{D}) = h(\underline{A}) + h(\underline{C}), \tag{6.2}$$

or simply *flat*, if it is clear what h is meant. We define the *slack* of a rhombus as

$$\sigma\big(\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D}), h\big) := \big(h(\underline{B}) + h(\underline{D})\big) - \big(h(\underline{A}) + h(\underline{C})\big).$$

It is clear that a rhombus $\Diamond$ is h-flat iff $\sigma\big(\Diamond, h\big) = 0$.

If a vertex labeling $h \in \mathbb{R}^H$ satisfies all rhombus inequalities, h is called a *hive*. The sum of two hives is again a hive. The difference of two hives is not necessarily a hive.

As the vertex set H is embedded into the plane, h can be interpreted as heights of the points H in $\mathbb{R}^3$:

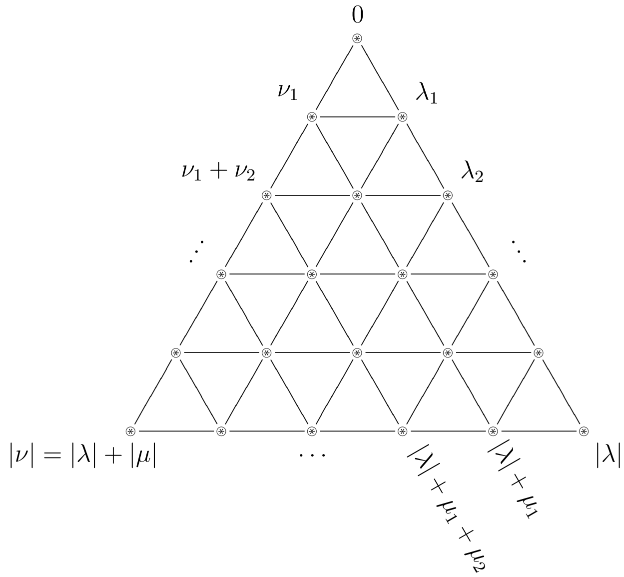

Figure 6.3: Border labelings of Δ resulting from λ, μ and ν.

Definition 6.2 (Hill function)**.** The convex hull $\mathrm{conv}(H)$ in the plane can be interpreted as the domain of $h' : \mathrm{conv}(H) \to \mathbb{R}$ where h' is induced by h via linear interpolation and thus $h'|_H = h$. We call h' the *hill function of* h. ■

It is essential that if h is a hive, then h' is a concave function.

A hive $h \in \mathbb{Z}^H$ is called an *integral hive*. Given partitions λ, μ and ν with $|\nu| = |\lambda| + |\mu|$. Let $b(\lambda, \mu, \nu) \in \mathbb{R}^B$ be a border with labels as in Figure 6.3.

Theorem 6.3 (cf. [KT99], [Buc00])**.** *Given partitions λ, μ, ν with $|\nu| = |\lambda| + |\mu|$. Then $c^{\nu}_{\lambda\mu}$ is the number of integral hives with border labels $b(\lambda, \mu, \nu)$.*

We remark that Theorem 6.3 can be derived from the Littlewood-Richardson rule (cf. [Buc00, PV05]).

The rhombus inequalities and the border labels can be encoded in a matrix A_n over $\{-1, 0, 1\}$ and a vector $b_{\lambda,\mu,\nu}$ over $\mathbb{N}$ such that the Littlewood-Richardson coefficient can be written as

$$c^{\nu}_{\lambda\mu} = \left|\left\{x \in \mathbb{Z}^{|H|} \mid A_n x \leq b_{\lambda,\mu,\nu}\right\}\right|.$$

Thus LRCoeff becomes a subproblem of #IP (see Section 2.2), namely

$$\text{LRCoeff} = \left\{(A_n, b_{\lambda,\mu,\nu}) \mapsto \left|\left\{x \in \mathbb{Z}^{|H|} \mid A_n x \leq b_{\lambda,\mu,\nu}\right\}\right|\right\}.$$

The associated polytope

$$P(\lambda, \mu, \nu) := P(A_n, b_{\lambda,\mu,\nu}) =$$
$$\{h \in \mathbb{R}^H \mid h|_B = b(\lambda, \mu, \nu),\ \forall\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D}) : h(\underline{B}) + h(\underline{D}) \geq h(\underline{A}) + h(\underline{C})\}$$

is denoted the *hive polytope* corresponding to λ, μ, ν.

Now [MS05] use the Saturation Conjecture (Theorem 6.1) and Theorem 6.3 to decide LRCOEFF$_{>0}$ in the following way.

Corollary 6.4.

$$P(\lambda, \mu, \nu) \neq \emptyset \Longleftrightarrow c^{\nu}_{\lambda\mu} > 0.$$

Proof. The direction " $\Leftarrow$ " is clear. Let $x \in \mathbb{Q}^m$ with $A_n x \leq b_{\lambda,\mu,\nu}$. Let $N \in \mathbb{N}$ with $Nx \in \mathbb{Z}^m$. Then $A_n(Nx) \leq Nb_{\lambda,\mu,\nu}$. From the definition of $b_{\lambda,\mu,\nu}$ we get $Nb_{\lambda,\mu,\nu} = b_{N\lambda,N\mu,N\nu}$. Hence $Nx \in P(N\lambda, N\mu, N\nu)$. So $c^{N\nu}_{N\lambda N\mu} > 0$ and with the Saturation Conjecture we get $c^{\nu}_{\lambda\mu} > 0$. □

Deciding whether a polyhedron $P(A, b)$ is empty can be done in polynomial time (see Section 2.1). Hence LRCOEFF$_{>0} \in \mathbf{P}$.

Purely combinatorial algorithms There are other problems where standard methods lead to polynomial-time algorithms. For example the maximum flow problem (see Definition 5.12) can be solved in polynomial time using ellipsoid method or interior point methods. But these standard methods are not as fast as algorithms that use the specific problem structure and operate directly on the graph, like the Ford-Fulkerson algorithm (see Section 5.6, also described in [AMO93, CLRS01]). We call algorithms of this kind (in the sense that they do not use an explicit linear programming algorithm) purely combinatorial algorithms. Such algorithms often have better runtime behaviour than the general methods in theory as well as in practice.

We design a purely combinatorial algorithm for LRCOEFF$_{>0}$ in Section 6.4 which we call the LRPA (Littlewood-Richardson Positivity Algorithm). In Chapter 7 we refine it into its polynomial-time counterpart LRP-CSA using a scaling approach. Its worst-case runtime behaviour is not as good as one might hope for. It is planned in the near future to implement the LRP-CSA and compare its running time with other methods that determine the positivity of Littlewood-Richardson coefficients. The worst-case analysis of LRP-CSA reveals interesting problems that are to deal with, which makes the algorithm an interesting result on its own. We quote [MS05] here:

> It is of interest to know if there is a purely combinatorial algorithm for this problem that does not use linear programming; i.e., one similar to the max-flow or weighted matching problems in combinatorial optimization. [...] It is reasonable to conjecture that there is a polynomial time algorithm that provides an integral proof of positivity of $c^{\nu}_{\lambda\mu}$ in the form of an integral point in P.

This is exactly what the LRP-CSA does, as there are bijections between the integer points in the quoted P and the integer points in the hive polytope (cf. [PV05]).

6.2 Hives and flows

In this section we want to transfer the problem of finding an integral hive into the language of flows and convert it into an optimization problem like the maximum flow problem.

As seen, partitions λ, μ, ν induce an integral vertex labeling $b := b(\lambda, \mu, \nu) \in \mathbb{R}^B$ on the border vertices of Δ (cf. Figure 6.3). This vertex labeling $b \in \mathbb{R}^B$ is called the *target border*. A border $b \in \mathbb{R}^B$ is called *regular*, if for all border vertices $\underline{A}, \underline{B}, \underline{C} \in B$ which are consecutive vertices in clockwise or counterclockwise direction on the same side of the big hive triangle, we have that

$$b(\underline{B}) - b(\underline{A}) > b(\underline{C}) - b(\underline{B}).$$

Note that $b(\underline{B}) - b(\underline{A}) > b(\underline{C}) - b(\underline{B}) \Leftrightarrow b(\underline{B}) - b(\underline{C}) > b(\underline{A}) - b(\underline{B})$ and thus is sufficient to look at the case where $\underline{A}, \underline{B}, \underline{C}$ are consecutive border vertices in clockwise direction. If λ, μ and ν are strictly decreasing partitions, then the target border $b(\lambda, \mu, \nu)$ is regular.

For $z \in \mathbb{R}$ we call a real number z-integral, if it is an integral multiple of z. We say $h \in \mathbb{R}^H$ is z-integral, if $h(\underline{A})$ is z-integral for all $\underline{A} \in H$.

6.2.1 The graph structure

Definition 6.5 (Throughput)**.** For a flow f on a digraph G, we define for each vertex v the *throughput* $\delta(v, f)$ as

$$\delta(v, f) := \delta_{\text{in}}(v, f) = \delta_{\text{out}}(v, f).$$

■

Note that this definition depends on the edge directions of G. All vertices that only have incident edges directed towards them or only have edges directed from them have throughput 0. For each vertex v that has exactly one edge e_1 directed towards v and one edge e_2 directed from v, we have $\delta(v, f) = f(e_1) = f(e_2)$.

We now define a bipartite planar digraph $G = (V, E, o)$, which is homeomorphic to the dual graph of Δ. The definition is similar to the definition in [Buc00]: G has one fat black vertex in the middle of each small triangle of Δ. In addition there is one circle vertex on every triangle side (see Figure 6.4). We denote a circle vertex between two upright triangle vertices $\underline{A}$ and $\underline{B}$ (read in counterclockwise direction) as $[\underline{A}, \underline{B}]$. Note that every circle vertex lies between two upright triangle vertices. Each fat black vertex is adjacent to the three circle vertices on the sides of its triangle. There is an additional fat black vertex o with edges from o to all circle vertices that lie on the border of the big triangle. The graph G is embedded in the plane in a way such that $\underline{0} \in H$ lies in the outer face. Note that G is essentially the dual graph of Δ with circle vertices added on each edge.

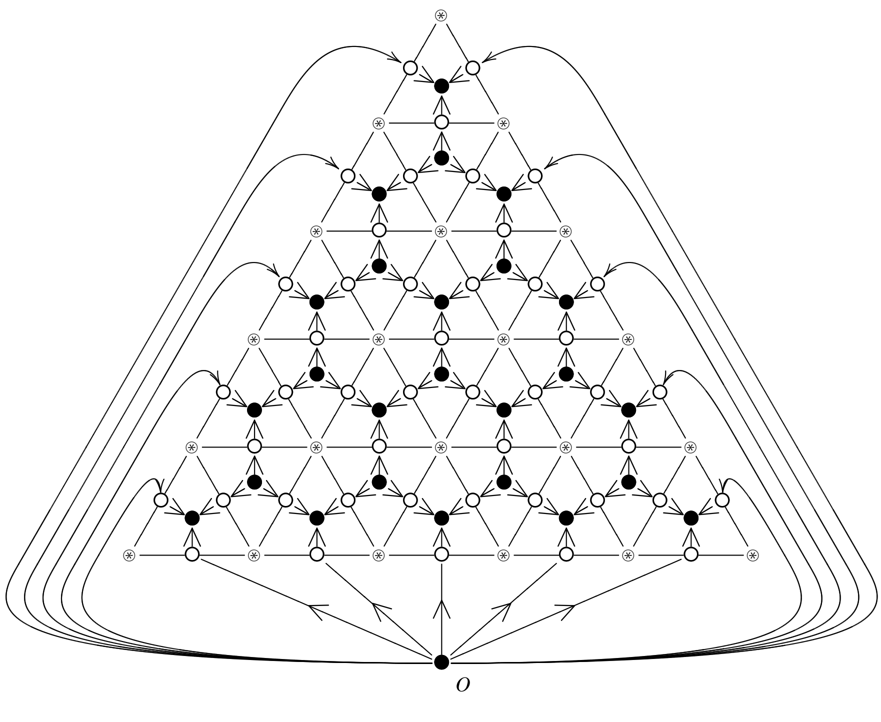

Figure 6.4: The digraph G and graph Δ.

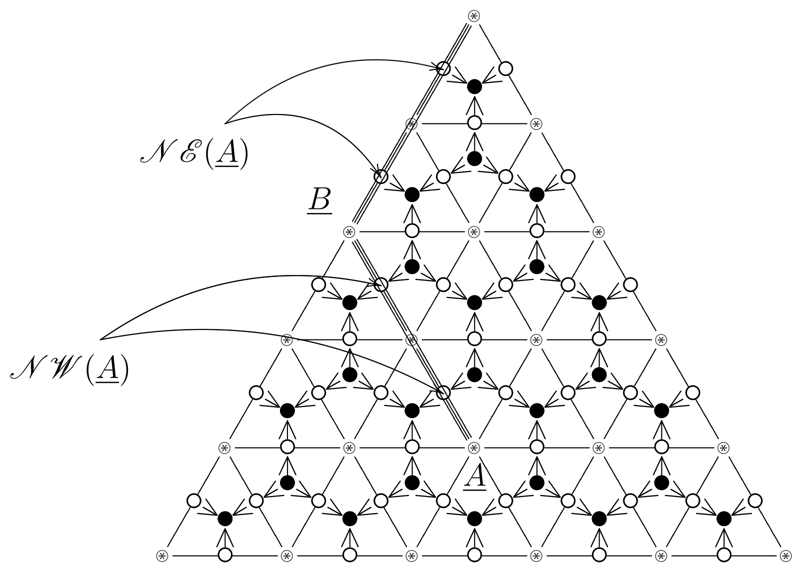

Figure 6.5: The sets $\mathscr{N}\mathscr{W}(\underline{A})$ and $\mathscr{N}\mathscr{E}(\underline{A})$. The vertex o is omitted here as in all following pictures as well.

Next we assign a direction to each edge in G (see Figure 6.4): The edges incident to o are directed from o towards the border of the big triangle graph. The edges in an upright triangle are directed towards the incident fat black vertex, while the edges in an upside down triangle are directed towards the incident circle vertex.

Winding numbers Let $\underline{A} \in H$. Then define $\mathscr{N}\mathscr{W}(\underline{A})$ to be the set of circle vertices in V that lie on the northwest diagonal drawn from $\underline{A}$ (see Figure 6.5). This diagonal hits a border vertex $\underline{B} \in B$. Define $\mathscr{N}\mathscr{E}(\underline{A})$ to be the set of circle vertices in V that lie on the northeast diagonal drawn from that border vertex $\underline{B}$ (see also Figure 6.5). Now define the *winding number* of a vertex $\underline{A} \in H$ with respect to a flow $f \in F$ as

$$\operatorname{wind}(\underline{A}, f) = \sum_{v \in \mathscr{N}\mathscr{W}(\underline{A})} \delta(v, f) - \sum_{v \in \mathscr{N}\mathscr{E}(\underline{A})} \delta(v, f).$$

The winding number is linear in the flow f.

Lemma 6.6. *For each $\underline{A} \in H$, $f \in F$, we have*

$$|\mathit{wind}(\underline{A}, f)| \leq n \cdot \max_{v \in V}\{|\delta(v, f)|\}.$$

Proof. Let $\underline{A} \in H$, $f \in F$. We have $|\mathscr{N}\mathscr{W}(\underline{A})| + |\mathscr{N}\mathscr{E}(\underline{A})| \leq n$. This proves the lemma. $\square$

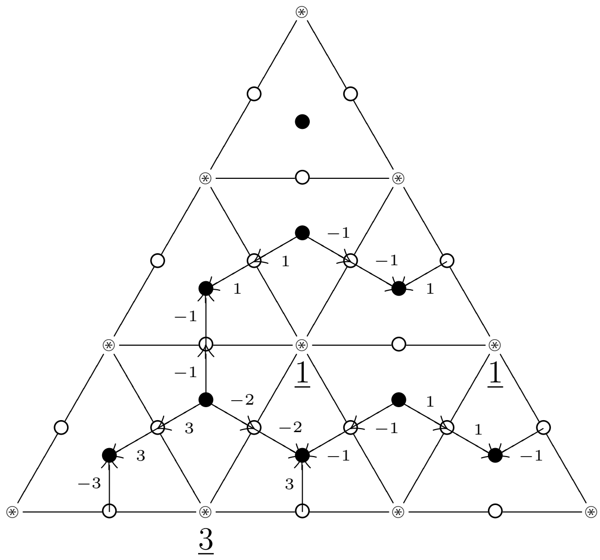

Figure 6.6: The isomorphism η. Here $\eta(h)$ is depicted where $h(\underline{A}) \neq 0$ for only three vertices $\underline{A}$. The figure only shows edges that have nonzero flow value in $\eta(h)$.

Flow vector space Let F denote the vector space $F(G)$ of flows on G. As G is connected, by Lemma 5.4 we have $\dim F = |E| - |V| + 1$. Note that a flow f on G is completely defined by its throughput $\delta([\underline{A}, \underline{B}], f)$ on each circle vertex $[\underline{A}, \underline{B}]$.

Theorem 6.7 (Vector space isomorphism)**.** *There is an explicit isomorphism* $\eta : \mathbb{R}^{H'} \to F$ *between the real vector space* $\mathbb{R}^{H'}$ *of vertex labels in* Δ *in which the top vertex* $\underline{0}$ *has value 0 and the real vector space* F *of flows on* G*: For* $h \in \mathbb{R}^{H'}$ *and each circle vertex* $[\underline{A}, \underline{B}]$*, set* $\delta([\underline{A}, \underline{B}], \eta(h)) := h(\underline{A}) - h(\underline{B})$*, which completely defines* $\eta(h)$*. The winding numbers give* η^{-1} *by* $\eta^{-1}(f)(\underline{A}) = wind(\underline{A}, f)$ *for* $f \in F$*.*

The isomorphism η is illustrated in Figure 6.6. Note that an integral hive h results in an integral flow $\eta(h)$ and that an integral flow f induces integral winding numbers and thus $\eta^{-1}(f)$ is integral. So η preserves integrality in both directions.

Also note that via η, all linear functions $H' \to \mathbb{R}$ can be converted to linear functions $F \to \mathbb{R}$.

We remark that the proof of Theorem 6.7 does not make use of the special problem structure and therefore this theorem can be generalized to any connected graph.

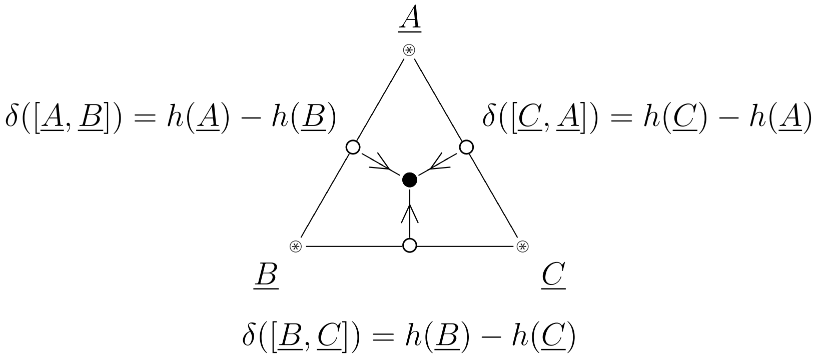

Figure 6.7: Illustration of η.

Proof of Theorem 6.7. Let $h \in \mathbb{R}^{H'}$. Define a flow $\eta(h) := f$ as described in Theorem 6.7 (see Figure 6.7). As for each circle vertex v there is exactly one edge e_1 directed towards v and exactly one edge e_2 directed from v, the flow on e_1 and e_2 is defined as $f(e_1) := f(e_2) := \delta([\underline{A},\underline{B}], f)$. This completely defines f on all edges of G as each edge in G is incident to exactly one circle vertex. It is easy to see that η is linear.

We show that $f \in F$:

The flow constraints are satisfied by definition on each circle vertex. For an upright triangle formed by $\underline{A}, \underline{B}, \underline{C}$ in counterclockwise direction, all the 3 edges that are both incident to the fat black vertex v in the center of the triangle and incident to $[\underline{A},\underline{B}], [\underline{B},\underline{C}]$ or $[\underline{C},\underline{A}]$ are directed towards v. So $\delta_{\text{in}}(v, f) = f(\{[\underline{A},\underline{B}], v\}) + f(\{[\underline{B},\underline{C}], v\}) + f(\{[\underline{C},\underline{A}], v\}) = h(\underline{A}) - h(\underline{B}) + h(\underline{B}) - h(\underline{C}) + h(\underline{C}) - h(\underline{A}) = 0$ and analogously $\delta_{\text{out}}(v, f) = 0$. For an upside down triangle the argument is similar. As the flow constraints are satisfied in all but one vertex o, the flow constraints must be satisfied in all vertices (see proof of Lemma 5.4). So f is a flow on G.

We show that $\mathbb{R}^{H'}$ and F have the same $\mathbb{R}$-dimension:

The number of faces of G equals $|H|$. As G is a connected planar graph, Euler's formula for planar graphs states that $|V| - |E| + |H| = 2$. So $\dim F = |E| - |V| + 1 = |H| - 1 = \dim R^{H'}$.

We show that η is an isomorphism:

With the rank-nullity theorem it only remains to show that η is injective. Let $h \in \mathbb{R}^{H'}$ with $\eta(h) = 0$. This means that for any two adjacent vertices $\underline{A} \in H$ and $\underline{B} \in H$ we have $h(\underline{A}) - h(\underline{B}) = 0$ and therefore $h(\underline{A}) = h(\underline{B})$. As $h(\underline{0}) = 0$ and G is connected, it follows that $h = 0$. Therefore η is injective.

We show how to compute η^{-1}:

Consider a standard basis vector h of $\mathbb{R}^{H'}$: Let $\underline{A} \in H'$ and $h \in \mathbb{R}^{H'}$ with $h(\underline{A}) = 1$ and $h(\underline{B}) = 0$ for all $\underline{B} \in H'$, $\underline{B} \neq \underline{A}$. Then it is easy to see that

$\mathrm{wind}(\underline{A}, \eta(h)) = 1$ and for all $\underline{A} \neq \underline{B}$ we have $\mathrm{wind}(\underline{B}, \eta(h)) = 0$. As for all flows of basis vectors the winding numbers give the vertex labels h and the winding number is linear in the flow, the winding numbers are a way to compute η^{-1}. $\square$

Hive inequalities on flows As η is an isomorphism, we can identify a flow $f \in F$ with its vertex labeling $\eta^{-1}(f) \in \mathbb{R}^{H'}$. For example we can now speak of f-flat rhombi. If for two flows $f, g \in F$ the induced hives have the same border, i.e. $\eta^{-1}(f)|_B = \eta^{-1}(g)|_B$, then we write $f|_B = g|_B$. As η is an isomorphism of vector spaces, the linear hive inequalities (6.1) can also be expressed as linear inequalities in F. Given a rhombus $\lozenge(\underline{A}, \underline{B}, \underline{C}, \underline{D})$. Let h be a hive and $f = \eta(h)$. Then

$$\begin{aligned} h(\underline{A}) + h(\underline{C}) \leq h(\underline{B}) + h(\underline{D}) &\Leftrightarrow (h(\underline{A}) - h(\underline{B})) \leq (h(\underline{D}) - h(\underline{C})) \\ &\Leftrightarrow \delta([\underline{A}, \underline{B}], f) \leq \delta([\underline{D}, \underline{C}], f), \end{aligned} \tag{6.3}$$

which is a restriction on the throughputs of circle vertices of this rhombus. This is equivalent to

$$\begin{aligned} h(\underline{A}) + h(\underline{C}) \leq h(\underline{B}) + h(\underline{D}) &\Leftrightarrow -(h(\underline{D}) - h(\underline{A})) \leq -(h(\underline{C}) - h(\underline{B})) \\ &\Leftrightarrow \delta([\underline{C}, \underline{B}], f) \leq \delta([\underline{D}, \underline{A}], f). \end{aligned} \tag{6.4}$$

We call a flow f a *hive flow*, if $\eta^{-1}(f)$ is a hive. We note that f is a hive flow, if for all rhombi $\lozenge(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ we have $\delta([\underline{A}, \underline{B}], f) \leq \delta([\underline{D}, \underline{C}], f)$. We can now express the slack of a rhombus as

$$\begin{aligned} \sigma\big(\lozenge(\underline{A}, \underline{B}, \underline{C}, \underline{D}), f\big) := \sigma\big(\lozenge(\underline{A}, \underline{B}, \underline{C}, \underline{D}), h\big) &= \delta([\underline{D}, \underline{C}], f) - \delta([\underline{A}, \underline{B}], f) \\ &= \delta([\underline{D}, \underline{A}], f) - \delta([\underline{C}, \underline{B}], f). \end{aligned}$$

6.2.2 Sources, sinks and b-boundedness

In this section we introduce the optimization problem to be solved for deciding whether a Littlewood-Richardson coefficient is positive.

Define the set $\mathscr{S} \subset V$ of *source vertices* as the set of all circle border vertices in G at the right or bottom border of the big triangle. Define the set $\mathscr{T} \subset V$ of *sink vertices* as the set of all circle border vertices in G at the left border of the big triangle. Note that for any flow $f \in F$, we have $\sum_{s \in \mathscr{S}} \delta(s, f) + \sum_{t \in \mathscr{T}} \delta(t, f) = 0$. The *throughput* $\delta(f)$ *of a flow* f on G is defined as

$$\delta(f) := \sum_{s \in \mathscr{S}} \delta(s, f) - \sum_{t \in \mathscr{T}} \delta(t, f) = 2 \sum_{s \in \mathscr{S}} \delta(s, f).$$

For all but three border vertices v we define the predecessor $\mathrm{pred}(v)$ as follows: For a vertex on the right border, it is its topleft neighbor border vertex. For a vertex on the bottom border, it is its right neighbor border vertex. For a vertex on the left border, it is its topright neighbor border vertex. We define the successor as $\mathrm{succ}(\mathrm{pred}(v)) := v$.

Now we put additional constraints on hive flows: Let $b \in \mathbb{R}^B$ be a border vertex labeling. We define the following bounds on the border vertices:

$$\begin{aligned} \forall [\underline{A}, \underline{B}] \in \mathscr{S} : \delta^b_{\max}([\underline{A}, \underline{B}]) &:= b(\underline{A}) - b(\underline{B}), \\ \forall [\underline{A}, \underline{B}] \in \mathscr{T} : \delta^b_{\min}([\underline{A}, \underline{B}]) &:= b(\underline{A}) - b(\underline{B}). \end{aligned}$$

Let $f \in F$ be a flow on G. We call f *b-bounded*, if it satisfies

$$\begin{aligned} \forall [\underline{A}, \underline{B}] \in \mathscr{S} : \delta([\underline{A}, \underline{B}], f) &\le \delta^b_{\max}([\underline{A}, \underline{B}]), \\ \forall [\underline{A}, \underline{B}] \in \mathscr{T} : \delta([\underline{A}, \underline{B}], f) &\ge \delta^b_{\min}([\underline{A}, \underline{B}]). \end{aligned} \tag{6.5}$$

These inequalities (6.5) together with the hive inequalities (6.3) on the flow vector space $F(G)$ define the *polyhedron $P^b \subseteq F(G)$ of all b-bounded hive flows.* The following lemma shows the significance of P^b:

Lemma 6.8. *Let the border $b = b(\lambda, \mu, \nu)$ come from partitions λ, μ and ν with $|\nu| = |\lambda| + |\mu|$. Then the following statements hold:*

(1) $\forall s \in \mathscr{S} : \delta^b_{\max}(s) \ge 0$ and $\forall t \in \mathscr{T} : \delta^b_{\min}(t) \le 0$.

(2) For any b-bounded flow f we have $\delta(f) \le 2|\nu|$.

(3) Let f be a b-bounded flow. $\delta(f) = 2|\nu|$ iff f satisfies all $3n$ inequalities in (6.5) with equality.

(4) A hive with border b exists iff $\max\{\delta(f) \mid f \in P^b\} = 2|\nu|$.

(5) If $\max\{\delta(f) \mid f \in P^b\} < 2|\nu|$, then $c^\nu_{\lambda\mu} = 0$.

(6) If there exists an integral flow $f \in P^b$ with $\delta(f) = 2|\nu|$, then $c^\nu_{\lambda\mu} > 0$.

(7) $\delta^b_{\max}(succ(s)) \le \delta^b_{\max}(s)$ for all source vertices $s \in \mathscr{S}$ that have a successor and $\delta^b_{\min}(succ(t)) \ge \delta^b_{\min}(t)$ for all sink vertices $t \in \mathscr{T}$ that have a successor.

Proof. (1) The first statement holds, because b comes from partitions as seen in Figure 6.3.

(2) The second statement is a result of a simple calculation using cancellation of telescoping sums: $\delta(f) = \sum_{s \in \mathscr{S}} \delta(s, f) - \sum_{t \in \mathscr{T}} \delta(t, f) \le (|\lambda| + |\mu| - 0) - (0 - |\nu|) = 2|\nu|$.

(3) If all $3n$ inequalities in (6.5) are satisfied with equality, then again by using cancellation of telescoping sums we get $\delta(f) = 2|\nu|$. On the other hand, as $\delta(f) \le 2|\nu|$, we can only get equality, if all summands in $\delta(f) = \sum_{s \in \mathscr{S}} \delta(s, f) - \sum_{t \in \mathscr{T}} \delta(t, f)$ are maximized, which means that the inequalities in (6.5) are satisfied with equality.

(4) If $\max\{\delta(f) \mid f \in P^b\} = 2|\nu|$, then with (3) we get a flow f that has all $3n$ inequalities in (6.5) satisfied with equality. Via η^{-1} a hive with border b can be created.

On the other hand, a hive with border b induces $f \in P^b$ that satisfies all $3n$ inequalities in (6.5) with equality. With (3) it follows that $\delta(f) = 2|\nu|$. With (2) we have $\max\{\delta(f) \mid f \in P^b\} = 2|\nu|$.

