Lorenz Wernisch

Dominance Relation on Point Sets and Aligned Rectangles

Dissertation

eingereicht am Fachbereich Mathematik und Informatik
der Freien Universität Berlin
1994

Betreuer: Prof. Dr. Emo Welzl

transparent verlag

1. Auflage, Berlin 1996

Gesamtherstellung und Vertrieb
H. u. E. Preuß
Reichenbergstraße 28
10999 Berlin

ISBN 3–927055–06–9

Acknowledgements

The thesis is based on common work with Stefan Felsner (see [7] and [6]). He deserves the greatest thank not only for making this work possible but also for his persistent encouragement and the numerous critical remarks on the manuscript. The original question, how to compute k-chains of permutations quickly, which motivated the work, was posed by Emo Welzl. I thank him for his strong and constant interest in the work. He gave me the opportunity to work for years in the Computer Science Department at the Free University Berlin. The discussions with my colleagues in the "Mittagsseminar" have also been helpful.

Contents

1 Introduction

An *order* on a set X is a relation R that is antisymmetric, $(p,p) \notin R$, and transitive, $(p,q),(q,r) \in R$ implies $(p,r) \in R$. Throughout the paper we assume that X is finite and has size n. If $(p,q) \in R$ then p is *less than* q and q is *greater than* p. If (p,q) or (q,p) in R then they are *comparable*, otherwise they are *incomparable*. An element of an order that is not less than any other is *maximal* as one not greater than any other is *minimal*. For a *weighted order* there additionally exists some weight function $w\colon X \to \mathbf{N}$ of the elements of X to natural numbers. The weight $w(Y)$ of a subset $Y \subseteq X$ then is the sum of the weights of the elements of Y.

We will be mainly concerned with a very special type of order, the *dominance order* on points in the euclidean k-dimensional space $\mathbf{R}^k$. If $x = (x_1, \ldots, x_k)$ and $y = (y_1, \ldots, y_k)$ are points in $\mathbf{R}^k$, then x is said to be *dominated* by y, if $x_i < y_i$, for all $i = 1, \ldots, k$. This order may be extended to aligned *boxes*, i.e., sets of the form $\{(x_1, \ldots, x_k) \in \mathbf{R}^k : l_i \leq x_i \leq u_i,\ 1 \leq i \leq k\}$ where $(l_1, \ldots, l_k)$ is the *lower corner* and $(u_1, \ldots, u_k)$ is the *upper corner* of the box. A box b dominates a box b' if the lower corner of b dominates the upper corner of b'. We refer to a set of boxes together with this dominance order as *box order*. See the right picture of Fig. 4 for an example of a box order with comparabilities as indicated. Note that the dominance order on points may be seen as a special case of a dominance order on boxes, where the upper and lower point of each box coincides.

The next sections deal with two special kinds of subsets of an order X. A *chain* is a set of pairwise comparable elements and an *antichain* is a set of pairwise incomparable ones. The *height* of an element $x \in X$ is the size of a longest chain with x as maximal element. Of course, two points of the same height cannot be comparable. Hence, collecting points with the same height in the same set yields a partition $\mathcal{A}$ of X into antichains, the *canonical antichain partition*. Observe that this partition is also obtained by a repeated removal of the set of minimal elements. By definition, the number of antichains in a canonical partition is the size of a largest chain in X, a *maximum chain* (see Fig. 1). Since, obviously, a chain and an antichain can have at most one point in common, there can be no partition into fewer antichains than there are in $\mathcal{A}$, i.e., it is a *minimal antichain partition*.

An antichain A as large as possible in an order is a *maximum antichain* and, according to Dilworth's Theorem [4], there exists a partition of the order

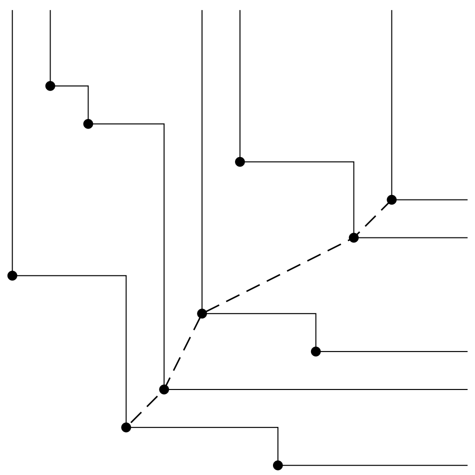

Figure 1: The canonical antichain partition and a maximum chain.

into a minimum number of chains, a *minimal chain partition*, such that each chain meets A. In the special case of points in the plane—let their coordinates be denoted by x and y—this easily follows from the above observation on minimum antichain partitions since chains and antichains interchange their roles by a reflection of the plane on the y coordinate, i.e., by mapping (x, y) to $(-x, y)$. Let us call this interchange of roles the *duality principle* for the dominance order on points in the plane. Note that a permutation $\sigma\colon [1 \ldots n] \to [1 \ldots n]$ may be represented by a planar point set $\{(i, \sigma(i)) \mid 1 \leq i \leq n\}$. Then a maximum chain corresponds to a longest increasing subsequence in σ as a maximum antichain corresponds to a longest decreasing subsequence.

If a subset C of X can be covered by k chains it is called a *k-chain*. A k-chain C is *maximum* if no other k-chain contains more elements than C and it is *strict* if at least k chains are needed to cover it. Similar definitions may be given for *(strict) maximum l-antichains*. The "greedy method" that repeatedly removes maximum chains may fail in computing a maximum k-chain. Try to find a maximum 2-chain in Fig. 1. It is possible to get one of size 8. But if one starts with the shown maximum chain of size 5 there remain only chains of size 2 (Nevertheless, the approximation yielded by the greedy method is

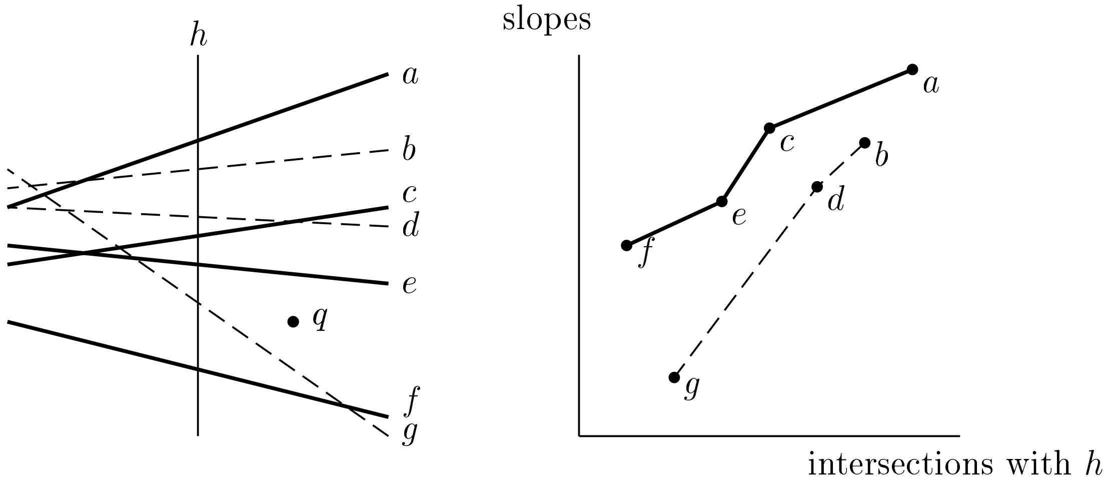

Figure 2: A set system $\mathcal{T}_R = \{R_1, R_2\}$ for line h.

not too bad as seen in Section 3). Analogously, given a weight function, a *maximum weighted k-chain* or a *maximum weighted l-antichain* is one with maximum weight among all weighted k-chains or l-antichains. Note that if we are asking for maximum weighted k-chains or -antichains, the restriction to natural numbers is not essential, since integer or rational values may be replaced by natural numbers after a proper rescaling.

Let us consider two examples of applications of maximum k-chains. Matoušek and Welzl [21] propose an algorithm that preprocesses a number of n points X in the plane such that for a given (nonvertical) query line the number of points in X above the line can be computed quickly. There exist mappings of lines to points and vice versa that preserve the relation of a point being above or below a line. Thus, we may alternatively think of n (nonvertical) lines X in the plane and the problem of finding the number of lines of X above a given query point q. The algorithm solving this problem allows $O(\sqrt{n}\,\log n)$ query time and needs $O(n \log n)$ space and $O(n^{3/2} \log n)$ preprocessing time.

The essential observation is the following. Suppose a vertical line h is given together with a set H of lines of X that are pairwise nonintersecting on the right side of h. Then it is easy to find the number of lines in H above a point q lying to the right of h by binary search in time $O(\log |H|)$. If we have two set systems of subsets of X of disjoint sets $\{L_1, \ldots, L_l\} = \mathcal{T}_L$ and disjoint sets

$\{R_1, \ldots, R_r\} = \mathcal{T}_R$, such that no two lines of L_i intersect on the left side of h and no two lines of R_i intersect on the right side of h, then we can find the number of lines in $\bigcup \mathcal{T}_L$ above q in time $O(l \log n)$, if q lies to the left of h, and the number of lines in $\bigcup \mathcal{T}_R$ above q in time $O(r \log n)$, if q lies to the right of h (see Fig. 2).

The algorithm now constructs a search tree with appropriately chosen h, $\mathcal{T}_L$, and $\mathcal{T}_R$ in the root, a left subtree for query points to the left of h and lines $X - \bigcup \mathcal{T}_L$, and a right subtree for query points to the right of h and lines $X - \bigcup \mathcal{T}_R$. Consequently, to otain a tree with small height (and thus small query time) we have to keep the sets $X - \bigcup \mathcal{T}_L$ and $X - \bigcup \mathcal{T}_R$ as small and balanced as possible. On the other hand, $l = |\mathcal{T}_L|$ and $r = |\mathcal{T}_R|$ must not become too large, since they affect the query time immediately.

For the details of the construction of a good search tree see Matoušek and Welzl [21]. However, one important step in the construction is to find such set systems $\mathcal{T}_L$ and $\mathcal{T}_R$ for a given vertical line h and a given integer k such that $l = |\mathcal{T}_L|$ and $r = |\mathcal{T}_R|$ are as large as possible under the constraint that $|\bigcup \mathcal{T}_L| \geq kl$ and $|\bigcup \mathcal{T}_R| \geq kr$. Now observe that no two lines of some set $H \subseteq X$ of lines do intersect to the right of h if the ordering of the slopes of lines in H coincides with the ordering according to the heights of the intersections of these lines with h. In other words, if all slopes of lines in X are ordered according to the heights of the intersections of the lines with h, H corresponds to an increasing subsequence of slopes. Similarly, a set H' of lines no two of them intersecting to the left of h corresponds to a decreasing subsequence of slopes. Thus, a set system $\mathcal{T}_R$ is a $|\mathcal{T}_R|$-chain (see Fig. 2) as a set system $\mathcal{T}_L$ is a $|\mathcal{T}_L|$-antichain in this permutation of slopes. Hence, we are searching for such set systems with maximum size under the above constraints.

Another application of k-chains is a kind of problem emerging in connection with VLSI design. A two-sided channel (see Fig. 3) is simply a pair of parallel lines with labeled terminals on the upper and on the lower side. The labels of terminals on one side are all different whereas the sets of labels on both sides coincide. A terminal on the upper side should be connected by some line to the terminal on the lower side with the same label. One problem consists in finding the maximum number of terminals that can be connected such that the connecting lines do not intersect. As one easily observes in Fig. 3, this amounts to computing a maximum chain in the permutation of labels of terminals on the lower side if the terminals are labeled in increasing order on the upper side. Sometimes it is possible to assume k layers where connecting lines are allowed

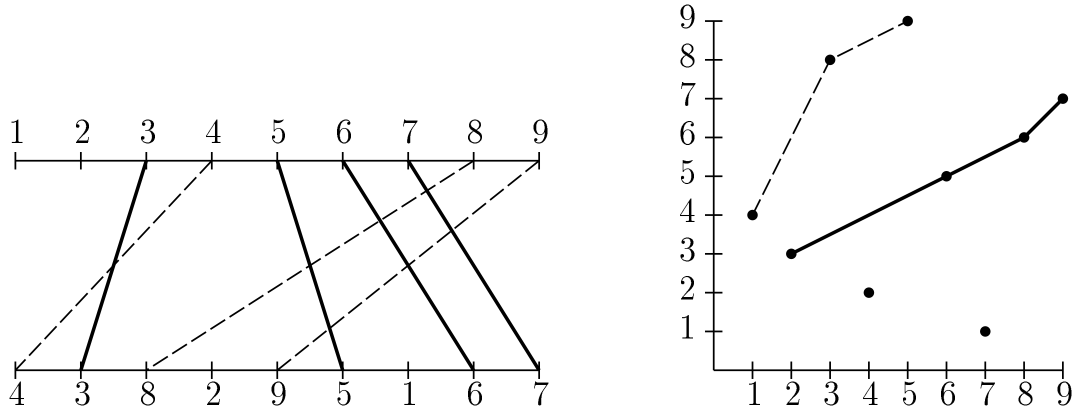

Figure 3: Connection of terminals in two layers without intersection.

to intersect provided that they lie in different layers. The maximum number of terminals that can thus be connected in k layers now is the size of a k-chain in the permutation of terminals. For some reason we might want to give some weight to the terminals, e.g., two terminals have to be connected by several lines. In this case we are searching for a maximum weighted k-chain.

How fast can k-chains or k-antichains be computed? In the sequel we do not necessarily assume k to be constant, it may vary with n. Based on considerations of Frank [9] one can design an algorithm using network flows to compute maximum k-chains in an arbitrary order in time $O(n^3)$. Gavril [11] uses network flow algorithms to compute them in time $O(kn^2)$. Based on this algorithm, Sarrafzadeh and Lou [25] propose an algorithm for maximum weighted k-chains in any order in time $O(kn^2)$. Nevertheless, the question arises whether one can do better if the type of orders considered is restricted.

There exists an algorithm to compute a longest increasing subsequence in permutations in time $O(n \log n)$ that pertains to mathematicians folklore (see Fredman [10]). Fredman [10] shows that, essentially, one cannot do with fewer comparisons in such algorithm. Anyway, restriction to dominance orders on points in the plane seems promising and Lou, Sarrafzadeh, and Lee [19] propose an algorithm to compute maximum 2-chains in time $O(n \log n)$, which again is optimal. Later they showed how to find maximum 3-chains within the same time bound [18].

In Section 3.1 we propose a general algorithm computing a maximum k-chain of a quasi-permutation in time $O(kn \log n)$ and linear space (this is joint work with Stefan Felsner [7]). (An outcome of an implementation of this algorithm is shown in Fig 31 on the last page.) The essential concept utilized in this algorithm is that of the skeleton of a point set in the plan, introduced by Viennot [33] and [32] and explained in Section 2.1. Viennots ideas are especially interesting (besides being satisfying esthetically), since they provide geometric interpretations of several results concerning permutations that otherwise remain unintuitive, e.g., of the Green-Kleitman Theorem about k-chains and l-antichains (see Section 3) or of the well-known Robinson-Schensted correspondence between permutations and pairs of Young tableaux (Robinson [3] and Schensted [26], see Section 2.2). We extend this geometric approach and show further properties of the skeleton that are applied to prove a theorem of Schützenberger concerning reversed permutations by geometric means in Section 2.3. There skew Young tableaux are also given a geometric treatment.

The algorithm of Section 3.1 is extended in Section 3.2 to work for weighted planar point sets, too. It is possible to obtain a maximum weighted k-chain in time $O(2^k n \log(2^k n))$ and space $O(2^k n)$. Of course, this makes sense only for small values of k. But note that even for constant k no better algorithm than that of Sarrafzadeh and Lou [25] with running time $O(kn^2)$ was known. Unfortunately, it is not obvious how to extend the algorithm of Section 3.3 computing a maximum k-antichain in a similar way to the weighted case.

Of course, by the duality principle we may use the algorithm for maximum k-chains to compute maximum k-antichains for planar points in time $O(kn \log n)$. On the other hand, if k gets large, say $\Omega(\sqrt{n})$, then we can do better. In Section 3.3 we describe an algorithm computing maximum k-antichains (and, by duality, k-chains) in time $O((n^2/k) \log n)$ and linear space following ideas of Viennot [32]. Consequently, a maximum k-chain can be computed in time $O(n^{3/2} \log n)$, for an arbitrary k.

After so much inspecting and computing k-chains and k-antichains one may wonder how large a “typical” k-chain in a planar point set can become at all. One possible interpretation of “typical” is to ask for the size of a k-chain in a random permutation of length n. In Section 3.4 we show how to extend known results on the expected size of a maximum chain, it is approximately $2\sqrt{n}$, for large n, to the case of k-chains.

Box orders, too, model a problem in VLSI design. Assume again a two-sided channel with labeled terminals on the upper and on the lower side. Now terminals on one side may be labeled by the same label. Again, terminals with the same label should be connected in a common net. Such a net may

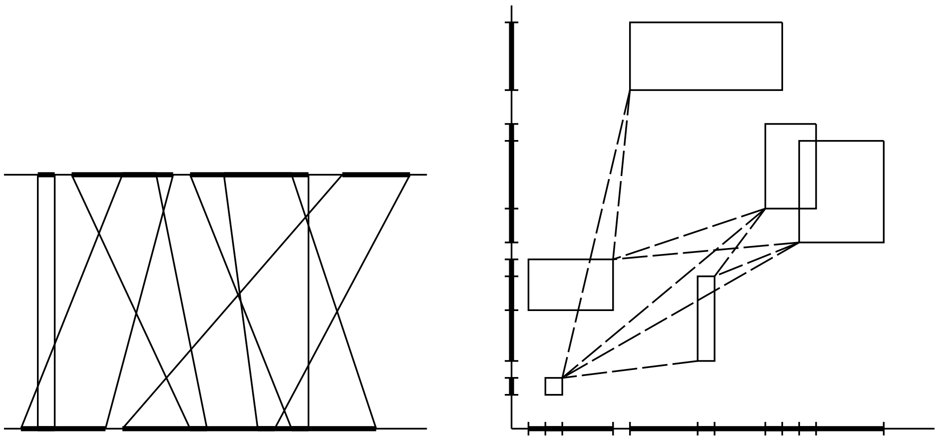

Figure 4: A box order with comparabilities and trapezoid representation.

be modeled by a trapezoid connecting the rightmost terminal of one label on the upper side to the rightmost terminal of the same label on the lower side and the leftmost upper terminal to the leftmost lower one (see Fig. 4). Nets then may be routed without intersection if the corresponding trapezoids do not intersect. The connection to box orders is the following. Interpret the terminals on the lower of the two lines of the channel as lying on the x-axis and that of the upper line as lying on the y-axis of the plane. Each trapezoid then corresponds to an axis-parallel box in the plane whose projection on the x- and y-axis coincides with the lower and upper side of the trapezoid (see Fig. 4). It is easily seen that two trapezoids are disjoint if the corresponding boxes are comparable.

Box orders are connected to *trapezoid graphs* which are introduced by Dagan, Golumbic, and Pinter [2]. A graph is a *trapezoid graph* if there exists a representation of the vertices of the graph by trapezoids in a two-sided channel such that two vertices are joined by an edge if and only if the corresponding trapezoids intersect. A chain in the box order corresponds to an independent set in the trapezoid graph and similarly, an antichain in the order corresponds to a clique in the graph. Dagan, Golumbic, and Pinter propose an algorithm computing the minimum number of independent sets to cover all vertices of a trapezoid graph, hence, computing a minimum chain partition in the corresponding box order, in time $O(n^2)$. They further give a less efficient backtracking algorithm finding a maximum clique in such a graph, i.e., finding a maximum antichain in the order.

In Section 4.1 we present an algorithm computing a maximum weighted chain in time $O(n \log n)$ and linear space (joint work with Stefan Felsner and Rudolf Müller [6]). A byproduct of this algorithm is the computation of a minimal antichain partition if started on unweighted box orders. We also show how to extend this algorithm from boxes in the plane to boxes in $\mathbf{R}^k$, for $k > 3$, using time $O(n(\log n)^{k-1})$ and space $O(n(\log n)^{k-2})$ (in the case $k = 3$ one can do slightly better). Section 4.2 shows how to compute a maximum weighted antichain for box orders also in time $O(n \log n)$ and linear space. Instead of trying to obtain a minimal chain partition simply as byproduct we propose another algorithm for this purpose working within the same time and space bounds. Note that to find a trapezoid or box representation of a trapezoid graph may take time $O(n^2)$ for a given trapezoid graph (see Ma and Spinrad [20]). Nevertheless, in the intended applications the graph is just given in form of such a representation. The question how to compute k-chains or l-antichains efficiently for box orders remains open.

Finally, let us make some general remarks. In the planar case we will often use somewhat sloppy terminology where *left* and *right* refer to decreasing and increasing x-coordinate and similarly, *above* and *below* means increasing and decreasing y-coordinate. We will even combine them, e.g., above and to the left means larger y- and smaller x-coordinate. We refer to the x- or y-coordinate of a point p by p_x and p_y, resp.

Extending the case of the representation of a permutation by points in the plane let a *quasi-permutation* (Viennot [32]) be a set of planar points with integer coordinates such that they are in *general position*, i.e., no two points have the same x- or y-coordinate. Sometimes we will be careful and provide algorithms even for points not in general position. But arguments are often much simpler if we simply assume such generality. Note that we can always perturb a set of points such that they are in general position without changing the comparability relations. Simply change the values of the x-coordinate of points with the same x-coordinate—by definition they are incomparable—by a small amount such they get increasing x-coordinates with decreasing y-coordinates. Points with the same y-coordinate are perturbed analogously. Box orders may be perturbed similarly (see Section 4).

We point to a standard method that improves most of the following algorithms. We will often have to maintain a finite set of real numbers such that values may be inserted or deleted from it and the predecessor or successor of a given query value can be found. Using balanced trees (e.g., red-black trees described in Cormen, Leiserson, Rivest [1]) all these operations can be done in $O(\log n)$ time and linear space. If we further assume the benefits of a random access machine and assume that the values are taken from a finite universe U (e.g., by sorting them once in time $O(n \log n)$) then the above operations take only $O(\log \log n)$ time and linear space when implemented on a data structure of van Emde Boas [30]. Hence, under these assumptions, the $\log n$ factor in the running time of the algorithms for 2-dimensional point or box orders may be replaced by a $\log \log n$ factor. For example, to find a maximum k-chain for planar points takes time $O(n \log n + kn \log \log n)$ for arbitrary real values on a random access machine, or, on the same machine, time $O(kn \log \log n)$ for a permutation.

2 The Skeleton

2.1 Basic Properties

Following Viennot [33], we define the *left shadow of point* $p = (x, y)$ (or *shadow* for short) as the set of all points (u, v) dominating p, i.e., points with $u > x$ and $v > y$. For an arbitrary set E of points, the *left shadow of* E is the union of the shadows of the points of E, i.e., the set of all points dominating at least one point of E (see the shaded region in Fig. 5). The *right shadow* of $p = (x, y)$ is the set of points (u, v) with $u < x$ and $v > y$. The term shadow suggests some light coming from the left below, or from the right below in the case of a right shadow (of course, this should not be taken too verbally, since the "shadows" have a form that is scarcely realizable physically). Imagining some light coming from above we define the *left downshadow* of (x, y) as the set of points (u, v) with $u < x$ and $v > y$, i.e., with light coming from left and going down (the reader may excuse the certainly awful terminology, but alternatives which may come to mind do not sound decisively better). Similarly, the *right downshadow* of (x, y) are points (u, v) with $u < x$ and $v < y$. Again, the corresponding shadows of a whole point set E are defined as the union of the according shadows of the points of E.

The *left jump line* or simply *jump line*, $L_{\nearrow}(E)$ or $L(E)$, of a point set E now is the topological boundary of the left shadow of E. Again, the *right jump line* $L_{\nwarrow}(E)$ (left and right jump line down, $L_{\searrow}$ and $L_{\swarrow}$) is defined similarly with respect to the right shadow (left, right downshadow). Let the unbounded half line of the jump line extending upwards be the *top outgoing line*, and let the unbounded half line extending to the right be the *right outgoing line*. Additionally, we use the terms *bottom outgoing line* and *left outgoing line* so that we are able to denote the corresponding infinite halflines of $L_{\nwarrow}$, $L_{\swarrow}$, and $L_{\searrow}$, too. It is easily seen that the jump line $L(A)$ of an antichain A is a downward staircase with the points of A in its lower corners. Let the points in the upper corners be the *skeleton points* or *skeleton* $S_{\nearrow}(A)$ or $S(A)$ of the antichain A. Formally, if $(x_1, y_1), \ldots, (x_k, y_k)$ are the points of A ordered by increasing x-coordinate then $S(A)$ contains the points $(x_2, y_1), \ldots, (x_k, y_{k-1})$. Hence, A has exactly $|A| - 1$ skeleton points (see Fig. 5).

