Sorting with two ordered stacks in series

M. D. Atkinson [a,*] M. M. Murphy [b] N. Ruškuc [b]

[a] *Department of Computer Science, University of Otago, Dunedin, New Zealand*

[b] *School of Mathematics and Statistics, University of St Andrews, St Andrews KY16 9SS*

Abstract

The permutations that can be sorted by two stacks in series are considered, subject to the condition that each stack remains ordered. A forbidden characterisation of such permutations is obtained and the number of permutations of each length is determined by a generating function.

Key words: Stacks, permutations, forbidden patterns, enumeration

1 Introduction

The question of which permutations can be sorted by a single stack, and how many there are of each length, was solved by Knuth in [5]. He showed that a permutation is stack sortable if and only if it has no subpermutation 231 (i.e. subsequence order isomorphic to 231) and that the number of such permutations of length n is the n^{th} Catalan number. At the same time he also introduced the problem of sorting permutations by two or more stacks in series and this was subsequently investigated further by Tarjan in [10].

Let S^k denote the set of permutations that can be sorted by k stacks in series. It is easy to see that this set is *closed* in the sense that subpermutations of permutations in S^k are also in S^k. Consequently S^k is characterised by a set of forbidden permutations, the set of minimal permutations not in S^k, called the *basis*. As noted above the singleton set $\{231\}$ is the basis of S^1.

* Corresponding author.
Email addresses: `mike@cs.otago.ac.nz` (M. D. Atkinson), `max@dcs.st-and.ac.uk` (M. M. Murphy), `nr1@st-and.ac.uk` (N. Ruškuc).

Preprint submitted to Theoretical Computer Science *12 December 2001*

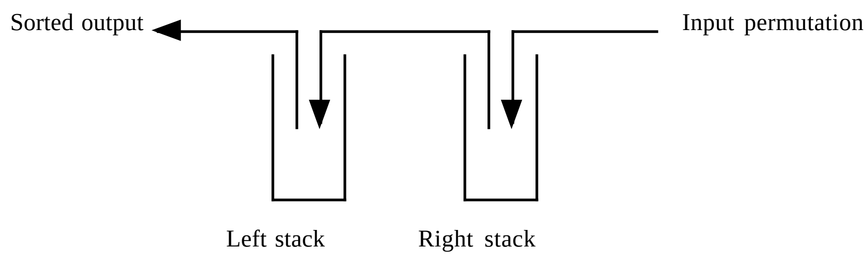

Fig. 1. Two stacks in series

For $k > 1$ it appears to be very difficult to identify the basis of S^k and to enumerate the permutations of each length in S^k. Notice that, if the basis of S^k was finite, we could decide whether a permutation was in S^k in polynomial time. However, there are some indications that the basis of S^2 is infinite; by computationally intensive methods we have found that its basis contains 22 permutations of length 7, 51 of length 8, and 146 of length 9. We conjecture not only that the basis of S^2 is infinite but that it is NP-complete to decide membership in S^2.

In this paper we study a subset M of S^2. The set M consists of all permutations that can be sorted by two stacks in series with each stack remaining sorted from top to bottom. Clearly M is a closed set. Having in mind Figure 1 we refer to the two stacks as the *right* stack (into which elements are first inserted) and the *left* stack (into which elements are transferred from the right stack before being output). An algorithm to sort a permutation using two stacks in series can be described as a sequence of operations, each operation being one of:

ρ: Move an element from the input permutation onto the right stack.
λ: Move an element from the right stack to the left stack.
μ: Move an element from the left stack onto the final output.

Our main results are a determination of the basis and an enumeration of the permutations of each length in the following theorems:

Theorem 1.1 *The basis of M is the infinite set*

$$B = \{(2, 2m-1, 4, 1, 6, 3, 8, 5, \dots, 2m, 2m-3) \mid m = 2, 3, 4, \dots\}.$$

Theorem 1.2 *Let z_n be the number of permutations of length n in M. Then*

$$\sum_{n=0}^{\infty} z_n x^n = \frac{32x}{-8x^2 + 20x + 1 - (1-8x)^{3/2}}$$

Somewhat surprisingly the same generating function appears also in [2]. It enumerates the permutations in the set defined by the single basis element 1342. Thus we have an example of two closed sets, one with an infinite basis the other with a single basis element, which have the same number of permutations of each length. This is an extreme example of a phenomenon that has been noted elsewhere in the literature on closed sets, of different closed sets being equinumerous [1], [8], and [9]; but all previous examples have had bases of the same size.

Our work also provides some common ground between the S^2 problem and a set W of permutations introduced by Julian West [12]. Although West did not describe his set in terms of two stacks in series it has the following description. Consider sorting by two stacks in series under a "greedy" restriction: at every stage, if it is possible to insert the next input symbol into the right stack (a ρ-operation) and keep it sorted then one must do so; otherwise, if it is possible to transfer the top element of the right stack to the left stack (a λ-operation) and keep it sorted then one must do so; otherwise one pops an element from the left stack into the output (a μ-operation). Then W is the set of permutations that this procedure sorts. Since the stacks remain sorted throughout we see that $W \subseteq M$. The set W was enumerated by Zeilberger [13] and further studied by Goulden and West [4] who related it to planar maps. As we shall see, our set M also has a relationship to planar maps.

The set M is related to W in another way. We observe below that there is a greedy algorithm to decide membership in M in linear time. However, while West's set depends on a greedy algorithm that favours operations "to the right" of the system, the set M is sortable by a "left" greedy algorithm. This left greedy algorithm (subsequently referred to simply as the greedy algorithm) prioritises the operations ρ, λ, μ in a different order to West: one carries out a μ-operation if that results in the correct next item being output, otherwise one carries out a λ-operation if the left stack remains sorted, otherwise one carries out a ρ-operation if the right stack remains sorted, and otherwise the algorithm fails. Then it is easily seen that:

Proposition 1.3 *Every permutation in M can be sorted by the greedy algorithm.*

Corollary 1.4 *There is an algorithm which decides whether a permutation σ belongs to M, and which has linear time complexity in the length of σ.*

In the next section we prove Theorem 1.1. Section 3 explains how we view sorting algorithms as words in the basic push and pop operations and Section 4 associates these words to certain labelled trees for which we prove an enumeration result. In the final section we relate these trees to the $\beta(0,1)$-trees of [2] and thereby prove Theorem 1.2.