(5) If $\max\{\delta(f) \mid f \in P^b\} < 2|\nu|$, then according to (4) no hive with border b exists. In particular, no integral hive with border b exists and thus according to Theorem 6.3 we have $c^{\nu}_{\lambda\mu} = 0$.

(6) If there exists an integral flow $f \in P^b$ with $\delta(f) = 2|\nu|$, then with (3) we get that f has all $3n$ inequalities in (6.5) satisfied with equality. Via η^{-1} we get an integral hive with border b. Theorem 6.3 shows that $c^{\nu}_{\lambda\mu} > 0$.

(7) Let $[\underline{B}, \underline{C}] \in \mathscr{S}$ be a source vertex and $[\underline{A}, \underline{B}] := \mathrm{succ}([\underline{B}, \underline{C}])$ its successor. W.l.o.g. $\delta^b_{\max}([\underline{B}, \underline{C}]) = b(\underline{B}) - b(\underline{C}) = \lambda_i$ for some i and $\delta^b_{\max}([\underline{A}, \underline{B}]) = b(\underline{A}) - b(\underline{B}) = \lambda_{i+1}$. As λ is a partition, we have $\delta^b_{\max}([\underline{A}, \underline{B}]) \leq \delta^b_{\max}([\underline{B}, \underline{C}])$. An analog proof can be applied to $\mathscr{T}$.

$\square$

6.3 Comments on two-commodity flow

The problem of deciding positivity of Littlewood-Richardson coefficients has a natural description as a so-called *homologous flow problem*. Itai (cf. [Ita78]) proved that solving a homologous flow problem is equivalent to solving a corresponding two-commodity flow problem with only linear loss of time. He also proved that solving this is polynomially equivalent to solving linear programs, for which no purely combinatorial algorithm is known. Thus for our goal of designing a combinatorial algorithm, we may not rely on the homologous flow description or the two-commodity flow description. Nevertheless we describe the approach here, because it might be a competitive way for deciding positivity of Littlewood-Richardson coefficients. Fast interior point methods designed for solving multicommodity flow problems as for example in [KP95] can be used to solve the problem efficiently in polynomial time.

A *homologous flow problem* is a maximum flow problem with additional constraints:

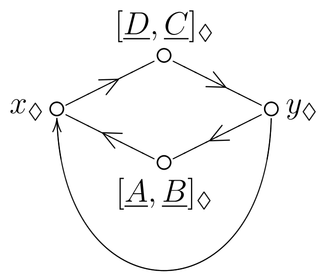

Figure 6.8: The gadget for the homologous flow description.

Definition 6.9 (Homologous flow problem)**.** Given a digraph $G = (V, E, o)$ with integral capacities $u_e \in \mathbb{Z}_{\geq 0}, l_e \in \mathbb{Z}_{\leq 0}$ on each edge e with one special edge $\{t, s\}$ directed from t towards s, a natural number $N \in \mathbb{N}$ and sets $E_1, \ldots, E_M \subseteq E$, the *homologous flow problem* is the problem of deciding whether a feasible flow f on G exists with $f\big(\{t, s\}\big) \geq N$ that satisfies the following constraints:

$$\forall 1 \leq i \leq M : \text{ if } e_1, e_2 \in E_i \text{ then } f(e_1) = f(e_2).$$

The sets E_i are called *homologous sets.* ■

If we use the Saturation Conjecture and Lemma 6.8(4), we can strengthen Lemma 6.8(6) to

(6′) If there exists a flow $f \in P^b$ with $\delta(f) = 2|\nu|$, then $c^\nu_{\lambda\mu} > 0$.

Then for deciding positivity of Littlewood-Richardson coefficients we need not care about integrality any more. We set $N := 2|\nu|$, start with the graph G and for each rhombus $\Diamond := \Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ we add the following gadget (see Figure 6.8) containing four vertices $x_\Diamond$, $y_\Diamond$, $[\underline{A}, \underline{B}]_\Diamond$ and $[\underline{D}, \underline{C}]_\Diamond$ and four uncapacitated edges:
$\{x_\Diamond, [\underline{A}, \underline{B}]_\Diamond\}$ directed from $[\underline{A}, \underline{B}]_\Diamond$ to $x_\Diamond$,
$\{y_\Diamond, [\underline{A}, \underline{B}]_\Diamond\}$ directed from $y_\Diamond$ to $[\underline{A}, \underline{B}]_\Diamond$,
$\{y_\Diamond, [\underline{D}, \underline{C}]_\Diamond\}$ directed from $[\underline{D}, \underline{C}]_\Diamond$ to $y_\Diamond$ and
$\{x_\Diamond, [\underline{D}, \underline{C}]_\Diamond\}$ directed from $x_\Diamond$ to $[\underline{D}, \underline{C}]_\Diamond$.
We add a fifth edge $e := \{y_\Diamond, x_\Diamond\}$ directed from $y_\Diamond$ to $x_\Diamond$ with $l_e = 0$, $u_e = \infty$. Then we create homologous sets that induce $\delta([\underline{A}, \underline{B}], f) = \delta([\underline{A}, \underline{B}]_\Diamond, f)$ and $\delta([\underline{D}, \underline{C}], f) = \delta([\underline{D}, \underline{C}]_\Diamond, f)$. Note that the capacity constraints on the gadget are equivalent to $\delta([\underline{A}, \underline{B}]_\Diamond, f) \leq \delta([\underline{D}, \underline{C}]_\Diamond, f)$.

As a last step, we split the vertex o into two vertices s and t such that s is connected with the source vertices and t is connected with the sink vertices and add an edge $\{t, s\}$ directed from t towards s. Then we have described the problem of deciding positivity of Littlewood-Richardson coefficients as a homologous flow problem in a natural way.

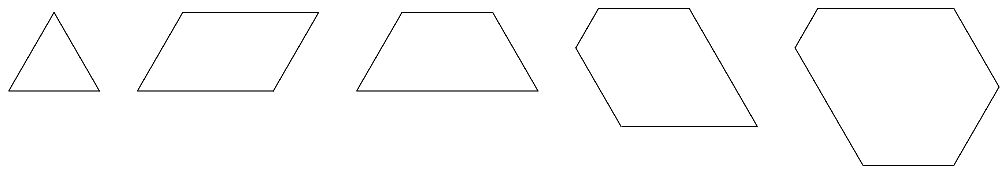

Figure 6.9: Possible shapes of flatspaces up to rotations, mirroring and different side lengths.

6.4 The basic algorithm LRPA

The main idea of the LRPA is to find $f \in P^b$ which maximizes δ in P^b by doing integral steps only. We will see that by doing so we can find an integral $f \in P^b$ which maximizes δ. If $\delta(f) = 2|\nu|$ this proves $c^{\nu}_{\lambda\mu} > 0$ as seen in Lemma 6.8(6). If $\delta(f) < 2|\nu|$, then Lemma 6.8(5) says that $c^{\nu}_{\lambda\mu} = 0$. The LRPA starts with $f := 0 \in P^b$ and increases the throughput $\delta(f)$ while preserving an integral b-bounded hive flow f.

The LRPA has a structure similar to the FFA presented in Section 5.5. But the first problems already appear when trying to construct a residual network. We manage in Section 6.4.2 to construct a residual network in which the hive inequalities are represented as edge capacities. We show in Section 6.4.5 how shortest cycles on this residual network can be used to make integral steps in P^b.

6.4.1 Flatspaces

The LRPA can only construct a residual network for so-called shattered flows f. Therefore in this section we introduce the notion of shatteredness.

A *small triangle* is a triangle formed by 3 pairwise adjacent vertices in the big triangle graph Δ. Two small triangles are denoted connected, if they share a side. An f-*flatspace* is a maximal connected union of small triangles such that any rhombus contained in it is f-flat. We simply write *flatspace*, if it is clear, which flow is meant. The flatspaces split the big hive triangle up in disjoint regions. The following properties are easy to verify (cf. [Buc00]):

(1) Flatspaces are convex.

(2) All flatspaces have one of the shapes in Figure 6.9 up to rotations, mirroring and different side lengths.

(3) A side of a flatspace is either on the border of Δ, or it is also a side of a neighbor flatspace.

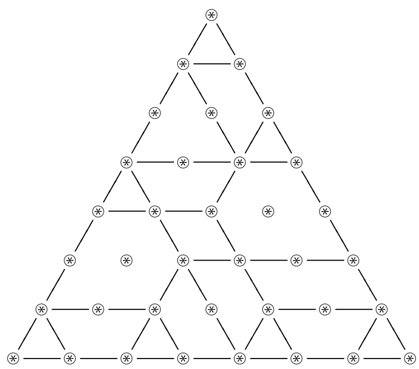

Figure 6.10: An example of a degeneracy graph.

(4) If the border of a hive is regular, then no flatspace has a side of length ≥ 2 on the border.

For a hive h we can draw the *degeneracy graph* by removing all "diagonal" edges $\{\underline{B}, \underline{D}\}$ from Δ for which $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ is flat. See Figure 6.10 for an example.

Flatspaces of rhombic shape that do not have side lengths $(1, 1, 1, 1)$ are called *big rhombi*. Recall that flatspaces of rhombic shape that have side lengths $(1, 1, 1, 1)$ are just called *rhombi*. We denote all flatspaces that are not small triangles or rhombi as *big flatspaces*.

Definition 6.10 (Shattered hive)**.** We call a hive $h \in \mathbb{R}^{H'}$ a *shattered hive*, if all of its flatspaces are small triangles or rhombi. We then call $\eta(h) \in F$ a shattered hive flow. ■

6.4.2 The residual network

In this section we introduce the residual network, in which the hive inequalities will be converted into edge capacities.

Fix a target border $b \in \mathbb{R}^B$ that comes from partitions and fix a b-bounded shattered hive flow f. The residual network $\mathsf{RES}^b(f)$ w.r.t. b and f is constructed as follows. The vertex and edge set of $\mathsf{RES}^b(f)$ are initially the vertex and edge set of G. Then each f-flat rhombus $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ is replaced by the following construction (illustrated in Figure 6.11):

Remove all inner vertices of $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ and keep $[\underline{A}, \underline{B}]$, $[\underline{C}, \underline{B}]$, $[\underline{D}, \underline{C}]$ and $[\underline{D}, \underline{A}]$. Then add *auxiliary vertices* $v_1, \ldots, v_{14}$. Now we add edges, some of which are marked with $^+$ or $^-$. We use the following syntax: $(w_1 \to^+ w_2 \leftarrow w_3)$ means that we add the edge $\{w_1, w_2\}$ directed from w_1 towards w_2 and marked with a $^+$ and we add $\{w_2, w_3\}$ directed from w_3 towards w_2. The intention of a $^+$ sign is

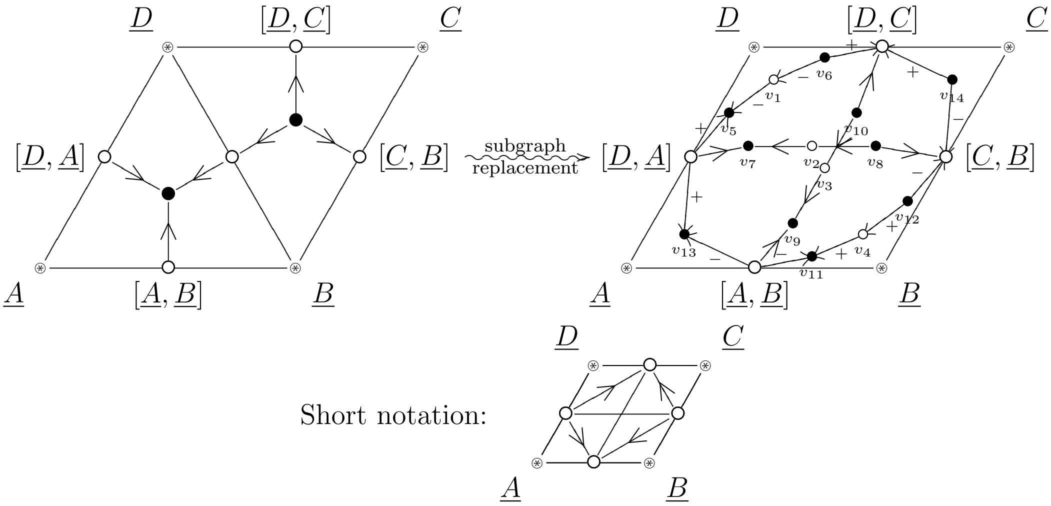

Figure 6.11: The subgraph replacement for an f-flat rhombus $\Diamond(\underline{A},\underline{B},\underline{C},\underline{D})$ and its short notation.

that the edge can only be passed by a well-directed cycle in the edge's direction while edges with a $^-$ sign can only be passed by a well-directed cycle against the edge's direction (compare Definition 5.9). The edges are the following:
$p_{\{[\underline{D},\underline{A}],[\underline{D},\underline{C}]\}} := ([\underline{D},\underline{A}] \rightarrow^+ v_5 \leftarrow^- v_1 \leftarrow^- v_6 \rightarrow^+ [\underline{D},\underline{C}])$,
$p_{\{[\underline{D},\underline{A}],[\underline{C},\underline{B}]\}} := ([\underline{D},\underline{A}] \rightarrow v_7 \leftarrow v_2 \leftarrow v_8 \rightarrow [\underline{C},\underline{B}])$,
$p_{\{[\underline{D},\underline{A}],[\underline{A},\underline{B}]\}} := ([\underline{D},\underline{A}] \rightarrow^+ v_{13} \leftarrow^- [\underline{A},\underline{B}])$,
$p_{\{[\underline{A},\underline{B}],[\underline{D},\underline{C}]\}} := ([\underline{A},\underline{B}] \rightarrow v_9 \leftarrow v_3 \leftarrow v_{10} \rightarrow [\underline{D},\underline{C}])$,
$p_{\{[\underline{A},\underline{B}],[\underline{C},\underline{B}]\}} := ([\underline{A},\underline{B}] \rightarrow^- v_{11} \leftarrow^+ v_4 \leftarrow^+ v_{12} \rightarrow^- [\underline{C},\underline{B}])$ and
$p_{\{[\underline{C},\underline{B}],[\underline{D},\underline{C}]\}} := ([\underline{C},\underline{B}] \leftarrow^- v_{14} \rightarrow^+ [\underline{D},\underline{C}])$.
We call the set of vertices and edges $p_{\{v,w\}}$ the *direct path between v and w*.

Note that in $\mathsf{RES}^b(f)$, the circle vertex $[\underline{B},\underline{D}]$ is no longer present. We note that $\mathsf{RES}^b(f)$ is still bipartite, but may not be planar. Up to here we defined the digraph $\mathsf{RES}(f)$ independent of b.

We now introduce capacities on edges. For each edge e put initially $l_e \leftarrow -\infty$ and $u_e \leftarrow \infty$. For each edge e that is marked with a $+$ sign, set $l_e \leftarrow 0$. This enforces that a well-directed cycle can only pass such e in the direction of e. For each edge e that is marked with a $-$ sign, set $u_e \leftarrow 0$. This enforces that a well-directed cycle can only pass such e in the reverse direction of e. We now introduce additional capacities that are dependent on b. For each edge $e = \{o, s\}$ with $s \in \mathscr{S}$ we set $u_e \leftarrow \delta^b_{\max}(s) - \delta(s, f)$. For each edge $e = \{o, t\}$ with $t \in \mathscr{T}$ we set $l_e \leftarrow \delta^b_{\min}(t) - \delta(t, f)$.

If we are not interested in the exact capacities, we write $\mathsf{RES}^{\mathrm{sgn}b}(f)$ and set $u_e \leftarrow \infty$ for all $e \in E$ with $u_e > 0$ and $l_e \leftarrow -\infty$ for all $e \in E$ with $l_e < 0$. We note that the feasible flows on $\mathsf{RES}^{\mathrm{sgn}b}(f)$ form a convex cone and that a cycle c

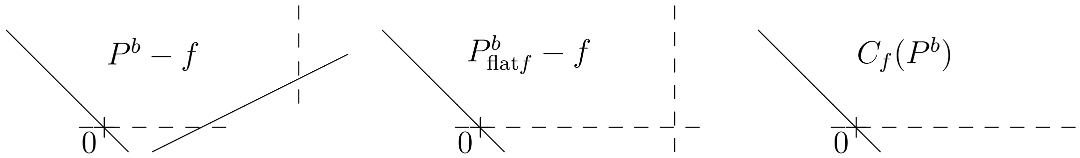

Figure 6.12: Examples of the polyhedra $P^b - f$, $P^b_{\mathrm{flat}f} - f$ and $C_f(P^b)$. Solid lines represent hive inequalities and dashed lines represent border inequalities.

on $\mathsf{RES}^{\mathrm{sgn}b}(f)$ is well-directed iff it is well-directed on $\mathsf{RES}^b(f)$. If we ignore the capacities, then the residual network is independent of b and we call it $\mathsf{RES}(f)$. Let E_{RES} denote the set of edges of $\mathsf{RES}(f)$.

Properties of the residual network We start with a general definition. Given a polyhedron P in a real vector space V and a vector $f \in P$. We can define the *cone of feasible directions $C_f(P)$ of P at f* as

$$C_f(P) := \{d \in V \mid \exists \varepsilon > 0 : f + \varepsilon d \in P\}.$$

Recall that $P^b \subseteq F(G)$ is the polyhedron of all b-bounded hive flows on G and thus

$$C_f(P^b) = \{d \in F(G) \mid \exists \varepsilon > 0 : f + \varepsilon d \in P^b\}.$$

Now for $f \in P^b$ relax P^b to $P^b_{\mathrm{flat}f} \supseteq P^b$ by removing every rhombus inequality that is not induced by an f-flat rhombus. Thus we keep only the rhombus inequalities which are satisfied with equality by f. Note that in a small neighborhood of f, $P^b_{\mathrm{flat}f}$ equals P^b. Figure 6.12 illustrates the relation between $P^b - f$, $P^b_{\mathrm{flat}f} - f$ and $C_f(P^b)$.

The next lemma shows that $C_f(P^b)$ can be understood in terms of the convex cone $P_{\mathrm{feas}}(\mathsf{RES}^{\mathrm{sgn}b}(f))$ of feasible flows on $\mathsf{RES}^{\mathrm{sgn}b}(f)$.

Lemma 6.11 (Residual Correspondence Lemma)**.** *Given a b-bounded shattered hive flow $f \in P^b$. Then there are $\mathbb{Z}$-linear maps*

$$F(G) \underset{\tau'}{\overset{\tau}{\overset{\hookrightarrow}{\twoheadleftarrow}}} F(\mathsf{RES}(f))$$

preserving the throughput of all vertices that are both in G and $\mathsf{RES}(f)$. In particular, these maps preserve the global throughput δ. Moreover $\tau' \circ \tau = id$ and we have the following properties:

(1) $\tau\big(C_f(P^b)\big) \subseteq P_{\text{feas}}(\mathsf{RES}^{\text{sgn}b}(f))$,

(2) $\tau\big(P^b_{\text{flat}f} - f\big) \subseteq P_{\text{feas}}(\mathsf{RES}^b(f))$,

(3) $\tau'\big(P_{\text{feas}}(\mathsf{RES}^{\text{sgn}b}(f))\big) = C_f(P^b)$,

(4) $\tau'\big(P_{\text{feas}}(\mathsf{RES}^b(f))\big) = P^b_{\text{flat}f} - f$.

So via this lemma, feasible flows on $\mathsf{RES}^{\text{sgn}b}(f)$ give the directions from $f \in P^b$ that do not point out of P^b. Moreover, if the border capacity constraints on $\mathsf{RES}^b(f)$ are satisfied for a flow d', then we have $f + \tau'(d') \in P^b_{\text{flat}f}$, which means that there are two cases: Either $f + \tau'(d') \in P^b$ or $f + \tau'(d')$ violates a rhombus inequality of a rhombus that is not f-flat.

Proof of Lemma 6.11. Note that $C_f(P^b)$ is the cone generated by $P^b_{\text{flat}f} - f$ and that $P_{\text{feas}}(\mathsf{RES}^{\text{sgn}b}(f))$ is the cone generated by $P_{\text{feas}}(\mathsf{RES}^b(f))$. Therefore it is sufficient to show the 2nd and 4th claim. For the 4th claim it suffices to show that $\tau'\big(P_{\text{feas}}(\mathsf{RES}^b(f))\big) \subseteq P^b_{\text{flat}f} - f$. The other direction follows from combining the 2nd claim and $\tau' \circ \tau = \text{id}$.

The first map τ Given a flow $d \in F(G)$. We define a flow $\tau(d) := d'$ on $\mathsf{RES}(f)$ as follows: d' equals d on each edge that does not lie in a flat rhombus. For each flat rhombus $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ we set d' to 0 on all edges but the following: The 4 edges $e_1, \ldots, e_4$ on the direct path from $[\underline{A}, \underline{B}]$ to $[\underline{D}, \underline{C}]$ get

$$d'(e_1) := -d'(e_2) := -d'(e_3) := d'(e_4) := \delta^G([\underline{A}, \underline{B}], d).$$

The 4 edges $e_1, \ldots, e_4$ on the direct path from $[\underline{D}, \underline{A}]$ to $[\underline{C}, \underline{B}]$ get

$$d'(e_1) := -d'(e_2) := -d'(e_3) := d'(e_4) := \delta^G([\underline{C}, \underline{B}], d).$$

The 4 edges $e_1, \ldots, e_4$ on the direct path from $[\underline{D}, \underline{A}]$ to $[\underline{D}, \underline{C}]$ get

$$d'(e_1) := -d'(e_2) := -d'(e_3) := d'(e_4) := \delta^G([\underline{D}, \underline{A}], d) - \delta^G([\underline{C}, \underline{B}], d).$$

We now show that d' is a flow on $\mathsf{RES}(f)$:

The flow constraints of d' are satisfied in each fat black vertex due to the fact that they are satisfied in d. We now consider the replacement of a single flat rhombus $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$. Let d' equal d on each edge outside this rhombus. We see from the edge directions that this single replacement only affects $\delta^{\mathsf{RES}(f)}_{\text{out}}([\underline{A}, \underline{B}], d')$, $\delta^{\mathsf{RES}(f)}_{\text{in}}([\underline{C}, \underline{B}], d')$, $\delta^{\mathsf{RES}(f)}_{\text{in}}([\underline{D}, \underline{C}], d')$ and $\delta^{\mathsf{RES}(f)}_{\text{out}}([\underline{D}, \underline{A}], d')$. We now show that these values are equal to their counterparts on G. Recall that d is a flow, which implies $\delta^G([\underline{A}, \underline{B}], d) + \delta^G([\underline{D}, \underline{A}], d) = \delta^G([\underline{C}, \underline{B}], d) + \delta^G([\underline{D}, \underline{C}], d)$.

$$\delta^{\mathsf{RES}(f)}_{\text{out}}([\underline{A}, \underline{B}], d') = \delta^G([\underline{A}, \underline{B}], d), \quad \delta^{\mathsf{RES}(f)}_{\text{in}}([\underline{C}, \underline{B}], d') = \delta^G([\underline{C}, \underline{B}], d),$$

$$\delta_{\text{in}}^{\mathsf{RES}(f)}([\underline{D},\underline{C}],d') = \delta^G([\underline{A},\underline{B}],d) + \delta^G([\underline{D},\underline{A}],d) - \delta^G([\underline{C},\underline{B}],d)$$
$$= \delta^G([\underline{D},\underline{C}],d),$$
$$\delta_{\text{out}}^{\mathsf{RES}(f)}([\underline{D},\underline{A}],d') = \delta^G([\underline{C},\underline{B}],d) + \delta^G([\underline{D},\underline{A}],d) - \delta^G([\underline{C},\underline{B}],d)$$
$$= \delta^G([\underline{D},\underline{A}],d).$$

We now show that $d \in P^b_{\text{flat} f} - f$ implies $d' \in P_{\text{feas}}(\mathsf{RES}^b(f))$.

We have to show that

$$\forall e \in E : l_e \le d'(e) \le u_e.$$

By construction this is satisfied on all edges incident to o.

Now consider an edge e that lies in the big triangle. To be capacitated, e must lie in a flat rhombus $\Diamond(\underline{A},\underline{B},\underline{C},\underline{D})$. If e is a capacitated edge and $d'(e) \neq 0$, then e must be one of the four edges $e_1,\dots,e_4$ on the direct path from $[\underline{D},\underline{A}]$ to $[\underline{D},\underline{C}]$. As d' is a flow, the capacity constraints of $e_1,\dots,e_4$ are satisfied iff the capacity constraint of e_1 is satisfied, which means $d'(e_1) \ge 0$. We have

$$d'(e_1) = \delta^G([\underline{D},\underline{A}],d) - \delta^G([\underline{C},\underline{B}],d).$$

As $\Diamond(\underline{A},\underline{B},\underline{C},\underline{D})$ is f-flat, we have

$$\delta^G([\underline{C},\underline{B}],f) = \delta^G([\underline{D},\underline{A}],f).$$

Combining both equations we get

$$\delta^G([\underline{C},\underline{B}],f+d) = \delta^G([\underline{D},\underline{A}],f+d) - d'(e_1).$$

From $d + f \in P^b_{\text{flat} f}$ it follows that

$$\delta^G([\underline{C},\underline{B}],f+d) \le \delta^G([\underline{D},\underline{A}],f+d)$$

and therefore $d'(e_1) \ge 0$.

The second map τ' The map τ' is defined in the obvious way: Given a flow $d' \in F(\mathsf{RES}(f))$. We define a flow $\tau'(d') := d \in F(G)$ as follows: d equals d' on each edge that does not lie in a flat rhombus. For each flat rhombus $\Diamond(\underline{A},\underline{B},\underline{C},\underline{D})$ we define the following:
The edge e directed from $[\underline{A},\underline{B}]$ gets $d(e) = \delta_{\text{out}}^{\mathsf{RES}(f)}([\underline{A},\underline{B}],d')$.
The edge e directed from $[\underline{D},\underline{A}]$ gets $d(e) = \delta_{\text{out}}^{\mathsf{RES}(f)}([\underline{D},\underline{A}],d')$.
The edge e directed towards $[\underline{D},\underline{C}]$ gets $d(e) = \delta_{\text{in}}^{\mathsf{RES}(f)}([\underline{D},\underline{C}],d')$.
The edge e directed towards $[\underline{C},\underline{B}]$ gets $d(e) = \delta_{\text{in}}^{\mathsf{RES}(f)}([\underline{C},\underline{B}],d')$.
The edge e directed from $[\underline{B},\underline{D}]$ gets $d(e) = \sum_{i=1}^4 \delta_{\text{out}}^{\mathsf{RES}(f)}(v_i,d')$.
The edge e directed towards $[\underline{B},\underline{D}]$ gets $d(e) = \sum_{i=1}^4 \delta_{\text{in}}^{\mathsf{RES}(f)}(v_i,d')$.

We now show that $g \in F(G)$:

For a given flat rhombus $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ the flow constraints on $[\underline{B}, \underline{D}]$ are satisfied, because they are satisfied on $v_1, \dots, v_4$. As for the first map, we only have to check the values of $\delta^G_{\text{out}}([\underline{A}, \underline{B}], d)$, $\delta^G_{\text{in}}([\underline{C}, \underline{B}], d)$, $\delta^G_{\text{in}}([\underline{D}, \underline{C}], d)$ and $\delta^G_{\text{out}}([\underline{D}, \underline{A}], d)$. But these are equal to their counterparts on $\mathsf{RES}(f)$ by definition.

From the definitions, it follows that $\tau' \circ \tau = \text{id}$.

We now show that $d' \in P_{\text{feas}}(\mathsf{RES}^b(f))$ implies $d \in P^b_{\text{flat}f} - f$:

The capacity constraints on the circle border vertices force $f + d$ to be b-bounded. We know that for each f-flat rhombus $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ we have

$$\delta^G([\underline{D}, \underline{A}], f) = \delta^G([\underline{C}, \underline{B}], f).$$

We must show that for each f-flat rhombus $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ it holds

$$\delta^G([\underline{D}, \underline{A}], f + d) \geq \delta^G([\underline{C}, \underline{B}], f + d).$$

Thus it suffices to show that

$$\delta^G([\underline{D}, \underline{A}], d) \geq \delta^G([\underline{C}, \underline{B}], d).$$

From the capacity constraints on the edges it follows that

$$\begin{array}{rcl} \delta^G([\underline{D}, \underline{A}], d) & = & \delta^{\mathsf{RES}(f)}([\underline{D}, \underline{A}], d') \\ & = & d'(\{[\underline{D}, \underline{A}], v_5\}) + d'(\{[\underline{D}, \underline{A}], v_7\}) + d'(\{[\underline{D}, \underline{A}], v_{13}\}) \\ & \geq & d'(\{[\underline{D}, \underline{A}], v_7\}) \ = \ d'(\{[\underline{C}, \underline{B}], v_8\}) \\ & \geq & d'(\{[\underline{C}, \underline{B}], v_8\}) + d'(\{[\underline{C}, \underline{B}], v_{12}\}) + d'(\{[\underline{C}, \underline{B}], v_{14}\}) \\ & = & \delta^{\mathsf{RES}(f)}([\underline{C}, \underline{B}], d') \ = \ \delta^G([\underline{C}, \underline{B}], d). \end{array}$$

□

Note that there can be well-directed cycles c on $\mathsf{RES}^{\text{sgn}b}(f)$ that are mapped by τ' to a flow that is not a cycle. See Figure 6.13 for examples.

