Mathematics Faculty Research   Mathematics

A bijection on classes enumerated by the Schröder numbers

Michael W. Schroeder

Rebecca Smith

A Bijection on Classes Enumerated by the Schröder Numbers

Michael W. Schroeder[1]　　　　　　　　Rebecca Smith[2*]

[1] *Marshall University, Huntington, WV USA*
[2] *SUNY Brockport, Brockport, NY USA*

We consider a sorting machine consisting of two stacks in series where the first stack has the added restriction that entries in the stack must be in decreasing order from top to bottom. The class of permutations sortable by this machine is known to be enumerated by the Schröder numbers. In this paper, we give a bijection between these sortable permutations of length n and Schröder paths of order $n-1$: the lattice paths from $(0,0)$ to $(n-1, n-1)$ composed of East steps $(1,0)$, North steps $(0,1)$, and Diagonal steps $(1,1)$ that travel weakly below the line $y = x$.

Keywords: algorithm, data structure, lattice path, permutation pattern, Schröder Number, sorting, stack

1　Introduction

A stack is a sorting device that works by a sequence of push and pop operations. This last-in, first-out machine was shown by Knuth [8] to sort a permutation if and only if that permutation avoids the pattern 231. That is, if there are not three indices $i < j < k$ with $\pi_k < \pi_i < \pi_j$, then it is possible to run π through a stack and output the identity permutation. The class of stack-sortable permutations is enumerated by the Catalan numbers.

In the language of permutation patterns, any downset of permutations in the permutation containment ordering is a *class*, and every class has a *basis*, which consists of the minimal permutations not in the class. Given that the basis for the class of permutations sortable by one stack contains only a single pattern of length three, considering two stacks in series is quite natural. However, the problem becomes rather unwieldy. In the case of two stacks in series, Murphy [11] showed that the class of sortable permutations has an infinite basis. The enumeration of this class also appears to be difficult. The best known bounds are given by Albert, Atkinson, and Linton [2].

We note that to sort a permutation by two stacks in series, the push and pop operations are such that when an entry is popped out of the first stack, it is immediately pushed into the second stack.

To get a better handle on this problem, many have considered different types of weaker sorting machines. One such weaker machine is a stack in which the entries must increase when read from top to bottom. Atkinson, Murphy, and Ruškuc [3] found an optimal algorithm for sorting permutations which are sortable

using two increasing stacks in series. Note that to obtain the identity permutation, the last stack will be an increasing stack even without declaring this restriction. Their left-greedy algorithm sorts all permutations sortable by this machine. Interestingly enough, the basis for these sortable permutations is still infinite, but the permutation class was found to be in bijection with the permutations that avoid 1342 as enumerated by Bóna [5]. Both enumerations were found by using a bijection with $\beta(0, 1)$ trees.

One can analogously define a *decreasing stack* as a stack in which the entries must decrease when read from top to bottom. Smith [14] considered sorting with a decreasing stack followed by an increasing stack, a machine called DI (we refer to the decreasing stack as D and the increasing stack as I). We illustrate how a permutation can be sorted with the DI sorting machine in Figure 1. Notice that this permutation contains the pattern 2341, and as such cannot be sorted by two increasing stacks in series. The class of DI-sortable permutations was shown to have a finite basis, $\{3142, 3241\}$. Kremer [9, 10] has shown previously that this class is enumerated by the large Schröder numbers.
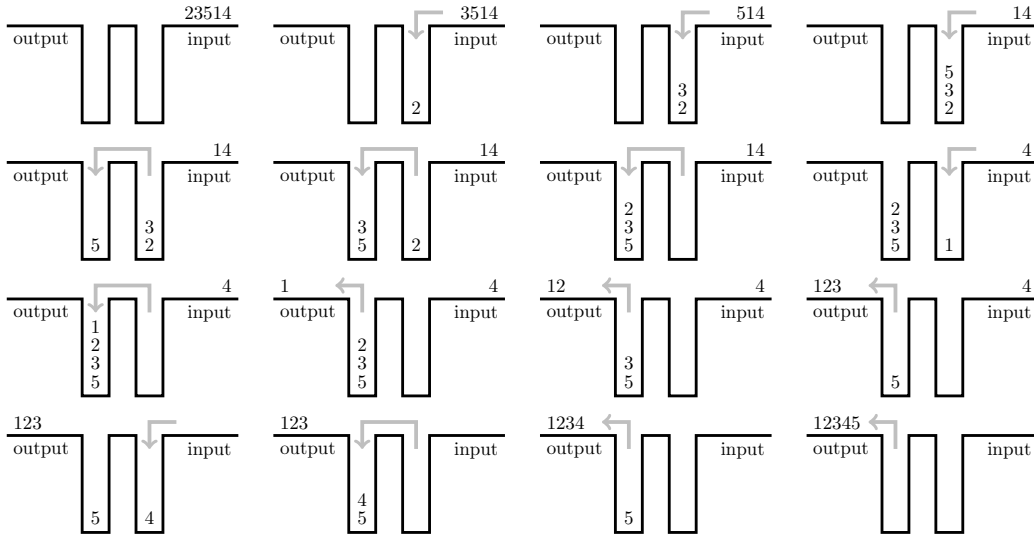

Fig. 1: Sorting the permutation 23514.

The Schröder numbers were introduced in somewhat modern times by Schröder [13] as legal bracketing of variables, though at least the initial terms of this sequence were known to Hipparchus [15]. Rogers and Shapiro [12] found bijections showing certain classes of lattice paths are enumerated by the Schröder numbers. While there are several such classes of lattice paths, we use the one given in the definition below.

Definition 1.1. Let $n \geq 1$. A lattice path from $(0, 0)$ to (n, n) taking only East $(1, 0)$, North $(0, 1)$, and Diagonal $(1, 1)$ steps while staying weakly below the main diagonal will be referred to as a *Schröder path*. We illustrate all of the Schröder paths from $(0, 0)$ to $(2, 2)$ in Figure 2.

In Section 2, we state the algorithm from [14] for producing a word from a DI-sortable permutation, then classify when a sorting word is the result of the algorithm. In Section 3, we give an algorithm which produces a Schröder path from a word produced by the algorithm in Section 2, then show there is a bijective correspondence between algorithmic DI words and Schröder paths. In Section 4, we discuss
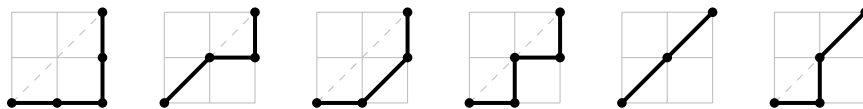
Fig. 2: Schröder paths from $(0,0)$ to $(2,2)$.

some properties of DI-sortable permutations which can be ascertained from aspects of their corresponding Schröder paths, and we conclude with some open questions.

On a related note, Ferrari [7] gave bijections between permutations sorted by restricted deques and a different class of lattice paths also enumerated by the Schröder numbers. Also, Bandlow, Egge, and Killpatrick [4] gave a bijection between a different permutation class and Schröder paths. Additionally, more background on stacks in general was consolidated by Bona [6].

2 The Sorting Algorithm

Let W be a word, i be a positive integer, and $\mathtt{L}$ be a letter. In what follows, we denote the letter in the ith position of W as $W(i)$, $\#_{\mathtt{L}}(W)$ as the number of occurrences of $\mathtt{L}$ in W, and W^i as the first i characters in W. We often refer to W^i as a *prefix* of W. We use $\mathtt{L}^i$ to denote i repetitions of $\mathtt{L}$.

In [14], Smith showed the following algorithm was an optimal way to sort permutations using the DI machine in the sense that this algorithm would sort any DI-sortable permutation.

Algorithm 2.1.

1. *If the top entry of the second (increasing) stack is the next entry of the output, then pop the entry to the output.*

2. *If all of the m entries in the first (decreasing) stack make up the next m entries of the output, then push those entries to the second stack.*

3. *Otherwise, if the next entry of the input is smaller than the top entry of the second stack and larger than the top entry of the first stack, then push it onto the first stack. We will apply the convention that the next entry of the input satisfies each of the aforementioned properties if there is no entry in the corresponding stack with which to compare it. Thus, this step can be thought of as pushing the next entry from the input to the top of the first stack if that entry can legally sit atop each of the two stacks at this stage.*

4. *Finally if neither of those moves are available, push the top entry from the first stack to the second.*

It was noted in [14] that Step 2 of this algorithm is not needed to produce an optimal algorithm. However, we continue to utilize it as outputting entries earlier in the process is useful in constructing a bijection between the DI-sortable permutations and Schröder paths.