2 The basis of M

To prove Theorem 1.1 we first note that none of the permutations in B can be sorted by the greedy algorithm. Furthermore, we can readily check that, if any symbol of one of these permutations is deleted, the resulting sequence can be sorted. This proves that B is a subset of the basis of M.

To prove that B is the whole of the basis we shall consider an arbitrary basis permutation β of length n, examine how the greedy algorithm must fail when applied to β, and thereby identify enough properties of β to demonstrate that $\beta \in B$. We denote by $\beta - i$ the sequence obtained by removing the symbol i from β.

Lemma 2.1 *Before the greedy algorithm applied to β fails, the symbol $n-1$ has been processed by a ρ-operation but not by a λ-operation.*

PROOF: If the greedy algorithm fails before attempting to apply a ρ-operation to the symbol n then it would also fail on $\beta - n$ and this contradicts the minimality of β. On the other hand if it fails after a ρ-operation has been successfully applied to n then, from the fact that $\beta - n$ can be sorted by the greedy algorithm, we easily see that β itself can be sorted. It follows that the greedy algorithm fails exactly at the point where it attempts to carry out a ρ-operation on n (failing because the right stack is non-empty).

We compare the action of the greedy algorithm on each of β and $\beta - (n-1)$. It is clear that, up to the point of failure in β, these algorithms must be performing identically except that, for $\beta - (n-1)$, all operations involving $n-1$ are absent. However, the greedy algorithm on $\beta - (n-1)$ would not fail on the ρ-operation to insert n into the right stack (by minimality) and so it follows that, when applied to β, $n-1$ must be present in the right stack at the point of failure. In other words $n-1$ has been processed by a ρ-operation but not by a λ-operation. ■

To complete the proof of Theorem 1.1 we shall construct a subsequence $a_{i_1}, a_{i_2}, \ldots$ of β order isomorphic to a permutation of B. By minimality this will be the whole of β. We shall label the subscripts i_j so that they suggest the relative values of the a_{i_j}.

Consider the point, guaranteed by Lemma 2.1, at which the greedy algorithm inserts $n-1$ into the right stack by a ρ-operation. The left stack is not empty (otherwise a λ-operation could be applied to $n-1$, contradicting Lemma 2.1) and contains a largest element a_2. Thus

$$\beta = \ldots a_2 \theta n - 1 \phi.$$

Now, it cannot be possible to empty the left stack by μ-operations (for that would permit a λ-operation on $n-1$) and so there must exist some a_1 within ϕ with $a_1 < a_2$ and we choose the rightmost such a_1. So we have

$$\beta = \ldots a_2 \theta n - 1 \phi_1 a_1 \ldots$$

Within θ there are no symbols larger than a_2. Indeed, since the right stack must be empty in order to insert $n-1$ into it, each symbol of θ must either be output or on the left stack; in either case it is smaller than a_2. However, within ϕ_1 there must be a symbol larger than a_2 (else when a_1 is processed by a ρ-operation the left stack and all of the right stack except for $n-1$ could be moved to the output and that would allow $n-1$ to move to the left stack). If ϕ_1 contains n then β contains the subsequence $a_2, n-1, n, a_1$ which is order isomorphic to $2341 \in B$ and we are finished. Otherwise the symbols in ϕ_1 that are larger than a_2 cannot be a set of contiguous symbols contiguous with a_2 (for the same reason as before, that they could all be output once a_1 was processed by a ρ-operation). Hence ϕ_1 contains some largest a_4 and there is a smaller a_3 to the right of a_1 also larger than a_2; we choose the rightmost such a_3. Now we have

$$\beta = \ldots a_2 \ldots n-1 \ldots a_4 \ldots a_1 \phi_3 a_3 \ldots$$

Essentially we now repeat the argument of the last paragraph until we run out of symbols. We do it explicitly once more for clarity. Within ϕ_3 there are some symbols larger than a_4 (else when a_3 is processed by a ρ-operation the contents of both stacks, except for $n-1$, could be moved to the output and $n-1$ could move to the left stack). If ϕ_3 contains n, then β contains $a_2, n-1, a_4, a_1, n, a_3$ which is order isomorphic to $254163 \in B$. Otherwise, the set of symbols in ϕ_3 that are larger than a_4 cannot be a contiguous set contiguous with a_4 (or again all the symbols in both stacks, except for $n-1$, could be output). Hence ϕ_3 has a largest symbol a_6 and there is a rightmost smaller symbol a_5 greater than a_4 but smaller than a_6 and to the right of a_3. The situation now is

$$\beta = \ldots a_2 \ldots n-1 \ldots a_4 \ldots a_1 \ldots a_6 \ldots a_3 \phi_5 a_5 \ldots$$

In this way we define more and more symbols of β:

$$a_2, n-1, a_4, a_1, a_6, a_3, \ldots, a_{2k}, a_{2k-3}, \phi_{2k-1}, a_{2k-1}$$

and we do this until ϕ_{2k-1} contains n, in which case we obtain a sequence order isomorphic to $(2, 2k-1, 4, 1, 6, 3, \ldots, 2k, 2k-3) \in B$ as required. This completes the proof of Theorem 1.1.

3 Algorithms as words

This section begins the proof of Theorem 1.2. In that theorem the generating function has constant term 1 corresponding to the empty permutation. However, for technical reasons, we shall from now on consider only non-empty permutations.

An algorithm for sorting a permutation of length n through two stacks in series is a sequence of appropriate stack operations, and so can be described as a word of length $3n$ over the alphabet ρ, λ, μ. We call these *S^2-words*. For a word W over $\{\rho, \lambda, \mu\}$ and $x \in \{\rho, \lambda, \mu\}$, we denote by $\sharp_x(W)$ the number of occurrences of x in W.

It is clear that a word W over $\{\rho, \lambda, \mu\}$ is an S^2-word if and only if it describes how to take a permutation through two stacks in series (without necessarily sorting it). Indeed, if W transforms a permutation $\sigma = (i_1, \dots, i_n)$ into the permutation $\tau = (j_1, \dots, j_n)$, then W sorts the permutation $\tau^{-1}\sigma$. From this it now easily follows that W is an S^2-word if and only if the following two conditions are satisfied:

(S1) $\sharp_\rho(W) = \sharp_\lambda(W) = \sharp_\mu(W)$;
(S2) for any initial subword (prefix) U of W we have $\sharp_\rho(U) \geq \sharp_\lambda(U) \geq \sharp_\mu(U)$.