The minimal elements of a point set X form an antichain A such that the rest $X - A$ lies completely in the shadow of A. Hence, by removing A and treating $X - A$ in the same way, we recursively obtain the canonical antichain

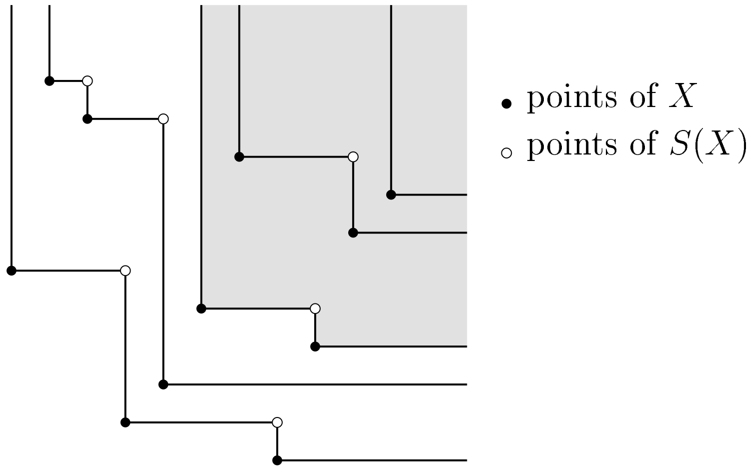

Figure 5: Point set X, its skeleton, and the shadow of 3rd layer.

partition $A = A_0, \dots, A_{\lambda-1}$ with nonintersecting jump lines $L(A_i)$, $0 \le i < \lambda$, which will be called the *layers* $L_i(X)$ *of* X. The *skeleton* or *left skeleton of* X, denoted by $S(X)$ or $S_{\nearrow}(X)$, is then defined as the union of the skeletons $S(A_i)$, $0 \le i < \lambda$. Since, as noted above, the i-th layer $L_i(X)$ has $|A_i| - 1$ skeleton points the size of $S(X)$ is $|X| - \lambda$. A picture of a point set X, its skeleton $S(X)$, its antichain layer partition, and the shadow of antichain 2 can be found in Fig. 5. Let us state an easy but quite useful observation.

Lemma 1 *Suppose a point set X is partitioned into k sets A_i in such a way that each jump line $L(A_i)$ contains all points of A_i and that the $L(A_i)$ are pairwise disjoint, for $0 \le i < k$. Then the sets A_i are the canonical antichain partition of X.*

Proof. Each A_i is an antichain. Furthermore, suppose the A_i are ordered by increasing x-coordinates of the top outgoing lines of the $L(A_i)$. Then $X - A_0$ is in the shadow of A_0. Hence, by the definition of the shadow, each point of $X - A_0$ is comparable to at least one of A_0 and no point of A_0 is comparable to any other point in X, i.e., A_0 are just the minimal elements of X. Repeat the procedure on $X - A_0$. □

SKELETON(X)
insert dummy d_u in L at height $+\infty$; $\mathit{link}(d_u) \leftarrow$ nil;
$k \leftarrow 0$; $S \leftarrow \emptyset$;
for each $p \in X$ from left to right and top to bottom **do**
    insert a new marker m' in L with $(m'_y, m'_x) \leftarrow (p_y, L_x)$;
    $\mathit{point}(m') \leftarrow p$;
    $\mathit{link}(p) \leftarrow \mathit{point}(m)$, m next marker below p on L;
    $m \leftarrow$ next marker above p on L;
    if $m = d_u$ **then**
        $A_k \leftarrow \{p\}$; $\mathit{antichain}(m) \leftarrow A_k$; $k \leftarrow k+1$;
    else
        add v to $\mathit{antichain}(m)$;
        add skeleton point (L_x, m_y) to S;
        $\mathit{antichain}(m') \leftarrow \mathit{antichain}(m)$;
        remove m from L;
$v \leftarrow \mathit{point}(m)$, m uppermost marker on L; $C \leftarrow \{v\}$;
while $\mathit{link}(v) \neq$ nil **do**
    $v \leftarrow \mathit{link}(v)$; $C \leftarrow C \cup \{v\}$;
return $S, C, A_0, \ldots, A_{k-1}$;

Figure 6: Algorithm SKELETON

The definition of a jump line as the definition of the canonical antichain partition given by a repeated removal of minimal points may be extended to points in d-dimensional euclidean space with respect to the dominance order. The definition of a shadow remains almost unaltered. The *jump surface* of a point set E is then again defined as the topological boundary of the shadow of E. The layers of the canonical antichain partition are the jump surfaces of repeatedly removed minimal elements. Then Lemma 1 remains true when extended to such layer structure, as is easily seen.

Let us describe a simple well-known algorithm to compute the skeleton, a maximum chain, and the canonical antichain partition of a point set X (see Fig. 6). Since we need this algorithm for points which may have equal x- or y-coordinates, we do *not* assume general position. A sweep line L going from left to right halts at every point of X. It contains an ordered set of markers m. We want to take care of cases where some points of X have the same x- or y-coordinate. Hence, the markers on L are organized in such a way that each marker m is assigned a pair (m_y, m_x) and m is said to be above a point p if $m_y > p_y$ or $m_y = p_y$ and $m_x < p_x$. Moreover, if at a moment there are several points of X on the sweep line they are treated from top to bottom. Conceptually, this amounts to slightly moving points with the same y-coordinate so that they get decreasing y-coordinates with increasing x-coordinates, and points with the same x-coordinate get decreasing x-coordinates with increasing y-coordinates. This shift of points preserves the dominance order on X.

Suppose L halts at some point p and the layers have been constructed for all points to the left of L and all points on L which are above p. Find the layer with its outgoing line above p. If there is no such layer (i.e., the marker found equals the dummy point), open a new one with p as its (yet) sole point. If there is one, add p to this layer and generate a new skeleton point. It is easily seen that the jump lines thus constructed cannot intersect and hence are the layers of a canonical antichain partition, by Lemma 1. Finally, a maximum chain is obtained by extracting a point from each of the antichains along a chain of properly established links. With L implemented as a dynamic binary tree we have

Theorem 2 *Algorithm SKELETON computes the skeleton, the canonical antichain partition, and a maximum chain of a point set X of size n which needs time $O(n \log(n))$ and linear space.* □

The dual definition of the *right skeleton* $S_{\nwarrow}(X)$ may be given using the right shadow and the right jump lines. Of course, we obtain the canonical chain partition instead of the antichain partition. Also, a lemma corresponding to Lemma 1 but dealing with chains instead of antichains is again true. This is seen simply by reflecting the plane on the y-axis. On the other hand, it is easy to obtain similar algorithms for the left and right down skeletons $S_{\searrow}(X)$ and $S_{\swarrow}(X)$ of X. Together with $S_{\swarrow}(X)$ we get an antichain partition, the *canonical right down antichain partition*, which can also be obtained by the repeated removal of maximal points. Together with $S_{\searrow}(X)$ we get a chain partition.

It is convenient to conceive the construction of the skeleton as an operator on finite point sets consisting of points in general position, since the points of the skeleton $S(X)$ again have pairwise different x- and y-coordinates. Thus, we may combine any of the operators $S_{\nearrow}$, $S_{\nwarrow}$, $S_{\searrow}$, and $S_{\swarrow}$. As usual, iteration of an operator O k times will be denoted by O^k; O^0 means identity. $S^k(X)$ will be called the *k-th skeleton* of X. An interesting algebraic property of $S_{\nearrow}$ and $S_{\nwarrow}$, they commute, will be shown in Section 2.3.

2.2 Skeletons and Young Tableaux

One of the properties that seem to lie behind the usefulness of skeletons is the fact that it is possible to reconstruct X from $S(X)$ with a small amount of additional information. Let $x_{\max}$ be the maximal x-coordinate of points in X, and let $y_{\max}$ be defined analogously. Then the *right marginal points* $M_R(X)$ of X are the points $(x_{\max}+1, y_1), \ldots, (x_{\max}+\lambda, y_\lambda)$, where λ is the number of layers of X and $y_1, \ldots, y_\lambda$ are the y-coordinates of the right outgoing lines of the layers ordered increasingly (see Fig. 7). Assuming $x_1, \ldots, x_\lambda$ to be the x-coordinates of the top outgoing lines of the layers in increasing order the *top marginal points* $M_T(X)$ of X are $(x_1, y_{\max}+1), \ldots, (x_\lambda, y_{\max}+\lambda)$ (see Fig. 7).

Theorem 3 *A point set X is the right down skeleton of the skeleton $S(X)$ enhanced by the top and left marginal points of X, i.e., $X = S_{\swarrow}(S(X) \cup M_R(X) \cup M_T(X))$.*

Proof. It is easily seen that the jump line $L(A)$ of an antichain A nearly coincides with the right down jump line $L_1 = L_{\swarrow}(S(A \cup \{s,t\}))$ of the skeleton of A with an arbitrary point s somewhere on the top outgoing line of $L(A)$ and some point t on the right outgoing line (see Fig. 7). They only differ in that L_1 has a left outgoing line to the left of s and a bottom outgoing line below t. Moreover, we have chosen the additional marginal points on the top and right outgoing lines in such a way that the new left and bottom outgoing lines of the right down jump lines cannot intersect any other of these lines. Hence, with Lemma 1 (of course, applied to $L_{\swarrow}$ instead of L) we are done. $\square$

A *partition* of an integer n is a sequence of integers $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{p-1} > 0$ such that $n = \lambda_0 + \cdots \lambda_{p-1}$. Such a partition may be represented graphically by a *Young shape*. This is a shape as that of the two figures in Fig. 9, which consists of p rows of rectangles or *cells* with λ_i cells in row i, when rows are taken from bottom to top (also called “French notation”). If integers are put in these cells in increasing order from left to right and from bottom to top we

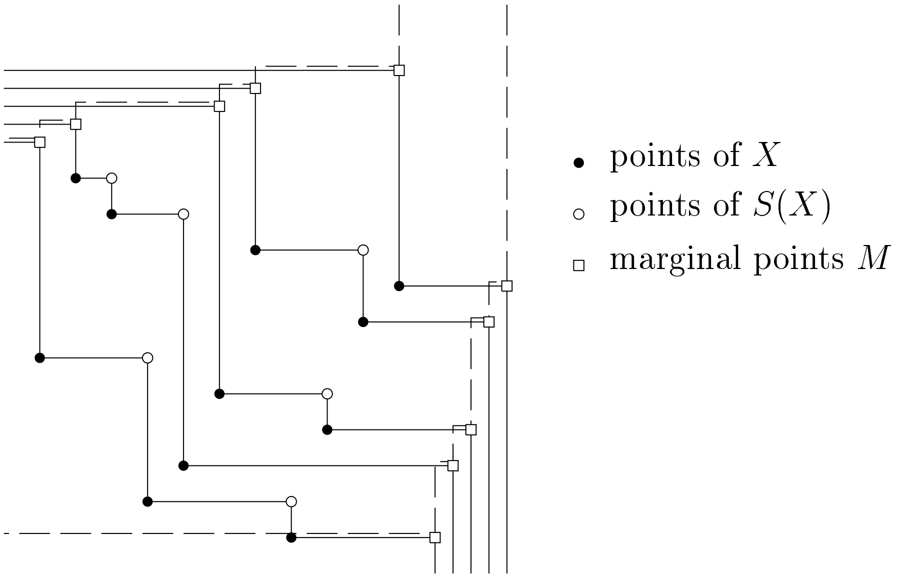

Figure 7: X is the right down skeleton of $S(X) \cup M$.

obtain a *Young tableau*. If Y is a Young tableau we denote the cell in the i-th column from the right and j-th row from below by $Y(i, j)$, with $i, j \geq 0$.

Since we will often refer to the number of layers of X, let us adopt the following notation. Let $p(X)$ be minimal with $S^{p(X)}(X) = \emptyset$. Then $\lambda_i(X)$, $0 \leq i < p(X)$, denotes the number of layers of $S^i(X)$. It is convenient to assume $\lambda_i(X) = 0$ for $i \geq p(X)$.

Lemma 4 *Let X be a point set then $\lambda_0(X) \geq \lambda_1(X) \geq \cdots \geq \lambda_{p-1}(X) > 0$, and $|S^k(X)| = \sum_{k \leq i < p} \lambda_i(X)$, where p is minimal with $S^p(X) = \emptyset$.*

Proof. By Theorem 3, the number of antichains in a minimal antichain partition of $S(X) \cup M_R(X) \cup M_T(X)$ is the same as $\lambda_0(X)$, the size of the canonical antichain partition of X. Hence, $\lambda_1(X)$, the size of a minimal antichain partition of $S(X)$, is at most $\lambda_0(X)$. The same argument shows the other inequalities. The sum over the $\lambda_i(X)$ is computed easily by using $|S(X)| = |X| - \lambda_0(X)$ and induction. □

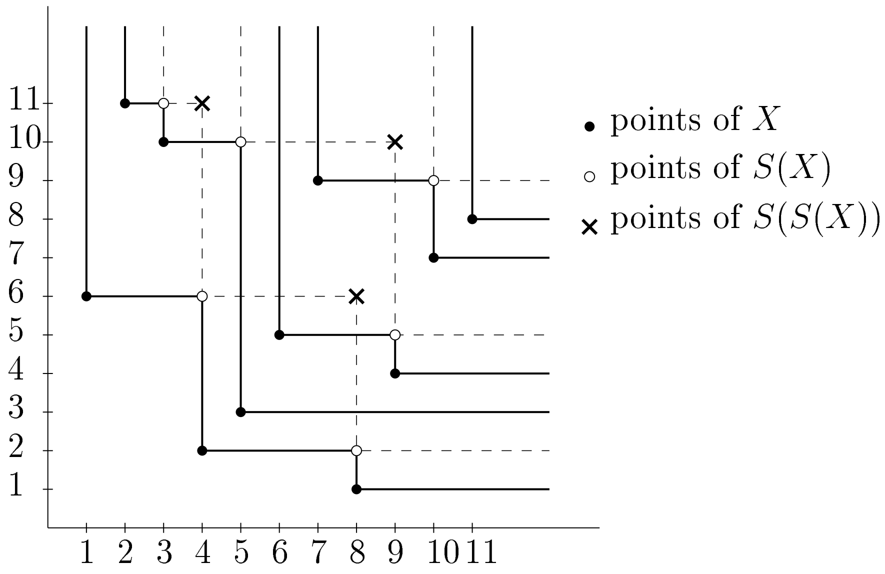

Figure 8: The first two layers $L_1(X)$ and $L_2(X)$ of X.

Let X be a quasi-permutation of n points. We associate two tableaux $P(X)$ and $Q(X)$ (the P- and Q-*symbol of* X) with X in the following way. The k-th row of $P(X)$, $k \geq 0$, are the y-coordinates of the right outgoing lines of $S^k(X)$ in increasing order. The k-th row of $Q(X)$, $k \geq 0$, are the x-coordinates of the top outgoing lines of $S^k(X)$ in increasing order. Compare the outgoing lines of the first two layers of Fig. 8 with the first two rows of the Young tableaux in Fig. 9. According to Lemma 4, $P(X)$ and $Q(X)$ have $\lambda_i(X)$ cells in their i-th row from below and $|X|$ cells altogether. Hence, the shape of the tableaux $P(X)$ and $Q(X)$ is a Young shape. We denote the number of cells in the i-th column (from left) with $\lambda_i^*(X)$. Obviously, $\lambda_0^*(X) \geq \lambda_1^*(X) \geq \cdots \geq \lambda_{l-1}^*(X)$ and $\sum_{0 \leq i < l} \lambda_i^*(X) = |X|$, where $l = \lambda_0(X)$. The $\lambda_i^*(X)$ are the *conjugate partition* of the $\lambda_i(X)$ for the integer $|X|$.

Our first observation about the P- and Q-symbol concerns the *inverse* X^{-1} *of* X, which is the point set that is obtained from X by the transposition $(x, y) \to (y, x)$, i.e., by reflection on the diagonal line $x = y$. Obviously, the corresponding P- and Q-symbols are simply interchanged.

11				
6	10			
2	5	9		
1	3	4	7	8

8				
4	9			
3	5	10		
1	2	6	7	11

Figure 9: The P- and Q-symbol of quasi-permutation X of Fig. 8.

Theorem 5 *For the inverse X^{-1} of a quasi-permutation X, $P(X^{-1}) = Q(X)$ and $Q(X^{-1}) = P(X)$.* $\square$

Theorem 6 (Robinson-Schensted) *The two tableaux $P(X)$ and $Q(X)$ of a quasi-permutation X are Young tableaux. Moreover, for any two Young tableaux P and Q with the same shape there exists a quasi-permutation X with $P = P(X)$ and $Q = Q(X)$, i.e., there is a bijection between quasi-permutations and pairs of Young tableau with the same shape.*

Proof. In Lemma 4, we have already seen that $P(X)$ and $Q(X)$ have the right shape. Also their entries are increasing along the rows, by definition. It remains to see why the right outgoing line of layer $L_j(X)$ lies below that of the corresponding layer $L_j(S(X))$ in the skeleton, i.e., why $P(0,j) \leq P(1,j)$. By applying the same considerations to X^{-1}, we then also have $Q(0,j) \leq Q(1,j)$, and induction implies that $P(X)$ and $Q(X)$ are Young tableaux. But note that a skeleton point of height j in the dominance order of $S(X)$ lies on some layer $L_i(X)$ with $i \geq j$, since any chain of skeleton points intersects a layer of X in at most one point. Hence, $L_j(S(X))$ lies in the shadow of $L_j(X)$ and its right outgoing line must lie above that of $L_j(X)$.

To show the second statement of the theorem suppose by induction that rows $1, \ldots, p-1$ of P and Q determine a quasi-permutation X' with antichains $A_0, \ldots, A_r$ and layers $L(A_0), \ldots, L(A_r)$ with y- and x-coordinates of their right and top outgoing lines corresponding to row 1 of P and Q. Let $p_{\max}$ and $q_{\max}$ be the maxima of all entries in rows 0 of P and Q and let λ be the length of these rows. The points $m_j^R = (p_{\max} + j, P(0,j))$ and $m_j^T = (Q(0,j), q_{\max} + j)$, $0 \leq j < \lambda$, form two chains lying completely to the right of X' and above X', respectively. Denote the set of these points by M. Since $P(0,j) \leq P(1,j)$ and $Q(0,j) \leq Q(1,j)$, we may add m_j^R and m_j^T to A_j obtaining antichains $A'_0, \ldots, A'_r$. With the supplementary antichains $A'_j = \{m_j^T, m_j^R\}$, for $r < j < \lambda$, we obtain an antichain partition of $X' \cup M$. This partition is minimal since its size coincides with the length of the chains in M. Now take $X = S_{\swarrow}(X' \cup M)$. It is easily seen that X' is $S(X)$ and M are just the top and right marginal points of X. Hence, the y- and x-coordinates of the right and top outgoing lines of the canonical layer partition of X are identical to row 0 of P and Q. □

The bijection of Theorem 6 is known as *Robinson-Schensted correspondence* and is usually demonstrated by analyzing the "bumping process" that will not be described here.

If we want to find a quasi-permutation X such that its P-symbol alone coincides with a given Young tableau P, we can do easier. Let the *standard quasi-permutation* X be the set of points $(\sum_{i<k<p} \lambda_k(P) + j, P(i,j))$, for $0 \leq i < p$ and $0 \leq j < \lambda_i(P)$, where p is the number of rows of P, $\lambda_k(P)$ is the length of row k of P, and $P(i,j)$ is the content of cell (i,j) of P. That means, if n is the number of cells in P, we plot n points with increasing x-coordinates from 1 to n and y-coordinates taken from P by reading rows from left to right, taking the rows from top to bottom. It is easily seen that the points corresponding to one column form an antichain and the points corresponding to one row form a chain (recall the properties of a Young tableau). Hence, the right outgoing lines are defined by the points corresponding to the bottom row of P, i.e., $P(X)$ and P coincide on this row. $S(X)$ consists of points $(\sum_{i-1<k<p} \lambda_k(P) + j, P(i,j))$, $i \geq 1$, which stand in the same order relation as the points $(\sum_{i<k<p} \lambda_k(P) + j, P(i,j))$, $i \geq 1$. Hence, by removing the bottom row and by induction, it is easy to see that the tableaux coincide on the other rows too.

2.3 A Theorem on Reversed Permutations

Whereas the effect of reflection of a quasi-permutation X on the diagonal on the corresponding Young tableaux is easy to explain (Theorem 5), reflection on the y-axis, i.e, the transformation that takes (x,y) to $(-x,y)$ (denoted by X^R, the *reversion of* X) has consequences for the Young tableaux that are more difficult to understand. Nevertheless, since an antichain of X becomes a chain in X^R and vice versa the following considerations shed some light on the relation between chains and antichains in X that will be treated in subsequent sections.

Note that by reversion the left shadow of X becomes the right shadow of X^R. Hence, instead of studying the properties of the skeleton of X^R we may also look at the right skeleton of X. To establish the main result of this section we need a theorem that seems to be interesting in itself. But at first, we show a technical lemma.

Lemma 7 *Suppose a finite nonempty sequence of items colored by the four colors red (R), pink (P), black (B), and grey (G) obeys the following rules:*

1. *The items come in pairs of a red or pink item followed by a black or grey one.*
2. *Between any two consecutive black (red) items there is exactly one pink (grey) item and there are no other pink (grey) ones.*

Then the sequence has the form of the regular expression

$$[RG]^* \, RB \, [PB]^* \, \Big[PG \, [RG]^* \, RB \, [PB]^* \Big]^*$$

Proof. Skip all RG pairs (if any) at the beginning of the sequence. Then there must be a RB pair (pink cannot occur before black by 2). Skip all PB pairs (if any). Then the sequence ist either empty or there is a PG pair (red is not possible before grey by 2). Skip all RG pairs (if any) and there must be a RB pair again (pink cannot occur before the next black and the last pink item has to be followed by a black one). Now repeat the arguments following that concerning the PB pair above until all items are seen. □

Theorem 8 *The left and the right skeleton operator commute, i.e., for a point set X $S(S_{\nwarrow}(X))) = S_{\nwarrow}(S(X))$. Furthermore, the right outgoing line of $L_i(X)$ coincides with that of $L_{i-1}(S_{\nwarrow}(X))$, $1 \le i < \lambda_0(X)$, and dually, the left outgoing line of $L_{\nwarrow,j}(X)$ coincides with that of $L_{\nwarrow,j-1}(S(X))$, $1 \le j < \lambda_{\nwarrow,0}(X)$. The right outgoing line of $L_0(X)$ is the prolongation of the left outgoing line of $L_{\nwarrow,0}(X)$.*

Proof. We want to construct the four layer structures of $S(X)$, $S_{\nwarrow}(X)$, $S(S_{\nwarrow}(X))$, and $S_{\nwarrow}(S(X))$ with a single sweep (see Fig. 10). The only sweep

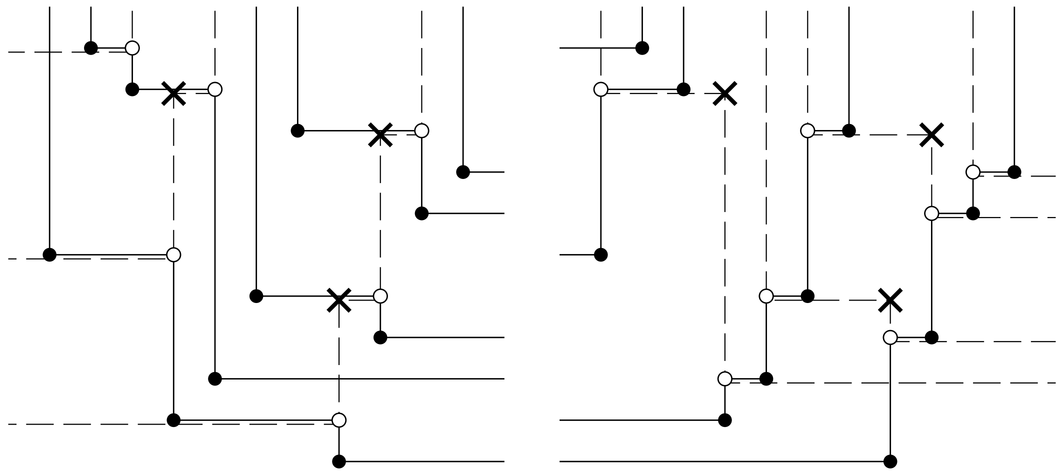

Figure 10: The skeletons $S_{\nwarrow}(S(X))$ (left) and $S(S_{\nwarrow}(X))$ (right) are identical.

line that may be used for these tasks simultaneously is one going from bottom to top (see Section 2.1). To avoid clumsy terminology let us color the left layers and skeleton points $S(X)$ of X with red (political connotations may help to keep overview) and the right layers of $S(X)$ with pink (layers of skeletons are fainter). Whereas the right layers and skeleton point $S_{\nwarrow}(X)$ of X are colored black and the left layers of $S_{\nwarrow}(X)$ are colored grey. Now we want to show by induction that the color pattern of the top outgoing lines is just that described by the regular expression

$$E = [RG]^* \, RB \, [PB]^* \, \Big[PG \, [RG]^* \, RB \, [PB]^* \Big]^*$$

of Lemma 7. In fact, each pair represents two lines with the same x-coordinate. For convenience, we assume that the red or pink line precedes the black or grey one. We show that after the sweep line has encountered a new point with y-coordinate greater than that of all points seen so far, the two conditions of Lemma 7 are again fulfilled and thus the theorem remains true. The reader may find it convenient to use a sheet of paper and colored pencils to check the following details.