For the construction of $\mathsf{RES}(f)$, we need f to be shattered. This is a fundamental restriction and the LRP-CSA spends much of its running time on keeping f shattered. The following example explains why shatteredness is important:

Example Consider the case where $n = 2$, depicted in Figure 6.14. Let $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ and $\Diamond(\underline{E}, \underline{C}, \underline{D}, \underline{B})$ be f-flat rhombi and let $\Diamond(\underline{F}, \underline{D}, \underline{B}, \underline{C})$ be not f-flat. Let no circle border vertex be on its δ-bound: For all $s \in \mathscr{S}$ we have $\delta(s, f) < \delta^b_{\max}(s)$ and for all $t \in \mathscr{T}$ we have $\delta(t, f) > \delta^b_{\min}(t)$. Then in $C_f(P^b)$ there is a flow d with the following throughput: $\delta([\underline{A}, \underline{B}], d) = 1$, $\delta([\underline{B}, \underline{E}], d) = 1$, $\delta([\underline{E}, \underline{C}], d) = -1$, $\delta([\underline{D}, \underline{C}], d) = 1$ and $\delta([\underline{D}, \underline{A}], d) = 0$ (This setting can be seen as a "tilting" operation: $\underline{E}$ is being raised by 2 units, while $\underline{B}$ and $\underline{C}$ are being raised by 1 unit). We use a shorter notation

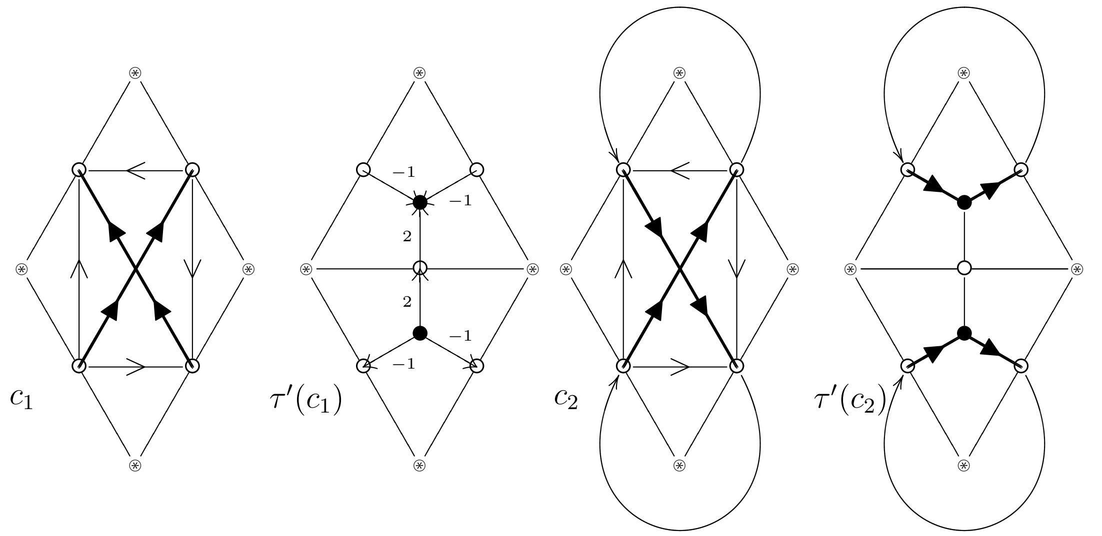

Figure 6.13: A well-directed cycle c_1 in a flat rhombus is mapped to a flow $\tau'(c_1)$ which has a flow value of 2 on some edges and a well-directed cycle c_2 in a flat rhombus is mapped to a flow $\tau'(c_2)$ which decomposes into at least two cycles.

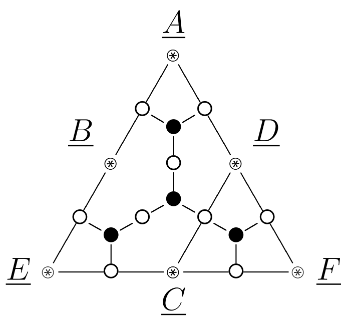

Figure 6.14: A hive which is not shattered.

for the throughput on the trapezoid $(\underline{E}, \underline{C}, \underline{D}, \underline{A}, \underline{B})$: $\delta_{\text{tra}}(d) = (1, 1, -1, 1, 0)$. Now assume that we could construct a residual network in this case, formally: Assume that there is a digraph $\mathsf{RES}^{\text{sgn}b}(f)$ and there is a pair of maps $\tau : F(G) \to F(\mathsf{RES}(f))$, $\tau' : F(\mathsf{RES}(f)) \to F(G)$ that preserve the throughput on all circle vertices and $\tau'(P_{\text{feas}}(\mathsf{RES}^{\text{sgn}b}(f))) = C_f(P^b)$. Then there is a feasible flow $d' \in P_{\text{feas}}(\mathsf{RES}^{\text{sgn}b}(f))$ with $\delta_{\text{tra}}(\tau'(d')) = (1, 1, -1, 1, 0)$. The flow d' can be decomposed into well-directed cycles. There are three possibilities:

- One cycle c_1 has $\delta_{\text{tra}}(c_1) = (1, 0, 0, 1, 0)$ and another cycle c_2 has $\delta_{\text{tra}}(c_2) = (0, 1, -1, 0, 0)$. But then $\tau(c_1) \notin C_f(P^b)$, which is a contradiction.
- One cycle c_1 has $\delta_{\text{tra}}(c_1) = (1, 0, -1, 0, 0)$ and another cycle c_2 has $\delta_{\text{tra}}(c_2) = (0, 1, 0, 1, 0)$. But then $\tau(c_1) \notin C_f(P^b)$, which is a contradiction.
- One cycle c_1 has $\delta_{\text{tra}}(c_1) = (1, 1, 0, 0, 0)$ and another cycle c_2 has $\delta_{\text{tra}}(c_2) = (0, 0, -1, 1, 0)$. But then neither $\tau'(c_1)$ nor $\tau'(c_2)$ can satisfy the flow constraints on G, which is a contradiction.

Basically these tilting operations permit constructions of residual networks for big flatspaces. So before constructing $\mathsf{RES}(f)$, it must be made sure that f is shattered.

Cycles on $\mathsf{RES}^b(f)$ can be used to determine whether f is optimal w.r.t. δ with the following lemma:

Lemma 6.12 (Optimality Test)**.** *Given a shattered, b-bounded hive flow $f \in P^b$ and any linear function $\delta : F \to \mathbb{R}$, then f maximizes δ in P^b iff $\mathsf{RES}^b(f)$ has no well-directed cycle c with $\delta(\tau'(c)) > 0$.*

Proof.

$$
\begin{aligned}
& f \text{ does not maximize } \delta \text{ in } P^b \\
\iff\; & \exists d \in F \text{ with } f + d \in P^b \text{ and } \delta(d + f) > \delta(f) \\
\iff\; & \exists d \in P^b - f \text{ and } \delta(d) > 0 \\
\overset{(*)}{\iff}\; & \exists d \in P^b_{\text{flat}f} - f \text{ and } \delta(d) > 0 \\
\overset{(**)}{\iff}\; & \exists d' \in P_{\text{feas}}(\mathsf{RES}^b(f)) \text{ and } \delta(\tau'(d')) > 0 \\
\overset{(***)}{\iff}\; & \exists \text{ a well-directed cycle } c \text{ on } \mathsf{RES}^b(f) \text{ with } \delta(\tau'(c)) > 0
\end{aligned}
$$

$(*)$ holds, because P^b equals $P^b_{\text{flat}f}$ in a small neighborhood of f.

$(**)$ is true due to the Residual Correspondence Lemma 6.11.

We now prove $(***)$: Let $d' \in P_{\text{feas}}(\mathsf{RES}^b(f))$ with $\delta(\tau'(d')) > 0$. Then Lemma 5.11 says that d' can be decomposed into well-directed cycles on $\mathsf{RES}^b(f)$: $d' = \sum_{i=1}^{M} \alpha_i c_i$ where $\alpha_i > 0$ for all $1 \le i \le M$. Thus

$$0 < \delta(\tau'(d')) = \delta\left(\tau'\left(\sum_{i=1}^{M} \alpha_i c_i\right)\right) = \sum_{i=1}^{M} \alpha_i \delta(\tau'(c_i))$$

and therefore there is a well-directed cycle c_i with $\delta(\tau'(c_i)) > 0$.

On the other hand, given a well-directed cycle c on $\mathsf{RES}^b(f)$ with $\delta(\tau'(c)) > 0$, according to Lemma 5.10 this gives rise to a feasible flow εc on $\mathsf{RES}^b(f)$ with $\delta(\tau'(\varepsilon c)) > 0$ for some $\varepsilon > 0$. □

Recall that $\delta(f) = \sum_{s\in\mathscr{S}} \delta(s, f) - \sum_{t\in\mathscr{T}} \delta(t, f)$. We give some intuition about cycles c on $\mathsf{RES}^b(f)$ with $\delta(\tau'(c)) > 0$:

Lemma 6.13. *Given a shattered flow $f \in P^b$ and a well-directed cycle c on $\mathsf{RES}^b(f)$ with $\delta(\tau'(c)) > 0$, then there are two circle border vertices $s \in \mathscr{S}$ and $t \in \mathscr{T}$ such that $\delta(s, c) = 1$, $\delta(t, c) = -1$ and $\delta(v, c) = 0$ for each circle border vertex $v \notin \{s, t\}$.*

Proof. As c is a cycle, c can only use the vertex o once. So we have $\delta(v, \tau'(c)) \neq 0$ for at most two circle border vertices v. As c satisfies $\delta(\tau'(c)) > 0$, those two vertices cannot lie on the same side of the big triangle and c must use at least one circle border vertex. Because of the flow constraints c must use exactly two such circle border vertices $s \in \mathscr{S}$ and $t \in \mathscr{T}$ with $\delta(s, c) = 1$ and $\delta(t, c) = -1$. □

6.4.3 Flatspace chains and increasable subsets

In this section we explain flatspace chains and increasable subsets and how they can be used to shatter a flow.

Definition 6.14 (Increasable subset)**.** An *increasable subset w.r.t. a hive* $h \in \mathbb{R}^{H'}$ is a subset of vertices of $S \subseteq H'$ such that $\varepsilon > 0$ exists with $h + \varepsilon\chi_S$ is a hive, where $\chi_S(\underline{A}) = 1$ if $\underline{A} \in S$ and $\chi_S(\underline{A}) = 0$ otherwise. ■

Definition 6.15 (Flatspace Chain)**.** A *flatspace chain* Ψ *w.r.t.* $h \in \mathbb{R}^{H'}$ is a region of connected flatspaces constructed in the following way (cf. [Buc00]):

(1) A flat hexagon is a flatspace chain on its own. If there are flat hexagons, then these are the only flatspace chains.

(2) If there are no flat hexagons, let m be the maximal length among all sides of flatspaces. If $m = 1$, then there are no flatspace chains and the hive is shattered. If $m \geq 2$, then start by taking a flatspace which has a side of length m and mark this side (see Figure 6.15). m is denoted the *width of* Ψ. Otherwise choose and fix a line crossing (the extension of) the marked side in an angle of 60° and call it the moving direction. If the flatspace is a triangle or a parallelogram, we furthermore mark an additional side. For a triangle, this is the other side not parallel to the moving direction, while for a parallelogram we mark the side opposite the one already marked. We construct the flatspace chain, starting with the chosen flatspace. This region will initially have one or two marked sides, depending on the shape

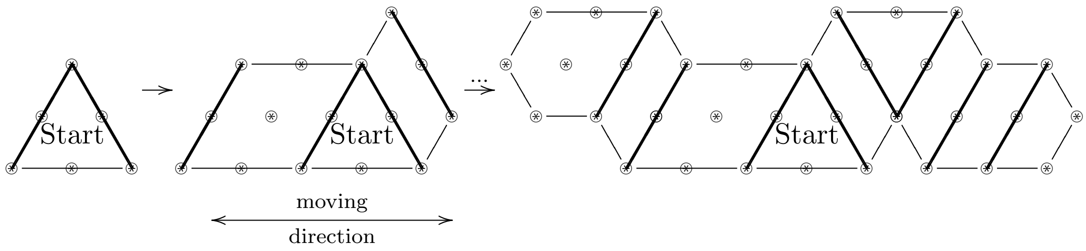

Figure 6.15: The construction of a flatspace chain.

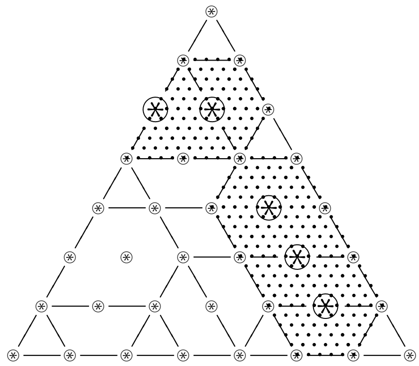

Figure 6.16: Two examples of flatspace chains. The upper one has an open ending on the left. The inner vertices are drawn bigger than others.

of the chosen flatspace. As long as the region has a marked side on its outer border and the marked side does not lie on the border, the flatspace on the opposite side is added to the region. If the new flatspace is a triangle, we mark its unmarked side which is not parallel to the moving direction. If the new flatspace is a parallelogram, we mark the side opposite the old marked side. If it is not a triangle or parallelogram, we do not mark any new sides.

■

Since the region always grows along the moving direction, it will never go in loops. If a flatspace chain stops with a marked side on the border, we call this side an *open ending*. By construction there can be at most 2 open endings. See Figure 6.16 for examples on how a flatspace chains look like in the degeneracy graph. We remark that the constructive definition of flatspace chains gives a straightforward way to compute a flatspace chain in polynomial time.

Let $\Psi_{\text{inner}} \subseteq H$ denote the set of inner vertices of the area of Ψ united with the inner vertices of open endings of Ψ. We call Ψ_{inner} the *set of inner vertices of* Ψ.

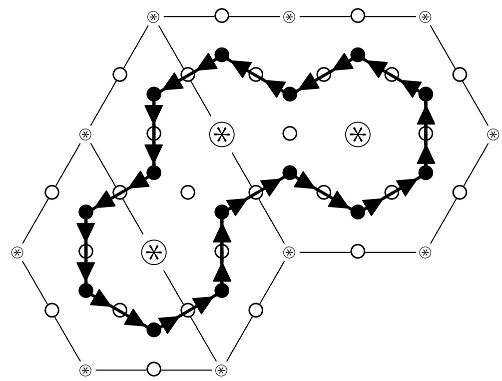

Figure 6.17: A flatspace chain Ψ consisting of a pentagon, a parallelogram and a trapezoid. The inner vertices are drawn bigger than others. Only the edges of G are drawn that carry nonzero flow in f_Ψ.

So if the sides of Ψ at the border of Δ have length 1, then Ψ_{inner} consists of only the inner vertices of the area of Ψ. We define $\chi_{\Psi_{\text{inner}}} : \mathbb{R}^{H'} \to \mathbb{R}$, $\chi_{\Psi_{\text{inner}}}(\underline{A}) = 1$ for all $\underline{A} \in \Psi_{\text{inner}}$ and $\chi_{\Psi_{\text{inner}}}(\underline{A}) = 0$ otherwise. For a flatspace chain Ψ we define f_Ψ to be the flow induced by raising Ψ_{inner} by 1 unit: $f_\Psi := \eta(\chi_{\Psi_{\text{inner}}})$. An example for f_Ψ is given in Figure 6.17. We see that f_Ψ can be interpreted as a cycle on G. Moreover, since each open ending cannot span more than one side of Δ, we have $\delta(f_\Psi) = 0$ for any flatspace chain Ψ.

Lemma 6.16. *Let $z \in \mathbb{R}$. Given a z-integral hive h and a flatspace chain Ψ of h whose sides at the border of Δ have length 1, then $h + z\chi_{\Psi_{\text{inner}}}$ is a hive.*

Proof. As h is z-integral, we have for each rhombus $\Diamond$ that $\sigma(\Diamond, h) = 0$ or $\sigma(\Diamond, h) \geq z$. It is important that flatspace chains have acute angles only at open endings. Therefore there are no flat rhombi $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ that have only one vertex contained in Ψ_{inner} and this vertex is an acute one $\underline{A}$ or $\underline{C}$. As flatspace chains do not have loops, there are no rhombi $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ that have only the acute vertices $\underline{A}$ and $\underline{C}$ in Ψ_{inner}. Therefore $\sigma(\Diamond, f_\Psi) \geq -z$ for all rhombi $\Diamond$ and $\sigma(\bar{\Diamond}, f_\Psi) \geq 0$ for all flat rhombi $\bar{\Diamond}$. This proves the claim.

□

If a hive has regular border, no flatspace chain can have an open ending. Therefore, if a hive has a regular border, all big flatspaces can be eliminated by increasing the increasable subsets induced by flatspace chains to their maximum. The border vertices are not touched during this operation. This creates a shattered hive.

The following lemma states that inner vertices of flatspace chains with open endings can also be raised in certain situations:

Lemma 6.17. *Let b be an integral, regular target border. Given an integral flow $f \in P^b$ and an f-flatspace chain Ψ, then $f + f_\Psi \in P^b$.*

Proof. If Ψ has no open ending, then Lemma 6.16 finishes the proof. Otherwise let $\{v_0, \mathrm{pred}(v_0), \ldots, \mathrm{pred}^m(v_0) =: v_m\} \subseteq \mathscr{S}$ lie in an open ending of Ψ on the border of the big triangle with $\delta(v_0, f_\Psi) = -1, \delta(v_m, f_\Psi) = 1$. Decreasing throughput on v_0 is not problematic, but in v_m the b-boundedness of $f + f_\Psi$ must be checked. As v_0 and v_m lie on the same side of an f-flatspace chain, we have $\delta(v_0, f) = \delta(v_m, f)$. As the target border is integral and regular, with Lemma 6.8(7) we have $\delta^b_{\max}(v_0) + 1 \leq \delta^b_{\max}(v_m)$. Then $\delta(v_m, f + f_\Psi) = \delta(v_m, f) + 1 = \delta(v_0, f) + 1 \leq \delta^b_{\max}(v_0) + 1 \leq \delta^b_{\max}(v_m)$. The proof for $\mathscr{T}$ is analogous. $\square$

6.4.4 The LRPA and the Saturation Conjecture

The basic algorithm LRPA is listed as Algorithm 3. The most interesting property is that *shortest* well-directed cycles on $\mathsf{RES}^b(f)$ can be used to increase $\delta(f)$ by 1 unit (see line 15) and so f stays integral all the time. The reason for this is explained in Section 6.4.5.

Theorem 6.18. *If given as input three strictly decreasing partitions λ, μ, $\nu \in \mathbb{N}^n$ with $|\nu| = |\lambda| + |\mu|$, then the LRPA returns true iff $c^\nu_{\lambda\mu} > 0$.*

Proof. First of all, the algorithm checks whether $\ell(\nu) < \max\{\ell(\lambda), \ell(\mu)\}$. If this is the case, then we have $c^\nu_{\lambda\mu} = 0$ and need no additional computation.

Note that during the algorithm f stays integral all the time, because inner vertices of flatspace chains in line 9 are raised by 1 unit and $\tau'(c)$ in line 15 is integral. Raising the inner vertices of flatspace chains by 1 unit is possible, even if they have open endings. This is due to Lemma 6.17. We do a rather involved proof for $f + \tau'(c) \in P^b$ in Section 6.4.5.

So if the algorithm returns true, an integral $f \in P^b$ with $\delta(f) = 2|\nu|$ is found. Lemma 6.8(6) shows that $c^\nu_{\lambda\mu} > 0$. If the algorithm returns false and did not exit in line 2, then there is $f \in P^b$ with $\delta(f) < 2|\nu|$ and f maximizes δ in P^b according to Lemma 6.12. Therefore with Lemma 6.8(5), we have $c^\nu_{\lambda\mu} = 0$. $\square$

The Saturation Conjecture Given $N \in \mathbb{N}$, λ, μ, ν strictly decreasing partitions with $|\nu| = |\lambda| + |\mu|$. If $c^{N\nu}_{N\lambda, N\mu} > 0$, then there is an integral hive with border associated with $N\lambda, N\mu$ and $N\nu$. This results in a rational hive with border $b = b(\lambda, \mu, \nu)$. Then there is a flow $f \in P^b$ with $\delta(f) = 2|\nu|$. In this case, the LRPA will find an *integral* flow $f \in P^b$ with $\delta(f) = 2|\nu|$ and therefore $c^\nu_{\lambda\mu} > 0$. So the correctness proof of the LRPA is a proof for the Saturation Conjecture in the case of strictly decreasing partitions. We will see in Section 7.7 that a variant

Algorithm 3 The LRPA

Input: $\lambda, \mu, \nu \in \mathbb{N}^n$ strictly decreasing partitions with $|\nu| = |\lambda| + |\mu|$.
Output: Decide whether $c^{\nu}_{\lambda\mu} > 0$.

if $\ell(\nu) < \max\{\ell(\lambda), \ell(\mu)\}$ **then**
  return **false**.
end if
Create the regular target border b and the digraph G.
Start with $f \leftarrow 0$.
`done` $\leftarrow$ **false**.
while not `done` **do**
  while there are f-flatspace chains **do**
    Raise the inner vertices of an f-flatspace chain Ψ by 1: $f \leftarrow f + f_\Psi$.
  end while
  // f is shattered now.
  Construct $\mathsf{RES}^b(f)$.
  if there is a well-directed cycle in $\mathsf{RES}^b(f)$ with $\delta(\tau'(c)) > 0$ **then**
    Find a shortest well-directed cycle c in $\mathsf{RES}^b(f)$ with $\delta(\tau'(c)) > 0$.
    Augment 1 unit over c: $f \leftarrow f + \tau'(c)$.
    // We have $f \in P^b$.
  else
    `done` $\leftarrow$ **true**.
  end if
end while
if $\delta(f) = 2|\nu|$ **then**
  return **true**.
else
  return **false**.
end if

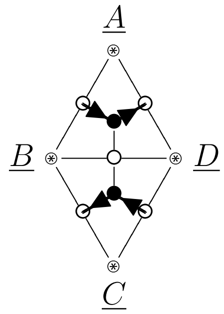

Figure 6.18: A rhombus $\lozenge := \lozenge(\underline{A},\underline{B},\underline{C},\underline{D})$ with $\sigma\big(\lozenge,c\big) = -2$.

of the LRP-CSA can also be used for partitions that are not strictly decreasing. This proves the Saturation Conjecture for arbitrary partitions.

6.4.5 Shortest well-directed cycles

In this section we show that in Algorithm 3, after executing line 15, we have $f \in P^b$. To simplify the notation, we define the throughput of a flow d on $\mathsf{RES}^b(f)$ and a vertex $v \in G$ as $\delta(v,d) := \delta(v,\tau'(d))$. In particular, we set $\delta(d) := \delta(\tau'(d))$. We do the same for the slack of any rhombus $\lozenge$ by setting $\sigma\big(\lozenge,d\big) := \sigma\big(\lozenge,\tau'(d)\big)$ for any flow d on $\mathsf{RES}^b(f)$. Recall that

$$\sigma\big(\lozenge(\underline{A},\underline{B},\underline{C},\underline{D}),f\big) = \delta([\underline{D},\underline{C}],f) - \delta([\underline{A},\underline{B}],f) = \delta([\underline{D},\underline{A}],f) - \delta([\underline{C},\underline{B}],f)$$

and that the slack is linear in the flow, i.e. for each rhombus $\lozenge$ we have for all flows f_1, f_2 on G and for all $z_1, z_2 \in \mathbb{R}$ that $\sigma\big(\lozenge, z_1 f_1 + z_2 f_2\big) = z_1\sigma\big(\lozenge, f_1\big) + z_2\sigma\big(\lozenge, f_2\big)$.

Unfortunately not all well-directed cycles c on $\mathsf{RES}^b(f)$ result in flows $\tau'(c)$ with $f + \tau'(c) \in P^b$:
Consider for example a rhombus $\lozenge := \lozenge(\underline{A},\underline{B},\underline{C},\underline{D})$ with $\sigma\big(\lozenge,f\big) = 1$ and a well-directed cycle c with $\delta([\underline{A},\underline{B}],c) = 1$, $\delta([\underline{D},\underline{A}],c) = -1$, $\delta([\underline{D},\underline{C}],c) = -1$ and $\delta([\underline{C},\underline{B}],c) = 1$ (see Figure 6.18). Then $\sigma\big(\lozenge,c\big) = \delta([\underline{D},\underline{C}],c) - \delta([\underline{A},\underline{B}],c) = -1-1 = -2$ and we have $\sigma\big(\lozenge,f+c\big) = \sigma\big(\lozenge,f\big) + \sigma\big(\lozenge,c\big) = 1-2 < 0$. Thus $f+\tau'(c)$ is not a hive flow, hence $f+\tau'(c) \notin P^b$. A first attempt for finding well-directed cycles c on $\mathsf{RES}^b(f)$ with $f + \tau'(c) \in P^b$ could be to find well-directed cycles c on $\mathsf{RES}^b(f)$ that have $\sigma\big(\lozenge,c\big) \geq -1$ for each rhombus $\lozenge$. But we note that in some situations well-directed cycles can be forced to have $\sigma\big(\lozenge,c\big) = -2$ on some rhombi $\lozenge$. See Figure 6.19 for examples: f-flat rhombi are drawn in short notation (cp. Figure 6.11). The edge directions in the small triangles are left out. In this notation, well-directed cycles can pass undirected edges in any direction and directed edges e only in the direction of e. The fat edges in the figure represent the cycles.

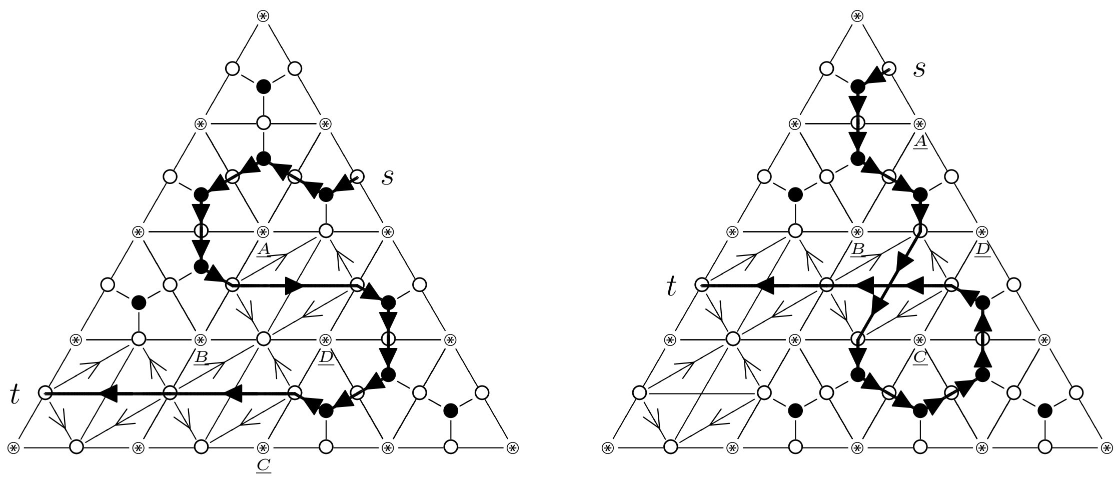

Figure 6.19: Well-directed cycles c that use s and t are sometimes forced to induce $\sigma\big(\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D}), c\big) = -2$.

We now show that any well-directed cycle c on $\mathsf{RES}^b(f)$ with minimal length $\ell(c)$ (i.e. number of edges) satisfies $f + \tau'(c) \in P^b$.

Theorem 6.19 (Shortest Cycle Theorem)**.** *Given a b-bounded integral shattered hive flow f. Given a well-directed cycle c on $\mathsf{RES}^b(f)$ with $\delta(c) > 0$. If $f + \tau'(c) \notin P^b$ then there is a well-directed cycle c' on $\mathsf{RES}^b(f)$ with $\ell(c') < \ell(c)$ and $\delta(c') > 0$.*

Corollary 6.20. *Given a b-bounded integral shattered hive flow f and a well-directed cycle c on $\mathsf{RES}^b(f)$ with $\delta(c) > 0$ that is a shortest cycle among all well-directed cycles $\tilde{c}$ on $\mathsf{RES}^b(f)$ that have $\delta(\tilde{c}) > 0$. Then $f + \tau'(c) \in P^b$.*

Proof of the Shortest Cycle Theorem 6.19. The rest of this section will be devoted to the proof of Theorem 6.19. For the rest of the proof we fix a b-bounded integral shattered hive flow $f \in P^b$ and a well-directed cycle c on $\mathsf{RES}^b(f)$ with $f + \tau'(c) \notin P^b$ and $\delta(c) > 0$. Let

$$\varepsilon := \max\{\varepsilon' \in \mathbb{R} \mid f + \varepsilon'\tau'(c) \in P^b\},$$

$$g := f + \varepsilon\tau'(c)$$

for the rest of the proof. Note that g is not necessarily shattered. Depending on f and g we introduce critical, loose, bending and rigid rhombi:

Definition 1. A rhombus is called *critical*, if it is not f-flat, but g-flat. A rhombus is called *loose*, if it is neither f-flat nor g-flat. A rhombus is called *bending*, if it is f-flat and not g-flat. A rhombus is called *rigid*, if it is both f-flat and g-flat. ■

Lemma 2. *We have the following properties:*

(1) *For all $v \in G$ we have $\delta(v, c) \in \{-2, -1, 0, 1, 2\}$.*

(2) *If $\delta(v, c) = 2$ for $v \in G$, then $v = [\underline{B}, \underline{D}]$ for an f-flat rhombus $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ and c uses all uncapacitated edges of this rhombus, from $[\underline{D}, \underline{C}]$ to $[\underline{A}, \underline{B}]$ and from $[\underline{C}, \underline{B}]$ to $[\underline{D}, \underline{A}]$.*

(3) *If $\delta(v, c) = -2$ for $v \in G$, then $v = [\underline{B}, \underline{D}]$ for an f-flat rhombus $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ and c uses all uncapacitated edges of this rhombus, from $[\underline{A}, \underline{B}]$ to $[\underline{D}, \underline{C}]$and from $[\underline{D}, \underline{A}]$ to $[\underline{C}, \underline{B}]$.*

(4) *For each rhombus $\Diamond$ we have $\sigma\big(\Diamond, c\big) \in \{-3, \ldots, 3\}$.*

(5) *There is at least one critical rhombus.*

(6) *For each critical rhombus $\Diamond$, we have $\sigma\big(\Diamond, c\big) \leq -2$.*

(7) *$\varepsilon \in \{\frac{1}{3}, \frac{1}{2}, \frac{2}{3}\}$.*

(8) *An f-flat rhombus $\Diamond$ is rigid iff c uses no capacitated edge in $\Diamond$.*

Proof. Recall that $\mathsf{RES}(f)$ and $\mathsf{RES}^b(f)$ have the same vertex set. Let $G \setminus \mathsf{RES}(f)$ denote the set of vertices that are in G and not in $\mathsf{RES}^b(f)$.

