Journal of Combinatorics
Volume 6, Number 3, 353–394, 2015

Flows on honeycombs and sums of Littlewood-Richardson tableaux

Glenn Appleby and Tamsen Whitehead

Suppose μ and μ' are two partitions. We will let $\mu \oplus \mu'$ denote the *direct sum* of the partitions, defined as the sorted partition made of the parts of μ and μ'. In this paper, we define a summation operation on two Littlewood-Richardson fillings of type $(\mu, \nu; \lambda)$ and $(\mu', \nu'; \lambda')$, which results in a Littlewood-Richardson filling of type $(\mu \oplus \mu', \nu \oplus \nu'; \lambda \oplus \lambda')$. We give an algorithm to produce the sum, and show that it terminates in a Littlewood-Richardson filling by defining a bijection between a Littlewood-Richardson filling and a *flow* on a honeycomb, and then showing that the overlay of the two honeycombs of appropriate type corresponds to the sum of the two fillings.

1. Introduction

Given two partitions μ and μ', the partition made from the sorted parts of μ and μ' is called the *direct sum* of the partitions and is denoted by $\mu \oplus \mu'$. Littlewood-Richardson fillings of type $(\mu, \nu; \lambda)$ are tableaux fillings of a skew shape λ/μ of content ν, for partitions μ, ν and λ, satisfying certain constraints (definitions and examples are given below). In this paper, we give an algorithm for summing two Littlewood-Richardson fillings of type $(\mu, \nu; \lambda)$ and $(\mu', \nu'; \lambda')$, which results in a Littlewood-Richardson filling of type $(\mu \oplus \mu', \nu \oplus \nu'; \lambda \oplus \lambda')$. We show that the algorithm terminates in a Littlewood-Richardson filling by defining a bijection between a Littlewood-Richardson filling and a flow on a honeycomb, and showing that the overlay of two flows corresponds to the direct sum of the two fillings.

Definition 1.1. *A* honeycomb *is finite graph embedded in the hyperplane $\mathbb{R}^3_\Sigma = \{(x, y, z) \in \mathbb{R}^3 : x + y = z\}$ where each vertex is joined by either finite segments (joining two vertices) or semi-infinite segments (connected to a single vertex). Any segment in a honeycomb (finite or semi-infinite) is in only one of the cardinal directions pointing northwest, northeast, or* *south. Thus, each segment will have one coordinate constant along that segment. We adopt the convention that segments with constant x coordinates are in the northwest direction, constant y coordinates in the southern direction, and segments with constant z coordinates lie in the northeast direction. Segments may have multiplicities, but we require that at each vertex the* "zero tension condition" *holds, meaning that the sum of the constant x and y coordinates of the incident segments (counting multiplicities) equals the sum of the constant z coordinate at every vertex of the honeycomb.*

It should be clear from these conditions that drawing a honeycomb on top of another honeycomb (an *overlay* of honeycombs) will produce another honeycomb.

There has been an active interest in relating the combinatorics of Littlewood-Richardson fillings to the study of other mathematical objects. Interest in such objects stems from the fact that the *number* of Littlewood-Richardson fillings associated to such a triple of partitions (called the "*Littlewood-Richardson coefficient*" and denoted $c^{\lambda}_{\mu\nu}$), enumerates, among other things, the multiplicity of V_λ, an irreducible GL_n representation of highest weight λ, in the tensor product $V_\mu \otimes V_\nu$, as well as forming the structure constants for products of Schubert classes. Survey papers by Fulton [6] and Zelevinsky [17] demonstrate that Littlewood-Richardson fillings appear in many other areas beyond their early use in representation and Schubert theory, and are now used to analyze problems including the eigenvalue structure of Hermitian matrices and modules over valuation rings. In particular, the relationships between Littlewood-Richardson fillings, honeycombs and combinatorial invariants called *hives* have been studied as a tool in these investigations. Linear bijections between Littlewood-Richardson fillings, integer-valued hives and honeycombs were implicit in the work of Knutson and Tao [11] where they first appeared, and have been explicitly constructed by Pak and Vallejo [14].

In 1962 Horn [7] conjectured criteria for triples of sequences of real numbers (μ, ν, λ) for which there exist $k \times k$ Hermitian matrices M and N such that the spectrum of M is μ, the spectrum of N is ν, and the spectrum of $M + N$ is λ. A proof of Horn's conjecture was not easily found, but it became clear that its proof was related to determining when some Littlewood-Richardson coefficients $c^{\lambda}_{\mu,\nu}$ were non-zero. Progress on these problems came from deep results concerning the spectra of Hermitian matrices and representation theory given by Klyachko [10], and it seems to have been an observation of Zelevinsky that suggested the "saturation conjecture" (that $c^{N\lambda}_{N\mu,N\nu} \neq 0$ implies $c^{\lambda}_{\mu,\nu} \neq 0$) would be sufficient, along with Klyachko's work, to prove the Horn Conjecture. Knutson and Tao's original work [11] proved the saturation conjecture and made essential use of honeycombs to provide a framework to relate Littlewood-Richardson fillings to problems concerning the spectra of sums of Hermitian matrices.

Subsequent work by Knutson, Tao, and Woodward [12] constructed independent proofs of both Klyachko's and Horn's results, making use of honeycombs, hives, and related objects called *puzzles* (not utilized in our paper here). Knutson, Tao and Woodward pointed out in that paper that the overlay of honeycombs *should* correspond to the direct sum of Hermitian matrices. This goal was achieved theoretically in the work of Speyer [16], where his construction used some deep results in algebraic and tropical geometry. Earlier work by the authors [1] demonstrated how to determine a Littlewood-Richardson filling from pairs of matrices over broad classes of valuation rings. In particular, it showed how to determine algebraically the Littlewood-Richardson filling associated to pairs of matrices sharing a common matrix block decomposition. Therefore, it is of great interest that the underlying combinatorics relating Littlewood-Richardson fillings to various matrix invariants and their direct sums be made evident, and it is to this task the current paper is addressed.

Recent work of Burgisser and Ikenmeyer [3] gives an algorithm that constructs a *flow* across a directed graph (called a hive graph) associated to a triple of partitions μ, ν and λ whenever $c^{\lambda}_{\mu\nu} \neq 0$. In particular, their results give a polynomial-time test for the nonnegativity of Littlewood-Richardson coefficients. Much of that work is devoted to proving that the existence of a flow is *sufficient* to imply $c^{\lambda}_{\mu\nu} \neq 0$. Our results here will also utilize flows on hive graphs, however, we will take up the more modest task of associating a flow to an *already given* Littlewood-Richardson filling (so, for example, the flow that we determine from a filling enumerated by $c^{\lambda}_{\mu\nu}$ need not be the same as that found by Burgisser and Ikenmeyer in the case $c^{\lambda}_{\mu\nu} > 1$). We will then be able to extend a flow on a hive graph to a flow on a honeycomb, and the analysis of this flow will be an essential component to our proofs relating overlays of honeycombs to associated Littlewood-Richardson fillings. The extension to flows on honeycombs is not used in the approach taken by Burgisser and Ikenmeyer, per se, due to the fact that the shape of the honeycomb on which one defines a flow is not known until the flow on the hive graph is already determined.

In this paper, we also decompose the flow on a given honeycomb in such a way that the decomposition determines the parts of the Littlewood-Richardson filling corresponding to the honeycomb. That is, we show that a Littlewood-Richardson filling not only determines a honeycomb, but a canonical flow on the honeycomb. This construction is equivalent to the numerical definitions, of course, but we further show that the approach using flows appears naturally in the study of the *overlay* operation on honeycombs. To this end, we construct independently an algorithm on pairs of Littlewood-Richardson fillings that produces a "sum" of the filling, and we show, by means of the flow construction, that this combinatorial algorithm corresponds to the *overlay* of two honeycombs.

Given partitions $\mu^{(1)}$ and $\mu^{(2)}$ (not necessarily the same length), let $\mu^{(1)} \oplus \mu^{(2)}$ denote the partition obtained by sorting the parts of $\mu^{(1)}$ and $\mu^{(2)}$ together. We will call this the *direct sum* of the partitions $\mu^{(1)}$ and $\mu^{(2)}$. Suppose μ, ν and λ are partitions all of length n, so that $\mu = (\mu_1, \mu_2, \ldots, \mu_n)$, etc. Let $I = (i_1, i_2, \ldots, i_k)$, $J = (j_1, j_2, \ldots, j_k)$, and $K = (\kappa_1, \kappa_2, \ldots, \kappa_k)$ be (increasing) sequences of indices from $\{1, 2, \ldots n\}$ of length k. The complementary sequences, I', J' and K', are the set theoretic differences $I' = \{1, 2, \ldots n\} - I$, $J' = \{1, 2, \ldots n\} - J$ and $K' = \{1, 2, \ldots n\} - K$. Let $\mu_I = (\mu_{i_1}, \mu_{i_2}, \ldots, \mu_{i_k})$ be the associated length k sub-sequence of μ, and define $\mu_{I'}$, $\nu_J, \nu_{J'}$, $\lambda_K, \lambda_{K'}$ similarly. We also let

$$|\mu_I| = \mu_{i_1} + \mu_{i_2} + \cdots + \mu_{i_k}.$$

Note that, with these definitions, $\mu = \mu_I \oplus \mu_{I'}$, etc.

In [11] Knutson and Tao completed a proof that a triple of eigenvalues $(\mu, \nu; \lambda)$ form the spectra for Hermitian matrices M, N, and L such that $M + N = L$ if and only if a collection inequalities of the form

$$|\lambda_K| \leq |\mu_I| + |\nu_J|$$

are true, along with the trace condition $|\mu| + |\nu| = |\lambda|$. Although the collection of index sets (I, J, K) determining the inequalities go by many names, we shall call them *Horn triples*, with the inequalities they define called *Horn inequalities*, since it was Horn [7] who first conjectured that a special set of inequalities of the form above could determine spectra for Hermitian matrix pairs. It was proved subsequently that the Horn inequalities are not minimal (see [12]), and that the Hermitian matrix existence problem is implied by the *essential Horn inequalities*, determined by a subset of the Horn triples, now known as *essential triples*.

King, Tollu, and Toumazet [9] considered Horn and essential triples in order to study hives and puzzles (combinatorial invariants related to honeycombs). Their results implied that if a triple (μ, ν, λ) satisfied all the Horn inequalities, and if a Horn triple (I, J, K) could be found so that the resulting inequality was tight:

$$|\lambda_K| = |\mu_I| + |\nu_J|,$$

then any honeycomb of type (μ, ν, λ) was an overlay of a honeycomb of type $(\mu_I, \nu_J, \lambda_K)$, and one of type $(\mu_{I'}, \nu_{J'}, \lambda_{K'})$. If the Horn triple (I, J, K) was *essential*, King, Tollu, and Toumazet proved the stronger statement that the collection of all honeycombs of type (μ, ν, λ) decomposed in such a way as to imply the Littlewood-Richardson coefficient $c^{\lambda}_{\mu,\nu}$ factored as

$$c^{\lambda}_{\mu\nu} = c^{\lambda_K \oplus \lambda_{K'}}_{\mu_I \oplus \mu_{I'}, \nu_J \oplus \nu_{J'}} = c^{\lambda_K}_{\mu_I \nu_J} \cdot c^{\lambda_{K'}}_{\mu_{I'} \nu_{J'}}.$$

We note that in order for a triple (I, J, K) to be Horn triple, it is necessary, but not sufficient that

$$(1) \qquad i_i + i_2 + \cdots + i_k + j_1 + j_2 + \cdots + j_k = \kappa_1 + \kappa_2 + \cdots + \kappa_k + k.$$

However, the honeycomb decompositions appearing in general matrix decompositions are not, typically, of the sort given in the results of King, Tollu, and Toumazet [9]. For example, the honeycomb:

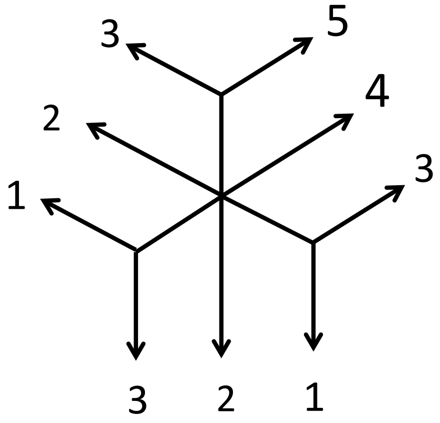
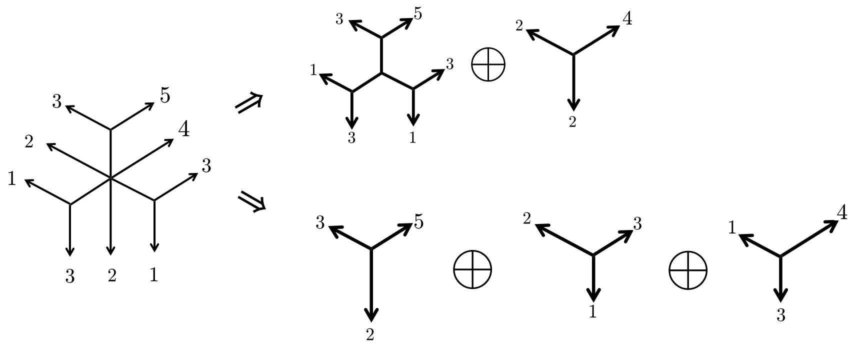


Figure 1: Honeycomb Example.

has a decomposition as an overlay in *two* inequivalent ways, as a sum of two honeycombs or three honeycombs:



In neither decomposition do the index sets satisfy Equation 1 necessary for Horn or essential triples. The first decomposition uses $((3,1),(3,1),(3,1)) \oplus ((2),(2),(2))$, while the second decomposition uses the length-one index sets $((1),(2),(1)) \oplus ((2),(3),(3)) \oplus ((3),(1),(2))$. This example shows that the relation between Littlewood-Richardson fillings and matrix (or honeycomb) decompositions is rather subtle, as there may be several inequivalent ways to realize a given honeycomb as a decomposition of honeycombs, with none determined by the Horn triples. In the above example, the first decomposition may be built using diagonal matrices, but the second would not.

