Parking functions, stack-sortable permutations, and spaces of paths in the Johnson graph

Catalin Zara
Department of Mathematics,
Yale University, New Haven, CT, USA
zara@math.yale.edu

Submitted: Apr 4, 2002; Accepted: Apr 10, 2003; Published: Apr 23, 2003
MR Subject Classifications: 05A05, 05C38

Abstract

We prove that the space of possible final configurations for a parking problem is parameterized by the vertices of a regular Bruhat graph associated to a 231-avoiding permutation, and we show how this relates to parameterizing certain spaces of paths in the Johnson graph.

1 Introduction

There are n seats on a row in a cinema hall, numbered $1, \ldots, n$ starting from the aisle; $n-k$ seats are already taken, but seats $q = \{q_1 < q_2 < \cdots < q_k\}$ are still available. Some k late patrons want to take these seats; they enter the row and start advancing. When they are in front of seats $p = \{p_1 < p_2 < \cdots < p_k\}$, the lights go off. Will they be able to find the empty seats? (They can only advance towards the end of the row, but not go back. Also, a patron can pass a patron in front of him only if that second patron is already seated. Intuitively, it is like the late patrons are choosing their seats in the order in which they entered the row.) It is clear that this is impossible if, for some i, the patron in front of seat p_i (for short, patron p_i) passed more than $k+1-i$ empty seats, so we assume that $p_i \leq q_i$, for all $i = 1, \ldots, k$; if this condition is satisfied, we say that $p \preceq q$ and call p, q an *initial position*.

A possible final arrangement is given by a permutation $u \in S_k$: for each $i = 1, \ldots, k$, patron p_i takes the seat $q_{u(i)}$. For an initial position p, q, not all permutations might appear, since a patron can't go back; we call a permutation $u \in S_k$ *attainable from* p, q (or, simply, *attainable*, when no confusion is possible), if u can appear as the permutation corresponding to a final arrangement if the initial position is p, q. For example, if everyone knows how many empty seats he passed by, then, for each $i = 1, \ldots, k$, the patron p_i could take the i^{th} available seat, q_i; the corresponding permutation is then the identity permutation. On the other hand, if everyone plays safe, then each patron will take the first seat available (next to or in front of him) after the lights went off. Then patron p_i will end up occupying seat $q_{w(i)}$, where $w = w_{p,q} \in S_k$ is a permutation determined by the relative ordering of the elements of p and q. We call such a permutation *the safe choice permutation* (or simply, *the safe permutation*) for the initial position p, q.

Example 1.1. If there are 5 seats in a row, with seats $2, 4$ and 5 empty, and patrons in front of seats $1, 2$ and 4, then the relative ordering is $p_1 < p_2 \leq q_1 < p_3 \leq q_2 < q_3$, the safe permutation is (312) and the attainable permutations are $(312), (132), (213)$, and (123). (Throughout this paper we are using the abbreviated notation for the two-line form of the permutation and *not* the cycle notation; in particular, (312) is the permutation that sends $1 \rightarrow 3$, $2 \rightarrow 1$, and $3 \rightarrow 2$).

