A Combinatorial Interpretation of Minimal Transitive Factorizations into Transpositions for Permutations with two Disjoint Cycles

by

Louis-François Préville-Ratelle

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2008

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis is about minimal transitive factorizations of permutations into transpositions. We focus on finding direct combinatorial proofs for the cases where no such direct combinatorial proofs were known. We give a description of what has been done previously in the subject at the direct combinatorial level and in general. We give some new proofs for the known cases. We then present an algorithm that is a bijection between the set of elements in $\{1, ..., k\}$ dropped into n cyclically ordered boxes and some combinatorial structures involving trees attached to boxes, where these structures depend on whether $k > n, k = n$ or $k < n$. The inverse of this bijection consists of removing vertices from trees and placing them in boxes in a simple way. In particular this gives a bijection between parking functions of length n and rooted forests on n elements. Also, it turns out that this bijection allows us to give a direct combinatorial derivation of the number of minimal transitive factorizations into transpositions of the permutations that are the product of two disjoint cycles.

Acknowledgements

First and foremost, I would like to thank my advisor Ian Goulden for his support, kindness, and encouragement while I worked as a graduate student. I am very grateful for his patience, especially when spending an enormous amount of time correcting this thesis and helping me improve it. I benefited greatly from our discussions on this subject. His influence has made this a great experience for me.

I would like to thank Marg for her help, Brendan Rooney for helping me set up Latex on my computer, and Ruben Escobar for spending the time to show me how Latex and Xfig work. Also, I would like to thank Chris Godsil and David Wagner for being on the thesis committee.

Contents

List of Figures

Chapter 1

Introduction and Background

1.1 Introduction

Motivated by the geometric problem of counting distinct ramified covers of the sphere by the sphere, Hurwitz [7], in 1891, considered the following combinatorial problem in the symmetric group S_n on $\{1, \ldots, n\}$. Let $\sigma \in S_n$ be a fixed permutation with m cycles and $\tau_1, \ldots, \tau_k$ be transpositions in the symmetric group S_n. We call $(\tau_1, \ldots, \tau_k)$ a *minimal transitive factorization into transpositions* of σ if it satisfies the following conditions:

1. $\tau_1 \tau_2 \ldots \tau_k = \sigma$,

2. $k = m + n - 2$,

3. $\langle \tau_1, \ldots, \tau_k \rangle$ acts transitively on $\{1, \ldots, n\}$.

We call these factorizations minimal because the number of factors in condition 2 is the minimal possible value consistent with conditions 1 and 3. For example, with $n = 6$ and $m = 2$, we have $(1,4)(2,3)(4,5)(3,5)(3,6)(1,2) = (1,6,3)(2,5,4)$, where we multiply permutations left to right. It is straightforward to check that these 6 transpositions act transitively on $\{1,\ldots,6\}$, so this is a minimal transitive factorization into transpositions of $(1,6,3)(2,5,4)$. We say that a permutation $\sigma \in S_n$ is in the conjugacy class $(c_1,\ldots,c_m)$ when $c_1,\ldots,c_m$ specify the lengths of the disjoint cycles for σ. The formula that Hurwitz obtained, published without proof, is as follows:

Theorem 1. *(See [7]) If $\sigma \in S_n$ and σ is in the conjugacy class $(c_1, c_2, \ldots, c_m)$ then the number of minimal transitive factorizations into transpositions of σ is*

$$C_\sigma = n^{m-3}(n+m-2)! \prod_{i=1}^{m} \frac{{c_i}^{c_i}}{(c_i-1)!}$$

Hurwitz sketched how he would prove it but did not complete the proof. In 1996, Strehl [13] reconstructed the proof of Hurwitz. In 1997, Goulden and Jackson [4] published a generating function proof of this result using a partial differential equation called the join-cut equation. In 2000, Bousquet-Mélou and Schaeffer [1] generalized the problem to arbitrary factors by considering h-tuples $(\sigma_1,\ldots.\sigma_h)$, for $\sigma \in S_n$ with m cycles. We call these *minimal transitive factorizations with h factors* of σ if they satisfy the following conditions:

1. $\sigma_1\sigma_2\ldots\sigma_h = \sigma$,

2. $\sum_{i=1}^{h}(n-\ell(\sigma_i)) = n+m-2$, where $\ell(\sigma_i)$ denotes the number of cycles of σ_i,

3. $\langle\sigma_1,\ldots,\sigma_h\rangle$ acts transitively on $\{1,\ldots,n\}$.

For example, take $\sigma_1 = (2,5)(4,3), \sigma_2 = (1,3,5)(2,4), \sigma_3 = id$ and $\sigma_4 = (2,6)$, omitting fixed points, then $\sigma_1\sigma_2\sigma_3\sigma_4 = (1,3,6,2)(4,5)$. It is straightforward to check the transitivity and we have $\sum_{i=1}^{4} n - l(\sigma_i) = 2+3+0+1 = 6+2-2 = n+m-2$, so it satisfies condition 2.

Theorem 2. *(See [1]) Let $\sigma \in S_n$ in the conjugacy class $(c_1,\ldots,c_m)$. For $h \geq 0$, the number of minimal transitive factorizations with h factors of σ is*

$$D_\sigma(h) = h\,\frac{[(h-1)n-1]!}{[(h-1)n-m+2]!}\prod_{i=1}^{m}\left[c_i\binom{h\,c_i-1}{c_i}\right]$$

Bousquet-Mélou and Schaeffer's proof is a direct bijection, using combinatorial structures called constellations and Eulerian trees. They also showed that Theorem 2 implies Theorem 1 in an indirect way via an inclusion-exclusion argument.

Only in a few special cases is a direct combinatorial proof of Theorem 1 known. For the simplest case when σ is a full cycle, several direct combinatorial bijection are known [6, 8, 5, 10, 2]. One of these bijections, involving parking functions, has been extended to give a combinatorial proof for the conjugacy classes $(n-1,1), (n-2,2)$ and $(n-3,3)$ [8, 11]. In this thesis, we give a direct combinatorial proof of Theorem 1 for the case of an arbitrary

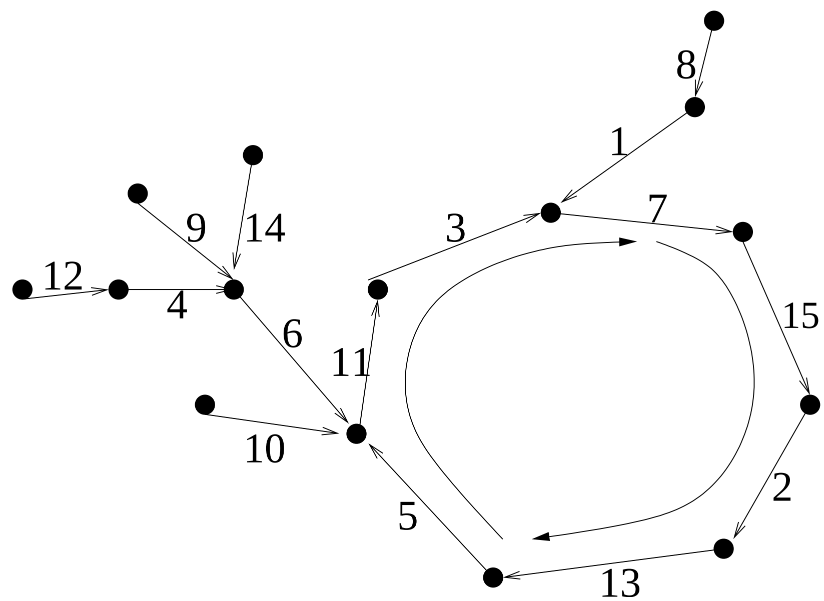

Figure 1.1: A directed graph in $A_{3,12}$.

conjugacy class with 2 cycles. The main combinatorial component of this proof is a simple bijective proof of the enumerative result given in Theorem 3 below. For positive integers θ, γ, let $A_{\theta,\gamma}$ be the set of connected directed graphs labelled on edges in the set $\{1, \ldots, \theta + \gamma\}$, with exactly one directed cycle of length at least 2 (and no other undirected cycles), θ descents along the oriented cycle (a descent is a vertex on the cycle, where the labels of the two edges adjacent to it on the cycle are in decreasing order with respect to the orientation of the cycle, and an ascent corresponds to increasing order). The descents on this cycle have total degree equal to 2. All other edges lie in rooted trees attached to ascents (of the cycle), and we direct all edges on such trees towards the root (on the cycle). See Figure 1.1 for an example of a directed graph in $A_{3,12}$. Then we get the following theorem:

Theorem 3. *For $\theta, \gamma \geq 1$, the cardinality of $A_{\theta,\gamma}$ is equal to:*

$$\gamma^{\gamma+\theta-1}$$

The proof of Theorem 3 will appear in Chapter 4 (it will be proven using similar structures). In the rest of this Chapter, we introduce some background material and give an outline of the thesis.

1.2 Background Material

1.2.1 The Symmetric Group

Let $N_n = \{1, \ldots, n\}$. The *Symmetric Group* S_n consists of all bijections from N_n to N_n. Clearly there are $n!$ of these. Each $\sigma \in S_n$ can be represented as the product of disjoint cycles such that each element in N_n appears in one of the cycles. We will assume multiplication is from left to right for S_n. We denote the length of the disjoint cycles of σ by the m-tuple $(c_1, c_2, \ldots, c_m)$, called the *conjugacy class of* σ (or cycle type), and obviously $\sum_{i=1}^{m} c_i = n$. When the c_i's are rearranged in decreasing order, this gives a *partition* of n. Let $\ell(\sigma) = m$ be the number of disjoint cycles in σ.

A *transposition* is a permutation that permutes only 2 elements, or symbolically, (i, j), where $i \neq j$, in which the transposition interchanges elements i and j of N_n (it is in the conjugacy class (2,1,...,1)).

Proposition 4. *Let $t_1, \ldots, t_k$ be transpositions in S_n. The subgroup $\langle t_1, \ldots, t_k \rangle$ acts transitively on N_n, if and only if the graph with n labelled points in N_n, with an edge between two points if there is a transposition that is composed of both endpoints, is connected.*

Proof. We will show that the fact that the graph is connected implies that the set of transpositions generates S_n which is stronger than transitivity. It is easy to see that if a set of permutations generates all transpositions then it generates S_n. Let $(i, j), i \neq j$, be a transposition. Consider a path from the point i to the point j (there must be such a path since the graph is connected). Call this path $w_1, \ldots, w_r$ where the w's are in $\{t_1, \ldots, t_k\}$. Then $(i, j) = (t_1 \cdot t_2 \cdot \cdots \cdot t_k)(t_{k-1} \cdot t_{k-2} \cdot \cdots \cdot t_1)$. The converse is trivial. □

Note that if we multiply a permutation $\sigma \in S_n$ with a transposition, say (i, j), to obtain $\sigma(i, j)$, and the values i, j are in different cycles, then these two cycles become one cycle in a specific way. Call this a *join*. Conversely, if they are in the same cycle, this cycle splits into two cycles, one containing i and the other containing j. Call this a *cut*. A join reduces the number of connected components (cycles) by 1 and a cut augments it by 1. From this, we get the following theorem:

Theorem 5. *(See [4]) Let $\sigma \in S_n$ with cycle type $(c_1, \ldots, c_m)$, then the minimal length k such that $t_1 \cdots t_k = \sigma$ and $\langle t_1, \ldots, t_k \rangle$ is transitive on N_n (so the graph is connected by the previous theorem) is equal to $n + m - 2$.*

Proof. Let $(t_1, \ldots, t_k)$ be a minimal transitive factorization of σ into transpositions. For any $i = 0, \ldots, k$, let $H(t_1, \ldots, t_i)$ be the graph with vertices in the set N_n and edges in the set N_i. Denote the edge with the value j to be the edge between the two vertices interchanged by transposition t_j. Since $H(t_1, \ldots, t_k)$ is a connected graph, construct a spanning tree by taking the edge t_r if this edge reduces the number of connected components of $H(t_1, \ldots, t_{r-1})$. This gives $(n-1)$ joins. Suppose there are x other joins and y cuts. Then since a join reduces the number of connected components by 1 and a cut augments the number of components by 1, we need to have $n - ((n-1) + x - y) = m$ since σ has m cycles. Then $x - y = m - 1$, and so $k = (n-1) + x + y \geq (n-1) + x - y \geq (n-1) + (m-1) = n + m - 2$. Now all that is left to show is that we can create such a factorization with $n + m - 2$ transpositions. Take the product $(12)(13)(14)\cdots(1n) = (1234\ldots n)$. Then simply use $(m-1)$ transpositions to cut the full cycle into a permutation θ such that $\ell(\theta) = m$ and θ has any cycle type with m disjoint cycles. By renaming, since this procedure depends only on cycle type, we can obtain σ with $(n + m - 2)$ steps. □

1.2.2 Parking Functions

We will consider all functions $f : N_n \mapsto N_n$ such that

$$| \{ j \in N_n : f(j) \leq i \} | \ \geq \ i \qquad \forall\, i \in N_n.$$

Such functions are called *parking functions* and date from 1966 [9]. The number of such functions is $(n+1)^{n-1}$. There exists more than one proof but we will only give one. We will use this theorem later to see how a simple generalization allows us to extend parking functions in a way and give some constructions from that extension. Before giving the proof, we will give an interpretation of parking functions that is well known. Suppose there are n cars and n parking spots in a parking lot (parking spots correspond to the image of the function of f), linearly arranged. One by one in increasing order, car i comes into the parking lot, going to $f(i)$ first, and parking itself in the unused parking spot with the smallest value greater or equal to $f(i)$. If no such spot exists for at least one car, then f is not a parking function. It is easy to see that each car will find a parking spot (i.e. f is a parking function) *if and only if*

$$|\{ j \in N_n : f(j) \leq i\}| \ \geq \ i \qquad \forall\, i \in N_n.$$

The following theorem is due to Pollack.