With the first point to be inserted a black right layer and a red left layer are opened with a RB pair of top outgoing lines and the two outgoing lines of $L_0(X)$ to the right and $L_{\nwarrow,0}(X)$ to the left on the same y-coordinate. Obviously, all assumptions are fulfilled.

For each of the next points p we consider two cases. At first, assume there is a red outgoing line r next to the right of p_x but no black one to the left. As is seen from expression E, p_x then must lie to the left of the leftmost RB pair. Point p is inserted in the layer of the red line r, which has x-coordinate r_x. We have a new red skeleton point at r_x. Since there is no pink line to the left of the RB pair, this skeleton point gives rise to a new pink layer. Similarly, the point itself gives rise to a new black layer not generating any black skeleton point. Hence, there are no new skeleton points in $S(S_\nwarrow(X))$ or $S_\nwarrow(S(X))$ to consider. Note that at p_x we now have a new RB pair and that the left outgoing line of the two new layers, the black layer of X and the pink layer of $S(X)$ coincide. But, by induction, these are just layers $L_{\nwarrow,j}(X)$ and $L_{\nwarrow,j-1}(S(X))$. Condition 2 of Lemma 7 is fulfilled since the red line r is replaced by that at p_x and no grey line can lie between r_x and p_x (according to E); and the new black line at p_x is followed on the right side by the pink line at r_x. Condition 1 is easily checked, too.

In the second case there is a red outgoing line r at r_x to the right and a black one b at b_x to the left. p is inserted in the layers of these lines generating a black skeleton point at b_x and a red one at r_x which in turn are inserted in the layers of the lines of the single PG pair that must lie between b_x and r_x, according to pattern E. Hence, the PG pair disappears and in its place we have a new point which at the same time belongs to $S(S_\nwarrow(X))$ and $S_\nwarrow(S(X))$. Note that no new left or right outgoing lines are generated in this case. Let us check Condition 2 of Lemma 7 for the red and grey lines (that for black and pink lines is then true by symmetry). At p_x we have a new RB pair. Note that the grey line of the PG pair has disappeared and is replaced by the grey line at b_x as the red line r is replaced by that at p_x. There are possibly some RG pairs between p_x and the grey line at b_x, but then there are grey lines in between and the whole red-grey sequence is as described in Condition 2.

A third case, sc., a black line to the left of p but no red one to the right, is symmetric to the first case. In particular, this is the case where layers $L_i(x)$ and $L_{i-1}(S_\nwarrow(X))$ with the same right outgoing line are generated. $\square$

The *transpose* P^T of a Young tableau P is defined by $P^T(i,j) = P(j,i)$, for each cell (j,i) of P. Hence, the graphical representation of P^T can be obtained by reflecting that of P on the diagonal $x = y$.

The Theorem of Schützenberger now specifies the relation between the P-symbol of X and that of X^R, which in turn are connected to antichain and chain partitions of X. Viennot [32] asks for a simple geometric proof of this relation. We give a proof that uses the geometrical properties of skeletons shown in Theorem 8 (though it probably is not as simple as envisaged by Viennot).

Theorem 9 (Schützenberger) *Let X be a quasi-permutation, then the P-symbol of X^R is the transposed P-symbol of X, $P(X^R) = P(X)^T$.*

Proof. The cells of the k-th row of $P(X)$ contain the y-coordinates of the right outgoing lines of the layers of $S^k(X)$, $0 \leq k < p(X)$. Let this sequence be $A_0(S^k(X))$, ..., $A_{\lambda_k - 1}(S^k(X))$, where $\lambda_k = \lambda_k(X)$ is the length of the k-th row of $P(X)$ (A stands for antichain layer). As discussed above, the l-th row of $P(X^R)$ is given by the y-coordinates of the left outgoing lines of the right layers of $S^l_{\nwarrow}(X)$, $0 \leq l < p(X^R)$. Let this sequence be $C_0(S^l_{\nwarrow}(X))$, ..., $C_{\lambda^*_l - 1}(S^l_{\nwarrow}(X))$, where $\lambda^*_l = \lambda_{\nwarrow,l}(X)$ is the length of the l-th row of $P(X^R)$ (C stands for chain layer). We have to show that the k-th row of $P(X^R)$ coincides with the k-th column of $P(X)$, i.e., $A_l(S^k(X)) = C_k(S^l_{\nwarrow}(X))$.

The second statement of Theorem 8 states that, for a quasi-permutation Y,

$$A_{k-1}(S_{\nwarrow}(Y)) = A_k(Y), \text{ for } 1 \leq k < \lambda_0(Y),$$

hence,

$$A_0(S^k_{\nwarrow}(Y)) = A_k(Y), \text{ for } 0 \leq k < \lambda_0(Y),$$

and dually, $C_0(S^l(Y)) = C_l(Y))$ $(0 \leq l < \lambda_{\nwarrow,0}(Y))$. Furthermore, $A_0(Y) = C_0(Y)$. If we recall the commutativity of S and $S_{\nwarrow}$ (in Theorem 8) we have collected all observations to finish the proof.

$$A_l(S^k(X)) = A_0(S^l_{\nwarrow}(S^k(X))) = C_0(S^k(S^l_{\nwarrow}(X))) = C_k(S^l_{\nwarrow}(X)).$$

□

To find out what happens to the Q-symbol of a quasi-permutation when it is reversed we consider an interesting operation on Young tableaux also due to Schützenberger [27]. In contrast to Schützenberger we provide a thoroughly geometric treatment of this question. Each tableau has some *upper corner* cells, i.e., cells that have no element above or to the right, and has one *lower*

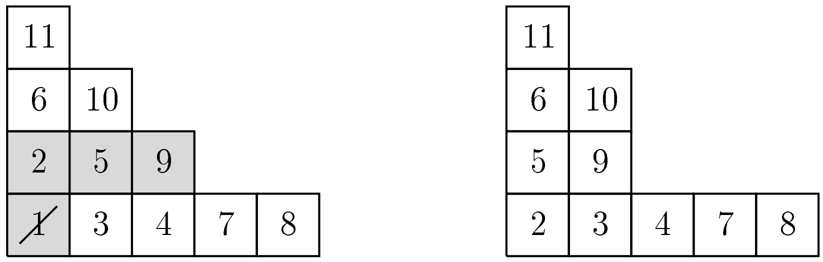

Figure 11: Slide $s(P(X))$ for point set X from Fig. 8.

corner cell, i.e., cell $(0,0)$. The *slide* $s(Y,c)$ *of a Young Tableau* Y *with respect to one of its cells* c is a Young tableau very similar to Y, but with the element contained in c removed. To fill the gap in cell $c=(i,j)$ look at cells $(i+1,j)$ and $(i,j+1)$, i.e., at the cell above and that to the right of c. If both contain elements move the smaller one to c. If only one contains an element, take that. The new gap in one of the two cells is treated similarly until there is no element above or to the right of the cell with the gap, i.e., until it is an upper corner cell. As is easily seen, this process in fact yields a Young tableau. The *slide* $s(Y)$ *of* Y then simply is $s(Y,(0,0))$, the slide with respect to the cell which contains the smallest element (see Fig. 11). $s(Y)$ has a straightforward geometric interpretation.

Lemma 10 *The slide of the* P*-symbol of a quasi-permutation* X *is the* P*-symbol of* $X-m$ *where* m *is the point with the smallest* y*-coordinate in* X*,* $s(P(X)) = P(X-m)$*. Also,* $s(Q(X)) = Q(X-m_l)$*, where* m_l *is the point with the smallest* x*-coordinate in* X*.*

Proof. Let $m=(x,y)$ and $m'=(x',y')$ be the points with smallest and second to smallest y-coordinate, respectively. We have to consider two cases. First, suppose y' is in cell $(0,1)$ above the lower corner cell $(0,0)$ of $P(X)$. Hence, by the definition of $P(X)$, there is no layer of X with right outgoing line at height y' and there must be a point to the right and below m'. But this point can only be m. Thus, when m is removed there is a new right outgoing line at y' and all other right outgoing lines of layers of X remain unchanged. But this means that row 0 of $P(X)$ remains unchanged save that y' replaces y in cell $(0,0)$, the first step in the sliding process, i.e., row 0 of $P(X-m)$ is that of

$s(P(X))$. What concerns the other rows, which just are the P-symbol of the skeleton of X, note that the gap in cell $(0,1)$ amounts to a gap in the lower corner cell of $P(S(X))$ and that with the removal of m from X the skeleton point (x, y') disappears from $S(X)$. Now induction yields that $s(P(S(X)))$ is $P(S(X) - (x, y'))$.

In the second case, y' is in cell $(1,0)$, i.e., in the cell to the left of the corner cell in $P(X)$. We simply switch to the first case by considering $P(X)^T$, which is the P-symbol of X^R by Theorem 9. Since m is the point with minimum y-coordinate in X^R too, and obviously $s(P(X)^T) = s(P(X))^T$, the first statement is shown. The second statement is a reformulation of the first one after the transformation $(x,y) \to (y,x)$. □

Perhaps the reader wants to verify Lemma 10 for Figs. 11 and 8. The *iterated slide* Y^J of a Young tableau Y contains x in cell (i,j) iff, for some k, x is the smallest element of $s^k(Y)$ and cell (i,j) is nonempty in $s^k(Y)$ but becomes empty in $s^{k+1}(Y)$, i.e., the element removed in a slide is put in the cell that becomes free. Let $-T$ denote the tableau obtained from a tableau T (not necessarily a Young tableau) by replacing each element x by $-x$. Of course, Y^J is not a Young tableau, but $-Y^J$ is.

Theorem 11 (Schützenberger) *Let X be a quasi-permutation, then for the Q-symbol of X^R $Q(X^R) = -(Q(X)^J)^T$.*

Proof. Recall that the l-th row of $Q(X^R)$ and $Q(X)$ are the x-coordinates of the top outgoing lines of the layers of $S^l(X^R)$ and $S^l(X)$. The rightmost point (x,y) of X corresponds to the leftmost point $(-x,y)$ in X^R. Hence, by Lemma 10, $Q(X^R - (-x,y))$ is the slide of $Q(X^R)$. Assume that the upper corner cell that becomes empty in the slide of $Q(X^R)$ is cell (i,j). On the other hand, assume we remove point (x,y). Suppose x appears in the l-th row of $Q(X)$. Hence, $S^l(X)$ has a top outgoing line above point (x,y) which disappears with the removal of (x,y) together with all skeleton points above (x,y) of skeletons $S^i(X)$, $0 < i \le l$, if $l > 0$ at all. Since (x,y) was the rightmost point of X, no other top outgoing lines are affected by its removal. Hence, $Q(X)$ differs from $Q(X-(x,y))$ solely in the cell of $Q(X)$ containing x. But since, by Theorems 6 and 9, the shapes of $Q(X)^T$ and $Q(X-(x,y))^T$ are equal to that of $Q(X^R)$ and $Q((X-(x,y))^R)$, respectively, the cell of $Q(X)$ containing x must be cell (j,i). This is the cell getting filled with $-x$ in the iterated slide $-(Q(X))^J$. Hence, cells (i,j) of both, $(-Q(X)^J)^T$ and $Q(X^R)$, contain $-x$. Applying induction to $Q(X^R - (-x,y))$ implies the theorem. □

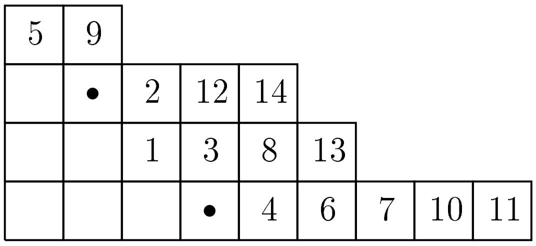

Figure 12: A skew Young tableau with dots indicating lower corner cells.

Consider the shape of a Young tableau Y. One can dismantle this shape by successively removing upper corner cells obtaining Young shapes in each step. Let us call the sequence of Young shapes generated in this way a *dismantling sequence of shapes* for Y.

Corollary 12 *For any given Young shape Y and any given dismantling sequence of shapes $Y = Y_0, \ldots, Y_n$ there is a Young tableau Q such that the shape of the i-th slide $s^i(Q)$ of Q is Y_i, $0 \leq i \leq n$.*

Proof. Let T be a tableau with shape Y and let a cell of T contain index i if the cell is in shape Y_{i-1} but disappears in Y_i, for $1 \leq i \leq n$. Of course, $-T$ is a Young tableau and, by Theorem 6, we find a quasi-permutation X with $Q(X) = -T$. By Theorem 11, $-T = Q(X) = Q((X^R)^R) = -(Q(X^R)^J)^T = -(Q(X^R)^T)^J$. By the definition of the iterated slide, this means that the Young tableau $Q = Q(X^R)^T$ does the job. □

The sliding operation for Young tableaux makes good sense in a more general setting, too. A *skew Young tableau* Z is similar to a Young tableau with the difference that some cells may be empty under the restriction that the empty cells form a Young shape, the *inner shape*, such that each empty cell has another cell of Z above it (not necessarily immediately above it) and to the right of it (not necessarily immediatley to the right of it). Hence, the diagram of Z is connected (see Fig. 12). A slide $s(Z, c)$ is well defined despite the fact that some cells of Z are empty if it starts at some nonempty cell of Z or starts at some upper corner cell c of the inner shape, since then no empty cells are involved. Let such slides be called *lower corner slides of Z*. We may repeat lower corner slides until the skew Young tableau is no longer skew, i.e., its inner tableau is empty. Call this tableau the *redrawing of* Z. Schützenberger [27] shows that the redrawing is a well defined notion. With the concepts developed so far it is easy to give a geometrically inspired proof of this fact.

Theorem 13 *The redrawing of a skew Young tableau* Z *is independent of the sequence of lower corner slides applied to obtain it.*

Proof. A sequence of lower corner slides $s(Z_i, c_i)$, $0 \leq i < n$, with $Z_0 = Z$, skew Young tableaux Z_i, for $1 \leq i < n$, and a Young tableau Z_n, corresponds to a dismantling sequence $Y_0, \ldots, Y_n$ of shapes for the inner shape Y of Z. Let Q be the Young tableau, provided by Corollary 12, whose slides yield the dismantling sequence. Fill the empty cells of the inner shape of Z with the entries of Q to obtain a full Young tableau Z'. Of course, we must diminish the entries of Q by a suitable amount such that they all lie below the smallest entry of Z. Let X be the standard quasi-permutation for Z', i.e., X are the points $(\sum_{k>i} \lambda_k(Z') + j, Z'(i,j))$, for each cell (i,j) of Z', and $P(X) = Z'$ (see Section 2.2). Tableau Q is constructed in such a way that the standard slides $s^1(Z'), \ldots, s^n(Z')$ of Z' correspond to the lower corner slides $s(Z_0, c_0), \ldots, s(Z_{n-1}, c_{n-1})$ of Z, when their effects on cells with entries from Z alone are considered. On the other hand, by Theorem 10, the sequence $s^i(P(X))$, $1 \leq i \leq n$, of slides yields the P-symbol of X with the n lowest points removed. But the n lowest points correspond to the added ones of tableau Q and the remaining points of X correspond to the entries of Z and, as a check of the definition of standard quasi-permutation X reveals, they are always the same regardless of the entries of the inner shape of Z. Hence, the redrawing of Z always is the P-symbol of the subset of points of X with y-coordinates from the entries of Z, independently of the lower corner slides. $\square$

The proof of Theorem 13 indicates how to obtain a standard quasi-permutation X for a skew Young tableau Z, too, such that $P(X)$ is the redrawing of Z. Plot points with increasing x-coordinates, while taking their y-coordinates from the entries of Z, reading each row from left to right, beginning with the topmost row, and skipping empty entries of Z.

Theorem 13 immediately implies that the following *product* $Z_1 * Z_2$ *on two skew Young tableaux* Z_1 *and* Z_2 is associative. If r_2 is the number of rows of Z_2 and c_1 is the number of columns of Z_1 then let $Z_1 * Z_2$ be the redrawing of the skew Young tableau with a rectangle of r_2 rows and c_1 columns of cells as inner shape and Z_1 put above it and Z_2 to the right of it. There even exists a neutral element, the empty Young tableau. Consequently, the product has the following geometric interpretation. If there are two quasi-permutations X_1 and X_2 such that X_2 lies completely to the right of X_1 and points of X_1 have different y-coordinates from that in X_2, then the P-symbol of $X_1 \cup X_2$ is the product of the P-symbols of X_1 and X_2, $P(X_1 \cup X_2) = P(X_1) * P(X_2)$. Hence, for a quasi-permutation $X = \{(x_1, y_2), \ldots, (x_n, y_n)\}$, where the x_i are ordered increasingly, and for single cells Z_i containing y_i, the product $Z_1 * \cdots * Z_n$ is identical to the P-symbol of X.

3 k-Chains and k-Antichains of Point Sets

Greene [13] demonstrates the tight interconnection between maximum k-chains and maximum l-antichains by showing the following theorem on arbitrary orders P.

Theorem 14 (Greene) *For any order R on n elements X there exists a Young shape Y with n cells such that*

1. *the number of cells in the first k rows of Y equals the size of a maximum k-chain, for $1 \leq k \leq n$,*
2. *the number of cells in the first l columns of Y equals the size of a maximum l-antichain, for $1 \leq l \leq n$.*

The proof of the correctness of the construction of maximum k-chains in Section 3.1 allows to derive part (1) of Theorem 14 for quasi-permutations by purely geometrical means (see Corollary 22). It furthermore implies a result of Greene [12] stating that the Young shape mentioned in Theorem 14 is just the shape of the P- and Q-symbol of the quasi-permutation. On the other hand, part (2) of Theorem 14 was deduced by Viennot [32] from his construction of l-antichains (see Section 3.3). Observe that one part can be deduced from the other by using the theorem of Schützenberger (Theorem 9).

Let us shortly describe how a network flow algorithm can be used to compute a maximum weighted k-chain for an arbitrary order R on a weighted set X with n elements and weight $w\colon X \to \mathbf{R}$ in time $O(kn^2)$ and space $O(n^2)$. This is an extension of Gavrils approach [11] due to Sarrafzadeh and Lou [25]. In Section 3.1 we show that at least for dominance orders on unweighted planar points we can do much better computing a k-chain in time $O(kn \log n)$ and linear space.

A network on vertices V (with source s and sink t) connected by arcs E, with capacity K equal to 1 for all arcs and a certain cost function C on them, is constructed as follows. V consists of two vertices v and v' for each element $v \in X$, where v is connected to v' (i.e., $(v, v') \in E$) with cost 0 (i.e., $C(v, v') = 0$). Furthermore, the source s is connected to each v, as each v' is connected to t, both with cost 0. Finally, each u' is connected to v iff $(u, v) \in R$. To compute the cost for the last type of edges at first each $v \in X$ gets a *chain weight* $W(v) = \max\{ W(u) \mid (u, v) \in R \} + w(v)$, i.e., the weight of a maximum weighted chain with v as maximal element. Note that W can be computed by considering all $v \in X$ in any linear ordering of R in time $O(n^2)$. Now the cost of arc (u', v) is defined by $C(u', v) = W(v) - w(v) - W(u)$ (for $(u, v) \in R$), i.e., the loss of weight in taking a chain ending in v that goes through u compared to the maximum possible weight of a chain ending in v. Additionally, $C(s, v) = W(v) - w(v)$ and $C(v', t) = W_{max} - W(v)$, where W_{max} is the maximum of all chain weights $W(v)$, for $v \in X$.

Since any flow entering a vertex $v \in V$ must go through $v' \in V$, too, along the connecting arc of capacity 1, it is clear that a flow of value k defines k vertex disjoint paths in (V, E), which by the transitivity of R correspond to k disjoint chains in (X, R). Conversely, k disjoint chains of (X, R) define a network flow of value k. A simple calculation shows that the weight of a k-chain corresponding to a flow f amounts to $kW_{max} - \text{cost}(f)$, where $\text{cost}(f)$ is the cost of flow f. Hence, by computing a flow f from s to t of value k with minimum cost among all possible flows by an algorithm of Edmonds and Karp [5] in time $O(kn^2)$, we obtain the maximum weighted k-chain of R.

Recall from Section 1 that a simple method to obtain large k-chains in an order is to remove k times a maximum chain from the shrinking set of elements, i.e., a k-chain $\bigcup_{i=1}^{k} M_i$ is obtained by taking maximum chains M_i of $X - \bigcup_{j=1}^{i-1} M_j$. Note that the size of such a k-chain is not independent of the algorithm chosen. Moreover, enhancing the order might cause a greedy algorithm to produce even smaller k-chains (e.g., remove the central point in Fig. 13). For planar points with dominance order such algorithm works in $O(kn \log n)$ time (e.g., by repeating the algorithm of Fig. 6 k times). The size of a k-chain thus obtained may be smaller than that of a maximum k-chain. Hence, let us estimate how bad a result the greedy method can yield.

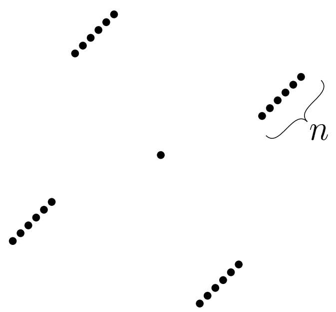

Figure 13: Point set X_n.

Lemma 15 *For any n and $k \geq 2$, there are planar point sets X with $|X| \geq n$ where the size of maximum k-chain exceeds that of a k-chain obtained by the greedy method by at least* $1/5(|X| - \lfloor k/2 \rfloor)$.

Proof. If k is even start with an antichain of size $k/2$. To get the point set X replace each point of the antichain by a copy of the point set X_n of Fig. 13

of size $4n+1$ so that there are no comparabilities besides that in X_n. If k is odd then similarly take $(k-1)/2$ copies and add a small chain of length $n = (|X_n|-1)/4$. It is easily seen that a maximum k-chain differs from an iterated k-chain by one of the four branches in each of the $k/2$ copies of X_n, for k even. Hence, the difference is $1/4(|X| - k/2)$. Similarly, the difference is at least $1/5(|X| - \lfloor k/2 \rfloor)$ in the odd case. □

On the other hand, the difference cannot become too large

Lemma 16 *The size of a maximum k-chain C of a planar point set X exceeds that of a k-chain M obtained by the greedy method by at most* $(1-1/k)^k|C| \leq |C|/e$.

Proof. Let the greedy algorithm remove maximum chains M_i from $X - \bigcup_{j=1}^{i-1} M_j$, for $1 \leq i \leq k$. Set $r_i = |C| - \sum_{j=1}^{i-1} |M_j|$, for $1 \leq i \leq k$, then $r_1 = |C|$ and, by the maximality of C, $r_i \geq 0$. Each $R_{i-1} = C - \bigcup_{j=1}^{i-1} M_j$ can be covered by k chains. Hence, the maximum chain M_i of $X - \bigcup_{j=1}^{i-1} M_j$ has size at least $|R_{i-1}|/k$. Of course, $|R_i| \geq r_i$, consequently, $|M_i| \geq r_{i-1}/k$ and $r_i \leq (1-1/k)r_{i-1}$. Thus, $|C| - |M| = r_k \leq (1-1/k)^k|C|$. □

3.1 Geometric Structure of k-Chains

We now want to propose an algorithm which computes k-chains in planar point sets quickly. The key notion, besides that of the skeleton, will be that of regions for k-chains. Suppose a subset C_S of the skeleton $S(X)$ of a planar point set X is given and let $C_1, \ldots, C_k$ be the canonical chain partition of C_S. The existence of such a chain partition implies that C_S is a k-chain (the converse is false, a k-chain can have a partition into fewer chains). We define the *i-th region of C_S*, for $2 \leq i \leq k$, to be the intersection of the right shadow of C_{i-1} with the complement of the right shadow of C_i, i.e., the region between the jump lines of C_{i-1} and C_i, containing the jump line of C_{i-1} but excluding that of C_i (see Fig. 14).

The *first region* is the complement of the right shadow of C_1 and the *$(k+1)$-st region* is the right shadow of C_k. These $k+1$ regions partition the whole plane.