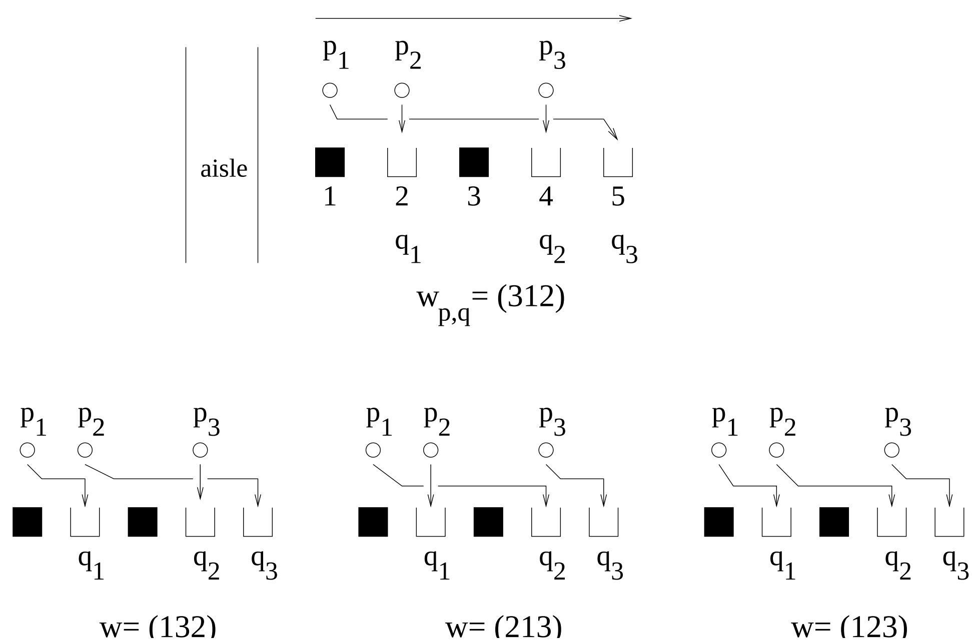

Figure 1: Safe permutation and other attainable permutations

In this note we answer the following questions:

1. Which permutations can appear as safe permutations? (For example, are there any initial positions for which the corresponding safe permutation is (146235)? Same question for (421365).)

2. If the safe permutation is w, which other permutations are attainable? (For example, if for some initial position the safe permutation is (532146), is it possible to obtain the permutation (524136)?)

3. What does this have to do with parking functions, stack-sortable permutations, or with parameterizing spaces of paths in the Johnson graph?

2 Statements of the results

In Section 4 we describe the permutations that can appear as safe permutations, and we prove the following theorem.

Theorem 2.1. *Let $w \in S_k$ and let* $\mathrm{is}(w)$ *be the length of a longest increasing subsequence in w.*

1. *If w is a safe permutation, then it avoids the pattern 231, i.e., there do not exist indices $i_1 < i_2 < i_3$ such that $w(i_3) < w(i_1) < w(i_2)$.*
2. *If w avoids the pattern 231, then w can be realized as the safe permutation for some initial position if and only if $n \geq 2k - \mathrm{is}(w)$.*

This answers the first of our questions. The permutation $u = (146235)$ can't appear as a safe permutation, since it contains the pattern 231, as 462 on positions 2, 3, and 4. On the other hand, $w = (421365)$ avoids the pattern 231, hence it can be realized as a safe permutation, if we have enough space. A longest increasing subsequence, $(1, 3, 5)$, has length 3, so we need at least 9 seats. The relative ordering of the initial positions that produce w is

$$p_1 < p_2 < p_3 \leq q_1 < q_2 < p_4 \leq q_3 < q_4 < p_5 < p_6 \leq q_5 < q_6 \ .$$

The proof of Theorem 2.1 is based on the connection between 231-avoiding permutations and *stack-sortable* permutations, and we introduce the necessary definitions and results about stack-sortable permutations in Section 3 (see [S, p.226], [K, p.239] for more details).

In Section 5 we determine which permutations are attainable from some initial position. Let τ_{ij} be the transposition interchanging $i < j$. If $w \in S_k$ is a permutation, we say that $w \prec w\tau_{ij}$ if $w(i) < w(j)$. The *Bruhat order* on S_k is the partial order generated by these relations (see [F]).

Remark 1. We are using the same notation, $\preceq$, both for comparing permutations and for comparing increasing sequences. While this might be somewhat annoying, there is a good reason for doing this: we are, in fact, talking about the same relation. To each increasing sequence $p_1 < p_2 < \cdots < p_k$ of elements from $\{1, \ldots, n\}$ corresponds a permutation $w_p = (p_1 p_2 \ldots p_k p'_1 \ldots p'_{n-k}) \in S_n$, where $p'_1 < p'_2 < \cdots < p'_{n-k}$ are the elements of the complement of p. Then $p \preceq q$ (as increasing sequences) if and only if $w_p \preceq w_q$ (as permutations). Moreover, if $u, v \in S_n$, then $u \preceq v$ (as permutations) if and only if

$$\{u(1), \ldots, u(r)\}\!\uparrow \;\preceq\; \{v(1), \ldots, v(r)\}\!\uparrow$$

(as increasing sequences) for all $r = 1, \ldots n$, where, for a set S, $S\!\uparrow$ is the sequence obtained by arranging the elements of S in increasing order.