Theorem 6. *(See [12]) The number of parking functions from N_n to N_n is $(n+1)^{n-1}$.*

Proof. Suppose we add an extra parking spot with value $n+1$ and then consider all functions $f : N_n \mapsto N_{n+1}$. Put these $n+1$ parking spots into circular order, so that the last parking spot is just before the first one in this circular order. As before, cars enter the parking lot, starting at parking spot with value $f(i)$, but now they take the empty spot that happens first circularly from $f(i)$. For each such function f, we have a unique empty spot $1, \dots, n+1$. If we consider $f + r$, for any $r \in N_{n+1}$, then we see when we compare with $f + r + 1$, that each car parks in the parking spot one more modulo $(n+1)$ compared to where it parked under $f + r$. In particular the empty spot is different $\forall r \in N_{n+1}$. So we then get $\frac{(n+1)^n}{n+1} = (n+1)^{n-1}$ of these functions such that the parking spot with value $(n+1)$ is empty. Each of these clearly corresponds to a parking function. Conversely, each parking function gives rise to the parking spot with value $(n+1)$ being empty. Thus, the number of parking functions is $(n+1)^{n-1}$. □

Note that since there is only one parking function in $\{f + r : r \in N_{n+1}\}$, where $f : N_n \mapsto N_{n+1}$, one can forget about the values of the parking spots linearly and just see them as ordered circularly. This fact is important since this is precisely the simple observation that will allow an extension of parking functions in the cases that there might be more or fewer cars than parking spots. From this easy statement, some simple algorithms will be stated in Chapter 4 that will give direct bijections with well known structures.

1.3 Outline of the Thesis

In Chapter 2, we will explain further what others have done. We will also explain some combinatorial interpretations for the results that are known.

In Chapter 3, we give some simple (new) proofs for known results.

In Chapter 4, we will give the extensions of parking functions that were discussed. This will be divided into 3 cases and some examples of simple algorithms that give combinatorial interpretations for well known structures will be provided.

In Chapter 5, we give a combinatorial interpretation for minimal transitive factorizations into transpositions, in the case where $\sigma \in S_n$ is in the conjugacy class (α, β). It builds on the construction of Chapter 4. Not only will a simple argument be given to prove this in a concise way, but the steps involved prior to obtaining this simple result will be provided.

In Chapter 6, general comments and questions that could not be solved will be addressed.

Chapter 2

Some simple definitions and a known combinatorial interpretation

2.1 Introduction

We will first give a graphical interpretation of a factorization of a permutation into transpositions and start with some easy definitions that will follow for the rest of the thesis. In particular, this is valid for minimal transitive factorizations of a permutation into transpositions. It is easy to represent a factorization of a permutation into transpositions graphically where the vertices are labelled with the elements in N_n and the edges represent the transpositions such that if a transposition is the i^{th} factor, we label this edge by the value i. If we denote the factorization by F, then we will call this graphical representation the *Picture of a product of transpositions*, and we denote it by *P(F)*. For example, in Figure 2.1 we give $P(F)$ for the factorization F given by $(1,2)(2,4)(1,3)(4,5)(1,4)(1,3) = (1,5,3,4,2)$.

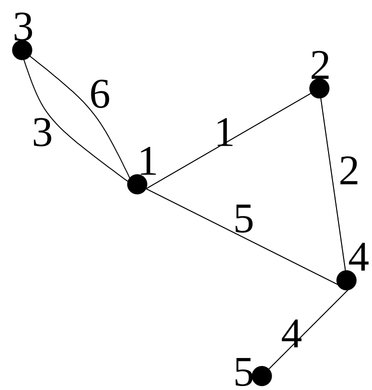

Figure 2.1: Picture of product of transpositions

Let F be a factorization as mentioned above and let $P(F)$ be the Picture of the product of the transpositions of F. For any vertex v in $P(F)$, we will call $PSIV(v)$, which denotes the *Path by the Smallest Increasing Value from vertex* v, to be the directed path (allow repeated vertices but not repeated edges) starting at vertex v with the following conditions. If v is not incident with any edge, the path is empty. Otherwise, follow the smallest edge by label from v, say e_1. Traverse this edge with smallest value and get to another vertex, call it v_1. Now, for $i = 1, 2, \ldots$, repeat the following until termination: Let S be the set of edges incident with v_i that have label larger than e_i; if $S = \emptyset$, then terminate with $t = i$, otherwise let e_{i+1} be the edge with the smallest label in S, and follow e_{i+1} from v_i to vertex v_{i+1}.

We will denote $PSIV(v)$ by $(v : e_1, e_2, \ldots, e_t)$ (since $v_1, \ldots, v_i$ are then uniquely implied). For example, in Figure 2.1, we have $PSIV(1) = (1 : 1, 2, 4)$, $PSIV(2) = (2 : 1, 3, 6)$, $PSIV(3) = (3 : 3, 5)$, $PSIV(4) = (4 : 2)$, and $PSIV(5) = (5 : 4, 5, 6)$. Note that in a product of transpositions equal to $\sigma \in S_n$, if $PSIV(v) = (v : e_1, e_2, \ldots, e_t)$ then v_t is just equal to $\sigma(v)$. We get the following simple proposition that will be useful later on.

Proposition 7. *Let $\mathcal{G}$ be a non-directed graph , $V(\mathcal{G})$ is the set of vertices in $\mathcal{G}$ and $\vec{E}(\mathcal{G})$ is the set of oriented edges of $\mathcal{G}$ (so each non-oriented edge of $\mathcal{G}$ gives rise to two oriented edges, one in each direction). Let F be a factorization of the permutation $\sigma \in S_n$ into transpositions, then*

$$\bigcup_{v \in V(P(F))} \vec{E}(PSIV(v)) = \vec{E}(P(F))$$

Proof. Take two vertices v_1 and v_2. Suppose that the paths $PSIV(v_1)$ and $PSIV(v_2)$ pass by the same edge in the same direction. By the description above, it is easy to see that the two paths will end at the same place. Then we get that $\sigma(v_1) = \sigma(v_2)$, so $v_1 = v_2$. We conclude each edge is traversed at most once in each direction.

To see that each edge is traversed exactly once in each direction, consider any edge e with a given direction. Let w be the vertex that is the tail of this directed edge. The only thing we need to do is to backtrack the procedure that was used for the $PSIV$. In order to construct a path that starts at the vertex w, follow the biggest edge smaller then e (if it does not exist, stop). Call this edge e_1 and this new vertex w_1. From w_1, look for the biggest edge smaller then e_1. Continue this process until it stops. Suppose it stops at the vertex w_x. Then it is straightforward to see that $PSIV(w_x)$ will pass through the oriented edge e as mentioned above. $\square$

Sometimes we will also refer to the concept of Picture of product of transpositions in the same way except that we will delete the labels on the vertices and just keep this information separately.

For example we will just label one or two of the vertices and say that all the labels of the other vertices could be easily obtained by following the $PSIV(v)$ iteratively for all the vertices v (starting with the label of the vertices known and continuing to label the vertices by looking at the permutation that is the product of the transpositions).

There is one gadget that is really simple that we will introduce here and will be used repeatedly for this problem of counting minimal transitive factorizations into transpositions. We will call this gadget a *tentacle*. This is simply a tree with at least 2 vertices and labelled on edges, that is rooted at one of the leaves. We will use this in the context where we will identify this root with a vertex from a different graph. More precisely, we will identify the root of a tentacle with a vertex from a cycle for many tentacles and this will help us to construct the map of Chapter 5. For example, at the top of Figure 2.2 we show three tentacles, in which the rooted leaf is circled in each case, together with a cycle of length four. At the bottom of Figure 2.2 we give an

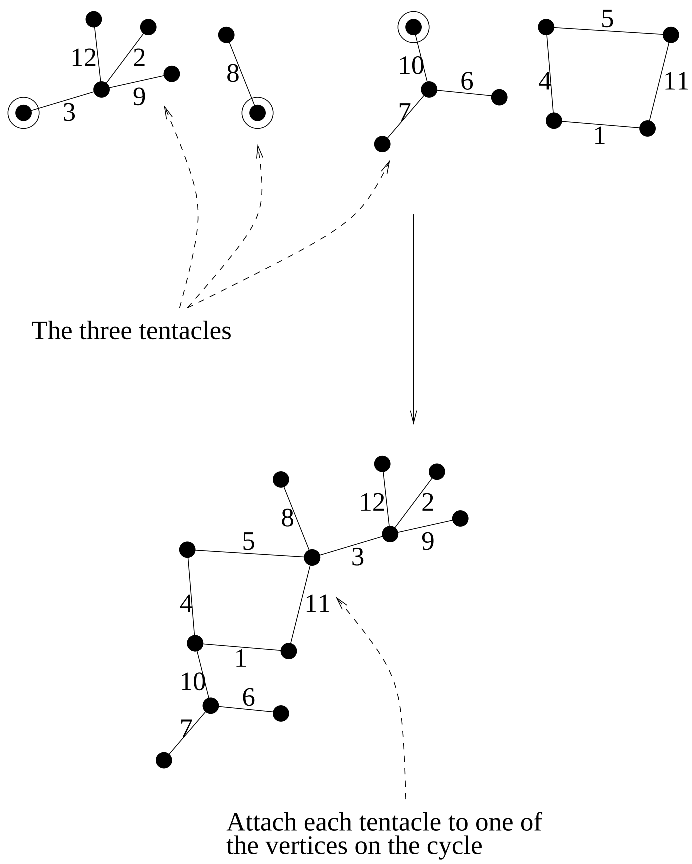

Figure 2.2: Example of tentacles that are attached to another graph

edge-labelled connected graph obtained by identifying the rooted leaves with vertices on the cycle. We have also some other questions related to tentacles in Chapter 5.

2.2 The proof of Moszkowski for the full cycle case

The following proof is similar in spirit to the one by Moszkowski [10] for the number of minimal transitive factorizations of the full cycle $(1, 2, \ldots, n)$. We present it using the notation that we have developed above. We also present this proof since it is based on the same kind of approach that we use for our mapping for minimal transitive factorization in the two cycle case in Chapter 5.

Theorem 8. *The number of minimal transitive factorizations of* $(1, 2, 3 \ldots, n)$, *or any other full cycle, is* n^{n-2}.

Proof. Let A denote the set of minimal transitive factorizations of $(1, 2, \ldots, n)$ into transpositions. Let B denote the set of rooted trees on n vertices in which the edges are labelled in N_{n-1}. If C is the set of trees with n vertices labelled in N_n, then there is a simple bijection between B and C: for a tree in C, root the tree at the vertex labelled n (removing the label), and "pull" the label on each other vertex onto the incident edge toward the root. In particular, this implies that $|B| = n^{n-2}$.

Now we give a bijection $f : A \to B$. Take an element of A and denote it by $t_1 t_2 \ldots t_{n-1}$ where the t_i's are transpositions. Since the set of transpositions $\{t_1, \ldots, t_{n-1}\}$ is transitive on N_n, the Picture of the factorization (as defined above) is connected by Theorem 5. It has $(n-1)$ edges and n

vertices, so it must be a tree. Now remove the labels of the vertices and simply root the resulting edge-labelled tree at the vertex which has label n. So define $f(t_1, t_2, \ldots, t_{n-1})$ to be this rooted edge labelled tree. Figure 2.3 shows an example of the bijection f for the factorization (in S_7) $(2,3)(4,5)(1,7)(2,4)(2,7)(6,7)$.

Now define $g : B \to A$ by labelling the root vertex n. Now follow $PSIV(n)$ until its end and label that vertex 1. Do the same for $PSIV(1)$ and denote this vertex 2. Repeat this process until all the vertices are labelled (this covers all the vertices, which can be seen from Theorem 5 with the fact that the set of transpositions is transitive and there are $(n-1)$ transpositions, so it has to be a full cycle). Just read the product of transpositions from the graph and get a factorization of $(1, 2, 3, \ldots, n)$. Clearly $f \circ g = id$ and $g \circ f = id$, so

$$|A| = |B| = n^{n-2}$$

□

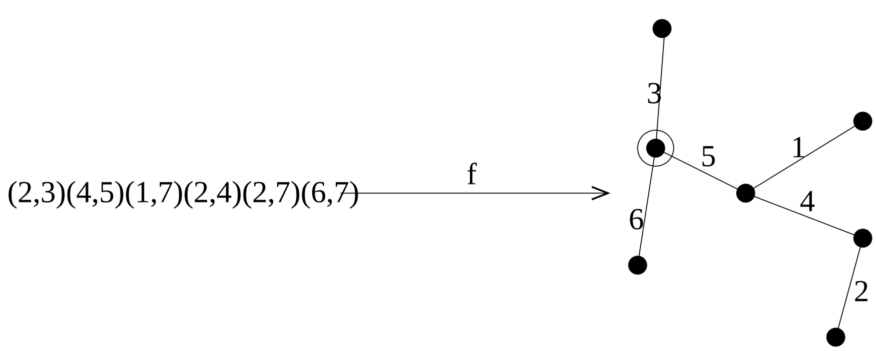

Figure 2.3: Example of the bijection f

Chapter 3

A simple proof for the $(n-1,1)$ case

In this chapter, we give a different proof from the one of Kim and Seo (See [8]) that the number of minimal transitive factorizations into transpositions of a permutation in the conjugacy class $(n-1,1)$ is equal to $(n-1)^n$. We will first introduce a basic lemma. We have been unable to find this result explicitly stated in the literature, although it is so simple that it must have been known previously.

Lemma 9. *The number of rooted labelled trees with n vertices such that the root has a smaller label than its neighbours is equal to $(n-1)^{n-1}$, for $n \geq 2$.*

Proof. Let A be the set of rooted labelled trees on n vertices such that the root has smaller label than its neighbours. Let B be the set of doubly rooted labelled trees on $(n-1)$ vertices. We will give a bijection $f : A \to B$.