It is also true that for every S^2-word there is a unique permutation σ which it sorts (σ can be found by applying the S^2-word in reverse so that it defines an algorithm for transforming an output sequence $1, 2, \dots, n$, via the two stacks, to produce σ in the input). The converse, however, is not necessarily true: it may be possible to sort a given permutation in several different ways.

In what follows we will find it useful to label the letters of an S^2-word W as follows. If $\pi = (a_1, \dots, a_n)$ is the permutation sorted by W, then we denote by ρ_i $(1 \leq i \leq n)$ the occurrence of ρ in W which corresponds to moving a_i from the input onto the right stack. Similarly, λ_i moves a_i from the right stack to the left stack, and μ_i outputs a_i from the left stack.

Those S^2-words which represent sortings of permutations whilst respecting the characteristic sorted stack property of M are called *M-words*; and those *M-words* that also represent greedy sortings are called *Greedy M-words*, or *GM-words* for short. We characterise M-words and GM-words in Propositions 3.3 and 3.4 but first we point out our reason for studying GM-words.

Lemma 3.1 *The number of GM-words of length $3n$ is equal to the number of permutations of length n in the set M.*

PROOF: There is a natural one-to-one correspondence between GM-words and permutations of M. Every permutation of M can be sorted by the greedy algorithm and so determines a GM-word. On the other hand, as already observed, each GM-word sorts a unique permutation which necessarily belongs to M. ■

Lemma 3.2 *Let W be an S^2-word, and let $\pi = (a_1, \dots, a_n)$ be the permutation it sorts. Then W is not an M-word if and only if in applying W to π there is a pair of elements a_i and a_j that are adjacent in both stacks.*

Proof: The 'if' part is obvious. For the 'only if' part let a_i, a_j $(i < j)$ be a pair that violates the stack ordering (necessarily on the right stack). Thus we have $a_i < a_j$, and at some stage a_j lies above a_i in the right stack, while at some later stage a_i lies above a_j in the left stack. In addition, assume that a_i and a_j are chosen so that the length of the subword $\lambda_j \dots \lambda_i$ of W is minimal possible. We claim that a_i and a_j are actually adjacent in both stacks.

Assume first that a_i and a_j are not adjacent on the right stack, and let a_k be an entry which lies between them. Since $a_j > a_i$, we must have $a_j > a_k$ or $a_k > a_i$. In the former case, the pair a_k, a_j violates the stack ordering and the sequence $\lambda_j \dots \lambda_k$ is a proper subword of $\lambda_j \dots \lambda_i$, while in the latter case the pair a_i, a_k violates the stack ordering and $\lambda_k \dots \lambda_i$ is a proper subword of $\lambda_j \dots \lambda_i$. In both cases we obtain a contradiction with the choice of a_i and a_j, and so they must be adjacent in the right stack.

Assume now that a_i and a_j are not adjacent on the left stack, and let a_l be an entry which lies between them. In particular, we have $a_l > a_i$. Since a_i and a_j were adjacent in the right stack, a_l must have been moved onto the right stack after a_j had left it, but before a_i had done so. Therefore, the pair a_i, a_l violates the stack ordering, and $\lambda_l \dots \lambda_i$ is a proper subword of $\lambda_j \dots \lambda_i$, which is again in contradiction with the choice of a_i and a_j. ■

Proposition 3.3 *An S^2-word W is an M-word if and only if it contains no subword of the form $\lambda U \lambda$, where U is empty or an S^2-word.*

Proof: Let $\pi = (a_1, \dots, a_n)$ be the permutation sorted by W. If W contains a subword $\lambda_j \lambda_i$ then obviously a_i and a_j are adjacent in both stacks, and W is not an M-word by Lemma 3.2. If W contains a subword of the form $\lambda_j U \lambda_i$, where U is an S^2-word, then after a_j has been moved to the left stack, U transfers a collection of elements from the input, via the two stacks, into the output, and then a_i is moved onto the left stack. We see that again a_i and a_j are adjacent in both stacks, and so W is not an M-word.

Conversely, if W is not an M-word, then, by Lemma 3.2, there is a pair a_i, a_j, such that they are adjacent in both stacks. Consider the subword $\lambda_j U \lambda_i$ of W, and assume that U is non-empty. We see that, after a_j has been moved onto the left stack, no element already on either of the stacks must be moved before a_i is moved on top of a_j. Therefore, U must transfer a group of symbols from the input, via the two stacks, to the output; in other words U must be an S^2-word. ■

Proposition 3.4 *An S^2-word W is a GM-word if and only if the following are satisfied:*

(GM1) *W does not contain a subword $\lambda\lambda$;*
(GM2) *W does not contain a subword $\rho\mu$;*
(GM3) *W does not contain a subword $U\lambda$, where U is an S^2-word.*

PROOF: ($\Rightarrow$) If W contains a subword $\lambda\lambda$ then it is not an M-word by Proposition 3.3. If W contains a subword $\rho\mu$, say $W = V\rho\mu\ldots$, W is not greedy, because a μ can follow V. Similarly, if $W = VU\lambda\ldots$, where U is an S^2-word, then W is not greedy, because a λ can follow V, while the first letter of U is ρ.

($\Leftarrow$) Assume now that W is not a GM-word. If W is not even an M-word then, by Proposition 3.3, it either contains a subword $\lambda\lambda$, or a subword $\lambda U\lambda$, where U is an S^2-word, and the proof is finished. So, let us now consider the case where W is an M-word, but is not greedy. Let π be the permutation sorted by M, and let V be the shortest initial segment of W after which the greedy algorithm condition fails. Thus, if we write $W = VxV_1$, where $x \in \{\rho, \lambda, \mu\}$, there exists another M-word sorting π of the form VyV_2, where x precedes y in the list ρ, λ, μ. So we can distinguish the following three cases.

Case 1: $x = \rho$, $y = \lambda$. Let a_j be the top element in the right stack after V has been applied to π, and write $W = V\rho V_3\lambda_j V_4$. Now note that ρV_3 must not move any of the elements which are already on either of the stacks. Indeed, the elements on the right stack cannot be moved before a_j (because a_j is on the top of the stack), and the elements on the left stack cannot be output before a_j (because a_j is smaller than any of them). Also, since any element input by ρV_3 is smaller than a_j, it must also be output by ρV_3 (i.e. before λ_j). Therefore, ρV_3 is an S^2-subword of W preceding a λ.