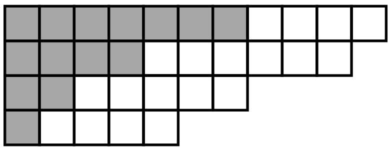
While the overlay of two honeycombs is easy to construct, the resulting Littlewood-Richardson filling formed by the overlay may be rather complicated. Conversely, it can be a challenge to detect whether a given honeycomb is an overlay of two others, and to determine the Littlewood-Richardson coefficients of the two associated honeycombs can be even more difficult (and, as the above example shows, not uniquely determined).

Consequently, it is natural to ask if one can find a *combinatorial sum* of Littlewood-Richardson fillings of types $(\mu, \nu; \lambda)$ and $(\mu', \nu'; \lambda')$ that will result in a filling of type $(\mu \oplus \mu', \nu \oplus \nu'; \lambda \oplus \lambda')$. Below, we define such a sum algorithmically and show that it terminates in a Littlewood-Richardson filling of the proper type, by defining a bijection between Littlewood-Richardson fillings of type $(\mu, \nu; \lambda)$ and flows on honeycombs of the same type. Our flow construction allows one to view the constraints appearing in the definition of Littlewood-Richardson fillings as requirements on the crossing of flows along honeycomb paths. As mentioned above, flows on hives (or their dual graphs) have been used to determine the positivity of Littlewood-Richardson coefficients [3], but our construction of flows on honeycombs is new and greatly simplifies our analysis. It appears to provide a link between the combinatorial constructions of Littlewood-Richardson coefficients and the variational definitions of eigenvalues and invariant factors. A numerical formula for the convolution of two hives (corresponding to the overlay of two honeycombs) appeared in [5] and was also described in [16]. While this formula is efficient for computations, it does not provide an avenue to the underlying combinatorics of the Littlewood-Richardson filling to which it is equivalent.

2. Notation and definitions

A *partition* of a positive integer n is a sequence of positive integers $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_k)$ such that $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_k$ and $\alpha_1 + \alpha_2 + \cdots + \alpha_k = n$. Each α_k is called a *part* of the partition. The *length* of the partition, $\mathit{length}(\alpha)$, is the number of parts, k, and the *weight* or *size* of the partition, $|\alpha|$, is n. See [13], chapter 1, and [15].

For two partitions α and β, we write $\beta \subseteq \alpha$, to mean $\beta_k \leq \alpha_k$ for all $k \geq 1$. This notation is suggested by the fact that if we represent the partitions by decreasing, left-justified rows of boxes (called the *diagram* or *Ferrers diagram* of the partition), then $\beta \subseteq \alpha$ implies the diagram for β fits inside the diagram of α. When $\beta \subseteq \alpha$, we will denote by α/β the *skew shape* consisting of the diagram of α, with the diagram of β removed. In the example below, $\alpha = (11, 10, 7, 5)$, $\beta = (7, 4, 2, 1)$ and and the skew shape α/β consists of the *unshaded* boxes of α.



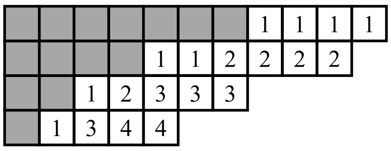
We create a *skew tableau* (or simply a *tableau*, if $\beta = \emptyset$), by filling the boxes of α/β with positive integers. Define the *content* of α/β to be the composition $\nu = (\nu_1, \nu_2, \ldots, \nu_m)$, where $\nu_i =$ the number of i's in the skew shape. If the entries in each row weakly increase, and the entries in each column strictly increase, then the tableau is said to be a *semi-standard Young tableau of shape α/β and content ν*. Note that the definition of the content of a tableau differs from the definition of the content of an unfilled Ferrers diagram. See [13] for this distinction. In general we will use the word *content* to refer collectively to entries in the skew shape, and the word *entry* to refer to the number in a particular box of the skew shape.

Our central combinatorial definition is the following:

Definition 2.1. *Let μ, ν, and λ be partitions, with $\mathit{length}(\mu)$, $\mathit{length}(\nu) \leq \mathit{length}(\lambda) \leq r$, $\mu \subseteq \lambda$ and $|\nu| = |\lambda/\mu|$. Then a semi-standard tableau of shape*

λ/μ and content ν is Littlewood-Richardson *if the word created by reading the entries of λ/μ, from top to bottom, right to left, is a* Yamanouchi word, *that is to say, it has the property that any prefix of the word contains at least as many i's as $(i+1)$'s, or, equivalently, the number of i's in rows i through j of the tableau is greater than or equal to the number of $(i+1)$'s in rows $(i+1)$ through $(j+1)$. We call such a filled diagram a* Littlewood-Richardson tableau.

For example, the skew tableau below is a Littlewood-Richardson tableau of λ/μ and content ν, where $\lambda = (11, 10, 7, 5)$, $\mu = (7, 4, 2, 1)$ and $\nu = (8, 5, 4, 2)$. The tableau word $1\,1\,1\,1\,2\,2\,2\,2\,1\,1\,3\,3\,3\,2\,1\,4\,4\,3\,1$ has the prefix condition given above.



We can express this definition more formally as follows:

Definition 2.2. *Let μ, ν, and λ be partitions, with length(μ), length$(\nu) \leq$ length$(\lambda) \leq r$. Let $S = (\lambda^{(0)}, \lambda^{(1)}, \cdots, \lambda^{(r)})$ be a sequence of partitions in which $\lambda^{(0)} = \mu$, and $\lambda^{(i)} \subseteq \lambda^{(i+1)}$. The sequence S is called a* Littlewood-Richardson Sequence *of type $(\mu, \nu; \lambda)$ if there is a triangular array of nonnegative integers $F = \{k_{ij} : 1 \leq i \leq j \leq r\}$ (called the* filling*) such that, for $1 \leq i \leq j$, $\lambda_j^{(i)} = \mu_j + k_{1j} + \cdots + k_{ij}$, subject to the conditions $(LR1)$, $(LR2)$, $(LR3)$ below. We shall say, equivalently, that any such set F determines a* Littlewood-Richardson filling *of the skew shape λ/μ with content ν.*

(LR1) (Sums) For all $1 \leq i \leq j \leq r$,

$$\mu_j + \sum_{s=1}^{j} k_{sj} = \lambda_j, \quad \textit{and} \quad \sum_{s=i}^{r} k_{is} = \nu_i.$$

(LR2) (Column Strictness) For each j, for $2 \leq j \leq r$ and $1 \leq i \leq j$ we require $\lambda_j^{(i)} \leq \lambda_{(j-1)}^{(i-1)}$, that is,

$$\mu_j + k_{1j} + \cdots k_{ij} \leq \mu_{(j-1)} + k_{1,(j-1)} + \cdots + k_{(i-1),(j-1)}.$$

(LR3) (Word Condition) For all $1 \le i \le r-1$, $i \le j \le r-1$,

$$\sum_{s=i+1}^{j+1} k_{(i+1),s} \le \sum_{s=i}^{j} k_{is}.$$

Let $LR(\mu, \nu; \lambda)$ *denote the set of Littlewood-Richardson fillings of type* λ/μ *and content* ν.

These two definitions are equivalent. The skew diagram λ/μ can be filled with ν_1 1's, ν_2 2's, etc. if and only if $|\nu| = |\lambda/\mu|$. The first equality of $(LR1)$ ensures that the sum of the number of boxes in row j of the filled diagram (including the empty boxes of the parts of μ) is λ_j, the j-th part of the partition λ, while the second equality ensures that the sum of the number of i's in all the rows of λ/μ is ν_i, the i-th part of the partition ν. $(LR2)$ says that the numbers in the filling are strictly increasing down columns. Lastly, $(LR3)$ indicates that the number of i's in rows i through j is greater than or equal to the number of $(i+1)$'s in rows $(i+1)$ through $(j+1)$. As a consequence, the filling must also be semi-standard.

Condition $(LR3)$ implies that the filling cannot have any entry bigger than i in row i. So row 1 can only contain 1's, row 2 can only contain 1's and 2's and so on.

Definition 2.3. *Given partitions* μ, ν, *and* λ, *we shall let* $c^{\lambda}_{\mu\nu}$ *denote the number of Littlewood-Richardson fillings of the skew shape* λ/μ *with content* ν. *The nonnegative integer* $c^{\lambda}_{\mu\nu}$ *is called the* Littlewood-Richardson coefficient *of the partitions* μ, ν, *and* λ. *So* $c^{\lambda}_{\mu\nu} = |LR(\mu, \nu; \lambda)|$.

3. The algorithm

We now present the algorithm that takes two Littlewood-Richardson tableaux of types $(\mu, \nu; \lambda)$ and $(\mu', \nu'; \lambda')$, and produces a Littlewood-Richardson filling of type $(\mu \oplus \mu', \nu \oplus \nu'; \lambda \oplus \lambda')$. We present it in outline form, and follow it with a detailed example.

Input: Two Littlewood-Richardson fillings of shape λ/μ and λ'/μ', and content ν and ν'.

Output: A Littlewood-Richardson filling of shape $(\lambda \oplus \lambda')/(\mu \oplus \mu')$ and content $\nu \oplus \nu'$.

We prove in Section 6 that all steps of the algorithm can be performed, and that the algorithm terminates in a Littlewood-Richardson filling of the proper type.

1. Relabel the content of both diagrams.
   (a) Sort the parts of ν and ν' to form a single partition, $\nu \oplus \nu'$.
   (b) Change the content of each Littlewood-Richardson filling so that the part of ν or ν' that became $(\nu \oplus \nu')_2$ becomes 2's, the part of ν or ν' that became $(\nu \oplus \nu')_3$ becomes 3's, etc.
2. Append the Littlewood-Richardson filling of shape λ'/ν' below that of shape λ/ν. We call this the *sum* diagram. All subsequent steps are performed on the sum diagram. Let $n = \text{length}(\lambda) + \text{length}(\lambda')$.
3. Sort the rows so that they are weakly decreasing in length.
4. Sort the columns that contain (empty) boxes of μ or μ' so that the empty boxes are at the top of the column. Leave the filled boxes in each column in the same relative order.
5. Fix "i in row i" violations: (A consequence of $LR3$, the Word condition, is that a Littlewood-Richardson filling only contains entries greater than or equal to i in row i.) For $i = 1$ to n
   (a) If row i contains $(i+k)$'s $(k > 0)$, then find the northeast-most box containing an i such that the strand of i's weakly northeast of that box is equal in length to the strand of $(i+k)$'s weakly northeast of it. Swap this strand of i's with the strand of $(i+k)$'s.
   (b) Let $k = k - 1$ and repeat Step 5(b) until $k = 0$.
6. Find the northeast most "bad" box (north dominates east), where a bad box is defined as one such that
   (a) The box contains some number i.
   (b) That i creates an ($LR3$) (Word) violation, so the number of $(i-1)$'s in rows strictly above the bad box is one less than the number of i's weakly northeast of the bad box, OR
   (c) That i creates an ($LR2$) (Column-Strict) violation, so that i lies directly above an entry k such that $k < i$.
7. Fix the violation caused by the bad box:
   (a) If the bad box creates an ($LR3$) violation, then, beginning at the bad box, find the southwest-most box containing an $i-1$, with south dominating, (called "the corner box") such that the strand of i's weakly to the northeast of the corner box is equal in length to the strand of $(i-1)$'s weakly northeast of the corner box. Swap the the strands of i's from the bad box to the corner box, with the strand of $(i-1)$'s from the bad box to the corner box.
   (b) If the bad box creates an ($LR2$) violation, swap the i and k.

8. Continue working southeast, identifying bad boxes and fixing the violations until no more exist.
9. If necessary, weakly order the content of each row.

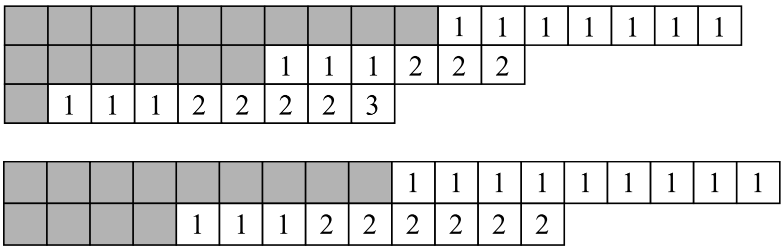


Figure 2: The two Littlewood-Richardson tableaux.