(1) For each fat black vertex $v \in G$, we have $\delta^G(v, \tilde{f}) = 0$ for any flow $\tilde{f}$ on G, in particular $\delta(v, c) = 0$ for each fat black vertex v of G. For each vertex v' of $\mathsf{RES}^b(f)$, we have $\delta^{\mathsf{RES}^b(f)}(v', c) \in \{-1, 0, 1\}$, because c is a cycle on $\mathsf{RES}^b(f)$. As τ' preserves the throughput on each vertex (see Residual Correspondence Lemma 6.11), we have $\delta^G(v', c) \in \{-1, 0, 1\}$. Let $v'' \in G \setminus \mathsf{RES}(f)$. Then $v'' = [\underline{B}, \underline{D}]$ for an f-flat rhombus $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$. The flow constraints on the fat black vertex in the upright triangle of this rhombus imply that $\delta([\underline{A}, \underline{B}], c) + \delta([\underline{B}, \underline{D}], c) + \delta([\underline{D}, \underline{A}], c) = 0$. As $[\underline{A}, \underline{B}]$ is a vertex of $\mathsf{RES}^b(f)$ and $[\underline{D}, \underline{A}]$ as well, we have $\delta([\underline{B}, \underline{D}], c) \in \{-2, -1, 0, 1, 2\}$.

(2) Each fat black vertex $v \in G$ has $\delta(v, c) = 0$. Each vertex v of $\mathsf{RES}^b(f)$ has $\delta(v, c) \in \{-1, 0, 1\}$, because c is a cycle on $\mathsf{RES}^b(f)$. So if $\delta(v, c) = 2$, it follows that $v \in G \setminus \mathsf{RES}(f)$ and thus we have that v is a vertex $[\underline{B}, \underline{D}]$ of an f-flat rhombus $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D}) =: \Diamond$. Recall that in $\Diamond$ there are auxiliary vertices $v_1, \ldots, v_{14}$. By construction of $\mathsf{RES}^b(f)$, we have

$$2 = \delta([\underline{B}, \underline{D}], c) = c(\{v_1, v_5\}) + c(\{v_2, v_7\}) + c(\{v_3, v_9\}) + c(\{v_4, v_{11}\}).$$

The capacity constraints ensure that $c(\{v_1, v_5\}) \leq 0$ and $c(\{v_4, v_{11}\}) \geq 0$.

Assume that $c(\{v_4, v_{11}\}) > 0$. As c is a well-directed cycle, the structure of $\mathsf{RES}^b(f)$ implies that $c(\{v_2, v_7\}) \leq 0$ and $c(\{v_3, v_9\}) \leq 0$. Therefore $c(\{v_4, v_{11}\}) \geq 2$, which is a contradiction.

So we have $c(\{v_4, v_{11}\}) = 0$. This implies that $c(\{v_1, v_5\}) = 0$, $c(\{v_2, v_7\}) = 1$ and $c(\{v_3, v_9\}) = 1$, i.e. c uses all uncapacitated edges of this rhombus, from $[\underline{D}, \underline{C}]$ to $[\underline{A}, \underline{B}]$ and from $[\underline{C}, \underline{B}]$ to $[\underline{D}, \underline{A}]$ as claimed.

(3) The proof is analog to (2).

(4) From (1) we know that for each rhombus $\Diamond := \Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ we have $\sigma\big(\Diamond, c\big) \in \{-4, -3, \ldots, 3, 4\}$, because $\sigma\big(\Diamond, c\big) = \delta([\underline{D}, \underline{C}], c) - \delta([\underline{A}, \underline{B}], c)$.

Assume $\sigma\big(\Diamond, c\big) \in \{-4, 4\}$. Consider the case $\sigma\big(\Diamond, c\big) = 4$. The proof for $\sigma\big(\Diamond, c\big) = -4$ is analog. With (1) we have $\delta([\underline{D}, \underline{C}], c) = 2$ and $\delta([\underline{A}, \underline{B}], c) = -2$. Then with (2) and (3) both $\Diamond(\underline{D}, \underline{A}, ., \underline{B})$ and $\Diamond(., \underline{D}, \underline{B}, \underline{C})$ are f-flat. But as $\sigma\big(\Diamond, c\big) = \delta([\underline{D}, \underline{A}], c) - \delta([\underline{C}, \underline{B}], c)$, we also have that $\delta([\underline{D}, \underline{A}], c) = 2$ and $\delta([\underline{C}, \underline{B}], c) = -2$ and thus that both $\Diamond(\underline{B}, \underline{D}, ., \underline{A})$ and $\Diamond(., \underline{C}, \underline{D}, \underline{B})$ are f-flat. This is a contradiction, because f is shattered.

Hence $\sigma\big(\Diamond, c\big) \in \{-3, \ldots, 3\}$.

(5) As $c \in P_{\text{feas}}(\mathsf{RES}^b(f))$, the Correspondence Lemma 6.11 implies that $f + \tau'(c) \in P^b_{\text{flat} f}$. As $f + \tau'(c) \notin P^b$, the rhombus inequality of some non-f-flat rhombus must be violated in $f + \tau'(c)$. We use this to show that the set of rhombi $\Diamond$ which have $\sigma\big(\Diamond, c\big) < 0$ is not empty: As f is a hive flow, we have $\sigma\big(\Diamond, f\big) \geq 0$ on each rhombus $\Diamond$. If for a rhombus $\Diamond$ we have $\sigma\big(\Diamond, c\big) \geq 0$, then $\sigma\big(\Diamond, f + \tau'(c)\big) \geq 0$. So a rhombus inequality can only be violated by a rhombus $\Diamond$ with $\sigma\big(\Diamond, c\big) < 0$.

Let

$$\varepsilon_{\max}(\Diamond) := -\sigma\big(\Diamond, f\big) / \sigma\big(\Diamond, c\big)$$

for each rhombus $\Diamond$ that has $\sigma\big(\Diamond, c\big) < 0$. Let $\Diamond'$ be a rhombus that minimizes $\varepsilon_{\max}$ among all rhombi $\Diamond$ that have $\sigma\big(\Diamond, c\big) < 0$. We have $\sigma\big(\Diamond', f + \varepsilon_{\max}(\Diamond')c\big) = 0$ and for any $\varepsilon'' > 0$ we have $\sigma\big(\Diamond', f + (\varepsilon_{\max}(\Diamond') + \varepsilon'')c\big) < 0$ and thus $f + (\varepsilon_{\max}(\Diamond') + \varepsilon'')\tau'(c) \notin P^b$. If we show that $f + \varepsilon_{\max}(\Diamond')\tau'(c) \in P^b$, then $\varepsilon = \varepsilon_{\max}(\Diamond')$ and $\Diamond'$ is critical.

Let $\bar{\Diamond}$ be an f-flat rhombus, i.e. $\sigma\big(\bar{\Diamond}, f\big) = 0$. As $c \in P_{\text{feas}}(\mathsf{RES}^b(f))$, the Residual Correspondence Lemma 6.11 ensures that $f + \tau'(c) \in P^b_{\text{flat} f}$ and thus $\sigma\big(\bar{\Diamond}, f + \tau'(c)\big) \geq 0$. Let $\tilde{\Diamond}$ be a rhombus with $\sigma\big(\tilde{\Diamond}, f\big) > 0$. If $\sigma\big(\tilde{\Diamond}, c\big) \geq 0$, then $\sigma\big(\tilde{\Diamond}, f + \tau'(c)\big) \geq 0$. If $\sigma\big(\tilde{\Diamond}, c\big) < 0$, then $\varepsilon_{\max}(\tilde{\Diamond}) \geq \varepsilon_{\max}(\Diamond')$ and thus $\sigma\big(\tilde{\Diamond}, f + \varepsilon_{\max}(\Diamond')\tau'(c)\big) \geq 0$. Hence $\varepsilon = \varepsilon_{\max}(\Diamond')$ and $\Diamond'$ is critical.

($*$) As an auxiliary result, we prove $0 < \varepsilon < 1$:

By definition we have $\varepsilon \geq 0$. As by (5) a critical rhombus exists, we have $\varepsilon \neq 0$, because critical rhombi are not f-flat, but g-flat. By assumption we have $f + \varepsilon\tau'(c) \in P^b$ and $f + \tau'(c) \notin P^b$ and thus $\varepsilon < 1$, because P^b is a polyhedron and therefore convex. Hence $0 < \varepsilon < 1$.

(6) Let $\Diamond$ be a critical rhombus. As c is a cycle, $\sigma(\Diamond, c)$ is integral. The flow f is integral and thus $\sigma(\Diamond, f)$ is integral. As f is a hive flow, we have $\sigma(\Diamond, f) \geq 0$. We have $\sigma(\Diamond, f) \geq 1$, because $\Diamond$ is not f-flat. The rhombus $\Diamond$ is critical and thus $\sigma(\Diamond, f) + \varepsilon\sigma(\Diamond, c) = \sigma(\Diamond, f + \varepsilon c) = \sigma(\Diamond, g) = 0$. By $(*)$ we have $0 < \varepsilon < 1$ and thus $\sigma(\Diamond, c) < -1$. Hence $\sigma(\Diamond, c) \leq -2$.

(7) According to (5) there exists a critical rhombus $\Diamond$. So $\sigma(\Diamond, f) + \varepsilon\sigma(\Diamond, c) = 0$. From (6) we have $\sigma(\Diamond, c) < 0$. With $(*)$ we know that $\varepsilon < 1$ and thus $\sigma(\Diamond, f) + \sigma(\Diamond, c) < 0$.

From (4) and (6) it follows that $\sigma(\Diamond, c) \in \{-3, -2\}$. We have $\sigma(\Diamond, f) > 0$, because $\Diamond$ is not f-flat. As $\sigma(\Diamond, f)$ is integral, it follows that $(\sigma(\Diamond, f), \sigma(\Diamond, c)) \in \{(1, -2), (1, -3), (2, -3)\}$. As $\sigma(\Diamond, c) \neq 0$ and $\sigma(\Diamond, f) + \varepsilon\sigma(\Diamond, c) = 0$, we have $\varepsilon = -\sigma(\Diamond, f)/\sigma(\Diamond, c)$. Hence $\varepsilon \in \{\frac{1}{3}, \frac{1}{2}, \frac{2}{3}\}$.

(8) Let $\Diamond := \Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ be f-flat. According to (7), we have $\varepsilon > 0$ and thus we have the following equivalences: $\Diamond$ is rigid $\Leftrightarrow \sigma(\Diamond, g) = 0 \Leftrightarrow \sigma(\Diamond, f) + \varepsilon\sigma(\Diamond, c) = 0 \Leftrightarrow \varepsilon\sigma(\Diamond, c) = 0 \Leftrightarrow \sigma(\Diamond, c) = 0$. Recall that in $\Diamond$ there are auxiliary vertices $v_1, \ldots, v_{14}$. By construction of $\mathsf{RES}^b(f)$ we have $\delta([\underline{A}, \underline{B}], c) = c(\{[\underline{A}, \underline{B}], v_9\}) + c(\{[\underline{A}, \underline{B}], v_{11}\}) + c(\{[\underline{A}, \underline{B}], v_{13}\})$ and $\delta([\underline{D}, \underline{C}], c) = c(\{[\underline{D}, \underline{C}], v_6\}) + c(\{[\underline{D}, \underline{C}], v_{10}\}) + c(\{[\underline{D}, \underline{C}], v_{14}\})$. As c is a flow, we have $c(\{[\underline{A}, \underline{B}], v_9\}) = c(\{[\underline{D}, \underline{C}], v_{10}\})$. Therefore $\sigma(\Diamond, c) = \delta([\underline{D}, \underline{C}], c) - \delta([\underline{A}, \underline{B}], c) = c(\{[\underline{D}, \underline{C}], v_6\}) + c(\{[\underline{D}, \underline{C}], v_{14}\}) - c(\{[\underline{A}, \underline{B}], v_{11}\}) - c(\{[\underline{A}, \underline{B}], v_{13}\})$. Note that c is well-directed and thus we have $c(\{[\underline{D}, \underline{C}], v_6\}) \geq 0$, $c(\{[\underline{D}, \underline{C}], v_{14}\}) \geq 0$, $c(\{[\underline{A}, \underline{B}], v_{11}\}) \leq 0$ and $c(\{[\underline{D}, \underline{C}], v_{13}\}) \leq 0$. Therefore we have the following equivalence:

$$\begin{aligned}
\sigma(\Diamond, c) = 0 \quad &\Leftrightarrow \quad c(\{[\underline{D}, \underline{C}], v_6\}) = 0 \wedge c(\{[\underline{D}, \underline{C}], v_{14}\}) = 0 \\
& \quad \wedge\, c(\{[\underline{A}, \underline{B}], v_{11}\}) = 0 \wedge c(\{[\underline{A}, \underline{B}], v_{13}\}) = 0 \\
&\Leftrightarrow \quad c \text{ uses no capacitated edges in } \Diamond.
\end{aligned}$$

This proves the claim.

□

Proof outline of the Shortest Cycle Theorem 6.19

Our goal is to find a well-directed cycle c' on $\mathsf{RES}^b(f)$ with $\delta(c') > 0$ and $\ell(c') < \ell(c)$. We now introduce one main tool called the *quasi-cycle-decomposition.*

Definition 3. Given a flow $\tilde{f} \in F(G)$. A finite set of well-directed cycles $\{c_1, \ldots, c_m\}$ on $\mathsf{RES}^b(f)$ is called a *quasi-cycle-decomposition of $\tilde{f}$ into m cycles,* if $\sum_{i=1}^{m} \tau'(c_i) = \tilde{f}$. ■

The name quasi-cycle-decomposition arises from the fact that $\tau'(c_i)$ is not necessarily a cycle on G. Also note that there can be cases where for a feasible flow d on $\mathsf{RES}^b(f)$ we have $\sum_{i=1}^m \tau'(c_i) = \tau'(d)$, but $\sum_{i=1}^m c_i \neq d$.

We will see that quasi-cycle-decompositions of $\tau'(c)$ into cycles c_i exist with $\ell(c_i) < \ell(c)$ for all i or that quasi-cycle-decompositions of $\tau'(c) + f_\Psi$ into cycles c_i exist with $\ell(c_i) < \ell(c)$ for all i. The following lemma then finishes the proof of Theorem 6.19:

Lemma 4. *(1) Given a quasi-cycle-decomposition $\{c_1, \dots, c_m\}$ of $\tau'(c)$ with $m \geq 1$, then there exists $1 \leq i \leq m$ with $\delta(c_i) > 0$.*

(2) Given a g-flatspace chain Ψ and a quasi-cycle-decomposition $\{c_1, \dots, c_m\}$ of $\tau'(c) + f_\Psi$ with $m \geq 1$, then there exists $1 \leq i \leq m$ with $\delta(c_i) > 0$.

Proof. By assumption we have $\delta(\tau'(c)) > 0$.

(1) $0 < \delta(\tau'(c)) = \delta(\sum_{i=1}^m \tau'(c_i)) = \sum_{i=1}^m \delta(\tau'(c_i)) \Rightarrow \exists i \in \{1, \dots, m\} : \delta(\tau'(c_i)) > 0$.

(2) $\delta(f_\Psi) = 0 \Rightarrow \delta(\tau'(c) + f_\Psi) > 0$.
$0 < \delta(\tau'(c) + f_\Psi) = \sum_{i=1}^m \delta(\tau'(c_i)) \Rightarrow \exists i \in \{1, \dots, m\} : \delta(\tau'(c_i)) > 0$.

$\square$

In order to find a quasi-cycle-decomposition, we do a distinction of cases:

In case 1 c uses at least one capacitated edge in an f-flat rhombus $\Diamond$ and at least 3 of the circle vertices of $\Diamond$. In this case, we will easily find a quasi-cycle-decomposition of $\tau'(c)$ into one or two cycles.

In case 2 we assume the contrary, namely in each f-flat rhombus $\Diamond$ c uses either no capacitated edge at all or c uses at most 2 of the circle vertices of $\Diamond$. In this case we do again a distinction of two cases:

In case 2.1 there is a critical rhombus $\Diamond$ that is not overlapping with any other g-flat rhombus. Here it will be relatively easy to find a quasi-cycle-decomposition of $\tau'(c)$ into two cycles by analyzing $\Diamond$ and rerouting c at $\Diamond$ and its connected small triangles.

In case 2.2 all critical rhombi are overlapping with at least one g-flat rhombus. Hence there is a g-flatspace chain Ψ. We will completely classify all possible shapes of g-flatspaces and additionally see which edges are used by c and in which direction. Then we will find a quasi-cycle-decomposition of $\tau'(c) + f_\Psi$ into $m \geq 1$ cycles.

The classification of all possible g-flatspaces and the behaviour of c in g-flatspaces is a major part of this proof. Note that each rhombus in a g-flatspace must either be critical or rigid. As f is shattered, rigid rhombi cannot overlap. Critical rhombi on the other hand can overlap in certain situations. We will distinguish g-flatspaces with overlapping critical rhombi and those without and analyze both situations independently.

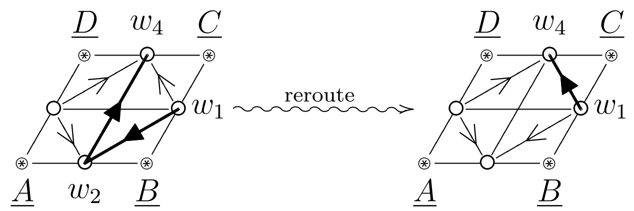

Figure 6.20: Shortest cycles only use direct paths. The f-flat rhombus is drawn in short notation and cycles are represented by fat arrows here as in all upcoming figures.

We now start with considering the first case.

Case 1:

Assumption: The cycle c uses at least one capacitated edge in an f-flat rhombus $\lozenge$ and at least 3 of the circle vertices of $\lozenge$.

We can handle this case with Lemma 4 and the following lemma:

Lemma 5. *There is a quasi-cycle-decomposition of $\tau'(c)$ into one or two cycles that are each shorter than c.*

Proof. Given a rhombus $\lozenge := \lozenge(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ such that c uses at least one capacitated edge in $\lozenge$ and at least 3 of its 4 circle vertices. Each vertex can only appear once in c. Thus c uses 3 or 4 successive circle vertices (case (a)) or c uses 4 circle vertices and two at a time are successive circle vertices (case (b)).

(a) c uses 3 or 4 circle vertices (denoted with $w_1, w_2, [w_3,]w_4$) that are successive circle vertices in c (see Figure 6.20). We get c' by doing local changes to c such that c' uses the direct path from w_1 to w_4 and omits w_2 [and w_3]. It is easy to check that c' is still well-directed and that these local changes preserve the throughput on all circle vertices. Thus $\{c'\}$ is a quasi-cycle-decomposition of $\tau'(c)$ into 1 cycle. The fact that $\ell(c') < \ell(c)$ finishes the proof.

(b) c uses 4 circle vertices that are not successive circle vertices in c (see Figure 6.21). W.l.o.g. let c use $[\underline{D}, \underline{A}]$ and the vertices on the direct path to $[\underline{A}, \underline{B}]$, then a set of vertices V_1, then $[\underline{C}, \underline{B}]$ and the vertices on the direct path to $[\underline{D}, \underline{C}]$ and then a set of vertices V_2. Then $\tau'(c) = \tau'(c_1) + \tau'(c_2)$ where c_1 and c_2 are two well-directed cycles on $\mathsf{RES}^b(f)$: c_1 uses $[\underline{C}, \underline{B}]$, the vertices on the direct path to $[\underline{A}, \underline{B}]$ and V_2. The cycle c_2 uses $[\underline{D}, \underline{A}]$, the vertices on the direct path to $[\underline{D}, \underline{C}]$ and V_1. So $\{c_1, c_2\}$ is a quasi-cycle-decomposition of $\tau'(c)$ into 2 cycles. Note that c, c_1 and c_2 each use exactly

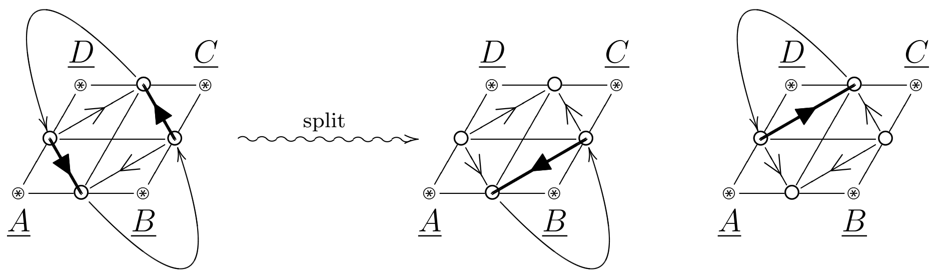

Figure 6.21: Shortest cycles do not pass a flat rhombus twice.

4 edges in $\Diamond$. So $\ell(c_1) = 4 + |V_1| < 4 + |V_1| + |V_2| = \ell(c)$ and analogously $\ell(c_2) < \ell(c)$.

$\square$

Case 2:

Assumption: In each f-flat rhombus $\Diamond$ c uses either no capacitated edge at all or c uses at most 2 of the circle vertices of $\Diamond$.

The assumption leads to the following observation concerning bending rhombi:

Lemma 6. *In each bending rhombus $\Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$, c uses exactly one of the four direct paths that have capacitated edges and c uses no uncapacitated edge. Moreover, we have $\delta([\underline{B}, \underline{D}], c) \in \{-1, 0, 1\}$.*

Proof. Let $\Diamond := \Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ be bending. Then by Lemma 2(8), c uses capacitated edges in $\Diamond$ and c uses only 2 of the four circle vertices in $\Diamond$ by the assumption of case 2. Therefore c uses exactly one of the 4 direct paths in $\Diamond$ that have capacitated edges and c uses no uncapacitated edge in $\Diamond$. If c uses the direct path from $[\underline{D}, \underline{A}]$ to $[\underline{A}, \underline{B}]$ or if c uses the direct path from $[\underline{C}, \underline{B}]$ to $[\underline{D}, \underline{C}]$, then $\delta([\underline{B}, \underline{D}], c) = 0$. If c uses the direct path from $[\underline{D}, \underline{A}]$ to $[\underline{D}, \underline{C}]$, then $\delta([\underline{B}, \underline{D}], c) = -1$ and if c uses the direct path from $[\underline{C}, \underline{B}]$ to $[\underline{A}, \underline{B}]$, then $\delta([\underline{B}, \underline{D}], c) = 1$. $\square$

Case 2.1:

Assumption: There exists a critical rhombus $\Diamond$ that is not overlapping with any other g-flat rhombi.

We can handle this case with Lemma 4 and the following lemma:

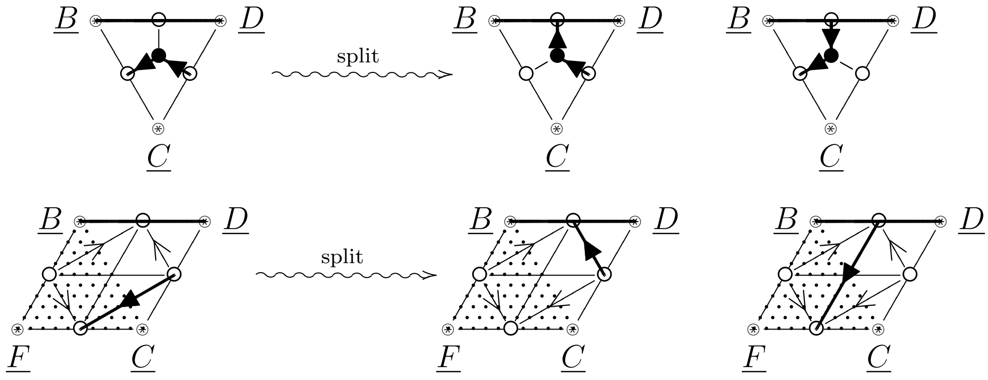

Figure 6.22: From c we obtain two cycles. This replacement operation is used twice, once on each side of the fat line from $\underline{B}$ to $\underline{D}$ using rotational symmetry. The new cycles both use $[\underline{B}, \underline{D}]$. The small triangle that does not belong to the g-flat rhombus is dotted.

Lemma 7. *There is a quasi-cycle-decomposition* $\{c_1, c_2\}$ *of* $\tau'(c)$ *into two cycles with* $\ell(c_1) < \ell(c)$ *and* $\ell(c_2) < \ell(c)$.

Proof. Let $\Diamond := \Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ be a critical rhombus that is not overlapping with any other g-flat rhombi. By Lemma 2(6) we have $\delta([\underline{D}, \underline{C}], c) - \delta([\underline{A}, \underline{B}], c) = \sigma(\Diamond, c) \leq -2$. By assumption of case 2.1, $\Diamond$ cannot overlap with other g-flat rhombi. Depending on their f-flatness, a rhombus overlapping with $\Diamond$ can be either bending or loose. If $\Diamond(., \underline{D}, \underline{B}, \underline{C})$ is loose, then $[\underline{D}, \underline{C}]$ is a vertex of $\mathsf{RES}^b(f)$ and therefore $\delta([\underline{D}, \underline{C}], c) \in \{-1, 0, 1\}$. If $\Diamond(., \underline{D}, \underline{B}, \underline{C})$ is bending, then by Lemma 6, we have $\delta([\underline{D}, \underline{C}], c) \in \{-1, 0, 1\}$. Using rotational symmetry, we get $\delta([\underline{A}, \underline{B}], c) \in \{-1, 0, 1\}$ as well. As $\sigma(\Diamond, c) \leq -2$, we have $\delta([\underline{D}, \underline{C}], c) = -1$ and $\delta([\underline{A}, \underline{B}], c) = 1$. With $\sigma(\Diamond, c) = \delta([\underline{D}, \underline{A}], c) - \delta([\underline{C}, \underline{B}], c)$ the same argument can be used to show that $\delta([\underline{D}, \underline{A}], c) = -1$ and $\delta([\underline{C}, \underline{B}], c) = 1$. On the left side of the "split" arrows in Figure 6.22 the possible cases of parts of g-flatspaces are depicted up to rotational and mirror symmetry and a g-flat rhombus results from glueing two parts together at the fat line from $\underline{B}$ to $\underline{D}$. We get the mirror symmetric situations by mirroring the figure and reversing the directions of all arrows including the fat ones. Let R denote the unification of $\Diamond$ with its overlapping f-flat rhombi. Then c uses vertices in R, then a set of vertices V_1, then again vertices in R and then a set of vertices V_2. We do local changes to c once on each side of the line from $\underline{B}$ to $\underline{D}$ as seen in Figure 6.22 and obtain two well-directed cycles c_1 and c_2 from which c_1 uses vertices on R and V_1 and c_2 uses vertices on R and V_2. See Figure 6.23 for an example. Both c_1 and c_2 use the vertex $[\underline{B}, \underline{D}]$ with $\delta([\underline{B}, \underline{D}], c_1) = 1$ and $\delta([\underline{B}, \underline{D}], c_2) = -1$. Note that c, c_1 and c_2 use exactly k edges in R for some $k \in \mathbb{N}$. Therefore $\ell(c) = |V_1| + |V_2| + k > |V_1| + k = \ell(c_1)$

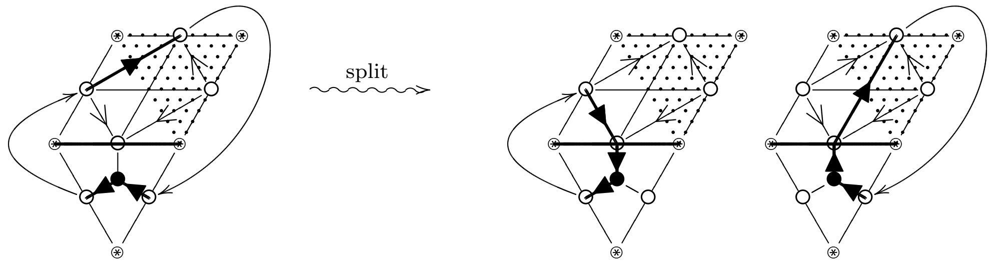

Figure 6.23: A situation where the cycle c is split up into shorter cycles c_1 and c_2 such that $\tau'(c) = \tau'(c_1)+\tau'(c_2)$. The small triangle that does not belong to the g-flat rhombus is dotted.

and analogously $\ell(c) > \ell(c_2)$. It is easy to check that on all circle vertices $v \in G$, we have $\delta(v, c) = \delta(v, c_1)+\delta(v, c_2)$ and thus $\tau'(c) = \tau'(c_1)+\tau'(c_2)$. Hence $\{c_1, c_2\}$ is a quasi-cycle-decomposition of $\tau'(c)$ into two cycles. $\square$

Case 2.2:

Assumption: Each critical rhombus is overlapping with at least one g-flat rhombus.