The notion of regions of a k-chain C_S of $S(P)$ may be slightly generalized. A *chain curve* is a continuous function $c\colon \mathbf{R} \to \mathbf{R}^2$ with the following properties.

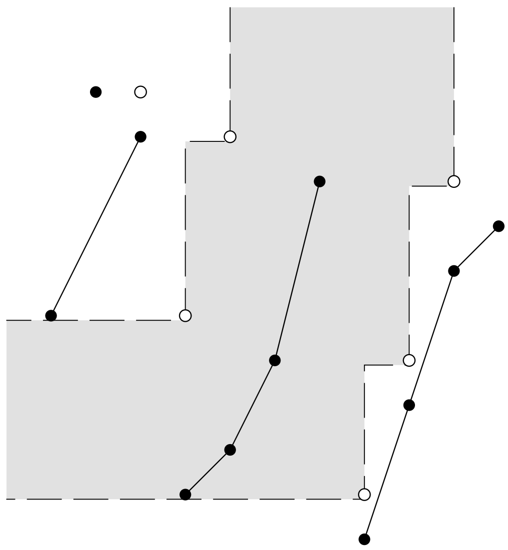

Figure 14: Three regions defined by a maximum 2-chain of the skeleton each containing one chain.

The curve extends from the left below to the right above (i.e., for any two points (x_1, y_1), (x_2, y_2) of c, $x_1 < x_2$ implies $y_1 \leq y_2$). Moreover, c is unbounded in both directions (i.e., for every $D, t \in \mathbf{R}$ there are $t_1 < t < t_2 \in \mathbf{R}$ with the distance of $c(t_1)$ and $c(t_2)$ from $(0,0)$ larger than D). Instead of right jump lines of a canonical chain partition $C_1, \ldots, C_k$ of C_S we now may take k chain curves $c_i\colon \mathbf{R} \to \mathbf{R}^2$, $1 \leq i \leq k$, connecting the points of C_i. When we speak of regions of C_S we tacitly assume that they may be defined by such chain curves.

A description of the main steps of an algorithm computing k-chains can be given with this concept of the region (a more detailed description can be found in Fig. 17). The reader may want to visualize the following steps on Fig. 14.

1. Compute the skeleton $S(X)$ of a planar point set X.
2. Compute recursively a $(k-1)$-chain C_{k-1} of $S(X)$.
3. For all regions R defined by C_{k-1}, extract a maximum chain from X intersected with R. The union of all chains is a maximum k-chain for X.

Note that a minimum chain partition of a k-chain thus computed may

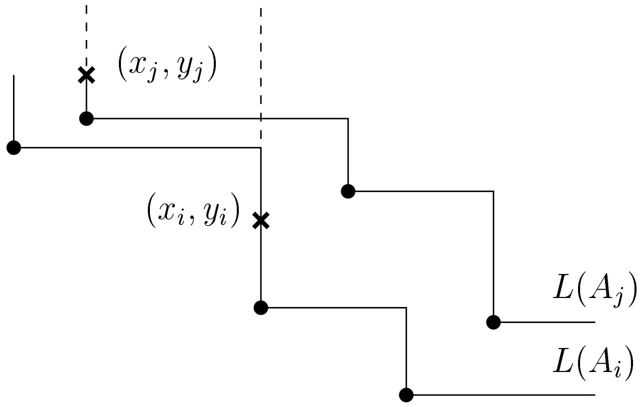

Figure 15: An impossible configuration.

consist of fewer than k chains. To demonstrate the correctness of this approach we need some further definitions. An antichain A of X and its jump line L *enter* a region R *vertically* (*horizontally*) if there is some point of A in R and the leftmost straight line segment of L that lies in R is vertical (horizontal). Similarly, A and L *leave* region R *vertically* (*horizontally*) if $A \cap R$ is nonempty and the rightmost straight line segment of L in R is vertical (horizontal). We are interested in antichains and their jump lines that enter R vertically and leave it horizontally; they are said to *cross R well.* (see Fig. 16).

The next lemma expresses the key property of well crossing antichains that makes them useful for our purposes.

Lemma 17 *Let R be a region of $C_S \subseteq S(X)$ and let $A_1, \ldots, A_\lambda$ be the canonical antichain partition of X. If I is the set of all indices i such that A_i is crossing R well, then the collection $\{A_i \cap R \mid i \in I\}$ is the canonical antichain partition of the underlying point set $\bigcup_{i \in I} A_i \cap R$.*

Proof. By definition, the leftmost line segments of all partial jump lines $L(A_i) \cap R$, $i \in I$, are vertical. Extend these vertical segments to vertical half lines. Similarly, the rightmost horizontal line segments are extended to horizontal half lines. Thus we get a set of jump lines $L(A_i \cap R)$, $i \in I$. Suppose two jump lines $L(A_i \cap R)$ and $L(A_j \cap R)$ with $i < j$ intersect. Since $L(A_j)$ lies in the shadow of $L(A_i)$, one of two outgoing lines of $L(A_i \cap R)$, say the top outgoing line, intersects the jump line $L(A_j \cap R)$. Let the upper bound of region R be a chain curve c. Let the intersection points of c with the top outgoing lines of $L(A_i \cap R)$ and $L(A_j \cap R)$ be (x_i, y_i) and (x_j, y_j), respectively. Since the

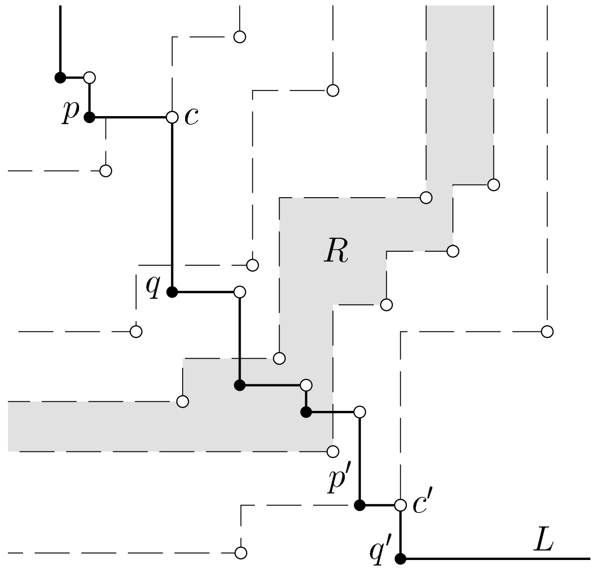

Figure 16: L crosses some region R well in the section between c and c'.

outgoing line of $L(A_i \cap R)$ intersects $L(A_j \cap R)$ we certainly have $x_j < x_i$. On the other hand, (x_i, y_i) and (x_j, y_j) are also points of $L(A_i) \cap c$ and $L(A_j) \cap c$, respectively. Since $L(A_j)$ lies in the shadow of $L(A_i)$, it is easily seen that $x_j < x_i$ implies $y_j > y_i$ (see Fig. 15), in contradiction to the definition of a chain curve. Hence, all $L(A_i \cap R)$, $i \in I$, are disjoint and Lemma 1 implies the lemma. □

Lemma 18 *Let $C_S \subseteq S(X)$ and let A be an antichain of the canonical partition of X. If m is the number of skeleton points on A that are in C_S then the number of regions of C_S crossed well by A is at least $m + 1$.*

Proof. Let c be a point of C_S on the jump line L of A. Let p and q be the points of A to the left and below c that define it (see Fig. 16). Then it is obvious that L leaves the region containing p horizontally and enters that of q vertically. Of course, the top outgoing line of L is vertical as the right outgoing line is horizontal.

Note that if L leaves one region vertically the region of the next point of A to the right is entered vertically too. Now consider the $m + 1$ sections of

L from left to right before, between, and after its m points in C_S (possibly m equals 0). Since in each such section A enters its first region vertically and leaves its last region horizontally, there must be some region crossed well by A in between. □

Lemma 19 *Let C_S be a $(k-1)$-chain in the skeleton $S(X)$ of X and let λ be the height of X. Taking a maximum chain of $X \cap R$ in each region R of C_S yields a k-chain of X of size at least $|C_S| + \lambda$.*

Proof. Let $A_1, \dots, A_\lambda$ be the canonical antichain partition of X. Each point of C_S is a skeleton point of exactly one antichain A_i. If m_i is the number of skeleton points of A_i in C_S, for $1 \leq i \leq \lambda$, then, according to Lemma 18, the antichains A_i cross the regions well in altogether at least $\sum_{1 \leq i \leq \lambda}(m_i + 1) = |C_S| + \lambda$ sections. On the other hand, by Lemma 17, each such section crossing well some region contributes one more point to the maximum chain of that region. □

Note that Lemma 19 does not require the $(k-1)$-chain C_S to be maximum. We now show a kind of reverse to Lemma 19.

Lemma 20 *Let C be a k-chain in X. There exists a $(k-1)$-chain in the skeleton $S(X)$ with size at least $|C| - \lambda$, where λ is the height of X.*

Proof. By Theorem 3, X is the right down skeleton of $S(X)$ enhanced by the top and right marginal points $M_T(X)$ and $M_R(X)$. Note that both chains of marginal points have size λ. Hence, we may reason similarly as in the proof of Lemma 19 after reflection of all points on the diagonal $x = -y$, i.e., after the transformation $T(x, y) = (-x, -y)$ of the plane. The right down skeleton X is thus transformed to a skeleton $T(X)$ of $T(S(X)) \cup T(M_T) \cup T(M_R)$. The image $T(C)$ of the k-chain $C \subseteq X$ is a k-chain in the skeleton $T(X)$ defining $k+1$ regions. Observe that $T(M_R)$ lies in the first region and $T(M_T)$ in the $(k+1)$-st region of $T(C)$ and that both are maximum chains of length λ. We apply Lemma 19 and obtain a $(k+2)$-chain C'_S in $T(S(X)) \cup T(M_T) \cup T(M_R)$ with $|C'_S| \geq |C| + \lambda$. According to the above observation, we may further assume that $T(M_R)$ and $T(M_T)$ are in C_S. When these two chains are removed from C'_S we obtain a $(k-1)$-chain $C_S \subseteq T(S(X))$ of size at least $|C| - \lambda$ and $T(C_S) \subseteq S(X)$ is the $(k-1)$-chain searched for. □

We denote the k-chain in X obtained from a $(k-1)$-chain C_S in $S(X)$ according to Lemma 19 by $\sigma_k(C_S)$.

Theorem 21 *Let C_1 be a maximum chain of $S_{k-1}(X)$ and $C_i = \sigma_i(C_{i-1})$, for $2 \le i \le k$, then C_k is a maximum k-chain in X.*

Proof. By induction, suppose that C_{k-1} is a maximum $(k-1)$-chain in $S(X)$ and let λ be the height of X. If there were a k-chain $C \subseteq X$ with more points than $\sigma_k(C_{k-1})$ then C would have more than $|C_{k-1}| + \lambda$ points, according to Lemma 19. Hence, by Lemma 20, there would be a $(k-1)$-chain of size larger than $|C_{k-1}|$ in $S(X)$, a contradiction. □

Note that the proof of the above theorem also shows that the number of additional points in each application of σ_l to a maximum $(l-1)$-chain of $S_{k-l+1}(X)$ is equal to the height of $S_{k-l}(X)$, for $2 \le l \le k$. Hence, we obtain the following corollary that is part (1) of Greene's Theorem (see Theorem 14) for permutations (recall that $\lambda_i(X)$ is the height of $S^i(X)$).

Corollary 22 *A maximum k-chain of a point set X has size $\sum_{0 \le i < k} \lambda_i(X)$.* □

We are now prepared to provide an algorithm MAXMULTICHAIN (see Fig. 17) that, given a quasi-permutation X and some k, computes a maximum k-chain of X. Some remarks about this algorithm are in order. To dispose X means to release any memory space holding the points of X, which is necessary to keep the space requirement small, as discussed below. As noted above, a maximum $(k-1)$-chain may consist of fewer chains. This happens if C_S equals S and can be partitioned into less than $k-1$ chains. Hence, there may be fewer than k regions of C_S. We assume the regions to be defined by chain curves that consist of the right jump lines of the canonical chain partition of C_S (see Fig. 14) and adopt the convention that R_1 is the whole plane if C_S is empty.

The partitioning of X according to the regions R_i of C_S can be done with a single sweep from left to right halting at every point of X. The sweep line L contains its intersection with all the right layers of C_S and is initialized to the y-coordinates of the left outgoing lines of these layers. Now a point $p \in X$ is easily assigned to its region. If the skeleton point immediately above p is in C_S then the height of the intersection point of the corresponding right layer with the sweep line has to be adapted accordingly.

MAXMULTICHAIN(X, k)
$C_S \leftarrow \emptyset$; $C \leftarrow \emptyset$;
if $k \geq 2$ **then**
    $S \leftarrow$ *skeleton*(X);
    $M \leftarrow$ *marginal points*(X);
    dispose X;
    $C_S \leftarrow$ MAXMULTICHAIN$(S, k-1)$;
    $X \leftarrow$ *right down skeleton*$(S \cup M)$;
$R_1, \ldots, R_l \leftarrow$ *regions*(C_S);
partition X into $X_i \leftarrow X \cap R_i$, $1 \leq i \leq l$;
for $i \leftarrow 1$ **to** l **do**
    $C \leftarrow C \cup$ *maximum chain*(X_i);
return C;

Figure 17: Algorithm MAXMULTICHAIN

Theorem 23 *Algorithm MAXMULTICHAIN(X, k) computes a maximum k-chain for a quasi-permutation X of size n in time $O(kn \log n)$ and linear space.*

Proof. According to Theorem 2, the skeleton, marginal points, and the canonical chain partition of a point set of size n can be computed in $O(n \log n)$ time. As is easily seen, the partitioning of X by a sweep as described above takes the same amount of time. The computation of the maximum chains in each region take, again by Theorem 2, time $O(\sum_{i=1}^{l} |X_i| \log(|X_i|))$ which is $O(n \log n)$. Since these estimations hold true in each recursive step, we have an overall time $O(kn \log n)$.

As far as the space requirement is concerned, the main problem is the computation of a new skeleton in each recursive step. But X is disposed and only its skeleton together with the marginal points is retained. The number of marginal points in the i-th step equals twice the number $\lambda_i(X)$ of layers of the i-th skeleton $S^i(X)$, $0 \leq i \leq k$. Thus, the amount of space that is needed for k recursions is $O(2 \sum_{i=0}^{k} \lambda_i(X) + |S^k(X)|) = O(|X|)$, by Lemma 4. □

3.2 Weighted k-Chains

Given some positive integer weight $w\colon X \to \mathbf{N}$ on the points of a quasi-permutation X, we define the weight of a k-chain as the sum of the weights of its points. A *maximum weighted* k-chain has maximum weight among all weights of k-chains of X. Such maximum weighted k-chains can be found in a similar way as maximum k-chains. Unfortunately, the corresponding algorithm is efficient only if k is small.

In the following we conceptually simulate the weighted case by the unweighted one. Though the algorithms work on the weighted point set itself the proofs of correctness are based on the following idea. We replace each point $p \in X$ by a tiny chain $C(p)$ of $w(p)$ points, where tiny means that the chain is contained within a small box of sidelength ϵ and ϵ is much smaller than the minimum distance in x- or y-coordinates of the points of X (of course, here we assume all points to have different x- and y-coordinates). Denote this set of new points by X'. Now consider the skeleton $S(X')$ of X'. Let p, q be two points of X with $p_x < q_x$. It is easily seen that if there are any skeleton points in $S(X')$ between points of the tiny chain $C(p) \subseteq X'$ and that of the other chain $C(q) \subseteq X'$ then they again fit into a small box B of sidelength ϵ. In this case, locate a *weighted skeleton point* between p and q (i.e., at (p_x, q_y)) and give it a weight equal to the number of skeleton points contained in box B. The resulting set of weighted skeleton points is the *weighted skeleton* $S(X, w)$.

It is easy to obtain the weighted skeleton $S(X, w)$ without resorting to set X' of multiplied points. As in the case of unweighted points, a sweep line L goes from left to right and halts at every point p of X. We have to translate the actions of Algorithm SKELETON (see Fig. 6 of Section 2.1) that constructs the skeleton $S(X')$ of X' into actions of an Algorithm SKELETON-WEIGHTED (see Fig. 18) that constructs the corresponding weighted skeleton $S(X, w)$ of X. The sweep line L of Algorithm SKELETON contains the y-coordinates of the right outgoing lines of the layer structure of X'. If there are some, say $W(m)$, right outgoing lines differing in their y-coordinates only by an amount smaller than ϵ then their collection corresponds to a *weighted outgoing line* on L at marker m with weight $W(m)$. Now the insertion of a point $p \in X$ with weight $w(p)$ in L corresponds to $w(p)$ successive insertions of all $w(p)$ points of the tiny chain $C(p) \subseteq X'$. Let m be the next marker above p. If $|C(p)|$ is greater or equal to the number $W(m)$ of outgoing lines on L then $W(m)$ new skeleton points in $S(X')$ are generated, i.e., a new skeleton point in $S(X, w)$ of

SKELETON-WEIGHTED(X, w)
insert dummy d_u in L with $W(d_u) \leftarrow +\infty$ at height $+\infty$;
$link(d_u) \leftarrow$ nil;
$S \leftarrow \emptyset$;
for each $p \in X$ from left to right and top to bottom **do**
    insert a new marker m' in L with $(m'_y, m'_x) \leftarrow (p_y, L_x)$;
    $W(m') \leftarrow w(p)$; $point(m') \leftarrow p$;
    $link(p) \leftarrow point(m)$, m next marker below p on L;
    $m \leftarrow$ next marker above p on L;
    $W \leftarrow w(p)$;
    while $W \geq W(m)$ **do**
        add skeleton point (L_x, m_y) with weight $W(m)$ to S;
        $W \leftarrow W - W(m)$;
        remove m from L;
        $m \leftarrow$ next marker above p on L;
    if $m \neq d_u$ and $W > 0$ **then**
        add skeleton point (L_x, m_y) with weight W to S;
        $W(m) \leftarrow W(m) - W$;
$v \leftarrow point(m)$, m uppermost marker on L; $C \leftarrow \{v\}$;
while $link(v) \neq$ nil **do**
    $v \leftarrow link(v)$; $C \leftarrow C \cup \{v\}$;
return S, C;

Figure 18: Algorithm SKELETON-WEIGHTED

weight $W(m)$ is generated and marker m removed. If $|C(p)| > W(m)$ there are still $W = |C(p)| - W(m)$ points of $C(p)$ to be inserted. Hence, the next mark on L is searched and the procedure is repeated until there is no further marker on L or there is one marker m_0 with $W(m_0) > W$. With these considerations in mind it is easy to verify that Algorithm SKELETON-WEIGHTED yields the weighted skeleton of (X, w).

The weighted skeleton thus computed has many points with equal x- or y-coordinate and we want to compute the weighted skeleton of a skeleton repeatedly. Hence, care must be taken of cases where some points have the same x- or y-coordinate. As in Algorithm SKELETON the sweep line is organized in a way to cope with this problem (see the description of this algorithm for more details). For the following arguments, we assume that the points of X have general position.

It is easily seen that the construction of the weighted skeleton $S(X, w)$ takes time $O((|X| + |S(X, w)|) \log(|X| + |S(X, w)|))$. Unfortunately, the number of skeleton points thus computed can be larger than that of X, but we have

Lemma 24 *The number of weighted skeleton points $S(X, w)$ is at most twice the number of points in X.*

Proof. If a skeleton point $s \in S(X, w)$ has no other skeleton point s' above it is assigned to the (unique) point $p \in X$ below it. Otherwise, it is assign to the (unique) point to the left of it. Since, as is easily seen by inspecting Algorithm SKELETON-WEIGHTED, a skeleton point either has no skeleton point to the right of it (i.e., it is generated at marker m under the condition $W \geq W(m)$) or it has no skeleton point above it (i.e., it is generated at marker m under the condition $0 < W < W(m)$), no point $p \in X$ gets assigned more than two skeleton points. □

The interesting fact now is that Algorithm MAXMULTICHAIN (see Fig. 17 of Section 3.1) may be used nearly without changes to compute a maximum weighted k-chain. There are only the following differences. We do not dispose any set of points, since the space requirement will be high anyway, as will be seen below. We further do not assume that the points of a $(k-1)$-chain in the skeleton are in general position. Hence, the right layers defining regions may intersect (but not cross). To construct such layers now simply use Algorithm SKELETON-WEIGHTED. The partitioning of X according to the regions is achieved as in Algorithm MAXMULTICHAIN.

Observe that either all or none of the points of $C(p)$, for some $p \in X$, are contained in a maximum chain in X'. Thus, a maximum chain in X' corresponds to a maximum weighted chain in X and vice versa. This fact is the reason that the chain returned by Algorithm SKELETON-WEIGHTED is maximum weighted chain, as is easily checked. The same holds true of maximum weighted k-chains in X and maximum k-chains in X'.

Theorem 25 *Let C_S be a maximum weighted $(k-1)$-chain in $S(X, w)$. Then taking a maximum weighted chain of points $X \cap R$ in each of the k regions R of C_S yields a maximum weighted k-chain of X.*

Proof. A maximum weighted $(k-1)$-chain in C_S corresponds to a maximum $(k-1)$-chain in some subset C'_S of the skeleton $S(X')$ of the set X' of multiplied points. The selection of a maximum chain in the point sets $X' \cap R$ for each region R of the k regions of C'_S would give a maximum k-chain of X'. This does not immediately yield a maximum weighted k-chain of X since some points of the chain $C(p) \subseteq X'$ replacing a point p of X may fall in one region and some in another. Since there is no skeleton point near $C(p)$ it is separated either by a vertical or horizontal segment of a jump line of some chain. But it cannot be vertical for there is no skeleton point below $C(p)$. On the other hand, any horizontal segment may be detoured along the right and the lower side of an aligned box of sidelength ϵ around $C(p)$. Since there is no skeleton point to the left of the box within its y-extension or in the box itself, the paths we thus obtain are a chain curves. Hence, according to the observation above, taking maximum chains from each of the k regions R'_i defined by these paths yields a maximum k-chain C'_k in X'. But, as is quickly verified, there is a correspondence between the regions R'_i of C'_S and the regions R_i of C_S, for $1 \leq i \leq k$, in such a way that a chain $C(p)$ is completely contained in R'_i iff p is contained in R_i. Consequently, all maximum weighted chains from regions R_i together form a maximum weighted k-chain. □

In analogy to the unweighted case, we set $S^0_w(X) = X$ and define $S^k_w(X) = S(S^{k-1}_w(X), w)$ to be the *k-th weighted skeleton* of X. We denote the weighted k-chain in X obtained from a weighted $(k-1)$-chain C_S in $S(X, w)$ according to Theorem 25 by $\rho_k(C_S)$. As noted above, a maximum weighted 1-chain may be computed by a single run of algorithm SKELETON-WEIGHTED.

Theorem 26 *Let C_1 be a maximum weighted chain of $S^{k-1}_w(X)$ and $C_l = \rho_l(C_{l-1})$, for $2 \leq l \leq k$, then C_k is a maximum weighted k-chain in X.* □

Theorem 27 *A maximum weighted k-chain of a point set X can be obtained in $O(2^k|X|\log(2^k|X|))$ time and $O(2^k|X|)$ space.*

Proof. As was already pointed out, to construct the weighted skeleton for a weighted point set X takes time $O(|X|\log|X|)$. The point location of points X in the regions defined by a $(k-1)$-chain C_S can be done with the help of a sweep line in time $O(|X|\log|C_S| + |C_S|))$. This amounts to a total running time of $O(\sum_{1\leq i\leq k} 2^i|X|\log(2^i|X|))$, since $|S^i_w| \leq 2^i|X|$ by Lemma 24. The bound on the space is given by $O(\sum_{1\leq i\leq k} 2^i|X|)$. □

The number of layers of a canonical antichain partition played an important role in the proof of Theorem 21, the unweighted counterpart of Theorem 25. This number corresponds to the sum of weights of outgoing lines after Algorithm SKELETON-WEIGHTED has computed the weighted skeleton. Let this weight be denoted by $\lambda_w(X)$. By recalling the fact that a maximum chain in the underlying point set X' corresponds to a maximum weighted chain in X the following theorem is a restatement of Lemmas 19 and 20 for the weighted case.

Theorem 28 *Let $\lambda_w(X)$ be the sum of weights of the right outgoing lines of the weighted skeleton $S(X,w)$. If C_S is a weighted $(k-1)$-chain in $S(X,w)$ of X then there exists a weighted k-chain of X of weight at least $|C_S| + \lambda_w(X)$. Conversely, if C is a weighted k-chain in X there exists a weighted $(k-1)$-chain in $S(X,w)$ with weight $S(X,w) \geq |C| - \lambda_w(X)$.* □

3.3 Geometric Structure of k-Antichains

In this section we prove some assertions made by Viennot [32] leading to an algorithm that efficiently computes k-antichains of a point set X if k is large.