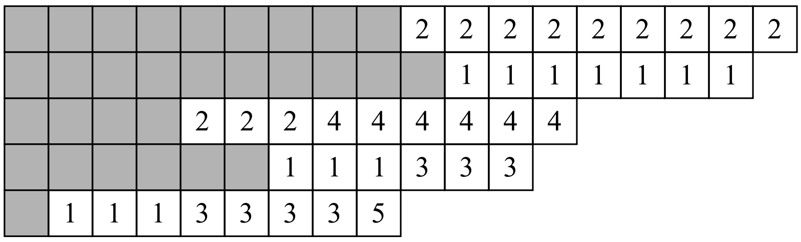


Figure 3: The initial summed diagram, with relabeled content.

We now present a detailed example, explaining each step more completely. Here, $\mu = (10, 6, 1)$, $\nu = (13, 7, 1)$, $\lambda = (17, 12, 9)$ $\mu' = (9, 4)$, $\nu' = (12, 6)$, $\lambda' = (18, 13)$. Hence $\mu \oplus \mu' = (10, 9, 6, 4, 1)$, $\nu \oplus \nu' = (13, 12, 7, 6, 1)$, and $(\lambda \oplus \lambda') = (18, 17, 13, 12, 9)$. The original Littlewood-Richardson tableaux are shown in Figure 2.

In Step 1, we relabel the content of both diagrams so that the summed diagram will contain 1's, 2's, ... and n's where $n = \text{length}(\lambda) + \text{length}(\lambda')$. For our example, since $\nu_1 \geq \nu'_1 \geq \nu_2 \geq \nu'_2 \geq \nu_3$, the partition ν gives us the number of 1's, 3's and 5's in the sum and ν' the number of 2's and 4's.

In Steps 2 and 3 of the algorithm we combine the parts of λ and λ' (without moving any entries within a given row) into one tableau, and sort the rows so that the row lengths weakly decrease. If two parts of λ and λ' are equal, we will order the rows so that the row whose content is smaller is the higher row of $\lambda \oplus \lambda'$. See Figure 3.

Next, in order to make the parts formed by μ and μ' into the partition $\mu \oplus \mu'$, in Step 4, we swap unfilled boxes in any column with the filled boxes above them, retaining the relative order of the filled boxes within that column. Filled boxes stay in their original columns, even if this leads to row-strict violations, as it does in rows 2 and 4 of Figure 4 below.

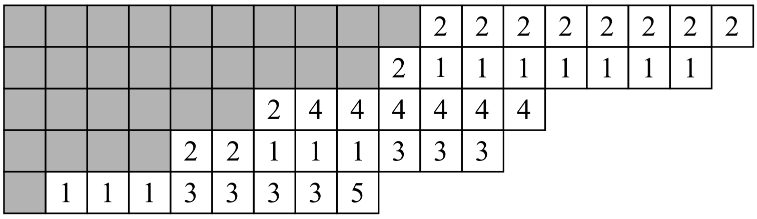


Figure 4: The summed diagram, after ordering the parts of μ and μ'.

At this point, the outer shape of the diagram is $\lambda \oplus \lambda'$, and the unfilled boxes inside form the shape $\mu \oplus \mu'$. These shapes of the summed diagram are correct, but the filling of the remaining skew shape $(\lambda \oplus \lambda')/(\mu \oplus \mu')$ will typically still have word, column-strict and/or row-strict violations (corresponding to conditions $(LR3)$, $(LR2)$ and the containment requirement $\lambda^{(i)} \subseteq \lambda^{(i+1)}$, respectively, in the definition of Littlewood-Richardson fillings).

First, in Step 5, we correct violations to the rule that entries in the boxes of row i are less than or equal to i. (Recall that this rule is a consequence of $(LR3)$, the word condition.) We correct these "i in row i" violations by following the "northeast" rule: Suppose there are some entries with value $(i+k)$ appearing in row i (with $k > 0$). We will prove in Section 6 that there cannot be more of these entries than there are i's in the diagram below row i. We look for the northeast-most box in a row below row i that contains an i such that the number of i's weakly northeast of that box equals the number of $(i+k)$'s weakly northeast of it (which includes the $(i+k)$'s of "i in row i" violations in row i, and may also include $(i+k)$'s in lower rows). There is, therefore, a horizontal strip, or "strand" of violating $(i+k)$ entries in row i and below, and a horizontal strip of i's of equal length. (The use of the term "strand" will be further justified when the relation between these portions of the Littlewood-Richardson filling are related to flows on honeycombs, to be discussed below. In particular, in Section 6, we will prove that such a strand of $(i+k)$'s always exists.) We then swap these strands.

In Figure 5 below, to fix the "1 in row 1" violation, we will swap a strand of 1's and 2's. Beginning at the last 2 in row 1, we look for the northeast-most box such that the strand of 1's and 2's weakly northeast of that box are equal in length. In Figure 5 below, we depict this process. Note that in

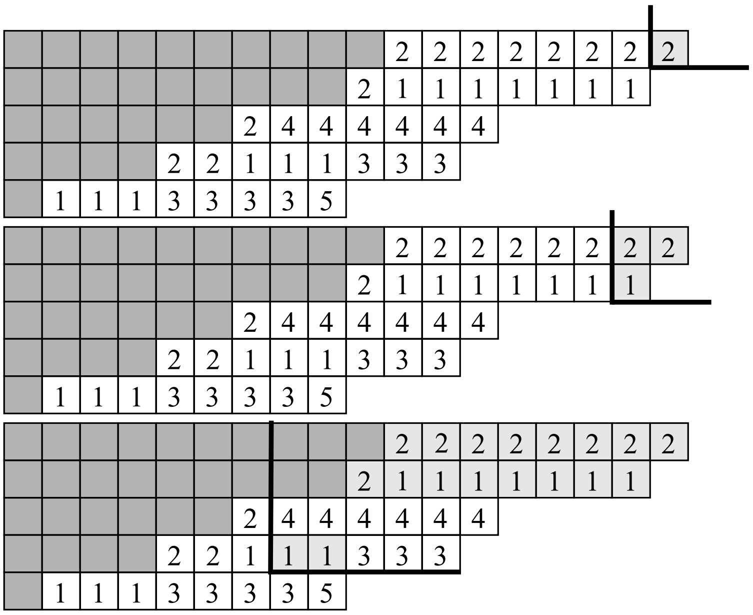


Figure 5: Sequence of summed diagrams, searching for strands of 1's and 2's of equal length (shaded).

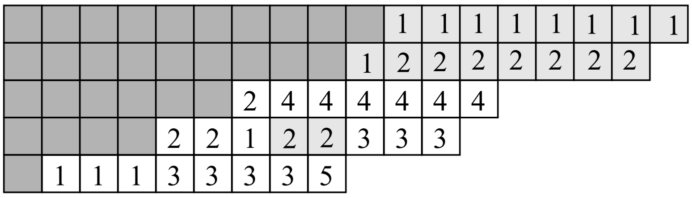


Figure 6: The summed diagram, with strands of 1's and 2's swapped.

the last diagram, the shaded strands of 1's and 2's both have length nine. We swap the contents of the two strands, which movesthe 2's out of row 1, and also moves a 2 out of row 2. The result of this swap is shown in Figure 6.

Once we have replaced any $(i+k)$'s appearing in row i with i's, if $k-1 > 0$, we repeat this process, now replacing any $(i+k-1)$'s in row i with a strand of i's appearing in lower rows, etc., until there are only the numbers 1 through i appearing in row i. The feasibility of this (and every) step of the algorithm will be addressed in our proof below for the correctness of the algorithm.

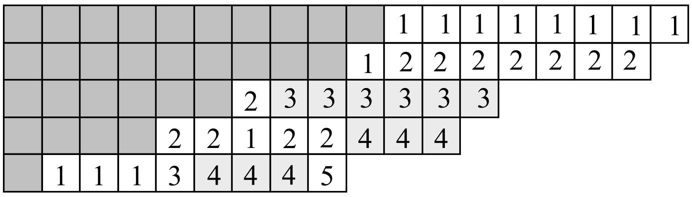


Figure 7: The summed diagram, with strands of 3's and 4's swapped.

We then continue this process, working down to the bottom row, until the content of each row is no bigger than the row index. In Figure 7, we have swapped strands of 3's and 4's to fix the "3 in row 3" problem.

The summed diagram may still contain column-strict ($(LR2)$), word ($(LR3)$), or row-strict violations. We will ignore the row-strict violations for now but will fix column-strict or word violations as they appear, by finding the northeast most "bad box".

Definition 3.1. *We say a filled box in a filling of the skew shape* $(\lambda \oplus \lambda')/(\mu \oplus \mu')$ *is a* bad box *if it is a witness to a violation of either the word or column-strictness conditions in Definition 2.2. In particular:*

In the case of word violations, a bad box in row j*, say, is one such that for some* i*, the number of* $(i+1)$*'s northeast of the bad box (including the bad box) exceeds the number of* i*'s through row* $j-1$*.*

For column-strict violations a bad box is a box that contains an entry i *which lies directly above an entry* k *where* $k < i$*.*

So, in Steps 6 and 7, we iteratively fix word or column-strict violations in the summed diagram by locating the northeast-most bad box of the diagram (meaning we look first for bad boxes in the northern-most row, and then choose the eastern-most bad boxes among these).

If the northeast-most bad box corresponds to a word violation, we fix it by swapping strands of i's and $(i-1)$'s of equal length. We address how to identify these strands in detail below. In the case of column-strict violations, we simply swap the two adjacent entries in the column so that they are strictly decreasing down the column.

Continuing with our example, working from the northeast corner of the top diagram of Figure 8, we first encounter a column-strict violation in the form of a pair of 3's above a pair of 2's, in rows 3 and 4, which we swap, as shown in the bottom diagram.

We then encounter (in row 3, column 7) a bad box corresponding to a word violation: the ten 2's in rows 2 and 3 exceed the nine 1's in rows 1 and 2. To identify the strand of 1's and 2's to be swapped, find the "corner

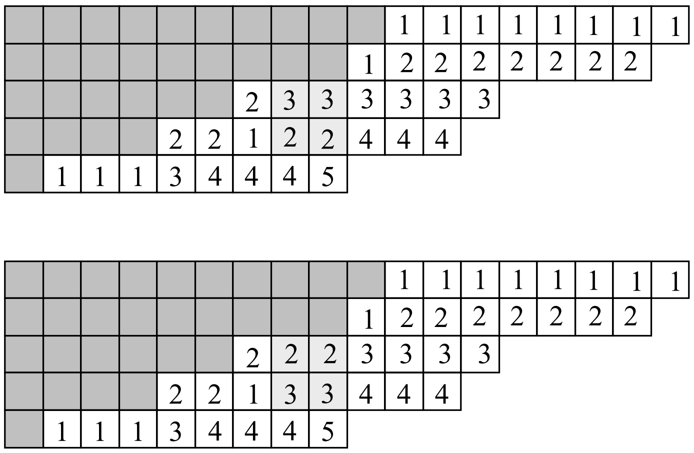


Figure 8: The summed diagram, with the pair of shaded 2's and 3's swapped.

box": the northeast-most box (with northern-most dominating), such that inside the frame made of boxes above and to the right of the corner box, the strand of 1's has length equal to the strand of 2's. In our example, the corner box is in row 4, column 7. We then swap these strands of 1's and 2's from the bad box to the corner box. The swap is shown in Figure 9 below.

In our example, there is one final bad box (in row 4, column 6), causing a word violation (the 2's through row 4 exceed the 1's through row 3). The

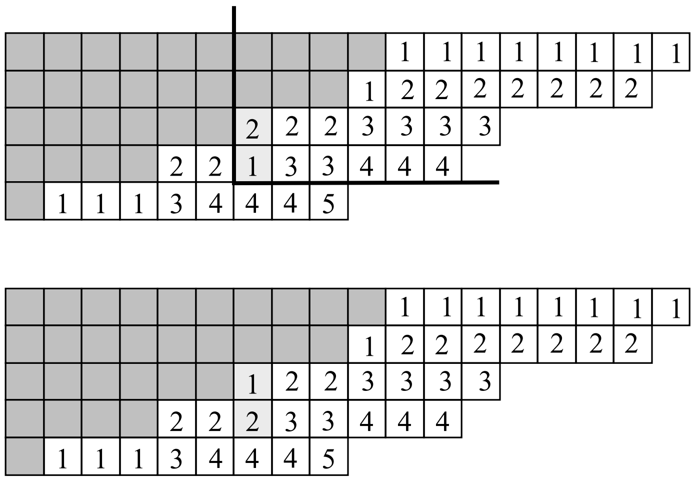


Figure 9: The summed diagram, with the shaded strands of 1's and 2's identified and swapped.

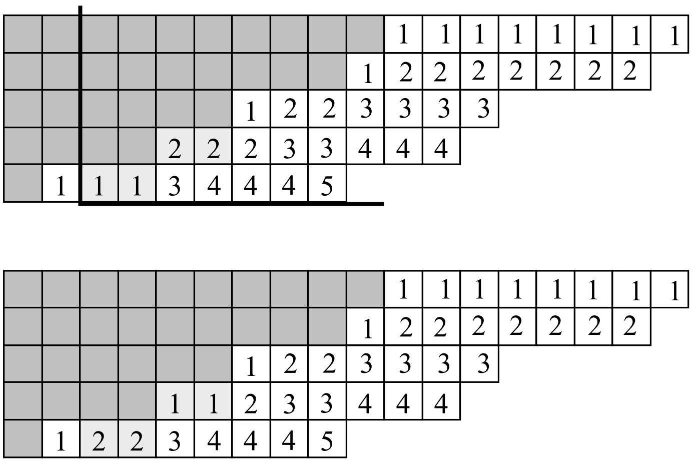


Figure 10: The summed diagram, with the final strands of 1's and 2's identified and swapped.

corner box is in row 5, column 3, as this is the northeast-most box in which the number of 1's and 2's in boxes above and to the right are equal. We swap the strand of 1's and 2's from the bad box to the corner box, as shown in Figure 10.

