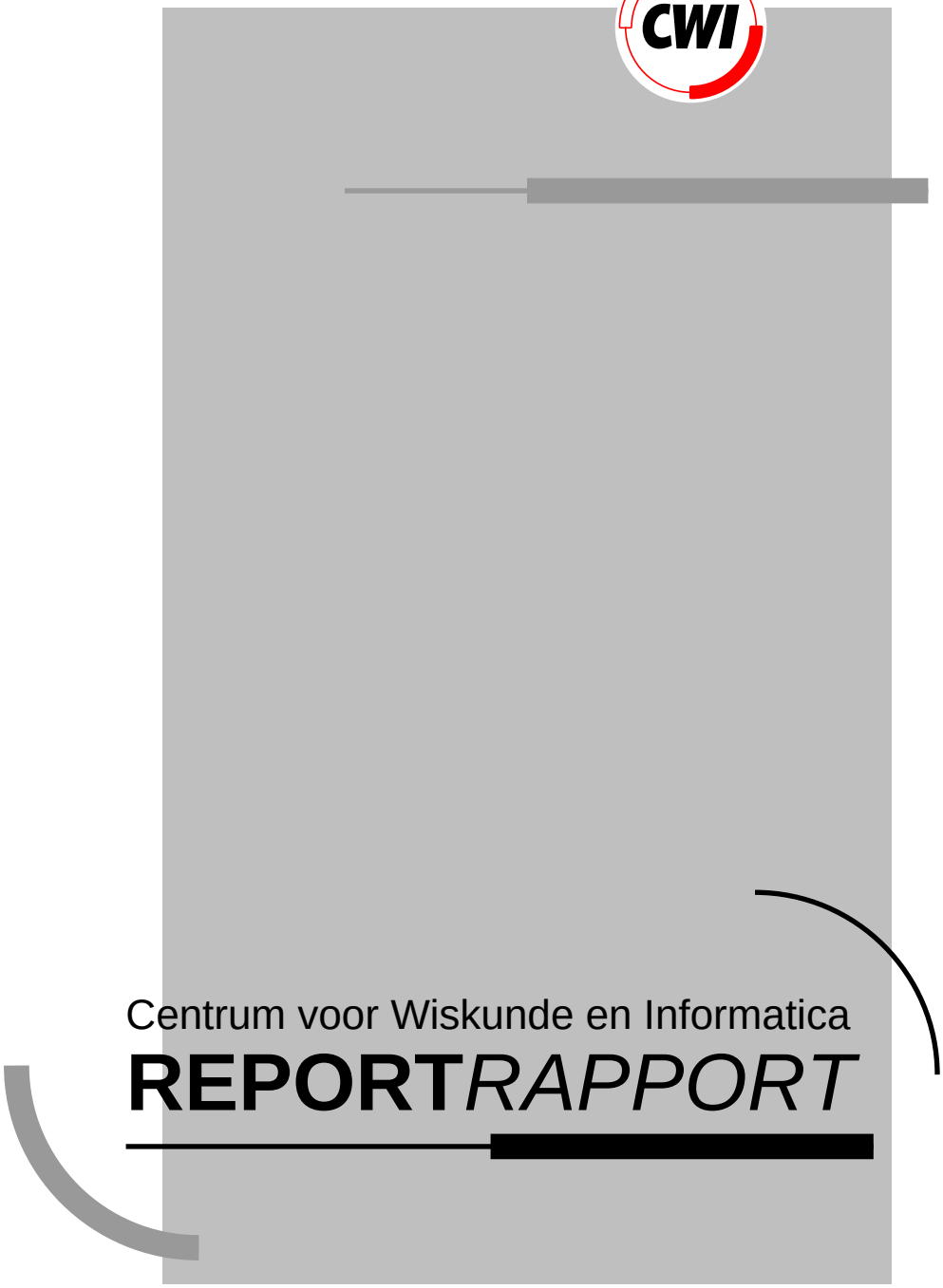


Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

The Robinson-Schensted and Schützenberger Algorithms Part I: New Combinatorial Proofs

M.A.A. van Leeuwen

Department of Analysis, Algebra and Geometry

AM-R9208 1992

The Robinson-Schensted and Schützenberger Algorithms

Part I: New Combinatorial Proofs

Marc A. A. van Leeuwen

CWI

P. O. Box 4079, 1009 AB Amsterdam, The Netherlands

email: M.van.Leeuwen@cwi.nl

Abstract

A new and self-contained approach to the Robinson-Schensted and Schützenberger algorithms is given, including elementary proofs for many of their combinatorial properties and representations of the algorithms by graphical computations. An analogue of the Robinson-Schensted algorithm for the hyperoctahedral groups instead of the symmetric groups is described, as well as relations of the algorithms with Schützenberger's theory of *glissements*, and the connection between the versions of the Robinson-Schensted algorithm defined by Robinson and Schensted respectively.

1991 Mathematics Subject Classification: 05E10.

Key Words and Phrases: Young tableaux, combinatorial algorithms, bijective proofs.

General introduction.

The two algorithms referred to in our title are combinatorial algorithms dealing with Young tableaux. The former was found originally by G. de B. Robinson [Rob], and independently rediscovered much later, and in a different form, by C. Schensted [Sche]; it establishes a bijective correspondence between permutations and pairs of Young tableaux of equal shape. The latter algorithm (which is sometimes associated with the term *jeu de taquin*) was introduced by M. P. Schützenberger [Schü1], who also demonstrated its great importance in relation to the former algorithm; it establishes a shape preserving involutive correspondence between Young tableaux. These algorithms have been studied mainly for their own sake—they exhibit quite remarkable combinatorial properties—rather than primarily serving (as is usually the case with algorithms) as a means of computing some mathematical value. In this paper we propose to improve the understanding of these two algorithms in two distinct but interrelated ways, and the paper is correspondingly divided into two parts. In Part I we give a novel approach to the combinatorial theory of the algorithms, which encompasses many of the known results about them, but treats them in a more systematic and rigorous manner. In Part II we give important interpretations of the algorithms (of which that of the Robinson-Schensted algorithm is due to R. Steinberg [Stb]), i.e., we give a mathematical context in which the algorithms do in fact provide computational answers to certain questions, which arise in the geometric study of the unipotent variety of the groups $\mathbf{GL}_n$.

Throughout the paper our main goal is a better understanding of the algorithms and their properties within a coherent framework, rather than generalising or finding new facts about them. This does not mean that we only prove facts known before, but it does mean that we shall pay much attention to the reasoning. Rather than attempting to obtain proofs with minimal effort, we shall try to clarify their structure and essential ingredients. In fact we will sometimes reconsider a known result from another

angle, or consciously refrain from using certain results (either from earlier in the paper or from the literature) which are not essential to the proof, in order to avoid unnecessary logical dependencies; thus relationships between different facts can be brought out more clearly. We shall also indicate connections with certain alternative approaches in the literature, although we cannot claim to be comprehensive on this point, in particular not with respect to the many recent publications on the subject.

For expository reasons the combinatorial part of this paper precedes the geometric interpretations; however, we arrived at studying these matters in the opposite order. In [vLee1] we formulated an extension of Steinberg's geometric interpretation of the Robinson-Schensted algorithm to the case of other classical groups (this extension lies beyond the scope of the current paper). From this it became clear that for $\mathbf{GL}_n$ a geometric interpretation could also be given for the Schützenberger algorithm, as shown in Part II (a brief account can be found in [vLee2], which already contains certain fragments of the current paper, but does not use its formalism). On the other hand the geometric problems enforced a view of the algorithms (viewing Young tableaux as chains of partitions, and the algorithms themselves as being defined recursively), which proved to be quite useful for a purely combinatorial understanding as well; from this observation Part I has evolved. Central in both parts of our paper is a formalism for defining the various algorithms recursively, which is introduced in §2 of Part I: it is made into a cornerstone of the combinatorial treatment (Part I), and also provides the proper framework for the proofs of the geometric interpretations (Part II). The geometric interpretations automatically provide (non-combinatorial) proofs to the main theorems of Part I of this paper, thereby shedding light on what might otherwise appear to be mysterious coincidences.

Introduction to Part I.

The first part of this paper is devoted to the purely combinatorial theory of the Robinson-Schensted and Schützenberger algorithms. The former is the oldest of the two, and was first described in 1938 by Robinson [Rob], in a paper dealing with the representation theory of the symmetric group; that description is rather obscure however, and apart from reference to it in later publications of the same author, it does not appear to have received much mention in the literature of the subsequent decades. The great interest the algorithm enjoys nowadays by combinatorialists was triggered by its independent reformulation by Schensted [Sche] published in 1961, whose main objective was counting permutations with given lengths of their longest increasing and decreasing subsequences. The combinatorial significance of the algorithm was indicated by Schützenberger [Schü1]: he stated a number of important identities satisfied by the Robinson-Schensted correspondence, in conjunction with the correspondence given by the Schützenberger algorithm, which he introduced in the same paper (the operation called I in his §5). In that paper these prominent results are somewhat obscured however, by the fact that its emphasis lies on treating the limiting case of (one sided) infinite sequences of distinct numbers instead of permutations, and correspondingly infinite Young tableaux, a generalisation which has been ignored in the further development of the subject. Another significant contribution is due to D. E. Knuth [Kn1], who gives a generalisation where repetitions are allowed in both tableaux (and permutations are correspondingly generalised), as well as a description of the classes of (generalised) permutations obtained by fixing one of the two tableaux. Knuth has probably also contributed considerably to the popularity of the algorithms by his exposition in [Kn2]. Schensted's theorem about increasing and decreasing subsequences is extended by C. Greene [Gre1], to give a direct interpretation of the shape of the Young tableaux corresponding to a permutation, and in [Schü2] a completely new approach is presented, based on the results of Knuth and Greene, in which the basic procedure of the Schützenberger algorithm plays a central rôle, rather than Schensted's construction.

Although since that point few new results have been added to the basic combinatorial theory of the algorithms, there have been new developments in several directions. One of these is based upon enumerative identities in representation theory which correspond to the Robinson-Schensted algorithm and its generalisation by Knuth. The ordinary Robinson-Schensted correspondence gives an identity which counts dimensions in the decomposition of the group algebra of $\mathbf{S}_n$ as representation of $\mathbf{S}_n \times \mathbf{S}_n$ (action from left and right). Knuth's generalisation (which at first glance does not add very much,

since it can be made to factor via the ordinary algorithm in an obvious way) leads to identities which describe the decomposition into irreducibles of $V^{\otimes n}$ as representation of $\mathbf{GL}(V) \times \mathbf{S}_n$, respectively the decomposition of $k[V \otimes W]$ as representation of $\mathbf{GL}(V) \times \mathbf{GL}(W)$; moreover they actually describe how the dimension of each individual weight space with respect to (a maximal torus in) $\mathbf{GL}(V)$ or $\mathbf{GL}(W)$ is distributed among the irreducible components. This has led to successful attempts to find variants of the Robinson-Schensted algorithm (often also termed Robinson-Schensted-Knuth algorithm) which are similarly related to the representation theory of other groups, see [Sag1], [Bere], [Sund1], [Stem], [Sund2], [Pro], [BeSt], [Oka2], [Ter]. A survey of a number of these generalisations can be found in [Sag2].

Another development centers around the observation that the definition of the Robinson-Schensted algorithm depends only on a few basic properties of the Young lattice, and that a large part of the theory can be developed similarly for other partially ordered sets which share these properties. The observation appears to have been made independently by S. V. Fomin [Fom2] and R. P. Stanley (who termed these sets 'differential posets') [Stan3]. The approach of the former is based on results from the study of finite partially ordered sets, which are closely related to the results of Greene, and it leads to explicit bijective algorithms; the latter approach is enumerative in nature, and leads to very general identities valid in arbitrary differential posets (efficiently formulated using a powerful machinery, involving such things as formal power series in non-commuting linear operators), but it is not mentioned whether corresponding bijections can be automatically derived from them. The two approaches are combined and extended in a series of recent papers [Roby], [Fom3], [Fom4], [FomSt].

Furthermore there is a number of papers in which a connection is made between the Robinson-Schensted algorithm and the Littlewood-Richardson rule, see [Schü2], [Thom2], [Whi], [Zel], which is a subject with a tradition, since it occurs already in Robinson's original paper. Finally, there are at least two instances of interpretations of the Robinson-Schensted correspondence in subjects outside combinatorics (which might prove to give the best explanation as to why some specific permutation should correspond to some specific pair of tableaux), namely an algebraic interpretation in terms of primitive ideals in envelopping algebras (see [Jos], [Vog], Theorem 6.5), or equivalently cells in Coxeter groups as defined in [KaLu], and a geometric interpretation in terms of subvarieties of the flag manifold [Stb]; the latter is treated in detail in Part II of the current paper.

In the current paper we do not pursue any generalisations or variants of the basic algorithms, except those which arise naturally from their study. Rather we develop the fundamental theory of the algorithms in a uniform and systematic way; in the present literature this theory is rather scattered and diverse in its methods and notations. We systematically interpret Young tableaux as chains of partitions, which leads to recursive forms of the algorithms; these are specified using a carefully chosen formalism (which is similar to a (functional) programming language), enabling the formulation of clear and rigorous proofs. We also give insightful pictorial representations of the proofs in which the symmetries stated by the theorems are also present in the representations of the proofs; in this way the right brain hemisphere shall be served equally well as the left. An additional advantage is that it becomes quite easy to understand the fixed points of the stated symmetries; in particular we derive a Robinson-Schensted algorithm for the hyperoctahedral groups, which is the natural combinatorial analogue of the ordinary Robinson-Schensted algorithm for the symmetric groups. Although most of these results have already been found by other methods, they come about very naturally here. We do not treat all the known properties of the algorithms: we focus on algebraic identities satisfied by the correspondences defined by them. Not mentioned are for instance the elementary transformations which keep one tableau invariant, described by Knuth [Kn1], nor the order theoretic interpretation of the Robinson-Schensted correspondence by Greene [Gre1]; in our approach their results are not required in the proofs of the algebraic identities.

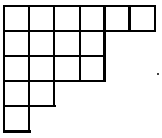
Since the form of the definitions of the algorithms is one of the main issues (and we will in fact meet several definitions for either algorithm), the discussion will start from the very basics, requiring in Part I hardly any prerequisites at all (only the principle of mathematical induction, one might say). The 7 sections of this part treat the following subjects. In §1, after defining the necessary combinatorial notions, we prove some simple purely enumerative propositions to whet the reader's appetite; they will be seen to be directly related to the Robinson-Schensted algorithm. In §2 the Robinson-Schensted algorithm is defined, first a traditional "cookbook style", then in a more formal recursive way, after introducing the

required formalism. The theorem expressing its most fundamental symmetry is given in §3, followed by a discussion of how the proof can be visualised pictorially, and by a consideration of the special case of the fixed points of the symmetry. In §4 we give a similar treatment (on all points) of the Schützenberger algorithm. In §5 we formulate and prove the most important theorem, which relates the two algorithms to each other, again followed by a visualisation and a consideration of fixed points, which in this case leads to a related algorithm for the hyperoctahedral groups. In §6 we elaborate on the consequences of this third main theorem, showing in particular how it can be used to understand Schützenberger's theory of '*glissements*', set forth in [Schü2]. Finally, in §7, we explain why the Robinson-Schensted algorithm is named after Robinson as well as after Schensted.

§1. Some simple enumerative combinatorics.

A *partition* λ of some $n \in \mathbf{N}$ is a weakly decreasing sequence '$\lambda_1 \geq \lambda_2 \geq \cdots$' of natural numbers, that ends with zeros, and whose sum $|\lambda| = \sum_i \lambda_i$ equals n. The terms λ_i of this sequence are called the *parts* of the partition. Although conceptually partitions are infinite sequences, the trailing zeros are usually suppressed, so we write $\lambda = (\lambda_1, \ldots, \lambda_m)$ if $\lambda_i = 0$ for $i > m$. We denote by $\mathcal{P}_n$ the (obviously finite) set of all partitions of n, and by $\mathcal{P}$ the union of all $\mathcal{P}_n$ for $n \in \mathbf{N}$.

To each $\lambda \in \mathcal{P}_n$ is associated an n-element subset of $\mathbf{N}_{>0} \times \mathbf{N}_{>0}$, called its *Young diagram* $Y(\lambda)$; it is defined by $(i,j) \in Y(\lambda) \iff j \leq \lambda_i$. The elements of a Young diagram will be called its *squares*, and we may correspondingly depict the Young diagram: the square (i,j) will be drawn in row i and in column j. For instance, for $\lambda = (6,4,4,2,1) \in \mathcal{P}_{17}$ we have

$Y(\lambda) =$  .

Clearly any partition $\lambda \in \mathcal{P}$ is completely determined by $Y(\lambda)$, and it is often convenient to mentally identify the two. In this spirit we shall use set theoretical notations for partitions, that are defined by passing to their Young diagrams: e.g., $\lambda \subseteq \mu$ for $\lambda, \mu \in \mathcal{P}$ is taken to mean $Y(\lambda) \subseteq Y(\mu)$. The set $\mathbf{N}_{>0} \times \mathbf{N}_{>0}$ has a natural partial ordering given by $(i,j) \leq (i',j')$ whenever $i \leq i'$ and $j \leq j'$. A finite subset S of $\mathbf{N}_{>0} \times \mathbf{N}_{>0}$ is a Young diagram if and only if $s \in S$ implies $s' \in S$ for all $s' \leq s$. From this characterisation it is clear that the set of all Young diagrams is closed under transposition (reflection in the main diagonal). We write $(i,j)^t = (j,i)$ for individual squares, and also write λ^t for the partition with $s \in Y(\lambda) \iff s^t \in Y(\lambda^t)$; this is called the transpose partition of λ. Obviously transposition is an involution on each set $\mathcal{P}_n$. The parts of λ^t can be interpreted as the column lengths of $Y(\lambda)$, so that we have $\lambda^t_j = \#\{\, i \mid \lambda_i \geq j \,\}$ (the operator '#' denotes the number of elements of a finite set).

The relation '$\subseteq$' makes $\mathcal{P}$ into a partially ordered set, which is called the *Young lattice*: one easily verifies that any $\lambda, \mu \in \mathcal{P}$ have an infimum and supremum, namely $\lambda \cap \mu$ respectively $\lambda \cup \mu$ (the notation follows the"partitions as diagrams" view). The partial ordering is graded by the subsets $\mathcal{P}_n$ of $\mathcal{P}$: whenever $\lambda \subset \mu$ we have $|\lambda| < |\mu|$, and one can find a chain of intermediate partitions connecting λ with μ which meets every $\mathcal{P}_i$ with $|\lambda| < i < |\mu|$. For $\lambda \in \mathcal{P}$ we introduce notations for the sets of its direct predecessors and successors in this lattice:

$$\lambda^- \stackrel{\text{def}}{=} \{\, \mu \in \mathcal{P}_{|\lambda|-1} \mid \mu \subset \lambda \,\}, \qquad \lambda^+ \stackrel{\text{def}}{=} \{\, \mu \in \mathcal{P}_{|\lambda|+1} \mid \mu \supset \lambda \,\}.$$

Clearly $\mu \in \lambda^-$ is equivalent to $\lambda \in \mu^+$; when it holds, the difference $Y(\lambda) \setminus Y(\mu)$ consists of a single square, which lies both at the end of a row and of a column of $Y(\lambda)$, while it lies one position beyond both the end of a row and of a column of $Y(\mu)$. We call such a square a *corner* of λ, and a *cocorner* of μ (so for $\lambda = (6,4,4,2,1)$ whose diagram is displayed above, we have as corners $(1,6)$, $(3,4)$, $(4,2)$, and $(5,1)$, and as cocorners $(1,7)$, $(2,5)$, $(4,3)$, $(5,2)$, and $(6,1)$). There is a corner in column j of $Y(\lambda)$ if and only if j occurs as a non-zero part of λ, while there is a cocorner in column j if and only if $j-1$ occurs as a part of λ (here we allow zero). Therefore we have the simple but important identity

$$\#\lambda^+ = \#\lambda^- + 1 \qquad \text{for all } \lambda \in \mathcal{P}. \tag{1}$$

Another identity, which is even more obvious than this one, will also be of importance, namely

$$\#(\lambda^+ \cap \mu^+) = \#(\lambda^- \cap \mu^-) \qquad \text{for } \lambda \neq \mu \tag{2}$$

since both sides are clearly 0 unless $|\lambda| = |\mu|$, and even then they can be at most 1, which happens when the equivalent conditions $\lambda^+ \cap \mu^+ = \{\lambda \cup \mu\}$ and $\lambda^- \cap \mu^- = \{\lambda \cap \mu\}$ are satisfied. The conditions (1) and (2) establish the fact that the Young lattice is a 'differential poset' as defined in [Stan3]; since the identities that shall be derived in this section only depend on these two equations, they remain valid when the Young lattice is replaced by any such set.

The principal reason for referring to the elements of a Young diagram $Y(\lambda)$ as squares (rather than as points), is that it allows one to represent maps $f\colon Y(\lambda) \to \mathbf{Z}$ by filling each square $s \in Y(\lambda)$ with the number $f(s)$. We shall call such a filled Young diagram a *Young tableau* (or simply a tableau) of *shape* λ if it satisfies the following condition*, which we shall refer to as the *tableau property*: all numbers are distinct, and they increase along each row and column. If T is a Young tableau of shape λ we write $\lambda = \operatorname{sh} T$; transposing the Young diagram and its entries leads to a tableau of shape λ^t which shall be denoted by T^t. The tableau property is equivalent to the map $f\colon Y(\lambda) \to \mathbf{Z}$ corresponding to the tableau being injective and monotonous (i.e., a morphism of partially ordered sets). It can also be formulated in a recursive way, which focuses on one square at a time. It is based on the following simple observation.

1.1. Proposition. *Let T consist of a diagram $Y(\lambda)$ filled with integer numbers. Then T is a Young tableau if and only if either*

(i) $\lambda = (0)$, *or*

(ii) the highest entry occurring in T appears in a unique square s, which is a corner of λ, and the restriction of T to $Y(\lambda) \setminus \{s\}$ is a Young tableau.

Proof. It is immediate from the tableau property that in a non-empty tableau the highest entry must be unique and occur at the end of a row and a column, whence the conditions of the proposition are necessary (note incidentally that s being a corner of λ is a prerequisite for the final statement of (ii) to make sense). An equally elementary verification shows that the conditions are sufficient. □

Since the tableau referred to in 1.1(ii) is strictly smaller than T, it is clear that the proposition can be used as a recursive characterisation of Young tableaux. This way of viewing tableaux will be central to our approach. We introduce operators $\lceil\cdot\rceil$ and $\cdot^-$ on non-empty tableaux, by defining $\lceil T \rceil$ to be the corner of $\operatorname{sh} T$ containing the highest entry of T (the square s in the proposition), and T^- to be the tableau obtained by removing that square (with its entry) from T (the restriction of T mentioned in the proposition). If we repeatedly apply the operator $\cdot^-$ to any tableau T, we eventually reach the empty tableau, and the sequence the shapes of the intermediate tableaux form a decreasing chain

$$\operatorname{ch} T \stackrel{\text{def}}{=} \left(\operatorname{sh} T, \operatorname{sh} T^-, \operatorname{sh} T^{--}, \ldots, (0)\right)$$

in the Young lattice. Conversely, from any sequence $\lambda, \lambda', \lambda'', \ldots, (0)$ in $\mathcal{P}$ with $\lambda^{(i+1)} \in (\lambda^{(i)})^-$ for all i, a tableau T may be constructed for which the sequence equals $\operatorname{ch} T$. This can be done by assigning the number $|\lambda|$ to the unique square in $Y(\lambda) \setminus Y(\lambda')$, and filling the remaining squares (i.e., the set $Y(\lambda')$) in the same way according to the shorter sequence $\lambda', \lambda'', \ldots, (0)$.