Let A_S be a k-antichain in the skeleton $S(X)$ and let $A_{S,1}, \ldots, A_{S,k}$ be the canonical antichain partition of A_S. It is easily seen that the intersection of X with the right down jump line $L_{\swarrow}(A_{S,i})$ is an antichain in X. Let us denote the k-antichain $\bigcup_{i=1}^{k} X \cap L_{\swarrow}(A_{S,i})$ of X by $\Delta(A_S)$ (see Fig. 19). On the other hand, given a k-antichain A with canonical partition $A_1, \ldots, A_k$ we define the k-antichain $\delta(A) = \bigcup_{i=1}^{k} S(X) \cap L(A_i)$ of $S(X)$. Recall that a strict k-antichain is one that cannot be covered by less than k antichains.

Lemma 29 *Let X be a planar point set.*

1. *If A_S is a strict k-antichain of $S(X)$ then $\Delta(A_S)$ is a k-antichain of X of size at least $|A_S| + k$.*
2. *If A is a strict k-antichain of X then $\delta(A)$ is a k-antichain of $S(X)$ of size at least $|A| - k$.*
3. *If A_S is a strict maximum k-antichain then equality holds in 1 and $\Delta(A_S)$ is a strict maximum k-antichain too.*

Proof. Let $A_{S,1}, \ldots, A_{S,k}$ be the canonical right down antichain partition of A_S. Each skeleton point $s \in S(X)$ has two defining points $p_L(s), p_D(s) \in X$,

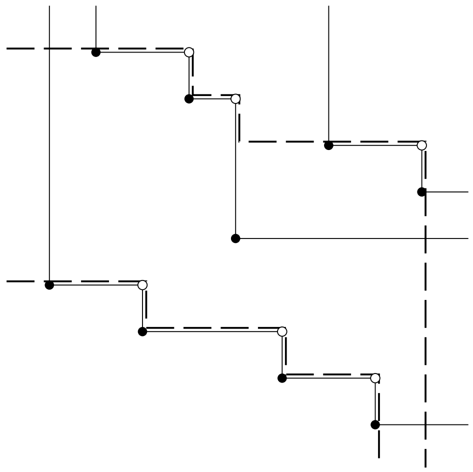

Figure 19: The Δ-operator.

one with the same y-coordinate to the left, the other with the same x-coordinate below. If we walk along a right down jump line $L_{\swarrow}(A_{S,i})$ from left to right we find between any two consecutive skeleton points s_1, s_2 of $A_{S,i}$ at least one of the defining points $p_D(s_1)$ or $p_L(s_2)$ on the jump line. Otherwise, the defining point $p_D(s_1)$ would have y-coordinate smaller than that of s_2 and point $p_L(s_2)$ would have x-coordinate smaller than that of s_1. But this implies that two layers of the canonical layer structure of X intersect, which is impossible (see Fig. 20). Since the left and down defining point $p_L(s_L)$ and $p_D(s_R)$ of the leftmost and rightmost skeleton points s_L and s_R of $A_{S,i}$ are always on the right down jump line, $X \cap L_{\swarrow}(A_{S,i})$ contains at least one point more than $A_{S,i}$. Summing over all $A_{S,i}$ we get the first inequality of the lemma.

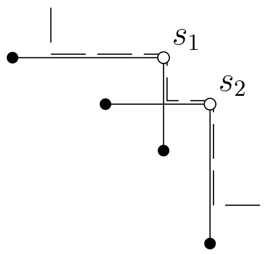

Figure 20: Intersecting layers.

The second inequality is obtained similarly. One may extend X by the top and right marginal points M_T and M_R and use the transformation $T(x, y) = (-x, -y)$. By Theorem 3, $T(X)$ is the skeleton of $T(S(X)) \cup T(M_T) \cup T(M_R)$. Hence, by the above argument, $|\delta(A)| \geq |\Delta(T(A))| - 2k \geq |T(A)| - k$ since the two chains $T(M_T)$ and $T(M_R)$ contribute at most $2k$ points to the k-antichain

$\Delta(T(A))$ in $T(S(X)) \cup T(M_T) \cup T(M_R)$.

Suppose A_S is a strict maximum k-antichain and let $\Delta(A_S)$ be a strict k'-antichain with $k' \leq k$. Then $\delta(\Delta(A_S))$ is a k'-antichain, hence k-antichain, of size at least $|\Delta(A_S)| - k' \geq |A_S| + k - k'$ in $S(X)$ and $k' = k$ since A_S is maximum. Consequently, if A' is a k-antichain of X, $|\Delta(A_S)| - k \geq |A_S| \geq \delta(A') \geq |A'| - k$, which implies that $\Delta(A_S)$ is a maximum k-antichain. □

Theorem 30 *Let X be a point set and let $k \leq \lambda_0(X)$. There is an l with $0 \leq l < p(X)$ and $\lambda_{l+1}(X) \leq k < \lambda_l(X)$. The l-th skeleton $S^l(X)$ contains a strict maximum k-antichain A_l and $\Delta^l(A_l)$ is a maximum k-antichain of X of size $|S^{l+1}(X)| + kl = \sum_{i=0}^{k} \lambda_i^*(X)$.*

Proof. Since, according to Lemma 4, the $\lambda_i(X)$ are decreasing in i and $\lambda_{p(X)}(X) = 0$, there is an l such that the inequalities are satisfied. $S^{l+1}(X)$ itself is a strict maximum $\lambda_{l+1}(X)$-antichain and $\Delta(S^{l+1}(X))$ is a strict maximum $\lambda_{l+1}(X)$-antichain of $S^l(X)$ of size $|S^{l+1}(X)| + \lambda_{l+1}(X)$, by Lemma 29. Take a maximum chain C in $S^l(X)$; it has size $\lambda_l(X)$. Since $\Delta(S^{l+1}(X))$ intersects C in at most $\lambda_{l+1}(X)$ points and $S^l(X)$ has size $|S^{l+1}(X)| + \lambda_l(X)$, there is no point of $S^l(X)$ outside $C \cup \Delta(S^{l+1}(X))$ and we may add $k - \lambda_{l+1}(X)$ arbitrary points of $S^l(X)$ to $\Delta(S^{l+1}(X))$ to get a strict maximum k-antichain A_l in $S^l(X)$ of size $|S^{l+1}(X)| + k$. Now induction and application of Lemma 29 shows the theorem. □

An algorithm MAXMULTIANTICHAIN (see Fig. 21) computing a maximum k-antichain of X for given X and k along the lines of the proof of Theorem 30 is now easy to provide.

Theorem 31 *Algorithm MAXMULTIANTICHAIN(X, k) computes a maximum k-antichain for a point set X of size n in time $O((n^2/k)\log n)$ and linear space.*

Proof. Since the algorithm simply mimics the proof of Theorem 30, it certainly computes a maximum k-antichain. The computation of the skeleton, marginal points, and the number of layers takes time $O(n \log n)$. The Δ operator is implemented straightforwardly. For a sweep line from right to left computing the right down layers of A_S may halt at points of X, too, and check whether they lie on a layer or not. Hence, the time needed for one recursive step is $O(n \log n)$. According to Theorem 30, a maximum k-antichain has size

MAXMULTIANTICHAIN(X, k)
$\lambda \leftarrow$ number of layers of (X);
if $\lambda \leq k$ **then**
  return (X, λ);
else
  $S \leftarrow$ *skeleton*(X);
  $M \leftarrow$ *marginal points*(X);
  dispose X;
  $(A_S, \lambda') \leftarrow$ MAXMULTIANTICHAIN(S, k);
  $X \leftarrow$ *right down skeleton*$(S \cup M)$;
  $A \leftarrow \Delta(A_S)$;
  if $\lambda' < k$ **then**
    add $k - \lambda'$ arbitrary points of X to A;
  return (A, k);

Figure 21: Algorithm MAXMULTIANTICHAIN

$|S^{l+1}| + kl \leq n$. Thus, the number $l+1$ of recursions is bounded by $n/k+1$. That the amount of space needed remains linear is seen as in the proof of Theorem 23. □

3.4 The Expected Size of Maximum k-Chains

What is the expected size of a k-chain in a random permutation taken from from all permutations of length n with equal distribution. Hammersley [14] showed that the expected length of a maximum increasing subsequence of such a permutation converges to $c\sqrt{n}$ with increasing n, for some constant c. Numerical calculations suggested that $c = 2$, which was subsequently proved by Logan and Shepp [17] ($c \geq 2$) (a result also claimed by Veršik and Kerov [31] but not thoroughly proven). A simple proof for $c \leq 2$ is given by Pilpel [24]. We recall this proof (slightly simplified) below.

The idea in the work of Logan and Shepp and that of Veršik and Kerov is to find a function approximating the shape of the Young tableaux of most permutations in the Robinson-Schensted correspondence. It is then shown that the number of permutations with a Young shape differing from that typical shape in a certain metric decreases rapidly. Moreover, we know from Corollary 22

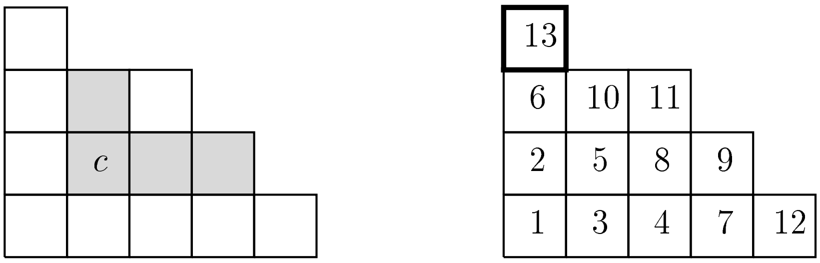

Figure 22: Cell c has hooklength 4. P-symbol of permutation π with $Z_{13}(\pi)$.

that the size of a maximum k-chain in a permutation can be computed by summing up the lengths of the first k columns of the Young shape of the corresponding P- or Q-symbol. This facts may be used to bound the expected size not only of maximum chains in permutations but also of maximum k-chains, though some additional technical details have to be settled.

Let Λ_n be the set of all Young shapes of size n and let $\lambda \in \Lambda_n$. Observe that we may look at λ as a piecewise continuous function from nonnegative real values $\mathbf{R}^+$ to $\mathbf{R}^+$. As noted above, the size of maximum k-chains in permutations with Young shape λ is given by $\sum_{0 \leq i < k} \lambda(i)$. The *hook length* h_c of cell c of λ is the number of cells above and to the right of c including c once (see Fig. 22). Then the number of Young tableaux with shape λ is given by the classical Frame, Robinson, Thrall Formula [8], ${n!}/{\prod_{c \text{ cell in } \lambda} h_c}$. Hence, by the Robinson Schensted correspondence, the probability that shape λ is the shape of the P- and Q-symbol of a random permutation is $n! \Big/ \left(\prod_{c \text{ cell in } \lambda} h_c\right)^2$ which for the rest of this section defines the *Plancherel-Schensted distribution* on Λ_n.

The next result is due to Pilpel [24]. We recall the slightly modified proof for completeness.

Lemma 32 *For $\lambda \in \Lambda_n$, the expected value of $\lambda(0)$ is at most $2\sqrt{n}$.*

Proof. Let $l_n(\pi)$ denote the size of a maximum antichain of permutation π, where π is from the set S_n of permutations of length n, and let l_n denote the corresponding random variable. Recall that $\lambda_i^*(\pi)$ denotes the height of the i-th column of $P(\pi)$. We know from Theorem 30 that $l_n(\pi)$ equals $\lambda_0^*(\pi)$. Consider the following event (see also the right part of Fig. 22) .

$Z_n(\pi)$ denotes the event that n appears in the first column of $P(\pi)$, i.e., by the properties of a Young tableau, n is in the single top cell of $P(\pi)$.

Furthermore, let $\pi^* \in S_{n-1}$ be the permutation obtained from $\pi \in S_n$ by deleting n in π. Note that each permutation of S_{n-1} can be obtained from exactly n permutations of S_n in this way.

If X is the point set corresponding to π, a point set X^* corresponding to π^* can be obtained from X by removing the highest point p_n, i.e., that with y-coordinate n, and decreasing the x-coordinates of all points to the right of p_n by one. It is obvious that the right outgoing lines of all skeletons $S^i(X)$ remain the same save the right outgoing line at height n (in some $S^i(X)$, for a similar observation see Theorem 11). By the definition of the P-Symbol, this implies that the P-Symbol of π^* can be obtained from that of π by deleting the cell with entry n. Hence, we have the following recursion

$$l_n(\pi) = \begin{cases} l_{n-1}(\pi^*) + 1 & \text{if } Z_n(\pi), \\ l_{n-1}(\pi^*) & \text{if not } Z_n(\pi). \end{cases}$$

Consequently we have

$$\begin{aligned} \mathbf{E}\, l_n &= \frac{1}{n!} \sum_{\pi \in S_n} l_n(\pi) \\ &= \frac{1}{n!}\left(\sum_{\pi \in S_n:\ Z_n(\pi)} (l_{n-1}(\pi^*) + 1) \quad + \sum_{\pi \in S_n:\ \text{not } Z_n(\pi)} l_{n-1}(\pi^*) \right) \\ &= \frac{1}{n!}\left(n \sum_{\pi \in S_{n-1}} l_{n-1}(\pi) + \sum_{\pi \in S_n:\ Z_n(\pi)} 1 \right) = \mathbf{E}\, l_{n-1} + \mathbf{P}[Z_n], \end{aligned}$$

which implies $\mathbf{E}\, l_n = \sum_{i=1}^n \mathbf{P}[Z_i]$.

If $\lambda \in \Lambda_{n-1}$ then let the Young shape $\lambda^+ \in \Lambda_n$ be obtained from λ by adding a single top cell, i.e., $\lambda^+(0) = \lambda(0) + 1$. By the Robinson-Schensted correspondence, a permutation π with $Z_n(\pi)$ is related to a P-Symbol of shape λ^+ with n in the top cell and some Q-Symbol of shape λ^+, for some $\lambda \in \Lambda_{n-1}$. Consequently, if $Y(\lambda)$ denotes the number of Young tableaux of shape λ, we have

$$\mathbf{P}[Z_n] = \frac{1}{n!} \sum_{\lambda \in \Lambda_{n-1}} Y(\lambda) Y(\lambda^+). \tag{1}$$

On the other hand, the Robinson Schensted correspondence means that

$$\begin{aligned}(n-1)! &= \sum_{\lambda\in\Lambda_{n-1}} Y(\lambda)^2, \text{ i.e., } n! = \sum_{\lambda\in\Lambda_{n-1}} (\sqrt{n}\,Y(\lambda))^2, \\ n! &= \sum_{\lambda\in\Lambda_n} Y(\lambda)^2 \geq \sum_{\lambda\in\Lambda_{n-1}} Y(\lambda^+)^2.\end{aligned}$$

By the Cauchy-Schwarz inequality, this yields $n! \geq \sum_{\lambda\in\Lambda_{n-1}} \sqrt{n}\,Y(\lambda)Y(\lambda^+)$ and combined with Equation 1 finally we get $\mathbf{P}[Z_n] \leq 1/\sqrt{n}$ and

$$\mathbf{E}\, l_n = \sum_{i=1}^{n} \frac{1}{\sqrt{i}} \leq \int_0^n \frac{1}{\sqrt{x}}\, dx = 2\sqrt{n}.$$

□

We next cite a claim of Veršik and Kerov [31] that can be used to deduce bounds on the expected size of k-antichanins (a proof may be found in the appendix). First we make some assumptions on λ that simplify the following considerations. We think of λ as a differentiable, monotone decreasing (hence invertible) function $\lambda(x)$ from $\mathbf{R}$ in $\mathbf{R}$. If we start with the border of the Young shape λ this my be achieved by slightly inclining the line segments and smoothing the edges in such a way that with arbitrarily small error the area below $\lambda(x)$ again has value n.

Now we scale $\lambda(x)$ such that it has integral unity, i.e., we define $f_\lambda(x) = (1/\sqrt{n})\, \lambda(\sqrt{n}x)$. Next let the graph of h_λ be that of f_λ scaled by $\sqrt{2}$ and rotated by $\pi/4$ around the origin, i.e., it is obtained by the transformation ρ: $(x, y) \to (x - y, x + y)$ of the plane (see Fig. 23). The following theorem says that there exists a kind of typical form of a Young shape under the Plancherel-Schensted distribution if n is large enough.

Theorem 33 *Let h_λ be defined as above and let $\epsilon > 0$, then*

$$\lim_{n\to\infty} \mathbf{P}\Big[\lambda \in \Lambda_n\colon\ \sup_{x\in\mathbf{R}} |h_\lambda(x) - h_0(x)| > \epsilon\Big] = 0$$

where

$$h_0(x) = \max\Big\{\, \frac{x}{\pi} \arcsin \frac{x}{2} + \sqrt{1 - \frac{x^2}{4}}, \frac{|x|}{2} \,\Big\} \text{ for } x \in \mathbf{R}.$$

Proof. See Appendix. □

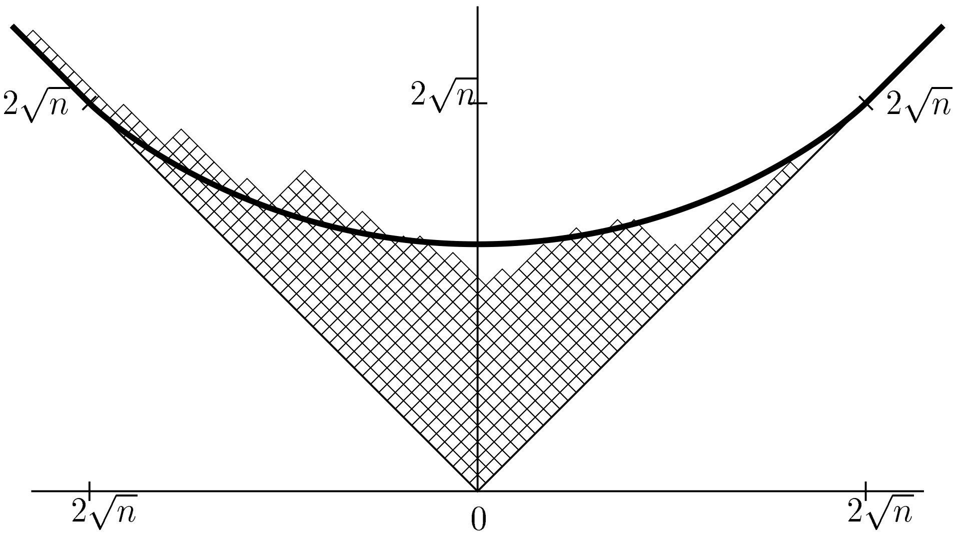

Figure 23: Form τ_n of typical Young tableaux.

The transformation that is inverse to ρ is $\rho^{-1}(u,v) = ((u+v)/2, (v-u)/2)$ and thus the image of the graph of f under ρ is the graph of a function $h = \Omega(f)$ with

$$f(x) = \frac{h(s)-s}{2} \qquad x = \frac{h(s)+s}{2}. \tag{2}$$

Let τ_n be the function approximating the "typical" Young tableau of size n, i.e., $\tau_n(x) = \sqrt{n} f_0(x/\sqrt{n})$, $x \geq 0$, where $f_0 = \Omega^{-1}(h_0)$. In the appendix it is shown that $\int_0^\infty f_0(x)\,dx = 1$ and thus $\int_0^\infty \tau_n(x)\,dx = n$. Unfortunately, Theorem 33 does not immediately imply a similar uniform convergence in probability for the untransformed shape λ. Instead we show the following lemma.

Lemma 34 *For any $\epsilon' > 0$,*

$$\lim_{n\to\infty} \mathbf{P}\Big[\lambda \in \Lambda_n\colon \sup_{x\geq 0}\Big\{\tau_n(x) - \lambda(x)\Big\} > \epsilon'\sqrt{n}\Big] = 0, \tag{3}$$

$$\lim_{n\to\infty} \mathbf{P}\Big[\lambda \in \Lambda_n\colon \sup_{x\geq \epsilon'\sqrt{n}}\Big\{\lambda(x) - \tau_n(x)\Big\} > \epsilon'\sqrt{n}\Big] = 0. \tag{4}$$

Proof. Let the image of the graph of h_0 under ρ^{-1}, i.e., the graph of f_0, be described by the curve $x_0(s) = (h_0(s)+s)/2$, $y_0(s) = (h_0(s)-s)/2$ and let

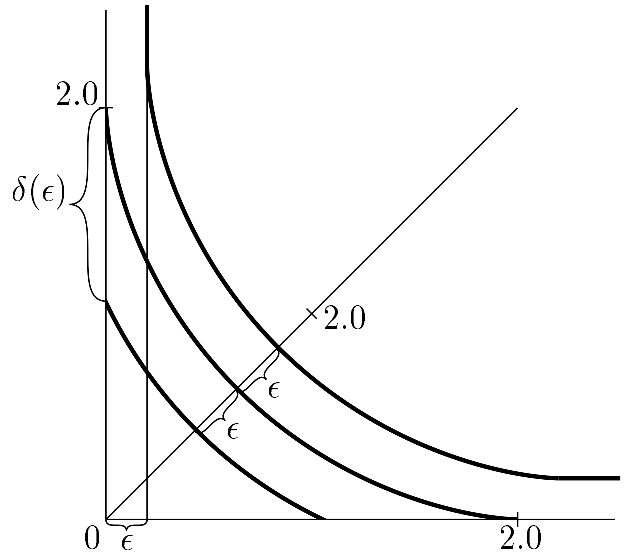

Figure 24: The function f_0 with a 2ϵ stripe.

the image of the graph $h_0 - \epsilon$ under ρ^{-1} be described by the curve $x_\epsilon(s) = (h_0(s) - \epsilon + s)/2$, $y_\epsilon(s) = (h_0(s) - \epsilon - s)/2$. Now let s_1 and s_2 be such that $0 \le x_0(s_1) = x_\epsilon(s_2)$ (obviously, $s_1 \le s_2$). From this equation we obtain $h_0(s_1) - h_0(s_2) = s_2 - s_1 - \epsilon$ and $y_0(s_1) - y_\epsilon(s_2) = s_2 - s_1$ (see Fig. 24). Define the continuous functions

$$\tilde{\delta}(u) = h_0(-2+u) - (h_0(-2) - u), \text{ for } u \ge 0, \text{ and } \delta(\epsilon) = \tilde{\delta}^{-1}(\epsilon), \text{ for } \epsilon \ge 0$$

(note that $h_0(-2) = 2$). Since h_0 is strictly convex, $\tilde{\delta}$ is strongly monotone increasing with $\tilde{\delta}(0) = 0$. Hence, δ is strongly monotone increasing and continuous, too, with $\delta(0) = 0$. It is easily seen from Fig. 24 that $\delta(\epsilon) = f_0(0) - y_\epsilon(s_0)$, for $x_\epsilon(s_0) = 0$. Since h_0 is convex and $s_2 \ge s_1$, $h_0(s_1) - h_0(s_2) \le h_0(-2) - h_0(-2 + s_2 - s_1) = s_2 - s_1 - \tilde{\delta}(s_2 - s_1)$. Thus, $\tilde{\delta}(s_2 - s_1) \le \epsilon$ or $s_2 - s_1 \le \tilde{\delta}^{-1}(\epsilon)$, and consequently $y_0(s_1) - y_\epsilon(s_2) = s_2 - s_1 \le \delta(\epsilon)$.

In summary, if point p lies above the graph of $h_0 - \epsilon$ then its image $\rho(p)$ lies above the graph of $f_0 - \delta(\epsilon)$. It follows that if $h_0(s) - \epsilon \le h_\lambda$, for all $s \ge 0$, then $\tau_n(x) - \delta(\epsilon)\sqrt{n} \le \lambda(x)$, for all $x \ge 0$. Consequently, by Theorem 33,

$$\lim_{n\to\infty} \mathbf{P}[\tau_n(x) - \lambda(x) > \delta(\epsilon)\sqrt{n}, \text{ for some } x \ge 0] = 0.$$

Since δ was continuous with $\delta(0) = 0$, i.e., $\delta(\epsilon)$ becomes arbitrarily small with ϵ small, i.e., ϵ can be choosen with $\delta(\epsilon) < \epsilon'$ and Equation 3 follows.

Considering $h_0 + \epsilon$ in place of h_0 and h_0 in place of $h_0 - \epsilon$ one similarly sees that if a point p lies below the graph of $h_0 + \epsilon$ then its image $\rho(p)$ lies below the graph of $f_0 + \delta(\epsilon)$ if $x(\rho(p)) \geq \epsilon$. By Theorem 33,

$$\lim_{n\to\infty} \mathbf{P}[\lambda(x) - \tau_n(x) > \delta(\epsilon)\sqrt{n}, \text{ for some } x \geq \epsilon\sqrt{n}] = 0.$$

By making $\delta(\epsilon) < \epsilon'$, Equation 4 is now easily derived.