After fixing all word or column-strict violations, any remaining row-strict violations can be fixed, by ordering the contents of each row. The algorithm terminates, and the summed filling, as we shall show, is Littlewood-Richardson.

4. The dual graph and flows

We will now proceed to develop the machinery necessary to prove that the above algorithm necessarily terminates in a Littlewood-Richardson filling for the summed diagram. In particular, we will review the construction of *hives*, first found in [11] in the work of Knutson and Tao. (See also the work of Buch [2] for a more detailed discussion of hives, and Pak and Vallejo [14] for their connection to Littlewood-Richardson fillings.)

Definition 4.1. *A* hive *of size r is a triangular array of numbers $(h_{ij})_{0\leq j\leq i\leq r}$ that satisfy the* rhombus inequalities*:*

1. Right: $h_{i,j} + h_{i-1,j-1} \geq h_{i,j-1} + h_{i-1,j}$*, for* $1 \leq j < i \leq r$*.*
2. Vertical: $h_{i,j-1} + h_{i,j} \geq h_{i+1,j} + h_{i-1,j-1}$*, for* $1 \leq j \leq i \leq r$*.*
3. Left: $h_{ij} + h_{i-1,j} \geq h_{i,j+1} + h_{i-1,j-1}$*, for* $1 \leq j < i < r$*.*

A hive of size 5 is shown below.

$$\begin{array}{ccccccccccc}
 & & & & & h_{00} & & & & & \\
 & & & & h_{10} & & h_{11} & & & & \\
 & & & h_{20} & & h_{21} & & h_{22} & & & \\
 & & h_{30} & & h_{31} & & h_{32} & & h_{33} & & \\
 & h_{40} & & h_{41} & & h_{42} & & h_{43} & & h_{44} & \\
h_{50} & & h_{51} & & h_{52} & & h_{53} & & h_{54} & & h_{55}
\end{array}$$

Note that we will always normalize hives so that $h_{00} = 0$.

These rhombus inequalities are so named because the terms in each inequality form a rhombus in the hive, made by adjacent entries in the array such that that the upper acute angle points to the right, vertically, and to the left, respectively. In each case the inequality asserts that the sum of the entries of the obtuse vertices of the rhombus is greater than or equal to the sum of the acute entries.

Note that the definition of a hive allows its entries to be real numbers, but we shall restrict our attention to hives whose entries are nonnegative integers.

Let $HIVE_r$ denote the set of nonnegative integer-valued hives of size r. We shall say a hive H is of *type* $\tau(H) = (\mu, \nu, \lambda)$ (for sequences of nonnegative integers μ, ν, and λ of length r) when

$\mu_i = h_{i0} - h_{(i-1),0}$	(the downward differences of entries along the left side)
$\nu_i = h_{ri} - h_{r(i-1)}$,	(the rightward differences of entries along the bottom)
$\lambda_i = h_{(i)(i)} - h_{(i-1)(i-1)}$	(the downward differences of entries along the right side).

Let

$$H(\mu, \nu, \lambda)_r = \{H \in HIVE_r : \tau(H) = (\mu, \nu, \lambda)\}.$$

As a consequence of the rhombus inequalities, the type of a hive H will be a triple of partitions of *weakly decreasing* nonnegative integers and, thus, form a triple of partitions. A necessary condition for the existence of a hive $H \in HIVE(\mu, \nu, \lambda)_r$ is:

$$|\mu| + |\nu| = |\lambda|.$$

As an example, the hive below is an element of $HIVE(\mu, \nu, \lambda)_5$ where $\mu = (10, 9, 5, 3, 1)$, $\nu = (12, 11, 7, 6, 1)$ and $\lambda = (18, 16, 12, 11, 8)$.

$$H = \begin{array}{ccccccccccc} & & & & & 0 & & & & & \\ & & & & 10 & & 18 & & & & \\ & & & 19 & & 27 & & 34 & & & \\ & & 24 & & 34 & & 42 & & 46 & & \\ & 27 & & 38 & & 48 & & 54 & & 57 & \\ 28 & & 40 & & 51 & & 58 & & 64 & & 65 \end{array} .$$

Figure 11: An example of a hive in $HIVE_5$.

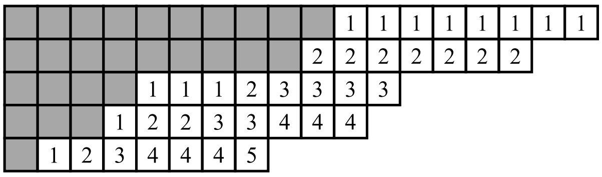
Pak and Vallejo [14] gave an injective map ϕ from $LR(\mu, \nu; \lambda)$ to $H(\mu, \nu, \lambda)_r$ with

$$\phi(\{k_{ij}\}) = \{h_{ij}\},$$

defined by

(2) $$h_{pq} = \sum_{i=1}^{q} \sum_{j=1}^{p} k_{ij} + \sum_{s=1}^{p} \mu_s.$$

For example,



is the Littlewood-Richardson tableau of type $((10, 9, 5, 3, 1), (12, 11, 7, 6, 1); (18, 16, 12, 11, 8))$ corresponding to the hive given above. Note that h_{pq} equals the sum of the parts of μ through row p added to the number of 1's, 2's, ..., and q's in rows 1 through p. So for example, $h_{52} = 51$ equals $\mu_1 + \ldots \mu_5 +$ (the total number of 1's and 2's through row 5).

In fact, ϕ is a surjection onto the set of nonnegative integer-valued hives, and so we also have

$$\phi^{-1}(\{h_{pq}\}) = \{k_{ij}\}$$

where

(3) $$k_{ij} = (h_{(j-1)(i-1)} + h_{ji}) - (h_{(j-1)i} + h_{j(i-1)})$$

for $i < j$. This difference is in fact the rhombus difference for right-slanted rhombi, and so is nonnegative. In the example above, $k_{24} = (h_{31} + h_{42}) - (h_{32} + h_{41}) = (34 + 48) - (42 + 38) = 2$.

In order to verify our algorithm does indeed terminate in a Littlewood-Richardson filling, we will need several more definitions, which will reformulate hive combinatorics in a more convenient form. We can view a hive as a vertex labeling on an underlying undirected graph, as shown:

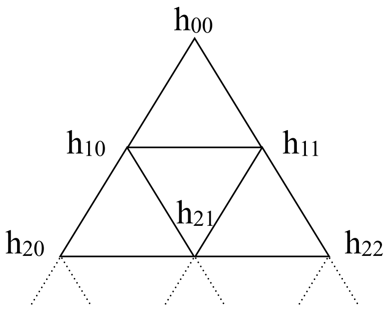


Figure 12: The hive graph.

So, for example, our hive shown in Figure 11 corresponds to the hive graph:

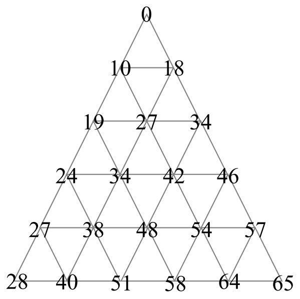


Figure 13: The hive graph for the hive in Figure 11.

We then can create the dual graph to the hive graph:

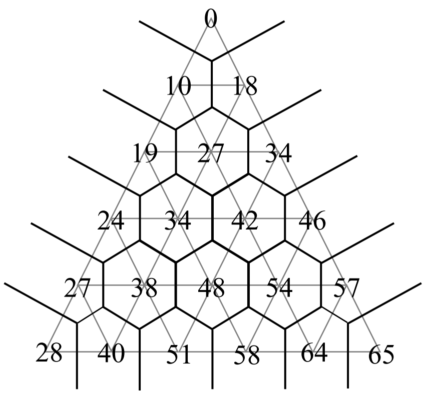


Figure 14: The dual graph.

Each hive value now lies in a unique cell of the dual graph. We weight each edge in the dual graph by the positive difference of the adjacent hive entries from the cells on either side of the edge:

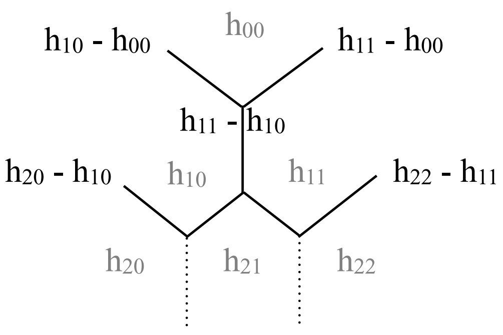


Figure 15: The weighted dual graph.

In Figure 15, the grey numbers are the hive entries, and the dark numbers are the corresponding edge weights of the dual graph.

So for our hive example, the top portion of the weighted dual graph is:

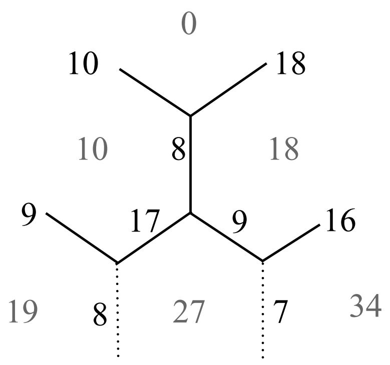


Figure 16: The top of the weighted dual graph.

By using Equation 2, we can compute, in terms of the filling, the weights of the edges on the boundary of the dual graph. The weights of the left edges are the parts of μ, the bottom weights are the parts of ν and the right edge weights are the parts of λ. Similarly, we may find formulas for edge weights for the interior edges. Using the diagram below to identify certain edge weights in a typical dual graph cell,

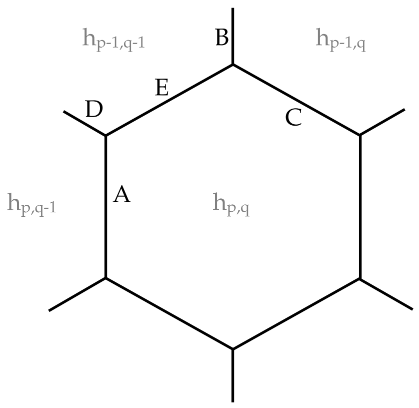


Figure 17: Edge weights identified.

we have (from Equation 3), that, for $q < p$, $k_{qp} = A - B = C - D$. Then, we have formulas for the edge weights A, B, C and E. We will call these the *flow equations*:

(4) $A = k_{q,q} + k_{q,q+1} + \cdots + k_{q,p}$
(5) $B = k_{q,q} + k_{q,q+1} + \cdots + k_{q,p-1}$
(6) $C = \mu_p + k_{1,p} + k_{2,p} + \cdots + k_{q,p}$
(7) $D = \mu_p + k_{1,p} + k_{2,p} + \cdots + k_{q-1,p}$
(8) $E = (\mu_p + k_{1,p} + k_{2,p} + \cdots + k_{q-1,p}) + (k_{q,q} + k_{q,q+1} + \cdots k_{q,p})$

Note, in Figure 17 above, that the flow equations imply

$$A + D = E = B + C.$$

Hence if we direct each edge as shown below, we may view these formulas as specifying a capacity for a *flow* such that the flow into any vertex equals the flow out of it. See Figure 18.

We will always identify the directions of our flows, called the “μ,” “ν” and “λ” directions, as shown in Figure 19.

For example, Figure 20 gives the directed graph with flow capacity, corresponding to the hive considered above in Figure 13.

We will make even more explicit use of the flow equations numbered 4 through 8. Consider the weight given for the edge labeled “C” in Figure 17:

$$C = \mu_p + k_{1,p} + k_{2,p} + \cdots + k_{q,p}.$$

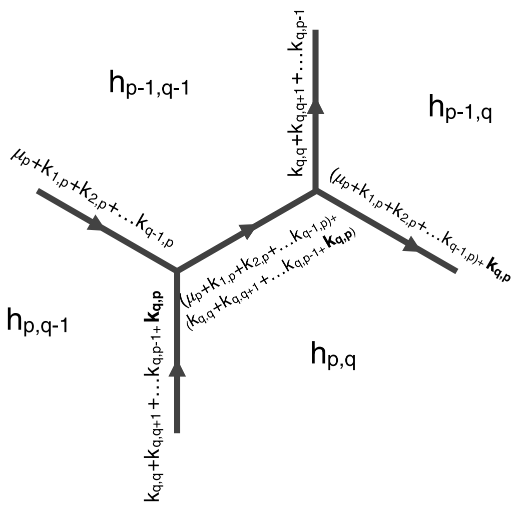

$h_{p-1,q-1}$
$h_{p-1,q}$
$h_{p,q-1}$
$h_{p,q}$
$k_{q,q}+k_{q,q+1}+\dots k_{q,p-1}$
$\mu_p+k_{1,p}+k_{2,p}+\dots k_{q-1,p}$
$(\mu_p+k_{1,p}+k_{2,p}+\dots k_{q-1,p})+\mathbf{k_{q,p}}$
$(\mu_p+k_{1,p}+k_{2,p}+\dots k_{q-1,p})+$
$(k_{q,q}+k_{q,q+1}+\dots k_{q,p-1}+\mathbf{k_{q,p}})$
$k_{q,q}+k_{q,q+1}+\dots k_{q,p-1}+\mathbf{k_{q,p}}$

Figure 18: Flows given by edge capacities on the dual graph.