Case 2: $x = \rho$, $y = \mu$. Let a_j be the top element of the left stack after V has been applied to π, and write $W = V\rho V_3\mu_j V_4$. Notice that a_j is the least element that has not yet been output. Therefore, V_3 cannot contain any occurrences of either λ or μ, and hence W contains a subword $\rho\mu$.

Case 3: $x = \lambda$, $y = \mu$. This case cannot occur, for if a_j is the top element on the left stack after V has been applied to π, then again it is the least element that has not yet been output, and so applying a λ move would violate the left stack ordering.

This completes the proof of the proposition. ■

4 Algorithms and plane trees

A GM-word W is *reducible* if $W = W_1W_2$, where both W_1 and W_2 are GM-words, and is *irreducible* (or IGM for short) otherwise. In this section, we are going to show how to associate a rooted plane tree with labelled edges to every IGM-word, and then we are going to establish a recurrence formula for the number of IGM-words corresponding to a rooted plane tree without labels.

IGM-words have a significance for the *indecomposable* permutations of M (those which have no proper decomposition into subwords as $\alpha\beta$ where $a < b$ for all $a \in \alpha$ and $b \in \beta$).

Lemma 4.1 *The number of IGM-words of length $3n$ is equal to the number of indecomposable permutations of length n in the set M.*

PROOF: Restrict the one-to-one correspondence given in the proof of Lemma 3.1 to IGM-words. ■

Let W be an IGM-word. Since it represents a greedy algorithm, W must begin with $\rho\lambda$, and it must end with μ; in other words W can be written as $W = \rho\lambda W'\mu$.

We define the *derived* word $\partial(W)$ of a GM-word W to be the word obtained from W by removing the λ symbols. The properties of $\partial(W)$ inherited from conditions (S1) and (S2) are those of well-balanced strings of parentheses and allow a well-known description by a plane tree. In this description $\partial(W)$ is obtained by walking around the tree beginning at the root traversing each edge twice, first downwards for a ρ symbol (opening parenthesis) and later upwards for the corresponding μ symbol (closing parenthesis). We shall use this only in the case of an IGM-word. For such words the tree corresponding to $\partial(W)$ has root of degree 1 and it is convenient to remove the root and its incident edge. The resulting rooted plane tree will be denoted by $T(W)$. By construction, if W has length $3n$, then $T(W)$ has n vertices.

Although $T(W)$ uniquely determines $\partial(W)$ it certainly does not determine W itself. To capture the more detailed information present in W we attach labels to the edges of $T(W)$.

Each edge of $T(W)$ corresponds, as described above, to a ρ-μ pair of W. To each such edge e we attach a label from the set

$$\mathcal{L} = \{(\rho, \mu), (\rho, \mu\lambda), (\rho\lambda, \mu), (\rho\lambda, \mu\lambda)\}$$

depending on whether ρ and μ corresponding to e are followed by λ in W. Trees arising in this way are called *IGM-trees*. We also call the unlabelled tree $T(W)$ the *shape* of the IGM-word W.

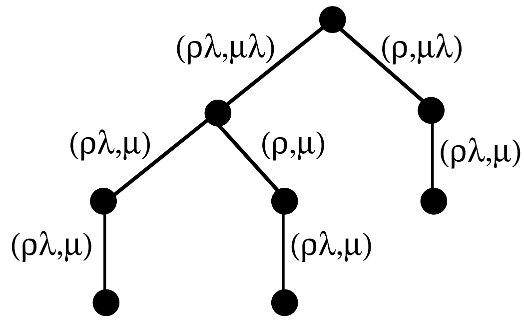

Fig. 2. An IGM-tree

Example 4.2 The IGM-word

$$\rho\lambda\rho\lambda\rho\lambda\rho\lambda\mu\mu\rho\rho\lambda\mu\mu\mu\lambda\rho\rho\lambda\mu\mu\lambda\mu$$

gives rise to the IGM-tree shown in Figure 2.

The IGM-word W can be reconstructed from its IGM-tree $T(W)$ in the intuitively obvious way, which can be formalised as follows. Let T be any plane tree with edges labelled by elements of the set $\mathcal{L}$. To each vertex V of T we associate two words $\sigma(V)$ and $\tau(V)$ defined recursively. If V is a leaf, and if the edge leading to it is labelled by (x, y) then

$$\sigma(V) = x\mu, \ \tau(V) = xy.$$

(Note that if T is an IGM-tree then $\sigma(V) = \tau(V) = \rho\lambda\mu$, because W does not contain a subword $\rho\mu$ or $\rho\lambda\mu\lambda$ by Proposition 3.4.) If V is neither a leaf nor the root, with children $C_1, \ldots, C_p$, and with the edge from V to its parent labelled by (x, y), then define

$$\sigma(V) = x\tau(C_1)\ldots\tau(C_p)\mu,$$
$$\tau(V) = x\tau(C_1)\ldots\tau(C_p)y.$$

Note that either $\sigma(V) = \tau(V)$ or else $\tau(V) = \sigma(V)\lambda$. Finally, if V is the root, and if its children are $C_1, \ldots, C_p$, then define

$$\sigma(V) = \tau(V) = \rho\lambda\tau(C_1)\ldots\tau(C_p)\mu.$$

Clearly, if $T = T(W)$ is an IGM-tree, and if R is its root, then $\sigma(R) = \tau(R) = W$.

We now give some properties of the words $\sigma(V)$ and $\tau(V)$.

Lemma 4.3 *Let $T = T(W)$ be an IGM-tree, and let V be any vertex of T.*

(i) *For every initial segment Z of $\sigma(V)$ we have $\sharp_\rho(Z) \geq \sharp_\lambda(Z)$.*
(ii) *For every terminal segment U of $\sigma(V)$ we have $\sharp_\mu(U) \geq \sharp_\lambda(U)$.*

(iii) *If* $\sharp_\rho(\sigma(V)) = \sharp_\lambda(\sigma(V))$ *then* $\sigma(V)$ *is an* S^2*-word, and hence* $\sigma(V) = \tau(V)$.
(iv) $\sharp_\rho(\tau(V)) \geq \sharp_\lambda(\tau(V))$.