Let $a \in A$ and let the vertex r be the the root of a. Let N' be the set $\{x_1, \ldots, x_{deg(r)}\}$ of neighbours of r, arranged in increasing order. Let the vertex x_t be the neighbour of r that contains vertex n among its descendants, including itself (For $n \geq 2, r \neq n$ since $deg(r) \geq 1$, and n cannot be smaller than any other label). Let $N = N' \setminus \{x_t\}$. We will define $f(a)$ by breaking a into smaller pieces and connecting them in a particular way to get $f(a)$: Delete the edge between each element of N and r. The result is a collection of subtrees; one contains r, the others each contain one element of N. Root them at the elements of N. Let S_1 be the set of these rooted trees that contain an element of N. Let R be the rooted tree that contains r (and n). In R, delete the edges adjacent to n and root the subtrees of n at its sons. Let S_2 be the set of (rooted) subtrees of n. The vertex n is deleted and will not appear in $f(a)$. Create a path that is first made of the roots of the trees in S_1 (if S_1 is not empty) in decreasing order and then followed by the vertex r. The first root of $f(a)$ is the first vertex of this path and the second root is the father of the element n in a. Now only the trees in S_2 must be placed. Add an edge from all the roots of the trees in S_2 to the vertex r. Then the doubly rooted tree $f(a)$ has been constructed. See Figure 3.1 for an example of this, in which $r = 4, n = 18$. It is easy to define an inverse function: the longest decreasing path starting at the first root contained in the path from this root to the second root finishes at the element that will become the root of the tree in the inverse map. Thus $f : A \to B$ is a bijection. The result follows since $|B| = (n-1)^{n-3}\,(n-1)^2$. $\square$

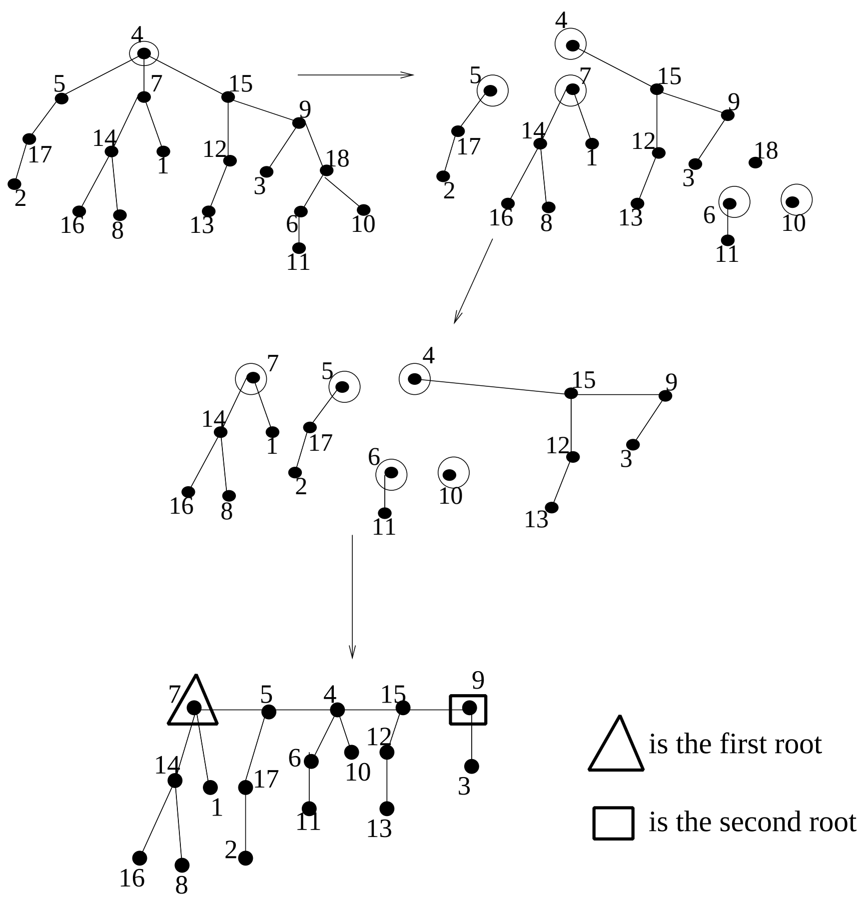

Figure 3.1: Bijection from rooted tree with root smaller than neighbours and doubly rooted trees with one less vertex.

From Lemma 9, we immediately obtain a proof of Kim and Seo's [8] result.

Theorem 10. *The number of minimal transitive factorizations of the permutation* $(1, 2, 3, \ldots, (n-1))(n)$ *into transpositions is equal to* $(n-1)^n$*.*

Proof. From Theorem 5, the number of transpositions in any factorization

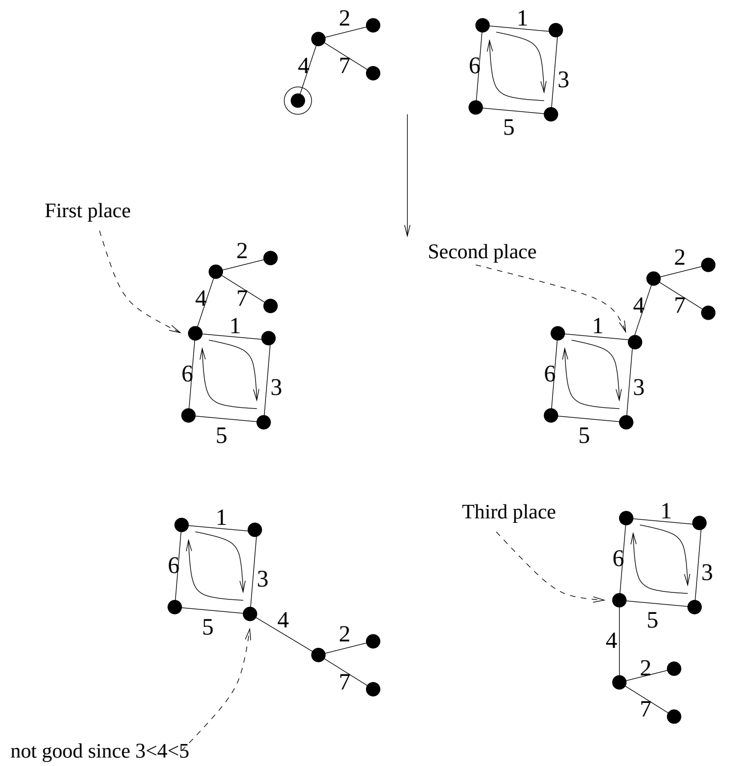

Figure 3.2: We show where a tentacle could be placed on a cycle

is n. We will use the description of the Picture of a factorization given at the beginning of Chapter 2 and the other constructions given there. So as before, if we take a look at the Picture (with only one cycle of length at least 2) of a factorization and delete all labellings of vertices, the underlying structure (edges labelled) has exactly the following characteristics (we keep in mind that we have only one cycle in the graph for the rest of this chapter): There exists a unique $v \in V$ such that $PSIV(v)$ finishes at v (n is the only fixed point in the product permutation, so v must have had label n). This is equivalent to structures where (the following is easy, but tedious, to show):

1. There exists an orientation of the cycle such that starting at one of the vertices, the entire cycle is increasing (this cycle must have length ≥ 2).

2. Take any edge incident to one of the vertices on the cycle, but not in the cycle. The label on this edge is not between ($mod\ n$) the labels on the two edges of the cycle that are incident with that vertex, following the orientation of the cycle. See Figure 3.2 for an example.

For example, in Figure 3.3, the cycle has length 5, with edge labels $2 < 5 < 8 < 9 < 12$. Edge labels 1 and 10 are not between 5 and 8, edge label 11 is not between 8 and 9 and edge label 3 is not between 9 and 12. We form a tentacle for each edge, say edge e, incident to a vertex on the cycle but not in the cycle. For each of these edges, take the vertex incident to it on the cycle. This vertex becomes the root of the tentacle with the (only) edge adjacent to it to be e (note that each vertex on the cycle might appear multiple times). Construct tentacles so that each edge not in the cycle is in exactly one tentacle (in other words take the tentacle to be as big as possible). So from condition 2 above, we see that given an increasing (oriented) cycle of length l and some tentacles, we can attach each tentacle to $l-1$ vertices on the cycle so that it is a picture of a minimal product of transpositions (the reason is that the value of the edge adjacent to the empty vertex of each tentacle will fall in the middle of two consecutive edges on the cycle $mod(n)$, following the orientation, in only one place since the cycle is increasing, so

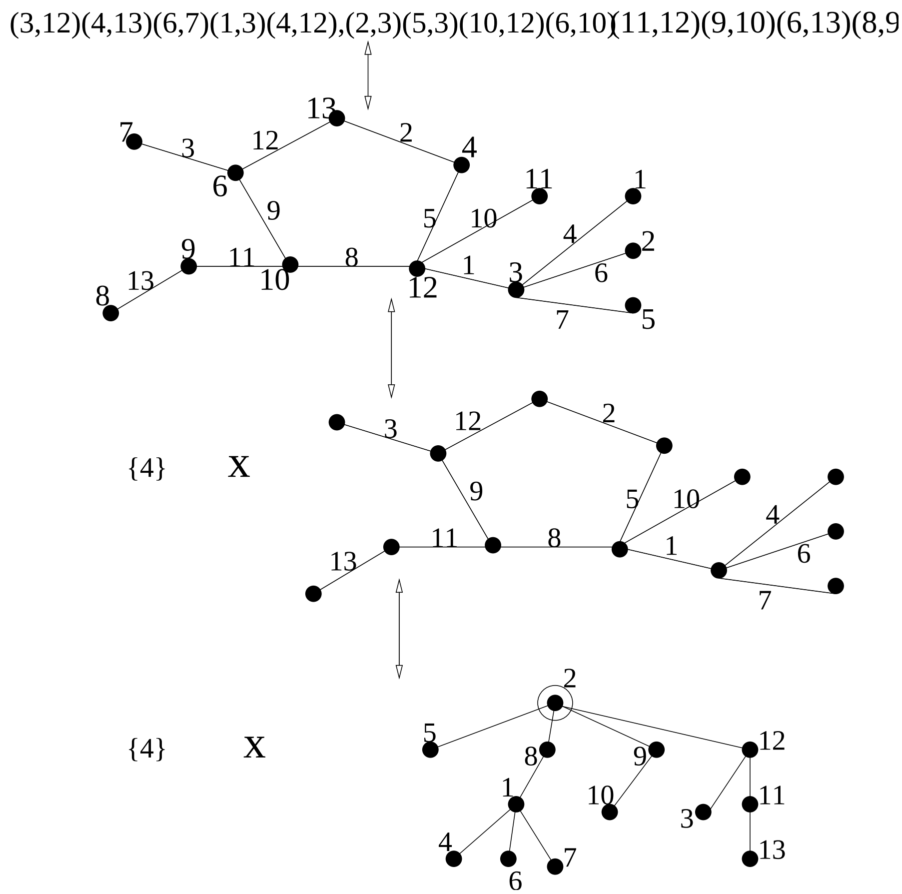

Figure 3.3: Bijection for the (n-1,1) case.

this tentacle could be attached to all vertices on the cycle except this one)

From these two characteristics, it is easy to see that the set of factorizations (minimal transitive) is in bijection with the rooted trees on n vertices such that the root has smaller index than its neighbours with a value in N_{n-1} attached to it. The bijection is as follows:

We will again start with a minimal transitive factorization, take the Picture of the factorization, break it into smaller pieces and then construct the image of this map from these pieces (except that the values on edges will become values on vertices) (see Figure 3.3). The element that is attached corresponds to the value other than n that is contained in the first transposition that contains n. From the underlying structure as described above, the root of the tree corresponds to the value of the edge that is the smallest on the cycle. Its neighbours are the other values of the edges on the cycle (note that the value on an edge becomes a vertex). Now clear the tentacles from the cycle as explained above and just remember to which vertex they were attached on the cycle. Then starting at the vertex that is mapped onto itself and following the orientation in condition 1 (the orientation that makes the cycle increasing), do the following for each tentacle. If the tentacle is attached to the i^{th} place (see Figure 3.2) where it could have been attached, then attach it to the i^{th} smallest neighbour of the new root (by pushing the values away from the empty vertex so that they become values on vertices instead). Then get a rooted tree with $n-1$ vertices such that the root has smaller index then its neighbours and a value in $\{1, 2, \ldots, (n-1)\}$ attached to it. It is easy to find an inverse function. Thus, this is a bijection.

From Lemma 10, since there are $(n-1)^{n-1}$ rooted trees on n vertices such that the root has smallest index than its neighbours and $(n-1)$ values in $\{1, 2, \ldots, (n-1)\}$, we get that the number of such factorizations is $(n-1)^n$. □

Chapter 4

An extension of Parking Functions

4.1 Some Definitions

We saw in Chapter 1 in Theorem 6 that parking functions are equivalent to dropping the values of N_n into $(n+1)$ cyclically ordered boxes. This could be obviously extended to dropping the elements of N_k into n cyclically ordered boxes. The number of ways to do this is n^{k-1}. We will construct a bijection from these constructions with other kinds of structures and this will be useful to give an interpretation of the problem of minimal transitive factorizations of permutations into transpositions where the permutations are the product of two disjoint cycles.

Dropping elements into boxes that are cyclically ordered could be done by putting the elements in each box in descending order (from left to right) and denoting the boundary between boxes with straight bars. We always read this from left to right cyclically. For example, in the following we have $k = 8$ elements dropped into $n = 5$ boxes, in which one box is empty:

$$7, 2 \big| 6, 4, 3 \big| \big| 1 \big| 8, 5 \big|$$

Thus the combinatorial objects we consider are cyclic sequences of n bars and the elements of N_k, for $k \geq 0, n \geq 1$, so that consecutive elements are descents if they are not separated by a bar. We denote this set by $D_{k,n}$ and let $D = \bigcup D_{k,n}$ where the union is over all $k \geq 0, n \geq 1$. First we need to introduce some simple facts and some notation. Suppose that we replace the elements with open parentheses and the bars with closed parentheses. Our goal is to pair up the elements with the bars by using the usual pairing of their associated parentheses. If an open parenthesis faces (is immediately followed by, cyclically) a closed one, then we say that they are *partners*. Now we erase these two partners and do the same thing recursively. At the end, we will either have that all elements and bars are paired up, some elements are not paired up but all bars are, or some bars are not paired up but all elements are. These three possibilities correspond exactly to whether the number of bars, which is the number of boxes, is equal, smaller or greater than the number of elements respectively. For example, suppose we have the following box structure with representation as parentheses below:

$$2||6,4,3||1|7,5|$$

$$()) ((())()(()$$

Remember that we read cyclically. The element 1 is partnered with the fifth bar, 2 with the first bar, 3 with the third bar, 4 with the fourth bar and 5 with the sixth bar. Also, the element 7, which corresponds to the sixth open parenthesis (linear order from the left) since it is the sixth element, is partnered with the second closed parenthesis. The element 6 has no partner.

Let x be an element in one of the box(es) that has a partnered bar. We will denote the *interval of* x by the sequence of elements and bars that starts at x and finishes at its partnered bar inclusively. Let the *strict interval of* x be the interval of x without x and its partnered bar. In the example above, the interval that starts at the element 7 is $7,5|2||$. The strict interval contained in this interval is $5|2|$. Note the convention that we use for "contained in": for an element x which has a partnered bar, we always consider the sequence to be cyclically rewritten so that x is to the left of its partnered bar. By considering the parenthesis representation, it is straightforward to obtain the following results, which we record as a Lemma for later use.