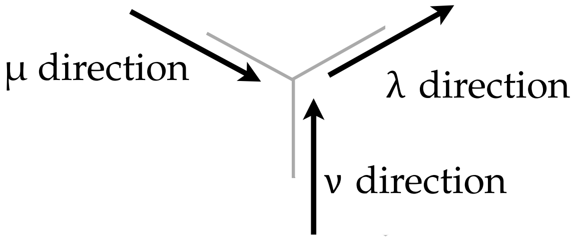


Figure 19: Labels for flow directions.

Not only will we use this equation to assign the *total* capacity for a flow across edge C, but we will use the terms in the sum to define a *decomposition* of this flow into pieces (determined uniquely by the underlying Littlewood-Richardson filling). This flow is *different* from a typical flow on a weighted, directed graph in that we identify the *constituent sources* of the flow, meaning that we track how much of the total flow on an edge comes from *each part* of μ and from *each part* of ν.

We will represent this decomposition by splitting the total flow along an edge into differently-colored strands. In a Littlewood-Richardson filling, each box of λ is either empty or filled with a number from 1 to r. We will color each strand according to these choices. We will use (for now) one color to represent any portion of the flow corresponding to a part μ_p, for any p. We will use a different color for each number 1 through r appearing in the

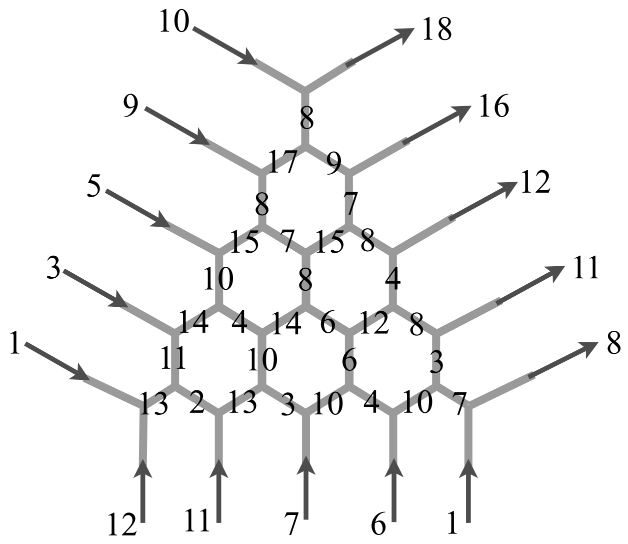
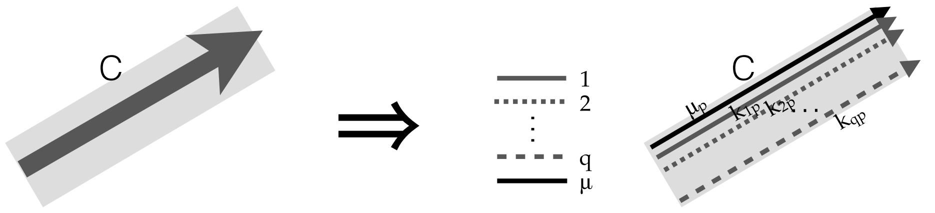


Figure 20: Dual graph edges, with flow capacities labeled.

Littlewood-Richardson filling, so that all terms of the form k_{ij}, for any j, will be colored the same. That is, instead of a single edge C with total flow $\mu_p + k_{1,p} + k_{2,p} + \cdots + k_{q,p}$, we will split the flow into colored strands:



Thus, while the *total* capacity (or weight) of an edge in the dual graph is given by the sums of terms in the flow equations 4 through 8, we will draw the flow on that edge using several colors, by assigning an *individual* weight to each colored strand in the flow for that edge.

A "row" of the dual graph is a path consisting of the alternating edges in the μ- and λ-directions, beginning on the left hand side. A graph corresponding to a hive of size r will have r-many such rows, which we will number from top to bottom. The equations for edges D, E, and C above show each edge in some row p has a weight decomposition containing a single μ_p term in it, with no other parts of μ appearing. Thus a flow of weight μ_p flows *into* the graph from the left side at row p, and this portion of the total flow,

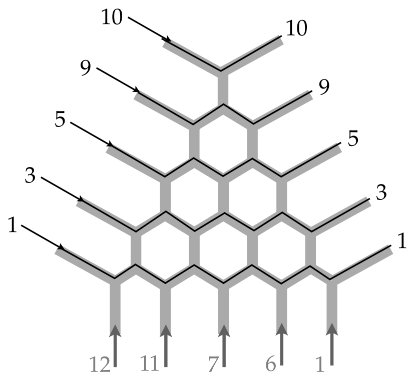


Figure 21: μ-flow through the dual graph.

called the "μ-flow," traverses the row (where it is joined by other parts of the flow determined by parts of the filling), flowing out of the diagram along the p-th edge with total weight λ_p. See Figure 21. We color the μ-flow portion of the total flow in the diagram with the same color, and in the "canonical flow" (defined below) given by the Littlewood-Richardson filling, the parts of the μ-flow necessarily do not intersect. When we consider colored flows on overlayed honeycombs, the μ-flow from one honeycomb may be colored differently than the μ-flow of some other, overlayed honeycomb, and such flows may indeed cross each other.

Using the flow equations we see that the weight of the i-th vertical edge along the bottom of dual graph, numbered from left to right, has weight ν_i. We will call the i-th *spine* of the dual graph the path starting from the i-th vertical edge at the bottom, and then proceeding north (in the ν direction), and then northeast (in the λ direction), alternating at each vertex. As a consequence of the flow equations 4 through 8, the flow along the vertical portions of the i-th spine will have *only* terms k_{ij} appearing in it. We color all portions of the flow labeled k_{ij}, for any j, the same color. The weights of the vertical edges of the spine will decrease as we proceed up and right by the amount k_{ij}, $(j = r, r-1, \ldots 1)$ with this portion of the flow diverted to the right. Thus, the colored portion of the flow associated to the parts k_{ij}, called the "i-flow" (since these terms determine the placement of the i's in the Littlewood-Richardson filling) will form a tree, rooted in the i-th vertical edge along the bottom of the dual graph. See Figure 22.

Figure 23 shows the μ-flow and the 1-flow (the dashed lines) of the dual graph.

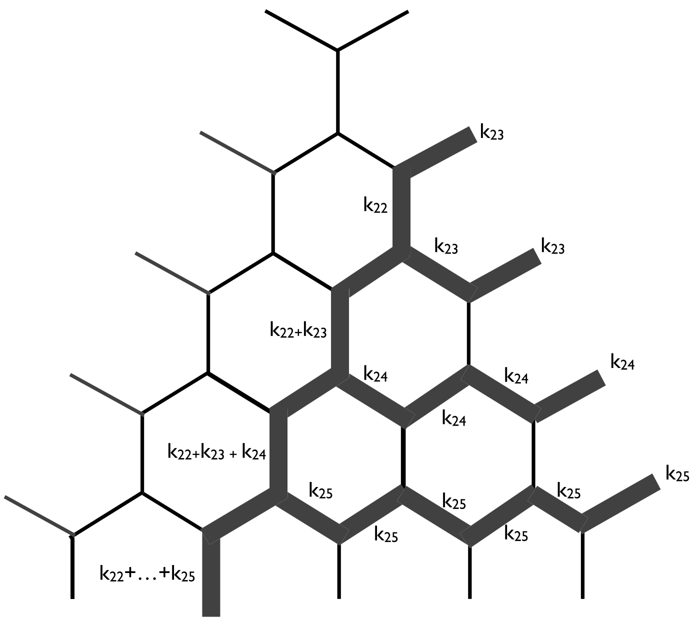


Figure 22: 2-flow of the dual graph.

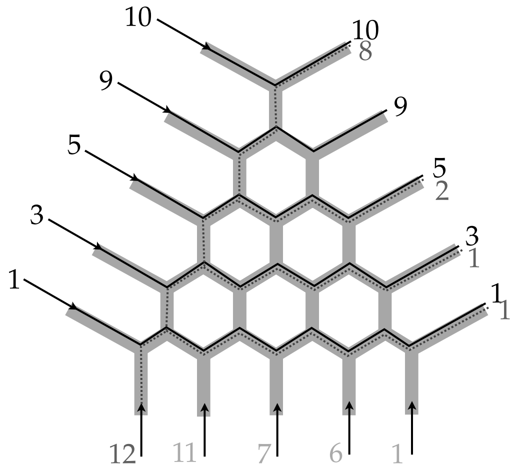


Figure 23: The 1-flow through the dual graph.

In particular, the 1-flow coming out of the edge of weight λ_i is indicated by the second number of that outflow, below the weight of the outgoing μ-flow for that edge.

We create the flow for the remaining parts of ν (the 2's through 5's, or in general, the 2's through the number of parts of ν) in the same manner. The flow of the 2's (the grey lines) is included in the diagram below.

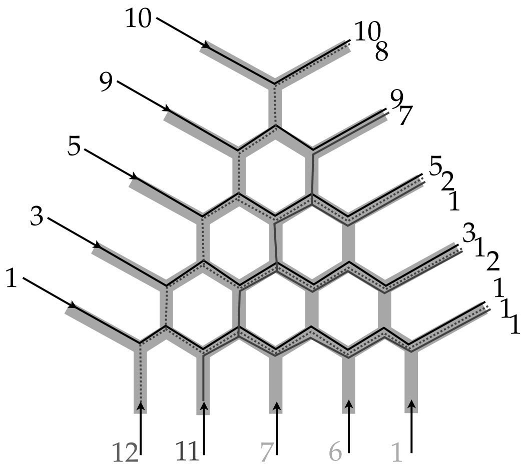


Figure 24: The 1 and 2 flow through the dual graph.

Let us summarize the above discussion. As a consequence of the flow equations, we obtain a colored flow along the edges of the dual graph such that:

1. The μ-flow in row j flows along row j with weight μ_j.
2. The i-flow begins in the i-th vertical edge with weight ν_i, and proceeds up the i-th spine.
3. At row j, the difference in the weight of the vertical edge of the i-th spine below row j, and the vertical edge of the spine above it, is k_{ij}.
4. In row j, starting at the i-th spine, we have a portion of the i-flow moving to the right by the amount k_{ij}, and flowing out the j-th edge.

We call this the "canonical flow" of the dual graph associated to the given Littlewood-Richardson filling. It is uniquely determined by the filling by means of the flow equations numbered 4 through 8. We may immediately read off this filling associated to the canonical flow on the dual hive graph by reading off the weights of the colored strands along each outgoing edge on the right side.

It is easy to show that the canonical flow of a weighted dual graph (subject to the inflow/outflow constraints at vertices) produces a Littlewood-Richardson filling as well. This follows from the observation that if a dual hive graph is assigned a set of weights such that at each vertex the inflow/outflow constraints are satisfied, then we may solve, uniquely if we assign the top hive value 0, for values for a hive (assigning numbers to the cells of the graph) that give rise to these edge weights. From the hive we may again determine the associated Littlewood-Richardson filling.

5. From flows on dual graphs to flows on honeycombs

Given the weighted dual graph (coming from the hive, with edge weights given by the positive difference of adjacent hive entries) we can view the edge weights on the edges incident to a given vertex as coordinates on that vertex for a point in a subspace S of $\mathbb{R}^3$, given by $(x, y, z) \in S$ if and only if $x+y = z$. So, each vertex in the dual graph corresponds to a coordinate in S, as shown in Figure 25 for the top of the weighted dual graph in Figure 20. Our vertex labeling (x, y, z) always has x as the label of the edge in the μ-direction, y as the label for the edge in the ν-direction, and z as the label of the edge in the λ-direction.

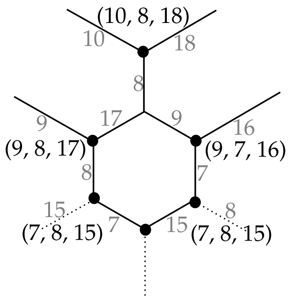


Figure 25: The weighted dual graph viewed in $\mathbb{R}^3$.

More than one vertex can correspond to the same point in S, as is the case with $(7, 8, 15)$.

This correspondence between vertices of the weighted dual graph and points in a hyperplane in $\mathbb{R}^3$ creates the *honeycomb* for the dual graph. We first plot in S the vertices of the dual graph (possibly with multiplicities) using the coordinates of the edge weights as described above. Vertices in the honeycomb are connected if the corresponding vertices were adjacent in the dual graph of the hive. By the flow requirements of vertices of the dual graph, all points (x, y, z) of a honeycomb lie on the hyperplane in $\mathbb{R}^3$ given by $x + y = z$, so we may represent a honeycomb as a planar graph lying in this subspace. The honeycomb for our hive example is shown below.