Definition 2.1. For a permutation $w \in S_k$, the *flow-down* of w is the set

$$X(w) = \{u \in S_k \ : \ u \preceq w\} \subseteq S_k \, .$$

Theorem 2.2. *A permutation $u \in S_k$ is attainable from the initial position p, q if and only if $u \in X(w_{p,q})$, where $w_{p,q}$ is the safe permutation for p, q.*

In particular, if $w_{p,q} = (532146)$ and $u = (524136)$, then $\{5,2,4\}\uparrow \not\preceq \{5,3,2\}\uparrow$, hence $u \not\preceq w_{p,q}$. Therefore u is not attainable.

In Section 6 we show how these questions and their answers are related to parameterizing certain spaces of paths in the Johnson graph.

The Johnson graph, $J(n,k)$, is the graph whose vertices are the k-element subsets of $\{1,\dots,n\}$, and whose edges are (V_1, V_2) such that $\#(V_1 \cap V_2) = k-1$ (see [BCN]). If $V_1 = V \cup \{i\}$ and $V_2 = V \cup \{j\}$, with $\#V = k-1$, we label the oriented edge $e = V_1V_2$ by (i,j); if $i < j$, we say that the edge $e = V_1V_2$ is *ascending*. We also say that V_2 has been obtained from V_1 by replacing i with j. A path

$$\gamma \ : V_1 \stackrel{(i_1,j_1)}{\longrightarrow} V_2 \stackrel{(i_2,j_2)}{\longrightarrow} \dots \stackrel{(i_m,j_m)}{\longrightarrow} V_{m+1}$$

is called *ascending* if each edge is ascending, and is called *relevant* if it is ascending and satisfies the condition

$$i_1 > i_2 > \cdots > i_m \, .$$

Let $\Omega_{p,q}$ be the set of relevant paths from p to q; this set is non-empty if and only if p, q is an initial position.

Example 2.1. The relevant paths from $p = \{1,2,4\}$ to $q = \{2,4,5\}$ are

$$\begin{array}{lll}
\gamma_1 & : & \{1,2,4\} \stackrel{(4,5)}{\longrightarrow} \{1,2,5\} \stackrel{(2,4)}{\longrightarrow} \{1,4,5\} \stackrel{(1,2)}{\longrightarrow} \{2,4,5\} \\
\gamma_2 & : & \{1,2,4\} \stackrel{(2,5)}{\longrightarrow} \{1,4,5\} \stackrel{(1,2)}{\longrightarrow} \{2,4,5\} \\
\gamma_3 & : & \{1,2,4\} \stackrel{(4,5)}{\longrightarrow} \{1,2,5\} \stackrel{(1,4)}{\longrightarrow} \{2,4,5\} \\
\gamma_4 & : & \{1,2,4\} \stackrel{(1,5)}{\longrightarrow} \{2,4,5\}
\end{array}$$

and $\Omega_{p,q} = \{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$.

The main result in Section 6 is the following:

Theorem 2.3. *There exists a bijection between $\Omega_{p,q}$ and $X(w_{p,q})$.*

Acknowledgements. I am grateful to Victor Guillemin, Ethan Bolker, and Thom Pietraho for their valuable comments on early drafts of this paper, to Ioana Dumitriu for her comments on the many faces of 231-avoiding permutations, to Michael Kleber and Anders Buch for suggesting the connection between initial positions and parking functions, and to Sara Billey for many useful comments, including (but not limited to) the possible relation between safe permutations and stack-sortable permutations. At last, but not at least, my thanks to a very careful referee, whose suggestions and comments improved the content and the presentation of this paper.

3 Stack-sortable permutations

Stack-sorting is an algorithm that takes a list L of distinct numbers and changes it to a list L'. At each stage of the process, we have a *right* list, a *left* list, and a *stack* (a last-in-first-out list) between them. The process starts with the left list and the stack empty, and the right list L, and ends with the stack and the right list empty, and the left list L', consisting of the same elements as L, but not necessarily in the same order. One step of the algorithm consists in moving an element from the right list to the stack or from the stack to the left list, according to the following rules:

1. If the stack is empty or if the first element of the right list is smaller than the top of the stack, we move the first element of the right list into the stack (as the new top).

2. If the right list is empty or if the first element of the right list is greater than the top of the stack, then we move the top out of the stack and add it as the last element of the left list.

The algorithm is successful if the list L' contains the elements of L sorted increasingly; if this is the case, we say that L is *stack-sortable.* To each permutation $u \in S_k$ corresponds a list $L_u = [u(1), \ldots, u(k)]$, and we say that u is stack-sortable if L_u is stack-sortable.