Lemma 11. *For any cyclic sequence in D, for any interval, all elements and bars in the corresponding strict interval must have a partner, and the strict interval must consist of a linearly ordered list of intervals. In particular,* *if a strict interval is non empty, then it terminates with a bar (so that the corresponding interval terminates with two consecutive bars). For any two intervals, either they are disjoint or one is contained in the other. Finally, any element or bar without a partner cannot be contained in any interval.*

Extend the set D as follows. Let D' be the set that consists of all ways to drop some (distinct) positive integers into an arbitrary (positive) number of boxes. Equivalently, these are cyclic sequences of bars and the elements of a set of distinct positive integers. Using D', we define a new set $E_{k,n}$ in 3 different ways depending if $k > n$, $k = n$ or $k < n$.

For $k > n$: The set of structures with a $d' \in D'$ such that the elements in the boxes of d' form a subset S of N_k, the bars are ascents, the elements in a box are in decreasing order and having the following characteristics. The elements of $N_k \setminus S$ are vertex labels in rooted trees such that each rooted tree is attached to one of the bars (note that an arbitrary number of rooted trees can be attached to a given bar). Also the number of boxes plus $(k - n)$ is equal to the number of elements in boxes. See Figure 4.1 (a) for an example in $E_{k,n}$ when $k = 13, n = 10$. (It is easy to see that the set $A_{\theta,\gamma}$ in Theorem 3 is equivalent to the set $E_{k,n}$ for this case, with $k = \theta + \gamma, n = \gamma$.)

For $k < n$: The set of structures with a $d' \in D'$ such that the boxes are empty, there are $(n - k)$ bars and the elements of N_k are vertex labels in rooted trees such that each rooted tree is attached to one of the bars. Figure 4.1 (b) gives an example in $E_{k,n}$ when $k = 11, n = 15$.

For $k = n$: The set of structures with a $d' \in D'$ with only one box and only one element (in N_k) in that box. The rest of the elements of N_k are vertex labels in rooted trees such that each rooted tree is attached to the bar. See Figure 4.1 (c) for an example in $E_{k,n}$ when $k = n = 8$.

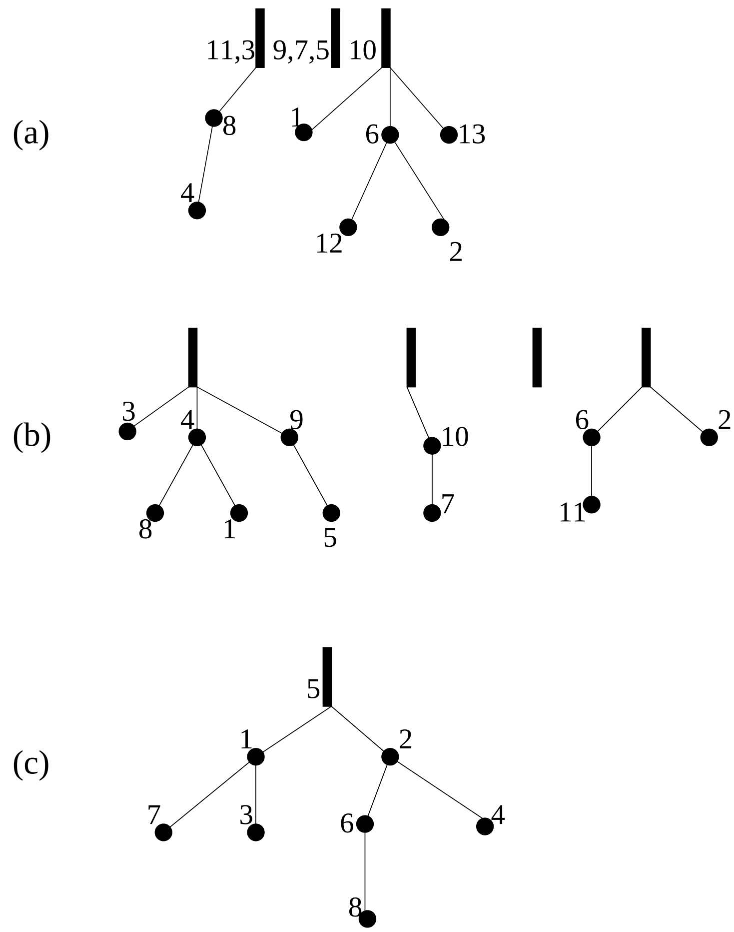

Figure 4.1: (a) For $k > n$, an element of $E_{13,10}$. (b) For $k < n$, an element of $E_{11,15}$. (c) For $k = n$, an element of $E_{8,8}$

4.2 Some Bijections

Now we will give an algorithm from $E_{k,n} \to D_{k,n}$ that turns out to be a bijection, as we prove later in this chapter.

Algorithm 1. *Repeat until there are no vertices attached to the bars (so all elements are contained in boxes). Take any bar that has some vertices attached to it. Take the vertex with the smallest label that is attached to it, clear its link to the bar, and drop its label in the box just before the bar it was attached to so that the elements are still in decreasing order in that box. Add a new bar immediately after this new element that was dropped in and attach its sons (with their subtrees), if any, to this new bar.*

See Figure 4.3 (at the end of the chapter) for an example of Algorithm 1 on an instance of $E_{k,n}$ for $k > n$. Figure 4.5 (at the end of the chapter) presents an example of Algorithm 1 on an instance of $E_{k,n}$ for $k < n$ (before the word "Reverse"). Figure 4.6 (at the end of the chapter) presents an example of Algorithm 1 on an instance of $E_{k,n}$ for $k = n$ (before the word "Reverse").

Note that, in performing Algorithm 1, in the underlying cyclic sequence at each stage, the new element is partnered with the new bar, and all previous pairings of elements with bars are maintained. Also all elements originally on trees, together with their partner bars, are inserted into the cyclic sequence to the left of the bar for that tree and to the right of all original bars. Thus this is a well defined Algorithm since what happens in each box is independent of the others, so the final result doesn't depend of the order we follow in Algorithm 1. Note that at every stage of Algorithm 1, we have a cyclic sequence in D' together with some rooted trees such that each rooted tree is attached to one of the bars.

These objects belong to the set $W_{k,n}$ for some $k \geq 0, n \geq 1$ defined by relaxing the conditions of $E_{k,n}$ in the following way: The set of structures with a $d' \in D'$ such that the elements in the boxes of d' form a subset S of N_k; the elements of $N_k \setminus S$ are vertex labels in rooted trees such that each rooted tree is attached to one of the bars. Also, the number of boxes plus the number of elements in the rooted trees is equal to n. Note the stronger statement that Algorithm 1 is a function from $W_{k,n} \to D_{k,n}$ by the same argument as above.

To prove that Algorithm 1 is a bijection from $E_{k,n} \to D_{k,n}$, we begin with an inverse algorithm and then prove that they are inverse functions, so they are both bijections. We begin with two conditions that will be useful for stating the inverse algorithm. These conditions apply to the elements of the boxes. For a member of $W_{k,n}$, any k, n, take an element, say θ in a box.

Condition 1: θ has a partnered bar, θ and this bar are consecutive (θ to the left of the bar), and the bar is either a descent or a left delimiter of an empty box.

Condition 2: If one reads to the left of θ and stops at the first bar, then this bar has no partner or its partner is smaller than or equal to the element immediately after this bar (adjacent to the right).

For example, take the box structure (where there might be some rooted trees attached to the bars, but they don't matter for the conditions):

$$7,6\big|4,3,2\big|\big|9,1\big|8,5\big|$$

The elements 6 and 2 satisfy Condition 1 and the elements $7,6,9,1,8$ and 5 satisfy Condition 2.

Lemma 12. *For any $k \geq 0, n \geq 1$, let $w \in W_{k,n}$. If there is an element in w that satisfies Condition* 1*, there must be an element that satisfies both Condition* 1 *and Condition* 2*.*

Proof. We will show first that an element that satisfies Condition 2 must exist (note that if an element in a box satisfies Condition 2, then all other elements in that box satisfy it). Then we will find using a simple construction that an element that satisfies Conditions 1 and 2 must exist.
From the fact that there is an element that satisfies Condition 1, it follows that at least one box is nonempty. If we have at least one element in one of the boxes, then we will show that there must exist an element that satisfies Condition 2. Take that element, look to its left until you get a bar. The element that immediately follows the bar (to the right of the bar) must be smaller than the partner of that bar and this partner must exist since otherwise we would have an element that satisfies Condition 2. Now we can

repeat the same process and look to the left of this new element until we get a bar and we see that if no element satisfies Condition 2 in this sequence (to the left), then the values of the elements are unbounded, which is impossible. Therefore there must exist an element that satisfies Condition 2.

Now we will show that there must exist an element that satisfies Condition 1 and Condition 2. Take one element that satifies Condition 2 and look to its right (previously we were looking left, cyclically, now we look to the right) until we see a bar. If the element that precedes this bar satisfies Condition 1, then we are done since it is in the same box as an element that satisfies Condition 2, so it must satisfy Condition 2 also. If not then the failure of Condition 1 implies that the box after the bar is not empty and the element before the bar is smaller than the element that follows the bar (to the right). But then we could look to the right of this bigger element and repeat. By iterating this process, if no element satisfies Condition 2, then we will get that all bars are ascents and so no element satisfies Condition 1, which violates the hypothesis. So there must exist an element that satisfies Condition 1 and Condition 2. $\square$

So now we can state Algorithm 2, the inverse algorithm from $D_{k,n} \rightarrow E_{k,n}$. The algorithm is stated iteratively. We will actually prove, for any $k \geq 0, n \geq 1$, that Algorithm 2 is a function from $W_{k,n} \rightarrow E_{k,n}$. In Algorithm 1 we removed elements from trees, and used the language of "dropping" them in boxes. In Algorithm 2, we remove elements from boxes and place them in trees, as labels for new vertices; here we use the language of "popping" the elements out of the box. Apply Algorithm 2 to any $w \in W_{k,n}$.

Algorithm 2. *Repeat until there are no more elements that satisfy Condition* 1*. Let* δ *be any element satisfying Condition* 1 *and Condition* 2*. We will pop out* δ *in the following way. Erase the partner bar of* δ *(that is adjacent to* δ *by Condition 1), attach* δ *to the first bar to the right of where the partner of* δ *was, and create edges from* δ *to the roots of the rooted trees that were attached to the erased bar.*

See Figure 4.4 (at the end of the Chapter) for an example of Algorithm 2 applied on an instance of $D_{k,n}$ for $k > n$. In Figure 4.5, after the word "Reverse", we perform Algorithm 2 on an instance of $D_{k,n}$ for $k < n$. Also, in Figure 4.6, after the word "Reverse", we perform Algorithm 2 on an instance of $D_{k,n}$ for $k = n$.

Note that in performing Algorithm 2, if an element is partnered with a bar, it is partnered with this bar at all stages until it gets popped out, if it does. Note that it is not clear that Algorithm 2 is a well defined function, though it is clear that at all stages we have an element of $W_{k,n}$. We will prove that it is well defined on any element on $W_{k,n}$, for any $k \geq 0, n \geq 1$.

We give some lemmas that will help us to finish the proof that Algorithm 2 is well defined.

Lemma 13. *For* $k \geq 0, n \geq 1$*, let* $w \in W_{k,n}$*, obtained at an intermediate stage of Algorithm* 2*, and* x *be one of the elements in the boxes that has a partnered bar. Suppose that the strict interval of* x *is not empty. Then* *there will always be an element in this interval that satisfies Condition* 1*. In particular, Algorithm* 2 *is never finished until all the strict intervals are empty. Morever, the strict interval of x needs to be empty before x could be popped out under Algorithm* 2*.*

Proof. Lemma 11 implies that the interval of x terminates with two consecutive bars. So reading to the left of the leftmost of these bars, we will eventually get an element (in the strict interval of x) and this element will obviously satisfy Condition 1 since it will be followed by an empty box. This is obvious since for x to satisfy Condition 1, it needs to be adjacent to its partner, which implies that its strict interval needs to be empty in order for x to be popped out. □

For $w \in W_{k,n}$, obtained at any stage of Algorithm 2, we next give a simple procedure that will tell us exactly which element will be popped out under Algorithm 2 and exactly which bar will be its father.

Take $w \in W_{k,n}$ and take an element in a box, say x, that has a partner bar in w. Look to the right of the partnered bar of x by skipping the strict intervals when there is one and stop at the first instance of the following (one of these must occur). We will call these the Possibilities:

Possibility 1: An element that has no partnered bar and that is bigger than x.

Possibility 2: A bar that has no partner.

Possibility 3: A bar that has a partner, say z, such that the strict interval of z contains the element x.

Possibility 4: An element, say θ, that is smaller than x.

Possibility 5: The element x.

Lemma 14. *For any $k \geq 0, n \geq 1$, when performing Algorithm* 2 *and obtaining $w \in W_{k,n}$ at an intermediate stage, we get the following for any instance of Algorithm* 2. *If Possibility* 1 *or Possiblity* 5 *happens first, then x is never popped out under Algorithm* 2. *If Possibility* 2 *happens first, x will be a son of that bar with no partner. If Possibility* 3 *happens first, then x will be the son of the partner of z. If Possibility* 4 *happens first, x will be a son of the bar that is the closest to the right of θ.*

Proof. For Possibilities 1 and 5, this is easy to show since the element x will never satisfy Condition 1 (the element that is adjacent on the right to the partnered bar of x will always be greater than or equal to x).

For Possibility 2, we note that since we have a bar that has no partner, we have more boxes than elements and therefore we will always have an empty box, so then all elements will be popped out at the end of Algorithm 2 (since there will always remain an element that satifies Condition 1 otherwise). So we just need to show that x can't be attached to any bar before the one mentioned in Possibility 4. Again, it is not hard to show that the sequence starting after the partnered bar of x to the bar mentioned in Possibility 4 will be a well defined bracketing sequence (linearly), so made of some consecutive intervals. Since it didn't satisfy the other possibilities before, we know all the elements not in the strict intervals of these intervals will be bigger than x, so this tells us that x will not be able to satisfy Condition 1 unless all these intervals are popped out. So x must be attached to the bar mentioned in Possibility 2 (the bar that has no partner).