Definition 2.2. Let $\mathtt{E}$ represent a push step from the input to the first stack, let $\mathtt{N}$ represent pop/push from the first stack to the second stack, and let $\mathtt{C}$ represent the pop from the second stack to the output. See Figure 3. A *sorting word* of a permutation is a word representing steps that can be taken to sort that permutation using two stacks in series (without restriction). We use $\mathtt{L}_i$ to denote the occurrence of the letter $\mathtt{L}$ in W which corresponds to the movement of symbol i.
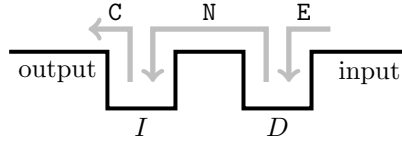
Fig. 3: The operations `E`, `N`, and `C` corresponding to the DI machine from Definition 2.2.

Observation 2.3. A word W is a sorting word of a sortable permutation of length n if and only if the following two conditions are met:

1. The length of W is $3n$ and W contains exactly n of each of the letters `E`, `N`, and `C`.

2. For all $x \in [3n]$, $\#_{\text{E}}(W^x) \geq \#_{\text{N}}(W^x) \geq \#_{\text{C}}(W^x)$. That is, in any prefix of W, the number of `C`s does not exceed the number of `N`s, which does not exceed the number of `E`s.

Note that for every sorting word, there exists a permutation π which it sorts. To find π, apply W to the identity permutation, and let σ be the output (which is not necessarily sorted). Then W sorts $\pi = \sigma^{-1}$.

Definition 2.4. A *DI word* of a permutation π is a sorting word representing steps that can be taken to sort π using the DI machine.

Example 2.5. The permutation 23514 has four DI words: `EEENNNENCCCENCC` (illustrated in Figure 1), `EEENNNENCCECNCC`, `EEENNNENCECCNCC`, and `EEENNNENECCCNCC`.

Definition 2.6. The *algorithmic DI word* (ADI word for short) of a permutation π is the unique word representing steps that will be taken when applying Algorithm 2.1 to sort π by the DI machine.

Example 2.7. Algorithm 2.1 was used in the process illustrated by Figure 1. So the ADI word of 23514 is `EEENNNENCCCENCC`. To use the indices outlined in Definition 2.2, the ADI word of 23514 can also be given as $\text{E}_2\text{E}_3\text{E}_5\text{N}_5\text{N}_3\text{N}_2\text{E}_1\text{N}_1\text{C}_1\text{C}_2\text{C}_3\text{E}_4\text{N}_4\text{C}_4\text{C}_5$.

For the duration of the paper, we say that $\text{X} <_W \text{Y}$ in a word W if `X` appears before `Y` in W. Similarly we say $\text{X} \prec_W \text{Y}$ if `X` appears immediately before `Y` in W.

We begin with some observations about the relative locations of symbols in a DI-word, then follow with a classification of when a sorting word is a DI-word and an ADI-word.

Observation 2.8. Let π be a DI-sortable permutation and let W be a DI word. Let $i, j \in [n]$.

(i) $\text{E}_{\pi_i} <_W \text{E}_{\pi_j}$ if and only if $i < j$; symbols appearing in π earlier enter the first stack earlier.

(ii) $\text{C}_{\pi_i} <_W \text{C}_{\pi_j}$ if and only if $\pi_i < \pi_j$; symbols must exit the second stack in increasing order.

(iii) If $\pi_i < \pi_j$ and $\text{N}_{\pi_i} <_W \text{N}_{\pi_j}$, then $\text{C}_{\pi_i} <_W \text{N}_{\pi_j}$; if a smaller number enters the second stack before a larger number, it must be output before the larger number can enter the second stack – otherwise the increasing condition for the second stack is not satisfied. Hence if $\pi_i < \pi_j$ and $\text{N}_{\pi_i} <_W \text{E}_{\pi_j}$, then $\text{C}_{\pi_i} <_W \text{N}_{\pi_j}$. We highlight this specifically as it will be used in later proofs.

(iv) If $\text{N}_{\pi_i} <_W \text{N}_{\pi_j} <_W \text{C}_{\pi_i}$, then $\text{C}_{\pi_j} <_W \text{C}_{\pi_i}$; if π_j enters the second stack after π_i and before π_i exits, then $\pi_j < \pi_i$ by the increasing condition of the second stack. So π_j must exit before π_i.

(v) If $\mathtt{E}_{\pi_j} <_W \mathtt{C}_{\pi_i} <_W \mathtt{N}_{\pi_j}$ (which implies $\pi_i < \pi_j$), then $\mathtt{N}_{\pi_i} <_W \mathtt{E}_{\pi_j}$; if a smaller number is pushed to the output while a larger number is in the first stack, the smaller number was popped from the first stack before the larger number entered the first stack – otherwise the larger number must be pushed to the second stack before the smaller number is pushed to the second stack.

(vi) If $\pi_i < \pi_j$ and $\mathtt{E}_{\pi_j} <_W \mathtt{E}_{\pi_i}$, then $\mathtt{E}_{\pi_j} <_W \mathtt{N}_{\pi_j} <_W \mathtt{E}_{\pi_i}$; if a larger number enters the first stack before a smaller number does, the larger number must move to the second stack before the smaller number enters the first stack – otherwise the decreasing condition for the first stack is violated.

We now give a classification of when a sorting word is a DI word.

Lemma 2.9. *A sorting word W of a permutation π of length n is a DI word if and only if for all $i \in [n]$, if no Ns appear in W between $\mathtt{N}_{\pi_i}$ and $\mathtt{C}_{\pi_i}$, then $\#_{\mathtt{E}}(W^x) = \#_{\mathtt{N}}(W^x)$ where x satisfies $W(x) = \mathtt{N}_{\pi_i}$; that is at step x, the first stack is empty.*

Alternatively, we can say a sorting word W is a DI word of some permutation π of length n if and only if every entry in π that preceded the entries output by a sequence of Cs must have moved to the second stack (and possibly the output) prior to this exodus.

Proof: Suppose that W is a DI word of a permutation π and for some $i \in [n]$, no Ns appear in W between $\mathtt{N}_{\pi_i}$ and $\mathtt{C}_{\pi_i}$, and $W(x) = \mathtt{N}_{\pi_i}$. Assume that the first stack is nonempty after stage x, and let π_j be a symbol in the first stack. Then $\mathtt{E}_{\pi_j} <_W \mathtt{N}_{\pi_i} <_W \mathtt{N}_{\pi_j}$, and since no Ns appear between $\mathtt{N}_{\pi_i}$ and $\mathtt{C}_{\pi_i}$, it follows that $\mathtt{N}_{\pi_i} <_W \mathtt{C}_{\pi_i} <_W \mathtt{N}_{\pi_j}$. By Observation 2.8 (v), we have that $\mathtt{N}_{\pi_i} <_W \mathtt{E}_{\pi_j}$, which gives a contradiction. So at stage x, the first stack must be empty.

Conversely, suppose W is a sorting word of a permutation π of length n which is not a DI word. Recall that since W represents the sorting of π through two stacks in series, any entries in the second stack will obey the increasing condition at all times. Consequently, the movements corresponding to W when applied to π must cause a violation of the decreasing condition on the first stack. In such a case, we have two entries π_i and π_j where $\pi_i > \pi_j$ and at some point π_i is below π_j in the first stack. That is, $\mathtt{E}_{\pi_i} <_W \mathtt{E}_{\pi_j} <_W \mathtt{N}_{\pi_j} <_W \mathtt{N}_{\pi_i}$.