PROOF: If V is a leaf then all the statements hold. Consider now the case where V is not a leaf, and assume inductively that all the statements (i)–(iv) hold for all of its children $C_1, \ldots, C_p$. By the definition we have

$$\sigma(V) = x\tau(C_1)\ldots\tau(C_p)\mu,$$
$$\tau(V) = x\tau(C_1)\ldots\tau(C_p)y.$$

(i) Let Z be an initial segment of $\sigma(V)$. If the length of Z is 1 or 2 the statement is obvious. Otherwise, if Z is a proper initial segment of length greater than 2, we can write

$$Z = x\tau(C_1)\ldots\tau(C_k)Z',$$

where Z' is either empty or else it is a proper prefix of $\tau(C_{k+1})$ (and hence a prefix of $\sigma(C_{k+1})$). We have

$$\sharp_\rho(Z) = 1 + \sum_{i=1}^{k} \sharp_\rho(\tau(C_i)) + \sharp_\rho(Z'),$$
$$\sharp_\lambda(Z) = \sharp_\lambda(x) + \sum_{i=1}^{k} \sharp_\lambda(\tau(C_i)) + \sharp_\lambda(Z').$$

Now note that $1 \geq \sharp_\lambda(x)$, as x is either ρ or $\rho\lambda$. Next note that, by induction, we have $\sharp_\rho(\tau(C_i)) \geq \sharp_\lambda(\tau(C_i))$ (property (iv)) and $\sharp_\rho(Z') \geq \sharp_\lambda(Z')$ (property(i)). We conclude that $\sharp_\rho(Z) \geq \sharp_\lambda(Z)$ in this case. Finally, the case where $Z = \sigma(V)$ follows also by noting that the last μ of $\sigma(V)$ does not contribute anything to either of $\sharp_\rho(\tau(Z))$ or $\sharp_\lambda(\tau(Z))$.

(ii) Let U be a terminal segment of $\sigma(V)$. If U has length 1, the statement is obvious. Otherwise, if U is a proper terminal segment of $\sigma(V)$ of length greater than 1 we can write

$$U = U'\tau(C_k)\ldots\tau(C_p)\mu,$$

where either (1) $U' = \lambda$, or (2) $\sigma(C_{k-1}) = \tau(C_{k-1})$ and U' is a terminal segment of $\sigma(C_{k-1})$, or (3) $U' = U''\lambda$ and U'' is a terminal segment of $\sigma(C_{k-1})$. We now have that

$$\sharp_\mu(U) = \sharp_\mu(U') + \sum_{i=k}^{p} \sharp_\mu(\tau(C_i)) + 1, \tag{1}$$
$$\sharp_\lambda(U) = \sharp_\lambda(U') + \sum_{i=k}^{p} \sharp_\lambda(\tau(C_i)). \tag{2}$$

By induction (property (iv)) we have

$$\sharp_\mu(\tau(C_i)) = \sharp_\rho(\tau(C_i)) \geq \sharp_\lambda(\tau(C_i)) \ (i = k, \dots, p). \tag{3}$$

Also, in each of the three possibilities for U' we have

$$\sharp_\mu(U') + 1 \geq \sharp_\lambda(U'). \tag{4}$$

Indeed in the case (1) this is obvious, while in the cases (2) and (3) it follows from the inductive hypothesis (property (ii)). Combining (1)–(4) we conclude that $\sharp_\mu(U) \geq \sharp_\lambda(U)$, as required.

(iii) We are going to show that conditions (S1) and (S2) are satisfied for $\sigma(V)$. Indeed, (S1) is satisfied by assumption. Also, if Z is an initial segment of $\sigma(V)$ then $\sharp_\rho(Z) \geq \sharp_\lambda(Z)$ by (i). Write now $\sigma(V)$ as $\sigma(V) = ZU$. By (ii) we have $\sharp_\mu(U) \geq \sharp_\lambda(U)$, and hence we have

$$\sharp_\lambda(Z) = \sharp_\lambda(\sigma(V)) - \sharp_\lambda(U) \geq \sharp_\mu(\sigma(V)) - \sharp_\mu(U) = \sharp_\mu(Z),$$

thus proving (S2) as well. The final statement follows from the assumption that W is an IGM-word and condition (GM3) for such words.

(iv) If $\tau(V) = \sigma(V)$ this follows from (i). Otherwise we have $\tau(V) = \sigma(V)\lambda$. We know that $\sharp_\rho(\sigma(V)) \geq \sharp_\lambda(\sigma(V))$. In fact, we must have $\sharp_\rho(\sigma(V)) > \sharp_\lambda(\sigma(V))$ by (iii). Since $\sharp_\lambda(\tau(V)) = \sharp_\lambda(\sigma(V)) + 1$, the statement follows. ∎

Definition 4.4 Let T be a plane tree with edges labelled by elements of the set $\mathcal{L}$, and let V be a vertex of T. The λ-*deficit* at V is the number

$$d(V) = \sharp_\rho(\tau(V)) - \sharp_\lambda(\tau(V)).$$

In the next result we give a characterisation of IGM-trees.

Proposition 4.5 *A plane tree T with edges labelled by elements of the set $\mathcal{L}$ is an IGM-tree if and only if the following conditions are satisfied:*

(T1) *every leaf edge is labelled by $(\rho\lambda, \mu)$;*
(T2) *$d(V) \geq 0$ for every vertex V;*
(T3) *$d(R) = 0$, where R is the root.*

Proof: ($\Rightarrow$) If $T = T(W)$ is an IGM-tree, then (T1) follows from (GM2) and (GM3), (T2) is Lemma 4.3 (iv), and (T3) follows from (S1).

($\Leftarrow$) Assume now that T satisfies (T1)–(T3). We are going to check that the word $W = \sigma(R)$, where R is the root of T, satisfies all conditions (S1), (S2), (GM1)–(GM3), and that it is irreducible. Indeed, (S1) follows from (T3), (GM1) follows from the definition of the words $\sigma(V)$, (GM2) follows from (T1), and (GM3) follows from (T2).

Next we prove that for every initial segment Z of W we have $\sharp_\rho(Z) \geq \sharp_\lambda(Z)$, (the first inequality in (S2)). We do this by induction on the length of Z. If Z has length 1, or, more generally, if Z contains no occurrence of μ, the assertion is obvious. Otherwise Z has one of the forms $Z_1\sigma(V)$ or $Z_1\tau(V)Z_2$, where V is a vertex and Z_2 contains no occurrences of μ. (This is obtained by 'reading' Z until its last μ, and then finding the corresponding ρ in front of it.) Then we have

$$\sharp_\rho(Z_1) \geq \sharp_\lambda(Z_1)$$

by induction,

$$\sharp_\rho(\sigma(V)) \geq \sharp_\lambda(\sigma(V)),\ \sharp_\rho(\tau(V)) \geq \sharp_\lambda(\tau(V))$$

by (T2), and

$$\sharp_\rho(Z_2) \geq \sharp_\lambda(Z_2)$$

since Z_2 contains no μ. Combining the above inequalities as appropriate we conclude that $\sharp_\rho(Z) \geq \sharp_\lambda(Z)$.