By Lemma 2(5) there exists at least one critical rhombus, which implies that g cannot be shattered. So there must be big g-flatspaces. We will classify all shapes of g-flatspaces up to rotational and mirror symmetry. We begin by proving the following auxiliary result:

Lemma 8 (Correct direction of c in critical rhombi)**.** *Given a critical rhombus* $\Diamond := \Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$*. We have* $\delta([\underline{A}, \underline{B}], c) > -1$ *and* $\delta([\underline{C}, \underline{B}], c) > -1$ *and* $\delta([\underline{D}, \underline{C}], c) < 1$ *and* $\delta([\underline{D}, \underline{A}], c) < 1$*.*

Proof. We only prove the first statement. The other three cases are analogous. We show that if $\delta([\underline{A}, \underline{B}], c) \le -1$, then $\Diamond$ is not critical: Let $\delta([\underline{A}, \underline{B}], c) \le -1$. Then we have $\sigma(\Diamond, c) = \delta([\underline{D}, \underline{C}], c) - \delta([\underline{A}, \underline{B}], c) \le \delta([\underline{D}, \underline{C}], c) + 1$. But according to Lemma 2(1), we have $-2 \le \delta([\underline{D}, \underline{C}], c) \le 2$ and thus $\sigma(\Diamond, c) \ge -1$. Lemma 2(6) then states that $\Diamond$ is not critical. $\square$

Lemma 8 will help to classify the shapes of g-flatspaces. We do the following distinction:

Definition 9. A g-flatspace that contains two overlapping critical rhombi is called *special*. Otherwise it is called *ordinary*. ■

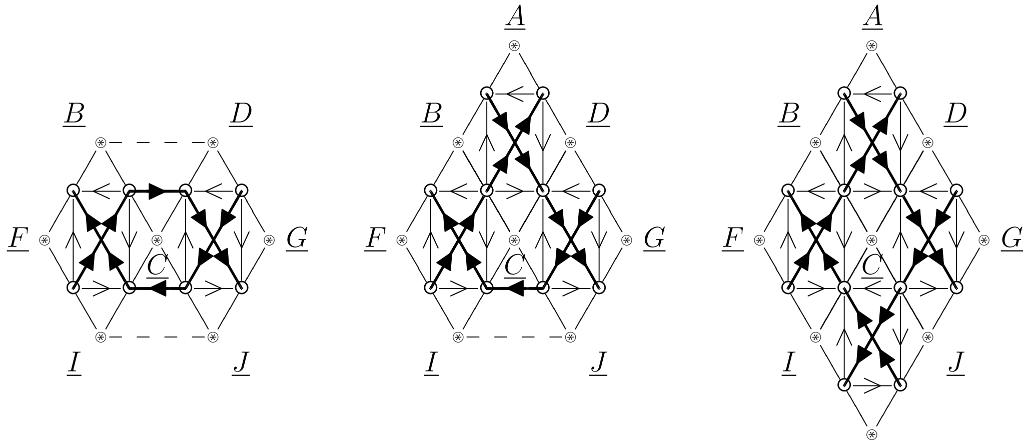

Figure 6.24: Three possibilities for g-flatspaces (hexagon, pentagon, big rhombus). The rhombi $\Diamond(\underline{F},\underline{C},\underline{D},\underline{B})$ and $\Diamond(\underline{G},\underline{D},\underline{B},\underline{C})$ are overlapping critical rhombi. A dashed line through a rhombus indicates that this rhombus can be either bending or loose.

We start with analyzing special g-flatspaces and then continue with ordinary g-flatspaces.

Lemma 10. *If a g-flatspace is special, then it is a hexagon with side lengths (1,1,1,1,1,1), a pentagon with side lengths (1,1,1,2,2), or a big rhombus with side lengths (2,2,2,2). Moreover, in each of these cases, the only possibilities for c in this flatspace are the ones illustrated in Figure 6.24.*

Proof. Let the vertices $\underline{A},\dots,\underline{J}$ be arranged as in Figure 6.24. Let $\Diamond(\underline{F},\underline{C},\underline{D},\underline{B})$ and $\Diamond(\underline{G},\underline{D},\underline{B},\underline{C})$ both be critical rhombi. The g-flatspace these two overlapping critical rhombi lie in is denoted by R. According to Lemma 2(6), we have $\delta([\underline{B},\underline{D}],c) - \delta([\underline{F},\underline{C}],c) = \sigma\big(\Diamond(\underline{F},\underline{C},\underline{D},\underline{B}),c\big) \leq -2$ and $\delta([\underline{C},\underline{G}],c) - \delta([\underline{B},\underline{D}],c) = \sigma\big(\Diamond(\underline{G},\underline{D},\underline{B},\underline{C}),c\big) \leq -2$. Therefore $\delta([\underline{B},\underline{D}],c) = 0$, $\delta([\underline{C},\underline{G}],c) = -2$ and $\delta([\underline{F},\underline{C}],c) = 2$. Lemma 2(2), Lemma 2(3) and Lemma 2(8) imply that both $\Diamond(\underline{D},\underline{C},\underline{J},\underline{G})$ and $\Diamond(\underline{B},\underline{F},\underline{I},\underline{C})$ are rigid and c uses all uncapacitated edges of these rhombi in only one possible way, which is illustrated in Figure 6.25. Due to the fact that flatspaces are convex, R spawns at least a hexagon of side length 1 and $\underline{C}$ is an inner vertex of R. So $\Diamond(\underline{J},\underline{C},\underline{F},\underline{I})$ and $\Diamond(\underline{I},\underline{J},\underline{G},\underline{C})$ are overlapping critical rhombi as well. Now Lemma 8 implies that $\Diamond(\underline{C},\underline{B},.,\underline{F})$ is not critical, because $\delta([\underline{C},\underline{B}],c) = -1$. Analogously, we get that $\Diamond(\underline{C},\underline{G},.,\underline{D})$, $\Diamond(.,\underline{I},\underline{C},\underline{F})$ and $\Diamond(.,\underline{G},\underline{C},\underline{J})$ are not critical. As neither of those 4 rhombi is rigid, because f is shattered, the border of R contains the vertices $\underline{I},\underline{F},\underline{B},\underline{D},\underline{G}$ and $\underline{J}$. So up to rotational symmetry there are 3 cases:

(1) Neither $\Diamond(\underline{A},\underline{B},\underline{C},\underline{D})$ nor $\Diamond(\underline{C},\underline{I},.,\underline{J})$ is g-flat.

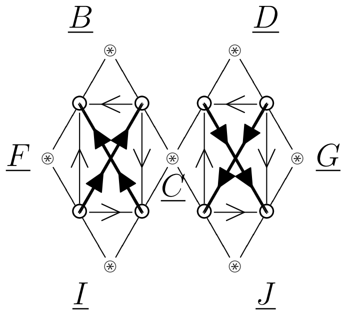

Figure 6.25: Situation where $\Diamond(\underline{F},\underline{C},\underline{D},\underline{B})$ and $\Diamond(\underline{G},\underline{D},\underline{B},\underline{C})$ are critical.

(2) $\Diamond(\underline{A},\underline{B},\underline{C},\underline{D})$ is g-flat, but $\Diamond(\underline{C},\underline{I},.,\underline{J})$ is not g-flat.

(3) Both $\Diamond(\underline{A},\underline{B},\underline{C},\underline{D})$ and $\Diamond(\underline{C},\underline{I},.,\underline{J})$ are g-flat.

So the shape of R is either a hexagon with side lengths $(1,1,1,1,1,1)$, a pentagon with side lengths $(1,1,1,2,2)$ or a big rhombus with side lengths $(2,2,2,2)$.

We show next that in each of these cases the only possibilities for c in this flatspace are those illustrated in Figure 6.24. We distinguish the cases where $\Diamond := \Diamond(\underline{A},\underline{B},\underline{C},\underline{D})$ is critical, rigid, loose or bending.

As $\delta([\underline{C},\underline{B}],c) = -1$, Lemma 8 prohibits that $\Diamond$ is critical.

If $\Diamond$ is rigid, then from $\delta([\underline{C},\underline{B}],c) = -1$ and $\delta([\underline{D},\underline{C}],c) = 1$ it follows with Lemma 2(8) that c must use all uncapacitated edges from $[\underline{C},\underline{B}]$ to $[\underline{D},\underline{A}]$ and from $[\underline{D},\underline{C}]$ to $[\underline{A},\underline{B}]$.

If $\Diamond$ is loose, then from $\delta([\underline{C},\underline{B}],c) = -1$ and $\delta([\underline{D},\underline{C}],c) = 1$ it follows that c must use the two edges from $[\underline{C},\underline{B}]$ to $[\underline{D},\underline{C}]$.

If $\Diamond$ is bending, then according to Lemma 2(8) c uses capacitated edges in $\Diamond$. By Lemma 6 c uses no uncapacitated edge and exactly one direct path in $\Diamond$. As we know already that $\delta([\underline{C},\underline{B}],c) = -1$ and $\delta([\underline{D},\underline{C}],c) = 1$, c must use the two edges on the direct path from $[\underline{C},\underline{B}]$ to $[\underline{D},\underline{C}]$.

Hence in each case there is only one possibility for c. We have analog results for $\Diamond(\underline{C},\underline{I},.,\underline{J})$. Note that these are the results illustrated in Figure 6.24. □

We have fully classified all special g-flatspaces and described c on these. We now classify the ordinary g-flatspaces by proving several restrictions on their shape.

Definition 11. A pair of critical rhombi that have the same orientation ('$\Diamond$', '▱' or '⏥') and which are both overlapping with the same f-flat rhombus is called a *forbidden pair of nearby critical rhombi.* ■

Lemma 12. *We have the following restrictions on g-flatspaces:*

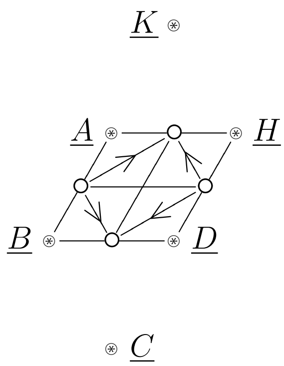

Figure 6.26: Forbidden pairs of nearby critical rhombi $\lozenge(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ and $\lozenge(\underline{K}, \underline{A}, \underline{D}, \underline{H})$ are impossible.

(1) Ordinary g-flatspaces do not contain three rhombi that share a small triangle.

(2) There is no forbidden pair of nearby critical rhombi.

(3) There are no ordinary g-flatspaces that have a side with length greater than 2.

Proof. (1) Note that the only situation in which three rhombi share a small triangle is when the three rhombi form a triangle of side lengths (2,2,2).

Each rhombus in a g-flatspace must be either rigid or critical. Rigid rhombi cannot overlap, because f is shattered. By Definition 9 ordinary g-flatspaces contain no overlapping critical rhombi. But we cannot assign rigidity and criticality to three rhombi that share a small triangle in a way that we have no pair of overlapping rigid rhombi and no pair of overlapping critical rhombi.

(2) Let $\underline{A}, \underline{B}, \underline{C}, \underline{D}, \underline{H}, \underline{K}$ as in Figure 6.26. Assume that $\lozenge(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ and $\lozenge(\underline{K}, \underline{A}, \underline{D}, \underline{H})$ are critical and $\lozenge(\underline{B}, \underline{D}, \underline{H}, \underline{A})$ is f-flat. By Lemma 2(6), we have $\delta([\underline{D}, \underline{A}], c) - \delta([\underline{C}, \underline{B}], c) \leq -2$ and $\delta([\underline{H}, \underline{K}]) - \delta([\underline{D}, \underline{A}]) \leq -2$. Lemma 2(1) implies that $\delta([\underline{D}, \underline{A}], c) = 0$, $\delta([\underline{H}, \underline{K}], c) = -2$ and $\delta([\underline{D}, \underline{A}], c) = 2$. With Lemma 2(2) and Lemma 2(3) this results in $\lozenge(., \underline{C}, \underline{D}, \underline{B})$ and $\lozenge(\underline{A}, \underline{H}, ., \underline{K})$ being f-flat and c using the uncapacitated edges to $[\underline{A}, \underline{H}]$ and $[\underline{B}, \underline{D}]$. Thus $\delta([\underline{A}, \underline{H}], c) = -1$ and $\delta([\underline{B}, \underline{D}], c) = 1$. The edge capacities in $\lozenge(\underline{B}, \underline{D}, \underline{H}, \underline{A})$ force c to use the direct path from $[\underline{B}, \underline{D}]$ to $[\underline{A}, \underline{H}]$. But this implies $\delta([\underline{A}, \underline{H}], c) = 1$, which is a contradiction.

The same proof can be applied in all mirrored and rotated settings.

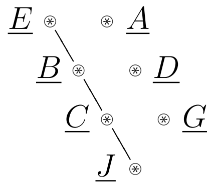

Figure 6.27: The arrangement of vertices in Lemma 12(3).

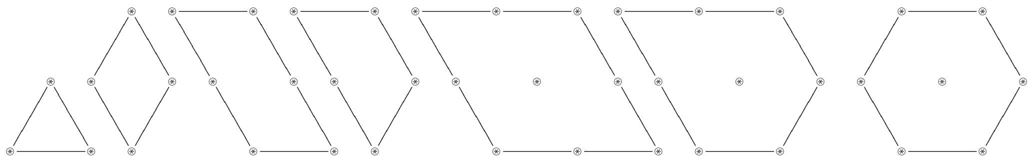

Figure 6.28: The possible shapes of g-flatspaces.

(3) Assume that the vertices $\underline{E}, \underline{B}, \underline{C}, \underline{J}$ are part of a side of an ordinary g-flatspace of length greater or equal to 3 as illustrated in Figure 6.27. Let other vertices of this g-flatspace be denoted with $\underline{A}, \underline{D}$ and $\underline{G}$ and w.l.o.g. be arranged as in Figure 6.27. The rhombi $\Diamond_1 := \Diamond(\underline{D}, \underline{C}, \underline{J}, \underline{G})$, $\Diamond_2 := \Diamond(\underline{G}, \underline{D}, \underline{B}, \underline{C})$, $\Diamond_3 := \Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ and $\Diamond_4 := \Diamond(\underline{D}, \underline{A}, \underline{E}, \underline{B})$ each must be g-flat and thus either rigid or critical. Rigid rhombi cannot overlap, because f is shattered. By Definition 9 ordinary g-flatspaces contain no overlapping critical rhombi. There are only two possibilities to assign rigidity and criticality to $\Diamond_1, \ldots, \Diamond_4$ that respect these rules for overlapping rhombi:

(a) $\Diamond_1$ and $\Diamond_3$ are critical and $\Diamond_2$ and $\Diamond_4$ are rigid.

(b) $\Diamond_2$ and $\Diamond_4$ are critical and $\Diamond_1$ and $\Diamond_3$ are rigid.

In case (a) the rhombi $\Diamond_1$ and $\Diamond_3$ form a forbidden pair of nearby critical rhombi and in case (b) the rhombi $\Diamond_2$ and $\Diamond_4$ form a forbidden pair of nearby critical rhombi. This is a contradiction to (2).

□

We can now fully classify the possible shapes of g-flatspaces. We note that we do not state that all *possible shapes* can really occur. We only say that all *other* shapes can *not* occur.

Lemma 13. *The possible shapes of g-flatspaces are exactly those depicted in Figure 6.28 up to rotational and mirror symmetry. The pentagon and the big rhombus are special. The hexagon can be special or ordinary. All other shapes are ordinary.*

Proof. Recall that Lemma 10 classifies all possible shapes of special g-flatspaces: There are only hexagons, pentagons big rhombi. For the ordinary shapes we now go through all possible shapes (cp. Figure 6.9):

There can be ordinary small triangles with side lengths $(1, 1, 1)$, but not with side lengths $(2, 2, 2)$ or larger because of Lemma 12(1).

There can be ordinary parallelograms with side lengths $(1, 1, 1, 1)$ and there can be ordinary big parallelograms with side lengths $(2, 1, 2, 1)$. Larger parallelograms are prohibited because of Lemma 12(1) and Lemma 12(3).

There can be ordinary trapezoids of side lengths $(1, 1, 1, 2)$, but larger trapezoids are prohibited because of Lemma 12(3).

Ordinary pentagons are prohibited because of Lemma 12(1).

There can be ordinary hexagons of side lengths $(1, 1, 1, 1, 1, 1)$, but larger hexagons are prohibited because of Lemma 12(1). □

We now want to analyze c on big g-flatspaces. Note that Lemma 10 already describes c on special g-flatspaces.

Lemma 14. *In each possible ordinary big g-flatspace, f-flatness and non-f-flatness are assigned to the contained rhombi as on the left side of the "reroute" arrows in Figure 6.29 and Figure 6.30 up to rotational and mirror symmetry. Moreover, the only possibilities for c in ordinary big g-flatspaces are the cases depicted in these figures on the left side of the "reroute" arrows.*

Proof. First consider a parallelogram of vertices $\underline{E}, \underline{B}, \underline{C}, \underline{G}, \underline{D}, \underline{A}$ as in Figure 6.30 (VI). There is only one possible way to assign f-flatness and non-f-flatness to the g-flat rhombi that avoids forbidden pairs of nearby critical rhombi: $\lozenge(\underline{D}, \underline{A}, \underline{E}, \underline{B})$ and $\lozenge(\underline{G}, \underline{D}, \underline{B}, \underline{C})$ are rigid and $\lozenge(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ is critical. As $\lozenge(\underline{A}, \underline{B}, \underline{C}, \underline{D})$ is critical, according to Lemma 2(6) we have $\delta([\underline{D}, \underline{A}], c) - \delta([\underline{C}, \underline{B}], c) \leq -2$. As $[\underline{D}, \underline{A}]$ is a vertex of $\mathsf{RES}^b(f)$ and $[\underline{C}, \underline{B}]$ is a vertex of $\mathsf{RES}^b(f)$ and c is a cycle on $\mathsf{RES}^b(f)$, we have $\delta([\underline{D}, \underline{A}], c) = -1$ and $\delta([\underline{C}, \underline{B}], c) = 1$. As $\lozenge(\underline{D}, \underline{A}, \underline{E}, \underline{B})$ and $\lozenge(\underline{G}, \underline{D}, \underline{B}, \underline{C})$ are rigid, according to Lemma 2(8) c must use the uncapacitated edges on the direct paths from $[\underline{B}, \underline{E}]$ to $[\underline{D}, \underline{A}]$ and from $[\underline{G}, \underline{D}]$ to $[\underline{C}, \underline{B}]$. c may use the other uncapacitated edges from $[\underline{E}, \underline{A}]$ to $[\underline{C}, \underline{G}]$ as well in any direction.

Now consider the trapezoid of vertices $\underline{F}, \underline{C}, \underline{G}, \underline{D}, \underline{B}$ as in Figure 6.29(I)-(V). Up to mirror symmetry there is only one possible way to assign f-flatness and non-f-flatness to the g-flat rhombi: $\lozenge(\underline{F}, \underline{C}, \underline{D}, \underline{B})$ is rigid and $\lozenge(\underline{G}, \underline{D}, \underline{B}, \underline{C})$ is critical. Thus $[\underline{B}, \underline{D}]$ is a vertex of $\mathsf{RES}^b(f)$ and $\delta([\underline{B}, \underline{D}]) \in \{-1, 0, 1\}$. If $\lozenge(\underline{D}, \underline{C}, ., \underline{G})$ is loose, then $[\underline{C}, \underline{G}]$ is a vertex of $\mathsf{RES}^b(f)$ and $\delta([\underline{C}, \underline{G}], c) \in \{-1, 0, 1\}$. If $\lozenge(\underline{D}, \underline{C}, ., \underline{G})$ is bending, then by Lemma 6 we have $\delta([\underline{C}, \underline{G}], c) \in \{-1, 0, 1\}$. As $\lozenge(\underline{G}, \underline{D}, \underline{B}, \underline{C})$ is critical, according to Lemma 2(6) we have $\delta([\underline{C}, \underline{G}], c) - \delta([\underline{B}, \underline{D}], c) \leq -2$. Thus $\delta([\underline{C}, \underline{G}], c) = -1$ and $\delta([\underline{B}, \underline{D}], c) = 1$. As $\lozenge(\underline{F}, \underline{C}, \underline{D}, \underline{B})$ is rigid, according to Lemma 2(8) c cannot use capacitated edges in $\lozenge(\underline{F}, \underline{C}, \underline{D}, \underline{B})$. Thus c uses the uncapacitated edges on the direct path from $[\underline{F}, \underline{C}]$ to $[\underline{B}, \underline{D}]$,

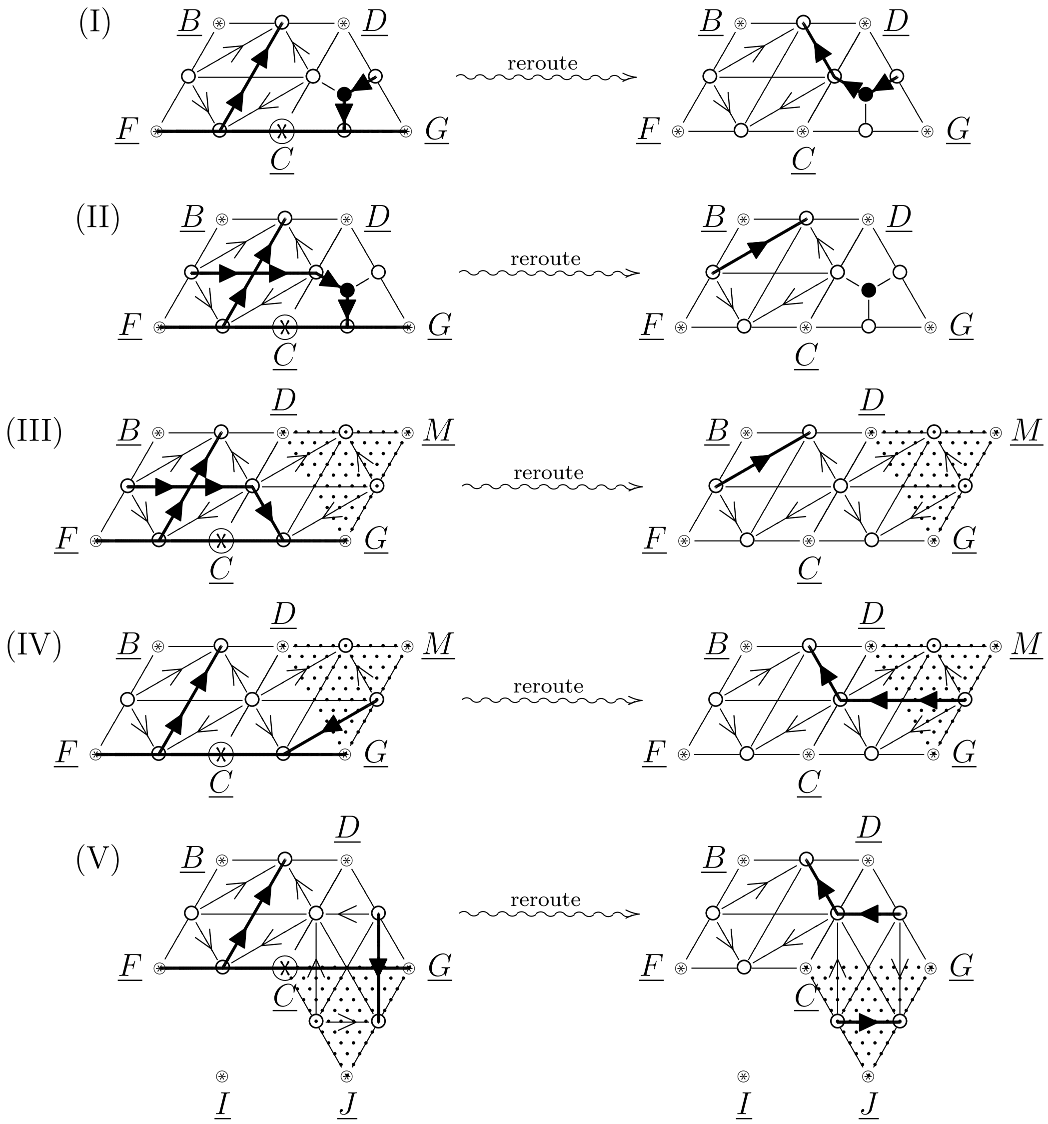

Figure 6.29: Rerouting c in all possible ordinary *trapezoid* cases up to rotation and mirroring. f-flat rhombi are drawn in short notation. The fat lines indicate where the g-flatspace chain is glued together. Inner vertices of g-flatspace chains are drawn bigger than others. Small triangles that do not belong to the g-flatspace are dotted. They are part of bending rhombi.

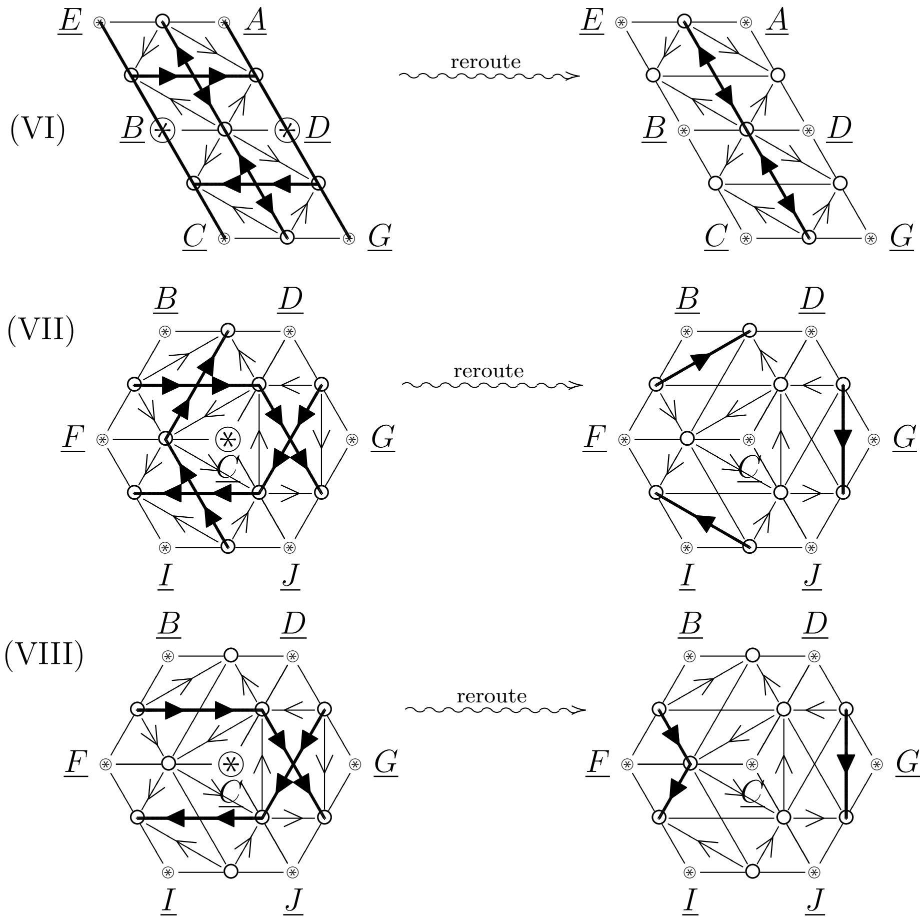

Figure 6.30: Rerouting c in all possible ordinary *non-trapezoid* cases up to rotation and mirroring, namely the cases of a parallelogram and a hexagon. f-flat rhombi are drawn in short notation. The fat lines indicate where the g-flatspace chain is glued together. Hexagons are g-flatspace chains on their own and are not glued together with other g-flatspaces. Inner vertices of g-flatspace chains are drawn bigger than others. The up-down-arrows in case (VI) indicate that c may use the uncapacitated edges from $[\underline{C}, \underline{G}]$ over $[\underline{B}, \underline{D}]$ to $[\underline{E}, \underline{A}]$ in any direction or c may not use them at all.

because $\delta([\underline{B},\underline{D}],c)=1$. Note that $\Diamond(\underline{D},\underline{C},.,\underline{G}) =: \Diamond_1$ and $\Diamond(\underline{C},\underline{G},.,\underline{D}) =: \Diamond_2$ can be bending or loose. They cannot both be bending, because f is shattered. So there are the following cases:

- $\Diamond_1$ and $\Diamond_2$ are both loose, i.e. the triangle $(\underline{D},\underline{C},\underline{G})$ is an f-flatspace. Then there are two possibilities: (I): $\delta([\underline{G},\underline{D}],c)=1$ or (II): $\delta([\underline{B},\underline{F}],c)=1$, illustrated in Figure 6.29.

- $\Diamond_1$ is loose and $\Diamond_2$ is bending. Then there are two possibilities: (III): $\delta([\underline{B},\underline{F}],c)=1$ or (IV): $\delta([\underline{G},\underline{D}],c)=1$, illustrated in Figure 6.29.

- $\Diamond_1$ is bending and $\Diamond_2$ is loose. Because of the edge capacities and Lemma 2(8) there is only one possibility to have $\delta([\underline{C},\underline{G}])=-1$: $\delta([\underline{G},\underline{D}],c)=1$ as illustrated in Figure 6.29 (V).

Now consider a hexagon with border vertices $\underline{D},\underline{B},\underline{F},\underline{I},\underline{J},\underline{G}$ and inner vertex $\underline{C}$ as in Figure 6.30(VII) and (VIII). There is only one possibility up to rotational symmetry to assign f-flatness and non-f-flatness to the hexagon's g-flat rhombi that avoids overlapping critical rhombi: $\Diamond(\underline{F},\underline{C},\underline{D},\underline{B})$, $\Diamond(\underline{J},\underline{C},\underline{F},\underline{I})$ and $\Diamond(\underline{D},\underline{C},\underline{J},\underline{G})$ are rigid and the other three rhombi $\Diamond(\underline{G},\underline{D},\underline{B},\underline{C})$, $\Diamond(\underline{C},\underline{B},\underline{F},\underline{I})$ and $\Diamond(\underline{I},\underline{J},\underline{G},\underline{C})$ are critical. As $\Diamond(\underline{G},\underline{D},\underline{B},\underline{C})$ is critical, according to Lemma 2(6) we have $\delta([\underline{C},\underline{G}],c)-\delta([\underline{B},\underline{D}],c)\le -2$. We have $\delta([\underline{B},\underline{D}],c)\in\{-1,0,1\}$, because $[\underline{B},\underline{D}]$ is a vertex of $\mathsf{RES}^b(f)$. Using Lemma 2(1) we have $\big(\delta([\underline{C},\underline{G}],c),\delta([\underline{B},\underline{D}],c)\big)\in\{(-2,1),(-1,1),(-2,0)\}$. Note that according to Lemma 2(8), c may use only uncapacitated edges in the whole hexagon.