□

With the help of the last two lemmas we can now estimate the expected size of a maximum k-antichain in a random permutation (and thus of k-chains too, of course). Recall that f_0 is the function $f_0((h_0(s)+s)/2) = (h_0(s)-s)/2$, for $s \geq -1$ and h_0 from Theorem 33, and that $\tau_n(x) = \sqrt{n}\, f_0(x/\sqrt{n})$.

Theorem 35 *Let the random variable $l_n(k)$ denote the size of a maximum k-antichain of a permutation of length n drawn with equal distribution from S_n. If $k_n \geq 0$, $n \geq 1$, is a sequence of integers with $k_n = o(\sqrt{n})$, and $\alpha_n \geq 0$, $n \geq 1$, is one of reals with $\lim_{n\to\infty} \alpha_n = \alpha$ then*

$$\frac{\mathbf{E}\, l_n(k_n)}{k_n\sqrt{n}} \to 2, \text{ for } n \to \infty, \tag{5}$$

$$\frac{\mathbf{E}\, l_n(\lfloor \alpha_n\sqrt{n}\rfloor)}{n} \to \int_0^\alpha f_0(x)\, dx, \text{ for } n \to \infty. \tag{6}$$

Proof. Throughout the proof let λ be a Young shape from Λ_n. Let us begin by proving Equation 5 first. Assume some arbitrary $\epsilon > 0$. Since for the size of a maximum k-chain of a permutation with Young shape λ, $\sum_{i=1}^{k_n} \lambda(i) < (2-\epsilon)k_n\sqrt{n}$ implies $\lambda(i_0) < (2-\epsilon)\sqrt{n}$, for some $i_0 \leq k_n$, by Lemma 34,

$$\mathbf{P}\Big[\lambda \in \Lambda_n\colon \sum_{i=1}^{k_n} \lambda(i) < (2-\epsilon)k_n\sqrt{n}\Big] < \epsilon, \text{ for } n \text{ large enough},$$

which gives

$$\mathbf{E}\Big[\sum_{i=1}^{k_n} \lambda(i), \lambda \in \Lambda_n\Big] \geq (2-\epsilon)k_n\sqrt{n}\,(1-\epsilon) \geq (2-3\epsilon)k_n\sqrt{n}.$$

On the other hand, since λ is monotone decreasing we know that $\sum_{i=1}^{k_n} \lambda(i) \le k_n\lambda(0)$, thus $\mathbf{E}\, l_n(k_n) \le 2k_n\sqrt{n}$, by Lemma 32.

To show Equation 6 we may first assume that $\alpha_n\sqrt{n}$ is an integer. Let $\epsilon > 0$ be arbitrary. Now assume that $\lambda(x) \ge \tau_n(x) - \epsilon\sqrt{n}$, for all x with $0 \le x \le \alpha_n\sqrt{n}$. Then

$$\begin{aligned}\sum_{i=1}^{\alpha_n\sqrt{n}} \lambda(i) &= \int_0^{\alpha_n\sqrt{n}} \lambda(x)\,dx \ge \int_0^{\alpha_n\sqrt{n}} \max(\tau_n(x) - \epsilon\sqrt{n}, 0)\,dx \\ &= \int_0^{\alpha_n\sqrt{n}} \tau_n(x)\,dx - 2\epsilon n\end{aligned}$$

and hence, by Lemma 34,

$$\mathbf{P}\Big[\lambda \in \Lambda_n \colon \sum_{i=1}^{\alpha_n\sqrt{n}} \lambda(i) < \int_0^{\alpha_n\sqrt{n}} \tau_n(x)\,dx - 2\epsilon n\Big] < \epsilon, \text{ for } n \text{ large enough.}$$

Again this implies (recall that $\int_0^\infty \tau_n(x)\,dx = n$),

$$\begin{aligned}\mathbf{E}\Big[\sum_{i=1}^{\alpha_n\sqrt{n}} \lambda(i), \lambda \in \Lambda_n\Big] &\ge \Big(\int_0^{\alpha_n\sqrt{n}} \tau_n(x)\,dx - 2\epsilon n\Big)(1-\epsilon) \\ &\ge \Big(\int_0^{\alpha_n} f_0(x)\,dx - 3\epsilon\Big)n. \qquad (7)\end{aligned}$$

Similarly as above, application of Lemma 34 gives

$$\mathbf{P}\Big[\lambda \in \Lambda_n \colon \sum_{i=\lfloor\epsilon\sqrt{n}\rfloor+1}^{\alpha_n\sqrt{n}} \lambda(i) > \int_{\lfloor\epsilon\sqrt{n}\rfloor}^{\alpha_n\sqrt{n}} (\tau_n(x) + \epsilon\sqrt{n})\,dx\Big] < \epsilon, \text{ for } n \text{ large enough.}$$

Hence, observing that trivially $\sum_{i=1}^{\alpha_n\sqrt{n}} \lambda(i) \le n$ and using again that for any k

$\mathbf{E}\, l_n(k) \le 2k\sqrt{n}$ we obtain

$$\begin{aligned}
&\mathbf{E}\Big[\sum_{i=1}^{\alpha_n\sqrt{n}} \lambda(i) \quad, \lambda \in \Lambda_n\Big] \\
&\qquad \le \quad \mathbf{E}\Big[\sum_{i=1}^{\lfloor\epsilon\sqrt{n}\rfloor} \lambda(i), \lambda \in \Lambda_n\Big] + \mathbf{E}\Big[\sum_{i=\lfloor\epsilon\sqrt{n}\rfloor+1}^{\alpha_n\sqrt{n}} \lambda(i), \lambda \in \Lambda_n\Big] \\
&\qquad \le \quad 2\epsilon n + \left(\int_0^{\alpha_n\sqrt{n}} \tau_n(x)\,dx + \alpha_n \epsilon n\right)(1-\epsilon) + n\epsilon \\
&\qquad \le \quad \left(\int_0^{\alpha_n} f_0(x)\,dx + 5\epsilon\right)n,
\end{aligned}$$

if we assume that $\alpha_n \le 2$, e.g., if $\alpha < 2$ and n is large enough. If $\alpha \ge 2$ we trivially have $\mathbf{E}\,\sum_{i=1}^{\alpha_n\sqrt{n}} \lambda(i) \le n$ (recall that $\int_0^\alpha f_0(x)\,dx = 1$ in this case).

Equation 6 now follows since $\lim_{n\to\infty} \int_0^{\alpha_n} f_0(x)\,dx = \int_0^\alpha f_0(x)\,dx$ and ϵ is arbitrary. It is easily seen that replacing $\alpha_n\sqrt{n}$ by $\lfloor \alpha_n\sqrt{n}\rfloor$ leads to an error of $O(\sqrt{n})$ for $\mathbf{E}\, l_n(\alpha_n\sqrt{n})$. □

With Theorem 35 it is also easy to show convergence in probability.

Theorem 36 *With the assumptions of Theorem 35,*

$$\lim_{n\to\infty} \mathbf{P}\Big[\lambda \in \Lambda_n\colon \left|\frac{l_n(k_n)}{k_n\sqrt{n}} - 2\right| > \epsilon\Big] = 0,$$

$$\lim_{n\to\infty} \mathbf{P}\Big[\lambda \in \Lambda_n\colon \left|\frac{l_n(\lfloor\alpha_n\sqrt{n}\rfloor)}{n} - \int_0^\alpha f_0(x)\,dx\right| > \epsilon\Big] = 0.$$

Proof. To show the first equation assume on the contrary that there exist $\epsilon > 0$ and $c_0 > 0$ such that

$$\mathbf{P}\Big[\lambda \in \Lambda_n\colon \sum_{i=1}^{k_n} \lambda(i) > (2+\epsilon)k_n\sqrt{n}\Big] \ge c_0, \text{ for infinitely many } n.$$

On the other hand, as in the proof of Theorem 35,

$$\mathbf{P}\Big[\lambda \in \Lambda_n\colon \sum_{i=1}^{k_n} \lambda(i) < (2-\epsilon c_0/3)k_n\sqrt{n}\Big] \le \epsilon c_0/6, \text{ for } n \text{ large enough.}$$

Both inequalities imply

$$\begin{aligned}\mathbf{E}\, l_n(k_n) &\geq \Big((2+\epsilon)c_0 + (2-\epsilon c_0/3)(1-c_0-\epsilon c_0/6)\Big)k_n\sqrt{n} \\ &\geq (2+\epsilon c_0/3)k_n\sqrt{n}, \quad \text{for infinitely many } n,\end{aligned}$$

which contradicts Theorem 35.

Let $I(\alpha)$ be an abbreviation for $\int_0^\alpha f_0(x)\,dx$ then the second equation is shown analogously by assuming

$$\mathbf{P}\Big[\lambda \in \Lambda_n\colon \sum_{i=1}^{\lfloor \alpha_n\sqrt{n}\rfloor} \lambda(i) > (I(\alpha)+\epsilon)n\Big] \geq c_0, \text{ for infinitely many } n.$$

On the other hand, as in the proof of Theorem 35,

$$\mathbf{P}\Big[\lambda \in \Lambda_n\colon \sum_{i=1}^{\lfloor \alpha_n\sqrt{n}\rfloor} \lambda(i) < (I(\alpha)-\epsilon c_0/3)n\Big] \leq \epsilon c_0/(3I(\alpha)), \text{ for } n \text{ large enough.}$$

Both inequalities imply

$$\begin{aligned}\mathbf{E}\, l_n(\lfloor \alpha_n\sqrt{n}\rfloor) &\geq \Big((I(\alpha)+\epsilon)c_0 + (I(\alpha)-\epsilon c_0/3)(1-c_0-\epsilon c_0/(3I(\alpha)))\Big)n \\ &\geq (I(\alpha)+\epsilon c_0/3)n, \quad \text{for infinitely many } n,\end{aligned}$$

which again contradicts Theorem 35. □

4 Box Orders

Recall from Section 1 that the dominance order on points in $\mathbf{R}^k$ may be extended to aligned boxes too. A box b dominates a box b' if the lower corner of b dominates the upper corner of b'. Note that this implies that every point of box b dominates all points of box b'. Let us use some more precise notation. The *weighted k-dimensional box order* (V, l, u, w) consists of mappings $u, l \colon V \to \mathbf{R}^k$ and $w \colon V \to \mathbf{N}$ such that $l(v)$ is the lower and $u(v)$ the upper corner of a box v and $w(v)$ the weight of v. The symbol $\mathrm{box}(v)$ is used to denote the set of points of box v. The *unweighted* case may be simulated by giving all points unit weights. In contrast to the weight $w(v)$ of a box v in a box order we will often attribute a *chain weight* $W(v)$ to v which is the maximum weight of a chain with v as its maximal element. We have already indicated how to compute the chain weights for elements of arbitrary orders in Section 3. They can be computed more efficiently for box orders, as seen below.

Throughout the next sections we assume that the points $l(v)$ and $u(v)$ of a box representation have mutually different x- and y-coordinates. Otherwise, we may obtain an equivalent box order fulfilling this requirement by perturbing the corner points with two line sweeps in the following way. Recall that two boxes v and v' where $u(v)$ has at least one coordinate equal to $l(v')$ count as incomparable according to the definition of the dominance order on boxes. Hence, to maintain the order relation we have to perturb points with, say, the same x-coordinates slightly in such a way that points which are lower corners have smaller x-coordinates than such which are upper corners. A similar perturbation is done for the y-coordinate. Boxes with coinciding upper and lower corners may be replaced by a small box in advance. It is easily seen, that in this manner the order relation on the boxes is not changed.

4.1 Maximum Chain and Minimum Antichain Partition for Boxes

Let a two-dimensional box order $\mathbf{B} = (V, l, u, w)$ be given. To find a maximum weighted chain, we use a sweep line L sweeping from left to right halting at every corner. L represents the currently available maximum weighted chains by markers. For each point y on L, the next marker below y contains the weight of such a chain dominated by y. The marker also points to the box

MAXCHAIN(V, l, u, w)
insert dummy d_l in L at $-\infty$ with $W(d) := 0$;
$\mathit{link}(d_l) \leftarrow$ nil;
for each $p \in \{l(v), u(v) \mid v \in V\}$ from left to right **do**
    if $p = l(v)$ for some $v \in V$ **then**
        $m \leftarrow$ first marker below p on L;
        $W(v) \leftarrow W(m) + w(v)$;
        $\mathit{link}(v) \leftarrow$ box of marker m;
    if $p = u(v)$ for some $v \in V$ **then**
        $m \leftarrow$ first marker below p on L;
        if $W(v) > W(m)$ **then**
            insert a new marker m_v at p_y in L with box v;
            $W(m_v) \leftarrow W(v)$;
            remove all markers m' above m_v with
                        $W(m') < W(m_v)$ from L;
$v \leftarrow$ box of the uppermost marker on L; $C \leftarrow \{v\}$;
while $\mathit{link}(v) \neq$ nil **do**
    $v \leftarrow \mathit{link}(v)$; $C \leftarrow C \cup \{v\}$;
return S;

Figure 25: Algorithm MAXCHAIN

which is the maximal box in this chain. Note that the chain weights increase with increasing y-coordinate on L.

If L meets a lower corner $l(v)$ the maximum weight $W(v)$ of a chain with box(v) as maximal element is computed and assigned to v. If L meets an upper corner $u(v)$ this corner is inserted as new marker in L. It gets the chain weight $W(v)$ of the corresponding box v. At the beginning, a dummy point is inserted in L with a y-coordinate smaller than that of all points to be processed and chain weight 0. Fig. 25 shows Algorithm MAXCHAIN.

Theorem 37 *Algorithm MAXCHAIN computes a maximum weighted chain of a weighted box order* (V, l, u, w).

Proof. We show that at the end of the main loop in MAXCHAIN the following invariant holds true.

> *If y is an arbitrary point on L and m the next marker below y, then a maximum weighted chain dominated by point y has weight $W(m)$.*

Of course, the invariant implies that Algorithm MAXCHAIN computes a maximum weighted chain, since all boxes are dominated by the uppermost point on L after the sweep has finished.

If L has stopped at some point $p = l(v)$ no new box has become available to increase any maximum weighted chain and no weight of any marker has been changed. But note that, by the invariance assumption, the weight $W(m)+w(v)$ assigned to v is the maximum weight of a chain with maximal element v, where m is the marker below $l(v)$.

On the other hand, suppose L has stopped at $p = u(v)$. If $y < u_y(v)$ or if no new marker is inserted in the sweep line, there is neither a new maximum weighted chain below y nor a new marker below y. Hence we assume $y \geq u_y(v)$ and that a new marker m_v has been inserted at height $u_y(v)$, i.e., there is a new chain with weight $W(v) = W(m_v)$ available for points above $u_y(v)$. Let m' and m be the marker immediately below y before and after the insertion of m_v. If $m' = m$ then $W(m') > W(m_v)$ (otherwise, m' would have been removed) and, by induction, $W(m) = W(m')$ remains the maximal weight of a chain dominated by y. If $m' \neq m$ then m can only be m_v and $W(m_v) > W(m')$ either by the condition for removing markers or by the condition for the insertion of m_v. Since, by induction, $W(m')$ was maximal among all chain weights save the new one ending in v, $W(m_v)$ surely is a maximal weight for y now. $\square$

The sweep line may be implemented to allow the execution of all operations save the removal of points in one iteration of the main while-loop in $O(\log n)$. Each point that is removed must have been inserted before and since the weights of the markers are sorted with increasing y-coordinate, in sum the removal steps only take linear time. Hence, a running time of $O(n \log n)$ is guaranteed.

The unweighted case may be simulated by unit weights. Of course, then Algorithm MAXCHAIN works in a similar manner as Algorithm SKELETON (see Fig. 6). The weights of all markers are different and the number of markers on L cannot exceed the length of a maximum chain in $\mathbf{B}$. If ω is the size of a longest chain the algorithm takes time $O(n \log \omega)$. If each element of $\mathbf{B}$ has unit weight then no two elements with the same chain weight are comparable. Hence, collecting all elements of the same weight in one set yields a partition of **B** into antichains. Thus, the canonical antichain partition is a byproduct of algorithm MAXCHAIN. We summarize these remarks in

Theorem 38 *If a box order* $\mathbf{B} = (V, l, u, w)$ *is given and the set* $\{l(v), u(v) : v \in V\}$ *is sorted by x-coordinate, then*

1. *in the unweighted case a minimum antichain partition of* **B** *can be obtained in* $O(n \log \omega)$*,*
2. *a maximum chain in* **B** *can be found in* $O(n \log \omega)$ *in the unweighted and in* $O(n \log n)$ *in the weighted case,*

where ω *is the size of a longest chain in* **B** *and* $n = |V|$. □

Things change drastically when we turn to higher dimensions. The symmetry between chains and antichains gets lost. When we use a sweep plane instead of a sweep line we no longer have a linear order on the points in the plane which makes it impossible to simply extend the algorithms for the two-dimensional case to higher dimensions.

The concept of a skeleton cannot be utilized for algorithmic tasks, since its size is no longer of the order of the point set as in two dimensions. Hence, it is not possible to compute maximum k-chains in a similar way as before. It is not clear how to beat the known algorithms using network flows (e.g., Sarrafzadeh and Lou [25], see also Section 3). It becomes a nontrivial task even to compute maximum chains or antichains.

Assume a k-dimensional box order **B** is given, for some higher dimension $k > 2$. We use dynamic multi-dimensional range trees (see, e.g., Smid [28] or Mehlhorn [22]) for the construction of a maximum chain for a point set in k dimensions. We need a data structure for a point set X in $(k-1)$-dimensional space that, for a given query point q, allows to find some $p \in P$ with maximum chain weight $W(p)$ among all points of X that are dominated by q. We also want to insert new points with some given chain weight. Given such a data structure, it is easy to compute a maximum weighted chain for a point set X in k dimensions. A sweep plane visits all points ordered by increasing last coordinate (see Fig. 26). For each point q on the sweep plane a point p is found in the range tree that has maximum chain weight $W(p)$ among all points dominated by q in the first $k-1$ coordinates. But since all points with smaller values than q in their last coordinate have been swept and have already been inserted in the range tree, p also has maximum weight

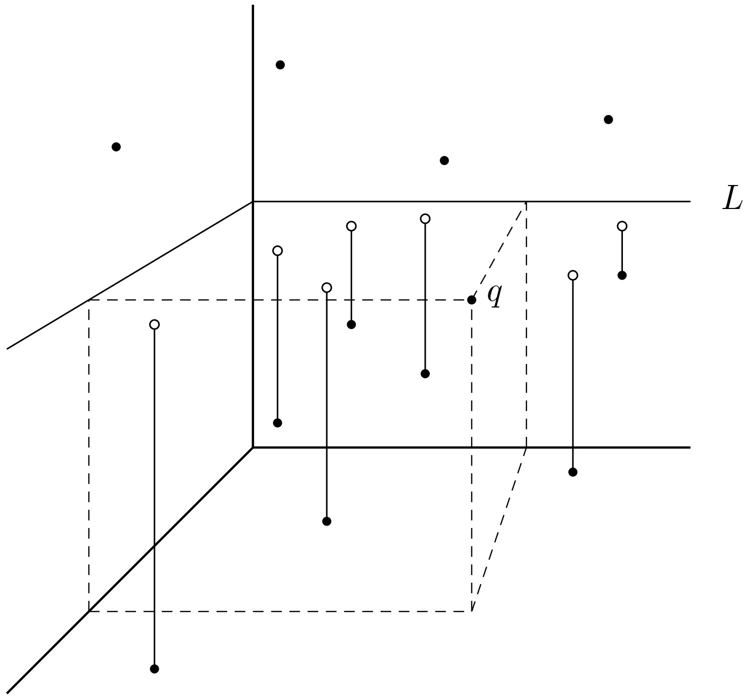

Figure 26: Plane L sweeping a three-dimensional point set halts at q.

among points dominated by q in all k dimensions. Hence, we may insert q in the range tree with chain weight $W(q) = W(p) + w(q)$. Along with $W(q)$ we may also store a link to point p. After insertion of all points a maximum weighted chain is easily found. At first, a point p_m with the highest weight ever computed during the sweep is searched. Then, beginning with p_m, the chain is extracted by following the corresponding chain of links. Again, if all points have unit weight then $W(p)$ is the size of a chain with maximal element p and two points with the same chain weight cannot be comparable. Hence, a minimum antichain partition is found as byproduct.

With the following changes the above approach is easily adapted to compute a maximum weighted chain for a box order. If point q on the sweep plane corresponds to a lower point $l(v)$ of some box v we calculate $W(v) = W(p) + w(v)$ as above but do not yet insert q in the tree. If $q = u(v)$ for some box v we insert q in the range tree with chain weight $W(q) = W(v)$ that has already been calculated before.

For convenience, let us briefly recall how such a $(k-1)$-dimensional range tree T works. We set $d = k-1$. Let the d coordinates be denoted by $x_1, \ldots, x_d$. Points S are represented by the leaves of a binary tree T ordered according to their x_d-coordinate. If $d = 1$ each node t of T points to the leaf of some point p with chain weight $W(p)$ maximal among all weights in the leaves below t, i.e., in the leaves of the subtree rooted at t. If $d > 1$ each node t of the main range tree points to a $(d-1)$-dimensional secondary range tree constructed recursively with respect to the first $d-1$ coordinates for the points in the subtree of t.

Suppose now we want to find, for a given query point q, the point with maximal weight among all points of X with all coordinates smaller than that of q. In a first step the point of X with smallest x_d-coordinate greater than that of q is searched in the main tree. Let the search path be S_q. Let L_q be the set of left children of nodes in S_q that are not itself in S_q. It is easily seen that each point with x_d-coordinate smaller than or equal to that of q has a leaf below some node in L_q. Hence, to find a point p with $W(p)$ maximal among all points dominated by q we proceed as follows. If $d = 1$ we check all leaves pointed to by the nodes in L_q and return the leaf with maximum weight. If $d > 2$ the range trees in nodes $t \in L_q$ allow to find points p_t with maximum weight among leaves below t and dominated by q in the first $d-1$ coordinates. In this case the point searched for is that with maximum weight among points p_t, for $t \in L_q$. On the other hand, if we want to insert point q into tree T this may be done by first inserting it in the main tree and then inserting it in all secondary range trees at nodes along the insertion path, if $d > 1$. If $d = 1$, pointers along the insertion path are set to the leaf belonging to q if its weight is the new maximum in the corresponding subtree.

It is easily seen that a query takes time $O(\log^d n)$ if all trees are balanced. If some trees become unbalanced during an insertion they must be rebalanced and it can be shown that this takes amortized time $O(\log^d n)$. For $d = 1$ we need linear space. And since a point is contained in at most $\log n$ secondary trees if $d > 1$, the total amount of space is $O(n \log^{d-1} n)$, by induction. We leave it to the reader to supply the omitted details that give a complete proof of the following statement.

Theorem 39 *A minimum antichain partition and a maximum chain of a box order on n boxes in $\mathbf{R}^k$ can be obtained in $O(n \log^{k-1} n)$ time and $O(n \log^{k-2} n)$ space.* □

For completeness, we describe a slight improvement on Theorem 39 if a maximum chain and the canonical antichain partition (see Section 2.1) is searched for an unweighted point set X in $\mathbf{R}^3$. Again, we use a sweep plane scanning the points along increasing last coordinate. During the sweep we maintain the intersection of the 3-dimensional antichain layers with the sweep plane. This intersection is a 2-dimensional layer partition $L_1 \dots , L_s$ of the projection π of a subset of X on the sweep plane.

Suppose the sweep plane reaches a new point p with largest z-coordinate among points seen so far. We want to insert p in the jump surfaces constructed so far with the help of the jump lines $L_1, \dots , L_s$ on the sweep plane. Suppose L_i lies in the shadow of L_{i-1} on the sweep plane and suppose the projection $\pi(p)$ of p lies between L_{i-1} and L_i. It is easily seen that in such a case p has to be inserted in the i-th jump surface. Hence, we have to change L_i to contain the projection $\pi(p)$. After the insertion we have a new jump line L'_i on the sweep plane. L'_i is computed from L_i by removing all points of L_i that lie in the (2-dimensional) shadow of $\pi(p)$ and inserting p in L_i instead. At the same time we set a pointer from p to some point on L_{i-1} dominated by p for the final quick extraction of a maximum chain. If p lies above the last jump line L_s then p simply opens a new jump line L_{s+1}. This procedure yields a partition of X by nonintersecting jump surfaces and by the appropriate generalization of Lemma 1 to $\mathbf{R}^3$ this implies that the corresponding set partition is the canonical antichain partition. Of course, a chain is maximum that is extracted along the sequence of pointers beginning with some point on the highest layer and taking one point from each layer below.