For Possibility 3, x is contained in the strict interval of z, then x must be popped out by Lemma 13 during Algorithm 2. The same reasoning as in the previous paragraph could be applied to prove that the element x can only become the son of the partner of z. So x will always become the son of the partner of z under Algorithm 2.

For Possibility 4, then we will first show that θ can't be popped out unless x is popped out. We will then show that when Algorithm 2 is finished, x must be popped out. We will then show that the only bar that x could be the son of is the bar mentioned in Possibility 4. Let's say this bar in Possibility 4 has partner y. We will first give a picture of the box structure of w for what matters here to us:

$$\overbrace{x\, S_x \,|}\ \overbrace{r_1\, S_{r_1} \,|}\ \overbrace{r_2\, S_{r_2} \,|}\ \ldots\ |\overbrace{r_j\, S_{r_j} \,|}\ \theta..y|$$

Here each $\overbrace{r_i\, S_{r_i} \,|}$ is the interval of r_i and all the intervals of $r's$ are consecutive (where S_q is the strict interval of the element q). The sequence $\theta..y$ is made only of elements (no bars). So $\theta \geq y$ and by above $x > \theta$. Since the first element of each of these intervals is bigger than x which is bigger than θ, then y can't be popped out unless x is popped out since all elements in the sequence from the value x to the bar before θ need to be popped out so that y could satisfy Condition 2. So y could be popped out only after x is popped out. Now to show that the only bar that x could be the son of is the partner of y, we just have to show that x can't be popped out unless all the bars between x and y have been popped out. But this is easy since all the consecutive intervals from x to θ have a first element that is bigger than x, so x can only satisfy Condition 1 when all these bars have been popped out. So the only place that x could be attached when popped out is the partnered bar of y (also since y can't be popped out before x). Now we show that if the element x is not popped out after performing Algorithm 2 on w, then there will be an element satisfying Condition 1. The sequence from θ to the element y will still be in boxes if x is not popped out by above, so just take the bar that is adjacent to θ at the left and this bar needs to have a partner since the sequence between the partnered bar of x and the element θ is a well defined bracketing sequence linearly (made of some consecutive intervals) initially, so it will always remain a well defined bracketing sequence as Algorithm 2 is performed (by Lemma 11). Therefore the bar before the element θ has a partner (after Algorithm 2 is performed) and we know by the picture above and by Lemma 13 that this element is one of the $r's$ or x, which are bigger than θ. So we will have that this bar has an adjacent partner (to the left) that satisfies Condition 1 or the strict interval of the partner of this bar is not empty, in which case Lemma 13 guarantees to us that there is an element that satisfies Condition 1. So this shows that at the end of Algorithm 2, x will be popped out always. So by above, x is always popped out and is always attached to the same bar.

□

Proposition 15. *For any* $k \geq 0, n \geq 1$, *Algorithm* 2 *is a well defined function from* $W_{k,n} \rightarrow E_{k,n}$. *In particular it is a well defined function from* $D_{k,n} \rightarrow E_{k,n}$.

Proof. Obvious from the above lemma. □

Now we still need to show that Algorithm 1 and Algorithm 2 are inverse functions in the special cases that are of interest to us. We will start by giving some simple lemmas that will be useful.

Lemma 16. *If Algorithm* 1 *is performed on an element of* $E_{k,n}$, *then when an element, say* θ, *is dropped in a box, the first element, if any, that is dropped in the box delimited to the left by the partner of* θ *is bigger than* θ.

Proof. This is true since if an element, say γ, is dropped first in the box after the partner of θ then that means that θ and γ were attached to the same bar and that θ was chosen before γ, so $\theta < \gamma$. □

Lemma 17. *If Algorithm* 1 *is performed on an element of* $E_{k,n}$, *then the element that is adjacent to a given bar at the right is weakly increasing as Algorithm* 1 *is performed.*

Proof. This is obvious since when an element is dropped in a box , it maintains the decreasing order in the box and the bar is put at the right of the element dropped. □

Lemma 18. *While Algorithm* 1 *is performed on an element of* $E_{k,n}$*, if an element, say* θ*, is followed by a bar and a smaller element, say* γ*, or an empty box, then* θ *has been dropped at an earlier stage of Algorithm* 1*.*

Proof. If θ had not been dropped earlier, then there should have been an element bigger or equal to θ in the box that follows θ before Algorithm 1 is performed. But then by Lemma 17, the element adjacent to the right of the partner of θ would still be bigger or equal to θ, contradicting the hypothesis. ☐

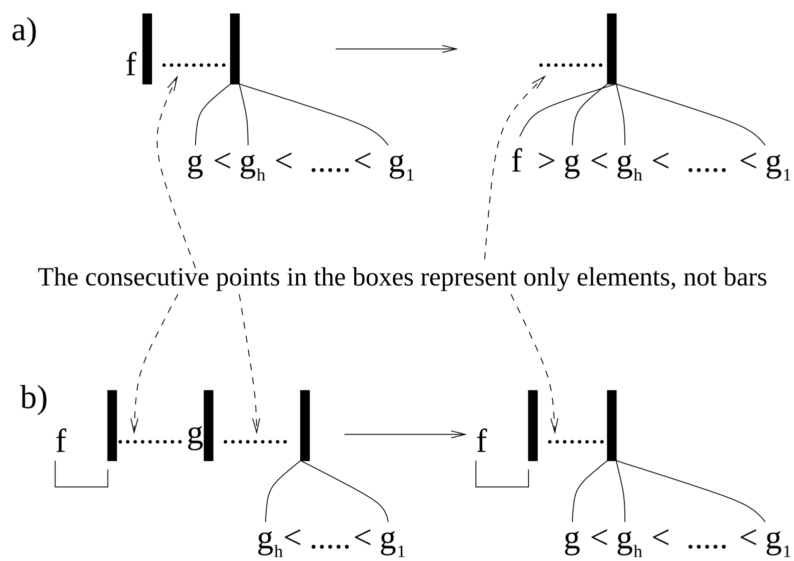

Figure 4.2: In (a) f is dropped in. In (b) g is dropped in.

Lemma 19. *For any* $k \geq 0, n \geq 1$*, if Algorithm* 2 *is performed on any element of* $D_{k,n}$*, then when an element is popped out, it is the smallest element that is attached to the bar where it gets attached.*

Proof. Suppose that such a statement is wrong and that this is the first occurence of a counterexample, so that an element, say f, is not the smallest (compared to the other elements that are attached to the same bar as f gets attached). So say elements $g_1, \ldots, g_h, g$ are in decreasing order and so the last one that was popped out is g (so $f > g$). Figure 4.2 (a) shows on the left the stage just before f is popped out and on the right, the stage just after (f is popped out). We know that g was popped out at an earlier stage than f was popped out. Figure 4.2 (b) shows on the left the stage just before g was popped out and on the right, the stage just after. Note that in Figure (b), it is easy to see that there are no bars between the partner of f and the element g (since otherwise there should have been another element that would have been popped out at a stage between when g is popped out and f is popped out and attached to the same bar as the one that g is attached to). So by Figure 4.2 (b) on the left, we see that there must be an element in the same box as g, say θ, that is bigger than f and so that g satisfies Condition 2. But then θ must be in the box after f in Figure 4.2 (a), preventing f from being popped out at that stage. This is a contradiction and the result follows. □

Proposition 20. *Each step of Algorithm* 1 *when applied to an element of* $E_{k,n}$ *can be reversed by Algorithm* 2*, and each step of Algorithm* 2 *when applied to an element of* $D_{k,n}$ *can be reversed by Algorithm* 1*.*

Proof. Suppose we get the following iteration of Algorithm 1, where γ has just been dropped in (we use this overbrace relation for the partnered relation):

$$\overbrace{\theta\, S_\theta}\big| r \ldots \gamma \big| \ldots \big| \quad or \quad \overbrace{\theta\, S_\theta}\big| \gamma \big| r \ldots \big|$$

Now suppose γ cannot be popped out under Algorithm 2, then we get $\theta > r, \gamma$ (r might not exist). This means that θ was dropped in previously by Lemma 18. But then this contradicts Lemma 17 and Lemma 16.

For the other direction, this is straightforward from Lemma 19.

□

From the propositions above, we get the following theorem.

Theorem 21. *Algorithm* 1 *is a bijection from* $E_{k,n} \to D_{k,n}$ *and Algorithm* 2 *is its inverse.*

Corollary 1.

$$\big|E_{k,n}\big| = n^{k-1}$$

For $k > n$, Figure 4.4 is the inverse (Algorithm 2) of Figure 4.3 that performs Algorithm 1. For $k < n$, Figure 4.5 performs first Algorithm 1 and then Algorithm 2. For $k = n$, Figure 4.6 performs Algorithm 1 and then Algorithm 2. We see that in the three cases, we always come back to the same original element in $E_{k,n}$.

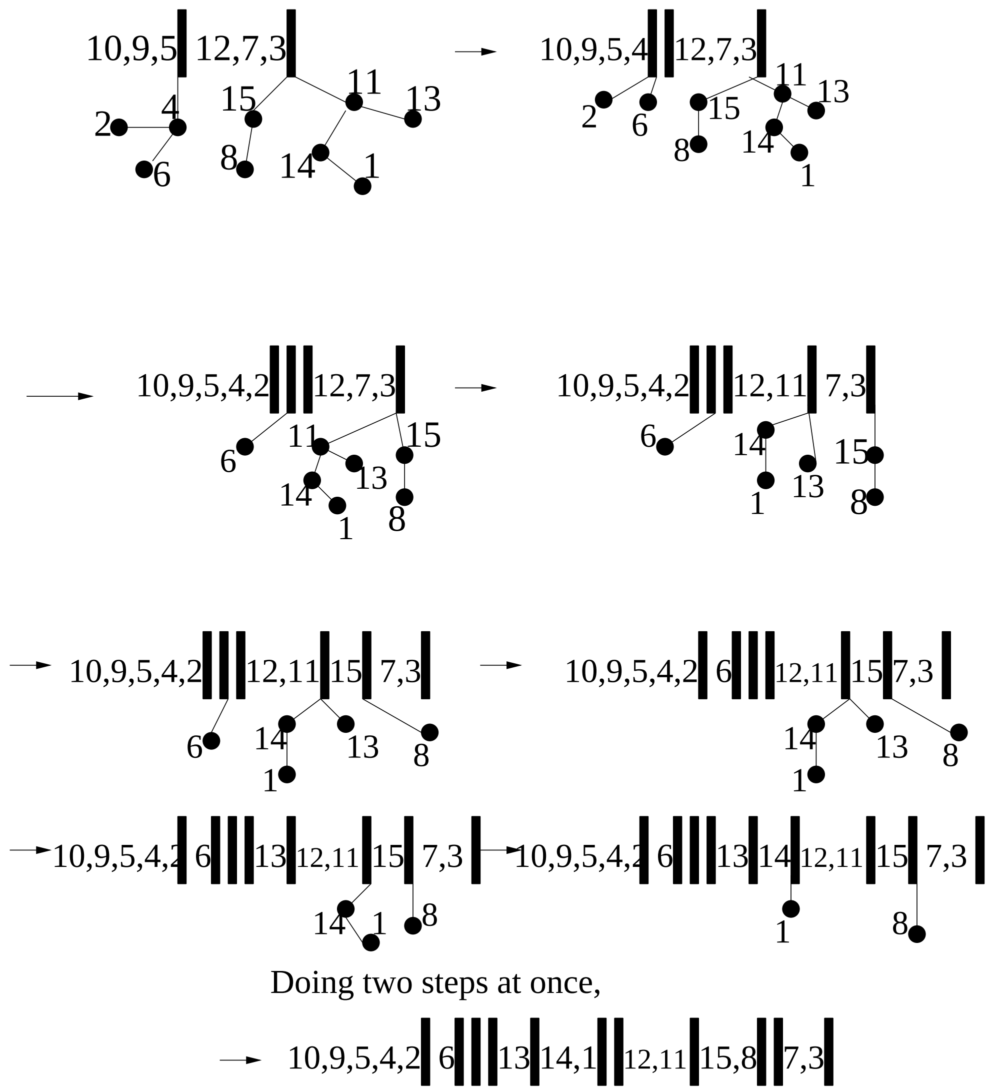

Figure 4.3: When $k = 15 > 11 = n$, an example of Algorithm 1.

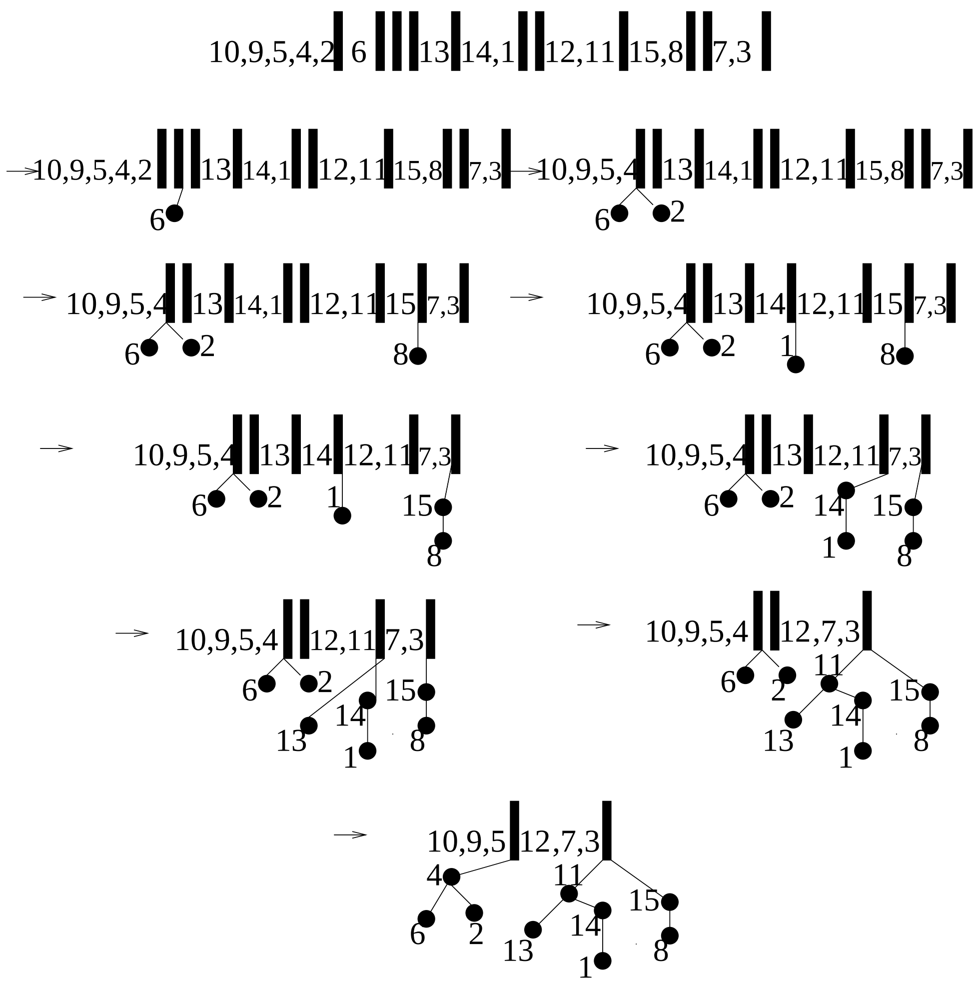

Figure 4.4: When $k = 15 > 11 = n$, Algorithm 2 is applied to the last part of the previous Figure.