If $\delta([\underline{B},\underline{D}],c)=1$ (see Figure 6.30 (VII)), then c uses the uncapacitated edges from $[\underline{I},\underline{J}]$ over $[\underline{F},\underline{C}]$ to $[\underline{B},\underline{D}]$. If $\delta([\underline{B},\underline{D}],c)=0$ (see Figure 6.30 (VIII)), then c does not use uncapacitated edges on the path from $[\underline{I},\underline{J}]$ over $[\underline{F},\underline{C}]$ to $[\underline{B},\underline{D}]$. If $\delta([\underline{C},\underline{G}],c)=-2$, then Lemma 2(3) states that c uses the direct paths from $[\underline{D},\underline{C}]$ to $[\underline{G},\underline{J}]$ and from $[\underline{G},\underline{D}]$ to $[\underline{J},\underline{C}]$, which results in c using additionally the direct paths from $[\underline{B},\underline{F}]$ to $[\underline{D},\underline{C}]$ and from $[\underline{J},\underline{C}]$ to $[\underline{I},\underline{F}]$.

This describes the cases where $\big(\delta([\underline{C},\underline{G}],c),\delta([\underline{B},\underline{D}],c)\big)\in\{(-2,0),(-2,1)\}$. If $\delta([\underline{C},\underline{G}],c)=-1$, then there are two cases: Either c uses the direct paths from $[\underline{B},\underline{F}]$ over $[\underline{D},\underline{C}]$ to $[\underline{G},\underline{J}]$ or c uses the direct paths from $[\underline{G},\underline{D}]$ over $[\underline{J},\underline{C}]$ to $[\underline{I},\underline{F}]$. These two cases for $\big(\delta([\underline{C},\underline{G}],c),\delta([\underline{B},\underline{D}],c)\big)=(-1,1)$ are rotationally symmetric to the case $\big(\delta([\underline{C},\underline{G}],c),\delta([\underline{B},\underline{D}],c)\big)=(-2,0)$. So up to rotational symmetry, we have the two cases depicted in Figure 6.30. □

We now want to find a quasi-cycle-decomposition of $\tau'(c)+f_\Psi$ into cycles that are shorter than c for a g-flatspace chain Ψ.

Recall that $\mathsf{RES}(f)$ and $\mathsf{RES}^b(f)$ have the same edge set E_{RES}. Note that the domain of the map $\tau' : F(\mathsf{RES}(f))\to F(G)$ can be extended from only the flows on $\mathsf{RES}(f)$ to all mappings $E_{\mathsf{RES}}\to\mathbb{R}$ and that τ' preserves $\delta_{\text{in}}(v)$ and $\delta_{\text{out}}(v)$ on each circle vertex v of $\mathsf{RES}(f)$.

Lemma 15. *Given a map $d : E_{\mathsf{RES}} \to \mathbb{R}$ such that for each f-flat rhombus $\Diamond$ we have $\delta_{in}(v_i, d) = \delta_{out}(v_i, d)$ on all auxiliary vertices $v_1, \ldots, v_{14}$ of $\Diamond$. If $\tau'(d)$ is a flow on G, then d is a flow on $\mathsf{RES}(f)$.*

Proof. The flow constraints for d are satisfied on each auxiliary vertex by assumption. The flow constraints on each fat black vertex are satisfied, because the restriction of τ' to edges adjacent to fat black vertices is the identity function. Let v be a circle vertex of $\mathsf{RES}(f)$. Then

$$\delta_{\text{in}}^{\mathsf{RES}(f)}(v, d) = \delta_{\text{in}}^{G}(v, \tau'(d)) = \delta_{\text{out}}^{G}(v, \tau'(d)) = \delta_{\text{out}}^{\mathsf{RES}(f)}(v, d)$$

and thus d is a flow on $\mathsf{RES}(f)$. □

Lemma 4 and the following lemma finish the proof:

Lemma 16. *There is a quasi-cycle-decomposition $\{c_1, \ldots, c_m\}$ of $\tau'(c) + f_\Psi$ into $m \geq 1$ cycles with $\ell(c_i) < \ell(c)$ for all $1 \leq i \leq m$ where Ψ is a g-flatspace chain.*

Proof. Let Ψ be a g-flatspace chain. Then for each g-flatspace of Ψ we want to apply to c the local changes depicted in Figure 6.29, Figure 6.30 and Figure 6.31. In most cases $\big($(I), (II) and (VI) up to (XI)$\big)$, the depicted region equals a g-flatspace. In the cases (III), (IV) and (V) the depicted region spans one extra small triangle $(\underline{D}, \underline{G}, \underline{M})$ or $(\underline{C}, \underline{J}, \underline{G})$. In the cases (III) and (IV) this triangle does not belong to Ψ and $(\underline{C}, \underline{G}, \underline{D})$ is the only small connected triangle belonging to Ψ. Therefore, if there is no case (V), if we apply the local changes, the order in which we apply them does not matter. Thus, if there is no case (V), the operation of applying local changes to Ψ as depicted in Figure 6.29, Figure 6.30 and Figure 6.31 is well-defined.

If there is a case (V), then we have $\delta([\underline{C}, \underline{G}], c) = -1$ and according to Lemma 8 $\Diamond(., \underline{G}, \underline{C}, \underline{J})$ is not critical and therefore not g-flat. By the properties of flatspaces, we have that the line from $\underline{F}$ over $\underline{C}$ to $\underline{G}$ is a side of a neighbor flatspace and thus both $\Diamond(\underline{J}, \underline{C}, \underline{F}, \underline{I})$ and $\Diamond(\underline{I}, \underline{J}, \underline{G}, \underline{C})$ are g-flat and therefore either critical or rigid. As by definition in an ordinary flatspace neither critical nor rigid rhombi overlap, we have that $\Diamond(\underline{J}, \underline{C}, \underline{F}, \underline{I})$ is rigid and $\Diamond(\underline{I}, \underline{J}, \underline{G}, \underline{C})$ is critical. As $\delta([\underline{F}, \underline{C}], c) = 1$, the rigidity of $\Diamond(\underline{J}, \underline{C}, \underline{F}, \underline{I})$ and Lemma 2(8) imply that c uses the uncapacitated edges from $[\underline{I}, \underline{J}]$ to $[\underline{F}, \underline{C}]$. As $\delta([\underline{J}, \underline{C}], c) = 0$, c does not use the other uncapacitated edges in $\Diamond(\underline{J}, \underline{C}, \underline{F}, \underline{I})$. As $\delta([\underline{C}, \underline{I}], c) = -1$ we can conclude with Lemma 8 that both $\Diamond(\underline{C}, \underline{I}, ., \underline{J})$ and $\Diamond(., \underline{I}, \underline{E}, \underline{C})$ are not g-flat. Thus Ψ spans exactly two trapezoids that form a hexagon together. We can see that the second trapezoid is a case (V) via mirror symmetry and that the local changes in the dotted triangle $(\underline{C}, \underline{J}, \underline{G})$ made by the first trapezoid are exactly the changes that are made by the second trapezoid in $(\underline{C}, \underline{J}, \underline{G})$. Hence the order of applying the two local changes does not matter. Hence the operation of applying the local changes is well-defined and from applying the local changes we obtain a mapping $d : E_{\mathsf{RES}} \to \mathbb{R}$ in all cases.

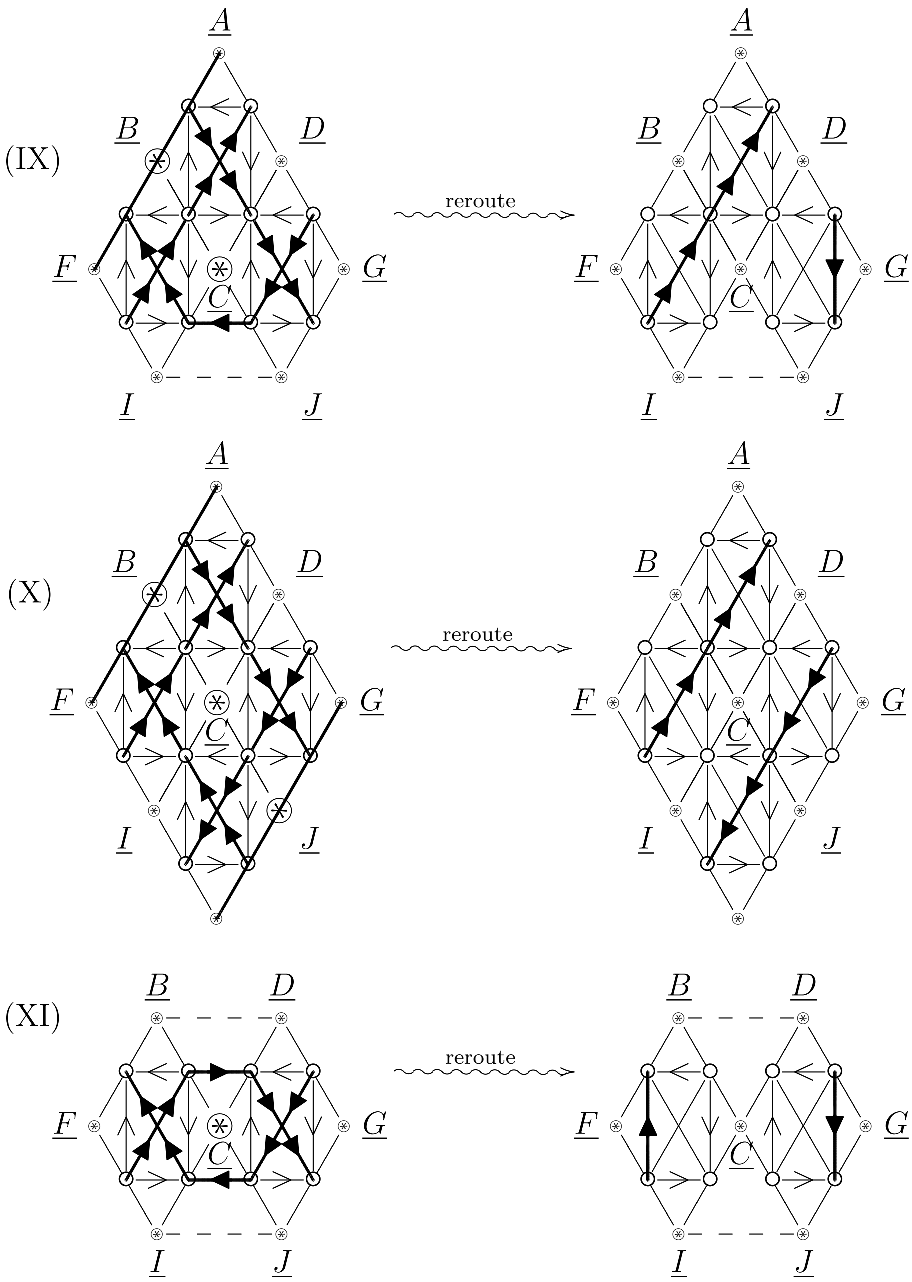

Figure 6.31: Rerouting c in all possible special cases up to rotation and mirroring. The fat lines indicate where the g-flatspace chain is glued together. Hexagons are g-flatspace chains on their own and are not glued together with other g-flatspaces. Inner vertices of g-flatspace chains are drawn bigger than others.

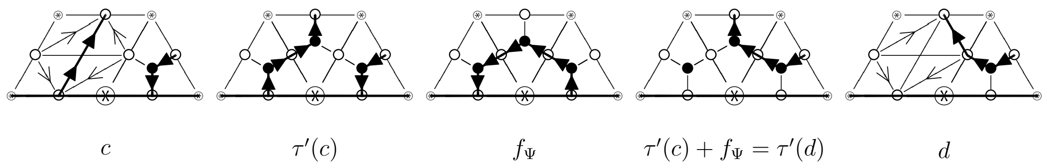

Figure 6.32: An example of rerouting on a trapezoid in case (I). The fat lines indicate where the g-flatspace chain Ψ is glued together. The inner vertex of Ψ is drawn bigger than others.

It is easy to check that $\tau'(c) + f_\Psi = \tau'(d)$, because $\delta(v, c) + \delta(v, f_\Psi) = \delta(v, d)$ for all $v \in G$. Figure 6.32 shows an example. As $\tau'(c) + f_\Psi$ is a flow on $\mathsf{RES}(f)$, by Lemma 15 we have that d is a flow on $\mathsf{RES}^b(f)$. By construction we have that $d = \sum_{i=1}^m c_i$ for well-directed vertex-disjoint cycles $c_1, \dots, c_m$ on $\mathsf{RES}^b(f)$ with $m \geq 0$. Thus $\{c_1, \dots, c_m\}$ is a quasi-cycle-decomposition of $\tau'(c) + f_\Psi$. Also by construction d uses less edges than c in each g-flatspace of Ψ and the same number of edges outside of Ψ. Therefore $\ell(c) > \ell(c_i)$ for all $i \in \{1, \dots, m\}$.

We show $m \geq 1$ by showing that $\tau'(c) + f_\Psi \neq 0$: The case that $\tau'(c)(e) + f_\Psi(e) = 0$ for all edges e in a g-flatspace R of Ψ can only happen, if R is a parallelogram of side lengths $(1, 2, 1, 2)$, as can be seen by looking at all cases in Figure 6.29, Figure 6.30 and Figure 6.31. So $\tau'(c) + f_\Psi = 0$ implies that Ψ completely consists of parallelograms of side lengths $(1, 2, 1, 2)$. Then Ψ has two open endings. But there can be no flatspace chain consisting of only parallelograms of side lengths $(1, 2, 1, 2)$ that has two open endings, because the long sides of all flatspaces in such a flatspace chain all share the same orientation —, \ or / and two open endings of a flatspace chain by construction never lie on the same side of Δ. $\square$

As all cases are considered, this proves the Shortest Cycle Theorem 6.19. $\square$

We can also prove the following three variants of the Shortest Cycle Theorem 6.19 with similar proofs. We will need these variants in Section 6.5 and Chapter 7. Let $\mathsf{RES}_\times(f)$ denote the digraph that results from adjusting in $\mathsf{RES}^b(f)$ the capacities on all edges e that are incident to o as follows: $l(e) \leftarrow u(e) \leftarrow 0$. Note that $\mathsf{RES}_\times(f)$ is independent of b.

Theorem 6.21 (Variant 1)**.** *Given a b-bounded integral shattered hive flow f and a well-directed cycle c on $\mathsf{RES}_\times(f)$ that is a shortest cycle among all well-directed cycles on $\mathsf{RES}_\times(f)$. Then $f + \tau'(c) \in P^b$.*

Theorem 6.22 (Variant 2)**.** *Let S be a subset of the set of circle border vertices. Let R denote the digraph that results from adjusting capacities in $\mathsf{RES}^b(f)$ as follows: For all edges e connecting o with a vertex from S we set $u_e \leftarrow l_e \leftarrow 0$. Let $z > 0$ such that $\vec{u}^R(v, w) = 0$ or $\vec{u}^R(v, w) \geq z$ for all vertices $v, w \in V$. Given* *a b-bounded z-integral shattered hive flow f and a well-directed cycle c on R with $\delta(c) > 0$ that is a shortest cycle among all well-directed cycles $\tilde{c}$ on R that satisfy $\delta(\tilde{c}) > 0$. Then $f + z\tau'(c) \in P^b$.*

Theorem 6.23 (Variant 3)**.** *Given any linear function $\mathbb{1} : \mathbb{R}^{H'} \to \mathbb{R}$ with non-negative coefficients, i.e. for all $\underline{A} \in H'$ we have $\mathbb{1}(\chi_{\underline{A}}) \geq 0$ where $\chi_{\underline{A}}(\underline{A}) = 1$ and $\chi_{\underline{B}} = 0$ for $\underline{A} \neq \underline{B}$.*

Let $z > 0$. Given a b-bounded z-integral shattered hive flow f and a well-directed cycle c on $\mathsf{RES}_\times(f)$ with $\mathbb{1}(\eta^{-1}(\tau'(c))) > 0$ that is a shortest cycle among all well-directed cycles $\tilde{c}$ on $\mathsf{RES}_\times(f)$ that satisfy $\mathbb{1}(\eta^{-1}(\tau'(\tilde{c}))) > 0$. Then $f + z\tau'(c) \in P^b$.

Dijkstra's algorithm and the Bellman-Ford algorithm Recall Lemma 6.13: If a well-directed cycle c on $\mathsf{RES}^b(f)$ has $\delta(c) > 0$, it uses one circle border vertex on the left side and one circle border vertex on one of the other two sides. It goes from o to the latter, traverses the big triangle, uses the former and returns to o. We search for such a well-directed cycle that uses a minimal number of edges. We can split the vertex o into two vertices o_1 and o_2 in a way that o_1 is connected with the source vertices and o_2 is connected with the sink vertices. Then Dijkstra's algorithm (see [CLRS01]) can be used to find a shortest path from o_1 to o_2 in polynomial time, which gives the desired cycle.

If we only search for a shortest well-directed cycle on $\mathsf{RES}_\times(f)$ and do not require that $\delta(c) > 0$ (e.g. in Variant 1), then we can use Breadth-First-Search in the following way: We start at a vertex v and do Breadth-First-Search until we find a well-directed cycle. Then we determine its length. We do this for each vertex v and take the shortest well-directed cycle.

Now we consider Variant 3: Given any linear function $\mathbb{1} : \mathbb{R}^{H'} \to \mathbb{R}$. Since $\mathbb{1} \circ \eta^{-1} \circ \tau' : F(\mathsf{RES}(f)) \to \mathbb{R}$ is a linear function and $F(\mathsf{RES}(f))$ is a subspace of $\mathbb{R}^{E_{\mathsf{RES}}}$, the function $\mathbb{1} \circ \eta^{-1} \circ \tau'$ can be continued linearly to $\omega : \mathbb{R}^{E_{\mathsf{RES}}} \to \mathbb{R}$ with $\omega|_{F(\mathsf{RES}(f))} = \mathbb{1} \circ \eta^{-1} \circ \tau'$. The function ω can be seen as an edge weight on $\mathsf{RES}(f)$. If the Bellman-Ford algorithm (see [CLRS01]) is started from a vertex v that is contained in a shortest well-directed cycle c on $\mathsf{RES}^b(f)$ with $\omega(c) > 0$, the algorithm is known to return such a cycle c in polynomial time. We can find the desired cycle by starting one instance of the Bellman-Ford algorithm from each vertex and comparing their lengths.

6.5 Checking multiplicity freeness

If λ, μ and ν are strictly decreasing partitions and $c^\nu_{\lambda\mu} > 0$, one can use for example the LRPA or the LRP-CSA as it will be explained in Chapter 7 to obtain an integral, *shattered* flow $f \in P^b$ with $\delta(f) = 2|\nu|$. Given such a flow f, then more can be said according to Lemma 6.8:

$$\begin{aligned} c^\nu_{\lambda\mu} \geq 2 \iff & \text{there is an integral flow } g \in P^b, g \neq f \text{ with } \delta(g) = 2|\nu|. \\ \iff & \text{there is an integral flow } 0 \neq d \in P^b - f \subseteq P^b_{\mathrm{flat}f} - f \\ & \text{ that uses no circle border vertex.} \\ \iff & \text{there is a well-directed cycle on } \mathsf{RES}_\times(f). \end{aligned}$$

The last equivalence holds because of Theorem 6.21 and the fact that by construction of $\mathsf{RES}_\times(f)$ each cycle c on $\mathsf{RES}_\times(f)$ has $\tau'(c) \neq 0$. As checking $\mathsf{RES}_\times(f)$ for a well-directed cycle can be done in polynomial time and obtaining the flow f can be done in polynomial time with the LRP-CSA, as we will see in Chapter 7, we can decide multiplicity freeness in polynomial time.

We get two corollaries from the above equivalences:

Corollary 6.24. *Let λ, μ, ν be strictly decreasing partitions. Given two distinct not necessarily integral hives $h_1, h_2 \in P(\lambda, \mu, \nu)$. Then $c^\nu_{\lambda\mu} \geq 2$.*

Proof. Let $h_1, h_2 \in P(\lambda, \mu, \nu)$, $h_1 \neq h_2$, $f := \eta(h_1)$, $g := \eta(h_2)$ with $\delta(f) = \delta(g) = 2|\nu|$. Then the LRPA finds an integral shattered hive flow $\bar{f} \in P^b$ with $\delta(\bar{f}) = 2|\nu|$. We have $\bar{f} \neq f$ or $\bar{f} \neq g$. W.l.o.g. $\bar{f} \neq g$. Then according to Lemma 6.11, $\tau(g - \bar{f})$ is a feasible flow on $\mathsf{RES}^b(\bar{f})$. As $\delta(v, g - \bar{f}) = 0$ on each circle border vertex v, we have that $\tau(g - \bar{f})$ is a feasible flow on $\mathsf{RES}_\times(\bar{f})$. With Lemma 5.11 $\tau(g - \bar{f})$ can be decomposed into well-directed cycles on $\mathsf{RES}_\times(\bar{f})$ and thus $c^\nu_{\lambda\mu} \geq 2$. ☐

Corollary 6.25. *Let λ, μ, ν be strictly decreasing partitions with $|\nu| = |\lambda| + |\mu|$. Then $c^\nu_{\lambda\mu} = 1 \Leftrightarrow c^{N\nu}_{N\lambda N\mu} = 1$ for all $N \in \mathbb{N}$.*

Proof. Note that for all hive flows f we have $\mathsf{RES}_\times(f) = \mathsf{RES}_\times(Nf)$. In particular there are no well-directed cycles on $\mathsf{RES}_\times(f)$ iff there are no well-directed cycles on $\mathsf{RES}_\times(Nf)$, which proves the claim. ☐

W. Fulton conjectured $c^\nu_{\lambda\mu} = 1 \Leftrightarrow c^{N\nu}_{N\lambda N\mu} = 1$ in the more general setting that the three partitions are not necessarily strictly decreasing. His conjecture was proved by Knutson, Tao and Woodward (cf. [KTW04]).

Chapter 7

The polynomial-time algorithm LRP-CSA

In this chapter we present the scaling method that turns the LRPA into its polynomial-time counterpart LRP-CSA. This method is basically about keeping $f \in P^b$ 2^k-integral for large k and finding well-directed cycles c in the residual network with $f + 2^k\tau'(c) \in P^b$. During the algorithm k decreases. One problem is that inner vertices of flatspace chains with open endings can be raised by 1 unit but one might not be able to raise them by 2^k units without leaving P^b. Therefore while $k \geq 1$ the algorithm preserves the regular border of f and thus prohibits that flatspace chains with open endings appear. For this reason we introduce a new residual network in Section 7.1. We describe the LRP-CSA in Section 7.2. The subsequent sections elaborate the technical details of the LRP-CSA.

7.1 The residual network

Recall that $\mathscr{S} \subset V$ is the set of source vertices and $\mathscr{T} \subset V$ is the set of sink vertices. To define the residual network, we first classify the circle border vertices into three types: small, medium and big. Let $k \in \mathbb{N}$ and let f be a 2^k-integral b-valid shattered hive flow. $\mathscr{V}^b_{\mathrm{big}}(f, 2^k)$ is the set of circle vertices that are at least 2^k away from their capacity bound and where adding 2^k units of flow preserves regularity on the border. They are called *big vertices*. $\mathscr{V}^b_{\mathrm{medium}}(f, 2^k)$ is the set of circle vertices that are at least 2^k away from their capacity bound and where adding 2^k units of flow does not preserve regularity on the border. They are called *medium vertices*. $\mathscr{V}^b_{\mathrm{small}}(f, 2^k)$ is the set of circle vertices that are less than 2^k away from their capacity bound. They are called *small vertices*. Formally:

$$\mathscr{V}^b_{\mathrm{small}}(f, 2^k) := \Big\{ s \in \mathscr{S} \mid \delta(s, f) > \delta^b_{\max}(s) - 2^k \Big\} \cup \Big\{ t \in \mathscr{T} \mid \delta(t, f) < \delta^b_{\min}(t) + 2^k \Big\}$$

$$\begin{aligned}\mathscr{V}^b_{\text{medium}}(f,2^k) &:= \Big\{s \in \mathscr{S} \mid \delta(s,f) \le \delta^b_{\max}(s) - 2^k, \delta(\text{pred}(s),f) = \delta(s,f) + 2^k\Big\} \\ &\cup \Big\{t \in \mathscr{T} \mid \delta(t,f) \ge \delta^b_{\min}(t) + 2^k, \delta(\text{pred}(t),f) = \delta(t,f) - 2^k\Big\}\end{aligned}$$

$$\begin{aligned}\mathscr{V}^b_{\text{big}}(f,2^k) &:= \Big\{s \in \mathscr{S} \mid \delta(s,f) \le \delta^b_{\max}(s) - 2^k, \delta(\text{pred}(s),f) > \delta(s,f) + 2^k\Big\} \\ &\cup \Big\{t \in \mathscr{T} \mid \delta(t,f) \ge \delta^b_{\min}(t) + 2^k, \delta(\text{pred}(t),f) < \delta(t,f) - 2^k\Big\}\end{aligned}$$

According to Lemma 6.8(7), each circle border vertex is either small, medium or big.

Definition 7.1 (The residual network $\mathsf{RES}^b_{2^k}(f)$)**.** We start with $\mathsf{RES}^b(f)$ which has capacities u, l as in Section 6.4.2. We get the new residual network $\mathsf{RES}^b_{2^k}(f)$ by adjusting the capacities to u', l' on all edges e by setting

$$\big(u'(e), l'(e)\big) := \begin{cases} (0,0) & \text{if } e \text{ connects } o \text{ with a small or medium vertex} \\ \big(u(e), l(e)\big) & \text{otherwise} \end{cases}.$$

■

Lemma 7.2. *Given a b-bounded, 2^k-integral, shattered hive flow $f \in P^b$ with a regular border. The set of well-directed cycles c on $\mathsf{RES}^b(f)$ for which $f + 2^k\tau'(c) \in P^b$ and $f + 2^k\tau'(c)$ has a regular border equals the set of well-directed cycles c on $\mathsf{RES}^b_{2^k}(f)$ that satisfy $f + 2^k\tau'(c) \in P^b$.*

Proof. Given a well-directed cycle c on $\mathsf{RES}^b_{2^k}(f)$ that has $f + 2^k\tau'(c) \in P^b$. Then c is a well-directed cycle on $\mathsf{RES}^b(f)$ and by construction of $\mathsf{RES}^b_{2^k}(f)$, c does not use small or medium vertices. Therefore $f + 2^k\tau'(c)$ has a regular border.

On the other hand let c be a well-directed cycle on $\mathsf{RES}^b(f)$ for which $f + 2^k\tau'(c) \in P^b$ and $f + 2^k\tau'(c)$ has a regular border. Then c does not use small or medium vertices. Thus c is a well-directed cycle on $\mathsf{RES}^b_{2^k}(f)$. □

7.2 The LRP-CSA

As a residual network can only be established, if the flow f is shattered, we need a mechanism to efficiently shatter a given flow f. This could be done by raising inner vertices of flatspace chains, but because of running time issues this is done by also optimizing a linear target function $\mathbb{1}$ on the set of inner vertices $I = H \setminus B$ of the big triangle graph Δ:

Definition 7.3 ($\mathbb{1}$-optimality)**.**

$$\mathbb{1} : \mathbb{R}^I \to \mathbb{R}, h \mapsto \sum_{i \in I} h(i).$$

A hive flow $f \in F$ is called $\mathbb{1}$-*optimal*, if there is no hive flow $g \in F$ with $f|_B = g|_B$ and $\mathbb{1}(g) > \mathbb{1}(f)$. ■

Lemma 7.4. *Let f be a hive flow with a regular border. If f is $\mathbb{1}$-optimal, then f is shattered.*

Proof. Assume that f is not shattered. Then according to Lemma 6.16 we can find a flatspace chain Ψ and increase its inner vertices Ψ_{inner} to get a better solution with respect to $\mathbb{1}$ that has the same border. This is a contradiction. □

We define $\mathbb{1}(d) := \mathbb{1}(\tau'(d))$ for flows d on $\mathsf{RES}(f)$. For notational convenience we call a flow d on $\mathsf{RES}(f)$ or on G δ-*positive*, if $\delta(d) > 0$. We do the same for $\mathbb{1}$-positivity.

The LRP-CSA is listed as Algorithm 4. We call each iteration of the for-loop a *round*. The LRP-CSA operates on b-bounded hive flows with a regular border and thus initial solution cannot be the 0-flow, because the 0-flow has no regular border. The construction of an initial solution is described in Section 7.4.

Theorem 6.22 ensures that $f \in P^b$ in line 13. Lemma 7.2 shows that additionally f has a regular border in line 13. Theorem 6.19 ensures that $f \in P^b$ in line 27. To regain shatteredness, the regular border is fixed and f is optimized w.r.t. $\mathbb{1}$ as described in Section 7.3. The intuition for optimizing w.r.t. $\mathbb{1}$ after each step is that many increasing steps of size 2^k should be made when k is still large. The correctness of the LRP-CSA is proved in Section 7.5. Running time issues are considered in Section 7.6.