Since the end result of the sorting is the identity permutation, we must have $\mathtt{C}_{\pi_j} <_W \mathtt{C}_{\pi_i}$. Therefore $\mathtt{E}_{\pi_i} <_W \mathtt{E}_{\pi_j} <_W \mathtt{N}_{\pi_j} <_W \mathtt{C}_{\pi_j} <_W \mathtt{N}_{\pi_i} <_W \mathtt{C}_{\pi_i}$. Let $\mathtt{N}_{\pi_k}$ be the last $\mathtt{N}$ in W to appear before $\mathtt{C}_{\pi_j}$. Since the second stack is increasing and W is a sorting word, $\pi_k \le \pi_j$ and so $\mathtt{N}_{\pi_k} <_W \mathtt{C}_{\pi_k} \le_W \mathtt{C}_{\pi_j}$. By the selection of k, no Ns appear in W between $\mathtt{N}_{\pi_k}$ and $\mathtt{C}_{\pi_k}$. Observe that π_i is in the first stack (and hence the first stack is nonempty) at the step corresponding to $\mathtt{N}_{\pi_k}$. $\qquad\square$

To similarly characterize our ADI words, we first introduce a lemma showing how "output-greedy" Algorithm 2.1 is.

Lemma 2.10. *Let $n \ge 1$ and $i \in [n]$ such that $\pi_i \neq n$. Let π be a DI-sortable permutation of length n with ADI word W. If $\mathtt{C}_{\pi_i} \not\prec_W \mathtt{C}_{\pi_i+1}$, then $\mathtt{C}_{\pi_i} <_W \mathtt{E}_{\pi_i+1}$. That is, if two consecutive values do not exit in two consecutive stages, then the larger value does not enter the machine until after the smaller value exits.*

Proof: Assume by way of contradiction that $\mathtt{C}_{\pi_i} \not\prec_W \mathtt{C}_{\pi_i+1}$ and $\mathtt{E}_{\pi_i+1} <_W \mathtt{C}_{\pi_i}$.

First suppose that $\mathtt{N}_{\pi_i+1} <_W \mathtt{C}_{\pi_i}$ and let x satisfy $W(x) = \mathtt{C}_{\pi_i}$. Then necessarily $\pi_i + 1$ must be immediately below π_i in the second stack after stage $x - 1$. So Step 1 of Algorithm 2.1 applies at stage $x + 1$, giving that $W(x + 1) = \mathtt{C}_{\pi_i+1}$. Hence $\mathtt{C}_{\pi_i} \prec_W \mathtt{C}_{\pi_i+1}$, a contradiction.

Alternatively, suppose $\mathrm{E}_{\pi_i+1} <_W \mathrm{C}_{\pi_i} <_W \mathrm{N}_{\pi_i+1}$, and let x satisfy $W(x) = \mathrm{E}_{\pi_i+1}$. Then $\mathrm{N}_{\pi_i} <_W \mathrm{E}_{\pi_i+1}$ by Observation 2.8 (v). Therefore $\mathrm{N}_{\pi_i} <_W \mathrm{E}_{\pi_i+1} <_W \mathrm{C}_{\pi_i}$, meaning that at stage x, $\pi_i + 1$ is pushed into the first stack while some symbol in the second stack is smaller than $\pi_i + 1$ (namely π_i). Therefore Step 3 of Algorithm 2.1 should not be applied at stage x, which is a contradiction. $\qquad\square$

Example 2.11. See Figure 1 and notice when entries $1, 2, 3$ are output, 4 is still in the input. This can also be seen in Example 2.7, where $\mathrm{C}_1 \prec_W \mathrm{C}_2 \prec_W \mathrm{C}_3 <_W \mathrm{E}_4$.

We now conclude this section with a classification of when a DI word is the ADI word produced by Algorithm 2.1 for some DI-sortable permutation.

Theorem 2.12. *A DI word W of a DI-sortable permutation π of length n is its ADI word if and only if $\mathrm{E}_1 \prec_W \mathrm{N}_1 \prec_W \mathrm{C}_1$ and for each $i \in [n]$ such that $\pi_i \neq 1$, either $\mathrm{C}_{\pi_i-1} \prec_W \mathrm{C}_{\pi_i}$ or $\mathrm{E}_{\pi_i} \prec_W \mathrm{N}_{\pi_i} \prec_W \mathrm{C}_{\pi_i}$. That is, a DI word W of a DI-sortable permutation π is its ADI word if and only if any maximal sequence of consecutive copies of Cs in W is immediately preceded by EN.*

Proof: Suppose W is the ADI word of π and $i \in [n]$. It follows by Algorithm 2.1 that $\mathrm{E}_1 \prec_W \mathrm{N}_1 \prec_W \mathrm{C}_1$, so we may assume going forward that $\pi_i \neq 1$. Additionally any references to *Steps* in the following argument refer to Algorithm 2.1.

Assume that $\mathrm{C}_{\pi_i-1} \not\prec_W \mathrm{C}_{\pi_i}$. By Lemma 2.10, $\mathrm{C}_{\pi_i-1} <_W \mathrm{E}_{\pi_i}$. Define x so that $W(x) = \mathrm{E}_{\pi_i}$. Therefore after stage $x - 1$, π_i is smaller than all symbols in the second stack and larger than all symbols in the first stack – otherwise Step 3 would not be applied at stage x. In fact, all symbols smaller than π_i have been output by stage $x - 1$ since $\mathrm{C}_{\pi_i-1} <_W \mathrm{E}_{\pi_i}$. Therefore after stage x, π_i is the only symbol in the first stack. Hence Step 2 applies at stage $x + 1$, giving $W(x + 1) = \mathrm{N}_{\pi_i}$. Then Step 1 applies at stage $x + 2$ giving that $W(x + 2) = \mathrm{C}_{\pi_i}$ and therefore $\mathrm{E}_{\pi_i} \prec_W \mathrm{N}_{\pi_i} \prec_W \mathrm{C}_{\pi_i}$.

Conversely, suppose the DI word W is not the ADI word of π, and let A denote the ADI word of π. Let $t \in [3n]$ be the largest integer such that $A^{t-1} = W^{t-1}$, that is t is the smallest integer so that $A(t) \neq W(t)$. We consider the state of the DI machine after stage $t-1$. Define $i \in [n]$ so that $\mathrm{C}_{\pi_i-1} \leq_W W(t-1) <_W \mathrm{C}_{\pi_i}$ (and also $\mathrm{C}_{\pi_i-1} \leq_A W(t-1) <_A \mathrm{C}_{\pi_i}$). By convention, we say that C_0 occurs at position 0, and since $W(t) < \mathrm{C}_n$, the index i is well-defined. Therefore all symbols smaller than π_i have been output by stage $t - 1$ and no Cs appear between $W(t - 1)$ and C_{π_i} in W and A. Hence by stage $t - 1$, π_i is either at

- the top of the second stack ($\mathrm{E}_{\pi_i} < \mathrm{N}_{\pi_i} \leq W(t - 1) < \mathrm{C}_{\pi_i}$ in W and A),

- at the bottom of the first stack ($\mathrm{E}_{\pi_i} \leq W(t - 1) < \mathrm{N}_{\pi_i} < \mathrm{C}_{\pi_i}$ in W and A), or

- part of the input yet to enter the stacks ($W(t - 1) < \mathrm{E}_{\pi_i} < \mathrm{N}_{\pi_i} < \mathrm{C}_{\pi_i}$ in W and A).

Suppose first that π_i is located at the top of the second stack by stage $t - 1$. See Figure 4(a). Observe that Step 1 of Algorithm 2.1 outputs π_i at stage t, so $A(t) = \mathrm{C}_{\pi_i}$. Since π_i is smaller than all symbols in the first stack (possibly vacuously), $W(t) \neq \mathrm{N}$. Thus $W(t) = \mathrm{E}$. Since π_i is smaller than any symbol in the first stack and in the remaining input, no N can appear in W between $W(t)$ and C_{π_i}; otherwise the second stack is no longer increasing. Therefore $\mathrm{E} \prec_W \mathrm{C}_{\pi_i}$, and hence neither $\mathrm{C}_{\pi_i-1} \prec_W \mathrm{C}_{\pi_i}$ nor $\mathrm{N}_{\pi_i} \prec_W \mathrm{C}_{\pi_i}$.