A similar induction shows that $\sharp_\mu(U) \geq \sharp_\lambda(U)$ for any terminal segment U of W. This, together with (T3) implies that $\sharp_\lambda(Z) \geq \sharp_\mu(Z)$ for any initial segment Z of W, thus completing the proof of property (S2). Finally, again by induction, one easily proves that $\sharp_\rho(Z) > \sharp_\mu(Z)$ for a proper initial segment Z of W, and this implies that W is irreducible. For, if $W = W_1W_2$, where W_1 and W_2 are GM-words, then W_1 is a proper initial segment of W with $\sharp_\rho(W_1) = \sharp_\mu(W_1)$. ■

Definition 4.6 Let T be a rooted plane tree, and let V be a non-root vertex of T. We define the *branch* of V to be the tree $H(V)$ obtained by taking the subtree of T rooted in V and adding to it the parent P of V and the edge linking P and V.

Definition 4.7 Let T be a rooted plane tree. To each non-root vertex V we associate a sequence

$$\Delta(V) = (\delta_0, \delta_1, \delta_2, \dots),$$

where δ_r is the number of different labellings of edges of $H(V)$ by elements of the set $\mathcal{L}$ which satisfy conditions (T1) and (T2) (but not necessarily (T3)) and for which $d(V) = r$.

Proposition 4.8 *Let T be a rooted plane tree, let R be its root, let $C_1, \dots, C_p$ be the children of R, and let*

$$\Delta(C_i) = (\delta_{i0}, \delta_{i1}, \delta_{i2}, \dots)\ (i = 1, \dots, p).$$

The number of IGM-trees with shape T is equal to $\delta_{10}\delta_{20}\dots\delta_{p0}$.

PROOF: For a given labelling of edges of T satisfying (T1) and (T2) from Proposition 4.5, we have that $\sigma(R)$ is an IGM-word if and only if (T3) is satisfied, i.e. if and only if $d(R) = \sharp_\rho(\tau(R)) - \sharp_\lambda(\tau(R)) = 0$. Since

$$\tau(R) = \rho\lambda\tau(C_1)\dots\tau(C_p)\mu,$$

and since $d(C_i) \geq 0$ $(i = 1, \dots, p)$, we have that $d(R) = 0$ if and only if $d(C_i) = 0$ for all $i = 1, \dots, p$. The number of different labellings of any $H(C_i)$ satisfying $d(C_i) = 0$ is precisely δ_{i0} and the result follows. ∎

We now give a recurrence for the sequences $\Delta(V)$.

Proposition 4.9 *Let T be a rooted plane tree, let V be a non-root vertex of T, and let*

$$\Delta(V) = (\delta_0, \delta_1, \delta_2, \dots).$$

If V is a leaf then

$$\delta_0 = 1, \;\; \delta_i = 0 \;\; (i > 0).$$

Otherwise, if $C_1, \dots, C_p$ are the children of V with

$$\Delta(C_i) = (\delta_{i0}, \delta_{i1}, \delta_{i2}, \dots),$$

then

$$\delta_r = \sum_{j_1+\dots+j_p=r-1} \delta_{1j_1}\delta_{2j_2}\dots\delta_{pj_p} + 2\left(\sum_{j_1+\dots+j_p=r} \delta_{1j_1}\delta_{2j_2}\dots\delta_{pj_p}\right) + \sum_{j_1+\dots+j_p=r+1} \delta_{1j_1}\delta_{2j_2}\dots\delta_{pj_p}.$$

PROOF: Note that

$$d(V) = \sharp_\rho(xy) - \sharp_\lambda(xy) + \sum_{i=1}^{p} d(C_i),$$

where (x, y) is the label of the edge connecting V to its parent. Clearly

$$\sharp_\rho(xy) = 1, \;\; \sharp_\lambda(xy) = \begin{cases} 0 \text{ if } x = \rho, \;\; y = \mu, \\ 1 \text{ if } x = \rho\lambda, \;\; y = \mu \text{ or } x = \rho, \;\; y = \mu\lambda, \\ 2 \text{ if } x = \rho\lambda, \;\; y = \mu\lambda. \end{cases}$$

Hence, to be able to label $H(V)$ so that $d(V) = r$, the trees $H(C_i)$ $(i = 1, \dots, p)$ must be labelled so that

$$\sum_{i=1}^{p} d(C_i) \in \{r-1, r, r+1\}.$$

A labelling of $H(C_i)$ $(i = 1, \dots, p)$ with $\sum_{i=1}^{p} d(C_i) = r \pm 1$ can be extended in a unique way to a labelling of $H(V)$ with $d(V) = r$ by setting $x = \rho\lambda$, $y = \mu\lambda$ or $x = \rho$, $y = \mu$ respectively. Similarly, a labelling of $H(C_i)$ $(i = 1, \dots, p)$ with $\sum_{i=1}^{p} d(C_i) = r$ can be extended in two ways to a labelling of $H(V)$ with $d(V) = r$ by setting $x = \rho\lambda$, $y = \mu$ and $x = \rho$, $y = \mu\lambda$. Finally note that the number of labellings of $H(C_i)$ $(i = 1, \dots, p)$ with $\sum_{i=1}^{p} d(C_i) = k \in \{r-1, r+1, r\}$ is precisely

$$\sum_{j_1+\dots+j_p=k} \delta_{1j_1}\delta_{2j_2}\dots\delta_{pj_p},$$

proving the formula. We remark that the argument remains valid for $r = 0$, when the term

$$\sum_{j_1+\dots+j_p=-1} \delta_{1j_1}\delta_{2j_2}\dots\delta_{pj_p}$$

is zero, reflecting the fact that the λ-deficit of every C_i is non-negative (Proposition 4.5 (T2)). ■

Remark 4.10 Although $\Delta(V)$ is an infinite sequence, only finitely many of its entries are non-zero. In other words the generating function $\Delta(V, x) = \sum \delta_i x^i$ of $\Delta(V)$ is a polynomial. From the recurrence of Proposition 4.9 we easily derive the polynomial equations:

$$\Delta(V, x) = 1 \tag{5}$$

if V is a leaf, and

$$\Delta(V, x) = \frac{1}{x}((1+x)^2 \prod_{i=1}^{p} \Delta(C_i, x) - \prod_{i=1}^{p} \Delta(C_i, 0)) \tag{6}$$

if V is a non-leaf, non-root vertex with children $C_1, \dots, C_p$. By Proposition 4.8 the number of IGM-trees of shape T is

$$\prod_{i=1}^{p} \Delta(C_i, 0) \tag{7}$$

where $C_1, \dots, C_p$ are the children of the root.