7.3 Optimizing w.r.t. $\mathbb{1}$

In this section we show how flows can be shattered by optimizing w.r.t. $\mathbb{1}$ (line 14 and line 28) with Algorithm 5 and Algorithm 5′. Algorithm 5′ is not listed separately and will be explained in this section. We first consider the case in line 14. Recall that $\mathsf{RES}_\times(g)$ is the digraph that results from adjusting in $\mathsf{RES}^b(g)$ the capacities on all edges e that are incident to o as follows: $l(e) \leftarrow u(e) \leftarrow 0$. The LRP-CSA uses Algorithm 5 as a subalgorithm. We prove its correctness with the following lemma:

Lemma 7.5. *Given a 2^k-integral, b-bounded, $\mathbb{1}$-optimal hive flow f with a regular border and a well-directed cycle c on $\mathsf{RES}^b_{2^k}(f)$ with $f + 2^k\tau'(c) \in P^b$. When Algorithm 5 terminates on input (k, f, c), it returns a flow $g \in P^b$ such that $(f + 2^k c)|_B = g|_B$ and g is 2^k-integral and $\mathbb{1}$-optimal.*

Algorithm 4 The LRP-CSA

Input: $\lambda, \mu, \nu \in \mathbb{N}^n$ strictly decreasing partitions with $|\nu| = |\lambda| + |\mu|$.
Output: decide whether $c^{\nu}_{\lambda\mu} > 0$.

if $\ell(\nu) < \max\{\ell(\lambda), \ell(\mu)\}$ **then**
    return false.
end if
Create the regular target border b and the digraph G.
Find an initial $2^{\lceil \log(|\nu|) \rceil + 1}$-integral, b-bounded, $\mathbb{1}$-optimal hive flow f with a regular border (see Algorithm 6).
for $k = \lceil \log(|\nu|) \rceil + 1$ down to 1 **do**
    `rounddone` $\leftarrow$ **false**.
    while not `rounddone` **do**
        // f is a 2^k-integral, b-bounded, $\mathbb{1}$-optimal hive flow with a regular border.
        Construct $\mathsf{RES}^b_{2^k}(f)$.
        if there is a δ-positive well-directed cycle on $\mathsf{RES}^b_{2^k}(f)$ **then**
            Find a shortest δ-positive well-directed cycle c on $\mathsf{RES}^b_{2^k}(f)$.
            Augment 2^k units over c: $f \leftarrow f + 2^k \tau'(c)$.
            Fix the border of f and optimize w.r.t. $\mathbb{1}$ with Algorithm 5 to obtain a 2^k-integral, b-bounded, $\mathbb{1}$-optimal hive flow f with a regular border.
        else
            `rounddone` $\leftarrow$ **true**.
        end if
    end while
end for
// Last round:
`rounddone` $\leftarrow$ **false**.
while not `rounddone` **do**
    // f is an integral, b-bounded, $\mathbb{1}$-optimal hive flow with a regular border.
    Construct $\mathsf{RES}^b(f)$.
    if there is a δ-positive well-directed cycle on $\mathsf{RES}^b(f)$ **then**
        Find a shortest δ-positive well-directed cycle c on $\mathsf{RES}^b(f)$.
        Augment 1 unit over c: $f \leftarrow f + \tau'(c)$.
        Optimize w.r.t. $\mathbb{1}$ with Algorithm $5'$ to obtain an integral, b-bounded, $\mathbb{1}$-optimal hive flow f with a regular border.
    else
        `rounddone` $\leftarrow$ **true**.
    end if
end while
// f is an integral, b-bounded, $\mathbb{1}$-optimal hive flow with a regular border and there are no well-directed δ-positive cycles on $\mathsf{RES}^b(f)$.
if $\delta(f) = 2|\nu|$ **then**
    return true.
else
    return false.
end if

Algorithm 5 Optimize w.r.t. $\mathbb{1}$

Input: $k \in \mathbb{N}$, a 2^k-integral, b-bounded, $\mathbb{1}$-optimal hive flow f with a regular border and a well-directed cycle c on $\mathsf{RES}^b_{2^k}(f)$ which satisfies $f + 2^k\tau'(c) \in P^b$ and for which $f + 2^k\tau'(c)$ has a regular border.

Output: A 2^k-integral, b-bounded, $\mathbb{1}$-optimal hive flow g on G such that $g|_B = \big(f + 2^k\tau'(c)\big)\big|_B$.

$g \leftarrow f + 2^k c$.
`done` $\leftarrow$ **false**.
while not `done` **do**
  while there are g-flatspace chains **do**
    Compute a g-flatspace chain Ψ.
    Augment Ψ_{inner} by 2^k: $g \leftarrow g + 2^k f_\Psi$. // This increases $\mathbb{1}$ by at least 2^k.
  end while
  // $g \in P^b$ is shattered and 2^k-integral.
  if there is a $\mathbb{1}$-positive, well-directed cycle on $\mathsf{RES}_\times(g)$ **then**
    Find a shortest $\mathbb{1}$-positive, well-directed cycle c' on $\mathsf{RES}_\times(g)$.
    Augment 2^k units over c': $g \leftarrow g + 2^k\tau'(c')$. // This increases $\mathbb{1}$ by at least 2^k.
    // We have $g \in P^b$ and g is 2^k-integral.
  else
    `done` $\leftarrow$ **true**.
  end if
end while
return g.

Proof. The inner vertices of flatspace chains in line 6 can be raised by 2^k, as $f + 2^k\tau'(c)$ has a regular border (cf. Lemma 6.16). Theorem 6.23 shows that we have $g \in P^b$ in line 12.

When the algorithm returns a flow g, then g is shattered and there are no $\mathbb{1}$-positive well-directed cycles on $\mathsf{RES}_\times(g)$. Assume that g is not $\mathbb{1}$-optimal. Then there exists a flow g' on P^b with $g|_B = g'|_B$ and $\mathbb{1}(g') > \mathbb{1}(g)$. Therefore $g' - g \in P^b - g$ and $g' - g$ has $\delta(v, g' - g) = 0$ for each border vertex v. With Lemma 6.11 we have that there exists a feasible flow $d \in P_{\text{feas}}(\mathsf{RES}^b(g))$ with $\tau'(d) = g' - g$ and thus $\mathbb{1}(d) > 0$. Lemma 5.11 shows that d can be decomposed into well-directed cycles. None of these cycles uses any border vertices, so they are all cycles on $\mathsf{RES}_\times(g)$ as well. Using the linearity of $\mathbb{1}$, one of those cycles c must have $\mathbb{1}(c) > 0$, which is a contradiction. □

We now want to analyze the running time of Algorithm 5. The idea is to show that $\mathbb{1}(g) \le \mathbb{1}(f + 2^k c) + \mathcal{O}(2^k n^5)$. This is sufficient to prove its polynomial running time, because $\mathbb{1}$ is increased by 2^k in line 6 and in line 11. We proceed by proving two lemmas.

Lemma 7.6. *Let f be a shattered hive flow and c a cycle on $\mathsf{RES}(f)$. Then $\mathbb{1}(c) \le n(n-1)(n-2)$.*

Note that this lemma holds for all cycles, not just for well-directed ones.

Proof. It is easy to check that $|I| = \frac{(n-1)(n-2)}{2}$. We have $\mathbb{1}(c) = \sum_{\underline{A} \in I} \operatorname{wind}(\underline{A}, c)$. As $|\delta^{\mathsf{RES}(f)}(v, \tau'(c))| \le 2$ for all $v \in V$ (cf. proof of Theorem 6.19, Lemma 2(1)), by Lemma 6.6 we have $\operatorname{wind}(\underline{A}, c) \le 2n$. Therefore $\mathbb{1}(c) \le |I| \cdot 2n = \frac{(n-1)(n-2)}{2} \cdot 2n = n(n-1)(n-2)$. □

Lemma 7.7. *Given a 2^k-integral, b-bounded, $\mathbb{1}$-optimal hive flow $f \in P^b$ with a regular border and a flow d on $\mathsf{RES}^b(f)$ that has a flow value $-m2^k \le d(e) \le m2^k$ on each edge e in $\mathsf{RES}^b(f)$ for some $m > 0$. Let d' be a flow on G with $\delta(v, d') = 0$ for all circle border vertices v, $f + \tau'(d) + d' \in P^b$ and $f + \tau'(d) + d'$ is $\mathbb{1}$-optimal. Then $\mathbb{1}(f + \tau'(d) + d') - \mathbb{1}(f + \tau'(d)) = \mathbb{1}(d') = \mathcal{O}(2^k n^5 m)$.*

Before proving Lemma 7.7 we recall the situation in Algorithm 5. Given a 2^k-integral, $\mathbb{1}$-optimal hive flow $f \in P^b$ with a regular border and a well-directed cycle c on $\mathsf{RES}^b_{2^k}(f)$ such that $f + 2^k\tau'(c) \in P^b$ and a flow g which is returned by Algorithm 5 and is $\mathbb{1}$-optimal with $f|_B = g|_B$. If we set $d := 2^k c$ and $d' := g - f - 2^k\tau'(d)$, we can apply Lemma 7.7 with $m = 1$ and get $\mathbb{1}(g) = \mathbb{1}(f + 2^k\tau'(c)) + \mathcal{O}(2^k n^5)$ as desired.

We note that in order to apply Lemma 7.7 f must be $\mathbb{1}$-optimal. So the LRP-CSA must reoptimize w.r.t. $\mathbb{1}$ after each step.

Definition 7.8. Given a cycle $c = (v_1, \dots, v_\ell, v_{\ell+1} = v_1)$ on $\mathsf{RES}^b(f)$. A tuple $(v_i, v_{i+1}), 1 \le i \le \ell$ is called *critical*, if $\vec{u}^{\mathsf{RES}^b(f)}(v_i, v_{i+1}) = 0$. ■

Proof of Lemma 7.7. Given $m > 0$ and f, d, d' as in Lemma 7.7. The flow $\tau(d')$ is a flow on $\mathsf{RES}_\times(f)$ which is not necessarily feasible. We want to find an upper bound for $\mathbb{1}(d')$.

Let M denote the number of edges in $\mathsf{RES}_\times(f)$. In each flat rhombus, $\mathsf{RES}_\times(f)$ has 20 edges. As f can have at most $n^2/2$ flat rhombi, we have $M \leq 10n^2$ $(*)$.

According to Lemma 5.6 the flow $\tau(d')$ can be decomposed into not necessarily well-directed cycles on $\mathsf{RES}_\times(f)$:

$$\tau(d') = \sum_{i=1}^{M} \alpha_i c_i$$

with cycles $c_1, \ldots, c_M$ on $\mathsf{RES}_\times(f)$ and $\alpha_1, \ldots, \alpha_M \in \mathbb{R}_{\geq 0}$ such that for all $1 \leq i \leq M$ and for all edges $e \in c_i$ we have $\operatorname{sgn}(c_i(e)) = \operatorname{sgn}(f(e))$.

Let $\tilde{c}$ be such a cycle in the decomposition with $\mathbb{1}(\tilde{c}) > 0$. Assume that $\tilde{c}$ is well-directed. Then Lemma 6.11 shows that $f + \varepsilon\tau'(\tilde{c}) \in P^b$ for some $\varepsilon > 0$. This is a contradiction to f being $\mathbb{1}$-optimal. So $\tilde{c}$ cannot be well-directed. Hence $\tilde{c}$ contains a critical tuple (v_i, v_{i+1}). Let $e := \{v_i, v_{i+1}\}$. W.l.o.g. $l(e) = 0$, $\tilde{c}(e) < 0$ and $\tau(d')(e) < 0$. By assumption we have $|d(e)| \leq m2^k$. As according to Lemma 6.11 we have $d + \tau(d') \in P_{\text{feas}}(\mathsf{RES}^b(f))$, it follows that $d(e) + \tau(d')(e) \geq 0$ and therefore $\tau(d')(e) > -m2^k$. Thus from the cycle decomposition we get that

$$\sum_{\substack{i=1 \\ c_i(e)\neq 0}}^{M} \alpha_i \leq m2^k.$$

In particular $\alpha_i \leq m2^k$ for all i with $c_i(e) \neq 0$. As each $\mathbb{1}$-positive cycle in the decomposition uses a critical tuple, we get

$$\begin{aligned}
\mathbb{1}(d') &\leq \sum_{\substack{i=1 \\ \mathbb{1}(c_i)>0}}^{M} \alpha_i \mathbb{1}(c_i) \overset{\text{Lemma 7.7}}{\leq} \sum_{\substack{i=1 \\ \mathbb{1}(c_i)>0}}^{M} \alpha_i n(n-1)(n-2) \\
&\leq \sum_{\substack{i=1 \\ \mathbb{1}(c_i)>0}}^{M} m2^k n(n-1)(n-2) \leq Mm2^k n(n-1)(n-2) \\
&\overset{(*)}{\leq} 10m2^k n^3(n-1)(n-2) = \mathcal{O}(2^k n^5 m).
\end{aligned}$$

□

Algorithm 5′ We now describe Algorithm 5′ used in Algorithm 4, line 28, and prove its correctness and polynomial running time. It is not listed separately, because it is nearly the same as Algorithm 5 with $k = 0$. The only difference is that the input $f + \tau'(c)$ not necessarily has a regular border. Therefore, in the first iteration of the while-loop in line 6, inner vertices of flatspace chains with open endings can be raised. We already know that raising inner vertices of flatspace chains with open endings works without leaving P^b, because the target border b is regular (see Lemma 6.17). This proves the correctness of Algorithm 5′.

Lemma 7.9. *Algorithm 5′ runs in polynomial time.*

Proof. To prove the polynomial running time of Algorithm 5′ it remains to show that the first iteration of the outer while-loop runs in polynomial time as only this iteration differs from Algorithm 5.

Let $\Psi_1, \ldots, \Psi_M$ denote the flatspace chains whose inner vertices are raised in line 6 during the first iteration of the outer while-loop. Our goal is to show that $M = \mathcal{O}(n^6)$. We have $f + \tau'(c) + f_{\Psi_1} + \ldots + f_{\Psi_M} \in P^b$. Let $J \subset \{1, \ldots, M\}$ be the set of indices i such that Ψ_i has an open ending. Let each Ψ_i have at most 1 open ending and let this open ending be on the right side of Δ. The proof for the other cases is similar. Let $\psi := \sum_{i \in J} f_{\Psi_i}$. Algorithm 5′ returns a flow g with $\delta(v, g) = \delta(v, f + \tau'(c) + \psi)$ for each circle border vertex v.

We bound M by first proving $|J| \leq n$. This bounds the absolute flow value on each edge of $\tau(\tau'(c) + \psi)$ and we can apply Lemma 7.7 with $d = \tau(\tau'(c) + \psi)$ to bound $M - |J|$. We now show that $|J| \leq n$:

Let i be the smallest element of J, if $|J| \neq \emptyset$. Note that the flatspace Ψ_i in $(f + \tau'(c) + f_{\Psi_1} + \ldots + f_{\Psi_{i-1}})$ has an open ending and width 2, because on each side of the big triangle graph Δ c uses at most 1 border vertex. We have $\delta(v, f_{\Psi_i}) = -1$ and $\delta(\text{pred}(v), f_{\Psi_i}) = 1$ for a circle border vertex v with $\delta(v, c) = 1$ and $\delta(\text{pred}(v), c) = 0$. Thus $\delta(v, \tau'(c) + f_{\Psi_i}) = 0$ and $\delta(\text{pred}(v), \tau'(c) + f_{\Psi_i}) = 1$. Let i' be the smallest element of $J \setminus \{i\}$ if $|J| \geq 2$. Note that the flatspace $\Psi_{i'}$ in $(f + \tau'(c) + f_{\Psi_1} + \ldots + f_{\Psi_{i'-1}})$ has and open ending and width 2 with $\delta(v, f_{\Psi_{i'}}) = 0$, $\delta(\text{pred}(v), f_{\Psi_{i'}}) = -1$ and $\delta(\text{pred}^2(v), f_{\Psi_{i'}}) = 1$. Thus $\delta(v, \tau'(c) + f_{\Psi_i} + f_{\Psi_{i'}}) = 0$, $\delta(\text{pred}(v), \tau'(c) + f_{\Psi_i} + f_{\Psi_{i'}}) = 0$ and $\delta(\text{pred}^2(v), \tau'(c) + f_{\Psi_i} + f_{\Psi_{i'}}) = 1$. We can continue this construction and see that $|J| \leq n$.

As there can be open endings on two sides, we have $|\psi(e) + \tau'(c)(e)| \leq 2n + 1$ for each edge $e \in E$. Note that $\tau(\psi + \tau'(c))$ is a flow on $\mathsf{RES}^b(f)$ with absolute flow value at most $4n + 2$ on each edge. Lemma 7.7 shows that $\mathbb{1}(f + \psi + \tau'(c) + d') - \mathbb{1}(f + \psi + \tau'(c)) = \mathcal{O}(2^k n^6)$ for the $\mathbb{1}$-optimal flow $f + \psi + \tau'(c) + d'$. Therefore $M - |J| = \mathcal{O}(n^6)$ and thus $M = \mathcal{O}(n^6)$. ☐

7.4 An initial solution

In this section we describe how to find an initial b-bounded, 2^k-integral, $\mathbb{1}$-optimal hive h_{init} with a regular border for a given $k \in \mathbb{N}$. We proceed step by step until we get a desired hive.

Each vertex in $\underline{A} \in H$ lies in a row $\varrho_{\downarrow}(\underline{A})$ counted from the top row (row 1) to the bottom row (row $n + 1$). Each vertex in $\underline{A} \in H$ lies in a column $\varrho_{\swarrow}(\underline{A})$

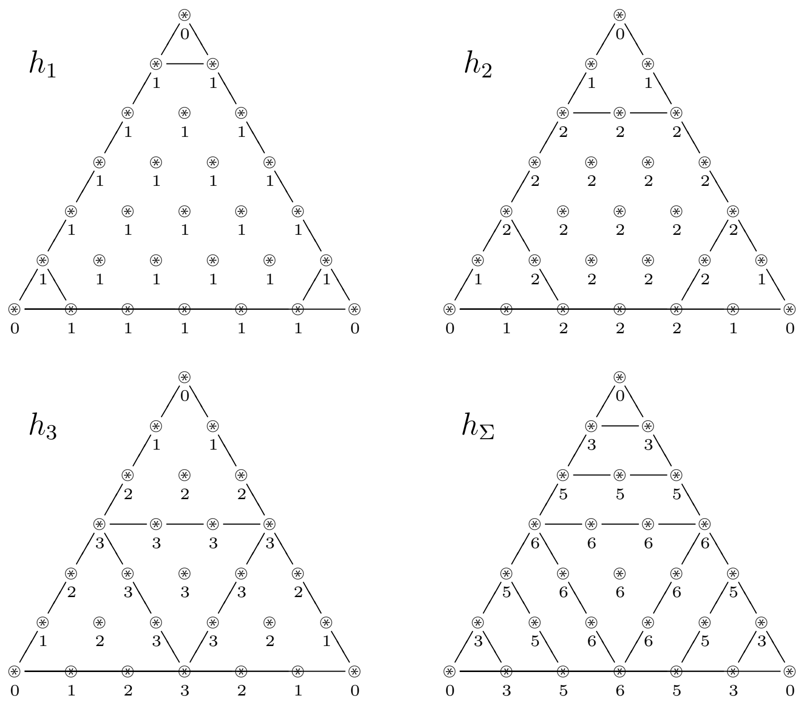

Figure 7.1: The construction of $h_\Sigma = \sum_i h_i$.

counted from the column 1 (the vertices on the right border) to the column $n+1$ (the vertex in the lower left corner). Generate a hive $h_\downarrow \in \mathbb{R}^H$ by setting

$$h_\downarrow(\underline{A}) = \varrho_\downarrow(\underline{A}) - 1$$

and generate a hive $h_\swarrow \in \mathbb{R}^H$ by setting

$$h_\swarrow(\underline{A}) = \varrho_\swarrow(\underline{A}) - 1.$$

Note that both hives consist of exactly one flatspace, namely one big triangle. Define $h_{\text{flat}} := h_\downarrow + h_\swarrow$. Let $f_{\text{flat}} := \eta(h_{\text{flat}})$. On all source vertices $s \in \mathscr{S}$ we have $\delta(s, f_{\text{flat}}) = 1$. On all sink vertices $t \in \mathscr{T}$ we have $\delta(t, f_{\text{flat}}) = -2$.

Each vertex in $\underline{A} \in H$ lies in a *layer* $\varrho(\underline{A})$, which is the shortest edge distance in Δ to a corner of Δ. The 3 corner vertices each have $\varrho(\underline{A}) = 0$. For $i \in \mathbb{N}$, define

$$h_i(\underline{A}) := \min\{i, \varrho(\underline{A})\}.$$

See Figure 7.1 for an illustration. Let $\varrho_{\max} := \left\lfloor \frac{n-1}{2} \right\rfloor$. If $1 \leq i \leq \varrho_{\max}$ and $n > 1$, then h_i is a hive that consists of 4 flatspaces: 3 triangles in the corners and 1 triangle or hexagon in the center. It is easy to see that this results in a hive. Now consider the sum of hives $h_\Sigma := \sum_{i=1}^{\varrho_{\max}} h_i$. Since h_Σ is a sum of hives, it is a hive itself. Let $f_\Sigma := \eta(h_\Sigma)$. On all source vertices $s \in \mathscr{S}$ we have $\delta(s, f_\Sigma) \in \{-\varrho_{\max}, \dots, \varrho_{\max}\}$. On all sink vertices $t \in \mathscr{T}$ we have $\delta(t, f_\Sigma) \in \{-\varrho_{\max}, \dots, \varrho_{\max}\}$.

Now fix the border of h_Σ and optimize w.r.t. $\mathbb{1}$ with Algorithm 6. Like Algorithm 5 it searches for shortest well-directed $\mathbb{1}$-positive cycles in $\mathsf{RES}^b(f)$ and augments over them. Whenever detecting any big flatspaces, it increases their inner vertices. Note that in line 11 we have $f + \tau'(c) \in P^b$ because of Theorem 6.23.

Algorithm 6 Initially optimize w.r.t. $\mathbb{1}$

Input: The hive $h_\Sigma \in \mathbb{R}^{H'}$.
Output: A $\mathbb{1}$-optimal hive $h \in \mathbb{R}^{H'}$ with $h_\Sigma|_B = h|_B$.

1: $f \leftarrow \eta(h_\Sigma)$.
2: `done` $\leftarrow$ **false**.
3: **while** not `done` **do** // at most $\frac{n^2(n-1)^2}{4}$ steps
4:   **while** there are f-flatspace chains **do**
5:     Compute an f-flatspace chain Ψ.
6:     Augment Ψ_{inner} by 1: $f \leftarrow f + f_\Psi$. // This increases $\mathbb{1}$ by at least 1.
7:   **end while**
8:   // f is integral and shattered.
9:   **if** there is a $\mathbb{1}$-positive well-directed cycle on $\mathsf{RES}_\times(f)$ **then**
10:     Find a shortest $\mathbb{1}$-positive well-directed cycle c on $\mathsf{RES}_\times(f)$.
11:     Augment 1 unit over c: $f \leftarrow f + \tau'(c)$. // This increases $\mathbb{1}$ by at least 1.
12:   **end if**
13: **end while**

Proposition 7.10. *Given a hive h_Σ generated as above, then Algorithm 6 finds a $\mathbb{1}$-optimal hive h with $h_\Sigma|_B = h|_B$ in polynomial time.*

Proof. The correctness of Algorithm 6 can be proved in the same way as the correctness of Algorithm 5. For any hive $h \in \mathbb{R}^H$ with fixed border $h|_B = h_\Sigma|_B$ and for each $\underline{A} \in I$, $h(\underline{A})$ can be bounded as follows: Recall that $h' : \operatorname{conv}(H) \to \mathbb{R}$ is a concave function (see Definition 6.2). Note that the top vertex $\underline{0}$ of Δ has $h_\Sigma(\underline{0}) = 0$ and the two adjacent vertices $\underline{B}$ have $h_\Sigma(\underline{B}) = \varrho_{\max}$. As h is a hive, h' must be concave and therefore each vertex $\underline{C} \in H$ can have height at most $h_\Sigma(\underline{C}) \le n \cdot \varrho_{\max}$.

As each operation of Algorithm 6 in line 6 and line 11 raises $\mathbb{1}(f)$ by 1 and $\mathbb{1}(h_\Sigma) \ge 0$ and $\mathbb{1}$ is bounded by $|I| \cdot n \cdot \varrho_{\max}$, the outer while-loop runs for at most $\frac{(n-1)(n-2)}{2} \cdot n \cdot \frac{n-1}{2} = \frac{n^2(n-1)^2}{4}$ steps. So an integral $\mathbb{1}$-optimal hive h with fixed border $h|_B = h_\Sigma|_B$ can be found in polynomial time. $\square$

Lemma 7.11. *Let $z \in \mathbb{R}$. A hive h is $\mathbb{1}$-optimal with border $h_\Sigma|_B$ iff $h + zh_{\mathit{flat}}$ is $\mathbb{1}$-optimal with border $h_\Sigma|_B + zh_{\mathit{flat}}|_B$.*

Proof. Note that adding or subtracting any multiple of h_{flat} does not change any rhombus' slack. Assume that $h + zh_{\text{flat}}$ is a hive with border $h_\Sigma|_B + zh_{\text{flat}}|_B$ that is not $\mathbb{1}$-optimal. Then there is a hive $h + zh_{\text{flat}} + \tilde{h}$ with border $h_\Sigma|_B + zh_{\text{flat}}|_B$ that is $\mathbb{1}$-optimal with $\mathbb{1}(\tilde{h}) > 0$ and $\tilde{h}|_B = 0$. But then $h + \tilde{h}$ is a hive with border $h_\Sigma|_B$ with $\mathbb{1}(h+\tilde{h}) > \mathbb{1}(h)$ which is a contradiction to the $\mathbb{1}$-optimality of h. □

Since $h(\underline{0}) = 0$, we can set $f := \eta(h)$. Consider $f - \varrho_{\max} f_{\text{flat}}$: On all source vertices $s \in \mathscr{S}$ we have $\delta(s, f_{\text{flat}}) = 1$ and $\delta(s, f) \leq \varrho_{\max}$. So we have

$$\delta(s, f - \varrho_{\max} f_{\text{flat}}) \leq 0.$$

On all sink vertices $t \in \mathscr{T}$ we have $\delta(t, f_{\text{flat}}) = -2$ and $\delta(t, f) \geq -\varrho_{\max}$. So we have

$$\delta(t, f - \varrho_{\max} f_{\text{flat}}) \geq 0.$$

This results in $f - \varrho_{\max} f_{\text{flat}}$ and any positive multiple of $f - \varrho_{\max} f_{\text{flat}}$ being b-bounded for any b that comes from partitions.

Let $k := \lceil \log(|\nu|) \rceil + 1$. Thus k is linear in the input size. Scale $f - \varrho_{\max} f_{\text{flat}}$ by setting $f_{2^k} := (f - \varrho_{\max} f_{\text{flat}}) \cdot 2^k$. We have that f_{2^k} is b-bounded, 2^k-integral and that it has a regular border.

$h_{\text{init}} := \eta^{-1}(f_{2^k})$ is the desired initial hive.

We now show that its δ-value is not very far from $2|\nu|$:

We have

$$\begin{aligned} \delta(f_{2^k}) &= 2^k \cdot \big(\delta(f) - \varrho_{\max}\delta(f_{\text{flat}})\big) \\ &\geq 2^k \cdot \big(0 - \varrho_{\max} \cdot 4n\big) = -2^k \cdot 4n \left\lfloor \frac{n-1}{2} \right\rfloor \\ &\geq -2^k \cdot 4n \frac{n-1}{2} = -2^k \cdot 2n(n-1). \end{aligned}$$

We also have $2|\nu| \leq 2 \cdot 2^k$. Hence

$$2|\nu| - \delta(f_{2^k}) \leq 2^k(2n(n-1) + 2) = \mathcal{O}(2^k n^2).$$

This ensures that the first round of Algorithm 4 runs in polynomial time.

7.5 Correctness

After introducing the LRP-CSA and all necessary subalgorithms, we can now prove the main result:

Theorem 7.12. *If given as input three strictly decreasing partitions λ, μ, $\nu \in \mathbb{N}^n$ with $|\nu| = |\lambda| + |\mu|$ that consist of natural numbers smaller than 2^k for some $k \in \mathbb{N}$, then the LRP-CSA returns true iff $c^\nu_{\lambda\mu} > 0$. Its running time is polynomial in n and k.*

Proof. The running time issues are considered in Section 7.6.

First of all the algorithm checks whether $\ell(\nu) < \max\{\ell(\lambda), \ell(\mu)\}$. If this is the case, then we have $c^{\nu}_{\lambda\mu} = 0$ and need no additional computation.

If the LRP-CSA returns true, an integral b-bounded hive flow f was found with $\delta(f) = 2|\nu|$. Lemma 6.8(6) shows that $c^{\nu}_{\lambda\mu} > 0$.

If the LRP-CSA returns false and has not returned in line 2, then there is an integral hive flow $f \in P^b$ with $\delta(f) < 2|\nu|$ and for which there is no δ-positive well-directed cycle in $\mathsf{RES}^b(f)$. The Optimality Test (Lemma 6.12) ensures that f maximizes δ in P^b. So by Lemma 6.8(5) we have $c^{\nu}_{\lambda\mu} = 0$. □

7.6 Running time

Each subalgorithm of the LRP-CSA runs in polynomial time as described in the respective sections and the number of rounds is linear in the input length. We will prove the polynomial running time of Algorithm 4 in this section by showing that the while-loop in line 8 runs only a polynomial number of times for each k and that the while-loop in line 25 runs only a polynomial number of times. As k is polynomial in the input length, the LRP-CSA runs in polynomial time.

Let $\delta_{\max} := \max\{\delta(f) \mid f \in P^b\}$.

Lemma 7.13 (Scaling-Lemma)**.** *Given a 2^k-integral, shattered hive flow $f \in P^b$. If there are no well-directed, δ-positive cycles on $\mathsf{RES}^b_{2^k}(f)$, then $\delta_{\max} - \delta(f) \le 2^k n^2$.*

Proof. Let $f \in P^b$ be a 2^k-integral, shattered hive flow such that there are no well-directed, δ-positive cycles in $\mathsf{RES}^b_{2^k}(f)$. A well-directed cycle on $\mathsf{RES}^b(f)$ that uses two big vertices is a well-directed cycle on $\mathsf{RES}^b_{2^k}(f)$ as well. So there are no well-directed, δ-positive cycles in $\mathsf{RES}^b(f)$ that use two big vertices.