We define two tableaux T, T' to be *similar* (written $T \sim T'$) when $\operatorname{ch} T = \operatorname{ch} T'$. This is the case if T' can be obtained from T by renumbering the entries in an order preserving way (for the corresponding maps $f, f'\colon Y(\lambda) \to \mathbf{Z}$ this means $f' = g \circ f$ for some monotonous $g\colon \mathbf{Z} \to \mathbf{Z}$). We call T *normalised* if its set of entries (i.e., $\operatorname{Im} f$) equals $\{1, 2, \ldots, |\lambda|\}$, and define $\mathcal{T}_\lambda$ to be the set of normalised Young

* The term tableau is used by different authors for quite different classes of filled Young diagrams, and several adjectives are used to indicate subclasses, in particular *standard*; unfortunately its meaning is not standard. We do not wish to declare our own standard, and follow the terminology of [Kn2]; we do not need any tableaux more general than this.

tableaux of shape λ. Clearly '$\sim$' is an equivalence relation, and every equivalence class contains a unique normalised element. As an example we have

$$T = \begin{array}{|c|c|c|}\hline 3&6&11\\\hline 5&8\\\cline{1-2} 7\\\cline{1-1} 19\\\cline{1-1}\end{array} \qquad \sim \qquad T' = \begin{array}{|c|c|c|}\hline 1&3&6\\\hline 2&5\\\cline{1-2} 4\\\cline{1-1} 7\\\cline{1-1}\end{array} \quad \in \mathcal{T}_{(3,2,1,1)},$$

since we have

$$\operatorname{ch} T = \left(\begin{array}{|c|c|c|}\hline \\\hline \\\cline{1-2} \\\cline{1-1} \\\cline{1-1}\end{array}\ , \ \begin{array}{|c|c|c|}\hline \\\hline \\\cline{1-2} \\\cline{1-2}\end{array}\ , \ \begin{array}{|c|c|}\hline \\\hline \\\cline{1-2}\end{array}\ , \ \begin{array}{|c|c|}\hline \\\hline \\\cline{1-1}\end{array}\ , \ \begin{array}{|c|c|}\hline \\\hline \\\cline{1-2}\end{array}\ , \ \begin{array}{|c|}\hline \\\hline \\\cline{1-1}\end{array}\ , \ \begin{array}{|c|}\hline \\\hline\end{array}\ , \ \emptyset \right) = \operatorname{ch} T'.$$

Since $\mathcal{T}_\lambda$ is in bijection with the set of saturated decreasing chains in the Young lattice starting in λ, we have the identity:

$$\#\mathcal{T}_\lambda = \sum_{\mu \in \lambda^-} \#\mathcal{T}_\mu \qquad \text{for all partitions } \lambda \neq (0). \tag{3}$$

This identity has a remarkable analogue for λ^+ instead of λ^-, which is directly related to the Robinson-Schensted algorithm.

1.2. Lemma. *For all $\lambda \in \mathcal{P}$*

$$(|\lambda| + 1)\#\mathcal{T}_\lambda = \sum_{\mu \in \lambda^+} \#\mathcal{T}_\mu.$$

Proof. By induction on $n = |\lambda|$. We have $\#\mathcal{T}_{(0)} = \#\mathcal{T}_{(1)} = 1$, so the lemma holds for $n = 0$; now assume that $n > 0$ and that the lemma holds for all $\mu \in \mathcal{P}_{n-1}$. We therefore have, using (3), the induction hypothesis, (1), (2) and once again (3):

$$\begin{aligned}(n+1)\#\mathcal{T}_\lambda = \#\mathcal{T}_\lambda + n\sum_{\mu\in\lambda^-}\#\mathcal{T}_\mu = \#\mathcal{T}_\lambda + \sum_{\mu\in\lambda^-}\sum_{\lambda'\in\mu^+}\#\mathcal{T}_{\lambda'} = (1+\#\lambda^-)\#\mathcal{T}_\lambda + \sum_{\mu\in\lambda^-}\sum_{\substack{\lambda'\in\mu^+\\ \lambda'\neq\lambda}}\#\mathcal{T}_{\lambda'}\\ = \#\lambda^+\#\mathcal{T}_\lambda + \sum_{\mu\in\lambda^+}\sum_{\substack{\lambda'\in\mu^-\\ \lambda'\neq\lambda}}\#\mathcal{T}_{\lambda'} = \sum_{\mu\in\lambda^+}\sum_{\lambda'\in\mu^-}\#\mathcal{T}_{\lambda'} = \sum_{\mu\in\lambda^+}\#\mathcal{T}_\mu. \quad \square\end{aligned}$$

We derive from this lemma a pair of interesting combinatorial identities.

1.3. Proposition. *The total number $t_n = \sum_{\lambda\in\mathcal{P}_n}\#\mathcal{T}_\lambda$ of normalised tableaux of n squares satisfies the recursion relation*

$$t_0 = 1 \qquad \text{and} \qquad t_{n+1} = t_n + nt_{n-1} \quad \text{for all } n \in \mathbf{N}$$

(to be interpreted in the obvious way for $n = 0$).

Proof. A straightforward computation:

$$\begin{aligned}\sum_{\lambda\in\mathcal{P}_{n+1}}\#\mathcal{T}_\lambda = \sum_{\lambda\in\mathcal{P}_{n+1}}\sum_{\mu\in\lambda^-}\#\mathcal{T}_\mu = \sum_{\mu\in\mathcal{P}_n}\#\mu^+\#\mathcal{T}_\mu = \sum_{\mu\in\mathcal{P}_n}(1+\#\mu^-)\#\mathcal{T}_\mu\\ = \sum_{\mu\in\mathcal{P}_n}\#\mathcal{T}_\mu + \sum_{\nu\in\mathcal{P}_{n-1}}\sum_{\mu\in\nu^+}\#\mathcal{T}_\mu = \sum_{\mu\in\mathcal{P}_n}\#\mathcal{T}_\mu + n\sum_{\nu\in\mathcal{P}_{n-1}}\#\mathcal{T}_\nu. \quad \square\end{aligned}$$

This proposition implies that the total number of normalised tableaux of size n is equal to the number of involutions in the symmetric group $\mathbf{S}_n$ (i.e., elements whose square is the identity, including the identity itself), since the latter number is easily seen to satisfy the same recursion. Indeed, an involution in $\mathbf{S}_{n+1}$ either fixes the last of the elements that $\mathbf{S}_{n+1}$ operates upon, in which case it is further determined by its action on the first n elements, or it exchanges the last element with one of the first n (say number i), in which case it is determined by i (with $1 \leq i \leq n$) and by its action on the remaining $n-1$ elements. From the proposition it also follows that the exponential generating function for the sequence t_n ($n \in \mathbf{N}$) is $e^{x+\frac{1}{2}x^2}$ (which means that t_n equals the n-th derivative evaluated at $x = 0$ of this function), see for instance [Stan2], Example 1.1.13. The following consequence of our lemma is even nicer than the first one.

1.4. Proposition.

$$\sum_{\lambda\in\mathcal{P}_n} (\#\mathcal{T}_\lambda)^2 = n! \qquad \textit{for all } n \in \mathbf{N}.$$

Proof. By induction:

$$\sum_{\lambda\in\mathcal{P}_n} (\#\mathcal{T}_\lambda)^2 = \sum_{\lambda\in\mathcal{P}_n}\sum_{\mu\in\lambda^-} \#\mathcal{T}_\lambda\#\mathcal{T}_\mu = \sum_{\mu\in\mathcal{P}_{n-1}}\sum_{\lambda\in\mu^+} \#\mathcal{T}_\lambda\#\mathcal{T}_\mu = n\sum_{\mu\in\mathcal{P}_{n-1}} (\#\mathcal{T}_\mu)^2 = n\cdot(n-1)! = n! \qquad \square$$

Remarks. The numbers $\#\mathcal{T}_\lambda$ for $\lambda \in \mathcal{P}_n$ appear in the representation theory of $\mathbf{S}_n$ as the dimensions of its irreducible representations. In that context proposition 1.4 states the well known relation between those dimensions and the order of the group. There is also an interpretation for 1.3, by a result of Frobenius and Schur (see [FrSch]) which in the case of groups such as $\mathbf{S}_n$, where all representations can be realised over the real numbers, can be formulated as follows: *for any $g \in G$ the number $\#\{\, x \in G \mid x^2 = g \,\}$ is the sum of all values at g of the irreducible characters.* (The cited result actually also tells how to take into account any possible non-real irreducible representations.) Our proposition 1.3 follows by taking for g the identity (the character values then become dimensions). We should also note that there is an explicit formula for the individual numbers $\#\mathcal{T}_\lambda$ (the Frame-Robinson-Thrall formula, see for instance [Kn2], theorem H), but no proof of that formula is known which is even nearly as simple as the proofs given above. (Nevertheless this formula may have been of crucial importance for the Robinson-Schensted algorithm, since it enabled Schensted to derive from his bijective correspondence the simple counting formula he was after; without it he might not have considered the bijection to be of much interest.)

These propositions show that there exist bijections from the set of normalised tableaux of size n respectively from the set of ordered pairs of such tableaux of equal shape to the set of involutions in $\mathbf{S}_n$ respectively to the whole $\mathbf{S}_n$. An obvious question is whether in each of these cases a particular such bijection can be defined in a natural way. The answer will be *yes*: in the second case there is the correspondence defined by the Robinson-Schensted algorithm, which is described in the next section, and from this a bijection for the first case can be obtained by embedding the set of all tableaux "diagonally" into the set of pairs of tableaux of equal shape (we shall see that involutions correspond to pairs of equal tableaux).

However, these facts do not yet fully express the relation between the results of this section and the Robinson-Schensted algorithm. It is possible, in a fairly straightforward manner, to *deduce* the Robinson-Schensted algorithm from the identities in this section and their proofs. Note that many of the quantities appearing in the identities are cardinalities of finite sets, and that only additions and multiplications occur—there are no cancellations. Therefore one may set out to systematically replace the identities by bijections between appropriate sets. In this process sums are replaced by disjoint unions, and products by Cartesian products; moreover identities established by an inductive proof lead to bijections defined by a recursive algorithm. We shall not describe this process in detail, since it is much more instructive for the reader to carry it out in his own way, using whatever formalism he prefers most to describe the results. We do note however that the recursive description of the Robinson-Schensted algorithm given in the next section is in a form such as could be obtained by the indicated construction.

A few remarks may help in performing this "derivation" of the Robinson-Schensted algorithm. The crucial step is to define a bijection corresponding to 1.2. The factor $|\lambda|+1$ occurring in it may be taken to correspond to the number of possibilities to extend a total ordering on a set of $|\lambda|$ elements to a set containing one more element. (Repeatedly performing such extensions, starting from the empty set, one finds the $n!$ possibilities to totally order a set of n elements, which correspond to the distinct permutations of n symbols; going from 1.2 to 1.4 proceeds along this line.) Therefore the left hand side of 1.2 can be interpreted as the number of ways to choose a similarity class of tableaux of shape λ, together with a place in the (total) ordering of the entries to fit in a new value (a candidate entry). The right hand side is the number of similarity classes of tableaux whose shape is in λ^+; the bijection to be constructed will associate such a class to each choice as indicated for the left hand side.

Since sums are to be translated into disjoint unions, the bijections will involve some form of "definition by cases"; in this context the splitting off of certain terms in the proofs will lead to exceptional cases

in the corresponding definitions of bijections. A choice has to be made in interpreting (1), since there are two natural injective maps from the corners of a partition to its cocorners, mapping each corner to the cocorner in the next row respectively to the cocorner in the next column. The former choice (for which the cocorner in the first row is the unmatched "exceptional cocorner") should be taken in order to find the usual Robinson-Schensted correspondence; the other choice leads to a transposed variant. Note that a similar construction of a bijective correspondence can be done for any differential poset (where saturated chains in the poset take the place of Young tableaux), provided a particular bijectivisation of (1) is chosen; this was already noted in [Fom2].

§2. The Robinson-Schensted algorithm, definitions.

We start this section with the "traditional" definition of the Robinson-Schensted correspondence. This term is not meant to imply that this definition is used throughout the literature: several definitions can be found, often without much direct resemblance (and the recursive definition below which will be our crucial ingredient appears to be a novel contribution to the list). The oldest definition, formulated in 1938 by Robinson, will be briefly discussed in §7, but here we shall reproduce the definition independently formulated by Schensted more than two decades later. This definition may be considered to be the basic one, as it is doubtlessly the easiest one to understand and to perform by hand.

The algorithm is based on a procedure to insert a new number into a Young tableau, thereby displacing certain entries and eventually leading to a tableau with one square more than the original one. More precisely, there is a pair of mutually inverse procedures that convert into each other the following sets of data: on one hand a tableau T and a number m not occurring as entry of T, and on the other hand a non-empty tableau P and a specified corner s of $\mathrm{sh}\, P$. We shall call the computation of P and s given T and m the *insertion* procedure I, and write $(P, s) = I(T, m)$. The inverse operation will be called the *extraction* procedure E, written $(T, m) = E(P, s)$.

The procedures are such that the following relations always hold: the set of entries of P is that of T together with the number m, and the $Y(\mathrm{sh}\, P) = Y(\mathrm{sh}\, T) \cup \{s\}$ (so that s is a corner of $\mathrm{sh}\, P$ and a cocorner of $\mathrm{sh}\, T$). Moreover, the procedures depend only on the relative order of the entries, i.e., replacing P by a tableau similar to it will correspond to the same renumbering applied to m and the entries of T. These procedures therefore define a bijection corresponding to lemma 1.2, as indicated in the previous section. Our description of these procedures will be somewhat informal; a more formal and elaborate description can be found in the excellent exposition [Kn2].

> *Insertion procedure.* Given a tableau T and a number m, the pair $(P, s) = I(T, m)$ is determined as follows. The first step is to insert m into the first row of T, where it either replaces the smallest entry larger than m, or, if no such entry exists, it is simply appended at the end of the row. Then the following (similar) step is repeated, as long as a number, say k, has been replaced at the most recent step. The number k is inserted into the row succeeding its original row, either replacing the smallest entry larger than itself, or, if no such entry exists, by being appended at the end of that row. The tableau obtained after the last step is P, while the square occupied during that step is s.

Since we are moving a row down at each step, it is obvious that the procedure must terminate, possibly creating a new row of length 1 at the last step. Note that s is the only new square to be occupied by a number. The number which occupies s at the final step exceeds all other numbers present in the same row, and therefore it cannot have been displaced from a square directly above any of them (since the original T is assumed to be a tableau). This excludes the possibility that this row has become longer than its predecessor, so although we have yet to show that P is a tableau, the set of squares it occupies certainly is a Young diagram, and the corresponding partition may be denoted by $\mathrm{sh}\, P$.

> *Extraction procedure.* Given a tableau P and a corner s of $\mathrm{sh}\, P$, the pair $(T, m) = E(P, s)$ is determined as follows. The first step is to remove the square s from P, together with the number it contains. Then repeat the following step until a number has been replaced or removed from the first row. The number removed or replaced in the previous step is moved to the row

preceding its original row, where it replaces the largest entry smaller than itself (such an entry exists, since the number originally directly above it is certainly smaller than it). The tableau obtained after the last step is T, while the entry removed or replaced from the first row is m.

Like for the insertion procedure, we have yet to show that the result T is a tableau, but since the square s is assumed to be a corner of $\mathrm{sh}\,P$, the squares of T occupy the Young diagram $Y(\mathrm{sh}\,P)\setminus\{s\}$, and $\mathrm{sh}\,T$ is a well defined partition. It can be verified immediately that the two procedures are inverses of each other in the sense that the effect of one can be undone by applying the other (even if the intermediate filled Young diagram were no tableau). We illustrate the rules by an example that involves four steps. We show the intermediate stages of the procedure I; for an example of the procedure E, read from right to left.

$$m=7,T=\begin{array}{|c|c|c|c|}\hline 2&5&6&8\\\hline 3&10&12\\\cline{1-3}9&13&15\\\cline{1-3}\end{array}\quad\begin{array}{|c|c|c|c|}\hline 2&5&6&7\\\hline 3&10&12\\\cline{1-3}9&13&15\\\cline{1-3}\end{array}\quad\begin{array}{|c|c|c|c|}\hline 2&5&6&7\\\hline 3&8&12\\\cline{1-3}9&13&15\\\cline{1-3}\end{array}\quad\begin{array}{|c|c|c|c|}\hline 2&5&6&7\\\hline 3&8&12\\\cline{1-3}9&10&15\\\cline{1-3}\end{array}\quad\begin{array}{|c|c|c|c|}\hline 2&5&6&7\\\hline 3&8&12\\\cline{1-3}9&10&15\\\cline{1-3}13\\\cline{1-1}\end{array}=P,s=(4,1)$$

At each stage except the rightmost there is one number missing: this is the entry that has been superseded but not yet inserted into another row.

We now check that the two procedures preserve the tableau property, and at the same time we establish a pair of recurrence relations that their results satisfy. If in the insertion procedure the entry m to be inserted is higher than any of the entries already present in T, then the insertion consists of just adding m at the end of the first row of T, and the result P is clearly a tableau, with moreover $P^-=T$. Now assume that the highest entry h of T exceeds m; we compare the computation of $I(T,m)$ with that of $I(T^-,m)$. The presence of h in T can only influence the insertion steps, if at some point it is the *only* entry in its row that exceeds the number being inserted into that row. If this occurs then the next step will be the final one of the computation of $I(T,m)$ (since h, being maximal, can replace no other element) while in the computation of $I(T^-,m)$, where h is absent, the insertion procedure already terminates without this last step. Therefore inserting m into T^- will yield P^- (to be read as "P without its highest entry h", since we have yet to show that P is a tableau; this is next on the programme). By earlier remarks it follows that in any case the numbers of P^- occupy a Young diagram, so that the entry h appears in a corner of $\mathrm{sh}\,P$. But then, applying 1.1, we establish by induction on the size of T that P is indeed a Young tableau.

The relation between inserting into T and T^- can be expressed more formally as

$$\text{if } I(T,m)=(P,s) \text{ then } I(T^-,m)=(P^-,s') \text{ for some square } s', \tag{4}$$

still under the assumption that m does not exceed all entries of T. Moreover, we can be more explicit about the square s' in (4). In the first place we must have

$$\{s,\lceil T\rceil\}=\{s',\lceil P\rceil\} \tag{5}$$

because both sets are equal to $Y(\mathrm{sh}\,P)\setminus Y(\mathrm{sh}\,T^-)$. When $s'\neq\lceil T\rceil$ this already implies that $s=s'$ and $\lceil P\rceil=\lceil T\rceil$; this is the case that h is not displaced during the insertion. In the case that h is displaced, the previous condition has to be supplemented by the additional relation

$$\text{if } s'=\lceil T\rceil \text{ then } s \text{ is the cocorner of } \mathrm{sh}\,T \text{ one row below the corner } s'. \tag{6}$$

Relations (5) and (6) determine s and $\lceil P\rceil$ when s' and $\lceil T\rceil$ are known, and this is the way they shall be used. With their aid, relation (4) completely determines P and s for given T and m, as soon as P^- and s' are determined.

For the extraction procedure may reason similarly. Here the exceptional case is when s coincides with $\lceil P\rceil$ and lies in the first row; we then have $T=P^-$ and m is the entry in P of the square s. In the

other cases we find that T is a tableau by a similar induction as given for the insertion procedure, and obtain the recurrence relation

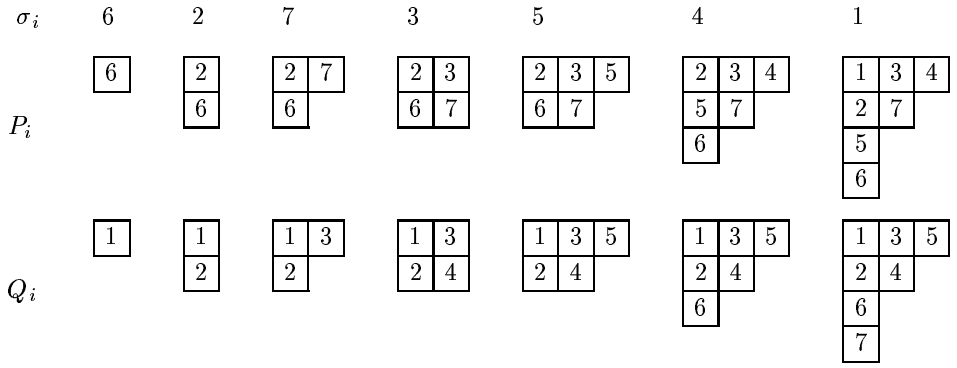
$$\text{if } E(P,s) = (T,m) \text{ then } E(P^-,s') = (T^-,m) \text{ for some square } s' \tag{7}$$

(this is equivalent to (4), but is used in the opposite direction). Again s and s' are related; the relations (5) and (6) are valid, but since they are now used to determine s' and $\lceil T \rceil$ from s and $\lceil P \rceil$, it is more natural to replace (6) by the equivalent

$$\text{if } s = \lceil P \rceil \text{ then } s' \text{ is the corner of } \operatorname{sh} P^- \text{ one row above the cocorner } s. \tag{8}$$

The full Robinson-Schensted algorithm will now be defined. It establishes a bijection corresponding to corollary 1.4, namely between the symmetric group $\mathbf{S}_n$ and the set $\bigcup_{\lambda\in\mathcal{P}_n} \mathcal{T}_\lambda \times \mathcal{T}_\lambda$ of pairs (P,Q) of normalised tableaux with $\operatorname{sh} P = \operatorname{sh} Q \in \mathcal{P}_n$. We represent a permutation $\sigma \in \mathbf{S}_n$, as a sequence $(\sigma_1,\ldots,\sigma_n)$ of distinct numbers (so σ maps $i \mapsto \sigma_i$); given such a sequence we build up the pair (P,Q) in n stages as follows. Put $P_0 = Q_0$ equal to the empty tableau, and let (P_{i-1},Q_{i-1}) be the pair at the beginning of the i-th stage. In that stage we compute $(P_i,s) = I(P_{i-1},\sigma_i)$, and obtain Q_i from Q_{i-1} by adding the number i in the position of square s. Finally we put $(P,Q) = (P_n,Q_n)$. Since the number added to Q_{i-1} is the highest until then, it is clear that each Q_i is a (normalised) tableau. The set of entries of P_i is $\{\sigma_1,\ldots,\sigma_i\}$, so P_n is also a normalised tableau. We have $\operatorname{sh} P_i = \operatorname{sh} Q_i$, so in particular $\operatorname{sh} P = \operatorname{sh} Q$ as claimed.

Like the auxiliary procedures, the full algorithm too can be directly reversed: start with putting $(P_n,Q_n) = (P,Q)$ and then repeatedly transform (P_i,Q_i) into (P_{i-1},Q_{i-1}) (for $i = n,\ldots,2,1$), meanwhile determining the numbers $\sigma_n,\ldots,\sigma_1$, namely by putting $Q_{i-1} = Q_i^-$ and $(P_{i-1},\sigma_i) = E(P_i,\lceil Q_i \rceil)$. We illustrate the algorithm, and its inverse, by an example: for the construction of (P,Q) read from left to right, for the inverse process from right to left .



If by these bijections the permutation $\sigma \in \mathbf{S}_n$ corresponds to the pair of normalised tableaux (P,Q), we shall write $\sigma = R(P,Q)$ and $(P,Q) = R^{-1}(\sigma)$. Note that R is the bijection from pairs of tableaux to permutations, which we called the 'inverse process' above; this choice is motivated by that greater notational convenience of functions with multiple arguments compared to those with multiple results. The procedures I and E have obvious transposed counterparts I^t and E^t, whose definition can be obtained by replacing all occurrences of the word 'row' by 'column'. These transposed procedures can be used to define another bijection $R^t\colon \bigcup_{\lambda\in\mathcal{P}} \mathcal{T}_\lambda \times \mathcal{T}_\lambda \xrightarrow{\sim} \mathbf{S}_n$ in exactly the same way that R is defined; it obviously satisfies $R^t(P,Q) = R(P^t,Q^t)$. The inverse of R^t will be written R^{-t}.

The Robinson-Schensted correspondence has some remarkable properties, and our main goal is to prove these in an elementary way. In order to do this, we have to recast the definition of the algorithms into a recursive form, which is based on the observation that the recurrence relations (4)–(8) are sufficient to completely define I and E. Since the recursive definitions play such a crucial rôle, we shall go to some lengths in defining an appropriate formalism for them. The formalism, which strongly resembles that of "functional programming", will be more precise than what is used for (4)–(8), in particular it better

exhibits the dependency between the different variables occurring, clearly indicating which values are computed from which other values.

For the contents of a square s of a tableau T we write T_s. We shall also want to specify a non-empty tableau T by giving T^-, the square $s = \lceil T \rceil$, and its contents T_s; for this we use the notation

$$T = T^- : (s, T_s).$$

The operator ':' may be read as "attach a new square at the indicated position with indicated entry (which exceeds all existing ones)". The operator will be left-associative, so that if for instance T has at least 3 squares, of which the squares $s_1 = \lceil T \rceil$, $s_2 = \lceil T^- \rceil$ and $s_3 = \lceil T^{--} \rceil$ have the three highest entries $c_i = T_{s_i}$, then we may write $T = T^{---} : (s_3, c_3) : (s_2, c_2) : (s_1, c_1)$. We also write '$\odot$' for the empty tableau (which we believe is clearer then '$\emptyset$'). In this way each tableau may be unraveled into $\odot$ followed by a colon-separated list of filled squares, in order of increasing entries; e.g., the intermediate tableau P_4 in the above display of the Robinson-Schensted algorithm becomes $\odot : ((1,1),2) : ((1,2),3) : ((2,1),6) : ((2,2),7)$.

In fact it is useful to have a similar notation for other kinds of sequences, such as permutations, which are (represented as) sequences of numbers. We shall write '()' for the empty sequence, and '$\sigma \,\&\, x$' for a non-empty sequence whose last term is x and where σ is the initial subsequence of the remaining terms, so $(x_1, \ldots, x_n) = (x_1, \ldots, x_{n-1}) \,\&\, x_n = () \,\&\, x_1 \,\&\, \cdots \,\&\, x_n$.