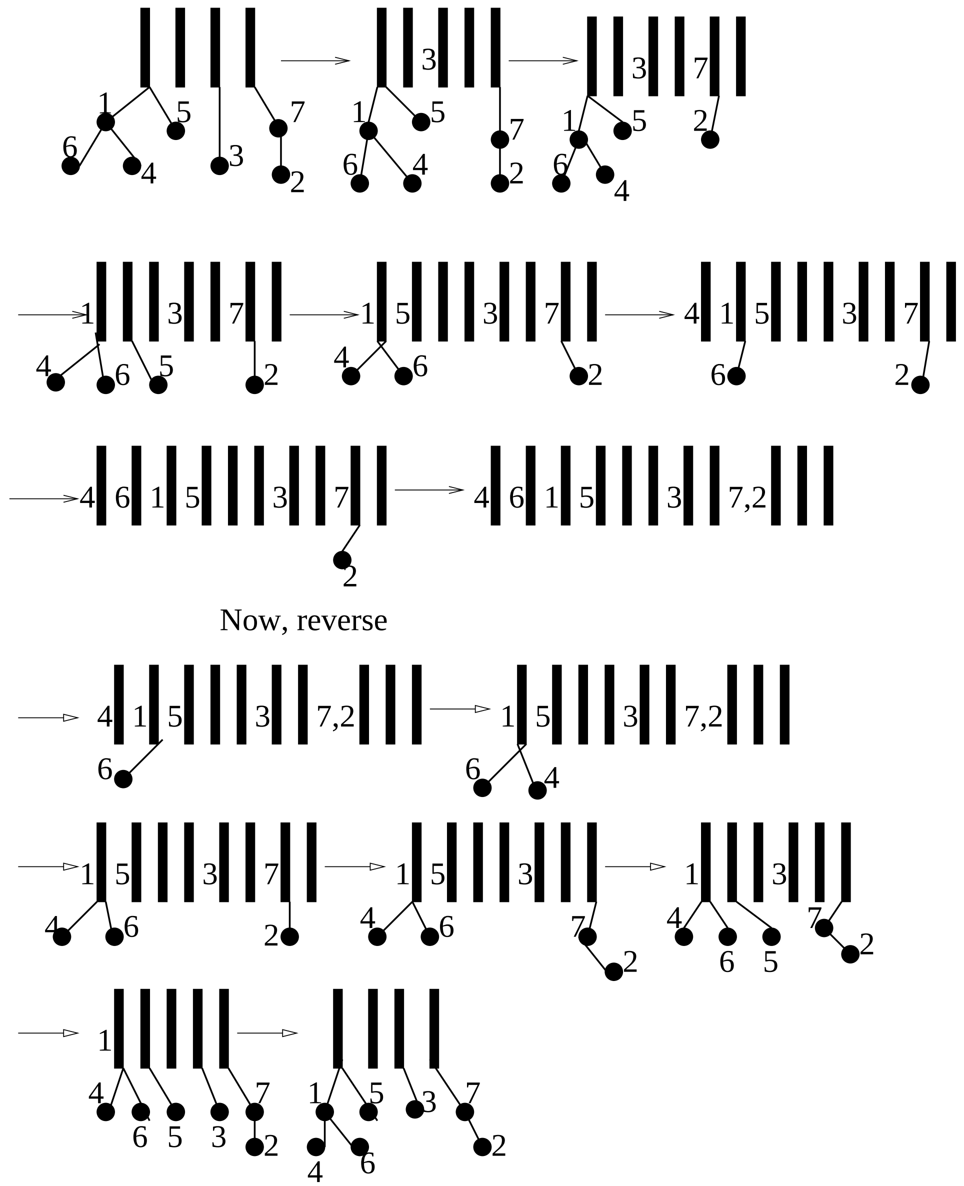

Figure 4.5: When $k = 7 < 11 = n$, Algorithm 1 followed by Algorithm 2

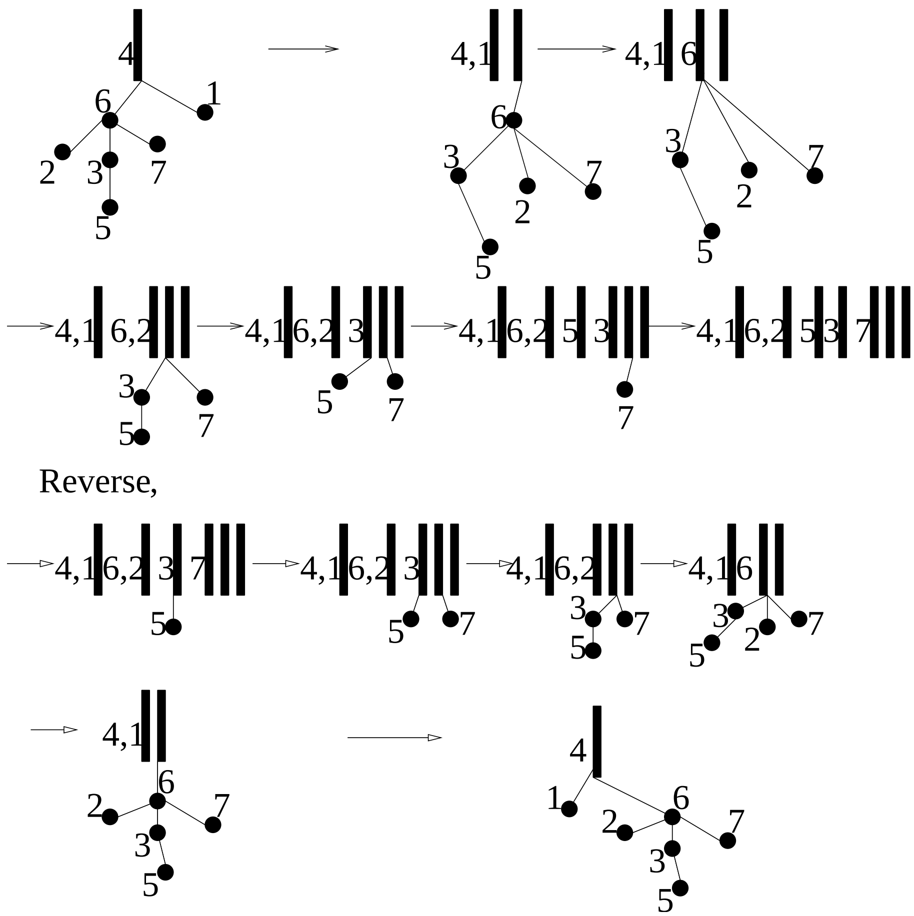

Figure 4.6: When $k = 7 = n$, Algorithm 1 followed by Algorithm 2

Chapter 5

The (α, β) case

In this chapter we give a direct, but artificial, combinatorial interpretation of the number of minimal transitive factorizations of permutations that are the product of two disjoint cycles. It is artificial since it is not symmetrical and a bit obscure. In the first section, we include the calculations because we hope that a similar reasoning for the more general case (where $\sigma \in S_n$ is arbitrary) might help. In the second section, we give the actual mapping.

5.1 Calculations and Details of Method

5.1.1 Counting

We consider minimal transitive factorizations into transpositions for a permutation with two disjoint cycles. One of these cycles has length α and the other β, with $\alpha + \beta = n$. Let, for example, the first cycle be $(1, 2, 3, \ldots, \alpha)$ and the second $((\alpha+1),\ldots,n)$, so consider the minimal transitive factorizations of the permutation $\mu = (1,2,3,\ldots,\alpha)((\alpha+1),\ldots,n)$, that is in the conjugacy class (α,β). Let $F(\alpha,\beta)$ be the number of minimal transitive factorizations of μ into transpositions (or any other permutation in the conjugacy class (α,β)). From the formula in Theorem 1, we know $F(\alpha,\beta) = \alpha^{\alpha}\beta^{\beta+1}\binom{n-1}{\alpha-1}$. We start by giving some easy statements. For the Picture of each factorization we have n edges, n vertices and a graph that is connected by Theorem 5. This implies we have only one cycle in the graph, which we will refer to as the *cycle of the factorization*. Define

$$PSIV(\alpha{-}cycle) := \bigcup_{v\in\alpha\text{–}cycle} PSIV(v),$$

$$PSIV(\beta{-}cycle) := \bigcup_{v\in\beta\text{–}cycle} PSIV(v).$$

In Figure 5.1, we start with a factorization of $\mu = (1,...,7)(8,..,15)$ at the top of the figure, and immediately below that we give its Picture of the factorization (with labels on vertices). Then in (a), we give the $PSIV(\alpha{-}cycle)$. In (b), we give the $PSIV(\beta{-}cycle)$.

(Note that the Picture has one cycle, described above, but the permutation μ has two cycles.) We give some simple lemmas that concern the picture of minimal transitive factorizations into transpositions of μ, without proofs. They are simple but tedious to prove. The argument that helps to establish them is Proposition 7.

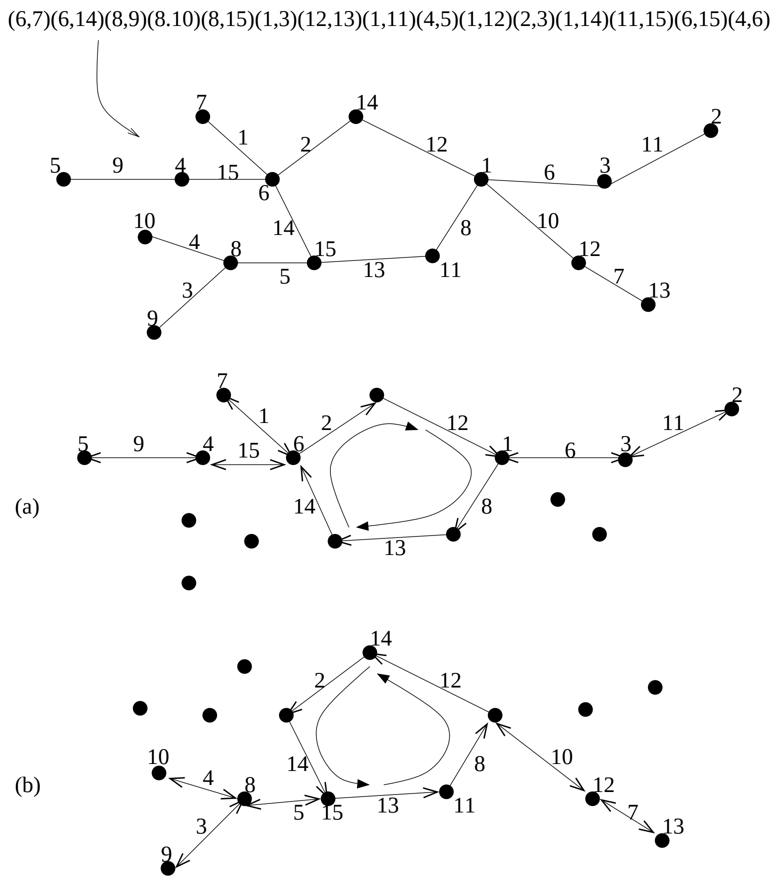

Figure 5.1: Example of (a) $PSIV(\alpha{-}cycle)$ and (b) $PSIV(\beta{-}cycle)$ for a minimal transitive factorization of μ for $\alpha = 7$ and $\beta = 8$

Lemma 22. *$PSIV(\alpha{-}cycle)$ traverses the cycle of the factorization in one direction and denote that direction to be the α-direction. $PSIV(\beta{-}cycle)$ traverses it in the other direction and denote it by the β-direction.*

For example, in 5.1(a), the α-direction in the cycle of the factorization is clockwise, as shown. In (b), we do the same thing for the β-cycle and show the β-direction (opposite to the α-direction). All the remaining lemmas apply for the β-cycle also.

Lemma 23. *The vertices on the cycle of the factorization that belong to the α-cycle of μ correspond to the vertices that are descents in the α-direction.*

For any Picture of a factorization (of a minimal transitive factorization into transpositions) of μ, it is made of a cycle and some tentacles where the root of each tentacle is identified with a vertex on the cycle. Figure 5.2 shows the tentacles split from the cycle of the factorization at the top of Figure 5.1 (where vertex labels have been removed).

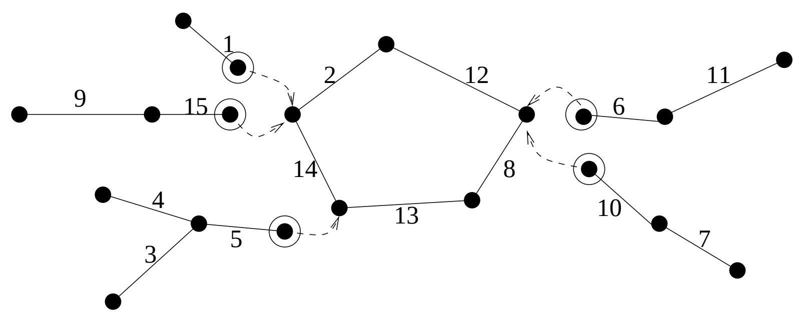

Figure 5.2: We split the tentacles from the cycle of the factorization from the top of Figure 5.1

Lemma 24. *For each of these tentacles, all of its vertices except the root are contained in the α-cycle or in the β-cycle (follows from Proposition 7). So we can call each tentacle whose vertices belongs to the α-cycle, an α-tentacle, or for the β-cycle, a β-tentacle.*

Figure 5.3 shows the α-tentacles in part (a) and the β-tentacles in part (b) of the factorization at the top of Figure 5.1.