Next suppose that π_i is at the bottom of the first stack by stage $t - 1$. Then $W(t - 1) <_A \mathrm{N}_{\pi_i}$ and hence $\mathrm{C}_{\pi_i-1} \leq_A W(t - 1) <_A \mathrm{N}_{\pi_i} <_A \mathrm{C}_{\pi_i}$. So $\mathrm{C}_{\pi_i-1} \not\prec_A \mathrm{C}_{\pi_i}$, and hence $\mathrm{C}_{\pi_i-1} <_A \mathrm{E}_{\pi_i} \leq_A W(t - 1)$ by Lemma 2.10. Since $W^{t-1} = A^{t-1}$, we have that $\mathrm{C}_{\pi_i-1} <_W \mathrm{E}_{\pi_i}$ as well. Let $W(x) = \mathrm{E}_{\pi_i}$ (so $x \leq t - 1$ and thus $A(x) = \mathrm{E}_{\pi_i}$). Then Step 3 applies at stage x, meaning that π_i is larger than all symbols in the first
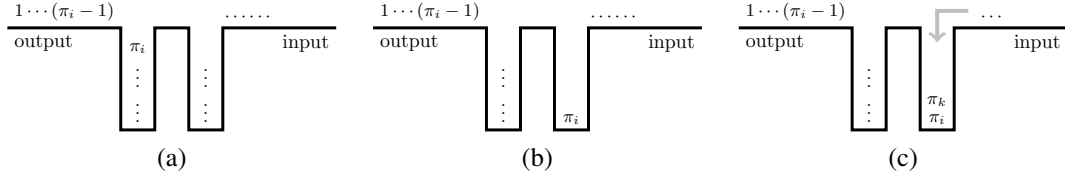
Fig. 4: Some states of the DI machine from the first two cases of the proof of Theorem 2.12.

stack by stage $x - 1$, but since $\mathtt{C}_{\pi_i-1} <_A \mathtt{E}_{\pi_i}$ all symbols smaller than π_i have been output. So π_i is alone in the first stack after stage x. Since π_i is smaller than any symbol (if any) in the second stack, it follows that $A(x) = \mathtt{E}_{\pi_i} \prec_A \mathtt{N}_{\pi_i} \prec_A \mathtt{C}_{\pi_i}$. So $A(x+1) = \mathtt{N}_{\pi_i}$. Since $A(t-1) <_A \mathtt{N}_{\pi_i} = A(x+1)$, $t-1 < x+1$. So $x = t - 1$ and hence $A(t) = \mathtt{N}_{\pi_i}$ and $A(t-1) = W(t-1) = \mathtt{E}_{\pi_i}$. Observe that π_i is alone in the first stack by stage $t - 1$ and $W(t) \neq \mathtt{C}$ (because $\mathtt{C}_{\pi_i-1} \leq_W W(t-1) = \mathtt{E}_{\pi_i} \prec_W W(t) <_W \mathtt{C}_{\pi_i}$), so $W(t) = \mathtt{E}_{\pi_k}$ for some $\pi_k > \pi_i$. Figures 4 (b) and (c) show the states of the DI-machine using W on π after stages $t - 1$ and t, respectively.

Since $\mathtt{C}_{\pi_i-1} \leq_W W(t-1) = \mathtt{E}_{\pi_i}$, no $\mathtt{C}$s appear between $\mathtt{E}_{\pi_i}$ and $\mathtt{C}_{\pi_i}$ in W. Hence $\mathtt{C} \not\prec_W \mathtt{N}_{\pi_i}$ and there are no $\mathtt{C}$s between $\mathtt{N}_{\pi_i}$ and $\mathtt{C}_{\pi_i}$. Furthermore $\mathtt{E} \not\prec_W \mathtt{N}_{\pi_i}$ since $\mathtt{E}_{\pi_i} \not\prec_W \mathtt{N}_{\pi_i}$ and otherwise W is not a sorting word. Therefore $\mathtt{N} \prec_W \mathtt{N}_{\pi_i}$. Furthermore, there are no $\mathtt{N}$s between $\mathtt{N}_{\pi_i}$ and $\mathtt{C}_{\pi_i}$, otherwise Observation 2.8 (iv) is violated. So either $\mathtt{N} \prec_W \mathtt{N}_{\pi_i} \prec_W \mathtt{C}_{\pi_i}$ or $\mathtt{E} \prec_W \mathtt{C}_{\pi_i}$.

Last, we show that π_i must be in one of the stacks by stage $t - 1$, which will conclude the proof. Assume to the contrary that π_i is not in either stack by stage $t-1$. Clearly both $A(t)$ and $W(t)$ are not $\mathtt{C}$ (necessarily $\mathtt{C}_{\pi_i}$), so either $A(t) = \mathtt{N}$ and $W(t) = \mathtt{E}$ or $A(t) = \mathtt{E}$ and $W(t) = \mathtt{N}$. Suppose at stage $t - 1$, π_k is the next symbol to enter the stacks, and if the stacks are nonempty, π_q sits atop the first stack and π_m sits atop the second stack. See Figure 5(a).
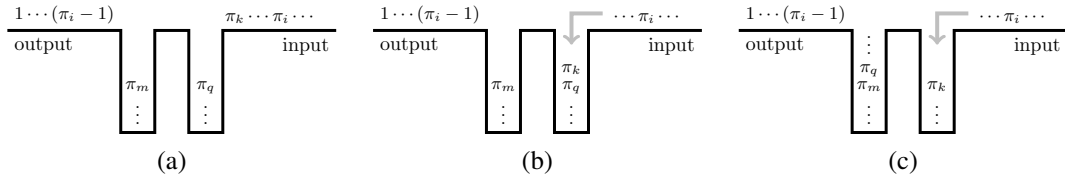

Fig. 5: Some stages of the DI machine from the final cases of the proof of Theorem 2.12.

First assume that $A(t) = \mathtt{N}_{\pi_q}$ and $W(t) = \mathtt{E}_{\pi_k}$ (and therefore the first stack is nonempty by stage $t - 1$). Then either Step 2 or Step 4 of Algorithm 2.1 applies at stage t. However Step 2 cannot be satisfied because π_i is the next symbol to be output, so Step 4 is applied, meaning the conditions of Step 3 are not satisfied. Hence either the second stack is nonempty and $\pi_k > \pi_m$, or $\pi_k < \pi_q$. Since W is a DI word and $W(t) = \mathtt{E}_{\pi_k}$, it follows that $\pi_k > \pi_q$ (and thus $\pi_k \neq \pi_i$), which implies that the second stack must be nonempty and $\pi_k > \pi_m$; hence $\mathtt{N}_{\pi_m} <_W \mathtt{E}_{\pi_k}$. Since $\mathtt{E}_{\pi_k} <_W \mathtt{E}_{\pi_i}$ we have that $W(t) = \mathtt{E}_{\pi_k} <_W \mathtt{N}_{\pi_k} <_W \mathtt{E}_{\pi_i}$ by Observation 2.8 (vi).

So $\mathtt{C}_{\pi_i} <_W \mathtt{C}_{\pi_m}$ (since $\pi_i < \pi_m$), $\mathtt{C}_{\pi_m} <_W \mathtt{N}_{\pi_k}$ (follows from $\mathtt{N}_{\pi_m} <_W \mathtt{E}_{\pi_k}$, $\pi_m < \pi_k$, and Observation 2.8 (iii)), and $\mathtt{N}_{\pi_k} <_W \mathtt{E}_{\pi_i}$ (because $\pi_i < \pi_k$, $\mathtt{E}_{\pi_k} <_W \mathtt{E}_{\pi_i}$, and Observation 2.8 (vi)). Therefore $\mathtt{C}_{\pi_i} <_W \mathtt{E}_{\pi_i}$, a contradiction. Hence we cannot have $A(t) = \mathtt{N}$ and $W(t) = \mathtt{E}$.

Therefore $A(t) = \mathrm{E}_{\pi_k}$ and $W(t) = \mathrm{N}_{\pi_q}$. This implies that $\pi_k > \pi_q$ (and thus $\pi_k \neq \pi_i$) and the first stack is nonempty at stage $t-1$, respectively. This also gives that $W(t) = \mathrm{N}_{\pi_q} <_W \mathrm{E}_{\pi_k}$. Furthermore $\pi_q > \pi_i$ since all other symbols smaller than π_i have been output by stage $t-1$.

So $\mathrm{C}_{\pi_q} <_W \mathrm{N}_{\pi_k}$ because from $\mathrm{N}_{\pi_q} <_W \mathrm{E}_{\pi_k}$, $\pi_q < \pi_k$, and Observation 2.8 (iii). Furthermore $\mathrm{N}_{\pi_k} <_W \mathrm{E}_{\pi_i}$ since $\pi_i < \pi_k$, $\mathrm{E}_{\pi_k} <_W \mathrm{E}_{\pi_i}$, and Observation 2.8 (vi). Since $\pi_i < \pi_q$, it follows that $\mathrm{C}_{\pi_i} <_W \mathrm{C}_{\pi_q}$. So $\mathrm{C}_{\pi_q} <_W \mathrm{N}_{\pi_k} <_W \mathrm{E}_{\pi_i} <_W \mathrm{C}_{\pi_i} <_W \mathrm{C}_{\pi_q}$, which is a contradiction. □

We can now strengthen Theorem 2.12. This result will be used in an argument in Section 4, specifically the proof of Theorem 4.11.