For example, the sorting processes for (312) and for (231) go as follows:

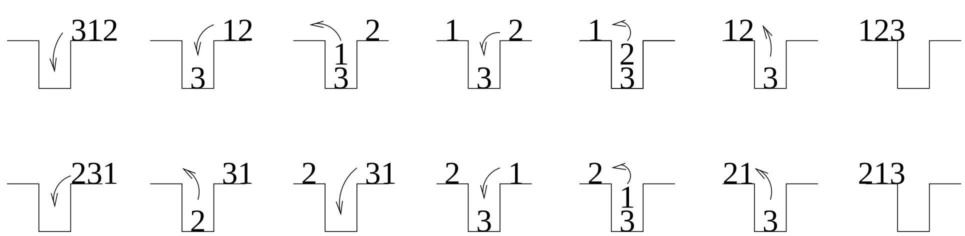

Figure 2: Sorting processes

In the examples above, (312) is stack-sortable, but (231) is not. In fact, in some sense, this is the only thing that can prevent a permutation from being stack-sortable.

Theorem 3.1. *A permutation is stack-sortable if and only if it avoids the pattern 231.*

4 Safe permutations and pattern avoidance

Let $p, q \in J(n, k)$, with $p_1 < \cdots < p_k$ and $q_1 < \cdots < q_k$. We assume that p, q is an initial position, hence that $p_i \leq q_i$ for all $i = 1, \ldots, k$. Let $w_{p,q}$ be the safe permutation associated to the initial position p, q.

Patron p_k takes the first empty seat next to or in front of him, hence

$$w_{p,q}(k) = \min\{j \ : \ p_k \leq q_j\} \ .$$

Patron p_{k-1} takes the first initially empty seat next to or in front of him, unless that seat has already been taken by p_k, in which case p_{k-1} advances and takes the first remaining seat. Similarly for the other patrons, hence, for $s = k-1, \ldots, 1$,

$$w_{p,q}(s) = \min\{j \ : \ j \neq w_{p,q}(s+1), \ldots, w_{p,q}(k) \text{ and } p_s \leq q_j\} \ .$$

We are now ready to prove that safe permutations avoid the pattern 231, and that any permutation that avoids the pattern 231 can be realized as a safe permutation, if we have enough spaces (Theorem 2.1).

Proof of Theorem 2.1. Let p, q be an initial position and let w be the safe permutation associated to p, q. Suppose w contains the pattern 231, i.e., suppose there exist $i_1 < i_2 < i_3$ such that $w(i_3) < w(i_1) < w(i_2)$. Then

$$p_{i_1} < p_{i_2} < p_{i_3} \leq q_{w(i_3)} < q_{w(i_1)} < q_{w(i_2)} \ ,$$

and this contradicts the definition of $w(i_2)$. This proves part 1 of the theorem.

To prove part 2, let w be a permutation that avoids the pattern 231. Then w is stack-sortable and we use the stack-sorting algorithm to construct an initial position p, q such that $w_{p,q} = w$, as follows:

1. Whenever we move an element from the right list to the stack we add an element p_i of p, strictly greater than all the elements of p, q already added.

2. Whenever we move an element x from the stack to the left list we add an element q_i of q, strictly greater than all the elements of p, q already added, *except* when the previous move in the sorting algorithm was moving x into the stack, in which case the new element of q_i may be greater than or equal to the last element of p, q already added.

Before discussing the general case, it might be clearer how this works if we consider a particular permutation. For example, the construction of an initial position for (421365) goes as follows:

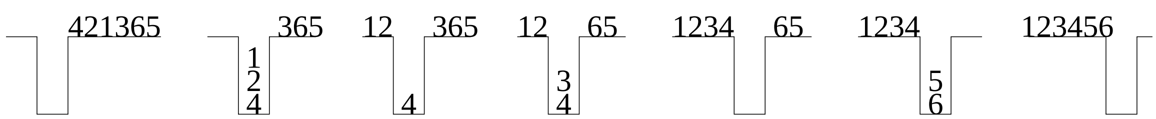

Figure 3: Sorting (421365)