In order to make our formalism as compact and self-contained as possible, we extend our mathematical notation with two new forms, which will be recognisable by the appearance of bold-face keywords. The first is the conditional expression, which takes the form

$$\textbf{if } \langle \text{condition} \rangle \textbf{ then } \langle \text{expression}_1 \rangle \textbf{ else } \langle \text{expression}_2 \rangle \textbf{ fi},$$

and specifies definition by cases, replacing the more clumsy form

$$\begin{cases} \langle \text{expression}_1 \rangle & \text{if } \langle \text{condition} \rangle \\ \langle \text{expression}_2 \rangle & \text{otherwise.} \end{cases}$$

The second form is the local definition, having as basic form

$$\textbf{let } \langle \text{variable} \rangle = \langle \text{expression}_1 \rangle \textbf{ in } \langle \text{expression}_2 \rangle.$$

It is defined to be equivalent to the result of substituting the value of $\langle \text{expression}_1 \rangle$ for all occurrences of $\langle \text{variable} \rangle$ in $\langle \text{expression}_2 \rangle$. The form can be used as an abbreviation when a complicated subexpression occurs several times. When used in some larger context it may not be immediately clear how far $\langle \text{expression}_2 \rangle$ extends (i.e., what the scope of $\langle \text{variable} \rangle$ is); it is taken to be the largest syntactically correct expression following '**in**'. Sometimes a series of such definitions is needed; in such cases we write

$$\textbf{let } x = E_1;\ y = E_2 \textbf{ in } E_3 \qquad \text{which stands for} \qquad \textbf{let } x = E_1 \textbf{ in let } y = E_2 \textbf{ in } E_3.$$

Another useful extension of this notation is to allow—in case $\langle \text{expression}_1 \rangle$ denotes a value in a Cartesian product—that a tuple of variables appears after **let**, instead of $\langle \text{variable} \rangle$. In this case each variable in the tuple denotes the corresponding component of the value of $\langle \text{expression}_1 \rangle$. The form then becomes something like

$$\textbf{let } (a, b, c) = E \textbf{ in } F,$$

which is equivalent to the result of substituting the first, second and third projections of E for a, b and c respectively into F. Thus we may avoid writing any projection functions explicitly. For sequences of unknown length we similarly allow patterns of the form '$(\sigma \,\&\, x)$' to be used following **let**; in this case $\langle \text{expression}_1 \rangle$ must be known to be a non-empty sequence, of which x denotes the final term and σ the (possibly empty) remaining initial sequence. As an example,

$$\textbf{let } (\sigma \,\&\, x) = E \textbf{ in } \sigma \,\&\, x \,\&\, x$$

denotes $(4,7,1,1)$ when $E=(4,7,1)$, while if $E=(9)$ it denotes $(9,9)$.

Both the '**if**' and the '**let**' expressions may be read in an algorithmic way. We may interpret a conditional expression as "first evaluate the condition, then choose the appropriate subexpression". In particular, the subexpressions after '**then**' and '**else**' need only to have a well defined meaning in case they are actually chosen. Similarly '**let** $v=E$ **in** F' may be interpreted as: "first evaluate E, then ascribe its value to v and evaluate F". This interpretation depends on the fact that we restrict our use of this form to situations where E has a well defined value whenever the form needs to be evaluated, independently of where it will end up after substitution into F (i.e., we need never resort to substituting unevaluated expressions).

Now we return to the Robinson-Schensted algorithm. Expressing the full algorithm R in terms of the extraction procedure E provides a good illustration of a recursive definition using our formalism.

$$R(P,Q) = \textbf{if } Q=\odot \textbf{ then } () \textbf{ else let } (T,m)=E(P,\lceil Q\rceil) \textbf{ in } R(T,Q^-)\,\&\,m \textbf{ fi} \tag{9}$$

This can be read as follows. To compute $R(P,Q)$, first see if Q is empty; if so the result is the empty sequence. Otherwise locate the square $\lceil Q\rceil$ and apply the extraction procedure E to P and this square, and call the resulting tableau T and the extracted entry m. The entry m will be the final term of the sequence $R(P,Q)$, and the sequence of terms before it can be obtained by computing $R(T,Q^-)$ using this very same prescription.

In this way a simple iteration can be expressed by recursion, although of course recursion can also be used for more complicated patterns of computation. However, for our purposes "linear" recursion will suffice, i.e., the evaluation of the right hand side of any definition will directly lead to at most one other invocation of the same definition. When writing recursive definitions such as (9), we must of course make sure that the recursive application on the right hand side is strictly simpler (in the sense of some well-founded ordering, in this case by the number of squares of the tableaux involved) than the original case to be computed. This condition will be met in an obvious way for all the recursive definitions given below, and we shall not mention it explicitly any more. If one fully understands definition (9), then reading the more complicated recursive definitions below should not present any fundamental difficulties (**else** try again **fi**).

Expressing R^{-1} in terms of I is slightly more difficult than (9) since a permutation does not explicitly record its own length, while this length is needed to determine the entries of Q. We shall compute length simultaneously as the third component of the result returned by the auxiliary function R':

$$\begin{aligned} R^{-1}(\sigma) &= \textbf{let } (P,Q,n)=R'(\sigma) \textbf{ in } (P,Q) \qquad \text{[the computed length } n \text{ is not needed in the end]},\\ R'(\sigma) &= \textbf{if } \sigma=() \textbf{ then } (\odot,\odot,0)\\ &\qquad \textbf{else let } (\sigma'\,\&\,m)=\sigma;\ (T,Q',i)=R'(\sigma');\ (P,s)=I(T,m)\\ &\qquad\quad \textbf{in } \big(P,Q':(s,i+1),i+1\big)\\ &\qquad \textbf{fi} \end{aligned} \tag{10}$$

Note that although σ_1 is the first number to be inserted during the computation of $R'(\sigma)$, we first isolate $m=\sigma_n$ from σ (this is easier in our formalism), and simply invoke R' recursively *before* applying I. When evaluating $R'(\sigma)$ one must of course carefully distinguish the different values ascribed to σ, σ', m, T, Q', i, P and s during the different nested invocations of R' (this is what makes recursion somewhat tedious to perform by hand).

Now we turn to the recursive definitions of the procedures I and E, which illustrate the formalism in full swing (and which motivated the introduction the formalism to begin with). We start with the extraction procedure; note that in $E(P,s)$ we must have $P\neq\odot$ since P should contain the square s.

$$\begin{aligned} E(P,s) = \textbf{let } &(r,c)=s;\ \lambda=\mathrm{sh}\,P^-;\ h=P_{\lceil P\rceil} \textbf{ in}\\ &\textbf{if } s\neq\lceil P\rceil \textbf{ then let } (T',m)=E(P^-,s) \textbf{ in } \big(T':(\lceil P\rceil,h),m\big)\\ &\textbf{else if } r=1 \textbf{ then } (P^-,h)\\ &\qquad \textbf{else let } s'=(r-1,\lambda_{r-1});\ (T',m)=E(P^-,s') \textbf{ in } \big(T':(s',h),m\big)\\ &\qquad \textbf{fi}\\ &\textbf{fi} \end{aligned} \tag{11}$$

Note that the square s'—when defined—is the corner of $\mathrm{sh}\,P^-$ one row above its cocorner s. Then, even without reference to the previous definition of E, it follows by induction that $E(P,s)$ is always well defined (provided that s is a corner of $\mathrm{sh}\,P$), and consists of a tableau T and a number m with the required relationship in shape and content to P and s.

It is easy to see that (11) reflects (5)–(8), with T' standing for T^- in those relations. For if $s \neq \lceil P \rceil$ then $s' = s$ and $\lceil T \rceil = \lceil P \rceil$ by (5), while if $s = \lceil P \rceil$ and this square does not lie in the first row, then $s' = \lceil T \rceil$ is computed as the the corner of $\mathrm{sh}\,P^- = \mathrm{sh}\,T$ in the row above s, as required by (8). The case that $s = \lceil P \rceil$ lies in the first row is also treated properly. We give a similar recursive definition for the insertion procedure.

$$
\begin{aligned}
I(T,m) = {} & \textbf{let } \lambda = \mathrm{sh}\,T \textbf{ in} \\
& \quad \textbf{if } T = \odot \vee m > T_{\lceil T \rceil} \textbf{ then let } s = (1, \lambda_1 + 1) \textbf{ in } \big(T : (s,m), s\big) \\
& \quad \textbf{else let } (P', s') = I(T^-, m);\ h = T_{\lceil T \rceil} \textbf{ in} \\
& \qquad \textbf{if } s' \neq \lceil T \rceil \textbf{ then } \big(P' : (\lceil T \rceil, h), s'\big) \\
& \qquad \textbf{else let } (r,c) = s';\ s = (r+1, \lambda_{r+1} + 1) \textbf{ in } \big(P' : (s,h), s\big) \\
& \qquad \textbf{fi} \\
& \quad \textbf{fi}
\end{aligned}
\tag{12}
$$

Similar remarks to those made for (11) apply here. The definitions of the transposed versions R^t, R^{-t}, E^t and I^t of these functions are almost identical to those above, and we shall not write them out. For R^t and R^{-t} it suffices to replace the calls to E and I by E^t and I^t respectively, for E^t we have to replace $r = 1$ in (11) by $c = 1$ and $(r-1, \lambda_{r-1})$ by $(\lambda^t_{c-1}, c-1)$, and for I^t two similar modifications to (12) must be made; we shall refer to these transposed definitions as (9^t)–(12^t).

The recursive definitions of E and I are less "efficient" than their original descriptions, in that in general more steps (recursive invocations) are needed to complete their computation; indeed, in those cases where the condition $s \neq \lceil P \rceil$ holds (respectively $s' \neq \lceil T \rceil$), no entry is moved to another place, and there is no corresponding step in the original procedures. However, this lack of efficiency is an advantage mathematically, since the individual steps are simpler (no searching for particular entries in a row: the position of the new square depends only on the shape of the tableau), and so will be the corresponding induction steps in proofs.

The relation between the proof of 1.2 and either (11) or (12) should be apparent even if one has not attempted constructing a bijection corresponding to that lemma. To illustrate the point, consider (12) and put in addition to $\lambda = \mathrm{sh}\,T$ also $\mu = \mathrm{sh}\,T^-$ and $\lambda' = \mathrm{sh}\,P'$, and let ν be the shape of P (i.e., of the tableau-part of the result). Then the square $\lceil T \rceil$ is the difference between (the diagrams of) λ and μ, while s' is the difference between λ' and μ. Hence the condition $s' \neq \lceil T \rceil$ is equivalent to $\lambda' \neq \lambda$, and in that case $\nu = \lambda \cup \lambda'$ (while $\mu = \lambda \cap \lambda'$). When $s' = \lceil T \rceil$ holds, then ν is the image of μ under the chosen injection $\lambda^- \to \lambda^+$ (see the previous section), while in the case taken aside by the outer conditional expression in (12), ν is the "exceptional" element of λ^+ that does not appear in the image of that map.

Although the algorithm R is intended to operate on pairs of normalised tableaux, its definition remains valid if the tableaux are not normalised. There is an asymmetry between the rôles of P and Q in $R(P,Q)$ in this respect, however, in that changing P to a similar tableau will affect the resulting number sequence by the same renumbering, while changing Q to a similar tableau has no effect at all on the result; indeed R^{-1} applied to the result will reproduce P exactly, but Q will be replaced by the normalised tableau similar to it. A way to restore the symmetry is to let the value yielded by R not just be a sequence of numbers, but rather a bijection $X \to Y$ between the sets $X, Y \subset \mathbf{Z}$ of the entries of Q and P respectively. Such a variant of the correspondence could be given by a modification of the definition of R, such that at each step a pair $(x,y) \in X \times Y$ is computed, where x is the entry removed from Q and y the entry extracted from P. However, one can just as well use the correspondence for normalised tableaux, by first normalising the tableaux, then applying R and finally replacing the resulting permutation by the corresponding map $X \to Y$ obtainable by monotonically pairing both X and Y with $\{1, \ldots, n\}$. Therefore such a "generalisation" has little mathematical content, and we shall not pursue it. When we occasionally write $R(P,Q)$ for non-normalised Q, the resulting sequence is often naturally expressed as another sequence from which certain terms (at positions corresponding to entries

missing from Q) are omitted; for this we shall use the common notation $(a_1, \ldots, \widehat{a_i}, \ldots, a_n)$ where the hat indicates an omitted number.

We close this section by describing a convenient method of performing the Robinson-Schensted algorithm on a computer. It is an iterative procedure, but it is related to the idea behind the recursive definitions, namely to consider the squares of a tableau not by their position in the tableau, but rather ordered by the size of their entries. For this purpose tableaux are represented linear instead of 2-dimensional arrays: the i-th entry of the array describes the position of the square whose entry is i. A convenient circumstance is that it is sufficient to record only the row numbers of the squares, since the tableau property guarantees that the c-th occurrence from left to right of some number r corresponds to a square in column c of row r. In fact the tableau condition implies an additional constraint, namely, if $r > 1$ then the c-th occurrence of $r-1$ must precede the c-th occurrence of r (for all applicable r, c). The number 0 can be used to denote an absent square. We call such an array of numbers a *row-encoded tableau*; in the literature such arrays are known under the hardly informative names 'lattice permutations' and *'mots de Yamanouchi'*.

The computation of $(P_i, s) = I(P_{i-1}, m)$ is performed by modifying a row-encoded tableau P (with entries written $P[j]$ for $1 \leq j \leq n$) which initially represents P_{i-1}, into one representing P_i. To this end we make an increasingly large subtableau correspond to P_i rather than to P_{i-1}; an index j indicates up to which number the entries have been incorporated, and a variable r records the row number of the square s by which the shape of the subtableau has been extended due to the insertion. Then according to (4) all that is needed when j is increased by 1 is to possibly update the values of $P[j]$ and r. By (5) nothing needs to be done if $P[j] \neq r$, and if $P[j] = r$ then by (6) both values should be increased by 1. Therefore, it is not difficult to see that the following Pascal fragment computes row-encoded tableaux P, Q corresponding to the permutation *sigma* represented by an array of length n:

```
for i := 1 to n do P[i] := 0;
for i := 1 to n do
begin r := 0
;   for j := sigma[i] to n do
    if P[j] = r then begin P[j] := r + 1;  r := r + 1 end
;   Q[i] := r
end
```

Note that the condition $P[j] = r$ should always be satisfied for $j = sigma[i]$, since we assume that each value $sigma[i]$ occurs only once in the sequence. The procedure can be reversed in an obvious way (but the number of iterations of the inner loop can not be predicted in this case: it will be of the form '**repeat** ... **until** $r = 0$', which should not fail to terminate, provided that we start with proper values for P and Q). In the limit of huge n these procedures will be less efficient on the average than a straightforward 2-dimensional approach, but their extreme simplicity probably makes them more favourable in most practical situations.

§3. The Robinson-Schensted algorithm, first properties.

We now turn to the first of the 'remarkable properties' announced above. The set $\mathbf{S}_n$ of permutations forms a group, so its elements can be inverted: in terms of sequences of numbers, the inverse $\tau = \sigma^{-1}$ of $\sigma = (\sigma_1, \ldots, \sigma_n)$ is the sequence $(\tau_1, \ldots, \tau_n)$ whose term τ_i is the unique index j such that $\sigma_j = i$.

3.1. Theorem.

$$R(Q, P) = R(P, Q)^{-1}$$

for all $(P, Q) \in \bigcup_\lambda \mathcal{T}_\lambda \times \mathcal{T}_\lambda$.

This theorem was already stated (without proof) by Robinson, and it was first proved (for Schensted's algorithm which was at that time not known to be related to Robinson's) by Schützenberger in [Schü1], §4 (at least a proof can be reconstructed from it, after correcting a number of misprinted formulae). Other authors have subsequently given different proofs, see for instance [Kn1], Theorem 3. The proof we give here is typical for the way our recursive formalism is used: the main tool is instantiating the relevant definitions, and carefully expanding and contracting, while distinguishing the possible cases, until the desired identity is proved. As this "substitutional algebra" is not so easily performed in ones head, the reader is advised to work out the indicated steps with pencil and paper if necessary.

Proof. We proceed by induction on $n = |\operatorname{sh} P|$, the case $n = 0$ being trivial. We first single out a particularly easy case, namely when $\lceil P \rceil = \lceil Q \rceil$ and this square lies in the first row. Then from (11) it follows immediately that $E(P, \lceil Q \rceil) = (P^-, n)$ and similarly $E(Q, \lceil P \rceil) = (Q^-, n)$ (both times the clause after $r = 1$ applies). Therefore by (9) we have $R(P, Q) = R(P^-, Q^-) \,\&\, n$ and $R(Q, P) = R(Q^-, P^-) \,\&\, n$, and the theorem follows by induction. Now assume we are in the remaining case. Let $(P', i) = E(P, \lceil Q \rceil)$ and $(Q', j) = E(Q, \lceil P \rceil)$; we claim the following crucial statements:

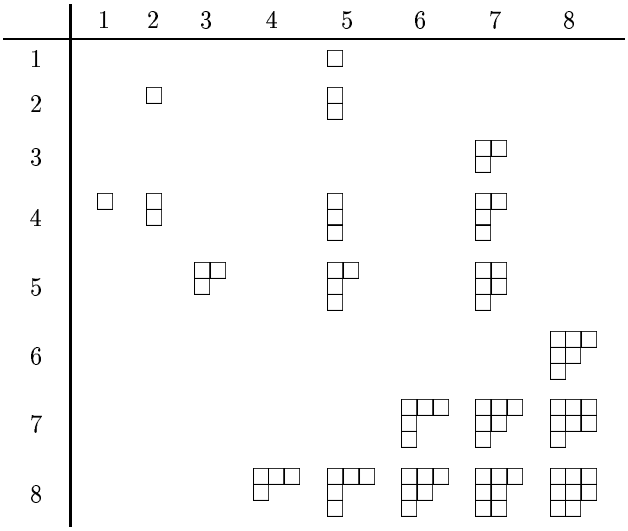
$$E(P^-, \lceil Q' \rceil) = (P'^-, i) \qquad \text{and similarly} \qquad E(Q^-, \lceil P' \rceil) = (Q'^-, j). \tag{$*$}$$

To prove it we distinguish two cases. First assume that $\lceil P \rceil \neq \lceil Q \rceil$, and consider (11) applied in the determination of $E(P, \lceil Q \rceil)$. The clause after $s \neq \lceil P \rceil$ applies, and we know that the result is (P', i); therefore we may conclude that $E(P^-, \lceil Q \rceil) = (P'^-, i)$ and $\lceil P' \rceil = \lceil P \rceil$. For the determination of $E(Q, \lceil P \rceil)$ we obtain similarly $E(Q^-, \lceil P \rceil) = (Q'^-, j)$ and $\lceil Q' \rceil = \lceil Q \rceil$; combining these results we get $(*)$. Now assume that $\lceil P \rceil = \lceil Q \rceil$, and consider the application of (11) in the computation of $E(P, \lceil Q \rceil)$ and of $E(Q, \lceil P \rceil)$. Either case leads to setting $(r, c) = \lceil P \rceil = \lceil Q \rceil$ and $\lambda = \operatorname{sh} P^- = \operatorname{sh} Q^-$, and selecting the **else**-branch of the outer conditional expression. Since the case that $r = 1$ was already treated separately, we also select the **else**-branch of the inner conditional, and set $s' = (r - 1, \lambda_{r-1})$; all this holds for both computations. Now we reason as before from the known outcome of the computation, to obtain $E(P^-, s') = (P'^-, i)$ and $\lceil P' \rceil = s'$, and similarly $E(Q^-, s') = (Q'^-, j)$ and $\lceil Q' \rceil = s'$; again these combine to prove $(*)$, which is thereby proved in both cases.

Now to prove the theorem it remains to judiciously apply $(*)$ and the induction hypothesis. Let $R(P, Q) = (\sigma_1, \ldots, \sigma_n)$, and let $\tau = \sigma^{-1}$ be its inverse; because of the case which has been excluded we have $\sigma_n \neq n \neq \tau_n$. By (9) we have $\sigma_n = i$ and $(\sigma_1, \ldots, \sigma_{n-1}) = R(P', Q^-)$. Applying the induction hypothesis we obtain $R(Q^-, P') = (\tau_1, \ldots, \widehat{\tau_i}, \ldots, \tau_n)$ since we have $\tau_i = n$. Comparing this to the second part of $(*)$, and using (9), we apparently must have $j = \tau_n$ and $R(Q'^-, P'^-) = (\tau_1, \ldots, \widehat{\tau_i}, \ldots, \tau_{n-1})$. Next, consider $R(Q, P)$; by (9) we see that its final term is $j = \tau_n$ as desired, and it remains to show that $R(Q', P^-)$ is equal to $(\tau_1, \ldots, \tau_{n-1})$. Now the first part of $(*)$ together with $i = \sigma_n$ gives us $R(P^-, Q') = R(P'^-, Q'^-) \,\&\, \sigma_n$, which by the known expression for $R(Q'^-, P'^-)$ and the induction hypothesis is equal to $(\sigma_1, \ldots, \widehat{\sigma_j}, \ldots, \sigma_n)$. A third and final application of the induction hypothesis gives the desired equation $R(Q', P^-) = (\tau_1, \ldots, \tau_{n-1})$, which completes the proof. □

This proof clearly exhibits the usefulness of the formalism for recursive definitions. Nevertheless it is a bit technical, and one would like to have some more transparent view of what is going on. There is a nice pictorial rendering of the proof which a once displays its global structure. For each permutation σ and corresponding pair (P, Q) of tableaux we define a certain structure that gives information about the intermediate values in the computation of $R(P, Q)$ or $R^{-1}(\sigma)$. This structure consists in ascribing to

certain points in $\mathbf{N} \times \mathbf{N}$ a partition or, equivalently, a Young diagram (we speak of points rather than squares here since the two are not directly related, and moreover we find the idea of filling a square with a Young diagram somewhat embarrassing). As an example, for $\sigma = (5, 2, 7, 1, 3, 8, 6, 4)$ it can be depicted as follows:



We now describe in general how this picture is related to σ and (P, Q).

The set of points that has a partition ascribed to it is $\Sigma_\sigma = \{\,(i,j) \mid \sigma_i \leq j \wedge {\sigma^{-1}}_j \leq i\,\} \cup \{(0,0)\}$ (the point $(0,0)$ is just a sentinel, and the partition (0) is always ascribed to it; as for the other points, think of poles being erected at all points (i, σ_i), and two lights shining from the top and left, then these are the points that either contain a pole or are completely in the shadow). Note that, for the natural partial ordering '$\leq$' on $\mathbf{N} \times \mathbf{N}$ the least upper bound of any subset of Σ_σ lies in Σ_σ; in particular, for any point $(i,j) \geq (0,0)$ the set of points $\leq (i,j)$ of Σ_σ has a unique maximal element.

For $i > 0$, row i of Σ_σ corresponds to the stages during the insertion (or extraction) of the number σ_i into (out of) the tableau. The partition ascribed to the last point of this row is the shape of the tableau P_i (obtained after completing the insertion of σ_i). The partition ascribed to an arbitrary point $(i,j) \in \Sigma_\sigma$ in this row, is the shape of the subtableau of P_i of squares with entries not exceeding j; the applicable j are precisely the entries $\geq \sigma_i$ occurring in P_i. So the partitions in this row display from right to left the chain $\mathrm{ch}\, P_i$, up to the partition ascribed to (i, σ_i). The remaining part of this chain is unchanged from $\mathrm{ch}\, P_{i-1}$, and the next partition can be found by moving from (i, σ_i) on to the maximal point in Σ_σ less than it (i.e., to the left and above), which we call the *base* point of (i, σ_i); then we proceed further to the left, possibly going from a point to its base point as necessary, until eventually the sentinel point $(0,0)$ is reached.

We can define for any point in $\Sigma_\sigma \setminus \{(0,0)\}$ its *left neighbour* to be the preceding point in the same row of Σ_σ, unless it is of the form (i, σ_i), in which case we take its base point. We similarly define its *upper neighbour* to be the preceding point in the same column of Σ_σ, again taking the base point for points (i, σ_i). One can verify that starting from any point not of the form (i, σ_i) the operations of moving to the left and upper neighbour commute. As indicated above, $\mathrm{ch}\, P_i$ can be obtained by tracing left neighbours, starting from the rightmost point in row i; moreover $\mathrm{ch}\, Q_i$ can be similarly obtained by tracing upper neighbours starting from the same point (in the latter case we find, apart from the first one, the same partitions as for $\mathrm{ch}\, Q_{i-1}$). In particular, both final tableaux P, Q can be read off from the picture: in the example we get

$$P = \begin{array}{|c|c|c|}\hline 1 & 3 & 4 \\ \hline 2 & 6 & 8 \\ \hline 5 & 7 \\ \cline{1-2}\end{array} \qquad \text{and} \qquad Q = \begin{array}{|c|c|c|}\hline 1 & 3 & 6 \\ \hline 2 & 5 & 7 \\ \hline 4 & 8 \\ \cline{1-2}\end{array}.$$

The recursive definition (12) may now be interpreted as giving the following rules for constructing the picture. The first clause says that to points (i, σ_i) we ascribe the partition obtained from the one of its

base point by increasing the first part by one. For any other non-zero point (i,j) we first need to know the shapes ascribed to its upper and left neighbour; in the setting of (12) these are $\lambda = \mathrm{sh}\,T$ and $\lambda' = \mathrm{sh}\,P'$ respectively, and $\lceil T \rceil$ and s' are the squares by which they differ with respect to the shape $\mathrm{sh}\,T^-$ ascribed to their common (left respectively upper) neighbour. Therefore the condition $s' \neq \lceil T \rceil$ is equivalent to $\lambda' \neq \lambda$, and when it holds the partition ascribed to (i,j) is $\lambda \cup \lambda'$ (which is in fact the only possibility if we realise that the difference between diagrams ascribed to neighbours is always just one square). In the remaining case we have that $s' = \lceil T \rceil$ is a corner of $\lambda' = \lambda$; we then ascribe to (i,j) the partition obtained by adding to λ' its cocorner s in the row following s'. Theorem 3.1 follows immediately from the symmetry with respect to transposition (of the points of Σ_σ, not the partitions ascribed to them!) of these rules. ⊞

Remarks.