Corollary 2.13. *Consider the ADI word W of a DI-sortable permutation π. Let $x \in [3n-1]$ and $\pi_i = \min\{\pi_k \mid W(x) <_W \mathrm{C}_{\pi_k}\}$. If $\mathrm{E}_{\pi_i} <_W W(x)$, then $W(x) \neq \mathrm{E}$. That is, Algorithm 2.1 will output all available entries in the stacks before another push from the input is allowed.*

Proof: If $W(x) = \mathrm{C}_{\pi_i - 1}$, then clearly $W(x) \neq \mathrm{E}$. If $\mathrm{C}_{\pi_i - 1} <_W W(x) <_W \mathrm{C}_{\pi_i}$, then by Theorem 2.12, $\mathrm{E}_{\pi_i} \prec_W \mathrm{N}_{\pi_i} \prec_W \mathrm{C}_{\pi_i}$. So $W(x) = \mathrm{N}_{\pi_i} \neq \mathrm{E}$. □

3   The Bijection

In this section we give a bijection that takes DI-sortable permutations to their corresponding Schröder paths via their ADI words. Let π be a DI-sortable permutation of length n with corresponding ADI word W. To produce a Schröder path, we use the following constructions. Suppose that W contains k maximal, consecutive substrings consisting only of Cs. Let $\tau = (\tau_1, \tau_2, \ldots, \tau_k)$ be the partition of n such that τ_i is the length of the ith such substring. Define a k-tuple ℓ so that $\ell_1 = \tau_1$ and $\ell_i = \tau_i + \ell_{i-1}$, when $2 \leq i \leq k$. Note that $\ell_k = n$ since it counts all the Cs of W. Finally, define an increasing k-tuple ρ so that for $x \in [3n]$ we have $\mathrm{E} \prec_W W(x) \prec_W \mathrm{C}$ if and only if $W(x) = \mathrm{N}$ and $\#_{\mathrm{N}}(W^x) = \rho_i$. Note that ρ is well-defined by Theorem 2.12. In other words, we have:

- ℓ_i is the number of Cs in the first i maximal subsequences of Cs in W.

- ρ_i is the number of Ns in W before the ith maximal subsequence of Cs. By Theorem 2.12, the $\rho_i{}^{\text{th}}$ N in W is preceded by an E and succeeded by a C.

Example 3.1. Let $\pi = 81736245$. Then $W = \mathrm{EN}$ ENC ENEENN ENCC ENC ENCCCC, which contains $k = 4$ maximal subsequences of consecutive Cs having lengths 1, 2, 1, and 4, respectively. So $\tau = (1, 2, 1, 4)$ and hence $\ell = (1, 3, 4, 8)$, and $\rho = (2, 6, 7, 8)$. The maximal substrings of the form ENC^{τ_i} are given in red.

Using this, we give an algorithm for producing a Schröder path from a DI-sortable permutation.

Algorithm 3.2. *Let π be a DI-sortable permutation of length n.*

1. *Use Algorithm 2.1 to find the ADI word W of π. From W, obtain τ and ℓ.*

2. *Construct a word T by replacing each maximal consecutive substring in W of the form ENC^a with an N. Observe that N is the last letter of T, and $\#_{\mathrm{N}}(T) = n$.*

3. *Construct a word S_{D} from T by replacing the $\ell_i{}^{\text{th}}$ N in T with a D for each $i \in [k]$. Observe that since $\ell_k = n$, the final character in this word is D. Remove this D to produce S.*

Example 3.3. Let $\pi = 81736245$ as given in Example 3.1, in which we give W, τ, and ℓ. We now apply Algorithm 3.2 to π. We first produce $T = \text{ENNENEENNNNN}$ by replacing the maximal consecutive substrings ENC^{τ_i} ($1 \leq i \leq 4$) with an N.

Next, we replace the $\ell_i{}^{\text{th}}$ Ns in T with D (in blue) to produce $S_\text{D} = \text{EDNEDEEDNNND}$, then remove the final D to produce $S = \text{EDNEDEEDNNN}$. See Figure 6.
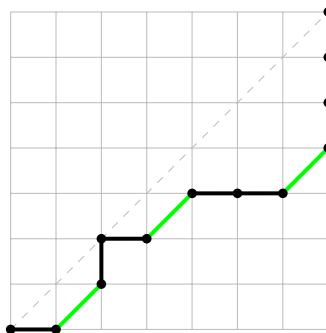

Fig. 6: The Schröder path EDNEDEEDNNN corresponding to $\pi = 81736245$.

We now give an observation, then prove two lemmas which collectively show that Algorithm 3.2 provides a one-to-one correspondence between ADI words and Schröder paths.

Observation 3.4. Let W be an ADI word for a permutation π of length n with k substrings of the form ENC^{τ_i} and k-tuples ρ and ℓ as defined earlier, and let $\ell_0 = 0$. Observe that C_m is the mth occurrence of C in W. Let $i \in [k]$ and $\hat{x}, x \in [3n]$ such that $W(\hat{x}) \prec_W \text{C}_{\ell_{i-1}+1}$ and $W(x) = \text{C}_{\ell_i}$; in other words $W(\hat{x})$ is the N which immediately precedes the ith maximal subsequence of Cs and $W(x)$ is the last C in the ith maximal substring of Cs. See Figure 7. Then, $\ell_i = \#_\text{C}(W^x) \leq \#_\text{N}(W^x) = \#_\text{N}(W^{\hat{x}}) = \rho_i$.

Observe that in Example 3.1, $\ell_i \leq \rho_i$ for each $i \in [4]$.

$$\overbrace{\cdots \text{ENC}_{\ell_{i-1}+1} \ \text{C}_{\ell_{i-1}+2} \ \cdots\cdots \ \text{C}_{\ell_{i-1}+\tau_i-1} \ \text{C}_{\ell_i} \ \text{E} \cdots}^{i^{\text{th}}\text{sequence of Cs}}$$
$$\phantom{\cdots \text{ENC}}\uparrow \phantom{_{\ell_{i-1}+1} \ \text{C}_{\ell_{i-1}+2} \ \cdots\cdots \ \text{C}_{\ell_{i-1}+\tau_i-1} \ \text{C}}\uparrow\phantom{_{\ell_i} \ \text{E} \cdots}$$
$$\text{Location number:} \quad \hat{x} \phantom{\text{ENC}_{\ell_{i-1}+1} \ \text{C}_{\ell_{i-1}+2} \ \cdots\cdots \ \text{C}} x \phantom{\text{E}}$$
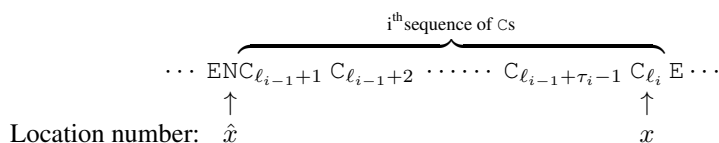
Fig. 7: An illustration of the indexing used for a word W in Observation 3.4.

Lemma 3.5. *The ADI word for a DI-sortable permutation of length $n > 0$ yields a Schröder path from $(0,0)$ to $(n-1, n-1)$ via Algorithm 3.2.*

Proof: Let W be the ADI word for a DI-sortable permutation π of length n. Suppose that W contains k disjoint maximal consecutive substrings of Cs. Let T, S_D, and S be the words produced in Algorithm 3.2. Then the length of T is $2n - k$ with $\#_\text{N}(T) = n$ and $\#_\text{E}(T) = n - k$. Therefore the length of S is $2n - k - 1$ with $\#_\text{N}(S) = \#_\text{E}(S) = n - k$ and $\#_\text{D}(S) = k - 1$. It follows that S corresponds to a path from $(0,0)$ to $(n-1, n-1)$.