An important step in the procedure is to find the next layer L_i above p. Mehlhorn and Näher [23] propose a data structure that allows to find those segments of n segments parallel to the x-axis in $\mathbf{R}^2$ which lie immediately above and below a query point in time $O(\log n \log\log n)$. This structure needs space $O(n \log n)$ and is fully dynamic. Insertions and deletions of segments take amortized time $O(\log n \log\log n)$. The segments we are interested in are the vertical line segments of all jump lines $L_1, \dots , L_s$ on the sweep plane. The jump lines are represented by doubly linked lists. We find the segment of L_i lying above the current point q and remove all segments on L_i from left to right until we find a segment outside the shadow of p. Next we have to insert two new vertical segments into L_i—the segment containing q together with the shortened old segment of L_i immediately above q. On the other hand, the segment below p immediately yields a comparable point on layer L_{i-1}.

Hence, altogether we have $O(|P|)$ queries and $O(|P|)$ insertions and removals of segments and obtain

Theorem 40 *A minimum antichain partition and maximum chain of a quasi-permutation of n points in $\mathbf{R}^3$ can be computed in $O(n \log n \log\log n)$ time and $O(n \log n)$ space.* □

4.2 Maximum Antichain and Minimum Chain Partition for Boxes

The computation of a maximum weighted chain for a box order in Section 4.1 was a straightforward generalization of the corresponding computation for a point set. Unfortunately, this is not true in the case of a maximum weighted antichain. The reason is that comparability of two boxes means that the dominating box is to the right and above the dominated one, i.e., in the very same relation as two comparable points. Whereas boxes may be incomparable not only if one is to the right and below the other but also because they intersect or because one is below the other etc. This enhanced set of possible relative situations of incomparable boxes asks for a treatment slightly different from that for points.

On the other hand, we may try to take over something from antichains of points. Recall the definition of a jump line for a given set of points in the plane given in Section 2.1. If S is a jump line and $l, u\colon V \to \mathbf{R}^2$ a set of boxes we denote the set of elements whose box intersect S by $A(S)$, i.e., $A(S) = \{v \in V \mid \mathrm{box}(v) \cap S \neq \emptyset\}$. Then we have the following lemma which shows that even in the case of boxes we have to deal with jump lines in computing an antichain of boxes.

Lemma 41 *Let $\mathbf{B}$ be a box order. If S is a jump line then $A(S)$ is an antichain. On the other hand, if A is an antichain of $\mathbf{B}$ then there exists a jump line S such that $A \subseteq A(S)$.*

Proof. Assume that $A(S)$ is not an antichain. Then there are boxes $v, v' \in A(S)$ with v' dominating v. Consequently, for any two different points $p \in \mathrm{box}(v) \cap S$ and $p' \in \mathrm{box}(v') \cap S$ on the jump line S, p' would dominate p. But recall that any two different points on a jump line are incomparable.

Now let A be an antichain of $\mathbf{B}$ and let $u(A) = \{u(v) : v \in A\}$. Let $S = L(u(A))$ be the jump line of $u(A)$. Suppose there is an element $v \in A$

with $\mathrm{box}(v) \cap S = \emptyset$. Since $u(v)$ is not part of the jump line of $u(A)$, $u(v)$ lies in the shadow of $u(A)$. Since $\mathrm{box}(v)$ cannot cross jump line S, $l(v)$ lies in the shadow of $u(A)$, too. But then, by the definition of a shadow, there is a box $a \in A$ with $u(a)$ dominated by $l(v)$. Thus, a is dominated by box $v \in A$ contradicting A being an antichain. □

Given a weighted box order $\mathbf{B}$ we define the *weight of a jump line* S as the sum of weights of all boxes intersecting S. If S is a jump line and $p \in S$, then we refer to the part of S that is above and to the left of p as *jump line ending in* p, denoted by S_p (see Fig. 28), and again its weight is the sum of weights of intersecting boxes.

Algorithm MAXANTICHAIN-WEIGHTED (see Fig. 27) computes an antichain of maximum weight. It uses two different data structures. The sweep line L halts at every point $l(v)$ and $u(v)$, for $v \in V$. Roughly, it contains a list of weighted markers, so that the weight $W(m)$ of marker m is the weight of a heaviest jump line ending in m. Moreover, a heaviest jump line ending in an arbitrary point y on L can be composed by joining the vertical line segment from y to the next marker m above y with a heaviest jump line ending in m. Structure L is initialized with dummy points d_u and d_l of weight 0 and $+\infty$ above and below all points that will ever be inserted into L. The second structure Δ contains a list of all *open* boxes, i.e., boxes which have their left sides already swept but not their right ones. Given two y-coordinates $y_2 \leq y_1$, the total weight of all open boxes v the upper sides of which have y-coordinate $u_y(v)$ with $y_2 \leq u_y(v) < y_1$ on L is denoted by $\Delta(y_1, y_2)$.

Theorem 42 *Algorithm MAXANTICHAIN-WEIGHTED computes a maximum weighted antichain of a weighted box order (V, l, u, w).*

Proof. A list of incomparable points is stored along with each marker m'. Let the corresponding jump line be called *associated with* m'. We will show that at the *end* of the main loop in Algorithm MAXANTICHAIN-WEIGHTED the following invariant holds true.

> *The jump line associated with a marker m' is a maximum weighted jump line among those ending in m'; it has weight $W(m')$. Generally, if y is an arbitrary point on L and m the next marker above or on y, then a maximum weighted jump line that ends in y on L has weight $W(m) + \Delta(m, y)$.*

MAXANTICHAIN-WEIGHTED(V, l, u, w)
insert dummy d_u with $W(d_u) \leftarrow 0$ in L at $+\infty$;
insert dummy d_l with $W(d_l) \leftarrow +\infty$ in L at $-\infty$;
for each $p \in \{l(v), u(v) \mid v \in V\}$ from left to right **do**
    if $p = l(v)$ for some $v \in V$ **then**
        add $w(v)$ to $W(m')$ for all markers m' in interval $[l_y(v), u_y(v)]$;
        insert a new item in Δ at height $u_y(v)$ with weight $w(v)$;
        $m \leftarrow$ next marker above p on L;
        $m^* \leftarrow$ next marker below p on L;
        while $W(m) + \Delta(m, m^*) \geq W(m^*)$ **do**
            remove m^* from L;
            $m^* \leftarrow$ next marker below p on L;
    if $p = u(v)$ for some $v \in V$ **then**
        $m \leftarrow$ next marker above p on L;
        insert a new marker m_v at p_y in L;
        $W(m_v) \leftarrow W(m) + \Delta(m, m_v)$;
        remove item at $u_y(v)$ from Δ;
        $list(m_v) \leftarrow list(m) \cup \{p\}$;
$T \leftarrow$ jump line of points in $list(lowest(L))$;
for each $v \in V$ **do**
    if v intersects T **then** $A \leftarrow A \cup \{v\}$;
return A;

Figure 27: Algorithm MAXANTICHAIN-WEIGHTED

This invariant implies the theorem, since structure Δ is empty after all boxes have been swept (i.e., there is no box left open).

Let W' denote the sweep line structure and Δ' denote the open box structure before a halt of the sweep line L and let W and Δ be the pair of structures after the halt. Let m be the first marker above y on L. Let S_y denote that part of the jump line of $list(m) \cup \{y\}$ that ends in point y (see Fig. 28). We show that S_y has weight $W(m) + \Delta(m, y)$ and that this weight is maximal among weights of jump lines ending in y.

At first, suppose the sweep line L halts at some point $l(v)$ for $v \in V$. If $y > u_y(v)$ there is no change to any jump line ending in y. Otherwise, we have $y < u_y(v)$. If furthermore $u_y(v) < m$ then S_y intersects the new box

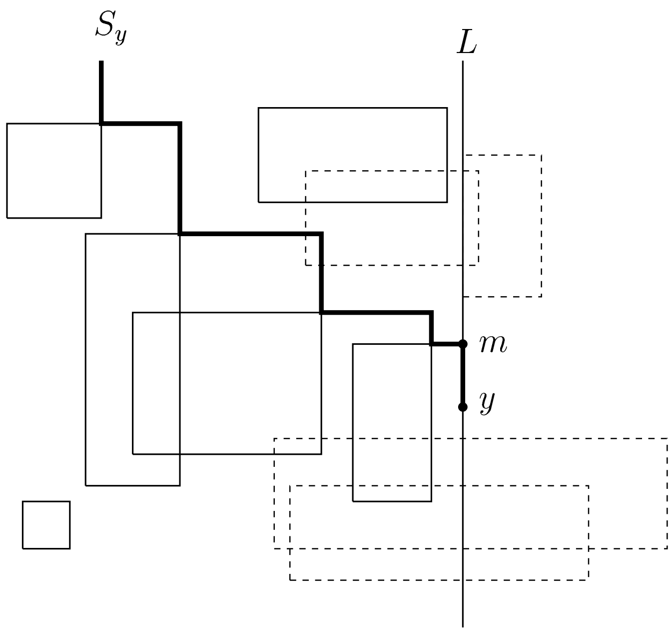

Figure 28: Sweep line L with open boxes and jump line S_y ending in y.

v and has weight $W(m) + \Delta(m, y) = W'(m) + (\Delta'(m, y) + w(v))$. By the invariance assumption, this weight is maximal among all jump lines ending in y. If $l_y(v) < m < u_y(v)$ then S_y has weight $W(m) + \Delta(m, y) = (W'(m) + w(v)) + \Delta'(m, y)$ which again is maximal. If $m < l_y(v)$ then the weight of S_y is $W(m)+\Delta(m, y) = W'(m)+\Delta'(m, y)$ which is the maximum weight of a jump line that avoids v. On the other hand, if a jump line S'_y ending in y intersects v then there is a $y' > l_y(v)$, such that S'_y is composed of a jump line ending in y' and a vertical segment from y to y'. Let m' be the first marker above y'. Since the case $m' > y' > l_y(v)$ has already been considered, by the definition of Δ, the weight of S'_y is at most

$$W(m') + \Delta(m', y') + \Delta(y', m) + \Delta(m, y) = W(m') + \Delta(m', m) + \Delta(m, y).$$

By the condition on the removal of markers in the algorithm, we know that $W(m) > W(m') + \Delta(m', m)$. Consequently, the weight of S'_y is less than $W(m)+\Delta(m, y)$, the weight of S_y. Note that the weight of markers m' between $l_y(v)$ and $u_y(v)$ is increased by $w(v)$. But the weight of the associated jump lines is increased by the same amount since they all intersect the box of v now.

Now suppose L halts at some point $u(v)$. Since v is no longer open we have to remove $u_y(v)$ from Δ'. On the other hand, we have to maintain the invariant. Thus, a new marker m_v is inserted in L with weight $W(m_v) = W'(m') + \Delta'(m', m_v)$, where m' is the next marker above $u_y(v)$. If m, the next marker above y, is different from m_v there again is no change in the weight of jump lines ending in y. On the other hand, if m_v is the next marker above y, then the weight of S_y is

$$W(m_v) + \Delta(m_v, y) = W'(m) + \Delta'(m, m_v) + \Delta'(m_v, y) = W'(m) + \Delta'(m, y),$$

which is maximal by the invariance assumption, since no new box has to be considered. Note that the associated jump line of m_v is constructed by enlarging that of m in such a way that it additionally intersects all open boxes between m and m_v with total weight $\Delta'(m, m_v)$. This means that it has weight $W(m_v)$ which is maximal. □

L may be implemented by a balanced binary tree. One has to be careful only about adding some weight w to $W(m')$ for markers m' lying in an interval $[l, u]$. We give a sketch of a possible implementation and invite the reader to supply the details. Suppose that each marker corresponds to a leave of the tree and that the leaves are sorted by increasing y-coordinate of the markers. Let h be the height of such a tree. Let each node of the tree have some extra field holding the increment in the weight for all leaves in its subtree. The weight of a marker is then easily computed in h steps by summing up all weights along the search path to the corresponding leave and finally adding this sum to the weight stored in the leave. On the other hand, note that the set of all leaves corresponding to an interval $[l, u]$ is the disjoint union of the leaves of at most $2h$ subtrees which can be found along the search paths of l and u. Consequently, if the weight of all leaves in interval $[l, u]$ have to be increased, it suffices to update the weight increment fields of the at most $2h$ roots of these subtrees.

The main operation in a rebalancing of a balanced tree, e.g., of a red-black tree (see [1]), is a rotation at some internal node t. But it is easily seen that, e.g., by a propagation of the weight increase from node t to its children before the rotation the overall weight of the markers remains the same. Hence, the computing time of the whole algorithm increases by a constant factor. Consequently, the addition of some weight to an interval as well as insertion, deletion, predecessor and successor queries, and the computation of the weight of some element can all still be done in time $h = O(\log n)$.

Δ may be implemented by any one dimensional range tree where insertion, deletion, and query again takes $O(\log n)$ time. The main loop is executed n times and each step therein takes logarithmic time save the while loop. But in total the while loop is executed at most n times since each removed point must have been inserted before. Of course, the test for intersection of a box v with jump line T can be done in time $O(\log n)$. The space requirement of both data structures is linear. In summary, we obtain

Theorem 43 *A maximum weighted antichain of a weighted box order on n boxes can be computed in time $O(n \log n)$ and linear space.* □

Note that one can do without the Δ structure if one uses subtraction in the L structure. But the above algorithm is easier to understand and it can be adapted to the case where no subtraction is allowed (e.g., in semigroups).

It is possible to define a minimum weighted chain partition that is dual to a maximum weighted antichain and to modify and extend Algorithm MAX-ANTICHAIN-WEIGHTED accordingly. The proof of correctness then may utilize the duality between chain partitions and antichains as does the proof of Theorem 44. But this would have made the Algorithm MAXANTICHAIN-WEIGHTED more complicated and less understandable, whereas the notion of a weighted chain partition does not seem to be very intuitive or to have interesting applications.

Now let us consider Algorithm MINCHAINPART computing a minimum chain partition for an unweighted box order (V, l, u). Although we may compute a maximum antichain by Algorithm MAXANTICHAIN-WEIGHTED using unit weights, we compute it directly in Algorithm MINCHAINPART again. This will turn out to be useful in the proof. A sweep line L halts at every point $l(v)$ and $u(v)$, for $v \in V$. Similarly as in Algorithm MAXANTICHAIN-WEIGHTED, it contains a list of markers with associated jump lines such that a maximum antichain to the left and above the marker is given by the boxes intersected by that jump line. At the same time each marker is associated with a maximum chain below the marker of boxes seen so far. For convenience, the sweep line is initialized with a dummy marker d_l below all points that are ever swept.

Theorem 44 *Algorithm MINCHAINPART computes a maximum antichain and a minimum chain partition of a box order $\mathbf{B} = (V, l, u)$.*

MINCHAINPART(V, l, u)
insert dummy d_l in L at $-\infty$;
$k \leftarrow 0$;
for each $p \in \{l(v), u(v) \mid v \in V\}$ from left to right **do**
    if $p = l(v)$ for some $v \in V$ **then**
        $m^* \leftarrow$ next marker below p on L;
        if $m^* \neq d_l$ **then**
            $chain(v) \leftarrow chain(box(m^*))$;
            add v to $chain(v)$;
            remove m^* from L;
        else
            $k \leftarrow k + 1$;
            $C_k \leftarrow \{v\}$; $chain(v) \leftarrow C_k$;
    if $p = u(v)$ for some $v \in V$ **then**
        $m \leftarrow$ next marker above p on L;
        insert a new marker m_v at p_y in L;
        $box(m_v) \leftarrow v$;
        $list(m_v) \leftarrow list(m) \cup \{p\}$;
$T \leftarrow$ jump line of points in $list(lowest(L))$;
for each $v \in V$ **do**
    if v intersects T **then** $A \leftarrow A \cup \{v\}$;
return $A, C_1, \ldots, C_k$;

Figure 29: Algorithm MINCHAINPART

Proof. Obviously, $C_1, \ldots, C_k$ is a chain partition. The upper sides of open boxes intersect the sweep line L. Let these intersection points on L be the *open lines*. Note that along with each marker a list of points is stored. Call the set of boxes already processed by the sweep and intersecting the jump line of these points the *associated antichain* of the marker. Then we have the following invariant at the end of the main loop in MINCHAINPART.

> *The size of an antichain associated with an arbitrary marker on L equals the number of markers and open lines above or on it. The number k of chains introduced so far equals the total number of markers and open lines on L.*

The invariant implies the theorem. For at the end of the sweep all open lines are removed. Hence, the size of the antichain associated with the lowest marker equals the number of chains in the partition. Since an antichain is at least as large as a chain partition this shows both to be optimal.

To show the invariant suppose first that L halts at point $l(v)$ for some $v \in V$. Let m^* be the next marker below $l(v)$ on L. If m^* differs from the lower dummy point d_l then the number of chains remains the same as remains the total number of markers and open lines since one marker is removed and a new open line (at $u(v)$) is introduced. The only markers that additionally have the new open line above it are those between $l(u)$ and $u(v)$ (there is no marker between m^* and $l(v)$). But their associated jump lines certainly intersect the new box v too. This observation also holds true if m^* equals the lower dummy d_l. In the latter case the number of chains in the partition increases as does the number of open lines.

Suppose L halts at point $u(v)$ for some $v \in V$. We only have to show that the invariant is true for the new marker m_v at $u(v)$. Let m be the next marker above m_v. By the invariant and the construction of the new jump line for marker m_v, the jump line intersects as many boxes as there are markers and open lines above or on m in addition to the boxes corresponding to the open lines between m_v and m, as box v itself. □

If L is implemented by any balanced tree all steps in the loop take time $O(\log n)$. Hence, we have

Theorem 45 *A maximum antichain and a minimum chain partition of a box order of n boxes can be computed in time $O(n \log n)$ and linear space.* □

5 Appendix

In Section 3.4 we cite a result stated in Kerov and Veršik [31] (see also Logan and Shepp [17]) saying that with high probability the Young shape of a random permutation will not differ much from a special standard shape that is given explicitly. This result is stated without proof. In Logan and Shepp [17] one finds the necessary calculations, which allow to derive the needed result in a simple manner. In the following, we supply some details which are left to the reader by Logan and Shepp. Finally, we apply the derived inequalities to give a complete proof of Theorem 33. The reader is supposed to have the paper of Logan and Shepp at hand to check other calculations which will not be repeated in full length here. Nevertheless, the following notes might be useful to gain some insight in the involved proof.

Recall from Section 3.4 that the probability that shape λ is the shape of the P- and Q-symbol of a random permutation is given by the Plancherel-Schensted distribution

$$\mathbf{P_S}(\lambda) = n! \Big/ \Big(\prod_{c \text{ cell in } \lambda} h_c \Big)^2,$$

where h_c is the hook length of cell c of λ, i.e., the number of cells above and to the right of c including c once.

We make some assumptions on λ that will simplify the following computations considerably. We think of λ as a differentiable, monotone decreasing (hence invertible) function $\lambda(x)$ from $\mathbf{R}$ in $\mathbf{R}$. If we start with the border of the Young shape λ this my be achieved by slightly inclining the line segments and smoothing the edges in such a way that with arbitrarily small error the area below $\lambda(x)$ again has value n and $h_c = \lambda(x_c) - y_c + \lambda^{-1}(y_c) - x_c$, where (x_c, y_c) is the center of cell c. By convexity of the logarithm, with arbitrarily small error

$$\log \Big(\prod_{c \text{ cell in } \lambda} h_c \Big) \geq \iint_R \log(\lambda(x) - y + \lambda^{-1}(y) - x)\, dx\, dy,$$

where R is the region between the x- and y-axis and $\lambda(x)$. Now we scale $\lambda(x)$ such that it has integral unity, i.e., we define $f_\lambda(x) = (1/\sqrt{n})\, \lambda(\sqrt{n}x)$ and set

$$H(f) = \int_0^\infty \int_0^{f(x)} \log\Big(f(x) - y + f^{-1}(y) - x\Big)\, dy\, dx.$$

Consequently,

$$\begin{aligned}
\mathbf{P_S}(\lambda) &\le n!\exp\Big(-2n\int_0^\infty\int_0^{f_\lambda(x)}\log(n^{1/2})+\log(f_\lambda(x)-y+{f_\lambda}^{-1}(y)-x)\,dy\,dx\Big)\\
&= (n!/n^n)\,e^{-2nH(f_\lambda)}\\
&= \sqrt{2\pi n}\,e^{-2n(H(f_\lambda)+1/2)}\Big(1+o(1/n)\Big),
\end{aligned}$$

using Sterling's formula.

To proof Theorem 33 we want to bound this probability for set

$$L(\epsilon)=\{\,\lambda\in\Lambda_n\colon\ \sup_{x\in\mathbf{R}}|h_\lambda(x)-h_0(x)|>\epsilon\,\} \tag{8}$$

where the graph of h_λ is that of f_λ scaled by $\sqrt{2}$ and rotated by $\pi/4$ (see Equation 11 below with $f=f_\lambda$ and $h=h_\lambda$) and h_0 is given by

$$h_0(x)=\max\Big\{\,\frac{x}{\pi}\arcsin\frac{x}{2}+\sqrt{1-\frac{x^2}{4}},\frac{|x|}{2}\,\Big\}\text{ for }x\in\mathbf{R}.$$

We will show in Lemma 49 below that, for $\lambda\in L$, $H(f_\lambda)\ge -1/2+\delta(\epsilon)$, where $\delta(\epsilon)>0$ is a constant depending on ϵ. Furthermore, the number $p(n)$ of shapes in Λ_n, i.e., of shapes of size n, is the number of partitions of n and can be bounded by $p(n)=e^{O(\sqrt{n})}$ (see Hardy and Wright [15]). Consequently, Theorem 33 is shown since

$$\lim_{n\to\infty}\mathbf{P_S}[L(\epsilon)]\le\lim_{n\to\infty}p(n)\sqrt{2\pi n}\,e^{-2n\delta(\epsilon)}\Big(1+o(1/n)\Big)=0$$

To show Lemma 49 below we need some preliminary results. Let $f\colon\ \mathbf{R}\to\mathbf{R}$ be a differentiable strongly monotone decreasing function with $f(0)=a$ and $f(b)=0$, for $a,b\in\mathbf{R}$, and $\int_0^b f(x)\,dx=1$. If there are no explicit integration bounds, integration is taken over $\mathbf{R}$.

Lemma 46 *A function f as above may be transformed to a continuous function $g\colon\ \mathbf{R}\to\mathbf{R}$ with $0\le g'(\zeta)\le 1$, for $\zeta<0$, $-1\le g'(\zeta)\le 0$, for $\zeta>0$, $g(x)=0$ for $x<-a$ and $x>b$, and $\int_{-\infty}^{\infty}g(x)\,dx=1$. Moreover,*

$$H(f)=I(g)=Q(g)+L(g),$$

with

$$Q(g)=1/2\int g'(x)\int g'(t)\ln|x-t|\,dt\,dx, \tag{9}$$

$$L(g)=-\int g'(x)(x\ln|x|-x)\,dx. \tag{10}$$

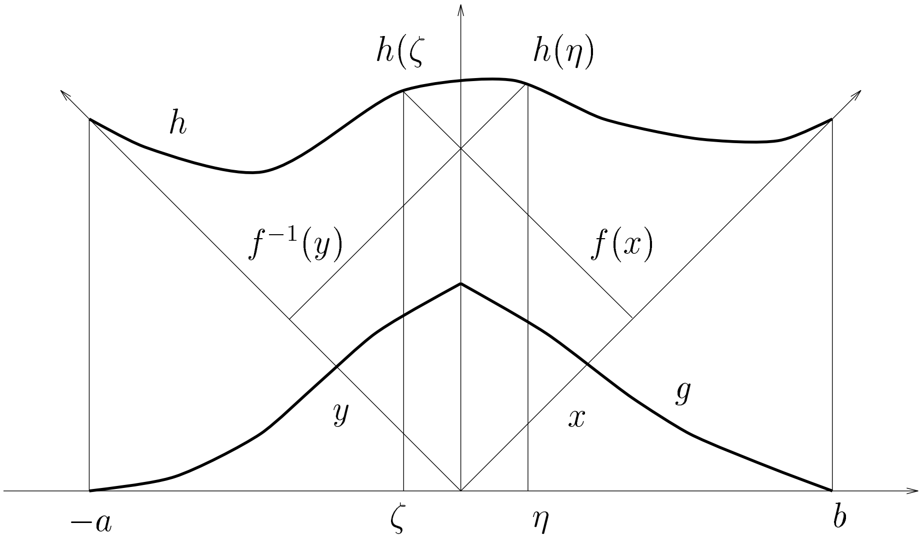

Figure 30: Functions h and g are obtained from f by transformations ρ and D.