(1) The picture makes clear why our formal proof needs three applications of the induction hypothesis: namely for the upper and left neighbour of the point under consideration, and for their common (left respectively upper) neighbour.

(2) There is a relation between our picture on one side and the "graph-theoretical viewpoint" of Knuth [Kn1], §4, and the (essentially identical) *"forme géométrique de la correspondance de Robinson-Schensted"* of Viennot [Vien] on the other side. Both these authors start by considering the set $\{\,(i,\sigma_i) \mid 1 \leq i \leq n\,\}$, partially ordered by the natural ordering $\leq$ on $\mathbf{N} \times \mathbf{N}$, classifying the points by the maximal length of a chain descending from them (it is easy to verify that this length equals the first part of the partition we have ascribed to such a point). However, from this only partial information is deduced directly (amounting to to the first rows of both tableaux), and an iterative procedure is needed to obtain a complete result: a new, smaller, set of points is constructed, for which the same construction is performed (giving the second rows), and this is repeated until the set of points is empty.

(3) The slightly irregular structure of our picture due to the fact that partitions are only ascribed to points in Σ_σ can be eliminated by copying to any other point the partition ascribed to the maximal point in Σ_σ smaller than it. Then each arrangement of partitions $\begin{smallmatrix}\mu & \lambda \\ \lambda' & \nu\end{smallmatrix}$ falls into one of the following cases: (a) $\mu = \lambda = \lambda'$, and ν is obtained from these by increasing the first part; (b) $\lambda = \lambda'$, and $\mu \in \lambda^-$ corresponds to $\nu \in \lambda^+$ (cocorner in the row following the corner); (c) $|\lambda| = |\lambda'|$ but $\lambda \neq \lambda'$ and $\mu = \lambda \cap \lambda'$, $\nu = \lambda \cup \lambda'$; (d) $\mu = \lambda \wedge \lambda' = \nu$ or $\mu = \lambda' \wedge \lambda = \nu$ or both. Case (a) applies if and only if the position of ν is some (i,σ_i), and case (d) if and only if that position is not in Σ_σ. In this setup it easy to see that the pictorial construction is also invertible, i.e., as initial data one may take instead of the permutation σ the chains $\mathrm{ch}\,P$ and $\mathrm{ch}\,Q$ of partitions along the right and bottom edges.

(4) Defining the structure, as we did, on Σ_σ only, dispenses with the partitions ν for which case (d) applies, without complicating the rules very much; note however that we can not as easily dispense with case (c), even though that case does not explicitly occur in the basic formulation of the Robinson-Schensted algorithm. Indeed there is no simple direct description of the set of points for which either (a) or (b) applies: it can be defined only by an iterative construction. Each partition ν ascribed to such a point has a distinguished corner by which it differs from its upper and left neighbour, and this corner lies in row 1 precisely when case (a) applies. From the set of points for which this corner lies in row r it is not difficult to construct the (smaller) set of points for which it lies in row $r+1$. This construction coincides with the iterative constructions of Knuth and Viennot mentioned under (2). Including case (c) is what makes our approach simpler than those ones.

(5) To any finite partially ordered set one may associate a partition, as is shown in [Gre2] and [Fom1]. Then, the partition we have ascribed to a point (i,j) is just the partition associated to the truncation to points $\leq (i,j)$ of the partially ordered set of points (i,σ_i) mentioned under (2). A non-trivial point to be proved in this approach is that one has inclusions of Young diagrams along the rows and columns. Along these lines a picture like ours has been constructed by Fomin in [Fom2]; however, we arrived at our construction by the proof of 3.1, and were unaware at the time of this earlier construction.

We now return to the statement of 3.1, and study the fixed points of the stated symmetry. Clearly

the map $P \mapsto R(P,P)$ defines a bijection, say Δ, from $\bigcup_{\lambda\in\mathcal{P}_n} \mathcal{T}_\lambda$ to the set of involutions in $\mathbf{S}_n$. In fact, if $\sigma \in \mathbf{S}_n$ is known to be an involution, then there is a method for finding $P = \Delta^{-1}(\sigma)$ which is slightly more efficient than computing $R^{-1}(\sigma)$; it is based on the observation that the picture described above will be symmetric about the main diagonal in this case, and therefore only half of it needs to be computed. The following informal description describes the reasoning. As the successive numbers σ_i are considered, we compute the shape at the points (i,i) along the main diagonal, or more precisely at the maximum within Σ_σ of points $\leq (i,i)$ (this means that at step i we ignore all squares with entries $> i$). Any number σ_i with $\sigma_i > i$ has no effect on the transition from $(i-1,i-1)$ to (i,i), so it is simply skipped over. A number $\sigma_i = i$ is incorporated by adding it at the end of the first row. A number $\sigma_i < i$ is first inserted into the tableau T in the usual way by computing $(P,s) = I(T,\sigma_i)$. Now $\mathrm{sh}\,P$ is the partition ascribed to the left neighbour of (i,i); by symmetry the partition ascribed to its upper neighbour is also $\mathrm{sh}\,P$. The latter partition gives the shape that T would have had, if we had not chosen to ignore the number i when it occurred (at position σ_i in the sequence) during the construction of T. We conclude that in that case the number i would have occupied square s at the time that σ_i was inserted, and would be pushed to the next row by that insertion. Therefore, to complete the incorporation of the number σ_i, we add i to P at the cocorner in the row below the square s.

The following formal algorithm describes the process (like in (10) we simultaneously compute the length of σ).

$$
\begin{aligned}
\Delta^{-1}(\sigma) &= \textbf{let } (P,n) = \Delta'(\sigma) \textbf{ in } P \qquad \text{[the length } n \text{ is not needed in the end]},\\
\Delta'(\sigma) &= \textbf{if } \sigma = () \textbf{ then } (\odot, 0)\\
&\qquad \textbf{else let } (\sigma' \,\&\, m) = \sigma;\ (T,n) = \Delta'(\sigma');\ i = n+1;\ \lambda = \mathrm{sh}\,T\\
&\qquad\quad ;\ P' =\\
&\qquad\qquad \textbf{if } m > i \textbf{ then } T \textbf{ else if } m = i \textbf{ then } T : ((1,\lambda_1+1),m)\\
&\qquad\qquad \textbf{else let } (P,(r,c)) = I(T,m) \textbf{ in } P : ((r+1,\lambda_{r+1}+1),i)\\
&\qquad\qquad \textbf{fi fi}\\
&\qquad\quad \textbf{in } (P',i)\\
&\qquad \textbf{fi}
\end{aligned}
\tag{13}
$$

The opposite procedure Δ does not have an equally simple recursive description, since the terms of the sequence σ are not computed in the right order.

Remark. Most of the things discussed up to this point can be done *mutatis mutandis* for an arbitrary differential poset instead of the Young lattice, once a bijectivisation of (1) is given (for (2) this is not necessary, since both sides can be shown to be at most 1). Young tableaux will be replaced be chains in the differential poset, and squares by edges in its Hasse diagram; the recursive definitions of E and I can be adapted to this setting (unlike the original iterative ones: there are no rows or columns) by inserting the bijections in the proper places, and the definitions of R and R^{-1} remain the same. Theorem 3.1, its proof and pictorial representation carry over to this context, as well as the bijections Δ and Δ^{-1} (with suitably adapted definition). There is no analogue however for the transposed algorithms: although each bijectivisation of (1) gives rise to its own algorithm, changing the bijection need not be reflected by an operation being applied to the individual chains (similar to transposition of tableaux).

The computation of Δ^{-1} has the following implication, which is due to Schützenberger, [Schü1], §4.

3.2. Proposition. *Let $\lambda \in \mathcal{P}_n$ and let $k = \sum_{i>0}(-1)^{i-1}\lambda_i$ be the number of columns of $Y(\lambda)$ of odd length, then for each tableau $P \in \mathcal{T}_\lambda$ the permutation $R(P,P)$ is an involution with k fixed points.*

Proof. Since the number of columns of $Y(\lambda)$ of length i is $\lambda_i - \lambda_{i+1}$ it is clear that k is indeed the number of odd-length columns, so it remains to show that k is equal to the number of fixed points of $\sigma = \Delta(P)$. This follows from the description of Δ^{-1} indicated above, since any term $\sigma_i = i$ contributes 1 to k by increasing the first part of $\lambda = \mathrm{sh}\,P$, while any pair $\{i,\sigma_i\}$ of distinct numbers exchanged by σ has the effect of increasing two consecutive parts of λ, thereby not affecting k. □

Further properties of the Robinson-Schensted correspondence will be formulated after we have introduced the Schützenberger algorithm. We can however already formulate the following statement (due

to Schensted), whose proof will follow from a stronger statement proved later. Let I_m denote the operation of inserting a number m into a tableau T not containing it, where only the tableau yielded is considered (so if $I(T,m)=(P,s)$ then $P=I_m(T)$); also let I^t_m be its transposed counterpart. Then *every I_m commutes with every $I^t_{m'}$*, i.e., for $m\neq m'$ and every tableau T not containing m or m' we have $I_m(I^t_{m'}(T))=I^t_{m'}(I_m(T))$. Since moreover $I_m(\odot)=I^t_m(\odot)$ we may convert the application of a composition of operators I_m to the empty tableau $\odot$ into an application of the reversed composition of the corresponding I^t_m to $\odot$. Consequently, reversing the order of the sequence σ affects the left tableau ('P') of the pair $R^{-1}(\sigma)$ by simply transposing it (but nothing is said here about the right tableau). Historically it was this fact, and the wish to extend the statement to describe the right tableau as well, which led to the Schützenberger algorithm described in the next section.

§4. The Schützenberger algorithm.

In this section we introduce an algorithm due to Schützenberger that is intimately related to the Robinson-Schensted algorithm. The Schützenberger algorithm defines a shape preserving transformation of normalised tableaux, i.e., for each $\lambda\in\mathcal{P}$ it defines a map $S\colon\mathcal{T}_\lambda\to\mathcal{T}_\lambda$. Like the Robinson-Schensted algorithm, it is based on the repeated application of a basic procedure that modifies a given tableau in a specific manner. In the current case we shall call this the *deflation* procedure D, since it starts by emptying the square in the upper left-hand corner, and the proceeds to rearrange the remaining squares to form a proper tableau. Like the other procedures we have seen, D can be reversed step by step, giving rise to an *inflation* procedure D^{-1}. More precisely, these procedures convert into each other the following sets of data: on one hand a non-empty tableau P, and on the other hand a tableau T, a specified cocorner s of $\mathrm{sh}\,T$, and a number m that is smaller than all entries of T; we write $(T,s,m)=D(P)$ and $P=D^{-1}(T,s,m)$. These procedures are such that we always have the following relations: the set of entries of P are those of T together with m, and $Y(\mathrm{sh}\,P)=Y(\mathrm{sh}\,T)\cup\{s\}$.

> *Deflation procedure.* Given a tableau P, the triple $(T,s,m)=D(P)$ is computed as follows. The first step is to put m equal to the smallest entry of P, and remove that entry, leaving an empty square in position $(1,1)$. Then the following step is repeated until the empty square is a corner of the shape $\mathrm{sh}\,P$ of the original tableau: move into the empty square the smaller one of the entries located directly to the right of and below it (if only one of these positions contains an entry, move that entry). When the position of the empty square finally is a corner of $\mathrm{sh}\,P$, then s is defined to be this corner, and T the tableau formed by the remaining non-empty squares.

Because the empty square moves down or to the right in each step, termination is evidently guaranteed. That T is indeed a tableau can be seen by observing that at each stage of the process the entries of the non-empty squares remain increasing along each row and column. In fact, when there are entries both to the right and below the empty square, the choice to move the smaller one is dictated by the tableau property.* By the same consideration it also becomes clear that D is invertible, and that its inverse procedure D^{-1} can be defined as follows:

> *Inflation procedure.* Given a tableau T, a cocorner s of $\mathrm{sh}\,T$ and a number m smaller than any of the entries of T, the tableau $P=D^{-1}(T,s,m)$ is computed as follows. The first step is to attach an empty square to T at position s. Then the following step is repeated until the empty square is at position $(1,1)$: move into the empty square the larger one of the entries located directly to the left of and above it (if only one of these positions contains an entry, move that entry). When the empty square has arrived at position $(1,1)$, it is filled with the number m to form the tableau P.

One easily verifies that the procedure reverses D step by step, and also preserves the tableau property.

* The rule stating which square to displace during the insertion procedure of the Robinson-Schensted algorithm cannot be characterised in such a way, since displacing the last *smaller* entry might equally well preserve the tableau property; the definition given in [Schü1], §1 of that insertion procedure is therefore incorrect.

We demonstrate these procedures by an example:

$P =$

1	2	5	10
3	4	9	
6	7	11	
8			

	2	5	10
3	4	9	
6	7	11	
8			

2		5	10
3	4	9	
6	7	11	
8			

2	4	5	10
3		9	
6	7	11	
8			

2	4	5	10
3	7	9	
6		11	
8			

2	4	5	10
3	7	9	
6	11		
8			

so that we have

$T =$

2	4	5	10
3	7	9	
6	11		
8			

, $\qquad s = (3,3), \qquad m = 1.$

Before we continue it is convenient to introduce the following notations.

4.1. Definition.

(i) Let P be a non-empty tableau, and $(T, s, m) = D(P)$. We define $P^{\downarrow} = T$.

(ii) Let $x = (r, c)$ and y be distinct squares. The relation $x \parallel y$ is defined to hold if either $y = (r+1, c)$ or $y = (r, c+1)$. In this case x and y are called adjacent.

In (i), the arrow is meant to suggest the lowest entry of P being squeezed out. It provides a convenient abbreviation, but we shall avoid using it in formal expressions, where we prefer to explicitly name the deflation procedure D. In (ii), which is on the other hand specifically intended for formal expressions, we did not include the cases that x lies to the right or below y, since we intend to apply it only in those cases when x is a corner and y a cocorner of one same shape.

There are recursion relations for D and D^{-1} similar to those for I and E. For any tableau P with at least two squares we compare the computation of $D(P)$ with that of $D(P^{-})$. Since the highest entry h of P lies at some corner of $\operatorname{sh} P$, it can only be moved in the final step; we conclude that $P^{\downarrow -} = P^{-\downarrow}$. Therefore we have, similarly to the case of I,

$$\text{if } D(P) = (T, s, m) \text{ then } D(P^{-}) = (T^{-}, s', m) \text{ for some square } s'. \tag{14}$$

Here too the relation between s and s' is expressed by a pair of additional requirements; the first is

$$\{s, \lceil T \rceil\} = \{s', \lceil P \rceil\} \tag{15}$$

since both sets are equal to $Y(\operatorname{sh} P) \setminus Y(\operatorname{sh} T^{-})$. Now if $s' \parallel \lceil P \rceil$ then s' is not a corner of $\operatorname{sh} P$, and since on the the other hand s is a corner of $\operatorname{sh} P$, it follows directly that $s = \lceil P \rceil$ and $\lceil T \rceil = s'$. In addition to (15) it therefore suffices to require

$$\text{if } s' \not\parallel \lceil P \rceil \text{ then } s = s'. \tag{16}$$

Since we already know that D and D^{-1} are inverse operations, the recurrence relations for D^{-1} (for non-empty T) can be derived from those for D. From (14) we get

$$\text{if } D^{-1}(T, s, m) = P \text{ then } D^{-1}(T^{-}, s', m) = P^{-} \text{ for some square } s', \tag{17}$$

where again s and s' are related by (15) and (16); the latter equation is more naturally written as

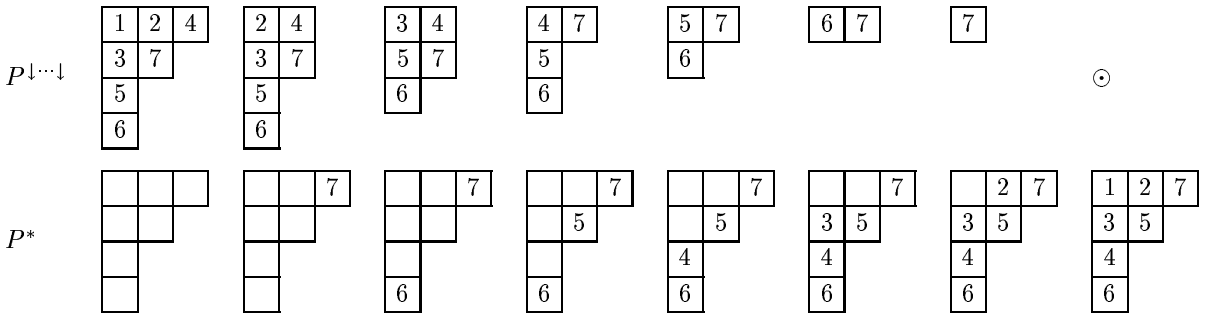
$$\text{if } s \not\parallel \lceil T \rceil \text{ then } s' = s. \tag{18}$$

As is the case for the Robinson-Schensted algorithm, the full Schützenberger algorithm essentially consists of repetition of the basic procedure. Repeating the application of the deflation procedure to P, we find a sequence of tableaux $P, P^{\downarrow}, P^{\downarrow\downarrow}, \ldots, \odot$, whose shapes form a saturated decreasing chain in the Young lattice starting in $\lambda = \operatorname{sh} P$; this chain equals $\operatorname{ch} P^{*}$ for a unique $P^{*} \in \mathcal{T}_{\lambda}$, and $S(P)$ is defined

to be this normalised tableau P^*. Note that while the smallest entries of P are removed first, it is the largest entries of P^* whose position is determined first, since if $D(P) = (P^\downarrow, s, m)$, then $\lceil P^* \rceil = s$.

By definition the algorithm S always returns normalised tableaux, and is constant on similarity classes. This obviously precludes reversibility of S on the set of all tableaux, but that is not a major problem, because we view tableaux as essentially representing just chains of partitions, and our main interest lies in applying S to normalised tableaux. Yet, it would have been possible to define S on the set of all tableaux in a reversible way, namely by changing the definition so that the entries of P^* are the negatives of those of P; then for any step $(P^{\downarrow(i)}, s, m) = D(P^{\downarrow(i-1)})$ occurring during the computation of $S(P)$, the square s would contain the entry $-m$ in $S(P)$. This would have certain advantages, but it would also imply that S fails to transform normalised tableaux into normalised ones, and it would agree with our actual definition only up to similarity.

The inverse algorithm S^{-1} of S on the set of normalised tableaux, which recomputes P from P^*, is the following: set P_0 equal to the empty tableau, and successively compute $P_i = D^{-1}(P_{i-1}, P^*[i], n+1-i)$ for $i = 1, \ldots, n$, where $P^*[i]$ denotes the square with entry i in P^*; then $P = S^{-1}(P^*) = P_n$. We give an example of performing the algorithm S: we display the successive stages $P, P^\downarrow, P^{\downarrow\downarrow}, \ldots$, and meanwhile the entries of P^* that are determined up to this point. Reading from right to left illustrates the computation of $S^{-1}(P^*)$, where those entries of P^* that have already served their purpose are erased.



It is obvious from the definition that S commutes with transposition: $S(P^t) = S(P)^t$.

Like for the Robinson-Schensted algorithm we can give formal recursive definitions of these algorithms, which express how the whole computation can be based on the recursion relations (14)–(18).

$$\begin{aligned} D(P) = {} & \textbf{if } P^- = \odot \textbf{ then } (\odot, \lceil P \rceil, P_{\lceil P \rceil}) \quad [\text{here } \lceil P \rceil = (1,1)] \\ & \textbf{else let } (T', s', m) = D(P^-) \\ & \quad ; \ (t, s) = \textbf{if } s' \parallel \lceil P \rceil \textbf{ then } (s', \lceil P \rceil) \textbf{ else } (\lceil P \rceil, s') \textbf{ fi} \\ & \quad \textbf{in } (T' : (t, P_{\lceil P \rceil}), s, m) \\ & \textbf{fi} \end{aligned} \tag{19}$$

$$\begin{aligned} D^{-1}(T, s, m) = {} & \textbf{if } T = \odot \textbf{ then } \odot : (s, m) \quad [\text{here } s = (1,1)] \\ & \textbf{else let } (s', p) = \textbf{if } \lceil T \rceil \parallel s \textbf{ then } (\lceil T \rceil, s) \textbf{ else } (s, \lceil T \rceil) \textbf{ fi} \\ & \quad \textbf{in } D^{-1}(T^-, s', m) : (p, T_{\lceil T \rceil}) \\ & \textbf{fi} \end{aligned} \tag{20}$$

The full Schützenberger algorithm and its inverse are given by

$$S(P) = \textbf{if } P = \odot \textbf{ then } \odot \textbf{ else let } (T, s, m) = D(P) \textbf{ in } S(T) : (s, |\operatorname{sh} P|) \textbf{ fi} \tag{21}$$

$$\begin{aligned} S^{-1}(P) &= S'(P, 1) \\ S'(P, i) &= \textbf{if } P = \odot \textbf{ then } \odot \textbf{ else } D^{-1}\big(S'(P^-, i+1), \lceil P \rceil, i\big) \textbf{ fi} \end{aligned} \tag{22}$$

Note that (21) uses the fact that S may be applied to arbitrary tableaux, but always yields a normalised result (the intermediate tableau T is not normalised). For S^{-1} an auxiliary procedure S' with an additional parameter i was introduced to ensure that the proper entries are inserted during the calculation

of $S^{-1}(P)$; one easily verifies that $S'(P,i) \sim S^{-1}(P)$ and the entries of $S'(P,i)$ form an interval of $\mathbf{Z}$ starting at i. These complications would not have occurred if we had worked with chains of partitions rather than tableaux.

The Schützenberger algorithm, like the Robinson-Schensted algorithm, has an unexpected symmetry, which is expressed by the following

4.2. Theorem. *For all $\lambda \in \mathcal{P}$ the algorithm S defines an involution $\mathcal{T}_\lambda \to \mathcal{T}_\lambda$, i.e.,*

$$S(P) = S^{-1}(P)$$

for all $P \in \mathcal{T}_\lambda$.

This fact was first stated and proved by Schützenberger in [Schü1], Sec5, but the proof is indirect, being based on the relation of the Schützenberger algorithm with the Robinson-Schensted algorithm that we shall formulate in the next section. To our knowledge the following is the first direct proof of the theorem.