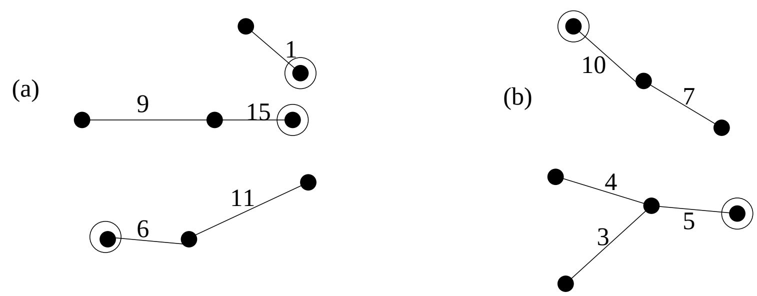

Figure 5.3: In (a), the α-tentacles of Figure 5.1, the β-tentacles of Figure 5.1

Lemma 25. *A tentacle is an α-tentacle* if and only if *the (only) edge adjacent to the root falls between the two consecutive edges on the cycle adjacent to the vertex that the tentacle is identified with following the α-direction.*

All the above lemmas are necessary conditions for the Picture of a factorization. Conversely, we see that if we build a Picture of a factorization (without labels on vertices) by first creating a cycle and then adding tentacles so that it gives rise to one cycle of length α and another of length β, then we just need to fix the values of one of the vertices for each cycle to give rise to a minimal transitive factorization of μ. In particular from Lemma 25, we get the following result.

Lemma 26. *Suppose we have a cycle of length at least* 2*, with distinct labels on edges, and a specified α-direction with d descents in the α-direction. Then there are d vertices on the cycle where for any tentacle its root can be identified so that it becomes an α-tentacle (all edge labels are distinct). In particular, these d vertices are independent of the fact that other tentacles might be identified (by their root) with vertices on the cycle.*

If you fix the number of edges on the cycle of the factorization and d, and decide which edges belong to the cycle of the factorization, the α-tentacles and the β-tentacles, the counting of the Pictures of the factorization with these characteristics is independent of the allocation of edges. To help find a combinatorial interpretation, I will first give an enumeration and then transform it into an interpretation. Everything that has been said about the α-cycle can be said about the β-cycle (with respect to the β-direction on the cycle).

Using all the details of above, we now give the enumeration. We are going to calculate $F(\alpha, \beta)$ directly (where as mentioned above, $F(\alpha, \beta)$ is the number of minimal transitive factorizations into transpositions for any permutation in the conjugacy class (α, β)). So the enumeration is as follows:

Lemma 27. *For $\alpha, \beta \geq 1$, with $\alpha + \beta = n$, we have*

$$F(\alpha, \beta) = \alpha\beta \sum_{i=2}^{n} \sum_{j=1}^{i-1} (L_{i,j}\, T_{|\alpha|,j}\, T_{|\beta|,i-j}) \binom{n}{i, \alpha - j, \beta - (i - j)} j(i - j),$$

where $L_{i,j}$ is the number of directed cycles of i distinct elements with j descents, $T_{x,x-y}$ (where $x > y \geq 0$) is the number of ways to put y labelled edges in $(x - y)$ cyclically ordered rooted trees and $\binom{n}{i,\alpha-j,\beta-(i-j)}$ is the multinomial coefficient.

Proof. The following proof make use of Lemmas 22, 23, 24, 25 and 26. In the outer summation, i represents the size of the cycle in the factorization, and in the inner summation, j is the number of descents for the $\alpha - cycle$ (so in the direction that the $\alpha - cycle$ traverses the cycle of the factorization). The external factors α and β correspond to the fact that these graphs have no values on vertices, so by fixing any vertex of each cycle, we fix all others. The term $T_{x,x-y}$ is precisely $E_{k,n}$, as in Chapter 4, with $n = x > k = y$. So we have $T_{x,x-y} = x^{y-1}$ from Corollary 1. The factor j corresponds to linearizing the orientation of the bars for $T_{|\alpha|,j}$ (in other words, breaking the cyclic order to obtain a total order). The factor $(i-j)$ corresponds to linearizing the orientation of the bars of $T_{|\beta|,i-j}$. In the term $\binom{n}{i,\alpha-j,\beta-(i-j)}$, i represents the edges in the cycle of the factorization, $(\alpha - j)$ the edges that belong to the α-tentacles (for the α-cycle) and $\beta - (i-j)$ the edges that belong to the β-tentacles (for the β-cycle). □

We now evaluate the triple summation in Lemma 27. For $x, y \geq 0$, $\binom{x}{y}$ denotes the binomial coefficient. We get for $L_{i,j}$ (from [3], p.495):

$$L_{i,j} = \sum_{t=1}^{i-j} t^{i-1}(-1)^{i+j+t}\binom{i}{t+j}, \tag{5.1}$$

and so $F(\alpha, \beta)$

$$\begin{aligned} &= \sum_{i=2}^{n}\sum_{j=1}^{i-1}\left(\sum_{t=1}^{i-j} t^{i-1}(-1)^{i+j+t}\binom{i}{t+j}\right) j(i-j)\alpha^{\alpha-j}\beta^{\beta-(i-j)}\binom{n}{i,\alpha-j,\beta-(i-j)} \\ &= \sum_{j=1}^{\alpha}\sum_{t}\left(\sum_{i=t+j}^{j+\beta} \beta^{\beta-(i-j)}(-t)^{i-1}(i-j)\binom{\beta-t}{i-j-t}\right) j\alpha^{\alpha-j}\frac{n!(-1)^{i+j+1}}{(t+j)!(\alpha-j)!(\beta-t)!} \end{aligned}$$

where the inner sum is equal to, when $t = \beta$:

$$\beta^{\beta-(\beta)}(-\beta)^{i-1}\beta \cdot 1 = -(-\beta)^i$$

and when $t \neq \beta$:

$$\begin{aligned}
&= \sum_{i=t+j}^{j+\beta} \beta^{\beta-(i-j)}(-t)^{i-1}(i-j-t)\binom{\beta-t}{i-j-t} - \sum_{i=t+j}^{j+\beta} \beta^{\beta-(i-j)}(-t)^{i}\binom{\beta-t}{i-j-t} \\
&= (\beta-t)(\beta-t)^{\beta-t-1}(-t)^{j+t} - (\beta-t)^{\beta-t}(-t)^{j+t} \\
&= 0
\end{aligned}$$

Thus we have

$$\begin{aligned}
F(\alpha,\beta) &= \sum_{j=1}^{\alpha} -(-\beta)^{j+\beta}(-1)^{j+\beta+1} \cdot j\alpha^{\alpha-j}\frac{n!}{(\beta+j)!(\alpha-j)!} \\
&= \sum_{j=1}^{\alpha} j\alpha^{\alpha-j}\binom{n}{\beta+j}\beta^{j+\beta} \\
&= (n-1)!\sum_{j=1}^{\alpha}(nj)\frac{\alpha^{\alpha-j}}{(\alpha-j)!}\frac{\beta^{j+\beta}}{(j+\beta)!} \qquad (5.2)
\end{aligned}$$

with $nj = \alpha(\beta + j) - (\alpha - j)\beta$,

$$
\begin{aligned}
&= (n-1)! \left(\sum_{j=1}^{\alpha} \frac{\alpha^{\alpha-j+1}}{(\alpha-j)!} \frac{\beta^{\beta+j}}{(\beta+j-1)!} - \sum_{j=1}^{\alpha-1} \frac{\alpha^{\alpha-j}}{(\alpha-j-1)!} \frac{\beta^{\beta+j+1}}{(\beta+j)!} \right) \\
&= (n-1)!(\alpha\beta) \left(\sum_{j=1}^{\alpha} \frac{\alpha^{\alpha-j}}{(\alpha-j)!} \frac{\beta^{\beta+j+1}}{(\beta+j-1)!} - \sum_{j=1}^{\alpha-1} \frac{\alpha^{\alpha-j-1}}{(\alpha+j-1)!} \frac{\beta^{\beta+j}}{(\beta+j)!} \right) \\
&= (n-1)!(\alpha\beta) \frac{\alpha^{\alpha-1}}{(\alpha-1)!} \frac{\beta^{\beta}}{\beta!} \\
&= \alpha^{\alpha} \beta^{\beta+1} \binom{n-1}{\alpha-1},
\end{aligned}
$$

which is consistent with Theorem 1 when $m = 2$.

5.1.2 Some Interpretations

We will sketch the main steps that lead us to the combinatorial interpretation.

The alternating formula (5.1) for $L_{i,j}$ follows simply from the principle of inclusion-exclusion (see [3], p.495). We get an involution τ from it.

The rest of this section is really sketchy. The involution τ is basically applied on sets of boxes with elements in them. We can extend the involution by adjoining rooted forests where each rooted tree is attached to boxes. The fact that the triple sum collapses to a single sum in the development after Lemma 27 can be used to give another involution, say ϕ, on the same set as the (extended) τ such that $(fix\,\tau) \neq (fix\,\phi)$. By alternating these two involutions, we can give a bijection from the set of factorizations and the quantity in (5.2), just before we do the telescope. Then we just have to give a combinatorial interpretation of the simple telescope and by combining these two bijections, we are done.

The fact that the triple sum becomes a simple sum was the way we guessed Theorem 21 for $k > n$. The inclusion-exclusion helped us to find the Algorithm 1 of Chapter 4 since we found that this algorithm was giving the same thing as the inclusion-exclusion map in a direct way. We now give the simplified reduction of the map described above.

5.2 The Mapping

Assume for the rest of the Chapter that $\alpha \leq \beta$.

The factor $\alpha\beta$ in the statement of Lemma 27 won't be considered here, so we will assume no values on vertices (just keep in memory the two values in the first transposition that contained an element from the cycle $(1, 2, 3, \ldots, \alpha)$ and another from $((\alpha+1), \ldots, n)$, with $\mu = (1, 2, 3, \ldots, \alpha)((\alpha+1), \ldots, n)$.

First I will give a simple bijection. Let $F_{t,\alpha}$ be the set of Pictures of factorization (of μ) such that the value n (on edge) belongs to the α-tentacles and the α-tentacles have t edges. Let $F_{t,\beta}$ be the set of Pictures of factorization (of μ) such that the value n doesn't belong to the α-tentacles and the α-tentacles have t edges, with $0 \leq t \leq \alpha - 1$. In the rest of Chapter 5, for $x, y \geq 0$, $\binom{x}{y}$ denotes the set of subsets of size y of the set N_x.

Let

$$S_{t,\alpha} = N_{\alpha}^{t-1} \times N_{\beta}^{n-t-1} \times N_{(\alpha-t)} \times \binom{n-1}{t-1}$$

$$S_{t,\beta} = N_{\alpha}^{t-1} \times N_{\beta}^{n-t-1} \times N_{(\alpha-t)} \times \binom{n-1}{t}$$

I will give a simple bijection from $F_{t,\alpha} \to S_{t,\alpha}$ and another from $F_{t,\beta} \to S_{t,\beta}$ that will be very similar.

Figure 5.4 (at the end of the chapter) gives an example of what is going on here, where the α-tentacles are on the left and the rest of the structure is on the right. Take an element in $F_{t,\alpha}$. We will first concentrate on the α-tentacles and change a bit the structure so that we can apply Algorithm 1 (of Chapter 4), after which we will do a similar thing for the rest of the structure. Take the cycle of the factorization with the α-direction, start at the vertex before visiting the biggest edge (on that cycle, since we need a point of reference), and we know the root of a tentacle could be identified with $(\alpha - t)$ vertices on the cycle of the factorization so that this tentacle is an α-tentacle (since we know there are $(\alpha - t)$ vertices on the cycle that belongs to the α-cycle). Put $(\alpha - t)$ bars in linear order (the bars and elements in boxes and trees correspond to the structure of $E_{k,n}$, for this case $k = t, n = \alpha$). So count the position where this tentacle is attached, say the j^{th} vertex (considering only the vertices that make it an α-tentacle), then identify the root of the tentacles with the j^{th} bar . Do this for all α-tentacles, and then put the $(\alpha - t)$ bars in cyclic order. Then perform Algorithm 1 on it. After, by convention, say the biggest element, which is n, is in box 1. This puts a total order on all the other edge-labels. The $(t-1)$ edge-labels that are chosen in the set N_{n-1} are the edge-labels in the α-tentacles except n. The sequence in N_{α}^{t-1}

corresponds to the position of the values, starting with the smallest value, and then the second smallest and so on (again all except n). The element in $N_{\alpha-t}$ corresponds to the order of the vertex on the cycle of the factorization where the root of the tentacle that contains the edge-label n is identified (or in other words, the position that this tentacle occupied when starting at the vertex on the cycle of the factorization just before the biggest edge of this cycle as explained above).
So we just need to explain the sequence in N_{β}^{n-t-1}. This will be done similarly to the α-tentacles. Consider all edge-labels not in the α-tentacles. So you have the cycle and the β-tentacles. Again take each β-tentacle and following the α-direction on the cycle of the factorization, start at the vertex before the edge with biggest label (on the cycle of the factorization). Put the edge-labels of the edges of the cycle of the factorization in boxes so that the order is the same as the way they are visited under the α-direction. The bars (delimiters of the boxes) are at each ascent. If a tentacle is attached to the j^{th} vertex (considering only the vertices that make it a β-tentacle), then attach it to the j^{th} ascent (bar) following this orientation (this corresponds to descents in the β-direction). Now we get a structure in $E_{k,n}$ for $k = \alpha - t + \beta = n - t$. Now apply Algorithm 1 to it and from it, by saying the biggest edge-label is in box 1, do as for the α-tentacles by putting the position of the edge-labels starting with the smallest (again all of them except the biggest edge-label).

Do the same thing for the bijection from $F_{t,\beta} \rightarrow S_{t,\beta}$, except that the subset of N_{n-1} will contain all the t values in the α-tentacles.

We summarize the above in the following theorem.