Example 4.11 Consider the plane tree of Figure 3. For each vertex V_i $(i = 1, \dots, 7)$ we calculate the corresponding polynomial $\Delta(V_i, x)$. First

$$\Delta(V_1, x) = \Delta(V_2, x) = \Delta(V_3, x) = 1,$$

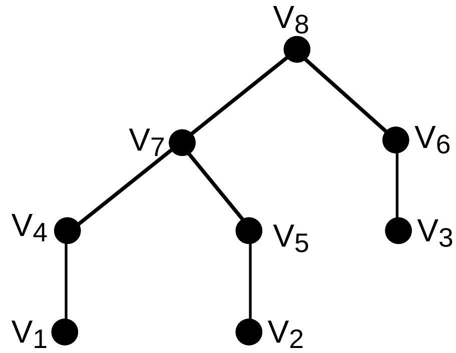

Fig. 3.

because they are the leaves. Now, V_4 has one child, V_1. Applying (6) we obtain

$$\Delta(V_4, x) = \frac{1}{x}((1+x)^2 - 1) = 2 + x.$$

Similarly,

$$\Delta(V_5, x) = \Delta(V_6, x) = 2 + x.$$

The vertex V_7 has two children, V_4 and V_5, so

$$\Delta(V_7, x) = \frac{1}{x}((2+x)^2(1+x)^2 - 4) = 12 + 13x + 6x^2 + x^3.$$

By Proposition 4.8 we can conclude that there are $\Delta(V_6, 0) \cdot \Delta(V_7, 0) = 2 \cdot 12 = 24$ IGM-trees with this given shape.

5 *IGM*-trees and $\beta(0,1)$-trees: enumeration

In this section we introduce the concept of $\beta(0,1)$-trees, as rooted plane trees with labelled vertices. Then we establish a recurrence formula, giving the number of $\beta(0,1)$-trees over a given rooted plane tree (with no labels), and establish connections between this recurrence relation and the one from the last section. From here it then follows that the numbers of IGM-trees and $\beta(0,1)$-trees over a given rooted plane tree are equal. Finally, we use this fact to give a proof of Theorem 1.2.

Definition 5.1 A $\beta(0,1)$-tree is a rooted plane tree with non-negative integer labels $l(V)$ on its vertices, satisfying the following conditions:

(B1) if V is a leaf then $l(V) = 0$;
(B2) if V is an internal vertex, and if $C_1, \ldots, C_p$ are its children then $l(V) \leq l(C_1) + l(C_2) + \ldots + l(C_p) + 1$;
(B3) if V is the root then $l(V) = 0$.

We note that this differs from the definition of a $\beta(0,1)$-tree given in [2] (and the more general definition in [3]) where one requires that $l(V) = l(C_1) + l(C_2) + \ldots + l(C_p)$ when V is the root with children $C_1, \ldots, C_p$. However, this difference will not affect the number of $\beta(0,1)$-trees, as both give no freedom of choice for the label of the root.

The following result was given in [2]. Proofs may be found in [6] and [7].

Proposition 5.2 *The number t_n of $\beta(0,1)$-trees on n vertices is equal to the number of rooted bicubic maps on $n-1$ vertices.*

From the enumeration of rooted bicubic maps given in [11] (see also [8] for a combinatorial proof) we have $t_1 = 1$ and, for $n > 1$, $t_n = 3 \cdot 2^{n-2} \cdot (2n-2)!/(n+1)!(n-1)!$.

Definition 5.3 Let T be a rooted plane tree. To each vertex V we associate a sequence

$$B(V) = (\beta_0, \beta_1, \beta_2, \ldots),$$

where β_r is the number of different labellings of the subtree of T rooted in V which satisfy conditions (B1) and (B2) of Definition 5.1 (but not necessarily condition (B3)), and in which V is labelled by r.

Remark 5.4 If R is the root of T, and if $B(R) = (\beta_0, \beta_1, \beta_2, \ldots)$, then the number of $\beta(0,1)$-trees with shape T is equal to β_0.

Remark 5.5 As with $\Delta(V)$, we see that $B(V)$ is an infinite sequence with only finitely many non-zero entries.

In the following proposition we give a recurrence for computing the sequences $B(V)$ in an arbitrary rooted plane tree.

Proposition 5.6 *Let T be a rooted plane tree, let V be any vertex in it, and let $B(V) = (\beta_0, \beta_1, \beta_2, \ldots)$. If V is a leaf, then*

$$\beta_0 = 1, \; \beta_i = 0 \; (i \geq 1).$$

If V is not a leaf, and if $C_1, \ldots, C_p$ are its children, with

$$B(C_i) = (\beta_{i0}, \beta_{i1}, \beta_{i2}, \ldots),$$

then

$$\beta_r = \sum_{j_1 + \ldots + j_p \geq r-1} \beta_{1j_1} \beta_{2j_2} \ldots \beta_{pj_p}.$$

Proof: A leaf must be labelled by 0, hence the first assertion. For the second, note that one is allowed to label V by r if and only if the sum of its children's labels is at least $r-1$. ■