Put 4, 2, and 1 into the stack – start with $p_1 < p_2 < p_3$. Move 1 (just added to the stack) and 2 out – add $q_2 > q_1 \geq p_3$. Put 3 in – add $p_4 > q_2$. Move 3 (just added) and 4 out – add $q_4 > q_3 \geq p_4$. (Notice that at this point, the stack is empty). Put 6 and 5 in – add $p_6 > p_5 > q_4$. Move 5 (just added) and 6 out – add $q_6 > q_5 \geq p_6$. The relative ordering of the elements of the initial position p, q is

$$p_1 < p_2 < p_3 \leq q_1 < q_2 < p_4 \leq q_3 < q_4 < p_5 < p_6 \leq q_5 < q_6 \ .$$

It is easy to check that p, q is an initial position and that the corresponding safe permutation is (421365). Moreover, p, q is feasible if and only if $n \geq 9$ $(= 12 - 3)$, and the length of a longest increasing subsequence in (421365) is 3 (for example, $1 < 3 < 5$).

Returning to the general case, we have to prove the following statements:

Claim 1: p, q is indeed an initial position.

To prove that $p \preceq q$, it suffices to show that for every $N \geq 1$, the number of elements of p that are less than N is greater or equal to the number of elements of q that are less than N. But this is clear, since in our construction process, at any given time, the number of elements of p added is the same as the number of elements that have entered the stack, the number of elements of q added is the same as the number of elements that have already been moved out of the stack, and the stack was initially empty.

Claim 2: The safe permutation corresponding to p, q is w.

To see this, we analyze the reverse "mixing" process: we start with the identity permutation $(12 \ldots k)$ as the left list and, by moving from right to left through the increasing sequence of elements of both p and q and reversing the actions taken during the sorting algorithm, we reconstruct w as our right list. Therefore, when we encounter an element of q, we move one element from the end of the left list into the stack, and when we encounter an element of p, we move the top of the stack to the beginning of the right list. If an element of p is equal to an element of q, then we read the element of q first.

The first elements encountered are from q, hence we start by moving $k, k-1, \ldots$ into the stack, until we arrive at the first element of p, p_k. At this point, the top of the stack is the index of the last element of q encountered. We move the top of the stack out, as the first entry of the right list. Since this reverse process reconstructs w, this first element is $w(k)$. So $q_{w(k)}$ is the smallest element of q which is not smaller than p_k, and therefore $w(k) = w_{p,q}(k)$. Now $w_{p,q}(k)$ is no longer in the stack. We continue our reverse procedure and add elements to the stack, until we arrive at p_{k-1}. The last element of q encountered is $q_{w(k-1)}$, and this is the smallest element of q not smaller than p_{k-1}, other than (possibly) $q_{w_{p,q}(k)}$, hence

$$w(k-1) = \min\{j \ : \ p_{k-1} < q_j \text{ and } j \neq w_{p,q}(k)\} = w_{p,q}(k-1) \, .$$

In general, when we arrive at the element p_s of p, the top of the stack is the index of the smallest element of q which is not smaller than p_s, other than the elements of q that have already been moved out of the stack. Therefore

$$w(s) = \min\{j \ : \ p_s \leq q_j \text{ and } j \neq w_{p,q}(k), w_{p,q}(k-1), \ldots, w_{p,q}(s+1)\} = w_{p,q}(s)$$

for all $s = k, \ldots, 1$, hence $w_{p,q} = w$.

Claim 3: We need $n \geq 2k - \mathrm{is}(w)$ to realize an initial position p, q with safe permutation w.

The relative ordering of the elements of p and q is completely determined by the safe permutation $w_{p,q}$, hence all we need to show here is that our construction of the initial position p, q works if and only if there $n \geq 2k - \mathrm{is}(w)$

Clearly any initial position can be realized if $n \geq 2k$. But we don't really need all these spaces: some of the inequalities between elements of p and q are non-strict, so we can save space by requiring that the non-strict inequalities be in fact equalities. How many such non-strict inequalities are there? We have such an inequality each time an element is moved out of the stack immediately after it has been added, and this happens precisely when we reach the end of a consecutive decreasing subsequence in w, so we may save at most as many spaces as the number of consecutive decreasing subsequences of w. Let

$$\begin{gathered} w(1) > w(2) > \cdots > w(a_1) \; , \\ w(a_1+1) > \cdots > w(a_2) \; , \\ \cdots \\ w(a_{d-1}+1) > \cdots > w(a_d) \; , \end{gathered}$$