Theorem 28. *The first process above is a bijection from the set of factorisations $F_{t,\alpha}$ to the set $S_{t,\alpha}$ and the second is a bijection from the set of factorisations $F_{t,\beta}$ to the set $S_{t,\beta}$, for $0 \leq t \leq \alpha - 1$.*

Proof. Straightforward. □

Define ρ to be the process above for all t and for the α and the β part. Then ρ is a bijection from the set of Pictures of factorizations (without values on vertices) to the set $\bigcup_{t=0}^{\alpha-1}(S_{t,\alpha} \times S_{t,\beta})$.

For $0 \leq t \leq \alpha - 1$, let

$$R_t = N_\alpha^t \times N_\beta^{n-t-1} \times \binom{n-1}{t}$$

I will give a bijection

$$\phi : R_{t-1} \times S_{t,\alpha} \times S_{t,\beta} \to R_t$$

By iterating the function ϕ until you reach an element in $R_{\alpha-1}$, you get a function from the set of factorizations to $R_{\alpha-1}$, which has the cardinality that we want (up to the factor $\alpha\beta$). Call this iterating function of ϕ by ψ. The function ϕ is described as follows (3 cases).

1. If $y \in R_{t-1}$, then

$$y = (\alpha_1, \ldots, \alpha_{t-1}) \times (\beta_1, \ldots, \beta_{n-t}) \times \{\gamma_1, \ldots, \gamma_{t-1}\}$$

for some α's, β's and γ's, where $\gamma_1 < \gamma_2 < \ldots < \gamma_{t-1}$. Let z be the β_{n-t}^{th} smallest value in $N_{n-1}\backslash\{\gamma_1, \ldots, \gamma_{t-1}\}$. Suppose $\gamma_{i-1} < z < \gamma_i$ for some i (assume $\gamma_0 = 0$ and γ_t is a number larger than n). Then

$$\phi(y) = (\alpha_1, \ldots, \alpha_{t-1}, i) \times (\beta_1, \ldots, \beta_{n-t-1}) \times \{\gamma_1, \ldots, \gamma_{t-1}, z\}$$

2. If $y \in S_{t,\alpha}$, then

$$y = (\alpha_1, \ldots, \alpha_{t-1}) \times (\beta_1, \ldots, \beta_{n-t-1}) \times (\delta_1) \times \{\gamma_1, \ldots, \gamma_{t-1}\}$$

for some α's, β, δ_1 and γ's, where $\gamma_1 < \gamma_2 < \ldots < \gamma_{t-1}$. Let z be the $(\delta_1+\beta)^{th}$ smallest element in $N_{n-1}\backslash\{\gamma_1, \ldots, \gamma_{t-1}\}$. Suppose $\gamma_{i-1} < z < \gamma_i$ for some i (assume $\gamma_0 = 0$ and γ_t is a number larger than n). Then

$$\phi(y) = (\alpha_1, \ldots, \alpha_{t-1}, i) \times (\beta_1, \ldots, \beta_{n-t-1}) \times \{\gamma_1, \ldots, \gamma_{t-1}, z\}$$

3. If $y \in S_{t,\beta}$, then

$$y = (\alpha_1, \ldots, \alpha_{t-1}) \times (\beta_1, \ldots, \beta_{n-t-1}) \times (\delta_1) \times \{\gamma_1, \ldots, \gamma_t\}$$

for some α's, β, δ_1 and γ's, where $\gamma_1 < \gamma_2 < \ldots < \gamma_t$. Then

$$\phi(y) = (\alpha_1, \ldots, \alpha_{t-1}, (\delta_1 + t)) \times (\beta_1, \ldots, \beta_{n-t-1}) \times \{\gamma_1, \ldots, \gamma_t\}$$

It is easy to see that ϕ is injective and to find its inverse ϕ^{-1} (tedious), so ϕ

is a bijection. By iterating ϕ, we get the following theorem.

Theorem 29. *The function ψ is a bijection from the set $\bigcup_{t=0}^{\alpha-1}(S_{t,\alpha} \times S_{t,\beta})$ to the set $R_{\alpha-1} = N_\alpha^{\alpha-1} \times N_\beta^\beta \times \binom{n-1}{\alpha-1}$, which has cardinality $\dfrac{\alpha^{\alpha-1}\beta^{\beta-1}(n-1)!}{(\alpha-1)!(\beta-1)!}$.*

We get by adjoining the factor $\alpha\beta$ for the values of the vertices the complete mapping.

Corollary 2. *The function $\psi \circ \rho$ is a bijection from the set of factorizations (by adjoining the values on vertices) to the set $N_\alpha \times N_\beta \times N_\alpha^{\alpha-1} \times N_\beta^\beta \times \binom{n-1}{\alpha-1}$, which has cardinality $\dfrac{\alpha^\alpha\beta^\beta(n-1)!}{(\alpha-1)!(\beta-1)!}$ as required.*

Proof. Immediate. □

See Figure 5.4 and Figure 5.5 for an example of $\psi \circ \rho$ (Figure 5.4 is applying ρ and Figure 5.5 is applying ψ).

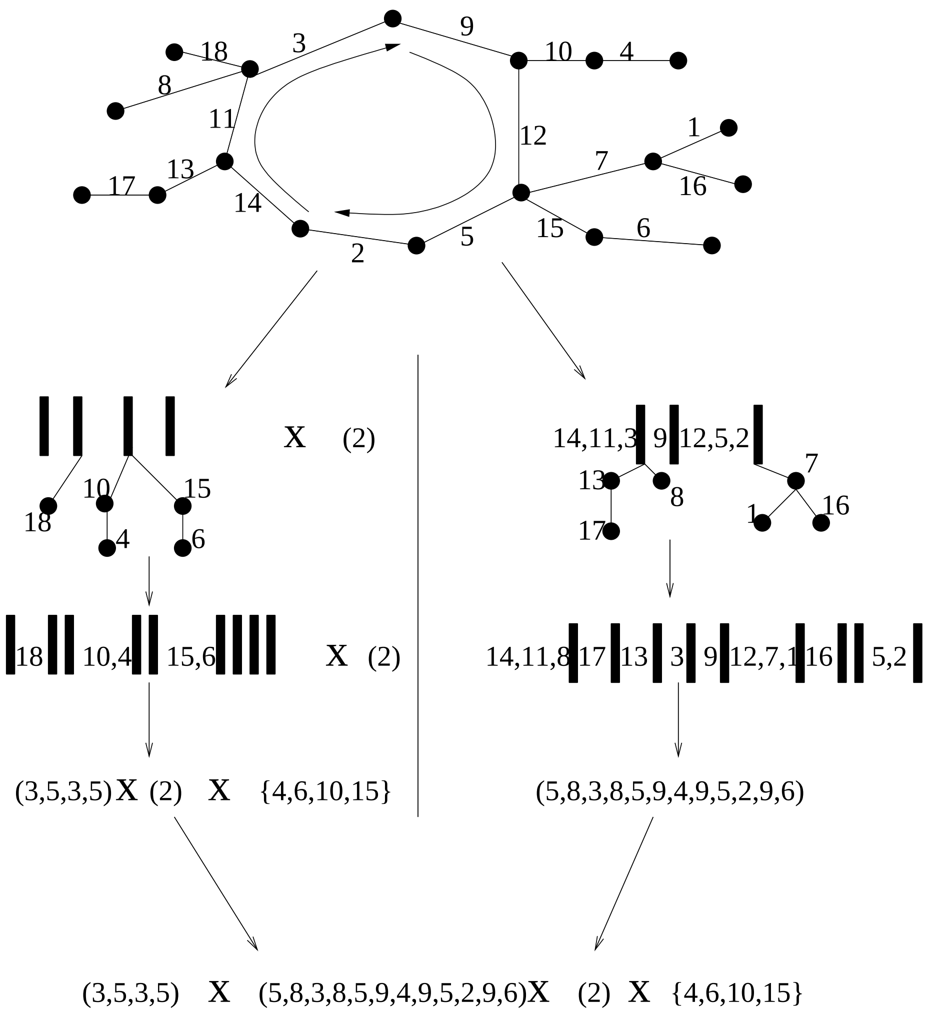

Figure 5.4: On a structure of a factorization (without values on vertices), then apply map ρ. The orientation given is the orientation that $PSIV(\alpha{-}cycle)$ traverses the cycle of the factorization

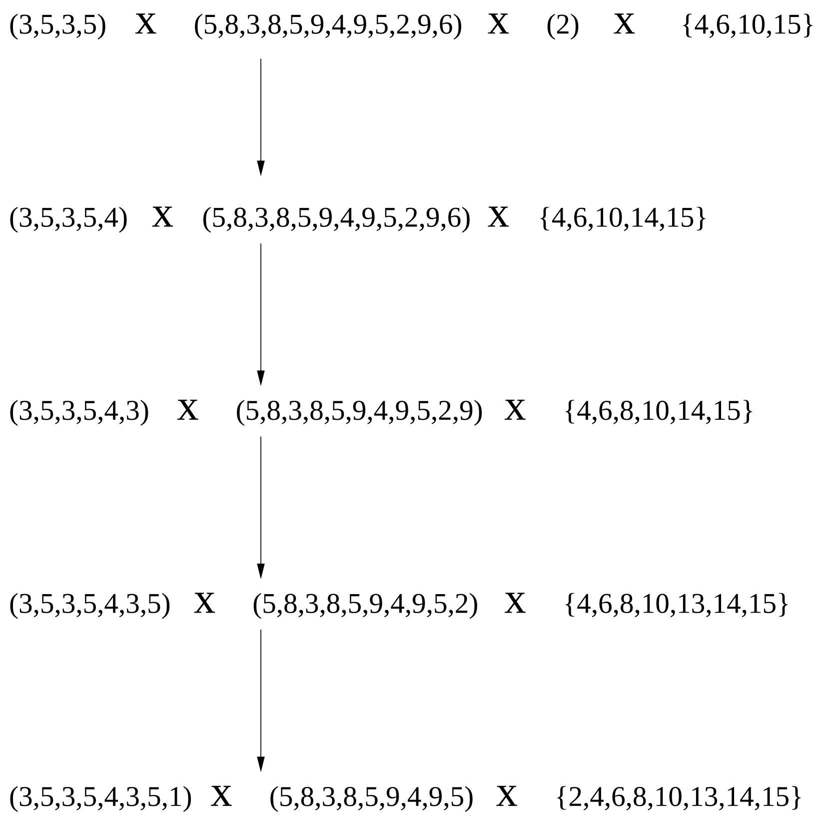

Figure 5.5: Apply ψ from Figure 5.4

Chapter 6

Conclusion

The map that is given in Chapter 5 is artificial. It is a direct map but it looks too complicated. Finding some structures that would make it nicer would be preferable. Towards it, maybe changing the map for the telescoping sum would help, although it would probably not make the argument that much nicer. Trying to give some other bijections with other kinds of known structures would be great especially for the case where $k > n$ in Chapter 4. This would be interesting and maybe could give some insights to what these nice objects that we are trying to find should be. Algorithm 1 is interesting in itself. We could compare Algorithm 1 to the other bijections in the literature between forests and Parking Functions (this is when $k = n - 1$).

The enumeration that we get in Chapter 4 for $k > n$ that basically counts the number of structures of factorizations (referred to as Picture of the factorization in the text) where the α-cycle is not allowed to have vertices in the tentacle but the β-cycle is allowed piques our curiosity in the sense that we are curious if we could get some reasonable formulas for the number of these structures where some cycles are allowed to have tentacles but some others are not (that is for the case where the permutation is an arbitrary permutation). An example of such an enumeration would be a permutation as a product of 3 disjoint cycles α, β and γ where the α and the β parts are allowed to have tentacles but the γ part is not. So that means that all the vertices of the γ-cycle are in the two connected components and the paths between them, but we allow the vertices of the α and β part to have vertices in tentacles. We point out that all these values could be obtained by solving a system of linear equations and using the number of minimal transitive factorizations of permutations as transpositions. If this is successful we could reverse this approach and try to give a direct enumeration for factorizations as transpositions for an arbitrary permutation. We must admit that we didn't have time to try the things mentioned in this paragraph, but propose to look at them in the future. We only realized using Maple that when a permutation is the product of 3 disjoint cycles, it is much more symmetrical with respect to this tentacles approach. In particular, you get some nice counting no matter who is allowed to have tentacles, and $(n+1)!$ for the number of Pictures when nobody is allowed to have tentacles. We will investigate this in the near future.

Index

Bibliography

[1] M. Bousquet-Mélou and G. Schaeffer, *Enumeration of planar constellations*, Adv. Appl. Math **24** (2000), 337–368.

[2] J. Dénes, *The representation of a permutation as the product of a minimal number of transpositions and its connection with the theory of graphs*, Publ. Math. Institute Hung. Acad. Sci. **4** (1959), 63–70.

[3] I.P. Goulden and D.M. Jackson, *Combinatorial enumeration*, John Wiley and Sons,Dover Reprint.

[4] ______, *Transitive factorizations into transpositions and holomorphic mappings of the sphere*, Proc. Amer. Math. Soc. **125** (1997), 51–60.

[5] I.P. Goulden and S. Pepper, *Labelled trees and factorizations of a cycle into transpositions*, Discrete Math. **113** (1993), 263–268.

[6] I.P. Goulden and A. Yong, *Tree-like properties of cycle factorizations*, J. Combin. Theory Ser. A **125** (1997), 51–60.

[7] A. Hurwitz, *Ueber Riemann'sche Flachen mit gegebenen Verzeigungspunkten*, Mathematische Annalen **39** (1891), 1–60.

[8] Dongsu Kim and Seunghyun Seo, *Transitive cycle factorizations and prime parking functions*, J. Combin Theory Series A **124** (2003), 125–135.

[9] A.G. Konheim and B.Weiss, *An occupancy discipline and applications*, Siam J. Appl. Math **14** (1966), 1266–1274.

[10] P. Moszkowski, *A solution to a problem of Dénes: a bijection between trees and factorizations of cyclic permutations*, European J. Combinatorics **10** (1989), 13–16.

[11] A. Rattan, *Permutation factorizations and prime parking functions*, Annals Comb. **10** (2006), 237–254.

[12] J. Riordan, *Ballots and trees*, J. Combin Theory **6** (1969), 408–411.

[13] V. Strehl, *Minimal transitive product of transpositions-the reconstruction of a proof of A. Hurwitz*, Sem. Lothar. Comb. **37** (1996).