Proof. We prove the theorem by induction on $n = |\lambda|$. For $n < 2$ the statement is obviously true, so assume $n \geq 2$. Following (21) let $(T, s, 1) = D(P)$; we have by definition

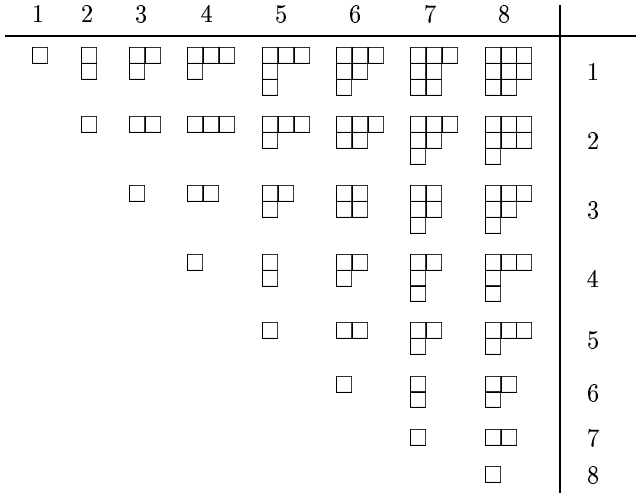
$$S(P) = S(T) : (s, n). \qquad (*)$$

According to (19) we have $D(P^-) = (T^-, s', 1)$ for some square s', and $(\lceil T \rceil, s)$ is determined from $(s', \lceil P \rceil)$ as in the third line of (19); from this it follows by (21) that $S(P^-) = S(T^-) : (s', n-1)$. Now put $U = S'(P^-, 2)$; it follows from the induction hypothesis applied to P^- that $U \sim S(P^-)$, so that $\lceil U \rceil = s'$ and $U^- \sim S(T^-)$. Applying the induction hypothesis for T^- to the latter statement, and noting that the entries of U form an interval of integers starting at 2, we obtain

$$U^- = S'(T^-, 2). \qquad (**)$$

Turning to the definition of $S^{-1}(P)$, we see that it equals $D^{-1}(U, \lceil P \rceil, 1)$. Renaming variables appropriately in (20) for the computation of $D^{-1}(U, \lceil P \rceil, 1)$, we find as right hand side of '**let** $(s', p) =$' the same expression that defined $(\lceil T \rceil, s)$ above, so we may conclude that $S^{-1}(P) = D^{-1}(U, \lceil P \rceil, 1) = D^{-1}(U^-, \lceil T \rceil, 1) : (s, n)$. Comparing this to $(*)$ we see that to complete the proof it suffices to show that $S(T) = D^{-1}(U^-, \lceil T \rceil, 1)$. To this end we use a final application of the induction hypothesis for T; expanding (22) for the computation of $S'(T, 1)$ and using $(**)$, the desired equation follows. □

Like for the proof of 3.1 there is a nice pictorial rendering of this proof. Again we associate partitions to a set of points in the plane; in the present case this set is the upper triangular part of a region of $n \times n$ points. Each row displays the partitions in some $\operatorname{ch} P^{\downarrow \cdots \downarrow}$, in ascending order. More precisely, assuming P is normalised, the partition at position (i, j) is $\operatorname{sh} P^{\downarrow \cdots \downarrow - \cdots -}$, where all entries $< i$ are removed by iterating the operator $\cdot^{\downarrow}$, and all entries $> j$ by iterating the operator $\cdot^-$. The chain $\operatorname{ch} S(P)$ of the computed tableau can be read off in the final column. As an example, illustrating the computation which for $P = \begin{array}{|c|c|c|}\hline 1&3&4\\\hline 2&6&8\\\hline 5&7\\\cline{1-2}\end{array}$ computes $S(P) = \begin{array}{|c|c|c|}\hline 1&2&4\\\hline 3&6&7\\\hline 5&8\\\cline{1-2}\end{array}$, we have:



The rule by which the picture is constructed is given by (19): when in some arrangement of partitions $\begin{smallmatrix}\kappa & \lambda \\ \mu & \nu\end{smallmatrix}$ the difference $Y(\lambda) \setminus Y(\mu)$ consists of two adjacent squares then κ and ν are both necessarily equal to the unique intermediate partition between λ and μ in the Young lattice, but otherwise (there are two intermediate partitions and) ν is chosen to be distinct from κ. In this way, starting from the top row, each successive row is constructed from left to right. The proof of 4.2 now comes down to proving that the same picture also displays the computation of $S^{-1}(P)$, where chains of the intermediate tableaux can be read in successive columns (proceeding from left to right). This proof succeeds because (20) effectively enforces the same rule for computing ν from κ, λ and μ. Another way of viewing this is that that rule is stated in such a way that it equally well allows us to determine κ from ν (for given λ, μ) as ν from κ, so that the computation of $S(S(P))$ from $S(P)$ is just the mirror image (in the anti-diagonal bisecting the triangular shape) of the computation of $S(P)$ from P, whence $S(S(P)) = P$. ⊞

Having established this symmetry of the Schützenberger correspondence it is interesting—as it was in the case of the Robinson-Schensted correspondence—to consider the fixed points of the symmetry. In the current case that are tableaux P with the property that $S(P) \sim P$. It will prove to be convenient to impose the additional restriction that the set of entries of P be symmetric around 0 (note that these are exactly the tableaux for which we would have $S(P) = P$ for the modified definition of S suggested above, where the negatives of the entries removed from P are inserted into $S(P)$).

4.3. Definition. *A Young tableau P is called a self-dual tableau if $S(P) \sim P$, and moreover for each i occurring as an entry of P the number $-i$ also occurs as entry of P. A normalised self-dual tableau is a self-dual tableau whose set of entries moreover forms a complete interval in $\mathbf{Z}$ from $-n$ to $+n$ for some n, with the possible exclusion of the number 0.*

For such tableaux the picture described above becomes symmetric about its anti-diagonal, where as before we do not reflect the Young diagrams themselves. This implies that for any configuration $\begin{smallmatrix}\kappa & \lambda \\ \mu & \nu\end{smallmatrix}$ with λ, μ on the anti-diagonal we have $\kappa = \nu$, whence, by the rule for constructing the picture, this must be the *only* intermediate partition between λ and μ. It follows that for any pair of successive partitions along the anti-diagonal their Young diagrams differ by a pair of adjacent squares, which we shall term a *domino*. Conversely, any tableau for which all partitions along the anti-diagonal are related in this way must be self-dual, since any partition directly above or below the anti-diagonal is uniquely determined by interpolation, and all other partitions follow by the rule for constructing the picture. This sequence of partitions can be encoded by filling a Young diagram with numbered dominoes; it will be useful to view these as special cases of a slightly more general class of combinatorial objects†.

4.4. Definition. *Let $r \in \mathbf{N}$ and $\lambda \in \mathcal{P}$; a domino tableau of rank r and shape λ is a Young diagram $Y(\lambda)$ filled with non-negative integers, such that 0 occurs at position (i, j) if and only if $i + j - 2 < r$, each other occurring number occurs precisely in a pair of adjacent squares, and such that entries are weakly increasing along both rows and columns. A total domino tableau is a domino tableau of rank 0 or 1. A domino tableau is called normalised if its set of positive entries is $\{1, \ldots, n\}$ for some n.*

For a domino tableau T the notation $\lceil T \rceil$ denotes the outer square of the highest numbered domino of T, i.e., the one which is a corner of $\mathrm{sh}\, T$. The meaning of T^- and the operator ':' are defined correspondingly. The set of positions with entry 0 equals $Y(\mu)$, where $\mu = (r, r-1, \ldots, 2, 1)$ is a "staircase partition" called the *core* of T; one cannot remove any dominoes from $Y(\mu)$ leaving another Young diagram. The shapes of the sequence $T, T^{--}, T^{----}, \ldots$, form a decreasing chain from λ to μ in the Young lattice, whose successive terms differ by a domino. If $\mu = (0)$ or $\mu = (1)$, this chain forms the anti-diagonal of the picture for the Schützenberger algorithm corresponding to a unique self-dual Young tableau. This informally proves the following

4.5. Proposition. *For any partition λ the set of normalised self-dual tableaux of shape λ is in bijection with the set of normalised total domino tableaux of shape λ.*

† the idea to do this was suggested by I. G. Macdonald, in connection with the 'Robinson-Schensted algorithm for hyperoctahedral groups' which is described in the next section.

We leave a detailed formal proof to the reader, but we shall give the main ingredient: a (non-pictorial) procedure A for computing the domino tableau corresponding to a given self-dual tableau P, and its inverse procedure A^{-1}. The given procedures do not require the self dual tableau P or the total domino tableau T to be normalised, and will compute a tableau with the same set of non-negative entries as their parameter.

$$A(P) = \textbf{if } |\operatorname{sh} P| \leq 1 \textbf{ then } P \quad \textbf{else let } (T,s,m) = D(P^{--});\ n = P_{\lceil P \rceil}\ [=-m] \textbf{ in } A(T) : (s,n) : (\lceil P \rceil, n) \quad \textbf{fi} \tag{23}$$

$$A^{-1}(T) = \textbf{if } |\operatorname{sh} T| \leq 1 \textbf{ then } T \quad \textbf{else let } n = T_{\lceil T \rceil} \textbf{ in } D^{-1}(A^{-1}(T^{--}), \lceil T^{-} \rceil, -n) : (\lceil T \rceil, n) \quad \textbf{fi} \tag{24}$$

One immediate consequence of this proposition is that a necessary and sufficient condition for a shape λ to admit any self-dual tableaux is that it admits total domino tableaux, i.e., that $Y(\lambda)$ or $Y(\lambda) \setminus \{(1,1)\}$ can be tiled with dominoes (it is easy to see that such a tiling can always be numbered so as to make it into a total domino tableau). To decide this is an easy matter, given that any shape admits domino tableaux of one unique rank only: one repeatedly removes dominoes from $Y(\lambda)$, leaving a Young diagram after each step, until a staircase shape $(r, r-1, \ldots, 2, 1)$ is obtained; if $r \leq 1$ a total domino tableau has been found, and otherwise none exist. The fact that λ admits only one rank is obvious if one draws $Y(\lambda)$ on a (suitably extended) chess board, and realises that counting black and white squares, the balance is different for each possible core, and unaffected by any domino.

Remark. A less opportunistic proof of this fact runs as follows. Call each boundary line segment (horizontal or vertical) between a square inside and one outside $Y(\lambda)$ an edge of $Y(\lambda)$, then $Y(\lambda)$ can be characterised by listing the orientations of its edges from upper right to lower left; we may take the sequence to start with infinitely many horizontal edges and similarly end with vertical edges. Such a sequence has a uniquely determined midpoint between two successive edges, such that there are equally many vertical edges before it as there are horizontal edges after it (this is the point where the main diagonal meets the boundary of $Y(\lambda)$); moreover this midpoint does not move whenever we interchange the orientations of any pair of edges. Now the basic observation is that the removal of a domino—regardless of whether it is a horizontal or a vertical one—is equivalent to interchanging the orientations of two edges which are two places apart, from vertical-horizontal to horizontal-vertical. Therefore if we split the sequence of edge orientations alternatingly into two subsequences, each domino removal will affect just one of these subsequences, and no further removal is possible when both subsequences consist of a sequence of just horizontal edges, followed by just vertical edges, with the turnover point being the midpoint defined for that subsequence. So the core is predetermined by the displacement of these midpoints for the subsequences, relative to the position inherited from the midpoint of the full sequence (the average displacement necessarily being zero). This analysis in fact defines a bijection from the set of shapes λ which admit domino tableaux of rank r with n dominoes to the set of ordered pairs (μ, ν) of partitions with $|\mu| + |\nu| = n$ (and in particular shows their number to be independent of r), which in turn implies a bijection from the set of normalised domino tableaux of such a shape λ corresponding to the pair (μ, ν), to the set of arrangements of the numbers $1, \ldots, n$ into an ordered pair of Young tableaux of shapes μ and ν, using each number once. The line of reasoning can also be directly generalised, replacing 2 by an arbitrary fixed positive integer q, in which case dominoes are replaced by so-called rim hooks of size q, and staircase shaped cores by q-cores (see for instance [JaKer], 2.7.16 or [FomSt]). Incidentally, the next section will provide yet another proof of the fact that any shape admits domino tableaux of one rank only.

As we shall see in the next section, arbitrary domino tableaux are in a sense a natural generalisation of total domino tableaux, but since only total domino tableaux correspond to self-dual Young tableaux, it remains to be seen whether another sensible interpretation can be given uniformly to all domino tableaux in this context. One fact about them is the following. Suppose we replace in the totally ordered set $\mathbf{Z}$ the element 0 by a sufficiently large ordered collection of "infinitesimal numbers"—which will be assumed to be neither positive nor negative—then we may fill the core of a given domino tableau in an arbitrary way

with such numbers so that it becomes an (infinitesimal) Young tableau. If we then apply procedure A^{-1}, with the condition $|\operatorname{sh} T| \leq 1$ replaced by the condition that $\operatorname{sh} T$ is the core, we obtain a tableau X with non-zero ordinary numbers mixed with infinitessimals. One easily shows that the positions of all positive numbers in X are the same as in X^*, where $X^* \sim S(X)$ is renumbered to obtain the same set of entries as X, and moreover these positions are independent of the original arrangement of the infinitesimals in the core (since the inflation procedure D^{-1} affects smaller numbers only after the larger ones are settled). Much less obviously, the same statements also hold for the negative numbers, because as we shall prove in §5, the positions of the k highest entries of any tableau T determine positions of the k lowest entries of $S(T)$. In this way domino tableaux may be considered to represent equivalence classes of Young tableaux which are "as self-dual as possible" given their shape, in the sense that there are only as many infinitessimals as the size of the core for the rank corresponding to that shape, and dualising only permutes the infinitesimal entries; equivalence is defined by all ordinary entries having the same positions. Note however that in these tableaux not all permutations of the infinitesimals compatible with the tableau condition necessarily are allowed: for some such permutations application of A may fail to produce dominoes, and hence to rearrange the infinitesimals back into the core. This fact prevents us from forgetting altogether about the arrangement of the infinitesimals.

§5. Relating the two algorithms.

We come now to the important theorem that exhibits the relationship between the Robinson-Schensted and Schützenberger algorithms.

5.1. Theorem. *Let $\lambda \in \mathcal{P}_n$ and $P, Q \in \mathcal{T}_\lambda$, and let $\sigma \in \mathbf{S}_n$ be determined by*

$$\sigma = R(P, Q)$$

then we also have

$$\tilde{w}\sigma = R^t(S(P), Q) \tag{25}$$

$$\sigma\tilde{w} = R^t(P, S(Q)) \tag{26}$$

$$\tilde{w}\sigma\tilde{w} = R(S(P), S(Q)) \tag{27}$$

where $\tilde{w} \in \mathbf{S}_n$ is the "order reversing" permutation given by $\tilde{w}_i = n + 1 - i$.

The permutation $\sigma\tilde{w}$ has as sequence of numbers the reverse of that of σ, so (26) generalises the fact announced earlier that under R^{-1} this reversal leads to transposition of the left tableau. That statement about the left tableau was proved by Schensted in [Sche], Lemma 7, while the remainder of the identity is proved by Schützenberger in [Schü1], §5. Alternative proofs have been given for Schensted's lemma (or equivalently, of the commutation of the insertion and transpose insertion procedures), but we know of no independent proof of Schützenberger's statement (although it does follow quite implicitly from the analysis in [Schü2]). This is unfortunate since [Schü1] not only very hard to understand (due in part to its cryptic notation and numerous minor errors), but the mentioned proof is incomplete in an essential way. It is not difficult to see that Schensted's lemma alone (together with the more obvious properties of the Robinson-Schensted correspondence) is not sufficient to prove (26) once S has been defined; it is necessary to use the following fact as well. If we write down a commuting square for the operations of insertion and transpose insertion as applied to a tableau of shape μ, and the shapes of the tableaux involved are $\begin{smallmatrix}\kappa & \lambda \\ \mu & \nu\end{smallmatrix}$ (the arrows going upward and to the right), then the same condition as was mentioned in §4 for the construction of S holds: we can *only* have $\kappa = \nu$ if there is no other intermediate partition between μ and λ in the Young lattice (the case of adjacent squares). This fact however—which is actually needed even in the process of proving Schensted's lemma itself—is not mentioned by Schützenberger; this omission can be traced down to a *non sequitur* in the proof of [Schü1], Remarque 1, 1° cas. The proof below not only puts right this point, but also provides a more rigorous proof of Schensted's lemma, whose proof, although admirably free of technicalities, has a deceptive simplicity: many of its subtle points are not only left to the reader to prove, but often only become apparent after close inspection.

From (26) and the commutation of S with transposition one can immediately deduce 4.2, and using 3.1 one also obtains (25) and (27). The current formulation of our theorem (more or less) is due to Knuth,

and can be found in [Kn2], Theorem D. He expresses the remarkable character of the theorem as follows (p. 60)

> "The reader is urged to try out these processes on some simple examples. The unusual nature of these coincidences might lead us to suspect that some sort of witchcraft is operating behind the scenes! No simple explanation for these phenomena is yet known; there seems to be no obvious way to prove even that $[\tilde{w}\sigma\tilde{w}]$ corresponds to tableaux having the same *shape* as P and Q."

Indeed we shall see that the theorem has far-reaching consequences. That the statement is far from obvious can also be seen from the length of its proof, which we shall now take up.

Proof of 5.1. We shall prove (25), which like (26) will imply the other two equations. For $n = 0$ the theorem is obvious, so assume $n > 0$ and put $(T, m) = E(P, \lceil Q \rceil)$ and $(T^*, m^*) = E^t(S(P), \lceil Q \rceil)$. Then according to (9) and (9^t) we have $\sigma = R(T, Q^-)$ & m and $R^t(S(P), Q) = R^t(T^*, Q^-)$ & m^*, so we have to prove $m^* = \tilde{w}_m$ and $R^t(T^*, Q^-) = \tilde{w}R(T, Q^-)$, since left-multiplication of a permutation by $\tilde{w}$ amounts to replacing each term t of its sequence by $\tilde{w}_t = n + 1 - t$ (indeed this is what is meant by $\tilde{w}R(T, Q^-)$, which is not really a product of permutations). When $m^* = \tilde{w}_m$ has been established, then the sets of entries of $R^t(T^*, Q^-)$ and $\tilde{w}R(T, Q^-)$ will certainly match, so it will then suffice to prove $R^t(T^{*\prime}, Q^-) = \tilde{w}'R(T', Q^-)$ for the normalised counterparts T' and $T^{*\prime}$ of T and T^*, and the analogue $\tilde{w}'$ of $\tilde{w}$ in $\mathbf{S}_{n-1}$. This will follow by induction if we can show that $T^* \sim S(T)$. Therefore the proof of the theorem has been reduced to that of the following

5.2. Lemma. *Let P be a normalised tableau containing $n \geq 1$ squares, let q be a corner of* $\operatorname{sh} P$*, and let T, T^*, m, and m^* be defined by $(T, m) = E(P, q)$ and $(T^*, m^*) = E^t(S(P), q)$, then $m^* = n + 1 - m$ and $T^* \sim S(T)$.*

Proof. The proof of this lemma involves the discrimination of quite a number of cases. This number can be reduced somewhat by applying the following symmetry principle that follows from 4.2 and the commutation of S with transposition: if the lemma holds for some pair (P, q) then it also holds for $(S(P)^t, q^t)$ (transpose all the ingredients of the hypothesis, and observe that this leads to a conclusion equivalent to the original one). The use of 4.2 is not essential however: one can do without the symmetry principle, thereby admitting a setup (as in the original proofs) where 4.2 is derived from 5.1. To bring out this point, we shall briefly indicate how the symmetry principle could be avoided wherever we use it; for this reason we also retain the distinction between S and S^{-1}.

The first case that we single out is $m^* = n$, i.e., that $q = \lceil S(P) \rceil$ and this square lies in the first column, so that $E^t(S(P), q) = (S(P)^-, n)$ (computed without a recursive application of E^t), and $T^* = S(P)^-$. Let $U = S'(T^*, 2) \sim S^{-1}(T^*)$ (this U is to be proved equal to T), then from $P = S^{-1}(S(P)) = S'(S(P), 1)$ and (22) we have $P = D^{-1}(U, q, 1)$. From this we wish to prove that $E(P, q) = (U, 1)$; indeed, for any tableau U whose entries all exceed 1, and for its cocorner q in the first column, we have

$$E(D^{-1}(U, q, 1), q) = (U, 1). \tag{28}$$

This follows immediately by induction from the definitions of D^{-1} and E, and can also be visualised easily: in the inflation procedure D^{-1} all entries in the first column of U are moved one place down, and in the extraction procedure E they are moved back up again. Note that since the equation holds for arbitrary U, and E has an inverse I, we may reverse the reasoning, showing that any tableau P which for some q gives $m = 1$ must be of the form $D^{-1}(U, q, 1)$ with q in the first column. As by (21) we then also have $S(P)^- = S(U)$, it follows that moreover $q = \lceil S(P) \rceil$; therefore the condition $m = 1$ is equivalent to $m^* = n$.

The case that $m = n$ follows from the previous case by the symmetry principle. Alternatively, one may prove directly for all tableaux T that, with q its cocorner in the first row and $n = |\operatorname{sh} T| + 1$, we have $E^t(S(T : (q, n)), q) = (T^*, 1)$ with $T^* \sim S(T)$; this can then be applied to $T = P^-$. The latter proof requires more rewriting then the previous case did, since the algorithm for S "starts at the wrong end" for an easy computation of $S(T : (q, n))$, and we must expand the application of D contained in it, distinguishing the two cases that can occur. Nonetheless this proof too uses only a straightforward single induction.

Having dealt with the cases that $m = 1$ or $m = n$, we treat the remaining case by an induction argument; note that we now have $n \geq 3$. There are still a number of subcases to be distinguished, according to the positions of the three corners q, $\lceil P \rceil$, and $\lceil S(P) \rceil$ of $\operatorname{sh} P$. In each of these subcases we could write out an argument establishing the induction step, as we have done in previous proofs. However, this would make the proof unacceptably long. Therefore we shall first formulate a general framework for the induction step, in which we shall have to assume certain conditions to hold in order to succeed. Then, in a case-by-case fashion we shall establish the validity of these assumptions. This leads to a considerably shorter proof; the price we pay is the introduction of a large number of variables, many of which will have identical values in each individual case.

To facilitate the proof we give names to the square-calculating parts of the procedures E and E^t, excluding the terminating cases. So if p, q are corners of λ not both in the first row, we define

$$e(\lambda, p, q) = \textbf{if } p \neq q \textbf{ then } q \textbf{ else let } (r, c) = p \textbf{ in } (r - 1, \lambda_{r-1}) \textbf{ fi}$$

so that under the same condition we may write instead of (11):

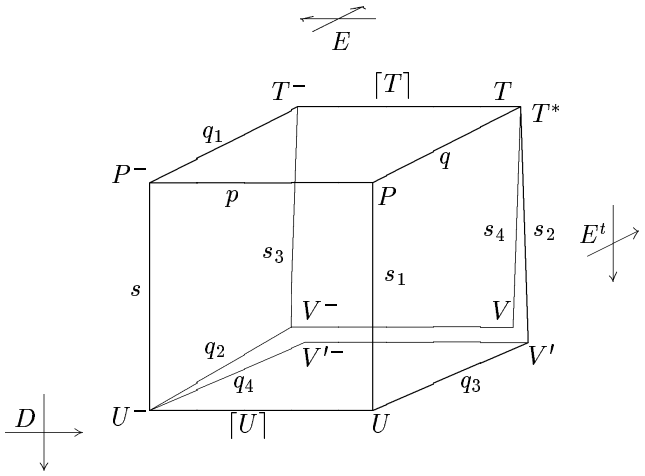
$$E(P, q) = \textbf{let } p = \lceil P \rceil;\ (T', m) = E(P^-, e(\operatorname{sh} P, p, q)) \textbf{ in } \big(T' : (e(\operatorname{sh} P, q, p), P_p), m\big).$$

Also define e^t like e, but replacing $(r - 1, \lambda_{r-1})$ by $(\lambda^t_{c-1}, c - 1)$. Another useful convention is to write $E(T, s) \stackrel{\sim}{=} (X, x)$ meaning that $E(T, s) = (X', x)$ for some tableau $X' \sim X$; so that for instance the statement of the lemma may be rendered as $E^t(S(P), q) \stackrel{\sim}{=} (S(T), n + 1 - m)$.

Now we introduce the numerous variables to be used. Each equation defines the variables appearing in its the left hand side.

$$\begin{array}{lll} p = \lceil P \rceil & q_1 = e(\operatorname{sh} P, p, q) & s_2 = e^t(\operatorname{sh} P, q, s_1) \\ s = \lceil S(P^-) \rceil & q_2 = e^t(\operatorname{sh} P^-, s, q_1) & s_3 = \lceil S(T^-) \rceil \\ (U, s_1, 1) = D(P) & q_3 = e^t(\operatorname{sh} P, s_1, q) & (V, s_4, 1) = D(T) \\ & q_4 = e(\operatorname{sh} U, \lceil U \rceil, q_3) & V' = S^{-1}(T^{*-}) \end{array} \tag{29}$$

The following diagram may aid the reader in remembering the definitions of the variables and in understanding the structure of the proof. The vertices of the cube are labeled by tableaux, but these in fact only indicate their shapes (for this reason the labels T and T^* apply to the same vertex), while the labels of the edges indicate the square by which the shapes associated to the vertices at its ends differ. Starting from the edges labeled s, p and q, the squares at the other edges are determined as indicated by the pairs of arrows. If we can match the edges at the back, then we shall be able to deduce that $T^* \sim S(T)$; so, informally speaking, the proof is by "zipping up the cube at its back".



The assumption in the general framework of the proof is that we can show that

$$q_2 = q_4 \qquad \text{and} \qquad s_2 = s_4. \tag{30a, b}$$

The induction hypothesis is that the lemma holds when P is replaced by a smaller tableau; we shall apply it for P^-, U and U^-.