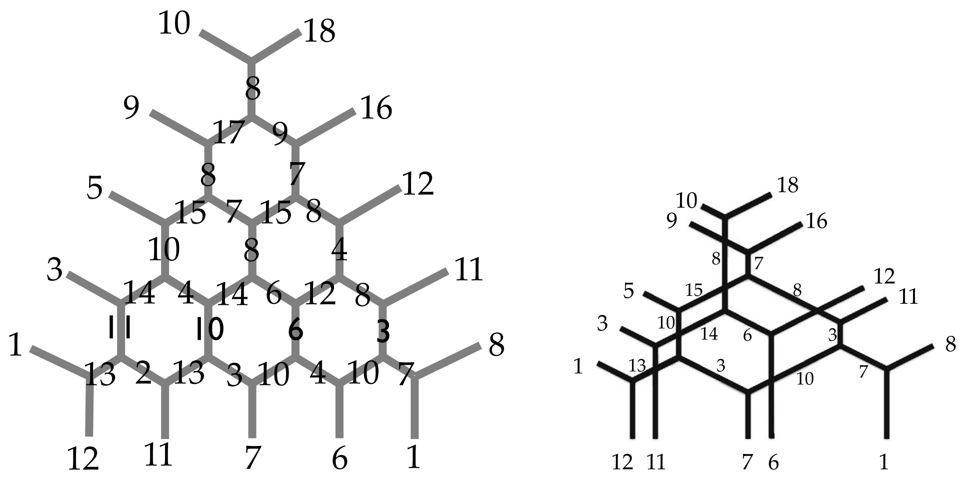


Figure 26: The hive and corresponding honeycomb.

Notice that all the points on a given edge in the honeycomb will have one coordinate that is constant. For example, the edge labeled 9 starting on the left side of the honeycomb consists of points of the form $(9, y, z)$. This will allow us to define a capacity for a flow on an edge: an edge's capacity is defined to be the value of the constant coordinate of the edge. As with the hive, we define the "μ direction" to be along edges going northwest to southeast, the "ν direction" to be along edges going south to north and the "λ direction" to be on edges going southwest to northeast. Note that the capacity of each edge in these three directions is the appropriate part of μ, ν or λ, so the "type" of the honeycomb is the same as the type of the Littlewood-Richardson filling (and the hive and the weighted dual graph).

Transverse crossings in the honeycomb correspond to two adjacent vertices that have the same coordinates in the dual graph. In Figure 27, the two vertices would be labeled $(m, n, m + n)$.

The canonical flow of the dual graph of a hive translates to a canonical flow on the honeycomb in the obvious manner. For example, Figure 28 below shows the honeycomb with the μ-flow and the flow of the 1's and 2's. This is the same flow as shown on the dual graph in Figure 24. The converse holds as well: a canonical flow on a honeycomb produces a Littlewood-Richardson filling. Knutson and Tao [11] show that every honeycomb may be associated to a unique hive, from which a canonical flow on the honeycomb may be determined, and from this flow we may read off the Littlewood-Richardson filling by reading the weights of the colored strand in each outgoing edge on

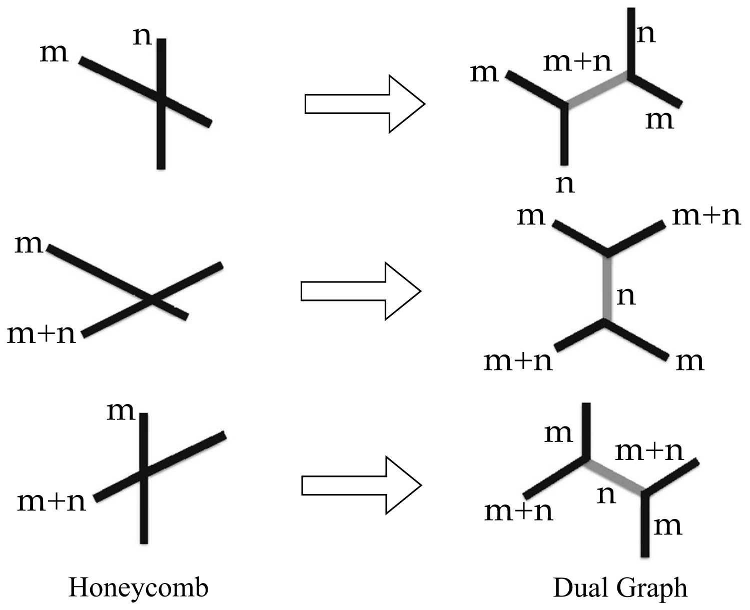


Figure 27: Transverse crossings correspond to dual graph vertices with the same coordinates.

the right side. The total weight of the i-th edge (from the top) is λ_i, and the weight of the μ-flow in that edge gives the size of the i-th row of μ inside of λ, with the weights of the other strands giving the size of $k_{1i}, k_{2i}, \ldots, k_{ii}$. We shall call these the outgoing strand weights of the canonical flow.

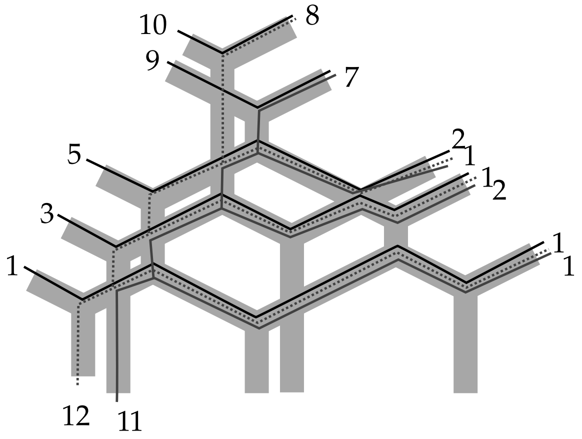


Figure 28: μ-flow and the flow of the 1's and 2's in the honeycomb.

The consequence of the above analysis is the following:

Theorem 5.1. *The outgoing strand weights of the canonical flow on a honeycomb determine its associated Littlewood-Richardson filling.*

6. Proof that the algorithm works

In this section we prove:

Theorem 6.1. *The algorithm described in Section 3 takes Littlewood-Richardson fillings of type $(\mu, \nu; \lambda)$ and $(\mu', \nu'; \lambda')$, respectively, producing a Littlewood-Richardson filling of type $(\mu \oplus \mu', \nu \oplus \nu'; \lambda \oplus \lambda')$. That is, the algorithm terminates in a Littlewood-Richardson filling of the proper type.*

This result will actually come as a consequence of the following:

Theorem 6.2. *Let $\mathcal{H}$ be a honeycomb determined by a Littlewood-Richardson filling of type $(\mu, \nu; \lambda)$, and $\mathcal{H}'$ be a honeycomb determined by a Littlewood-Richardson filling of type $(\mu', \nu'; \lambda')$. Then the outgoing strand weights of the canonical flow of the overlay honeycomb $\mathcal{H} \oplus \mathcal{H}'$ are the same as the parts of the filling appearing in the output of the algorithm applied to the two given Littlewood-Richardson fillings.*

And so, in particular, the output of the algorithm is a Littlewood-Richardson filling of the appropriate type.

We prove this result by working with the two corresponding canonical flows on the honeycombs associated with the Littlewood-Richardson fillings of type $(\mu, \nu; \lambda)$ and $(\mu', \nu'; \lambda')$.

Our plan is as follows: In [11] Knutson and Tao provide a definition of a honeycomb that is independent of the underlying dual graph of a hive, and show that the associated hive may be derived directly from the honeycomb. Furthermore, this definition implies that the overlay of two honeycombs (in the hyperplane $x + y = z$ in $\mathbb{R}^3$) is another honeycomb (indeed, the overlay is one of the ways that multiplicities for edges and vertices may be realized). However, the overlay of the two associated canonical flows is *not* typically a canonical flow for the overlay honeycomb. We shall show that the process of resolving non-canonical flows for the overlay of two honeycombs will match, step by step, the algorithm we presented for sums of Littlewood-Richardson tableaux, thus proving the algorithm terminates in a Littlewood-Richardson filling for the sum of the partitions.

So we begin with the two Littlewood-Richardson fillings. These correspond to two honeycombs, each with their canonical flow. Note again how we may read the size of the parts of the filling (the size of the μ_i and the k_{ij}'s) by reading the weights of the flow on the outgoing edges on the right side.

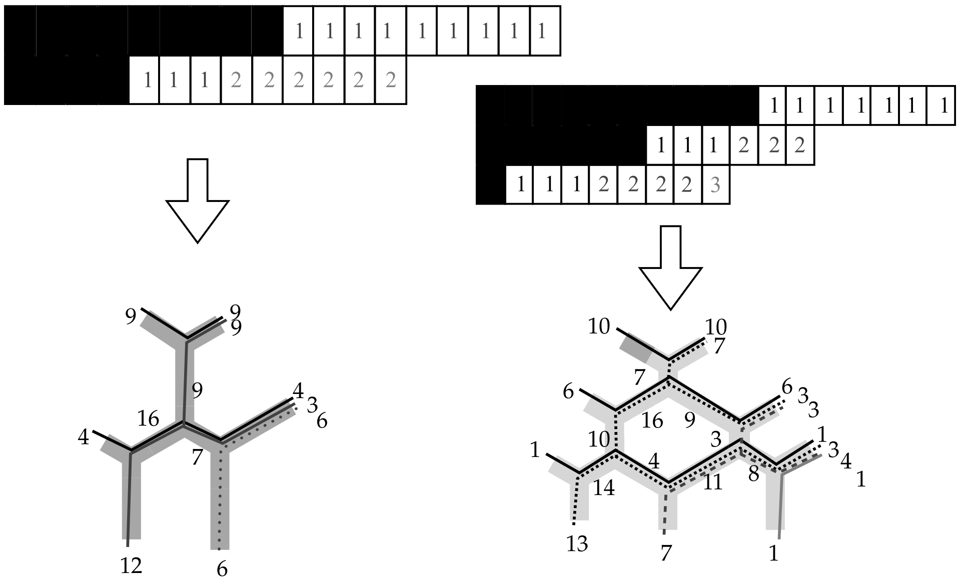


Figure 29: The two honeycombs associated with the two Littlewood-Richardson fillings.

We create the honeycomb that will correspond to the summed tableaux by overlaying the two honeycombs. The associated flows for each honeycomb are also overlayed, producing what we shall call the "canonically wrong" flow. By the term "canonically wrong" flow we mean the uniquely determined non-canonical flow that results from the overlay of two canonical flows. We shall resolve this flow, and the steps we follow in transforming it into a canonical flow will both mimic and justify the steps of our algorithm on the corresponding sum of Littlewood-Richardson fillings. The circled vertices in Figure 30 show the corresponding "errors" in the honeycomb flow, that is, departures from it being canonical.

This overlay honeycomb has type $(\mu \oplus \mu', \nu \oplus \nu'; \lambda \oplus \lambda')$; the content comes from relabeling the content from each diagram (as was done in the summed diagram,) so that the number of 1's is greater than or equal to the number of 2's, etc. If two parts of $\nu \oplus \nu'$ are equal, assign the smaller number to the part that ends in the higher row of the honeycomb.

The overlayed flow will not necessarily be canonical. In fact, it will have a non-canonical flow that mimics exactly the errors in the initial summed filling. We can determine the "canonically wrong filling" of the shape $\lambda \oplus \lambda'$ from the canonically wrong flow on the honeycomb above by using the same procedure as before. In the edge whose total weight corresponds to the i-th

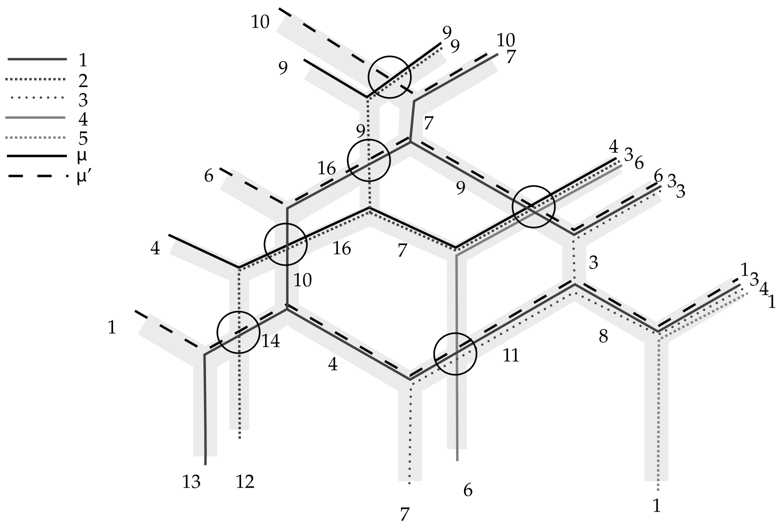


Figure 30: Overlay of two honeycombs and the "canonically wrong" flow.

part of $\lambda \oplus \lambda'$, use the weights of the strands to determine the filling in the i-th row of the shape $\lambda \oplus \lambda'$. That is, if a strand in row i of the honeycomb (counting from the top) comes from a μ or μ' flow (canonical or not), put that many un-numbered boxes in row i. Recall, we will actually retain the *order* of the strands in each outgoing edge. We proceed down through each strand, and use the weight of the strand to tell us how many numbers to put in row i, and *which* number we use is determined by the color of the strand.

In the above honeycomb, with the associated canonically wrong flow, the first row has a μ-flow of 9, and a strand of 2's of weight 9, so we put nine empty boxes in the first row of $\lambda \oplus \lambda'$, followed by nine 2's. The third row, for example, has a μ-flow of 4, followed by a strand of 2's of weight 3, then a strand of 4's of weight 6. This gives, in the third row of $\lambda \oplus \lambda'$, four empty boxes, followed by three 2's, then six 4's, etc.