Proof. In Section 3.4 we used the transformation $\rho(x,y) = (x-y, x+y)$ on the plane which amounts to a scaling of the x- and y-axis by a factor of $\sqrt{2}$ and a rotation by $\pi/4$. The inverse transformation is $\rho^{-1}(u,v) = ((u+v)/2, (v-u)/2)$ and thus the image of the graph of f under ρ (see Figure 30) is described by function h with

$$f(x) = \frac{h(\zeta) - \zeta}{2} \qquad x = \frac{h(\zeta) + \zeta}{2} \tag{11}$$

$$f^{-1}(y) = \frac{h(\eta) + \eta}{2} \qquad y = \frac{h(\eta) - \eta}{2}. \tag{12}$$

Hence, $f(x)-y+f^{-1}(y)-x = \eta-\zeta$, $dx = (h'(\zeta)+1)/2\, d\zeta$, $dy = (h'(\eta)-1)/2\, d\eta$, and

$$\begin{aligned} H(f) &= \int_0^b \int_0^{f(x)} \ln(f(x) - y + f^{-1}(y) - x)\, dy\, dx \\ &= -\frac{1}{4} \int_{-a}^b \int_\zeta^b \ln(\eta - \zeta)(h'(\eta) - 1)\, d\eta\, (h'(\zeta) + 1)\, d\zeta \end{aligned}$$

(the sign stems from changing the integration from 0 to $f(x)$ to that from ζ to b). We may now simplify function h by a further transformation D to function

$g = D(h)$ with

$$g(\zeta) = \begin{cases} (h(\zeta)+\zeta)/2 & \zeta < 0, \\ (h(\zeta)-\zeta)/2 & \zeta \geq 0. \end{cases} \tag{13}$$

Consequently,

$$h'(\zeta) = \begin{cases} 2g'(\zeta) - 1 & \zeta < 0, \\ 2g'(\zeta) + 1 & \zeta \geq 0. \end{cases} \tag{14}$$

Since h is obtained from the monotone decreasing function f through a rotation by $\pi/4$, $-1 \leq h'(\zeta) \leq 1$ and the inequalites on g' follow. Similarly, by the scaling in transformation ρ, the area A between the graph of h and the rotated x- and y-axis is twice the area below the graph of f, so A is 2. It is easily seen that the area between g and the ζ-axis is half of A, hence $\int g(\zeta)\,d\zeta$ is 1 again. Now

$$\begin{aligned} -4H(f) = {} & \int_{-a}^{0} 2g'(\zeta)\Big(\int_{\zeta}^{0} \ln(\eta-\zeta)(2g'(\eta)-2)\,d\eta + \int_{0}^{b} \ln(\eta-\zeta)2g'(\eta)\,d\eta\Big)\,d\zeta \\ & + \int_{0}^{b} (2g'(\zeta)+2) \int_{\zeta}^{b} \ln(\eta-\zeta)2g'(\eta)\,d\eta\,d\zeta \\ = {} & 4\int_{-a}^{b} g'(\zeta) \int_{\zeta}^{b} \ln(\eta-\zeta)g'(\eta)\,d\eta\,d\zeta \\ & - 4\int_{-a}^{0} g'(\zeta) \int_{\zeta}^{0} \ln(\eta-\zeta)\,d\eta\,d\zeta + 4\int_{0}^{b} g'(\eta)\int_{0}^{\eta} \ln(\eta-\zeta)\,d\zeta\,d\eta \\ = {} & 4\frac{1}{2}\int_{-a}^{b}\int_{-a}^{b} \ln|\eta-\zeta|\,g'(\zeta)g'(\eta)\,d\eta\,d\zeta + 4\int_{-a}^{b} g'(\zeta)(\zeta\ln|\zeta|-\zeta)\,d\zeta, \end{aligned}$$

using $\int \ln(x)\,dx = x\ln(x) - x$.

□

Note that $g'(x)$ for g in the above lemma is not defined for $x = 0$. We tacitly assume that all integrations involving $g'(x)$ are taken on $\mathbf{R}\backslash\{0\}$.

Recall that the Fourier transform $\widehat{g}$ and the Hilbert transform $\widetilde{g}$ of g are defined by

$$\begin{aligned} \widehat{g}(x) &= \int_{-\infty}^{\infty} g(t)\,e^{-ixt}\,dt, \\ \widetilde{g}(x) &= \frac{1}{\pi}\mathbf{P}\int_{-\infty}^{\infty} \frac{g(t)}{x-t}\,dt = \frac{1}{\pi}\lim_{\epsilon\to 0}\Big(\int_{-\infty}^{x-\epsilon} \frac{g(t)}{x-t}\,dt + \int_{x+\epsilon}^{\infty} \frac{g(t)}{x-t}\,dt\Big). \end{aligned}$$

One also says that the latter integral is taken as "principal value".

Lemma 47 *Functional $Q(g)$ for g as in Lemma 46 may be rewritten as*

$$Q(g) = -\frac{\pi}{2}\int g'(x)\tilde{g}(x)\,dx = \frac{1}{4}\int |\widehat{g}(x)|^2|x|\,dx.$$

Proof. Using integration by parts we obtain

$$\int_{-\infty}^{x-\epsilon} g(t)\frac{1}{x-t}\,dt = g(t)(-\ln(x-t))\Big|_{t=-\infty}^{x-\epsilon} - \int_{-\infty}^{x-\epsilon} g'(t)(-\ln(x-t))\,dt,$$
$$-\int_{x+\epsilon}^{\infty} g(t)\frac{1}{t-x}\,dt = -g(t)\ln(t-x)\Big|_{t=x+\epsilon}^{\infty} + \int_{x+\epsilon}^{\infty} g'(t)\ln(t-x)\,dt,$$

hence (since $g(x)$ is zero outside $[-a,b]$),

$$\mathbf{P}\int g(t)\frac{1}{x-t}\,dt = \lim_{\epsilon\to 0}\Big(-g(x-\epsilon)\ln\epsilon + g(x+\epsilon)\ln\epsilon\Big) + \int g'(t)\ln|x-t|dt.$$

By the mean value theorem, for some $\zeta \in [x-\epsilon, x+\epsilon]$,

$$\lim_{\epsilon\to 0}\Big(-g(x-\epsilon)\ln\epsilon + g(x+\epsilon)\ln\epsilon\Big) = \lim_{\epsilon\to 0}(-g'(\zeta)\epsilon\ln\epsilon) = 0,$$

since $|g'(\zeta)| \le 1$ for all ζ. The first equality follows.

By partial integration, one easily computes the Fourier transform of g' as $ix\widehat{g}(x)$. The Fourier transform of $\tilde{g}$ is $-i\,\mathrm{sgn}(x)\widehat{g}(x)$ (see[1] Titchmarsh[29], p. 120). Furthermore, if z^* denotes the complex conjugate of a number $z \in \mathbf{C}$,

$$\int g_1 g_2^* = \frac{1}{2\pi}\int \widehat{g_1}(\widehat{g_2})^* \tag{15}$$

(Parseval's identity, Titchmarsh[29], p. 73). Since $\tilde{g}(x)$ is real,

$$\int g'(x)\tilde{g}(x)\,dx = \int g'(x)\tilde{g}(x)^*\,dx = \frac{1}{2\pi}\int \widehat{g'}(x)(\widehat{\tilde{g}}(x))^*\,dx$$
$$= \frac{1}{2\pi}\int i\,x\widehat{g}(x)(i\,\mathrm{sgn}(x)\widehat{g}(x)^*)\,dx = -\frac{1}{2\pi}\int |x||\widehat{g}(x)|^2\,dx.$$

□

[1] Logan and Shepp give the false sign here. Note also that Titchmarsh uses a different definition for the Fourier transform.

For the proof of the following lemma see p. 214ff of Logan and Shepp [17]).

Lemma 48 *The function*

$$g_0(x) = \begin{cases} \frac{2}{\pi}\sqrt{1-x^2/4} + \frac{x}{\pi}\arcsin(x/2) - \frac{|x|}{2}, & \textit{if } |x| \le 2 \\ 0, & \textit{if } |x| \ge 2 \end{cases}$$

has Hilbert transform

$$\widetilde{g_0}(x) = \frac{x}{\pi} - \frac{x}{\pi}\ln|x|, \;\; \textit{if } |x| \le 2.$$

Furthermore, g_0 has the properties of g as stated in Lemma 46 and for any g with these properties

$$I(g) \ge -Q(g_0) + Q(g - g_0).$$

According to the considerations at the beginning of this section, we are done when we finally show

Lemma 49 *For $\epsilon > 0$ and $\lambda \in L(\epsilon)$, $H(f_\lambda) \ge -1/2 + \delta(\epsilon)$, where $\delta(\epsilon) > 0$ depends on ϵ only.*

Proof. By the definition of $L(\epsilon)$ (see Equation 8) we know that $\sup_{x\in\mathbf{R}} |h_\lambda(x) - h_0(x)| \ge \epsilon$ and with $g = D(h_\lambda)$ and $g_0 = D(h_0)$ (where D is the transformation given in Equation 13) also $\sup_{x\in\mathbf{R}} |g(x) - g_0(x)| \ge \epsilon/2$. This inequality will be used in the sequel.

We first show that $Q(g_0) = 1/2$. By Lemma 47,

$$Q(g_0) = \frac{\pi}{4}\int_{-2}^{2}\left(\frac{2\arcsin(x/2)}{\pi} - \operatorname{sgn} x\right)\left(\frac{x}{\pi} - \frac{x}{\pi}\ln|x|\right)\,dx.$$

This integral is easily simplified by repeated partial integration and we leave this task to the reader. The only problem may arise with the following expression for which we give a hint how to solve it.

$$\int_0^1 \sqrt{1-x^2}\ln(x)\,dx = \int_0^{\pi/2}\sqrt{1-(\sin y)^2}\ln(\sin y)\cos y\,dy,$$

by substituting $x = \sin y$. Furthermore, by partial integration with $u = \cos x$ and $v' = \ln(\sin x) \cos x$,

$$\begin{aligned}\int_0^{\pi/2} & \ln(\sin x) \cos x \cos x \, dx \\ &= \int_0^{\pi/2} \ln(\sin x) \, dx - \int_0^{\pi/2} (\cos x)^2 \ln(\sin x) \, dx - \int_0^{\pi/2} (\sin x)^2 \, dx.\end{aligned}$$

Consequently,

$$2 \int_0^{\pi/2} \ln(\sin x)(\cos x)^2 \, dx = \int_0^{\pi/2} \ln(\sin x) \, dx - \frac{\pi}{4} = -\frac{\pi}{2} \ln 2 - \frac{\pi}{4},$$

where we used $\int_0^{\pi/2} \ln(\sin x) \, dx = -(\pi/2) \ln 2$ which may be found, e.g., in Lewin [16], p. 103.

Let g_1 denote $g - g_0$. Hence, $\sup_{x \in \mathbf{R}} |g_1(x)| > \epsilon/2$. We derive a lower bound on $Q(g_1)$. Observe that $\int |g(x)| \, dx = \int |g_0(x)| \, dx = 1$. Hence,

$$|\widehat{g_1}(x)|^2 = \Big| \int g_1(t) \, e^{-ixt} \, dt \Big|^2 \leq \Big(\int |g_1(t)| \, |e^{-ixt}| \, dt \Big)^2 \leq 4. \tag{16}$$

Furthermore, by the properties of g' and g_0' as given in Lemma 46, $|g_1'(x)| \leq 1$. On the other hand, there exists some x_0 with $|g_1(x_0)| \geq \epsilon/2$. Since g_1 is not allowed to increase or decrease with a slope greater than 1, $|g_1(x)|$ dominates the function $\Delta(x) = \max\{-|x - x_0| + \epsilon/2, 0\}$ and so

$$\int |g_1(x)|^2 \, dx \geq \int \Delta(x)^2 \, dx = 2 \int_0^{\epsilon/2} x^2 \, dx = \frac{\epsilon^3}{12}. \tag{17}$$

Finally, we derive

$$\begin{aligned}\int_{-\infty}^{\infty} |\widehat{g_1}(x)|^2 |x| \, dx &\geq a \int_{-\infty}^{a} |\widehat{g_1}(x)|^2 \, dx + a \int_{-a}^{a} |\widehat{g_1}(x)|^2 \, dx + a \int_{a}^{\infty} |\widehat{g_1}(x)|^2 \, dx \\ &\qquad - a \int_{-a}^{a} |\widehat{g_1}(x)|^2 \, dx + a \int_{-a}^{a} |\widehat{g_1}(x)|^2 |x| \, dx \\ &\geq a \Big(\int_{-\infty}^{\infty} |\widehat{g_1}(x)|^2 \, dx - \int_{-a}^{a} |\widehat{g_1}(x)|^2 \, dx \Big) \\ &\geq a \Big(2\pi \int_{-\infty}^{\infty} |g_1(x)|^2 \, dx - 8a \Big) \geq a \Big(\frac{2\pi}{12} \epsilon^3 - 8a \Big),\end{aligned}$$

using Equations 16 and 17 and Parseval's Identity. Setting $a = \pi\epsilon^3/48$ we have the inequality

$$Q(g_1) = \int |\widehat{g_1}(x)|^2 |x| \, dx \geq \pi^2\epsilon^6/576,$$

and the lemma is shown by recalling that (see Lemmas 46 and 48)

$$H(f_\lambda) = I(g) \geq -Q(g_0) + Q(g - g_0) \geq -\frac{1}{2} + \frac{\pi^2\epsilon^6}{576}.$$

□

References

[1] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms.* MIT Press, 1989.

[2] I. Dagan, M. Golumbic, and R. Pinter. Trapezoid graphs and their coloring. *Discr. Appl. Mathematics*, 21:35–46, 1988.

[3] G. de B. Robinson. On the representation of the symmetric groups. *American J. Math.*, 60:745–760, 1938.

[4] R. P. Dilworth. A decomposition theorem for partially ordered sets. *Ann. of Math.*, 51:161–165, 1950.

[5] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. of the ACM*, 19:248–264, 1972.

[6] S. Felsner, R. Müller, and L. Wernisch. Trapezoid graphs and generalizations, geometry and algorithms. In *Proc. 4. Scand. Workshop on Alg. Theory*, 1994. to appear in *SIAM J. of Computing.*

[7] S. Felsner and L. Wernisch. Maximum k-chains in planar point sets: Combinatorial structure and algorithms. In *Proc. 25. Ann. ACM Symp. on the Theory of Computing*, pages 146–153, 1993. to appear in *Discr. Appl. Mathematics.*

[8] J. S. Frame, G. de B. Robinson, and R. M. Thrall. The hook graphs of the symmetric group. *Canad. J. Math.*, 6:316–325, 1954.

[9] A. Frank. On chain and antichain families of partially ordered sets. *J. Comb. Th. (B)*, 29:176–184, 1980.

[10] M. L. Fredman. On computing the length of longest increasing subsequences. *Discr. Math.*, 11:29–35, 1975.

[11] F. Gavril. Algorithms for maximum k-colorings and maximum k-coverings of transitive graphs. *Networks*, 17:465–470, 1987.

[12] C. Greene. An extension of Schensted's theorem. *Adv. in Math.*, 14:254–265, 1974.

[13] C. Greene. Some partitions associated with a partially ordered set. *J. Comb. Th. (A)*, 20:69–79, 1976.

[14] J. M. Hammersley. A few seedlings of research. In *Proc. 6. Berkeley Symp. on Math. Stat. and Prob.*, volume 1, pages 345–394, 1972.

[15] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers.* Oxford Univ. Press, London, 1954.

[16] L. Lewin. *Polylogarithms and associated functions.* North Holland, New York, 1981.

[17] B. F. Logan and L. A. Shepp. A variational problem for random Young tableaux. *Advances in Math.*, 26:206–222, 1977.

[18] R. D. Lou and M. Sarrafzadeh. An optimal algorithm for the maximum 3-chain problem. to appear in SIAM J. on Comp., 1992.

[19] R. D. Lou, M. Sarrafzadeh, and D. T. Lee. An optimal algorithm for the maximum two-chain problem. In *Proc. of the 1. Ann. ACM-SIAM Symp. on Discr. Alg.*, pages 149–158, 1990.

[20] T. Ma and J. Spinrad. An $O(n^2)$ time algorihtm for the 2-chain cover problem and related problems. In *Proc. 2. Ann. ACM Symp. on Discrete Algorithms*, pages 363–372, 1991.

[21] J. Matoušek and E. Welzl. Good splitters for counting points in triangles. *J. of Algorithms*, 13:307–319, 1992.

[22] K. Mehlhorn. *Data Structures and Algorithms 3. Multi-dimensional Searching and Computational Geometry*, volume 3. Springer, Berlin Heidelberg New York, 1984.

[23] K. Mehlhorn and S. Näher. Dynamic fractional cascading. *Algorithmica*, 5:215–241, 1990.

[24] S. Pilpel. Descending subsequences of random permutations. *J. of Comb. Theory (A)*, 53:96–116, 1990.

[25] M. Sarrafzadeh and R. D. Lou. Maximum k-covering of weighted transitive graphs with applications. *Algorithmica*, 9:84–100, 1993.

[26] C. Schensted. Longest increasing and decreasing sequences. *Canad. J. Math.*, 13:179–191, 1961.

[27] M. Schützenberger. La correspondence de Robinson. In D. Foata, editor, *Combinatoire et Représentation du Groupe Symétrique*, pages 59–113. Lecture Notes in Mathematics 579, Berlin Heidelberg New-York, 1977.

[28] M. Smid. Range trees with slack parameter. *Algorithms Review*, 2(2):77–87, 1991.

[29] E. E. Titchmarsh. *Introduction to the Theory of Fourier Integrals*. Oxford Univ. Press, London, 2 edition, 1948.

[30] P. van Emde Boas. Preserving order in a forest in less than logarithmic time and linear space. *Inf. Proc. Letters*, 6:80–82, 1977.

[31] A. M. Veršik and S. V. Kerov. Asymptotics of the plancherel measure of the symmetric group and the limiting form of Young tables. *Soviet Mathematics, Dokl. Akad. Nauk. SSSR*, 18(1):527–531, 1977.

[32] G. Viennot. Une forme géométrique de la correspondance de Robinson-Schensted. In D. Foata, editor, *Combinatoire et Représentation du Groupe Symétrique*, pages 29–58. Lecture Notes in Mathematics 579, Berlin Heidelberg New-York, 1977.

[33] G. Viennot. Chain and antichain families, grids and young tableaux. In M. Pouzet, editor, *Orders: Description and Roles*, pages 409–463. North-Holland Math. Stud. 99, Amsterdam-New York, 1984.

Index

Zusammenfassung

Gegeben seien n Punkte in der Ebene. Dabei *dominiert* ein Punkt x einen anderen y, wenn x sich, anschaulich gesprochen, rechts oben von y befindet. Eine *Kette in* X ist eine Menge von Punkten aus X in der einer von je zweien den anderen dominiert. eine *maximale* k*-Kette* ist eine Vereinigung von jeweils k Ketten, die die meisten Elemente unter allen solchen Vereinigungen hat. Die Arbeit beschreibt einen Algorithmus, der eine solche Kette in Zeit $O(kn \log n)$ berechnet. Gibt man den Punkten Gewichte und fragt nach einer maximalen gewichteten k-Kette, so wird die Berechnung schwieriger. Es läßt sich dann aber ein für kleine k immer noch effizienter Algorithmus angeben.

Diese Algorithmen beruhen auf Strukturen, die der Robinson-Schensted Korrespondenz von Young Tableaux und Permutationen zugrundeliegen, und durch Viennots Konstruktion von *Skeletten* geometrisch anschaulich gemacht werden können. Die Arbeit gibt eine Einführung in diese Theorie. Klassische Resultate werden durchgehend geometrisch bewiesen. Insbesondere führt die Beantwortung von Viennots Frage nach einem geometrischen Beweis für Schensteds Theorem auf eine interessante Eigenschaft von Skeletten.

Was ist die erwartete Größe einer maximalen k-Kette in einer zufälligen Permutation der Länge n, wenn n groß wird? Es wird gezeigt, wie bekannte Resultate für den Fall einfacher Ketten auf k-Ketten ausgedehnt werden können.

Der Begriff der Dominanzordnung bleibt sinnvoll, wenn man Punkte in höherdimensionalen Räumen betrachtet, oder Punkte durch achsenparallele Rechtecke ersetzt. Für diese Gebilde wird selbst die Berechnung einfacher Ketten schwierig. Allerdings ist die Berechnung von Ketten und *Antiketten* (d.h. von Teilmengen, worin kein Punkt irgendeinen anderen dominiert) für zweidimensionale Mengen von Rechtecken in optimaler Zeit möglich. Die Algorithmen arbeiten auch für gewichtete Versionen des Problems.

Lebenslauf

Lorenz Wernisch, geboren 3.3.1961 in Klagenfurt, Österreich.

1967–1971 Volksschule in Klagenfurt.
1971–1979 Gymnasium in Klagenfurt.
1979 Matura (Abitur).
1979–1982 Versuche mit Studium der Medizin und Philosophie an der Universität Wien.
1982–1989 Studium der Mathematik an der Universität Wien.
1989 Diplom in Mathematik.
1994 Promotion.

Seit 1990 Wissenschaftlicher Mitarbeiter in der Gruppe von Prof. Emo Welzl am Fachbereich für Mathematik und Informatik an der Freien Universität Berlin.

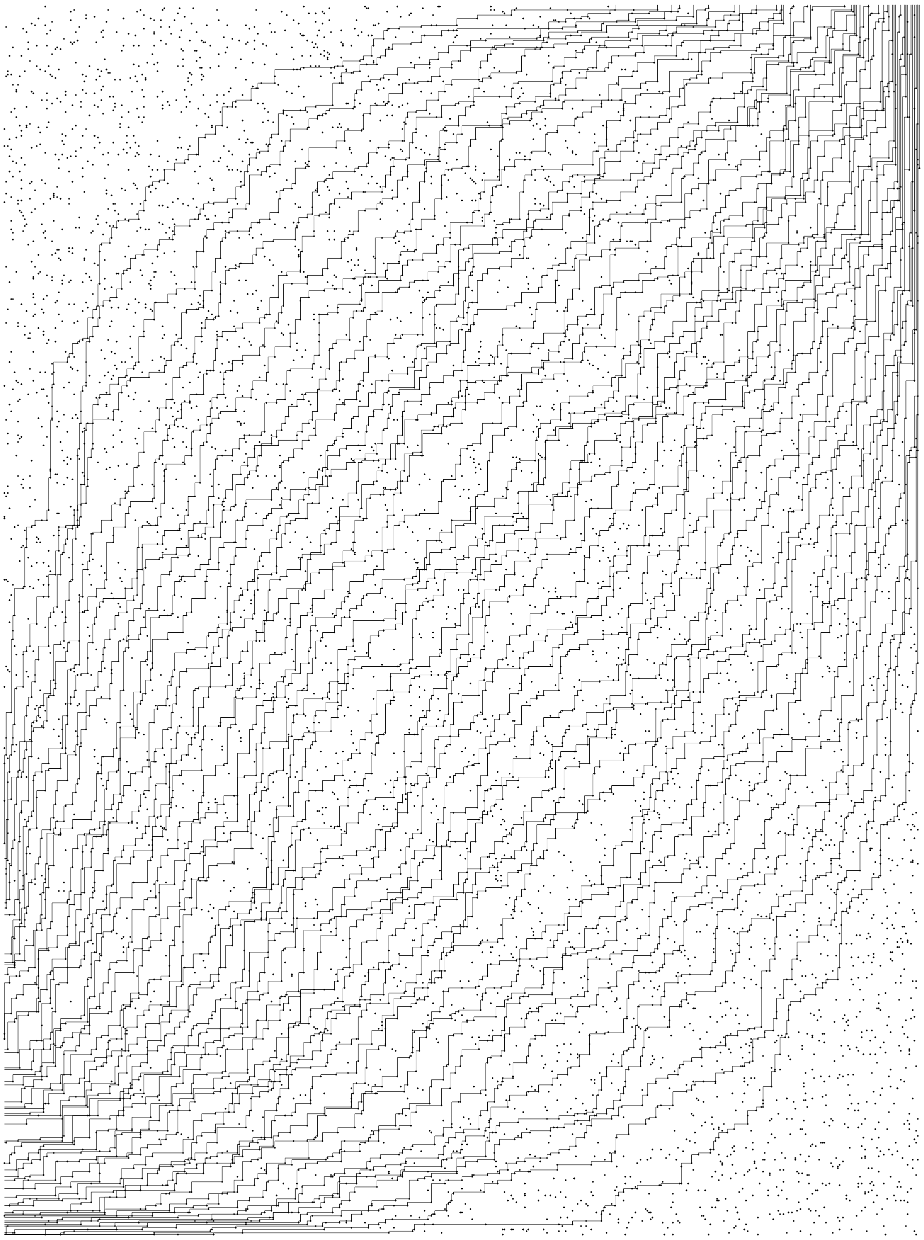

Abbildung 31: A maximum 50-chain of size 6069 among 10000 random points.