Comparing the definitions of U and V with (21), we find that $S(P)^- = S(U)$ and $S(T)^- = S(V)$, while comparing them with (19) we find $D(P^-) = (U^-, s, 1)$, $S(P^-)^- = S(U^-)$ and also $D(T^-) = (V^-, s_3, 1)$, $S(T^-)^- = S(V^-)$. We have defined q_1 in such a way that $E(P^-, q_1) = (T^-, m)$. Applying the induction hypothesis for P^- we obtain $E^t(S(P^-), q_1) \cong (S(T^-), n-m)$. On one hand this gives us

$$s_3 = e^t(\operatorname{sh} P^-, q_1, s), \tag{31}$$

which shall be useful for verifying (30b); on the other hand it gives us, using the above identities for $S(P^-)^-$ and $S(T^-)^-$, that

$$E^t(S(U^-), q_2) \cong (S(V^-), n-m). \tag{32}$$

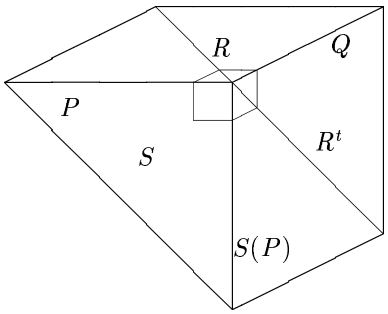
We may proceed similarly starting from $(T^*, m^*) = E^t(S(P), q)$ in place of $(T, m) = E(P, q)$, as follows. Since clearly $\lceil S(P) \rceil = s_1$ we first find that $s_2 = \lceil T^* \rceil$ and $E^t(S(P)^-, q_3) = (T^{*-}, m^*)$. Now using $S(P)^- = S(U)$ and applying the induction hypothesis for U, we get $E(U, q_3) \cong (V', n-m^*+1)$ (the term '$+1$' stems from the fact that the entries of U start at 2). From this it follows by the definition of q_4 that $E(U^-, q_4) \cong (V'^-, n-m^*+1)$, and by application of the induction hypothesis for U^- we get $E^t(S(U^-), q_4) \cong (S(V'^-), m^*-1)$. Comparing this to (32) and using (30a) we conclude that $m^* = n+1-m$ as desired, and that $V'^- \sim V^-$. In order to deduce from the latter statement that $V' \sim V$ it suffices to show that $\operatorname{sh} V' = \operatorname{sh} V$; as these shapes differ from $\operatorname{sh} T = \operatorname{sh} T^*$ by the squares s_2 and s_4 respectively, this follows from (30b). The proof is completed by noting that $\operatorname{sh} T = \operatorname{sh} T^*$ together with $T^{*-} \sim S(V') = S(V) = S(T)^-$ implies $T^* \sim S(T)$.

It remains to verify the assumption (30) in all cases, using the equations in (29) and (31). We consider the possible positions of the three squares s, p, and q that in conjunction with $\lambda = \operatorname{sh} P$ determine the positions of all other squares. The first major distinction is whether or not $s \parallel p$, which by (19) is equivalent to $s_1 = p$.

First assume that $s \not\parallel p$, then s and p are distinct corners of λ. If moreover the corner q of λ is distinct from both s and p, then one immediately sees that $q_i = q$ for $i = 1, \ldots, 4$; also $s_3 = s \not\parallel \lceil T \rceil = p$ so that also $s_i = s$ for $i = 1, \ldots, 4$. If on the other hand $q = p$, then we still have $s = s_1 = s_2$, $\lceil U \rceil = p$ and $q_3 = q$, but now $q_1 = \lceil T \rceil$ is the corner of $\operatorname{sh} P^-$ in the row above p, and q_4 is the corner of $\operatorname{sh} U^-$ in that same row. In case s (which forms the difference between $Y(\operatorname{sh} P^-)$ and $Y(\operatorname{sh} U^-)$) also lies in that row above p, we must have $q_1 = s$ and q_4 lies immediately to the left of it; in the other case $s \neq q_1 = q_4$. In the former case q_2 is defined as the corner of $\operatorname{sh} U^-$ in the column to the left of $s = q_1$, and hence it is equal to q_4, establishing (30a). Since $s_3 = q_2 \parallel q_1 = \lceil T \rceil$ we also have $s_4 = \lceil T \rceil = q_1 = s = s_2$ giving (30b). In the latter case ($s \neq q_1 = q_4$) we have $q_2 = q_1$, and $s_3 = s \not\parallel q_1 = \lceil T \rceil$ (since both sides are corners of $\operatorname{sh} P^-$), so that $s_i = s$ for $i = 1, \ldots, 4$, thus also establishing (30). This completes the case $s \not\parallel p = q$. From this the case $q = s \not\parallel p$ follows by the symmetry principle, or alternatively by reasoning in a similar manner about the positions of the squares.

Now assume that $s \parallel p$, and therefore $s_1 = p$ and $\lceil U \rceil = s$; since s is no corner of λ we also necessarily have $s \neq q$. If in addition $q \neq p$, then it directly follows that $q_i = q$ for $i = 1, \ldots, 4$, as well as $s_3 = s \parallel p = \lceil T \rceil$, so that $s_4 = \lceil T \rceil = p = s_1 = s_2$, and (30) is satisfied. So finally consider the case $s \parallel p = q$. By the symmetry principle we may assume that s and p lie in the same row (or otherwise do the vertical case similarly). Then $q_3 = s_2$ is the corner of $\operatorname{sh} U = \operatorname{sh} P^-$ in the column to the left of p, and therefore equal to $s = \lceil U \rceil$; it follows that q_4 is the corner of $\operatorname{sh} U^-$ in the row above it. Furthermore $q_1 = \lceil T \rceil$ is the corner of $\operatorname{sh} P^-$ in the row above p (which is the same row as the one above s), and since clearly $q_1 \neq s$ we have $q_2 = q_1$, which, being a corner of $\operatorname{sh} U^-$ in the same row, is equal to q_4. The squares $s_3 = s$ and $\lceil T \rceil = q_1$ are both corners of $\operatorname{sh} P^-$, and therefore certainly not adjacent, so that $s_4 = s_3 = s = q_3 = s_2$, and (30) has been proved for this final case. This completes the proofs of 5.2 and of 5.1. □ □

Like for 3.1 and 4.2, there is a pictorial rendering of the proof of this theorem, but it is a three-dimensional instead of a plane structure; its general outline of the picture will be



The three visible faces correspond to the pictorial representations of the computation of $S(P)$, $R(P,Q)$, and $R^t(S(P),Q)$ respectively, as indicated; it is also indicated how the cube displayed above fits into this larger structure. The permutations σ and $\tilde{w}\sigma$ occurring in the proof can be visualised by marking the points given type (a) in our earlier discussion (i.e., the points (i,σ_i) respectively $(i,\tilde{w}\sigma_i)$) on the faces labeled 'R' and 'R^t'; we have suggested this by small circles in the illustration, for the particular permutation $\sigma=(5,6,2,4,1,3)\in\mathbf{S}_6$. Note that these markings exhibit exactly the same pattern on both faces, although otherwise (namely in the Young diagrams ascribed to the points) the two faces are quite different.

As with the earlier theorems, interesting information can be derived here by looking at fixed points. Equation (27) of the theorem implies that if we apply R to a pair of self-dual tableaux, then we obtain a permutation commuting with $\tilde{w}$ (to be precise, we get a permutation of a set of numbers which is symmetric around 0, and $\tilde{w}$ corresponds to multiplication by -1), and conversely any such permutation corresponds to a pair of self-dual tableaux. Now in the symmetric groups $\mathbf{S}_{2n}$ and $\mathbf{S}_{2n+1}$ the centraliser of $\tilde{w}$ is isomorphic to the *hyperoctahedral group* $\mathbf{H}_n$, and we have seen that self-dual tableaux may be conveniently represented by total domino tableaux, whose rank is 0 or 1 according as the self-dual tableau has an even or odd number of squares. So we obtain in this way a pair of Robinson-Schensted correspondences for hyperoctahedral groups, associating to each element of such a group a pair of total domino tableaux of equal shape, and of predetermined rank 0 or 1.

Before we describe this correspondence in more detail, we rephrase 5.1 in several ways as expressing a certain form of commutation between the procedures E, E^t and D (and their inverses). These commutation rules can be derived from the fact that given a sequence of numbers representing a permutation σ, the operations of removing the first, last, least and largest entries all commute with each other, except when these entries coincide (e.g., the operations of removing the last entry commutes with removing the least one, unless the last entry is actually also the least). By 5.1, if $(P,Q)=R^{-1}(\sigma)$, then each of these four operations applied to σ changes P (or more precisely the left tableau of the image under R^{-1}) to the tableau T given respectively by $(T,f)=E^t(P,\lceil S(Q)\rceil)$, $(T,l)=E(P,\lceil Q\rceil)$, $T=P^{\downarrow}$ and $T=P^{-}$. The last of these operations is not of interest (we already know how removing the highest entry commutes with E, E^t and D), but the other three cases, together with the observation that Q is independent of P except for its shape, lead to the following corollary (we have used D^{-1} rather than D to facilitate the formulation).

5.3. Corollary. *Let P be a non-empty tableau of shape λ.*

*(a) (Commutation of E and E^t) Assume $\lambda\neq(1)$; let x be a corner of λ, and let y be a corner of the corresponding partition $\lambda'\in\lambda^-$. Define squares x',y' by $(x',y')=$ **if** $y\parallel x$ **then** (y,x) **else** (x,y) **fi**. Then the following two expressions have the same value:*

$$\textbf{let } (P_1,m_1)=E(P,x);\ (P_3,m_2)=E^t(P_1,y) \textbf{ in } (m_1,m_2,P_3),\text{ and}$$
$$\textbf{let } (P_2,m_2)=E^t(P,y');\ (P_3,m_1)=E(P_2,x') \textbf{ in } (m_1,m_2,P_3).$$

(b) (Commutation of D^{-1} and E) Let x be a cocorner of λ, and let y be a corner of the corresponding partition $\lambda'\in\lambda^+$; let m_1 be a number less than any entry of P. If $x=y$ and this square lies in the first column, then $E(D^{-1}(P,x,m_1),y)=(P,m_1)$. Otherwise define squares $x'=e^t(\lambda',y,x)$ and $y'=e^t(\lambda',x,y)$; the following two expressions have the same value:

$$\textbf{let } P_1=D^{-1}(P,x,m_1);\ (P_3,m_2)=E(P_1,y) \textbf{ in } (m_2,P_3),\text{ and}$$
$$\textbf{let } (P_2,m_2)=E(P,y');\ P_3=D^{-1}(P_2,x',m_1) \textbf{ in } (m_2,P_3).$$

(c) *(Commutation of D^{-1} and E^t) Let x be a cocorner of λ, and let y be a corner of the corresponding partition $\lambda' \in \lambda^+$; let m_1 be a number less than any entry of P. If $x = y$ and this square lies in the first row, then $E^t(D^{-1}(P,x,m_1),y) = (P,m_1)$. Otherwise define squares $x' = e(\lambda',y,x)$ and $y' = e(\lambda',x,y)$; the following two expressions have the same value:*

$$\textbf{let } P_1 = D^{-1}(P,x,m_1);\ (P_3,m_2) = E^t(P_1,y) \textbf{ in } (m_2,P_3), \text{ and}$$
$$\textbf{let } (P_2,m_2) = E^t(P,y');\ P_3 = D^{-1}(P_2,x',m_1) \textbf{ in } (m_2,P_3).$$

Each of these statements is in a sense equivalent to 5.1: their direct proof would involve the same cases, and they would readily imply the full theorem (e.g., 5.3(b) could be used to prove lemma 5.2); in the prism drawn above the statements concern the sequences of cubes along the edges labeled Q, P and $S(P)$ respectively.

We now consider the case of self-dual tableaux and the hyperoctahedral group $\mathbf{H}_n$. Elements of $\mathbf{H}_n$ are represented by sequences of numbers that are invariant under reversal of the sequence while simultaneously negating each individual term. Such sequences can be built up starting with either an empty sequence or a sequence consisting of a single 0, by repeatedly extending it at both ends by opposite numbers. Using 5.1 we see that applying R^{-1} to any of the intermediate sequences will yield a pair of self-dual tableaux (more precisely, the left tableau is self-dual, while the right tableau is similar to a self-dual tableau). Also according to 5.1 we see that if (P,Q) is one such pair, then the previous pair (i.e., for the sequence of length 2 less) can be found as (P_3,Q') after putting $x = \lceil Q \rceil$, $(Q',y,i) = D(Q^-)$ and performing the computation of 5.3(a). Because Q is similar to a self-dual tableau, we have $y \parallel x$, but x and y are unrelated to P. Therefore we apparently have in 5.3(a) that if P is self-dual and $y \parallel x$ then P_3 is also self-dual, and $m_2 = -m_1$. We shall now investigate this transition in terms of domino tableaux.

The function we are interested in, which we call F, and which is the counterpart of E (and E^t) for domino tableaux, can be written as follows. We are given a total domino tableau P and a domino $\{y,x\} \subseteq Y(\mathrm{sh}\,P)$ such that $Y(\mathrm{sh}\,P) \setminus \{x,y\}$ is again Young diagram; we wish to compute

$$F(P,x,y) = \textbf{let } (P',m) = E(A^{-1}(P),x);\ (P'',m') = E^t(P',y) \textbf{ in } (A(P''), m\ [=-m']) \qquad (33)$$

but without reference to the conversion procedures A and A^{-1}. By 5.3(a) we may interchange the applications of E and E^t if we like (also interchanging m and m', but keeping everything else in place); in practice it is convenient to use the stated order if y and x are *horizontally* adjacent, but to apply E^t first if they are *vertically* adjacent. The task of rewriting (33) may seem a formidable one, as it involves a cascade of no less than 4 recursive procedures, but thanks to the commutation relations of 5.3 everything simplifies quite nicely.

Clearly we are after a recursive definition of F, in the style of E and E^t. Let $t = \lceil P \rceil$ and $s = \lceil P^- \rceil$, so $\{s,t\}$ is highest domino of P, and let $n > 0$ be its entry; we should analyse the rôle of this domino, depending on its position relative to the domino $\{y,x\}$. Consider first the case that these two dominoes are disjoint; we may hope that this case will be the simplest one. Now by definition of A^{-1} we have $\lceil A^{-1}(P) \rceil = t$, and by the disjointness assumption its entry will not move under the applications of E and E^t, and so, by the definition of A, we may conclude that $\lceil F(P,x,y) \rceil = t$. To determine the other square of the highest domino of $F(P,x,y)$ is less straightforward: first we must compute $D^{-1}(A(P^{--}),s,-n)$ then apply E and E^t, and finally apply D again. However, the commutation rules 5.3(b,c) guarantee that the result of this final application will be of the form $(T,s,-n)$, so that the highest domino of $F(P,x,y)$ is identical to that of P; this is similar to what we know for our single-square procedures. Moreover, T is such that $A(T)$ is the tableau component of $F(P^{--},x,y)$, so that in this case we find $F(P,x,y) = \textbf{let } (P',m) = F(P^{--},x,y) \textbf{ in } (P' : (s,n) : (t,n), m)$.

Although the other cases are more involved then the disjoint one, they can nevertheless be treated by variations of its computation: some of the trivial cases in (11) and 5.3(b,c) will be replaced by less trivial ones. Let us consider first the case that the application of E terminates directly (without recursion); this happens when $t = x$ lies in the first row, in which case $s = y$ is horizontally adjacent to it. We then have $E(A^{-1}(P),x) = (P',n)$ where $P' = D^{-1}(A^{-1}(P^{--}),s,-n)$; by the first case of 5.3(c)

$E^t(P',y) = (A^{-1}(P^{--}),-n)$, whence we get $F(P,x,y) = (P^{--},n)$ for this case. The case that $t = x$ lies in the first column and $s = y$ is vertically adjacent to it is similar, except that now the application of E^t—which, as we said above, it is convenient to perform first—extracts the number n, while the application of E extracts $-n$; therefore we now have $F(P,x,y) = (P^{--},-n)$.

Next consider the other cases where $\{s,t\} = \{y,x\}$; assume first that this domino is a horizontal one. Since x coincides with $t = \lceil A^{-1}(P)\rceil$, the entry of that square will move one row up to $\hat{x} = e(\lambda,x,x)$ when E is applied. In fact that application $E(A^{-1}(P),x)$ expands to

$$\textbf{let } P_1 = D^{-1}(A(P^{--}),s,-n);\ (P_3,m) = E(P_1),\hat{x}) \textbf{ in } (P_3 : (\hat{x},n),m)$$

which, by 5.3(b) and $\hat{x} \neq s$ can also be written as

$$\textbf{let } (P_2,m) = E(A(P^{--}),\hat{x});\ P_3 = D^{-1}(P_2),s,-n) \textbf{ in } (P_3 : (\hat{x},n),m)$$

We temporarily retain these values of P_2 and P_3; we have to compute $E^t(P_3,y)$ next. For this we can apply 5.3(c): since the squares s and y coincide we have to determine $e(\mathrm{sh}\,P_3,y,y)$, which is the square to the left of $\hat{x}$; we call this square $\hat{y}$. The result is that we can determine m' and P'' in (33) by putting $(T,m') = E^t(P_2,\hat{y})$ and then $P'' = D^{-1}(T,\hat{y},-n)$. Since $A(P^{--})$ is self-dual we find that $m' = -m$, as it should; moreover it follows that $(A(T),m) = F(P^{--},\hat{x},\hat{y})$. Putting everything together we find that in this case $F(P,x,y) = \textbf{let } (T',m) = F(P^{--},\hat{x},\hat{y}) \textbf{ in } (T' : (\hat{y},n) : (\hat{x},n),m)$. The case that $\{s,t\} = \{y,x\}$ is a vertical domino is just the transpose of the horizontal case: now $\{\hat{y},\hat{x}\}$ is a vertical domino at the end of the column to the left of $\{y,x\}$.

A final case occurs when the intersection of $\{s,t\}$ with $\{y,x\}$ is a single square; this necessarily is the corner $t = x$. We may assume that $\{y,x\}$ is horizontal and $\{s,t\}$ vertical, because again the other case will be its transpose. This case starts off in the same way as the previous one, with the entry of square x moving up to the previous row: in the current case this is to square s. Therefore $E(A^{-1}(P),x)$ expands in the same way as above, but with s in the place of $\hat{x}$. Again we can apply 5.3(b), but now we are in the case of equal squares, with $e(\mathrm{sh}\,P^{--},s,s) = y$, so that we obtain for $E(A^{-1}(P),x)$:

$$\textbf{let } (P_2,m) = E(A(P^{--}),y);\ P_3 = D^{-1}(P_2),y,-n) \textbf{ in } (P_3 : (s,n),m)$$

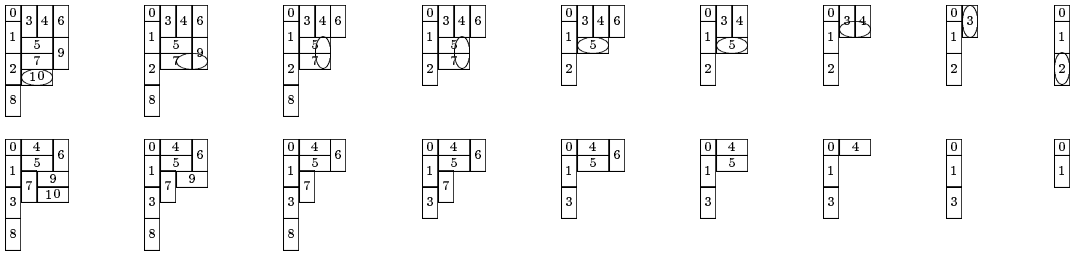
The computation of $E^t(P_3,y)$ proceeds exactly as above: the case of equal squares (namely y) in 5.3(c) is applied, and $e(\mathrm{sh}\,P_3,y,y)$ is the square above y (and to the left of s), which we call z. We find $(T,m') = E^t(P_2,z)$ and $P'' = D^{-1}(T,z,-n)$, where $m' = -m$ and $(A(T),m) = F(P^{--},y,z)$. As final result we get $F(P,x,y) = \textbf{let } (T',m) = F(P^{--},y,z) \textbf{ in } (T' : (z,n) : (s,n),m)$. Note that in this case the domino to be cleared (i.e., $\{y,x\}$) and the domino containing n (i.e., $\{s,t\}$) flip from horizontal to vertical and vice versa in the recursion (namely to $\{z,y\}$ respectively to $\{z,s\}$). Note also that we arrived at the recursive call $F(P^{--},y,z)$ with E being applied before E^t, but since $\{y,z\}$ is vertical, we shall choose to interchange the order of these applications when evaluating $F(P^{--},y,z)$ further.

Summarising these considerations, we have the following recursive description of F:

$$\begin{array}{l}
F(P,x,y) = \textbf{let } t = \lceil P\rceil;\ s = \lceil P^{-}\rceil;\ n = P_t \textbf{ in} \\
\quad \textbf{if } t \neq x \textbf{ then let } (T,m) = F(P^{--},x,y) \textbf{ in } (T : (s,n) : (t,n),m) \\
\quad \textbf{else let } (r,c) = x \textbf{ in} \\
\qquad \textbf{if } s \neq y \\
\qquad \textbf{then let } z = (r-1,c-1);\ (T,m) = F(P^{--},y,z) \textbf{ in } (T : (z,n) : (t,n),m) \\
\qquad \textbf{else if } r = 1 \vee c = 1 \textbf{ then } (P^{--},\ \textbf{if } r = 1 \textbf{ then } n \textbf{ else } -n \textbf{ fi}) \\
\qquad\quad \textbf{else let } (r',c') = y;\ \lambda = \mathrm{sh}\,P^{--};\ (\hat{x},\hat{y}) = \\
\qquad\qquad\qquad \textbf{if } r' = r \textbf{ then } ((r-1,\lambda_{r-1}),(r-1,\lambda_{r-1}-1)) \\
\qquad\qquad\qquad\qquad \textbf{else } \ ((\lambda^t_{c-1},c-1),(\lambda^t_{c-1}-1,c-1)) \\
\qquad\qquad\qquad \textbf{fi} \\
\qquad\qquad ;\ (T,m) = F(P^{--},\hat{x},\hat{y}) \\
\qquad\qquad \textbf{in } (T : (\hat{y},n) : (\hat{x},n),m) \\
\qquad\quad \textbf{fi} \\
\qquad \textbf{fi} \\
\quad \textbf{fi}
\end{array} \tag{34}$$

Despite the complications, the similarity to (11) is quite clear.

We illustrate this procedure by an example which involves all the cases. The first row displays the domino tableaux P for which an application $F(P,x,y)$ takes place, with the domino $\{y,x\}$ indicated by an oval; this row is to be read from left to right. The second row displays beneath each tableau the result of the corresponding application of F; this row is to be read from right to left. The final result of this procedure is $(T,-2)$, where T is the first tableau of the second row.



With the auxiliary procedure F it is not difficult to define the full Robinson-Schensted algorithm for the hyperoctahedral group, which is quite analogous to (9). Although elements of $\mathbf{H}_n$ correspond to permutations of $\{-n,\ldots,n\}$, they are fully determined by the images of the positive numbers only, because we have $\sigma_{-i} = -\sigma_i$ for all i. Since these values are given in the second half of the sequence representing the permutation, we content ourselves with computing only that half of the sequence.

$$G(P,Q) = \textbf{if } |\operatorname{sh} Q| \leq 1 \textbf{ then } () \textbf{ else let } (T,m) = F(P,\lceil Q\rceil,\lceil Q^{-}\rceil) \textbf{ in } G(T,Q^{--}) \,\&\, m \textbf{ fi} \qquad (35)$$

Similarly to E and R, the algorithms F and G can be inverted in an obvious way; in the case of G we must of course specify the desired rank for the domino tableaux, which determines which core to use at the bottom of the recursion. From the remarks made earlier, and from the fact that F has been defined in a way so as to satisfy (33), it is not difficult to see that G computes the Robinson-Schensted correspondence for pairs of self-dual tableaux represented as domino tableaux. This fact can be formally stated as

5.4. Theorem. *For any pair of normalised total domino tableaux P,Q of the same shape, $G(P,Q)$ represents the element of the hyperoctahedral group $\mathbf{H}_n$ which under the standard embedding of $\mathbf{H}_n$ in $\mathbf{S}_{2n}$ or $\mathbf{S}_{2n+1}$ corresponds to $R(A^{-1}(P),A^{-1}(Q))$.* □

Remarks. The idea of defining a Robinson-Schensted algorithm for $\mathbf{H}_n$, using a class of special Young tableaux, is first mentioned by Barbasch and Vogan in [BaVog], p.171. They define a procedure equivalent to A, which is used mainly as a criterion to recognise self-dual tableaux. They do not prove their claim that this class of tableaux corresponds via R to $\mathbf{H}_n$ as subgroup of $\mathbf{S}_{2n}$ or $\mathbf{S}_{2n+1}$, however; their "proof" is merely a description of a method of computing $R(P,Q)$ for self-dual P,Q, alternatingly using E and E^t to produce numbers at either end of the sequence. The algorithms F and G were introduced by Devra Garfinkle in [Garf] (where they are called β (definition 1.2.9) and B (definition 1.2.11), with inverses α (definition 1.2.5) and A' (definition 1.2.7) respectively), by means of constructions that are essentially the same as (34) and (35), although the formalism used is rather different. She mentions in the introduction that the correspondence obtained between pairs of domino tableaux and elements of $\mathbf{H}_n$ is the same as that of Barbasch and Vogan (as stated in our theorem 5.4), but the paper contains no proof of such a statement; nonetheless it seems reasonable to assume that the algorithm was derived along much the same lines as our reasoning above. It should be noted that neither of the papers mentioned here concerns itself primarily with combinatorial questions; indeed the algorithm found is used only as a first step towards generalising an interpretation of he Robinson-Schensted algorithm, concerning so-called primitive ideals in the enveloping algebra of simple Lie groups of type A_n, to the other classical types B_n, C_n and D_n (these matters can also be formulated in terms of the cells in the corresponding Weyl groups, as defined for instance by means of the Kazhdan-Lusztig polynomials). In this context it is also appropriate to mention

the current author's thesis [vLee1], in which the geometric interpretation of the Robinson-Schensted algorithm which is discussed in Part II of this paper is generalised from type A_n to the other classical types. In that instance too the algorithm G is not the whole story: in fact the generalisation of the Robinson-Schensted algorithm for this problem is very similar, but not quite identical to G (it appears that despite the differences between these two algorithms, their results never diverge very far from each other, but as yet we have not been able to formulate their exact relationship).