be the consecutive descending subsequences in w, with $a_1 < a_2 < \cdots < a_d$. A subsequence of w with length strictly greater than d must have two terms in the same consecutive descending subsequence of w, hence it can not be increasing; therefore $\mathrm{is}(w) \leq d$. But since w is 231-avoiding, we have $w(a_1) < w(a_2) < \cdots < w(a_d)$, hence w has an increasing subsequence of length d. This proves that $\mathrm{is}(w) = d$. (In the example considered at the beginning of the proof, $w = (421365)$, the consecutive decreasing subsequences are $(4,2,1)$, (3), and $(6,5)$ and an increasing subsequence of maximal length is obtained by taking the last elements of these decreasing subsequences.) $\square$

There is a nice (though not efficient) criterion to recognize 231-avoiding permutations: for $w \in S_k$ we define $d(w) = (d_1(w), \ldots, d_k(w))$, where, for each $i = 1, \ldots, k$,

$$d_i(w) = \#\{j \ : \ j < i \text{ and } w(j) < w(i)\} \; .$$

Hence $d(146235) = (0,1,2,1,2,4)$ and $d(421365) = (0,0,0,2,4,4)$.

Proposition 4.1. *A permutation $w \in S_k$ is 231-avoiding if and only if the sequence $d(w)$ is weakly increasing.*

Proof. For $w \in S_k$, let $M_i(w)$ be the set whose cardinality is denoted by $d_i(w)$.

Assume that $d(w)$ is not weakly increasing, so there exists a value $i > 1$ such that $d_{i+1}(w) < d_i(w)$. Then necessarily $w(i+1) < w(i)$ and $M_{i+1}(w)$ is strictly included in $M_i(w)$. If $j \in M_i(w) - M_{i+1}(w)$, then $j, i, i+1$ corresponds to a 231 pattern in w.

Conversely, assume that w contains a 231 pattern and let $i_1 < i_2 < i_3$ be the positions where such a pattern appears, with i_3 minimal. Then $i_1, i_3 - 1, i_3$ also corresponds to a 231 pattern, $M_{i_3}(w)$ is included in $M_{i_3-1}(w)$, and because $i_1 \in M_{i_3-1}(w) - M_{i_3}(w)$, the inclusion is strict. Therefore $d_{i_3}(w) < d_{i_3-1}(w)$, so $d(w)$ is not weakly increasing. $\square$

Remark 2. It is easy to see that $d_i(w) = c_{w_0(i)}(ww_0)$, where $c(u)$ is the code of the permutation $u \in S_k$ (see [M]) and $w_0 = (k \;\; k-1 \;\; \ldots 2 \;\; 1) \in S_k$. (The *length* of a permutation w is the number of inversions in w and w_0 is the longest permutation in S_k.)

Since w is 231-avoiding if and only if ww_0 is 132-avoiding, Proposition 4.1 is in fact a restatement of the following observation of Sara Billey ([B]):

Proposition. *A permutation is 132-avoiding if and only if its code is weakly decreasing.*

Remark 3. We can formulate the initial problem as a parking problem (see [S, p.94]): There are k parking spaces on a one-way street, in front of houses $q_1 < \cdots < q_k$. Cars $C_1, \ldots, C_k$ (belonging to occupants of houses $p_k > p_{k-1} > \cdots > p_1$) enter the street in that order and try to park. Each driver has a preferred spot: the first space in front or after his house. If that space is taken, the car will be parked in the next available space. Hence the sequence of preferences is $\alpha_{p,q} = (\alpha_1, ., \alpha_k)$, where

$$\alpha(i) = \min\{j \ : \ p_{k+1-i} \leq q_j\} \ .$$

If $\alpha_{p,q}$ is a sequence of preferences that allows every car to park, then $\alpha_{p,q}$ is called a parking function; this is the case if and only if $p \preceq q$. When cars are parked according to $\alpha_{p,q}$, car C_i occupies space $q_{w(i)}$, where $w = w(\alpha_{p,q})$ is a permutation in S_k; this permutation is related to our safe permutation $w_{p,q}$ by

$$w(\alpha_{p,q}) = w_{p,q} w_0 \ ,$$

where w_0 is the longest permutation in S_k.