Let $x \in [2n - k - 1]$ and suppose that $S(x) = \text{N}$. It is sufficient to show that $\#_\text{E}(S^x) \geq \#_\text{N}(S^x)$. We have $T(x) = \text{N}$. Let $i \in [n]$ such that $\rho_{i-1} < \#_\text{N}(T^x) \leq \rho_i$, with the convention that $\ell_0 = 0, \rho_0 = 0$. Since $\ell_{i-1} \leq \rho_{i-1}$ and in producing S we replace the ℓ_j^{th} N in T with a D whenever $1 \leq j < k$, we have $\#_\text{D}(S^x) \geq i - 1$. Let $\hat{x} \in [3n]$ so that $W(\hat{x}) = \text{N}$ and corresponds to the N at $T(x)$. Then $\#_\text{N}(T^x) = \#_\text{N}(W^{\hat{x}})$. Furthermore $\text{C}_{\ell_{i-1}} <_W W(\hat{x}) <_W \text{C}_{\ell_{i-1}+1}$, and since, in producing T, we remove $i - 1$ instances of E that precede the N at location $\hat{x}$ in W, it follows that $\#_\text{E}(W^{\hat{x}}) = \#_\text{E}(T^x) + (i - 1)$. In addition, observe that $\#_\text{E}(S^x) = \#_\text{E}(T^x)$ and $\#_\text{N}(T^x) = \#_\text{N}(S^x) + \#_\text{D}(S^x)$. So

$$\begin{aligned} \#_\text{E}(S^x) &= \#_\text{E}(T^x) = \#_\text{E}(W^{\hat{x}}) - (i - 1) \\ &\geq \#_\text{N}(W^{\hat{x}}) - (i - 1) = \#_\text{N}(T^x) - (i - 1) = \#_\text{N}(S^x) + \#_\text{D}(S^x) - (i - 1) \geq \#_\text{N}(S^x). \end{aligned}$$

Hence S is a Schröder path from $(0, 0)$ to $(n - 1, n - 1)$. $\qquad\square$

Lemma 3.6. *For every Schröder path S from $(0, 0)$ to $(n - 1, n - 1)$, there is an ADI word W of a DI-sortable permutation of length n such that Algorithm 3.2 applied to W gives S.*

Proof: We begin with a Schröder path from $(0, 0)$ to $(n - 1, n - 1)$ and consider it as a word S. Let $k = \#_\text{D}(S) + 1$. Then $\#_\text{E}(S) = \#_\text{N}(S) = n - k$. Furthermore, we have $\#_\text{E}(S') \geq \#_\text{N}(S')$ for any prefix S' of S, since the Schröder path stays weakly below the main diagonal. Following Algorithm 3.2 backwards, append a D to the end of S to create the "extended" Schröder path S_D.

For each $i \in [k]$, let t_i be the location of the ith D in S_D, and let ℓ_i be the location of the ith D in the substring of S_D consisting of all of the Ns and Ds of S_D. Since the substring of S_D consisting of all of the Ns and Ds of S_D has exactly n letters, observe that $\ell_k = n$ and for convention, we set $\ell_0 = 0$.

Next create T by replacing each D with an N in S_D. Then $\#_\text{N}(T) = n = k + \#_\text{E}(T)$, and observe that if $t_i \leq x < t_{i+1}$ for some $i \in [k - 1]$, then $\#_\text{D}(S_\text{D}^x) = i$ and $\#_\text{N}(S_\text{D}^x) \leq \#_\text{E}(S_\text{D}^x) = \#_\text{E}(T^x)$. So

$$\#_\text{N}(T^x) = \#_\text{N}(S_\text{D}^x) + \#_\text{D}(S_\text{D}^x) \leq \#_\text{E}(T^x) + i.$$

Define $r_i = \min\{x \mid \#_\text{N}(T^x) = \#_\text{E}(T^x) + i\}$ for each $i \in [k]$. In other words, r_i is the minimal length for a prefix of T which has exactly i more Ns than Es. By convention, let $r_0 = 0$.

Observe that $T(r_i) = \text{N}$ and $\#_\text{N}(T^{r_i-1}) = \#_\text{E}(T^{r_i-1}) + i - 1$ for each $i \in [k]$ and that $r_k \leq 2n - k$ since $\#_\text{N}(T^{2n-k}) = \#_\text{E}(T^{2n-k}) + k$. Furthermore, note that $r_i \geq t_i$ for each $i \in [k]$. Indeed,

$$\#_\text{D}(S_\text{D}^{r_i}) = \#_\text{N}(T^{r_i}) - \#_\text{N}(S_\text{D}^{r_i}) = \#_\text{E}(T^{r_i}) + i - \#_\text{N}(S_\text{D}^{r_i}) = \#_\text{E}(S_\text{D}^{r_i}) + i - \#_\text{N}(S_\text{D}^{r_i}) \geq i,$$

since S_D is a Schröder path from $(0, 0)$ to (n, n). In particular, $r_k \geq t_k = 2n - k$ hence $r_k = 2n - k$.

Now create $\hat{T}$ by inserting an E before the k Ns at locations $r_1, r_2, \ldots, r_k$ of T. Let $\hat{r}_i$ be the new location of these Ns in $\hat{T}$, that is $\hat{r}_i = r_i + i$, meaning the inserted Es are at locations $\hat{r}_i - 1$ in $\hat{T}$. Again by convention, let $\hat{r}_0 = 0$. We have $\#_\text{E}(\hat{T}) = n = \#_\text{N}(\hat{T})$ and $\hat{r}_k = 2n$.

We now establish that $\#_\text{N}(\hat{T}^x) \leq \#_\text{E}(\hat{T}^x)$ for each $x \in [2n]$. First suppose $x = \hat{r}_i - 1$ for some $i \in [k]$. Then $x - i = r_i - 1$. Furthermore $\#_\text{N}(\hat{T}^x) = \#_\text{N}(T^{x-i}) = \#_\text{N}(T^{r_i-1}) = \#_\text{E}(T^{r_i-1}) + i - 1$ and

$$\#_\text{E}(\hat{T}^x) = \#_\text{E}(T^{r_i-1}) + i = \#_\text{N}(T^{r_i-1}) + 1 = \#_\text{N}(\hat{T}^x) + 1.$$

Observe that since $\hat{T}(\hat{r}_i) = \text{N}$, we have $\#_\text{E}(\hat{T}^{\hat{r}_i}) = \#_\text{N}(\hat{T}^{\hat{r}_i})$. This will be used later.

Now suppose $\hat{r}_i \leq x < \hat{r}_{i+1} - 1$ where $0 \leq i \leq k-1$. Then $r_i \leq x - i < r_{i+1}$, so $\#_{\text{N}}(\hat{T}^x) = \#_{\text{N}}(T^{x-i})$. So by definition of r_{i+1},

$$\#_{\text{N}}(\hat{T}^x) = \#_{\text{N}}(T^{x-i}) \leq \#_{\text{E}}(T^{x-i}) + i = \#_{\text{E}}(\hat{T}^x).$$

Additionally since $\hat{r}_k = 2n$, we have $\#_{\text{N}}(\hat{T}^{\hat{r}_k}) = n = \#_{\text{E}}(\hat{T}^{\hat{r}_k})$. Thus $\#_{\text{N}}(\hat{T}^x) \leq \#_{\text{E}}(\hat{T}^x)$ for all $x \in [2n]$.

Now, create the partition $\tau = (\tau_1, \tau_2, \ldots, \tau_k)$ of n where $\tau_i = \ell_i - \ell_{i-1}$ for each $i \in [k]$. Next, create W by inserting τ_i copies of C after the Ns in $\hat{T}$ located at $\hat{r}_i$ for each $i \in [k]$. Let $\hat{\hat{r}}_i$ be the new location of these Ns in W, that is $\hat{\hat{r}}_i = \hat{r}_i + \ell_{i-1}$, meaning the inserted Cs are at locations $\hat{\hat{r}}_i + 1, \ldots, \hat{\hat{r}}_i + \tau_i$.