Let w be a circle border vertex, w.l.o.g. $w \in \mathscr{S}$, and let

$$\delta^b_{\max}(w) - \delta(w, f) \le \zeta$$

for some $\zeta \in \mathbb{R}$, e.g. the case where w is a small vertex and $\zeta \le 2^k - 1$. Let $\mathrm{succ}(w)$ be a medium vertex. Since from Lemma 6.8(7) we know that $\delta^b_{\max}(\mathrm{succ}(w)) \le \delta^b_{\max}(w)$, we have $\delta^b_{\max}(\mathrm{succ}(w)) - \delta(w, f) \le \zeta$. As $\mathrm{succ}(w)$ is medium, we have $\delta(w, f) = \delta(\mathrm{succ}(w), f) + 2^k$. So

$$\delta^b_{\max}(\mathrm{succ}(w)) - \delta(\mathrm{succ}(w), f) \le \zeta + 2^k.$$

So if there are consecutive medium vertices that have a small vertex as predecessor, for all those medium vertices v we have

$$\delta^b_{\max}(v) - \delta(v, f) \le (2^k - 1) + (n - 1) \cdot 2^k < n \cdot 2^k.$$

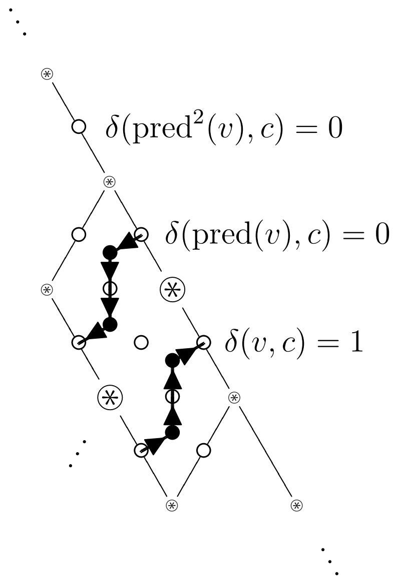

Figure 7.2: A flatspace chain Ψ in $f + 2^k\tau'(c) + \psi$ which has an open ending on the right side of Δ. The inner vertices are drawn bigger than others. The fat arrows represent f_Ψ.

We call these medium vertices *minor medium vertices*. All other medium vertices are called *major medium vertices*.

So far we have bounded $\delta^b_{\max}(v) - \delta(v, f)$ for small and minor medium vertices v. Additionally we know that no well-directed δ-positive cycle uses two big vertices.

We want to show that there is no δ-positive well-directed cycle on $\mathsf{RES}^b(f)$ which uses two major medium vertices or one big and one major medium vertex. Assume the contrary, i.e. that that there is a δ-positive, well-directed cycle c on $\mathsf{RES}^b(f)$ which uses two major medium vertices or one big and one major medium vertex. Let v be a major medium vertex used by c. Let w be the other border vertex used by c, i.e. a big or major medium vertex. The flow $f+2^k\tau'(c)$ has big flatspaces, because its border is not regular. Therefore flatspace chains without open endings can be found and raised by 2^k each until there are only flatspace chains left that have an open ending. Let $\psi \in F(G)$ be the flow corresponding to this raise. Then $f + 2^k\tau'(c) + \psi \in P^b$ and $f + 2^k\tau'(c) + \psi$ is 2^k-integral. The flatspace chains of $f + 2^k\tau'(c) + \psi$ each have width 2, because c uses at most one border vertex on each side of the big triangle graph Δ. At least one of the these flatspace chains Ψ has an open ending containing v and $\mathrm{pred}(v)$. W.l.o.g. $v \in \mathscr{S}$, $\mathrm{pred}(v) \in \mathscr{S}$ and $w \in \mathscr{T}$. Then $f_\Psi(v) = -1$ and $f_\Psi(\mathrm{pred}(v)) = 1$ (see Figure 7.2). Note that since v is major medium and w is major medium or big, we have $f + 2^k\tau'(c) + \psi + 2^k f_\Psi \in P^b$. Also note that $\delta(v, 2^k\tau'(c)+\psi+2^k f_\Psi) = 0$ and $\delta(\mathrm{pred}(v), 2^k\tau'(c)+\psi+2^k f_\Psi) = 2^k$. If w is major medium, then depending on whether Ψ has an open ending containing w, we have $\delta(\text{pred}(w), 2^k\tau'(c)+\psi+2^k f_\Psi) = -2^k$ or $\delta(w, 2^k\tau'(c)+\psi+2^k f_\Psi) = -2^k$. The flow $\tau(2^k\tau'(c)+\psi+2^k f_\Psi)$ on $\mathsf{RES}^b(f)$ can be decomposed into well-directed cycles and one of these cycles must use pred(v) and w or pred(v) and pred(w). This cycle is δ-positive, while the other cycles c' have $\delta(c') = 0$. pred(v) and pred(w) each are major medium or big. Repeat this argument until a well-directed δ-positive cycle on $\mathsf{RES}^b(f)$ is found that uses two big vertices. This is a contradiction.

So a δ-positive well-directed cycle that uses a major medium vertex must also use a small or a minor medium vertex. And a δ-positive well-directed cycle that uses a big vertex must also use a small or a minor medium vertex.

Now consider a flow d on G with $f + d \in P^b$ and $\delta(f + d) = \delta_{\max}$. If the sum of throughput in d through big and major medium vertices exceeds the sum of throughput through minor medium or small vertices, then d must decompose in at least one well-directed cycle on $\mathsf{RES}^b(f)$ that uses two big vertices or a big and a major medium vertex or two major medium vertices, which is a contradiction. As $|\mathscr{T}| = n$, we have $\delta_{\max} - \delta(f) = \delta(d) \le n^2 \cdot 2^k$. $\square$

How many δ-positive well-directed cycles on $\mathsf{RES}^b_{2^k}(f)$ can be found during a round in Algorithm 4? The first iteration runs in polynomial time as seen at the end of Section 7.4. After the kth round we have $\delta_{\max} - \delta(f) \le n^2 \cdot 2^k$. So how many δ-positive well-directed cycles can be found in the next round on $\mathsf{RES}^b_{2^{k-1}}(f)$? Clearly at most $2n^2$. So we know that every round besides the last one run in polynomial time. At the beginning of the last round, we have $\delta_{\max} - \delta(f) \le n^2 \cdot 2^1$. This ensures that the last round runs in polynomial time as well.

7.7 Handling weakly decreasing partitions

The LRPA and the LRP-CSA can only handle triples of strictly decreasing partitions λ, μ and ν. What if at least one of these input partitions is only weakly decreasing? We will adjust λ, μ and ν to $\tilde\lambda$, $\tilde\mu$ and $\tilde\nu$ such that they are strictly decreasing and $c^{\nu}_{\lambda\mu} > 0 \iff c^{\tilde\nu}_{\tilde\lambda\tilde\mu} > 0$.

Recall the hives h_Σ and h_{flat} from Section 7.4. and that $\varrho_{\max} := \left\lfloor \frac{n-1}{2} \right\rfloor$. Figure 7.3 shows an example of $h_\Sigma + \varrho_{\max} h_{\text{flat}}$. Let $\bar b \in \mathbb{R}^B$ be the border of $h_\Sigma + \varrho_{\max} h_{\text{flat}}$. Then $\bar b$ is regular and weakly increasing from top to bottom on the left and on the right and from right to left on the bottom. Thus we have for any three consecutive border vertices $\underline{A}, \underline{B}, \underline{C}$ that

$$\bar b(\underline{A}) - \bar b(\underline{B}) > \bar b(\underline{B}) - \bar b(\underline{C}).$$

Therefore adding $\bar b$ to any border b coming from any three partitions will eliminate the irregularities:

$$b(\underline{A}) - b(\underline{B}) \ge b(\underline{B}) - b(\underline{C}) \Rightarrow (b+\bar b)(\underline{A}) - (b+\bar b)(\underline{B}) > (b+\bar b)(\underline{B}) - (b+\bar b)(\underline{C}).$$

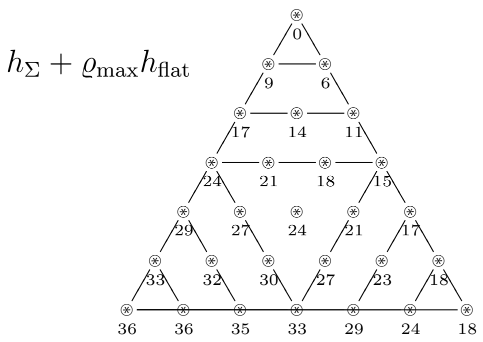

Figure 7.3: An example of $h_\Sigma + \varrho_{\max} h_{\text{flat}}$.

Let $b := b(\lambda, \mu, \nu)$ be the border induced by λ, μ and ν. Given $N \in \mathbb{N}$, we can define $\tilde{\lambda}$, $\tilde{\mu}$ and $\tilde{\nu}$ to be the partitions that induce $b(\tilde{\lambda}, \tilde{\mu}, \tilde{\nu}) = Nb + \bar{b}$. Note that $Nb + \bar{b}$ is a regular border.

For every $N \in \mathbb{N}$ we have $c^{\nu}_{\lambda\mu} > 0 \Longrightarrow c^{\tilde{\nu}}_{\tilde{\lambda}\tilde{\mu}} > 0$.

We must choose N large enough to get the other direction as well. We need an important lemma for this approach that is a slight generalization of [Buc00].

Lemma 7.14. *Given partitions $\lambda, \mu, \nu \in \mathbb{N}^n$. For each flow $f \in P^{b(\lambda,\mu,\nu)}$ that maximizes δ in $P^{b(\lambda,\mu,\nu)}$, we have that $\delta(f) \in \mathbb{Z}$.*

Proof. Given $\lambda, \mu, \nu \in \mathbb{N}^n$, $b := b(\lambda, \mu, \nu)$. Note that we can also define $b(\lambda_{\text{rat}}, \mu_{\text{rat}}, \nu_{\text{rat}}) \in \mathbb{Q}^B$ for rational vectors $\lambda_{\text{rat}}, \mu_{\text{rat}}, \nu_{\text{rat}} \in \mathbb{Q}^{3n}$ as in Figure 6.3. With the constructions from Section 7.4 we can show that $P^{b'}$ contains a rational flow for any rational border b'. Let $\{1\} \cup \{a_i | i \in H\}$ be a set of real numbers that is linearly independent over $\mathbb{Q}$ and for which $a_i > 0$ for all $i \in H$, e.g. $a_i = \sqrt{p_i}$, where p_i denotes the ith element in the sequence of primes. Define

$$\mathbb{1}^* : \mathbb{R}^H \to \mathbb{R}, h \mapsto \sum_{i \in H} a_i h(i).$$

Note that for any two distinct $h_1, h_2 \in \mathbb{Q}^H$ we have $z_1 + \mathbb{1}^*(h_1) \neq z_2 + \mathbb{1}^*(h_2)$ for all $z_1, z_2 \in \mathbb{Q}$. Define

$$\delta^*_M : F(G) \to \mathbb{R}, f \mapsto M\delta(f) + \mathbb{1}^*(\eta^{-1}(f)).$$

Then for each $(\lambda_{\text{rat}}, \mu_{\text{rat}}, \nu_{\text{rat}}) \in \mathbb{Q}^{3n}$ there exists exactly one rational flow which maximizes δ^*_M in $P^{b(\lambda_{\text{rat}},\mu_{\text{rat}},\nu_{\text{rat}})}$, because the problem is bounded and feasible, and according to [Sch98] there is always at least one vertex of $P^{b(\lambda_{\text{rat}},\mu_{\text{rat}},\nu_{\text{rat}})}$ which maximizes δ^*_M. Let $\varepsilon > 0$. Then there is $M(\varepsilon) \in \mathbb{N}$ such that for all rational $f \in F(G)$ we have

$$f \text{ maximizes } \delta^*_{M(\varepsilon)} \text{ in } P^{b'} \Rightarrow f \text{ maximizes } \delta \text{ in } P^{b'}$$

for all b' with $||b'-b|| < \varepsilon$. Define

$$\ell_\varepsilon : \mathbb{Q}^{3n} \to \mathbb{Q}^H, (\lambda_{\text{rat}}, \mu_{\text{rat}}, \nu_{\text{rat}}) \mapsto f \text{ such that}$$
$$\delta^*_{M(\varepsilon)}(f) = \max\{\delta^*_{M(\varepsilon)}(g) | g \in P^{b(\lambda_{\text{rat}}, \mu_{\text{rat}}, \nu_{\text{rat}})}\}.$$

Note that $\ell_\varepsilon(\lambda,\mu,\nu) = f$ such that $\delta(f) = \max\{\delta(g)|g \in P^b\}$ for ε small enough.

We show that $\ell_\varepsilon(\lambda,\mu,\nu)$ is integral. The function ℓ_ε is continuous, which follows from [Sch98], ch. 10.4 "Sensitivity analysis". Note that $\ell_\varepsilon(\lambda_{\text{rat}}, \mu_{\text{rat}}, \nu_{\text{rat}})$ is a vertex of the polyhedron $P^{b(\lambda_{\text{rat}}, \mu_{\text{rat}}, \nu_{\text{rat}})}$, because there is only one vector that maximizes $\delta^*_{M(\varepsilon)}$ in $P^{b(\lambda_{\text{rat}}, \mu_{\text{rat}}, \nu_{\text{rat}})}$. Let $\left(\lambda^j_{\text{rat}}, \mu^j_{\text{rat}}, \nu^j_{\text{rat}}\right)_{j \in \mathbb{N}}$ be a sequence in $\mathbb{Q}^{3n}$ with $||b(\lambda^j_{\text{rat}}, \mu^j_{\text{rat}}, \nu^j_{\text{rat}}) - b|| < \varepsilon$ and for which $b(\lambda^j_{\text{rat}}, \mu^j_{\text{rat}}, \nu^j_{\text{rat}})$ is a regular border for all $j \in \mathbb{N}$ and which satisfies $\lim_{j\to\infty}(\lambda^j_{\text{rat}}, \mu^j_{\text{rat}}, \nu^j_{\text{rat}}) = (\lambda,\mu,\nu)$. If we can show that $\ell_\varepsilon(\lambda^j_{\text{rat}}, \mu^j_{\text{rat}}, \nu^j_{\text{rat}})(\underline{A})$ is a $\mathbb{Z}$-linear combination of entries from $\lambda^j_{\text{rat}}, \mu^j_{\text{rat}}$ and ν^j_{rat} for each $\underline{A} \in H$, we have that $\ell_\varepsilon(\lambda,\mu,\nu)$ is integral, which proves the lemma.

We define $b' := b(\lambda^j_{\text{rat}}, \mu^j_{\text{rat}}, \nu^j_{\text{rat}})$. Let f maximize $\delta^*_{M(\varepsilon)}$ in $P^{b'}$. Recall that f has a regular border. Then f is shattered: If we assume the contrary, then we can raise inner vertices of a flatspace chain and increase $\delta^*_{M(\varepsilon)}$, which is a contradiction. Since $\frac{f}{2} \in P^{b'}$ is shattered as well, we can construct $\mathsf{RES}^{b'}(\frac{f}{2})$. As $f \in P^{b'}$, Lemma 6.11 shows that $\tau(\frac{f}{2}) \in P_{\text{feas}}(\mathsf{RES}^{b'}(\frac{f}{2}))$. Therefore $\tau(f) \in P_{\text{feas}}(\mathsf{RES}^{\text{sgn}b'}(\frac{f}{2}))$. In each $\frac{f}{2}$-flat rhombus $\Diamond$, $\tau(\frac{f}{2})$ uses no capacitated edge, since $\Diamond$ is f-flat as well. Thus $\tau(f)$ uses no capacitated edges in any f-flat rhombus. There can be no cycles in $\mathsf{RES}_\times(f)$ that only use uncapacitated edges, because for such cycles c we have $f + \varepsilon'\tau'(c) \in P^{b'}$ and $f - \varepsilon'\tau'(c) \in P^{b'}$ for some $\varepsilon' > 0$. This means that f is no vertex of $P^{b'}$, which is a contradiction.

We know which rhombi are f-flat and we know that $\tau(f)$ uses no capacitated edges in any f-flat rhombus and that there are no cycles on $\mathsf{RES}_\times(f)$ that use only uncapacitated edges. Therefore, if we know the troughputs on the circle border vertices, we can uniquely assign throughputs $\delta(v,f)$ to circle vertices v iteratively starting at the border respecting the flow constraints. For each vertex v we have that $\delta(v,f)$ is a $\mathbb{Z}$-linear combination of the throughput on the circle border vertices.

It remains to show that the throughputs on the border vertices are $\mathbb{Z}$-linear combinations of entries in λ^j_{rat},μ^j_{rat} and ν^j_{rat}: We delete all capacitated edges including the edges incident to o from $\mathsf{RES}(f)$ and obtain a digraph G^*. The digraph G^* has no cycles and each connected component of G^* contains at least 2 circle border vertices. Circle border vertices $s \in \mathscr{S}$ with $\delta(s,f) < \delta^{b'}_{\max}(s)$ and circle border vertices $t \in \mathscr{T}$ with $\delta(t,f) > \delta^{b'}_{\min}(t)$ are called *open*. No connected component has two open border vertices, as this would induce a well-directed cycle c on $\mathsf{RES}^{b'}(f)$ and a well-directed cycle $-c$ on $\mathsf{RES}^{b'}(f)$, which with Lemma 6.11 is a contradiction to f being a vertex of a polytope. As in each connected component there is at most one open vertex, the throughputs on the circle border vertices are $\mathbb{Z}$-linear combinations of entries in λ^j_{rat},μ^j_{rat} and ν^j_{rat}. $\square$

Proposition 7.15. *Given three partitions λ, μ, $\nu \in \mathbb{N}^n$ with $|\nu| = |\lambda| + |\mu|$ and $\bar{b}$ as described above. Let $b := b(\lambda, \mu, \nu)$ and $b(\tilde{\lambda}, \tilde{\mu}, \tilde{\nu}) = Nb + \bar{b}$. For $N > 3^{3n^2+3n} \cdot 2n^3(n-1)$ we have $c_{\lambda\mu}^{\nu} > 0 \Leftrightarrow c_{\tilde{\lambda}\tilde{\mu}}^{\tilde{\nu}} > 0$.*

Proof. One direction is clear. To figure out how big N must be for the other direction, we write the problem of optimizing δ in P^b as a linear program and do a sensitivity analysis (cf. [Sch98], ch. 10.4 "Sensitivity analysis"): We want to optimize $\delta(f) = 2\sum_{t \in \mathscr{T}} f(\{t, o\})$ subject to the constraints

$$\begin{array}{rcl} \forall v \in V \setminus \{o\}: & \sum_{e \in \delta_{\text{in}}(v)} f(e) - \sum_{e \in \delta_{\text{out}}(v)} f(e) & = 0 \\ \forall \Diamond(\underline{A}, \underline{B}, \underline{C}, \underline{D}): & \delta([\underline{A}, \underline{B}]) - \delta([\underline{D}, \underline{C}]) & \le 0 \\ \forall [\underline{A}, \underline{B}] \in \mathscr{S}: & \delta([\underline{A}, \underline{B}]) & \le b(\underline{A}) - b(\underline{B}) \\ \forall [\underline{A}, \underline{B}] \in \mathscr{T}: & -\delta([\underline{A}, \underline{B}]) & \le b(\underline{B}) - b(\underline{A}) \end{array}$$

In standard form the first equalities each become two inequalities of the form ≤ 0 and ≥ 0. Note that $\delta([\underline{A}, \underline{B}])$ is in fact a flow value $f(e)$ on a single edge e. Putting this system of inequalities in matrix notation $A'f \le b'$, then A' has at most 3 nonzero entries in each row, namely each flow inequality of a fat black vertex has 3 nonzero entries. We now determine $|V|$ and $|E|$: We have n^2 small triangles with a fat black vertex each, $\frac{n(n-1)}{2}$ upright triangles with 3 circle vertices each and 1 additional vertex o, so we get $|V| = n^2 + 3\frac{n(n-1)}{2} + 1 = \frac{5}{2}n^2 - \frac{3}{2}n + 1$ and $|E| = 3n^2 - 3n$, because we have twice as many edges than circle vertices. This results in A' having at most $2(|V| - 1) + 3|V| + 3n = \frac{25}{2}n^2 - \frac{9}{2}n + 3$ rows.

So in each square submatrix B' of A', according to the Leibniz formula, we have $\det(B') \le 3^{3n^2-3n}$. As $B'^{-1} = \operatorname{adj}(B')/\det(B')$, each entry of B'^{-1} has a bounded absolute value of at most 3^{3n^2-3n}.

As seen in [Sch98, ch. 10, eq. (22)], for a second right-hand side b'' we have

$$\left|\max\{\delta(f) \mid A'f \le b'\}\right| - \left|\max\{\delta(f) \mid A'f \le b''\}\right| \le n\Delta \left\|\delta\right\|_1 \cdot \left\|b' - b''\right\|_\infty,$$

where $\Delta = 3^{3n^2-3n}$ is an upper bound on the absolute values of entries in B'^{-1} for each square submatrix B' of A' and $\|\delta\|_1 = 2n$. In particular, since we have

$$\left\|Nb + \bar{b} - Nb\right\|_\infty = \left\|\bar{b}\right\|_\infty \le 2n\varrho_{\max}$$

by construction of $\bar{b}$, we get

$$\begin{aligned} &\left|\max\{\delta(f) \mid A'f \le Nb\}\right| - \left|\max\{\delta(f) \mid A'f \le Nb + \bar{b}\}\right| \\ &\qquad \le n \cdot 3^{3n^2-3n} \cdot 2n \cdot 2n\varrho_{\max} \le 3^{3n^2-3n} \cdot 2n^3(n-1). \end{aligned}$$

Let $\delta_{\max}(\lambda, \mu, \nu) := \max\{\delta(f) \mid f \in P^{b(\lambda,\mu,\nu)}\}$. If $c_{\lambda\mu}^{\nu} = 0$, according to Lemma 6.8(6) and Lemma 7.14 we have $\delta_{\max}(\lambda, \mu, \nu) \le 2|\nu| - 1$. Then $\delta_{\max}(N\lambda, N\mu, N\nu) \le 2N|\nu| - N$. Choose N to be larger than $3^{3n^2-3n} \cdot 2n^3(n-1)$ to get $\delta_{\max}(\tilde{\lambda}, \tilde{\mu}, \tilde{\nu}) < 2N|\nu| < 2|\tilde{\nu}|$. Therefore $c_{\tilde{\lambda}\tilde{\mu}}^{\tilde{\nu}} = 0$. $\square$

We note that the bitsize of $\tilde{\lambda}$, $\tilde{\mu}$ and $\tilde{\nu}$ is polynomial in the bitsize of λ, μ and ν. Therefore using this precalculation, the LRP-CSA can be used to determine the positivity of Littlewood-Richardson coefficients in polynomial time even in the case of weakly decreasing partitions. This again can be used to prove the Saturation Conjecture in the case of weakly decreasing partitions. However, although it does not directly use the Saturation Conjecture, the precalculation uses almost all ideas from Buch's proof of the Saturation Conjecture.

Bibliography

[AMO93] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows: theory, algorithms, and applications.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[BI08] Peter Bürgisser and Christian Ikenmeyer. The complexity of computing Kronecker coefficients. To appear in DMTCS, special issue dedicated to FPSAC '08, 2008.

[BK99] C. Bessenrodt and A. Kleshchev. On Kronecker products of complex representations of the symmetric and alternating groups. *Pacific Journal of Mathematics*, 190(2):201–223, 1999.

[BO07] Cristina M. Ballantine and Rosa C. Orellana. A combinatorial interpretation for the coefficients in the Kronecker product $s_{(n-p,p)} * s_\lambda$. *Sém. Lothar. Combin.*, 54A:Art. B54Af, 29 pp. (electronic), 2005/07.

[Buc00] Anders Skovsted Buch. The saturation conjecture (after A. Knutson and T. Tao) with an appendix by William Fulton. *Enseign. Math.*, 2(46):43–60, 2000.

[BZ92] A. D. Berenstein and A. V. Zelevinsky. Triple multiplicities for sl(r + 1) and the spectrum of the exterior algebra of the adjoint representation. *J. Algebraic Comb.*, 1(1):7–22, 1992.

[CLRS01] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* The MIT Press, 2nd edition, 2001.

[FF62] L.R. Ford and D.R. Fulkerson. *Flows in Networks.* Princeton University Press, Princeton, N.J., U.S.A., 1962.

[FFK91] Stephen A. Fenner, Lance Fortnow, and Stuart A. Kurtz. Gap-definable counting classes. In *Structure in Complexity Theory Conference*, pages 30–42, 1991.

[FH91] William Fulton and Joe Harris. *Representation Theory - A First Course*, volume 129 of *Graduate Texts in Mathematics*. Springer, 1991.

[For97] Lance Fortnow. Counting Complexity. In *Alan L. Selman, Editor, Complexity Theory Retrospective, In Honor of Juris Hartmanis on the Occasion of His Sixtieth Birthday, July 5, 1988*, volume 2. 1997.

[Ful97] William Fulton. *Young tableaux*, volume 35 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, 1997.

[Ita78] Alon Itai. Two-commodity flow. *J. ACM*, 25(4):596–611, 1978.

[Jun04] Dieter Jungnickel. *Graphs, Networks and Algorithms (Algorithms and Computation in Mathematics)*, volume 5. Springer-Verlag New York, 2004.

[Kar84] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–396, 1984.

[Kha80] L. G. Khachian. Polynomial algorithms in linear programming. *Zhurnal Vychislitelnoi Matematiki i Matematicheskoi Fiziki*, 20:53–72, 1980.

[KP95] Anil Kamath and Omri Palmon. Improved interior point algorithms for exact and approximate solution of multicommodity flow problems. In *SODA '95: Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 502–511, Philadelphia, PA, USA, 1995. Society for Industrial and Applied Mathematics.

[KT99] Allen Knutson and Terence Tao. The honeycomb model of $\mathrm{GL}_n(\mathbf{C})$ tensor products. I. Proof of the saturation conjecture. *J. Amer. Math. Soc.*, 12(4):1055–1090, 1999.

[KTW04] Allen Knutson, Terence Tao, and Christopher Woodward. The honeycomb model of gl(n) tensor products ii: Puzzles determine facets of the littlewood-richardson cone. *J. Amer. Math. Soc.*, 17(1):19–48, 204.

[Las80] Alain Lascoux. Produit de Kronecker des représentations du groupe symétrique. In *Séminaire d'Algèbre Paul Dubreil et Marie-Paule Malliavin, 32ème année (Paris, 1979)*, volume 795 of *Lecture Notes in Math.*, pages 319–329. Springer, Berlin, 1980.

[LM04] Joseph M. Landsberg and Laurent Manivel. On the ideals of secant varieties of Segre varieties. *Found. Comput. Math.*, 4(4):397–422, 2004.

[MS01] Ketan D. Mulmuley and Milind Sohoni. Geometric complexity theory. I. An approach to the P vs. NP and related problems. *SIAM J. Comput.*, 31(2):496–526 (electronic), 2001.

[MS05] Ketan D. Mulmuley and Milind Sohoni. Geometric complexity theory III: On deciding positivity of Littlewood-Richardson coefficients. cs.ArXive preprint cs.CC/0501076, 2005.

[MS06] Ketan D. Mulmuley and Milind Sohoni. Geometric complexity theory II: Towards explicit obstructions for embeddings among class varieties. cs.ArXive preprint cs.CC/0612134. To appear in SIAM J. Comput., 2006.

[Nar06] Hariharan Narayanan. On the complexity of computing Kostka numbers and Littlewood-Richardson coefficients. *J. Algebraic Combin.*, 24(3):347–354, 2006.

[OH91] Mitsunori Ogiwara and Lane A. Hemachandra. A complexity theory for feasible closure properties. In *Structure in Complexity Theory Conference*, pages 16–29, 1991.

[Pap94] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, Reading, MA, 1994.

[PV05] Igor Pak and Ernesto Vallejo. Combinatorics and geometry of Littlewood-Richardson cones. *Eur. J. Comb.*, 26(6):995–1008, 2005.

[Rem89] Jeffrey B. Remmel. A formula for the Kronecker products of Schur functions of hook shapes. *J. Algebra*, 120(1):100–118, 1989.

[Rem92] Jeffrey B. Remmel. Formulas for the expansion of the Kronecker products $S_{(m,n)} \otimes S_{(1^{p-r},r)}$ and $S_{(1^k 2^l)} \otimes S_{(1^{p-r},r)}$. *Discrete Math.*, 99(1-3):265–287, 1992.

[Ros01] Mercedes H. Rosas. The Kronecker product of Schur functions indexed by two-row shapes or hook shapes. *J. Algebraic Combin.*, 14(2):153–173, 2001.

[RW94] Jeffrey B. Remmel and Tamsen Whitehead. On the Kronecker product of Schur functions of two row shapes. *Bull. Belg. Math. Soc. Simon Stevin*, 1(5):649–683, 1994.

[Sag01] Bruce E. Sagan. *The symmetric group*, volume 203 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 2001. Representations, combinatorial algorithms, and symmetric functions.

[Sch98] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, 1998.

[Sta99] Richard P. Stanley. *Enumerative Combinatorics Volume 2*, volume 62 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1999.

[Sta00] Richard P. Stanley. Positivity problems and conjectures in algebraic combinatorics. In *Mathematics: frontiers and perspectives*, pages 295–319. Amer. Math. Soc., Providence, RI, 2000.

[Str83] Volker Strassen. Rank and optimal computation of generic tensors. *Lin. Alg. Appl.*, 52:645–685, 1983.

[Val79] Leslie G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8:189–201, 1979.