▨	▨	▨	▨	▨	▨	▨	▨	▨	2	2	2	2	2	2	2	2	2
▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	1	1	1	1	1	1	1	
▨	▨	▨	▨	2	2	2	4	4	4	4	4	4					
▨	▨	▨	▨	▨	▨	1	1	1	3	3	3						
▨	1	1	1	3	3	3	3	5									

Figure 31: Filling of $\lambda \oplus \lambda'$, determined by the honeycomb overlay.

We will use this procedure, of determining a filling from a flow, *throughout* the process of resolving a canonically wrong flow into its uniquely determined canonical flow. That is, independent of the twisting or turnings of strands in the interior of the honeycomb, the weights along the outgoing edges will always uniquely determine *some* filling of the shape $\lambda \oplus \lambda'$. We shall show that each operation on the honeycomb, towards resolving it into a canonical flow, will have the effect on the associated filling of $\lambda \oplus \lambda'$ that follows the steps described in our algorithm. At the point our flow becomes canonical, then our previous results will guarantee that the associated filling must indeed be Littlewood-Richardson.

Let us begin to analyze how to resolve a canonically wrong flow on the overlay honeycomb into a canonical one. The overlay of honeycombs and their flows can create two types of non-canonical flow. Each non-canonical flow has a unique method of resolution:

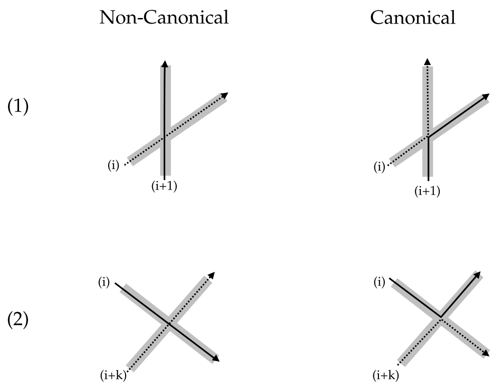


Figure 32: Types of non-canonical flows in the honeycomb overlay and their corresponding canonical flows.

We now show how to correct the overlay flow to produce its canonical flow and that this canonical flow is indeed the one associated to the corrected Littlewood-Richardson filling of the sum of the two diagrams. This will prove Theorem 6.2 and, as a consequence, Theorem 6.1.

First, the two types of non-canonical flows that can be seen in the overlay flow (Figure 32) correspond to the following types of errors in the sum filling: Type (1) flow errors are are reflected in the diagram as "i in row i" or word violations; Type (2) flow errors are represent "μ switching" or column-strict violations in the associated diagram. There are no λ violations in the overlay; the parts of $\lambda \oplus \lambda'$ are ordered from largest to smallest in the honeycomb. Although it is unimportant to the final, corrected, flow, to be consistent with the order in which errors are corrected in the Littlewood-Richardson diagram, we will correct each type of non-canonical flow from top to bottom, east to west, as encountered.

A "μ switching" error occurs when the honeycomb overlay produces a transverse crossing of the μ-flow, as shown in Figure 33 below. In this figure we see the portion of the μ-flow associated to μ_i starts above the flow associated to μ'_j on the left, but in the overlay of the two honeycombs $\mathcal{H}$ and $\mathcal{H}'$ the flows cross, reflecting, as we shall see, a μ-switching error in the associated filling of $\lambda \oplus \lambda'$.

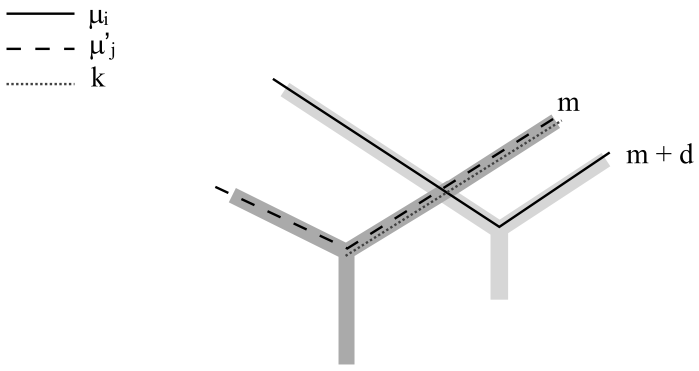


Figure 33: μ switching error in flow.

The μ-flow determined by $\mu \oplus \mu'$ comes from different honeycombs, but in the uncorrected overlay of the honeycombs, a portion of the μ-flow of larger weight crosses below a portion of smaller weight. We correct this error by mimicking the process done on the Littlewood-Richardson diagrams. Let $\mu_i = m + d$ and $\mu'_j = m$. We need to swap $m + d$ units of flow on the upper edge, which will consist of μ-flow and ν-flow, with $m + d$ units of flow on the lower edge, which will be entirely μ-flow. Note that this is always possible because each edge's label (and hence the total flow on that edge) corresponds to its z-coordinate, which is always at least as big as its x coordinate, the μ-flow value.

So, we correct the crossing by taking $m + d$ units of flow (μ, 1's, 2's, etc. in order) on the upper edge and swapping that with $m + d$ units of flow on the lower edge. In our example, there is a flow of weight 6 associated to the μ-flow leaving the edge labeled λ_4, but only a μ-flow of weight 4 leaving the edge labeled λ_3. Since $\lambda_3 \geq \lambda_4$, there is enough total flow in the λ_3-row to match the μ-flow of weight 6 leaving the λ_4 edge. Thus, we swap the μ-flow of weight 6 leaving row 4 with the μ-flow of weight 4, along with a portion of the 2-flow in that row of weight 2 (we proceed downwards in the strand in the λ_3). Note that by using the exiting weights of the flows in the honeycombs to determine the associated fillings (before and after the swap) we exactly mimic Step 4 of the algorithm in Section 3.

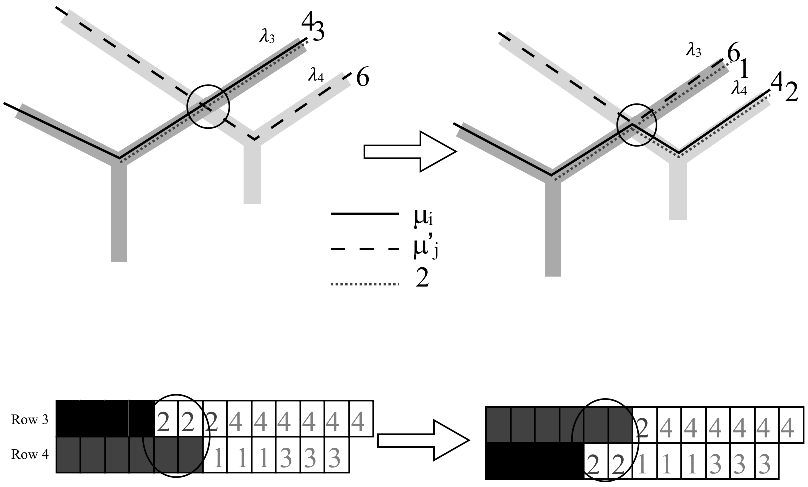


Figure 34: Strand swapping to correct the μ-flow error.

After all μ switching errors are fixed (our example has one more at the top circled crossing in Figure 30), we correct "i in row i" errors. These errors occur when, in overlaying the two honeycombs, an $i + k$ ends in row i. In the figure below, $k = 1$, and we see a portion of the 2-flow (of weight 8) appearing in row 1 (whose total weight is λ_1). In reading off the associated filling of $\lambda \oplus \lambda'$ from the edge weights of the honeycomb flow, we immediately see that there would be eight 2's appearing in row 1 of the diagram, which the algorithm would seek to correct. See Figure 35.

We correct this error much as in the previous case, by swapping equivalent amount of flow at the "bad" intersection, so that the flow is canonical there. Let's describe more completely the nature of this error. As shown below, this sort of transverse crossing in the overlay of honeycombs comes from two vertices in the dual graph that map onto the same point in the honeycomb. Such a non-canonical flow on the overlayed honeycombs must correspond to a flow on the dual graph in which the dashed flow on the edge labeled "$\ell - n$" consists of n units flowing *backwards*. This retrograde flow cannot occur in a canonical flow on the dual graph but the analysis of it on the dual graph is somewhat complicated, and this is one reason analyzing flows on honeycombs is simpler. See Figure 36.

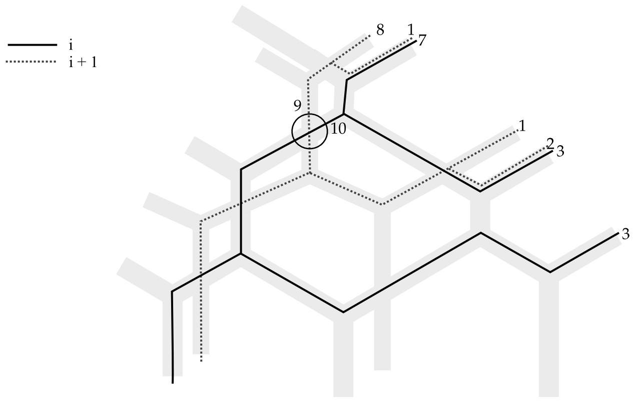


Figure 35: "i in row i error in the overlay of the honeycombs.

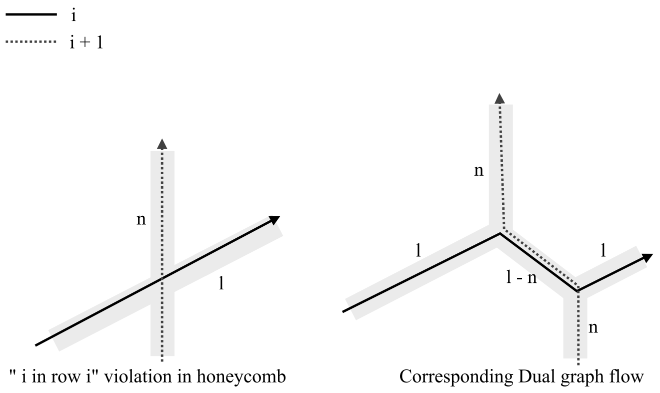

i
i + 1
n
l
" i in row i" violation in honeycomb
n
l
l - n
l
n
Corresponding Dual graph flow

Figure 36: Dual graph flow.

Back to our example. Recalling that the sum of the flows in the μ and ν directions equals the flow in the λ direction, we see that $\ell \geq n$. So, we exchange n units of i flow in the λ direction with n units of $i+k$ flow in the ν direction. For example, to fix the error at the circled vertex, we need to swap $n = 9$ units of 2-flow leaving the circled vertex with 9 units of 1-flow leaving the circled vertex. The swapped flow is depicted by the dashed (2-flow) and black (1-flow) lines. Again, if we determine the associated fillings before and after the swap (using the exiting weights along the right edges), we recover the associated step in our algorithm. See Figure 37.

Below, Figure 38 shows the flow swap on the "3 in row 3" problem, indicated by the non-canonical flow circled.

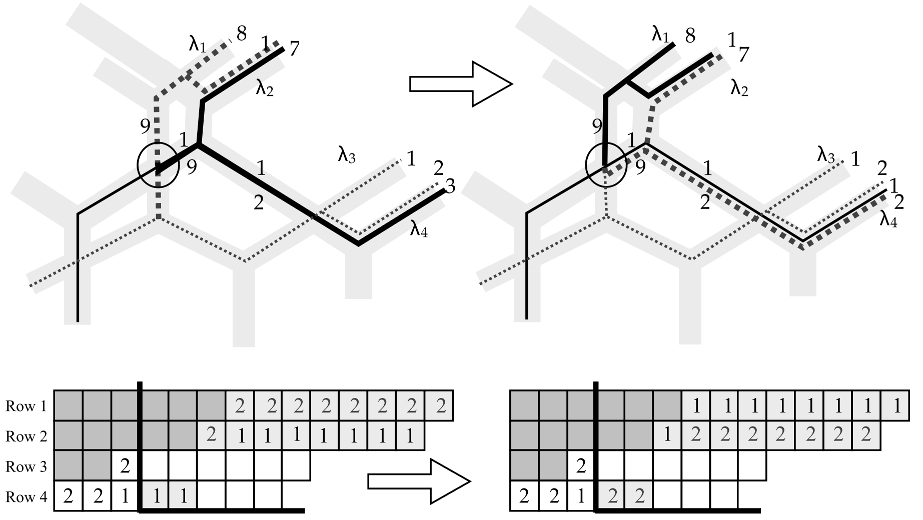


Figure 37: The flow, before and after the swap.