5 Attainable permutations

A permutation $u \in S_k$ is attainable from an initial position p, q if and only if

$$p_i \leq q_{u(i)} \qquad \text{for all } i = 1, \ldots, k \ . \tag{1}$$

Let $W_{p,q}$ be the set of attainable permutations. Theorem 2.2 states that $W_{p,q}$ is the set of permutations that precede (or are equal to) $w_{p,q}$ in the Bruhat order, and the proof follows from the next two lemmas.

Lemma 5.1. *If $u \in W_{p,q}$ and $u\tau_{ij} \prec u$, then $u\tau_{ij} \in W_{p,q}$.*

Proof. Assume $i < j$. Then $u(i) > u(j)$, since $u\tau_{ij} \prec u$. But then

$$p_i < p_j \leq q_{u(j)} = q_{u\tau_{ij}(i)} \ , \quad p_j \leq q_{u(j)} < q_{u(i)} = q_{u\tau_{ij}(j)} \ ,$$

and if $r \neq i, j$, then $p_r \leq q_{u(r)} = q_{u\tau_{ij}(r)}$; therefore $u\tau_{ij} \in W_{p,q}$. □

Lemma 5.2. *The only maximal element of $W_{p,q}$ is $w_{p,q}$.*

Proof. Let $u \in W_{p,q}$ be a maximal element, and assume that $u \neq w_{p,q}$. Then there exists $r \geq 1$ such that $u(i) = w_{p,q}(i)$, for $i = r+1, \ldots, k$, but $u(r) \neq w_{p,q}(r)$.

Let $t = u(r)$. Then $p_r \leq q_t$ and

$$t \notin \{u(k), \ldots, u(r+1)\} = \{w_{p,q}(k), \ldots, w_{p,q}(r+1)\} \ ,$$

hence $w_{p,q}(r) \leq t$. Then $w_{p,q}(r) = t' < t$ and

$$t' \notin \{w_{p,q}(k), \ldots, w_{p,q}(r+1)\} = \{u(k), \ldots, u(r+1)\} \ ,$$

hence $u^{-1}(t') = r' < r$. Then $r' < r$ and $u(r') = t' < t = u(r)$, hence $u \prec u\tau_{r'r}$. But

$$p_{r'} = p_r \leq q_{u(r)} = q_{u\tau_{r'r}(r')} \qquad \text{and} \qquad p_r \leq q_{w_{p,q}(r)} = q_{u(r')} = q_{u\tau_{r'r}(r)} \ ,$$

hence $u\tau_{r'r} \in W_{p,q}$. This contradicts the assumption that u is maximal in $W_{p,q}$, so $u = w_{p,q}$, and this completes the proof. □

6 Parameterizations of spaces of paths

Let p, q be vertices of the Johnson graph $J(n,k)$, such that $p \preceq q$. In Section 2 we defined the relevant paths from p to q to be the ascending paths

$$\gamma \ : \ p = V_1 \stackrel{(i_1,j_1)}{\longrightarrow} V_2 \stackrel{(i_2,j_2)}{\longrightarrow} \dots \stackrel{(i_m,j_m)}{\longrightarrow} V_{m+1} = q \tag{2}$$

which satisfy the condition

$$i_1 > i_2 > \cdots > i_m \ . \tag{3}$$

In [Z], we attach to each relevant path γ from p to q a rational expression $E(\gamma)$ in the variables $x_1, \dots, x_n$, and, by summing over the space $\Omega_{p,q}$ of relevant paths from p to q, we get a *polynomial*

$$S_{p,q} = \sum_{\gamma \in \Omega_{p,q}} E(\gamma) \in \mathbb{C}[x_1, \dots, x_n] \ . \tag{4}$$

This procedure is akin to a discrete version of integration over $\Omega_{p,q}$ and in this section we prove that the space $\Omega_{p,q}$ of relevant paths from p to q can be parameterized by the set $X(w_{p,q})$ of attainable permutations for the initial position p, q (Theorem 2.3).

Proof of Theorem 2.3. Let $p = \{p_1 < \cdots < p_k\}$, $q = \{q_1 < \cdots < q_k\}$ be an initial position. A key consequence of (3) is that along a relevant path (2), an element that has been added can't be replaced, and hence all the added elements are distinct. Therefore we can define a map $\Phi : \Omega_{p,q} \to S_k$ by associating a permutation $v = \Phi(\gamma) \in S_k$ to each relevant path (2), according to the following rules:

1. If $p_i = i_r$, then $v(i)$ is defined by $j_r = q_{v(i)}$.
2. If $p_i \notin \{i_1, \dots, i_m\}$, then $v(i)$ is defined by $p_i = q_{v(i)}$.