By its creation, it follows from the properties of $\hat{T}$ that $\#_{\text{E}}(W) = \#_{\text{N}}(W) = \#_{\text{C}}(W) = n$ and $\#_{\text{N}}(W^x) \leq \#_{\text{E}}(W^x)$ for all $x \in [3n]$. To show that $\#_{\text{C}}(W^x) \leq \#_{\text{N}}(W^x)$ for all $x \in [3n]$, we need only to show the inequality holds when $x = \hat{\hat{r}}_i + \tau_i$ for each $i \in [k]$. So let $x = \hat{\hat{r}}_i + \tau_i$ for some $i \in [k]$. Recall t_i was defined to be the location of the ith D in S_{D}, and it was established that $t_i \leq r_i$ for all $i \in [k]$. Then

$$\#_{\text{N}}(W^x) = \#_{\text{N}}(\hat{T}^{\hat{r}_i}) = \#_{\text{N}}(T^{r_i}) \geq \#_{\text{N}}(T^{t_i}) = \#_{\text{N}}(S^{t_i}) + \#_{\text{D}}(S^{t_i}) = \ell_i = \#_{\text{C}}(W^x).$$

Therefore $\#_{\text{C}}(W^x) \leq \#_{\text{N}}(W^x) \leq \#_{\text{E}}(W^x)$ for each $x \in [3n]$. So by Observation 2.3, W is a sorting word for a permutation π of length n. Moreover, $\#_{\text{E}}(W^{\hat{r}_i}) = \#_{\text{N}}(W^{\hat{r}_i})$ for each $i \in [k]$ since $\#_{\text{E}}(\hat{T}^{\hat{r}_i}) = \#_{\text{N}}(\hat{T}^{\hat{r}_i})$. From Lemma 2.9, W is a DI word of π.

Finally, one can see by the construction of W that $\text{E}_{\ell_{i-1}+1} \prec_W \text{N}_{\ell_{i-1}+1} \prec_W \text{C}_{\ell_{i-1}+1}$ for each $i \in [k]$ and $\text{C}_{\pi_i-1} \prec_W \text{C}_{\pi_i}$ if $\pi_i \notin \{\ell_{i-1} + 1 \mid i \in [k]\}$. Hence from Theorem 2.12, W is an ADI word for some DI-sortable permutation π. Moreover, applying Algorithm 3.2 to W gives S. $\qquad\square$

By Lemmas 3.5 and 3.6, our main result follows:

Theorem 3.7. *For any $n > 0$, Algorithm 3.2 produces a bijection between the set of ADI words from DI-sortable permutations of length n and Schröder paths from $(0,0)$ to $(n-1, n-1)$.*

Example 3.8. Suppose we start with the Schröder path $S = \text{EDNEDEEDNNN}$ illustrated in Figure 6. We demonstrate the inverse of Algorithm 3.2 described in Lemma 3.6.
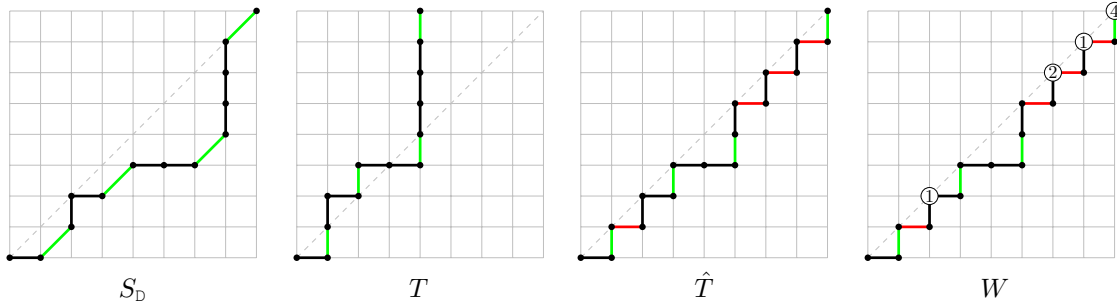

Fig. 8: The process by which the Schröder path in Figure 6 becomes an ADI-word. Circled numbers indicate consecutive copies of C.

We begin by producing $S_{\text{D}} = \text{EDNEDEEDNNND}$ by appending a diagonal (Figure 8(a)). We identify $\ell = (1, 3, 4, 8)$ and $\tau = (1, 2, 1, 4)$ based on the relative locations of the diagonal edges with respect to

the north edges. We construct $T = \texttt{ENNENEENNNNN}$ by converting each diagonal edge to a north edge (Figure 8(b)).

Next we find $r = (3, 10, 11, 12)$ and construct $\hat{T} = \texttt{ENENENEENNENENEN}$ by inserting $\texttt{E}$ before the $\texttt{N}$ at each of the locations in T specified by r. Note that $\hat{r} = (4, 12, 14, 16)$, and with this we construct $W = \texttt{ENENCENEENNENCCENCENCCCC}$. Observe that W is the ADI word of 81736245, and Algorithm 3.2 applied to W gives S (see Example 3.3).

4   Permutation properties observable from their Schröder paths

While the bijection obtained from Algorithm 3.2 can be used to show exactly which Schröder path corresponds to a given DI-sortable permutation, there are several properties that translate prominently between permutation and path.

Definition 4.1. A *right-to-left minimum* of a permutation is an element that is the least element seen thus far when reading the permutation from right to left.

Example 4.2. The permutation $\pi = 81736245$ has four right-to-left minima, namely $1, 2, 4, 5$.

Lemma 4.3. *If W is the ADI word of a DI-sortable permutation π, then $\texttt{E}_{\pi_i} \prec_W \texttt{N}_{\pi_i} \prec_W \texttt{C}_{\pi_i}$ if and only if π_i is a right-to-left minimum. Furthermore, the number of right-to-left minima of a DI-sortable permutation π is one more than the number of diagonal steps in its corresponding Schröder path.*

Proof: Suppose π_i is a right-to-left minimum of π with ADI word W. There must be some $k \in [n]$ that satisfies $\texttt{E}_{\pi_k} \prec_W \texttt{N}_{\pi_k} \prec_W \texttt{C}_{\pi_k} \prec_W \cdots \prec_W \texttt{C}_{\pi_i}$ by Theorem 2.12. Assume by way of contradiction that $k \neq i$. Then $\texttt{E}_{\pi_i} <_W \texttt{E}_{\pi_k}$ and $\texttt{C}_{\pi_k} <_W \texttt{C}_{\pi_i}$. Hence $i < k$ and $\pi_k < \pi_i$, which is a contradiction to π_i being a right-to-left minimum. So $\texttt{E}_{\pi_i} \prec_W \texttt{N}_{\pi_i} \prec_W \texttt{C}_{\pi_i}$.

Now suppose that $\texttt{E}_{\pi_i} \prec_W \texttt{N}_{\pi_i} \prec_W \texttt{C}_{\pi_i}$ for some $i \in [n]$ and let $j \in [n]$ such that $i < j$. Then $\texttt{E}_{\pi_i} <_W \texttt{E}_{\pi_j}$. Therefore $\texttt{C}_{\pi_i} <_W \texttt{E}_{\pi_j}$, so $\texttt{C}_{\pi_i} <_W \texttt{C}_{\pi_j}$. Hence $\pi_i < \pi_j$. So π_i is a right-to-left minimum.

Hence there is a one-to-one correspondence between consecutive sequences $\texttt{ENC}$ in W and right-to-left minima in π. And as seen in Algorithm 3.2, these sequences give rise to one fewer diagonal in the corresponding Schröder path of π. $\qquad\square$

Lemma 4.3 leads to a visual proof to the Corollary 4.4.

Corollary 4.4. *The DI-sortable permutations with one right-to-left minimum, that is the DI-sortable permutations that end in 1, are enumerated by the Catalan numbers. In particular, the number of DI-sortable permutations of length $n + 1$ ending in 1 is $C_n = \dfrac{1}{n+1}\dbinom{2n}{n}$.*

Proof: The Schröder paths from $(0, 0)$ to (n, n) without diagonal steps are Dyck paths, known to be enumerated by the Catalan numbers.

Alternatively, recall that the basis for our DI-sortable permutations is $\{3142, 3241\}$. Thus, the DI-sortable permutations of length $n + 1$ ending in 1 are exactly the permutations whose first n entries must avoid 213, which are known to be enumerated by Catalan numbers, C_n [8]. $\qquad\square$

For a given diagonal in a Schröder path S and prefix S' of S terminating after the given diagonal, the we say the *height* of the diagonal is $\#_\texttt{N}(S') + \#_\texttt{D}(S')$. In other words, its height is the y-coordinate of the point in the path where the diagonal terminates.

Example 4.5. Let $\pi = 81736245$ with corresponding Schröder path S given in Figure 6. Then the height of its diagonals are $1, 3,$ and 4.