There exists a family of algorithmically defined bijections between pairs of equal shaped tableaux consisting of a fixed q-core to which n rim hooks of size q are added, and "q-coloured permutations of n" (i.e., sequences representing a permutation of n, in which each term has al additional attribute (colour) $\in \mathbf{Z}/q$); it was first defined in [StWhi], and a succinct formulation appears in [FomSt]. The case $q = 2$ gives a bijection between domino tableaux and 2-coloured permutations of n, which are naturally identified with elements of $\mathbf{H}_n$; we would like to stress however that this is *not* the same bijection as defined by G above. In fact, using the bijection sketched in the previous section between domino tableaux with n dominoes and pairs of Young tableaux with a total of n squares, the 2-rim hook algorithm translates into a pair of ordinary Robinson-Schensted algorithms running together, each one with its own colour, and with each entry being inserted into the tableau of the same colour* (a similiar statement holds for arbitrary q). No similar factorisation holds for G, as we shall see below. (In the language of differential posets, the set $\mathcal{P}$ equipped with the partial order induced by removal of dominoes is a 2-differential poset, which means that in (1) the number 1 is replaced by 2, and it factors into disjoint components according to the rank associated to the partitions. The two different algorithms correspond to different bijections realising the modified equation (1), with the bijection corresponding to the 2-rim hook algorithm being derived from the bijection of each connected component of $\mathcal{P}$ as 2-differential poset with $\mathcal{P} \times \mathcal{P}$, where both factors have the ordinary inclusion ordering of the Young lattice.) It would appear that, from the enumerative combinatorial point of view, the rim hook algorithms are the preferred ones, since thay are defined for all q; however, when the symmetric group is viewed as Weyl group of type A_n, then G is the more natural generalisation to the other classical types.

As remarked earlier, definition (35) in fact defines two correspondences, namely for domino tableaux of rank 0 and of rank 1. To see that there is no difficulty with a non-empty core, it suffices to observe that in any proper application $F(P, x, y)$ (i.e., where $\{x, y\}$ can be removed from $Y(\mathrm{sh}\, P)$ leaving a Young diagram) the domino $\{x, y\}$ does not meet the core. This holds in particular for all recursive applications of F, so that the computation of $F(P, x, y)$ can be completed without problems. Hence the successive applications of F during the recursion of G will reduce the shape of the tableau, until finally only the core remains. This observation can be extended without modification to the case that P is a domino tableau of arbitrary rank (this fact was our main motivation to introduce such domino tableaux). This means we can define $G(P, Q)$ for arbitrary pairs of domino tableaux P, Q of equal shape, provided only that the termination criterion becomes that Q is reduced to its core. Since no domino can be removed from a staircase shape leaving a Young diagram, it will be clear that P is reduced to its core at the same step that Q is, whence P and Q must have had the same rank to begin with; this gives yet another proof of that the rank of a domino tableau is determined by its shape. We thus obtain a countable family, parametrised by a natural number r, of bijections between the set of equal shaped pairs of domino tableaux of rank r with n dominoes and the group $\mathbf{H}_n$.

The definition of these bijections is independent the rank r, but this does not imply that the bijections are all equivalent in some sense. For instance, the equivalence classes induced in $\mathbf{H}_n$ of elements corresponding to pairs of tableaux of one same shape are *not* the same for each rank, as can already be seen for $n = 2$ and $r = 0, 1$ (yet, by our earlier counting argument, the number of such classes is independent of r). Contrarily, due to the factoring into separate ordinary Robinson-Schensted algorithms, the similiar classes defined by the rim hook algorithms are all the same; this also shows that a similar factoring cannot hold for G. The existence of more than one non-equivalent version of G may serve as an explanation for the fact that apparently it does not give a complete answer to those problems from out-

* Incidentally, the description of such an algorithm for $\mathbf{H}_n$ can already be found in [Stan1], §6, it is inspired mainly by enumerative considerations. See also [Oka1].

side combinatorics where a generalisation of the Robinson-Schensted correspondence to hyperoctahedral groups has been sought for.

It is interesting to consider the limit $r \to \infty$. One easily sees that if $r + 1 \geq n$, then the dominoes will be separated into a group of horizontal dominoes at the right end of the core and a group of vertical dominoes at the bottom of the core, and further increase of r will not essentially change anything. Moreover, shifting the lines in the former group of dominoes so as to line up the leftmost dominoes of each row, respectively shifting the columns of the latter group to line up the topmost dominoes, we obtain arrangements that are essentially stretched versions of ordinary Young tableaux. This would suggest that this limiting case might in fact be the same as the 2-rim hook correspondence; this is not quite the case, but the difference is limited to a transposition of the Young tableau at the bottom of the core. Therefore the algorithms G may be viewed as deformations of a variant of the 2-rim hook algorithm, which is easier to understand due to factoring; it would therefore be interesting to study in detail the effect of increasing the rank on the correspondence defined by G.

Considerations as these raise numerous interesting questions about G, but we shall not pursue these in this paper; there are however still a few obvious properties to be noted. First, G exhibits the same symmetry with respect to its arguments as expressed by 3.1 for R: for each pair P, Q of domino tableaux of equal shape, $G(Q, P)$ represents the element of $\mathbf{H}_n$ which is inverse to $G(P, Q)$. Second, the transpose of G is identical to it, up to a change of sign in the sequence of numbers computed: when $G(P, Q) = \sigma$ then $G(P^t, Q^t) = -\sigma$, where the sequence $-\sigma$ is obtained by termwise negation of the terms of σ (which corresponds to multiplication by the central element $\tilde{w}$ of $\mathbf{H}_n$). For ranks 0 and 1 these facts can be deduced immediately via 5.4 from 3.1 and 5.1 respectively; alternatively, a direct proof can be given, which works for all ranks. For the latter statement the direct proof is almost trivial, for the former statement the proof is similar to that of 3.1: the global structure of the proof will be identical, and although the basic step will involve a larger number of cases, a moment's reflection about (34) will show that it has the symmetry required to make the induction step succeed. As a consequence we obtain for each fixed rank a bijection between the set of involutions in $\mathbf{H}_n$ and the set of domino tableaux of that rank with n dominoes. Like for the proper Robinson-Schensted algorithm, one may define a slightly more efficient version of G (and its inverse) for this special case; the remarks about Δ in §3 can be adapted without difficulty. From this one may derive the following analogue of 3.2:

5.5. Proposition. *Let P be a domino tableau, let k be the difference between the number of odd-length columns of P and that of the core of P, and let l be the similar difference for the odd-length rows. Then k and l are non-negative even numbers, and for $\sigma = G(P, P)$ the number of (positive) i with $\sigma_i = i$ equals $\frac{k}{2}$, while the number of (positive) i with $\sigma_i = -i$ equals $\frac{l}{2}$.* □

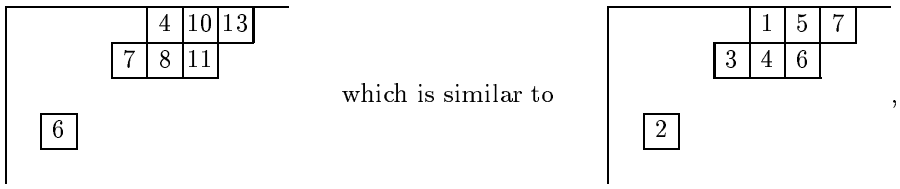
For total domino tableaux this proposition can also be deduced via 5.4 from 3.2 itself.

Via the indirect way of involutions in $\mathbf{H}_n$, we obtain bijections between the sets of domino tableaux of all fixed ranks. These bijections do not always preserve equality of shape between tableaux, whence they cannot commute with G for pairs of different tableaux, nor do they coincide with the bijections defined by passage to pairs of Young tableaux. The interesting question of giving a direct description of these bijections is likely to require techniques different from those in this paper, since they do not commute with removal of the highest domino: while $\begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline \end{array}$ corresponds to $\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 2 \\ \cline{1-1} \end{array}$, we find that $\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 \\ \cline{1-1} \end{array}$ corresponds to $\begin{array}{|c|c|}\hline 0 & 1 \\ \hline 2 & 3 \\ \hline \end{array}$. It seems that techniques similar to those in [Garf], §5 can be useful for answering this question and related ones.

§6. Glissement.

In this section we shall discuss some remarkable consequences of 5.1. Basically, we use this theorem to show that certain constructions with a non-deterministic element will still produce uniquely defined results. Our main goal is to show how the approach to the Robinson-Schensted and Schützenberger algorithms developed in [Schü2], which is apparently completely different from ours, can be understood in this way. It is unclear whether historically that approach has emerged from a similar line of reasoning, although the equivalent of theorem 5.1 had been known for quite some time. The style of this section will be somewhat less formal than the previous ones, in order to allow a more intuitive discussion.

In this section we shall extend the class of tableaux under consideration from Young tableaux to so-called skew tableaux. A *skew diagram* is a finite subset of $\mathbf{N}_{>0} \times \mathbf{N}_{>0}$ that is convex with respect to the natural ordering, i.e., for a skew diagram χ and any pair of squares $s, t \in \chi$ we have $p \in \chi$ whenever $s \leq p \leq t$. A skew diagram is a Young diagram if and only if it is either empty or contains the square $(1,1)$, and every skew diagram χ can be expressed as the difference set $Y(\lambda) \setminus Y(\mu)$ for some $\lambda, \mu \in \mathcal{P}$ with $\mu \subseteq \lambda$; in general λ and μ are not uniquely determined by χ. A skew diagram is partially ordered by the natural ordering on $\mathbf{N}_{>0} \times \mathbf{N}_{>0}$; a *skew tableau* K of shape χ is an injective morphism of partially ordered sets $\chi \to \mathbf{Z}$. We define $\operatorname{sh} K = \chi$ in this case, and define a similarity relation $K \sim K'$ for skew tableaux K, K' that holds whenever $K' = f \circ K$ for some monotonic function $\mathbf{Z} \to \mathbf{Z}$. Like Young tableaux, a skew tableau can be depicted by filling each square with its image in $\mathbf{Z}$, which may look something like



where we have drawn lines to indicate the limits of $\mathbf{N}_{>0} \times \mathbf{N}_{>0}$. As with Young tableaux, rows and columns are increasing, and similar skew tableaux are obtained from one another by renumbering the entries in an order preserving way.

One obvious way to obtain a skew tableau is to start with a Young tableau T, and to retain only those squares whose entries lie in a specified interval J of $\mathbf{Z}$; we shall call this the restriction to J of T, written $T|_J$. Every non-empty skew tableau K can be obtained as the restriction of some Young tableau T to an interval containing the highest entry of T; we call such a tableau T an *extension* of K to a Young tableau. By removing from T all the squares of K we obtain another Young tableau which we shall denote as $T \setminus K$. Our main interest shall be in a relation between skew tableaux called *'glissement'* in [Schü2]; we shall not translate the term.

6.1. Definition *Let K and K' be a skew tableaux with the same set of entries; K is related by glissement to K', written as $K \triangleright K'$, if there exists an extension T of K to a Young tableau, say $K = T|_J$ such that $K' = T^{\downarrow\cdots\downarrow}|_J$ for some number of applications of $\cdot^\downarrow$ (this number cannot exceed $|\operatorname{sh}(T \setminus K)|$, lest entries of K would be lost); T is called a witness for the relation $K \triangleright K'$.*

We deduce from the definition of D that $T^\downarrow|_J$ only depends on K and the square s defined by $(T', s, m) = D(T \setminus K)$. This square s is such that $\{s\} \cup \operatorname{sh} K$ is again a skew diagram, and s is a minimal point of it. We have $T^\downarrow|_J \neq K$ if and only if $s \parallel t$ for some $t \in \operatorname{sh} K$, and in this case we say that K and $T^\downarrow|_J$ are related by one-step glissement into square s, written $K \triangleright_s T^\downarrow|_J$. It is obvious that whenever $K \triangleright K'$, we can find a sequence $K \triangleright_{s_1} \cdots \triangleright_{s_k} K'$. The converse is also true:

6.2. Lemma. *The relation of glissement is transitive.*

Proof. Since any glissement can be broken up into one-step glissements, it is sufficient to prove that $K \triangleright_s K' \triangleright K''$ implies $K \triangleright K''$. There exist a witness T for $K' \triangleright K''$, and J such that $K' = T|_J$ and $K'' = T^{\downarrow\cdots\downarrow}|_J$; if we can find a Young tableau P with $K = P|_J$ and $P^\downarrow = T$, it will witness $K \triangleright K''$. Let p be the square such that $\{s\} \cup \operatorname{sh} K = \operatorname{sh} K' \cup \{p\}$. If p is a cocorner of $\operatorname{sh} T$, then we can

take $P = D^{-1}(T, p, m)$ (for sufficiently small m) which clearly satisfies the requirements. However, p is not necessarily a cocorner of $\mathrm{sh}\, T$ (counterexamples with $\#\,\mathrm{sh}\, K = 1$ are easily found), the possible obstruction being the existence of squares $t < p$ with $t \notin Y(\mathrm{sh}\, T)$. In fact, such a t excludes the existence of any appropriate P for this T, as we would necessarily have $t \in Y(\mathrm{sh}\, P)$; therefore, the only solution is to replace T by another witness for $K' \triangleright K''$. Because $\{s\} \cup \mathrm{sh}\, K = \mathrm{sh}\, K' \cup \{p\}$ is a skew diagram we cannot have $k < t$ for any square $k \in \mathrm{sh}\, K'$, and hence if t is chosen to be minimal among the obstructing squares (and therefore a cocorner of $\mathrm{sh}\, T$) and m is sufficiently small, the tableau $T' = D^{-1}(T, t, m)$ witnesses $K' \triangleright K''$ and has one less obstructing square than T. Replacing T by T', and repeating the process until p has become a cocorner of $\mathrm{sh}\, T$, we eventually construct P witnessing $K \triangleright K''$. □

As a corollary to the proof of the lemma we see that whenever $K \triangleright K'$, we can find an witness P for $K \triangleright K'$ which is of the minimal size possible (given $\mathrm{sh}\, K$), i.e., with $Y(\mathrm{sh}\, P) = \{\, s \mid \exists t \in \mathrm{sh}\, K \colon s \leq t \,\}$; we call such a P a *minimal witness* for $K \triangleright K'$.

A skew tableau K is related by glissement to any skew tableau K' obtained from K by translating all its squares by $(-x, -y)$ for $x, y \in \mathbf{N}$. To see this it suffices to consider the case $x = 0, y = 1$. In this case K has no squares in the first column; in each row that meets $\mathrm{sh}\, K$ consider the square directly before the first square of $\mathrm{sh}\, K$, and let $s_1, \ldots, s_k$ be the sequence of these squares ordered by decreasing row number. Now form the sequence $K = K_0 \triangleright_{s_1} K_1 \triangleright_{s_2} \cdots \triangleright_{s_k} K_k$. One easily sees that each K_i is obtained from K_{i-1} by shifting the row containing s_i one place to the left (the glissements only involve horizontal motion of squares); consequently $K_k = K'$ so that $K \triangleright K'$ as claimed.

A skew diagram χ may be partitioned into *connected components*, which are equivalence classes for the equivalence relation generated by $s \parallel t$. The above argument may be extended to the case that K' is obtained from K by applying possibly different translations to each of the connected components of $\mathrm{sh}\, K$, provided that the resulting sets form the connected components of $\mathrm{sh}\, K'$, and that their relative order (as determined for instance by the column numbers of arbitrarily chosen representative squares in each of the components) is unaltered.

The following direct consequence of 5.1 is crucial in our approach to glissements. It also justifies the claim at the end of §4 concerning "almost self-dual tableaux with infinitessimal entries".

6.3. Lemma. *Let Q, Q' be two Young tableaux of equal shape, which coincide with each other in the position of each of their k highest entries, for some $k \in \mathbf{N}$. Then $S(Q)$ and $S(Q')$ coincide with each other in the position of each of their their k lowest entries $1, \ldots, k$.*

Proof. Let $\lambda = \mathrm{sh}\, Q = \mathrm{sh}\, Q'$, and choose any fixed tableau $P \in \mathcal{T}_\lambda$. Now compute the permutations $\sigma = R(P, Q)$ and $\sigma' = R(P, Q')$. From the definition of R it follows that the last k terms in the sequence of σ coincide with those of σ', since the first k extractions proceed identically in both computations. By (27), we have $R(S(P), S(Q)) = \tilde{w}\sigma\tilde{w}$ and $R(S(P), S(Q')) = \tilde{w}\sigma'\tilde{w}$, and since conjugation by $\tilde{w}$ means reversing the sequence while applying the permutation $\tilde{w}$ to the individual entries, we find that $\tilde{w}\sigma\tilde{w}$ and $\tilde{w}\sigma'\tilde{w}$ coincide in their *first* k terms. But since $S(Q)$ and $S(Q')$ are obtainable as the respective right tableaux after applying R^{-1} to these sequences, and the first k insertions during these computations are identical, we find that $S(Q)$ and $S(Q')$ coincide in their k lowest entries. □

Remark. We have used the weakest identity (27) of 5.1, but we could equally well have used (26), in combination with R^{-t}, instead (in fact this would be somewhat more natural). Alternatively, the reader may check that a similar proof can be based on the observation that the positions of the k *highest* terms in the sequence of $R(Q, P)$ and $R(Q', P)$ are identical, and then using (25) or (27).

6.4. Theorem. *Each skew tableau is related by glissement to a unique Young tableau.*

Proof. Then existence of such a Young tableau is obvious. Now let K be a skew tableau, and suppose T and T' are both Young tableaux such that $K \triangleright T$ and also $K \triangleright T'$. Choose minimal witnesses Q and Q' for these relations, and let $k = \#\,\mathrm{sh}\, K$. We have $\mathrm{sh}\, Q = \mathrm{sh}\, Q'$, so that the lemma applies, giving $S(Q)|_J = S(Q')|_J$, where $J = \{1, \ldots, k\}$. From the definitions of S and of glissement it follows that $S(Q)|_J = S(T)$ and similarly $S(Q')|_J = S(T')$. From the invertibility of S on normalised tableaux it follows that $T \sim T'$, and since both T and T' have the same set of entries as K, we get $T = T'$. □

Remark. Using the language of term rewrite systems, the relation $\triangleright$ is a reduction relation on the set of skew tableaux, which is strongly normalising, with Young tableaux as normal forms. The theorem is equivalent to the statement that this rewrite system is confluent (or equivalently it has the Church-Rosser property), i.e., if $K \triangleright L_1$ and $K \triangleright L_2$ then there exists an M with $L_1 \triangleright M$ and $L_2 \triangleright M$. A slightly unusual aspect is that the proof depends on going all the way down to the normal form. In particular, it is *not* true that $K \triangleright K'$, $K \triangleright K''$ and $\operatorname{sh} K' = \operatorname{sh} K''$ imply $K' = K''$.

As a consequence of the theorem, the assumption of equal shapes may be dropped in 6.3. The various ways indicated to prove that lemma lead to various ways of determining the tableau T of the theorem, or equivalently $S(T)$, without using any glissement at all. For instance, according to the proof using (25), one may extend the skew tableau K to a partially indeterminate Young tableau P without actually choosing any entries, by assigning to each new square a value "unknown, but smaller than any entry of K". One may verify that for arbitrary Q the computation of $R(P,Q)$ can still be simulated, resulting in a sequence containing the entries of K at well defined positions, interspersed with unknown but smaller terms. (This process is very closely related to the construction in [SaSt], Theorem 2.1.) Multiplying to the left by $\tilde{w}$ means applying the anti-monotonic map $\tilde{w}$ to each term; this makes the unknown terms greater than any of the known ones. Since inserting a term by the procedure I^t only affects the positions of the entries greater than it, the positions of the known numbers is not affected by the unknown ones, so by (25) simply leaving out the unknown terms and applying R^{-t} gives us $S(T)$ as desired. As a variation of this process one can use (27), replacing left multiplication by $\tilde{w}$ by conjugation, and R^{-t} by R^{-1}; since R builds its sequence from right to left while R^{-1} uses its sequence from left to right, the extra reversal of the intermediate sequence needed in this case means that as soon a an entry x is extracted from K the corresponding number $\tilde{w}_x$ can be inserted into $S(T)$.

Both these computations allow a tableau Q controlling the extraction process to be freely chosen, without affecting the final result. The reader may check that by an appropriate choice of Q one can achieve that the entries of K are extracted by columns, going down each column from top to bottom and proceeding from right to left through the columns; for another choice of Q the entries are extracted by rows, going through each row from right to left and ordering the rows from top to bottom. (For the first case fill $\operatorname{sh} Q$ columnwise, for the second put the highest entries of Q at the bottoms of the columns, proceeding from right to left, then continue similarly with the remaining columns that are not yet filled up, etc. In both cases show that each extraction step in the computation of $R(P,Q)$ involves only a single column.) For instance, for the first skew tableau depicted above the entries may be extracted in the order $13,10,11,4,8,7,6$ or alternatively $13,10,4,11,8,7,6$, either of which leads us (after renumbering anti-monotonically, viz. substituting $1,2,3,\ldots$ for $13,11,10,\ldots$, and then applying R^{-1}) to conclude that

$$S(T) = \begin{array}{|c|c|c|c|c|}\hline 1 & 2 & 4 & 5 & 6 \\ \hline 3 & 7 \\ \cline{1-2}\end{array} \qquad \text{which implies that} \qquad T = \begin{array}{|c|c|c|c|c|}\hline 4 & 7 & 8 & 10 & 13 \\ \hline 6 & 11 \\ \cline{1-2}\end{array}.$$

One can apply 5.1 once more to find two ways of finding T without first computing $S(T)$. We get

6.5. Proposition. *Let K be a skew tableau, and let σ be the sequence of its entries, listed either by going up the columns, taking the columns in order from left to right, or by going from left to right along the rows, ordering the rows from bottom to top. Then the unique Young tableau P with $K \triangleright P$ of Theorem 6.4 can be found as the left tableau of the pair $R^{-1}(\sigma)$. Variations of this computation can be obtained using 5.1, in particular, to compute $S(P)$ one may replace σ by the permutation obtainable by listing the entries of K in an opposite order and applying an anti-monotonic map to them.* □

Theorem 6.4 shows that there is a map from the set of all skew tableaux to the set of Young tableaux defined by the relation of glissement. We study this map in some special cases. First we consider a skew tableau K whose shape is an *anti Young diagram*, i.e., a skew diagram with a unique *maximal* element in the natural ordering. Let $\lambda \in \mathcal{P}$; a skew diagram χ is called an an anti Young diagram of type λ if it has the form $\chi = \{\, (n-i, m-j) \mid (i,j) \in Y(\lambda) \,\}$ for some fixed pair n, m. An anti Young tableau of type λ is a skew tableau K whose shape is an anti Young diagram of type λ; we write $\lambda = \operatorname{tp} P$. If K is a non-empty anti Young tableau, then K_- will denote the anti Young tableau obtained from it by removing its lowest entry. For any anti Young tableau K the types of the sequence $K, K_-, K_{--}, \ldots$ form a saturated decreasing chain in the Young lattice, and hence define an ordinary Young tableau.

6.6. Definition *Let K be an anti Young tableau of type λ, then $K^\diamond \in \mathcal{T}_\lambda$ is the Young tableau such that* $\mathrm{ch}\, K^\diamond = (\mathrm{tp}\, K, \mathrm{tp}(K_-), \mathrm{tp}(K_{--}), \ldots, (0))$.

We can obtain $K^\diamond$ from K by rotating it by 180° and renumbering the entries in the opposite order. Another Young tableau associated to K is provided by 6.4, and together these two operations define a non-trivial map from the set of Young tableaux to itself. An obvious question is whether this map coincides with S; this is in fact the case, although it is not at all evident from the definitions.