Remark 5.7 The recurrence of Proposition 5.6 also gives relations between the polynomials $B(V,x)=\sum \beta_i x^i$. Indeed we have:

$$B(V,x)=1 \tag{8}$$

if V is a leaf, and

$$B(V,x)=\frac{1}{x-1}(x^2\prod_{i=1}^{p}B(C_i,x)-\prod_{i=1}^{p}B(C_i,1)) \tag{9}$$

if V is a non-leaf vertex with children $C_1,\ldots,C_p$. An easy consequence of (9) is

$$\prod_{i=1}^{p}B(C_i,1)=B(V,0). \tag{10}$$

Also, from the definition $B(V,0)=\beta_0$. So if R is the root node and $C_1,\ldots,C_p$ are the children of the root then the number of $\beta(0,1)$-trees of shape T is

$$B(R,0)=\prod_{i=1}^{p}B(C_i,1). \tag{11}$$

Example 5.8 Consider the plane tree shown in Figure 3. For each vertex V_i we calculate the corresponding polynomial $B(V_i,x)$. First

$$B(V_1,x)=B(V_2,x)=B(V_3,x)=1.$$

because they are the leaves. The vertex V_4 has but one child, V_1, giving:

$$B(V_4,x)=\frac{x^2-1}{x-1}=1+x.$$

Similarly,

$$B(V_5,x)=B(V_6,x)=1+x.$$

Now, V_7 has two children, V_4 and V_5. From their polynomials we obtain:

$$B(V_7,x)=\frac{(1+x)^2x^2-4}{x-1}=4+4x+3x^2+x^3.$$

Finally, a similar calculation for V_8 involving its children V_6 and V_7, gives:

$$B(V_8,x)=\frac{(4+4x+3x^2+x^3)(1+x)x^2-24}{x-1}=24+24x+20x^2+12x^3+5x^4+x^5.$$

We conclude that there are $B(V_8,0)=24$ $\beta(0,1)$-trees with this given shape.

If we compare the equations of Remarks 4.10 and 5.7 we obtain:

Proposition 5.9 *Let T be a plane tree and let V be a non-root vertex of T. Then the following polynomial equality holds:*

$$\Delta(V, x-1) = B(V, x)$$

PROOF: We prove the proposition by induction. If V is a leaf then

$$\Delta(V, x-1) = 1 = B(V, x)$$

Otherwise, if V has children $C_1, \ldots, C_p$ and if we assume that the proposition holds for $C_1, \ldots, C_p$, then

$$\begin{aligned}
\Delta(V, x-1) &= \frac{1}{x-1}(x^2 \prod_{i=1}^{p} \Delta(C_i, x-1) - \prod_{i=1}^{p} \Delta(C_i, 0)) && \text{(by (6))} \\
&= \frac{1}{x-1}(x^2 \prod_{i=1}^{p} B(C_i, x) - \prod_{i=1}^{p} B(C_i, 1)) && \text{(induction)} \\
&= B(V, x) && \text{(by (9))}
\end{aligned}$$

as required, thus completing the proof.

■

Theorem 5.10 *Let T be a rooted plane tree. The number of IGM-trees with shape T is equal to the number of $\beta(0,1)$-trees with shape T.*

PROOF: Let R be the root of T and let $C_1, \ldots, C_p$ be its children. Then the number of IGM-trees with shape T is $\prod_{i=1}^{p} \Delta(C_i, 0)$, by Remark 4.10. On the other hand, by Remark 5.7, the number of $\beta(0,1)$-trees with shape T is $\prod_{i=1}^{p} B(C_i, 1)$. By Proposition 5.9 these are equal. ■

PROOF OF THEOREM 1.2: First we observe that the number of IGM-words of length $3n$ is equal to the number, t_n, of $\beta(0,1)$-trees on n vertices. This follows from Theorem 5.10 by summation over all tree shapes, and the fact that IGM-words and IGM-trees are in one-to-one correspondence. By Lemma 4.1 this number is also equal to the number of indecomposable permutations of length n in M; so this number is t_n.

We complete the proof by following an argument similar to that used in [2]. Every permutation σ of M has a unique factorisation (as a word) $\sigma = \tau_1\tau_2\ldots\tau_m$ with $a < b$ whenever $a \in \tau_i, b \in \tau_{i+1}$. The subwords τ_i are order isomorphic to indecomposable permutations of M. Conversely, every sequence of indecomposable permutations of M determines a permutation of M in this way. It follows from this that the generating function for the numbers z_n of permutations of length n in the set M, including the empty permutation, is

$$\sum_{k=0}^{\infty} F(x)^k = \frac{1}{1 - F(x)}$$

where $F(x)$ is the generating function for the (non-empty) indecomposable permutations of M. However, Tutte [11] has proved that

$$F(x) = \sum_{n=1}^{\infty} t_n x^n = \frac{8x^2 + 12x - 1 + (1 - 8x)^{3/2}}{32x}$$

and our theorem now follows. ■

Acknowledgement. The authors are grateful to the anonymous referee for his/her receptive reading of the paper and useful comments. In particular, the use of generating polynomials $\Delta(V, x)$ and $B(V, x)$ has helped us to streamline part of the exposition in Sections 4 and 5.

References

[1] M. D. Atkinson: Restricted permutations, Discrete Math. 195 (1998), 27–38.

[2] M. Bóna: Exact enumeration of 1342-avoiding permutations; A close link with labeled trees and planar maps, Journal of Combinatorial Theory, Series A, 80 (1997) 257–272.

[3] R. Cori, B. Jacquard, and G. Schaeffer: Description trees for some families of planar maps, Paper presented at 9th Conference on Formal Power Series and Algebraic Combinatorics, Vienna 1997. See also http://www.liafa.jussieu.fr/ cp/FPSAC/FPSAC97/ARTICLES/Schaeffer.ps.gz

[4] I. P. Goulden and J. West: Raney paths and a combinatorial relationship between nonseparable rooted planar maps and two-stack-sortable permutations, J. Combinatorial Theory (A) 75 (1996) 220–242.

[5] D. E. Knuth: *Fundamental Algorithms, The Art of Computer Programming* Vol. 1 (Second Edition), Addison-Wesley, Reading, Mass. (1973).

[6] B. Jacquard: Cartes et arbres; enumeration, generation et dessins, PhD thesis, École Polytechnique, Palaiseau, 1997.

[7] M. Bousquet-Mélou, G. Schaeffer: Enumeration of planar constellations, Adv. in Applied Math.(to appear).

[8] Z. E. Stankova: Forbidden subsequences, Discrete Math. 132 (1994), 291–316.

[9] Z. E. Stankova: Classification of forbidden subsequences of length 4, European J. Combin. 17 (1996), no. 5, 501–517.

[10] R. E. Tarjan: Sorting using networks of queues and stacks, Journal of the ACM 19 (1972), 341–346.

[11] W. T. Tutte: A census of planar maps, Can. J. Math, 15 (1963), 249–271.

[12] Julian West: Sorting twice through a stack, Theoretical Computer Science, 117(1-2) (1993):303–313

[13] D. Zeilberger: A proof of Julian West's conjecture that the number of 2-stack-sortable permutations of length n is $2(3n)!/((2n+1)!(n+1)!)$, Discrete Math. 102 (1992), 85–93 .