Theorem 4.6. *The heights of the diagonals determine the values of the right-to-left minima. In particular, if the heights of the diagonals in the Schröder path corresponding to the permutation π are $h_1, h_2, \ldots, h_k$, the right-to-left minima of π are $1, h_1 + 1, h_2 + 1, \ldots, h_k + 1$.*

Proof: Suppose that S has k diagonals (and hence π has $k + 1$ right-to-left minima), and let W be the ADI word of π. Notice since ℓ_i counts the Cs in the first i maximal substrings of Cs, we have ℓ_i counts $\mathtt{C}_1, \mathtt{C}_2, \ldots, \mathtt{C}_{\ell_i}$. Thus $\mathtt{E}_a \prec_W \mathtt{N}_a \prec_W \mathtt{C}_a$ if and only if $a = \ell_i + 1$ for some i where $0 \leq i \leq k$. (Recall $\ell_0 = 0$ by convention.) From Lemma 4.3 the right-to-left minima are exactly $\ell_i + 1$ for $0 \leq i \leq k$.

Let $x_1, x_2, \ldots, x_k$ be locations in S so that x_i is the location of the ith D; that is $S(x_i) = \mathtt{D}$ and $\#_{\mathtt{D}}(S^{x_i}) = i$. From Step 3 in Algorithm 3.2, we observe that $\#_{\mathtt{N}}(S^{x_i}) + \#_{\mathtt{D}}(S^{x_i}) = \ell_i$ is the height of the ith diagonal for each $i \in [k]$. So the heights of the diagonals of S are one less than the right-to-left minima of π (excluding 1). $\qquad\square$

Definition 4.7. An *interval* of a permutation π is a consecutive subsequence of π that contains consecutive values.

Example 4.8. The permutation $\sigma = 685712943$ contains intervals $6857, 12, 9, 43$.

Definition 4.9. A permutation π is said to be *plus-decomposable* if π is the concatenation of two non-empty intervals ω and τ' where the values of ω are less than those of τ'. Further, if we rescale the entries of τ' by subtracting the length of ω from each entry of τ' to get a permutation τ, we denote $\pi = \omega \oplus \tau$. If a permutation is not plus-decomposable, we say the permutation is *plus-indecomposable*.

Example 4.10. The permutation $\pi = 43126758$ is plus-decomposable and can be written as $\pi = 4312 \oplus 231 \oplus 1$. The permutation $\sigma = 685712943$ is plus-indecomposable.

Theorem 4.11. *The Schröder path corresponding to a plus-decomposable DI-sortable permutation $\pi = \omega \oplus \tau$ is the Schröder path for ω followed by a diagonal step (on the main diagonal) followed by the Schröder path for τ. Furthermore, this means the Schröder paths with diagonal steps on the main diagonal correspond exactly to the plus-decomposable DI-sortable permutations.*

Proof: Consider the plus-decomposable permutation $\pi = \omega \oplus \tau$ and let W be the ADI word of π. Let $x \in [3n - 1]$ satisfy $W(x) = \mathtt{E}_{\pi_k}$ with $k = |\omega| + 1$ and $\pi_i = \min\{\pi_\ell \mid W(x) <_W \mathtt{C}_{\pi_\ell}\}$. Then by Corollary 2.13, we have $\mathtt{E}_{\pi_k} = W(x) \leq_W \mathtt{E}_{\pi_i}$, so $k \leq i$. Since $\pi = \omega \oplus \tau$ and $i \geq k = |\omega| + 1$, we have $\pi_i > |\omega|$. Hence by the definition of π_i, we have $\mathtt{C}_{|\omega|} <_W W(x) = \mathtt{E}_{\pi_k}$. Thus we know Algorithm 2.1 will force the exit of all the entries of ω before any entry of τ' (entries of π corresponding to τ) enters the stacks. Therefore the ADI word for π will be the ADI word for ω followed by the ADI word for τ.

When Algorithm 3.2 acts on the ADI word for π, it will convert the first portion of the ADI word to the Schröder path for ω, but followed by D as this D is no longer the last D of the word. At the same time, it converts the remaining portion of the ADI word to the Schröder path for τ. $\qquad\square$

Example 4.12. In Figure 9, we see that this Schröder path for $\pi = 43126758 = 4312 \oplus 231 \oplus 1$ is the concatenation of the Schröder paths for $4312, 231,$ and 1 (a degenerate path), connected by diagonal edges.
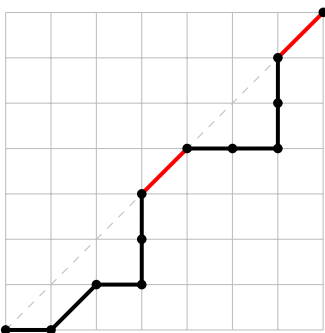
Fig. 9: The plus-decomposition of the Schröder path of 43126758 as given in Example 4.12.

Aguiar and Moreira [1] showed that the number of Schröder paths from $(0,0)$ to (n,n) that do not have diagonal steps on the main diagonal equals the number of Schröder paths from $(0,0)$ to (n,n) that do have diagonal steps on the main diagonal as a particular case of a larger result. That is, they showed the number of Schröder paths that do not have diagonal steps on the main diagonal are enumerated by the small Schröder numbers, which gives us the following corollary.

Corollary 4.13. *The plus-decomposable DI-sortable permutations are in bijection with the plus-indecomposable DI-sortable permutations. In particular, both classes are enumerated by the small Schröder numbers.*

We conclude with a few open questions:

1. Are there any other properties of the DI-sortable permutations that are easily seen by looking at the corresponding Schröder paths?

2. Is there another bijection between the DI-sortable permutations and the Schröder paths that conserves more or different properties of the DI-sortable permutations in the Schröder paths?

Acknowledgements

References

[1] AGUIAR, M., AND MOREIRA, W. Combinatorics of the free Baxter algebra. *Electron. J. Combin. 13*, 1 (2006), Research Paper 17, 38.

[2] ALBERT, M., ATKINSON, M., AND LINTON, S. Permutations generated by stacks and deques. *Ann. Comb. 14*, 1 (2010), 3–16.

[3] ATKINSON, M. D., MURPHY, M. M., AND RUŠKUC, N. Sorting with two ordered stacks in series. *Theoret. Comput. Sci. 289*, 1 (2002), 205–223.

[4] BANDLOW, J., EGGE, E. S., AND KILLPATRICK, K. A weight-preserving bijection between Schröder paths and Schröder permutations. *Ann. Comb. 6*, 3-4 (2002), 235–248.

[5] BÓNA, M. Exact enumeration of 1342-avoiding permutations: a close link with labeled trees and planar maps. *J. Combin. Theory Ser. A 80*, 2 (1997), 257–272.

[6] BÓNA, M. A survey of stack-sorting disciplines. *Electron. J. Combin. 9*, 2 (2002/03), Article 1, 16. Permutation patterns (Otago, 2003).

[7] FERRARI, L. *Permutation classes, sorting algorithms, and lattice paths*. PhD thesis, Università di Bologna, 2013.

[8] KNUTH, D. E. *The art of computer programming. Vol. 1: Fundamental algorithms*. Second printing. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont, 1969.

[9] KREMER, D. Permutations with forbidden subsequences and a generalized Schröder number. *Discrete Math. 218*, 1-3 (2000), 121–130.

[10] KREMER, D. Postscript: "Permutations with forbidden subsequences and a generalized Schröder number" [Discrete Math. **218** (2000), no. 1-3, 121–130; MR1754331 (2001a:05005)]. *Discrete Math. 270*, 1-3 (2003), 333–334.

[11] MURPHY, M. M. *Restricted Permutations, Antichains, Atomic Classes, and Stack Sorting*. PhD thesis, University of St Andrews, 2002.

[12] ROGERS, D. G., AND SHAPIRO, L. W. Some correspondences involving the Schröder numbers and relations. In *Combinatorial mathematics (Proc. Internat. Conf. Combinatorial Theory, Australian Nat. Univ., Canberra, 1977)*, vol. 686 of *Lecture Notes in Math.* Springer, Berlin, 1978, pp. 267–274.

[13] SCHRÖDER, E. Vier kombinatorische probleme. *Z. Math. Phys. 15* (1870), 361–376.

[14] SMITH, R. Two stacks in series: a decreasing stack followed by an increasing stack. *Ann. Comb. 18*, 2 (2014), 359–363.

[15] STANLEY, R. P. Hipparchus, Plutarch, Schröder, and Hough. *Amer. Math. Monthly 104*, 4 (1997), 344–350.