6.7. Theorem. *Let K be an anti Young tableau, and let T be the Young tableau such that $K \triangleright T$; then $K^\diamond = S(T)$.*

Proof. We may assume the entries of K to be such that T is normalised. Various proofs are possible; we sketch two of them, leaving the details to the reader. One proof, which is based on the connection between the definition of S and the relation of glissement, is by induction on the size of $\mathrm{sh}\, K$, using 6.4 and applying the induction hypothesis twice, namely for K_- and for the anti Young tableau $K^\uparrow$ satisfying $K^{\uparrow\diamond} \sim K^{\diamond\downarrow}$. In this approach one also needs an auxiliary fact, namely that if two *successive* numbers appear as horizontally adjacent entries of K, then they can never be vertically adjacent in any skew tableau related to K by glissement; this is easily proved either directly or from 6.4. A simpler proof of the theorem can be given however, based on using 6.5 to compute $S(T)$. When we successively extract the entries from K in a columnwise fashion, and apply $\tilde{w}$ to the terms of the resulting sequence (i.e., renumber in the opposite order), then we obtain a sequence that is also obtainable by reading the columns of $K^\diamond$ from bottom to top, and proceeding from left to right. It is not difficult to see that $K^\diamond$ can be reconstructed from this sequence by applying R^{-1} and selecting the left tableau—once again each insertion step only moves entries down a single column—and since this is just what 6.5 prescribes, the theorem follows. □

This theorem is implicit in [Schü2], where in fact the operation S is *defined* by a combination of central symmetry and glissement, while our basic definition of S is stated as an alternative way of computing it. There is a particular instance of the theorem that is particularly noteworthy, as it can be formulated without any reference to glissement, and moreover it implies the general case of the theorem without much difficulty. In fact, it is even possible to derive 6.4 from it without using 5.1. The case referred to is that in which $\mathrm{sh}\, K$ is rectangular, i.e., when K is at the same time a Young tableau and an anti Young tableau; we then have $T = K$ in the theorem, and we therefore have

6.8. Corollary. *Let $\lambda \in \mathcal{P}_n$ be a rectangular partition, and let $T \in \mathcal{T}_\lambda$. Then $S(T) = T^\diamond$, i.e., $S(T)$ can be obtained from T by 180° rotation followed by renumbering its entries in the opposite order.* □

In the opinion of the author, this corollary is the most amazing of all the remarkable facts about the Robinson-Schensted and Schützenberger algorithms, having such a simple formulation, yet apparently without a correspondingly simple proof (in our setup it depends essentially on 5.1; note however that a relatively simple non-combinatorial proof is given in Part II of this paper). It implies that, in the computation of $S(T)$ by its original definition, using the sequence of intermediate tableaux $T^{\downarrow\cdots\downarrow}$, the position of the square coming free in the i-th step is determined only by λ and the position of the entry i in the original tableau T, and by nothing else. Therefore that entry i, while moving towards its position $(1,1)$ in $T^{\downarrow(i-1)}$, has like magic managed to encode the information about its original position into the position of the remaining squares (and the other entries are simultaneously performing the same feat)! We mention another surprising fact that follows from it.

6.9. Corollary. *Let $n > 0$, let λ be a rectangular partition of n, and consider the operation g on the set of Young tableaux of shape λ whose set of entries form an interval of $\mathbf{Z}$, defined by*

$$g(P) = \textbf{let } (T, s, i) = D(P) \textbf{ in } T : (s, i + n)$$

(here s always is the unique corner of λ). Then from any such tableau T one can obtain $g^n(T)$ by increasing all entries by n.

Proof. Divide each rectangular tableau $g^i(T)$ into the Young tableau T_i of entries $\leq n$ and the anti Young tableau U_i of entries $> n$. We have $T_i = T^{\downarrow(i)}$, and hence $\mathrm{ch}(S(T)) = (\lambda, \mathrm{sh}\, T_1, \ldots, \mathrm{sh}\, T_{n-1}, (0))$;

applying the central symmetry in the centre of T we get $\mathrm{ch}(S(T)^\diamond) = (\lambda, \mathrm{tp}\, U_{n-1}, \ldots, \mathrm{tp}\, U_1, (0))$. Using the notation K^- to indicate removal of the highest entry of a skew tableau K in the same way as for Young tableaux, we have the relation $U_{i-1} \triangleright_{s_i} U_i^-$, where s_i is such that $(T_i, s_i, m) = D(T_{i-1})$. Observing that central symmetry transforms glissements into reverse glissements, we obtain $U_i^{\diamond\downarrow} \sim U_{i-1}^\diamond$. We therefore have $\mathrm{ch}(S(U_n^\diamond)) = (\lambda, \mathrm{tp}\, U_{n-1}, \ldots, \mathrm{tp}\, U_1, (0)) = \mathrm{ch}(S(T)^\diamond)$. By 6.8 we have $S(T)^\diamond \sim T \sim S(T^\diamond)$, and since S and $\cdot^\diamond$ are invertible operations and $U_n = g^n(T)$ we conclude that $g^n(T) \sim T$, and since the set of entries of $g^n(T)$ is clearly that of T shifted up by n, the corollary follows. □

Remark. To prove the corollary it would have sufficed that the operations S and $\cdot^\diamond$ on the set of rectangular tableaux commute, while in fact they coincide, as is stated by 6.8. The corollary was suggested to the author as a conjecture by A. E. Brouwer.

Finally we consider skew diagrams χ, of which all connected components are singleton sets. This means that χ is totally ordered by the column number of the squares, and that any injective map $\chi \to \mathbf{Z}$ is a skew tableau of shape χ. Therefore such skew tableaux correspond to arbitrary sequences of distinct numbers. The following is immediate from 6.5, together with 3.1; it shows that the Robinson-Schensted algorithm can be computed just by glissement.

6.10. Corollary. *Let σ be a permutation, K a skew tableau such that the connected components of* $\mathrm{sh}\, K$ *are all singletons, whose entries, listed by increasing column numbers, form the sequence for σ, and let K' be similarly defined, but for σ^{-1} in place of σ. Then the unique Young tableaux P, Q for which $K \triangleright P$ and $K' \triangleright Q$ satisfy $(P, Q) = R^{-1}(\sigma)$.* □

In fact one can simulate Schensted's original definition of the Robinson-Schensted algorithm directly by glissements, where the intermediate Young tableaux P_i during the insertion process are represented by skew tableaux whose connected components correspond to the rows of P_i, with the entries yet to be inserted to their upper right (this method is used in [Schü2]). The crucial ingredient is that the insertion of a number a into a row $x_1, \ldots, x_k$ yielding a new row $x_1', \ldots, x_k'$ while displacing the number b, corresponds to the transformation

obtained by a $k-1$-step glissement. Using this one shows how the squares of K can be incorporated one by one, in order of increasing column number. After simulating the last insertion step, the rows of P_n are shifted to the left so as to make their left edges align, which produces P as a proper Young tableau.

We do not elaborate any further on the many questions that arise from the viewpoint of glissement. More interesting facts can be found in [Schü2], such as the following set of invariants. For each $k \in \mathbf{N}$ define a function d_k that associates to a skew tableau K the maximal value of $\#X$ among the subsets $X \subseteq \mathrm{sh}\, K$ that have the property that within X one can form no sequence of length exceeding k of squares whose entries in K *and* whose row numbers are at the same time strictly increasing. (Such X can be formed as the union of at most k disjoint subsets of $\mathrm{sh}\, K$ in each of which—after ordering them by increasing entries—the row numbers are weakly *decreasing*; to see this take as first such subset the set of squares in X whose entry is smaller than all entries in the rows of X strictly above it, and proceed inductively for the remainder of X.) The functions d_k are constant under glissement, and together determine the shape λ of the Young tableau related to K by glissement, as we have $d_k(K) = \sum_{i \le k} \lambda_i$ (in fact λ is the partition associated, in the way mentioned in remark (5) following Theorem 3.1, to a finite partially ordered set determined by K). It follows that the values $d_k(K^{-\cdots-})$ for all k and all repeated applications of $\cdot^-$ together determine (up to similarity) the tableau to which K is related by glissement. This theorem, a generalisation by Schützenberger of a similar theorem in a somewhat simpler context due Greene, [Gre1], Theorem 3.1, forms the basis for all the main results of [Schü2]; our derivation shows that the basic facts can also be derived without it. An alternative proof of 6.4 appears in [Thom1].

We should also mention the interesting generalisation by A. V. Zelevinsky of the Robinson-Schensted correspondence to so-called "pictures" [Zel], a concept that generalises both permutations and skew tableaux. His results seem to blend very nicely with the theory of glissements, and it seems worth while to try to develop this theory in the general context of pictures.

§7. Robinson's original construction.

In this final section, which is quite independent of the remainder of the paper, we investigate the relationship between the Robinson-Schensted algorithm as discussed above (which, as was indicated, is essentially Schensted's construction), with the construction defined by G de B. Robinson in [Rob], §5. The fact that the algorithm is now named after both these men indicates that their constructions are in some sense equivalent, but the constructions are so dissimilar that even after Schensted's paper was published, their interrelation was not recognised for quite a few years. The first reference to a connection between the algorithms is made in [Kn1], where it is attributed to Schützenberger; in the later literature the fact is soon accepted as common knowledge. Nowhere, however, did we find a precise description of the relationship, let alone a proof of such a claim. Apparently not much further work has been based on Robinson's construction, so the omission is not a very serious one, but nevertheless it is interesting to fill this void in the literature, and this is as good a place as any to do so.

Robinson's paper [Rob] is not very accessible to the (modern) reader for a number of reasons, ranging from trivialities as the indistinguishability of a's and α's in subscripts, to the use of an almost obscurantist language for describing formal constructions and procedures. Fortunately though, in order to decipher §5 one does not have to fully understand the previous sections (indeed we do not, although we tried). Here we shall attempt to reformulate Robinson's construction in a manner that allows comparison with Schensted's construction.

Robinson's construction is not in terms of Young tableaux, but rather of sequences of letters called permutations, although not all letters are distinct; a case of particular interest are the 'lattice permutations', which we have already met at the end of our §2, where we called them 'row encoded tableaux'. (It would appear that the very definition of lattice permutations or row encoded tableaux requires a sequence of numbers, not letters, but the letters are in fact just numbers in disguise, as they are limited to come from the alphabet $\{c_1, c_2, \ldots\}$.) Robinson does note the correspondence between lattice permutations and tableaux, and it is clear that he uses the row (as opposed to column) encoding. At this point the reader might expect Robinson's algorithm to be the same as the one we described at the end of §2, but this is not the case.

For more general "permutations", each letter is given an (extra) index j, indicating that it is the j-th appearance from left to right of that letter, and for each change of the sequence these indices are immediately reassigned. This suggests that we extend the correspondence between lattice permutations and tableaux to the case of arbitrary sequences of letters, as follows. To a sequence σ of length n we associate an arrangement Σ of the numbers $1, \ldots, n$ into rows numbered $1, 2, \ldots$, by the rule that if the k-th letter of σ is the j-th occurrence of the letter c_i, then the square (i, j) of Σ will be filled with the number k. In this way each row of Σ is increasing and left-justified (i.e., there are no vacant positions before the last entry of the row), but empty rows are allowed; it is immediate that σ may be reconstructed form such Σ. From this point on we shall describe everything in terms of these 2-dimensional arrays Σ of numbers. This is particularly convenient because the basic operation on the "permutations" σ is not permuting its terms, but rather changing a letter into another letter: in Σ this corresponds to moving a number from one row of into another, while adjusting any larger numbers in those rows to keep the rows increasing and left-justified.

Robinson describes a procedure for "making any permutation lattice"; this translates to moving the numbers in Σ around until it becomes a Young tableau. To describe the procedure we assign to each number k occupying a square (i, j) in Σ with $i > 1$ a quantity which we shall call its *overhang*: it equals j minus the amount of numbers $< k$ in row $i-1$ of Σ (so if the number were to be moved to the previous row, then its column number would decrease by its overhang minus 1). One easily sees that Σ is a Young tableau if and only if none of its numbers has a positive overhang. Robinson's procedure prescribes repeating the following step until Σ is a Young tableau: search for the first row containing a number with a positive overhang; find in that row the first number with the maximal overhang occurring in the row, and move it to the previous row (adjusting the remaining numbers). Such a move may or may not cause a positive overhang to occur in that previous row, either for the number moved itself or for some larger number. If the move does introduce a positive overhang in the previous row, then the next step

will move a number out from that row, and we shall consider that move to be part of the same "sweep" of moves as its predecessor; a sweep ends when the new positive overhangs introduced by its first move have died out. To each sweep Robinson associates a formal operator $C_{i,i'}$ where i is the row in which the sweep starts, and $i' < i$ is the row in which it ends; his complete procedure returns not only the Young tableau eventually obtained, but also a formal composition of operators $C_{i,i'}$.

Although this procedure is quite general, its main interest lies in the case that one starts off with a real permutation σ, so that Σ initially consists of consecutive rows of length 1: its numbers, read off from the first row to the last, form the sequence corresponding to σ^{-1}. It is also clear that no overhang will exceed 1 initially, and this will remain the case during the whole process (since each move will be into a row in which at that point no number has a positive overhang); hence the rule to look for *maximal* overhang is irrelevant to this case.

The process obviously associates a Young tableau to each permutation; to obtain another tableau, the composition of operators $C_{i,i'}$ produced by the process is applied to a 2-dimensional array Σ', in which the numbers $1, \ldots, n$, are initially arranged into a single increasing column. Here $C_{i,i'}$ is interpreted as "move a number from row i to row i'"; since in each application the value of i will exceed that of its predecessors, there is no ambiguity as to which number to move (namely i). Like with Σ, the rows of Σ' are kept left justified and increasing. Each number moves to the end of a row and will remain in the same square from then on; the shapes of Σ and Σ' will be the same after each sweep, and eventually becomes a Young diagram, whence Σ' eventually becomes a Young tableau. Robinson claims that, given the final values of Σ and Σ' each of the steps of the procedure can be reversed, and that it is not difficult to see that interchanging the tableaux will correspond to inverting the permutation σ (cf. our Theorem 3.1); it is doubtful whether anyone could substantiate these claims without translating to Schensted's formulation.

Despite this 2-dimensional description, the equivalence with Schensteds procedure is not at all evident: it would seem that we are moving in the wrong direction (towards the lower numbered rows), and the rule prescribing the moves is also quite different. Actually, we do not obtain the same tableaux as in Schensted's construction (as is already apparent in the examples Robinson gives), but rather their transposed tableaux. The basic ingredient needed in proving this claim is the following

7.1. Proposition. *Let σ be a sequence of numbers representing a permutation. Arranging the numbers of this sequence into the first column, and then applying Robinson's construction, consider the arrangement obtained after a sweep is completed which started by moving the number in row i. At this point the numbers in the first i rows form a Young tableau, which is the transpose of the left tableau obtained during the application of Schensted's algorithm to σ, after the first i terms have been inserted.*

Proof. The fact that the indicated first i rows form a Young tableau is clear, since no square in these rows has a positive overhang. It is also clear that each sweep starts by moving the next number from the original sequence upwards across any empty lines until it reaches the Young tableau built up during the previous sweeps; the point left to prove is that from this point on the sweep is equivalent to I^t, the transpose of Schensted's insertion procedure. The first observation to be made is that each number which moves to a previous row will end up in the same column as where it came from, since its overhang must have been exactly 1 before the move. Also, if it still has overhang 1 after the move, then it will be the first such number in its row (since the numbers to its left have not be displaced), and it will then be moved up again at the next step. In this way the number moves up in its column until it reaches a smaller number in that column, or the top of the column. Any numbers present in the rows between the one from which the number started to move and the one in which it came to rest will eventually end up in the same position as they were initially, although some may have been bumped temporarily to the right.

We inductively define certain numbers to be 'active' during a sweep, as follows: the number that moves up during the first step of the sweep is active, and if in any step of the sweep an active number moves to a different position and does not have a positive overhang after that step, then the number positioned one square to its right after that step, if any, is also active. Note that an active number can directly cause at most one other number to be declared active, since thereafter it will move no more; it is

possible however that it propagates its activity in the same step as in which it was itself declared active. Since during a sweep the row numbers from which moves take place decrease, each active number except the initial one will have been shifted one column to the right during the step at which it was declared active. After that step the active number may possibly move upwards to rid itself of a positive overhang, but in any case there will be no further change of its column. It is also easy to see that it will eventually occupy the topmost square in that column which did not contain a smaller number before the sweep, and that if that square did contain a larger number, then that is the next active number.

Therefore the sequence of active squares is determined in the same way as the sequence of numbers that are displaced during an application of the transpose insertion procedure I^t, and they move to the same positions. It remains to verify that any non-active number ends up in the same position after the sweep as is was before; this follows because if it is moved to the right at some step by an active number smaller than it entering its row, then there must also be an active number smaller than it leaving its row at the next step (or else it would itself have been active). □

It is remarkable that one move of I^t may correspond to a long sequence of moves in Robinson's procedure, while at the other hand the last move of such a sequence may at the same time achieve the effect of a number of subsequent moves of I^t. To complete the comparison between Robinson's and Schensted's algorithm, observe that the effect of any operator $C_{i,i'}$ is to place the number i in the right tableau at the position by which the shape of the left tableau is increased by the sweep starting in row i; this is analogous to Schensted's rule for building up the right tableau. Taking into account the fact that in (the 2-dimensional form of) Robinson's algorithm one starts with writing down vertically the inverse σ^{-1} of the given permutation, and using Theorem 3.1, we finally obtain

7.2. Proposition. *If for a permutation σ we have $(P,Q) = R^{-1}(\sigma)$ (as computed by Schensted's algorithm), then applying Robinson's algorithm to σ yields the pair of lattice permutations corresponding to the pair of tableaux (Q^t, P^t).* □

References.

[BaVog] D. Barbash and D. Vogan, "Primitive ideals and orbital integrals in complex classical groups", *Mathematische Annalen* **259**, (1982), 153–199.

[BeSt] G. Benkart and J. Stroomer, "Tableau and insertion schemes for spinor representations of the orthogonal Lie algebra $\mathfrak{so}(2r+1,\mathbf{C})$", *J. Combin. Theory, Ser. A* **57**, (1991), 211–237.

[Bere] A. Berele, "A Schensted-type correspondence for the symplectic group", *J. Combin. Theory, Ser. A* **43**, (1986), 320–328.

[Fom1] S. V. Fomin, "Finite partitally ordered sets and Young tableaux", *Soviet Math. Dokl.* **19**, (1978), 1510–1514.

[Fom2] S. V. Fomin, "Generalised Robinson-Schensted-Knuth correspondence", *Journal of Soviet Mathematics* **41**, (1988), 979–991. (Translation from Записки научных семинаров ЛОМИ **155** (1986) 156–175; authorised translation available from the author).

[Fom3] S. Fomin, *Dual graphs and Schensted correspondences*, Proceedings of the 4th International conference on Formal power series and Algebraic combinatorics, Montreal, (1992). Extended version in Institut Mittag-Leffler reports No. 15 & 16 (1991/92).

[Fom4] S. Fomin, *Schur operators and Knuth correspondences*, Institut Mittag-Leffler report No. 17, (1991/92).

[FomSt] S. Fomin and D. Stanton, *Rim hook lattices*, Institut Mittag-Leffler report No. 23, (1991/92).

[FrSch] F. G. Frobenius and I. Schur, "Über die reellen Darstellungen der endlichen Gruppen", *Sitzungsberichte der Preussische Akademie*, (1906), 186–208. *Also in* F. G. Frobenius: *Gesammelte Abhandlungen* (J.-P. Serre ed.), Springer Verlag (1968) 355–377.

[Garf] D. Garfinkle, "On the classification of primitive ideals for complex classical Lie Algebra's, I", *Compositio Mathematica* **75**, (1990), 135–169.

[Gre1] C. Greene, "An extension of Schensted's Theorem", *Advances in Mathematics* **14**, (1974), 254–265.

[Gre2] C. Greene, "Some partitions associated with a partially ordered set", *J. Combin. Theory, Ser. A* **20**, (1976), 69–79.

[JaKer] G. James and A. Kerber, *The representation theory of the symmetric group*, Encyclopedia of mathematics and its applications (G.-C. Rota, ed.) Vol. 16, Addison-Wesley, Reading MA, 1981.

[Jos] A. Joseph, "Sur la classification des idéaux primitifs dans l'algèbre enveloppante d'une algèbre de Lie reductive", *Comptes rendus de l'Acad. Sciences, serie A* **284 #8**, (1977), 425–427.

[KaLu] D. Kazhdan and G. Lusztig, "Representations of Coxeter groups and Hecke algebras", *Inventiones Math.* **53**, (1979), 165–184.

[Kn1] D. E. Knuth, "Permutations, matrices and generalized Young tableaux", *Pacific Journal of Math.* **34**, (1970), 709–727.

[Kn2] D. E. Knuth, *The art of Computer programming, Vol. III Sorting and Searching*, pp. 48–72, Addison-Wesley, 1975.

[vLee1] M. A. A. van Leeuwen, *A Robinson-Schensted algorithm in the geometry of flags for Classical Groups*, Thesis (Rijksuniversiteit Utrecht, the Netherlands), (1989).

[vLee2] M. A. A. van Leeuwen, "The Robinson-Schensted and Schützenberger algorithms and interpretations", pp. 65–88 in *Computational aspects of Lie group representations and related topics* (A. M. Cohen, ed.), CWI Tract 84, 1991.

[vLee3] M. A. A. van Leeuwen, *The Robinson-Schensted and Schützenberger algorithms, Part II: Geometric interpretations*, CWI report AM-R9209, (1992).

[Oka1] S. Okada, "Wreath products by the symmetric groups and product posets of Young's lattice", *J. Combin. Theory, Ser. A* **55**, (1990), 14–32.

[Oka2] S. Okada, "A Robinson-Schensted-type algorithm for $SO(2n, \mathbf{C})$", *Journal of Algebra* **142**, (1991), 334–372.

[Pro] R. A. Proctor, "A Schensted algorithm which models tensor representations of the orthogonal group", *Canad. J. Math.* **42**, (1990), 28–49.

[Rob] G. de B. Robinson, "On the representations of the symmetric group", *American Journal of Math.* **60**, (1938), 745–760.

[Roby] T. W. Roby, *Applications and extensions of Fomin's generalization of the Robinson-Schensted correspondence to differential posets*, Thesis (M. I. T.), (1991).

[Sag1] B. E. Sagan, "An analog of Schensted's algorithm for shifted Young tableaux", *J. Combin. Theory, Ser. A* **27**, (1979), 10–18.

[Sag2] B. E. Sagan, "The ubiquitous Young tableau", pp. 262–298 in *Invariant theory and tableaux* (D. Stanton, ed.), The IMA volumes in mathematics and its applications 19, Springer-Verlag, New York, 1990.

[SaSt] B. E. Sagan and R. P. Stanley, "Robinson-Schensted Algorithms for Skew Tableaux", *J. Combin. Theory, Ser. A* **55**, (1990), 161–193.

[Sche] C. Schensted, "Longest increasing and decreasing subsequences", *Canadian Journal of Math.* **13**, (1961), 179–191.

[Schü1] M. P. Schützenberger, "Quelques remarques sur une construction de Schensted", *Math. Scandinavica* **12**, (1963), 117–128.

[Schü2] M. P. Schützenberger, "La correspondence de Robinson", pp. 59–113 in *Combinatoire et Représentation du Groupe Symétrique* (D. Foata, ed.), Lecture Notes in Mathematics 579, 1976.

[Stan1] R. P. Stanley, "Some aspects of groups acting on finite posets", *J. Combin. Theory, Ser. A* **32**, (1982), 132–161.

[Stan2] R. P. Stanley, *Enumerative Combinatorics, Vol. 1*, Wadsworth and Brooks/Cole, Pacific Grove, CA, 1986.

[Stan3] R. P. Stanley, "Differential posets", *Journal of the Am. Math. Soc.* **1**, (1988), 919–961.

References

[StWhi] D. W. Stanton and D. E. White, "A Schensted algorithm for rim hook tableaux", *J. Combin. Theory, Ser. A* **40**, (1985), 211–247.

[Stb] R. Steinberg, "An Occurrence of the Robinson-Schensted Correspondence", *Journal of Algebra* **113**, (1988), 523–528.

[Stem] J. R. Stembridge, "Rational tableaux and the tensor algebra of $\mathfrak{gl}_n$", *J. Combin. Theory, Ser. A* **46**, (1987), 79–120.

[Sund1] S. Sundaram, *On the combinatorics of representations of* $Sp(2n, \mathbf{C})$, Thesis (M. I. T.), (1986).

[Sund2] S. Sundaram, "Orthogonal tableaux and an insertion algorithm for $SO(2n+1)$", *J. Combin. Theory, Ser. A* **53**, (1990), 239–256.

[Ter] I. Terada, *A Robinson-Schensted-type correspondence for a dual pair on spinors*, to appear in J. Combin. Theory, Ser. A.

[Thom1] G. P. Thomas, "On a construction of Schützenberger", *Discrete Mathematics* **17**, (1977), 107–118.

[Thom2] G. P. Thomas, "On Schensted's construction and the multiplication of Schur functions", *J. Combin. Theory, Ser. A* **32**, (1982), 132–161.

[Vien] G. Viennot, "Une forme géométrique de la correspondance de Robinson-Schensted", pp. 29–68 in *Combinatoire et Représentation du Groupe Symétrique* (D. Foata, ed.), Lecture Notes in Mathematics 579, 1976.

[Vog] D. Vogan, "A generalised τ-invariant for the primitive spectrum of a semisimple Lie-Algebra", *Mathematische Annalen* **242**, (1979), 209–224.

[Whi] D. White, "Some connections between the Littlewood-Richardson rule and the construction of Schensted", *J. Combin. Theory, Ser. A* **30**, (1981), 237–247.

[Zel] A. V. Zelevinsky, "A Generalisation of the Littlewood-Richardson Rule and the Robinson-Schensted-Knuth Correspondence", *Journal of Algebra* **69**, (1981), 82–94.