To can make the relation between a relevant path γ and its associated permutation $v = \Phi(\gamma)$ more suggestive by representing the path as

$$\gamma \ : \quad p \stackrel{(p_k, q_{v(k)})}{\longrightarrow} \dots \stackrel{(p_1, q_{v(1)})}{\longrightarrow} q \ ; \tag{5}$$

if $p_i = q_{v(i)}$, then the corresponding "edge" is in fact a loop that starts and ends at the same vertex, and we can delete this loop from our path.

From (1) and (5) it follows that Φ is injective and the image of Φ is $X(w_{p,q})$, the set of attainable permutations. Therefore Φ induces a bijection between $\Omega_{p,q}$ and $X(w_{p,q})$. $\square$

Example 6.1. For the paths in Example 2.1, the permutations are

$$\Phi(\gamma_1) = (123), \quad \Phi(\gamma_2) = (132), \quad \Phi(\gamma_3) = (213), \quad \Phi(\gamma_4) = (312) \ .$$

The Bruhat graph is the Cayley graph associated to (S_k, T), where T is the set of transpositions in S_k: the vertices are permutations $w \in S_k$ and two vertices w_1 and w_2 are joined by an edge if $w_1^{-1} w_2$ is a transposition. This is a regular graph, since each vertex has $k(k-1)/2$ neighbors. For a permutation $w \in S_k$, the *Bruhat graph of* w is the subgraph of the Bruhat graph whose set of vertices is $X(w)$, the flow-down from w.

If w avoids the patterns 3412 and 4231, then the Bruhat graph of w, $X(w)$, is a regular subgraph of S_k (see [LS], [C]): each vertex of $X(w)$ has $\ell(w)$ neighbors in $X(w)$, where $\ell(w)$ is the length of w.

For an initial position p, q, we have proved in Theorem 2.2 that the set of attainable permutations is $X(w_{p,q})$. By Theorem 2.1, $w_{p,q}$ avoids the pattern 231, hence it also avoids the patterns 3412 and 4231. Therefore Theorem 2.3 states that $\Omega_{p,q}$ is parameterized by the vertices of a regular graph, and in [Z] we use a localization formula and the parameterization given in Theorem 2.3 to show that (4) is indeed an integral over a smooth manifold.

Remark. It may seem bizarre that a finite sum *is* actually an integral over a smooth manifold, but this is one of the beauties of integrating *equivariant* classes (see [GS] for details). In the situation considered in [Z], an equivariant class is determined by its values at the fixed points, and, by the localization formula of Atiyah-Bott-Berline-Vergne, its integral is computed by adding the contributions of the finitely many fixed points.

References

[B] S. Billey, personal communication.

[BCN] A. E. Brouwer, A. M. Cohen, and A. Neumaier, *Distance-regular graphs*, Springer-Verlag, Berlin, 1989.

[C] J. Carell, On the smooth points of a Schubert variety, *CMS Conference Proceedings*, **16** (1994) 15–24, *Proc. of the conference on Representations of Groups: Lie, Algebraic, Finite, and Quantum*, June 1994.

[LS] V. Lakshmibai and B. Sandhya, Criterion for smoothness of Schubert varieties in $\mathrm{Sl}(n)/B$, *Proc. Indian Acad. Sci. Math. Sci.* **100**(1), 45–52, 1990.

[F] W. Fulton, *Young tableaux*, Cambridge Univ. Press, Cambridge, 1997.

[GS] V. Guillemin and S. Sternberg, *Supersymmetry and Equivariant de Rham Theory*, Springer-Verlag, Berlin, 1999.

[K] D.E. Knuth, *The Art of Computing Programming. Vol. 1: Fundamental Algorithms*, Addison-Wesley Publishing Co., Reading MA, 1973.

[M] I.G. Macdonald, *Notes on Schubert polynomials*, Publ. du LACIM, vol. 6, Univ. du Québec à Montreal, 1991.

[S] R. Stanley, *Enumerative combinatorics, Vol. 2*, Cambridge Univ. Press, Cambridge, 1999.

[Z] C. Zara, Generators for the equivariantly cohomology ring of GKM manifolds, *in preparation.*