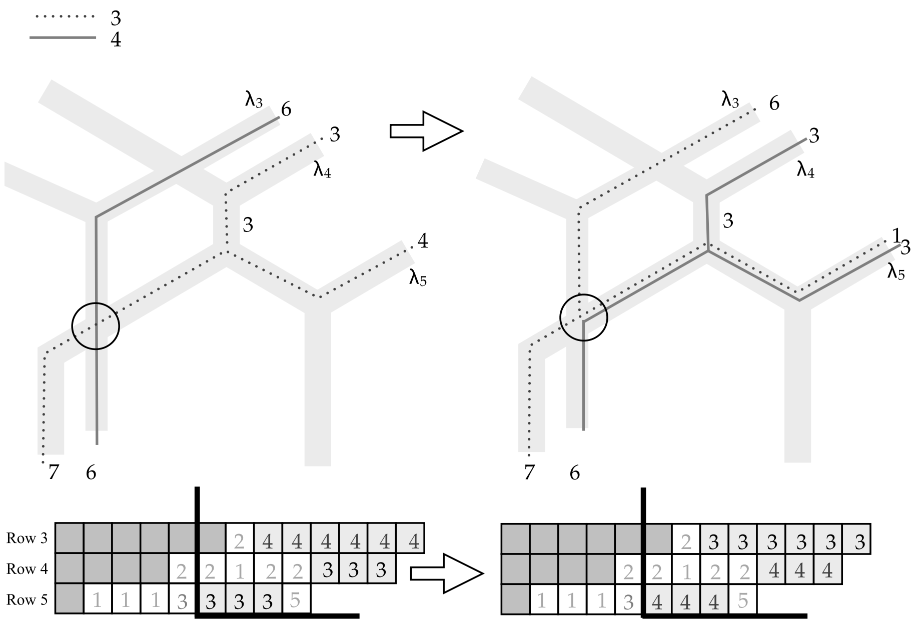


Figure 38: “3 in row 3” flow fixed, and the corresponding rows of the Littlewood-Richardson diagram.

Continuing to follow the algorithm described in Section 3, we now tackle word or column-strict flow violations. These may be of the same *type* of error that we encountered above, but we must correct all such errors (which correspond to the iterative steps of the algorithm), so we shall describe these cases as well. Word violations correspond to Type (1) non-canonical flows (see Figure 32). Column-strict violations correspond to Type (2). As such, these errors are corrected in exactly the same manner as a μ-flow (another Type (1) error) and "i in row i" (a Type (2) error).

For example, Figure 8 shows the Littlewood-Richardson diagram with a column-strict violation fixed. The corresponding non-canonical flow and strand swap are shown in Figure 39.

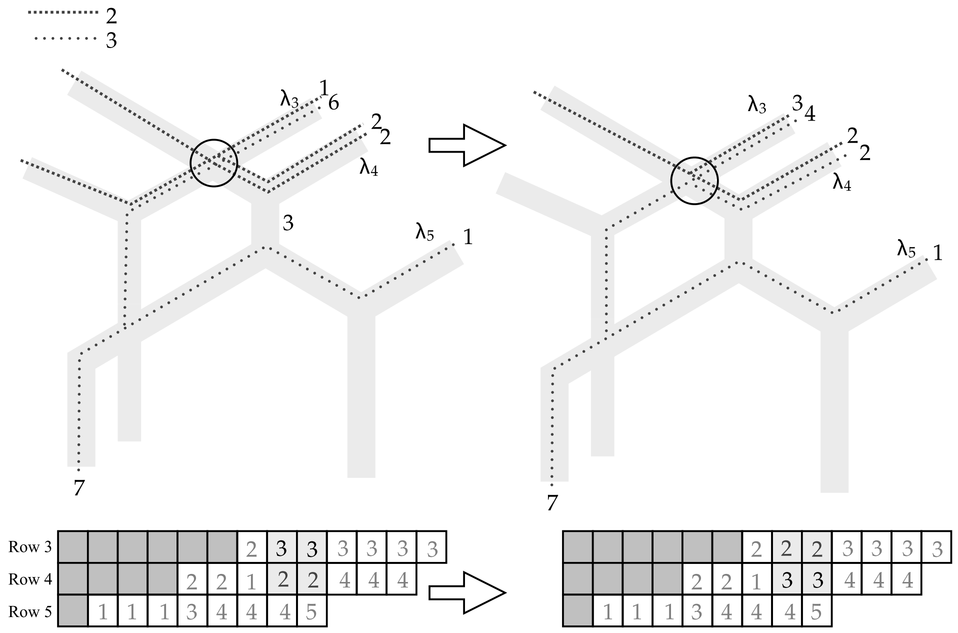


Figure 39: Column-strict flow correction, and the corresponding rows of the Littlewood-Richardson diagram.

After all non-canonical flow intersections are corrected, as described above, the resulting flow is canonical, and hence, the corresponding diagram, the one produced by the algorithm in Section 3, is indeed Littlewood-Richardson, save for any row-strict violations that may remain in the diagram. However, we may, as claimed, resolve those violations without risk of producing any new errors in the diagram since the counts of 1's, 2's, etc., in the diagram will remain unchanged, and will match the counts of the necessarily canonical flow on the right edge of the honeycomb. Hence, reordering within the row can only result in a filling matching that of the canonical flow.

Finishing our example, the corrected, canonical, flow on the honeycomb overlay is shown below. The circled vertices are those that had non-canonical flow in the original overlay of honeycombs; the flow is now canonical at each of the vertices.

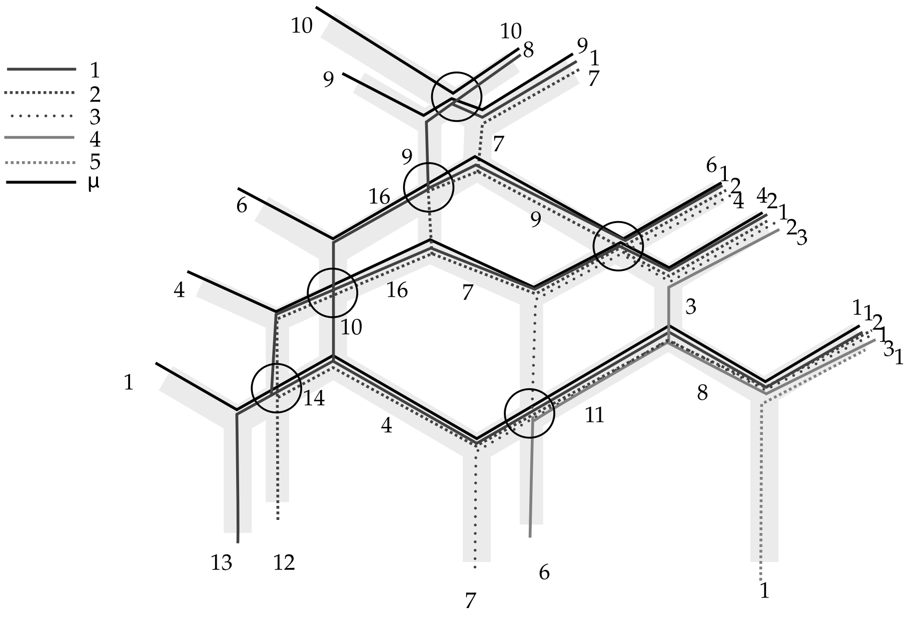


Figure 40: The corrected, canonical flow of the overlay.

This flow corresponds to the corrected Littlewood-Richardson diagram in Figure 10.

7. Conclusion

There are a number of interesting questions to which this analysis may contribute. By Theorem 6.1 the map:

$$LR(\mu, \nu; \lambda) \times LR(\mu', \nu'; \lambda') \to LR(\mu \oplus \mu', \nu \oplus \nu'; \lambda \oplus \lambda')$$

given by the algorithm described above exists, but as our examples have shown, its properties are not simple to analyze. King, Tollu, and Toumazet [9] provide conditions on μ, ν and λ under which this map is a bijection, but often it fails to be onto, and sometimes it is not one-to-one. Much work has gone into the study of the polytope of Littlewood-Richardson fillings associated to a given triple [4, 8, 9, 12, 14], and analysis of the sum of fillings map suggests that when fillings have distinct, non-trivial decompositions, it may indicate these points (Littlewood-Richardson fillings) are *vertices* of this polytope. The ability to deform a filling in more than two independent directions in this polytope is reflected in that filling having more than one decomposition. Further, our methods, based (ultimately) on the combinatorics of honeycombs, demonstrate that some problems in the geometry of Littlewood-Richardson fillings may be analyzed more easily using honeycombs, instead of hives as has more often been the case: The "strand swapping" at the heart of the algorithm is rather difficult to describe on a hive and is still difficult to work with on the dual graph, but the process becomes transparent when represented on the honeycomb.

This leads to other questions to consider. Previous work by the authors [1] showed how to determine Littlewood-Richardson fillings from matrix pairs with entries over valuation rings. It seems very likely that this map, too, would preserve overlays under simultaneous block decomposition, but this has yet to be proved. Furthermore, one can consider finding other combinatorially-defined maps corresponding to other matrix operations. For example, suppose a honeycomb is associated to a pair of $n \times n$ matrices over a discrete valuation ring R

$$(M, N) \mapsto \mathcal{H}.$$

(The process of this map is described in [1], but let us take it as given.)

Let $M^{(k)}$ and $N^{(k)}$ denote the k-th compound matrices (determined uniquely up to some ordering of the k-th exterior power of standard basis elements). Then we may also consider the map

$$(M^{(k)}, N^{(k)}) \mapsto \mathcal{H}'$$

for some honeycomb $\mathcal{H}'$. Studying examples suggests there is a combinatorial rule that assigns to the honeycomb $\mathcal{H}$ its "k-th exterior power," and that this rule may be analyzable from considering the canonical flows of the associated honeycombs.

Similarly, if

$$(M_1, N_1) \mapsto \mathcal{H}_1,$$

and

$$(M_2, N_2) \mapsto \mathcal{H}_2,$$

then we may also compute

$$(M_1 \otimes_R M_2, N_1 \otimes_R N_2) \mapsto \mathcal{H}^*,$$

and we might ask if there is a combinatorial relation allowing one to compute $\mathcal{H}^*$ from $\mathcal{H}_1$ and $\mathcal{H}_2$.

References

[1] G. Appleby and T. Whitehead, "Matrix pairs over valuation rings and $\mathbb{R}$-valued Littlewood-Richardson fillings," *Linear and Multilinear Algebra*, 61, no. 8, pp. 1063-1115, (2013). MR3175349

[2] A. Buch, "The saturation conjecture (after A. Knutson and T. Tao)," *Enseign. Math.* (2) 46, no. 1-2, pp. 43-60, (2000). MR1769536

[3] P. Burgisser, C. Ikenmeyer, "A max-flow algorithm for positivity of Littlewood-Richardson coefficients," *DMTCS Proc. AK.* pp. 265-276, (2009). MR2721518

[4] I. Dimitrov and M. Roth, "Geometric realization of PRV components and the Littlewood-Richardson cone," *Contemporary Math.* 490 (2009). MR2555971

[5] V. I. Danelov and G. A. Koshevoy, "Discrete convexity and Hermitian matrices," *Tr. Mat. Inst. Steklova* 241, pp. 68-89, (2003). MR2024044

[6] W. Fulton, "Eigenvalues, invariant factors, highest weights, and Schubert calculus," *Bull. Amer. Math. Soc.* 37, pp. 209-249, (2000). MR1754641

[7] A. Horn, "Eigenvalues of sums of Hermitian matrices," *Pacific J. Math.* 12, pp. 225-241, (1962). MR0140521

[8] C. Ikenmeyer, "Geometric complexity theory, tensor rank, and Littlewood-Richardson coefficients," PhD diss., Universität Paderborn, 2013.

[9] R. C. King, C. Tollu, and F. Toumazet, "Factorisation of Littlewood-Richardson coefficients," *J. of Combinatorial Theory, Series A* (116), no. 2, pp. 314-333, (February 2009). MR2475020

[10] A. A. Klyachko, "Stable bundles, representation theory and Hermitian operators," *Selecta Mathematica, New Series* 4.3, pp. 419-445, (1998). MR1654578

[11] A. Knutson and T. Tao, “The honeycomb model of $GL_N(\mathbb{C})$ tensor products I: Proof of the saturation conjecture,” *J. Amer. Math. Soc.* 12, pp. 1055-1090, (1999). MR1671451

[12] A. Knutson, T. Tao, and C. Woodward, “The honeycomb model of GL_n tensor products II: Facets of the Littlewood-Richardson cone,” *J. Amer. Math. Soc.* 17, no. 1, pp. 19-48, (2004). MR2015329

[13] I. G. Macdonald, *Symmetric Functions and Hall Polynomials*, Oxford Univ. Press, London/New York, (1979). MR0553598

[14] I. Pak and E. Vallejo, “Combinatorics and geometry of Littlewood-Richardson cones,” *Europ. J. Combinatorics* 26, pp. 995-1008, (2005). MR2143205

[15] B. Sagan, *The Symmetric Group: Representations, Combinatorial Algorithms, and Symmetric Functions,* Springer Verlag, (2001). MR1824028

[16] D. Speyer,“Horn’s problem, Vinnikov curves and hives,” *Duke Journal of Mathematics* 127, no. 3, pp. 395-428, (2005). MR2132865

[17] A. Zelevinsky, “Littlewood-Richardson semigroups,” *New Perspectives in Algebraic Combinatorics* (L. J. Billera, A. Björner, C. Greene, R. E. Simion, R. P. Stanley, eds.), Cambridge University Press (MSRI Publication), pp. 337-345, (1999). MR1730235

Glenn Appleby
Department of Mathematics and Computer Science
Santa Clara University
USA
E-mail address: gappleby@scu.edu

Tamsen Whitehead
Department of Mathematics and Computer Science
Santa Clara University
USA
E-mail address: tmcginley@scu.edu

Received 9 October 2013