Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 349 (2005) 309–317
Theoretical Computer Science
www.elsevier.com/locate/tcs

Permuting machines and priority queues

R.E.L. Aldred[a], M.D. Atkinson[b,*], H.P. van Ditmarsch[b], C.C. Handley[b], D.A. Holton[a], D.J. McCaughan[a]

[a] *Department of Computer Science, University of Otago, New Zealand*
[b] *Department of Mathematics and Statistics, University of Otago, New Zealand*

Received 9 March 2004; received in revised form 4 April 2005; accepted 24 July 2005

Communicated by M. Jerrum

Abstract

Machines whose sole function is to re-order their input data are considered. Every such machine defines a set of allowable input–output pairs of permutations. These sets are studied in terms of the minimal disallowed pairs (the *basis*). Some allowable sets with small bases are considered including the one defined by a priority queue machine. For more complex machines defined by two or more priority queues in series or parallel, the basis is proved to be infinite.

Keywords: Permutation; Pattern; Priority queue; Basis

1. Introduction

The question of which permutations can be sorted by a stack was posed and answered by Knuth in [9]. Since then the same question has been studied for various generalisations of a stack and the survey [6] lists many newer results. A wider context for these questions and others was defined in [1] where permuting machines were introduced. These are (non-deterministic) machines that receive an input stream of distinct tokens and transform it into an output stream that is a permutation of the input stream. Two properties are usually demanded of such machines:

1. Suppose the machine is able to transform an input α into an output β and let α' be a subsequence of α that becomes rearranged as the subsequence β' of β. Then the machine must be able to transform α' (if presented as an input sequence in its own right) into β'.
2. The names (values) of the input items are unimportant. For example, the possible behaviours of the machine on the input 3, 1, 4, 2 are the same as on the input 1, 2, 3, 4 (e.g. if it can reverse one input it can reverse the other).

Property 2 is normally exploited by renaming the items so that the input is $1, 2, \ldots, n$ in which case we think of the machine as a permutation generator, or by renaming the items so that the output is $1, 2, \ldots, n$ in which case we think of the machine as a sorting machine.

Property 1 ties the theory of permuting machines to the notion of subpermutations: we say a permutation π is a *subpermutation* of a permutation σ if σ has a subsequence whose values are ordered relatively the same as π. The

* Corresponding author. Tel.: +64 3 479 8538; fax: +64 3 479 8529.
E-mail address: mike@cs.otago.ac.nz (M.D. Atkinson).

0304-3975/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2005.07.039

combinatorial properties of the subpermutation relation have been studied very intensively in the last decade. A sample of recent work can be found in [7].

Together, properties 1 and 2 tell us that the set of permutations that are sortable by a permuting machine is closed under forming subpermutations. That sets the stage for investigating two questions: how many sortable permutations of each length are there, and what are the minimal unsortable permutations?

The starting point of this paper is that there are some interesting machines where the second assumption does not hold. For example, a *priority queue* is a permuting machine which allows items to be input and output except that, when an item is output, it is the *smallest* of those currently in the machine. It is easily seen that the first assumption holds. However, since the output rule depends on the value of the output item the second assumption does not hold. For example, the possible output streams when the input stream is 312 are 123, 132 and 312 whereas, for the input stream 213, the possible output streams are 123 and 213. Nevertheless, although a priority queue produces different behaviours depending on the input values, two input streams with the same *relative* values are not treated differently. That suggests a weaker definition of a permuting machine in which the second property is replaced by the property:

3. Only *relative* order matters. More precisely, suppose σ, σ' are two sequences related by $\sigma\pi = \sigma'$ where π is an order preserving bijection from the values in the sequence σ to those in σ'. Let T, T' be the sets of possible outputs if the machine is presented with the inputs σ, σ', respectively. Then $T\pi = T'$.

For example, a priority queue can transform the input 231 into any of 123, 213 and 231; therefore, it can transform the input 472 into any of 247, 427 and 472.

From now on we assume only properties 1 and 3. Although we cannot reduce the study of these machines to questions about the subpermutation relation progress can be made nevertheless. We define a pair of permutations (α, β) to be *allowable* for a machine if it is possible for the machine to produce β as output when given α as input (otherwise (α, β) is *disallowable*). For example, a machine that can transform 42513 into 51324 can transform 453 into 534 (property 1) and therefore can transform 231 into 312 (property 3). In other words knowing that (42513, 51324) is allowable we can infer that (231, 312) is also allowable. From this example we see that, rather than studying subpermutations, we need to study *pairs* of permutations and a more complicated form of the subpermutation idea that we formalise as follows:

Definition 1. A pair of permutations (α, β) is *involved* in another pair (σ, τ) if σ and τ have subsequences σ' and τ' *on the same set of values* such that α is order isomorphic to σ' and β is order isomorphic to τ'. This condition is written $(\alpha, \beta) \preccurlyeq (\sigma, \tau)$.

These definitions have been set up so that we have the following obvious result.

Proposition 1. *The set of allowable pairs of permutations for a permuting machine is closed downwards under the involvement relation.*

We shall occasionally require a more algebraic formulation of involvement.

Lemma 2. *Let α, β be permutations of $[m]$ and σ, τ permutations of $[n]$. Then $(\alpha, \beta) \preccurlyeq (\sigma, \tau)$ if and only if there exist monotonic increasing maps $\lambda, \mu, \nu : [m] \to [n]$ satisfying*

$$\alpha\lambda = \mu\sigma,$$
$$\beta\lambda = \nu\tau.$$

In these equations maps are composed from left to right.

Proof. Suppose the equations hold. Then $\alpha\lambda, \beta\lambda$ are permutations (rearrangements) of $[m]\lambda$ order isomorphic to α, β, respectively, while $\mu\sigma, \nu\tau$ are subsequences of σ, τ, respectively. Since the equations hold $(\alpha, \beta) \preccurlyeq (\sigma, \tau)$. The converse follows by reversing the argument. □

Example 1. $(3142, 4213) \preccurlyeq (16527843, 35841276)$. We can see this either by spotting the subsequences (6284, 8426) of the second pair or by using the monotonic maps $\lambda : \{1, 2, 3, 4\} \to \{2, 4, 6, 8\}$; $\mu : \{1, 2, 3, 4\} \to \{2, 4, 6, 7\}$; $\nu : \{1, 2, 3, 4\} \to \{3, 4, 6, 8\}$.

It is trivial to check that involvement is a partial order on pairs. The theory of subpermutations suggests how it should be studied. In particular, we define a *closed* class of pairs as one that is closed downwards under the involvement relation. Thus, Proposition 1 says that the set of allowable pairs for a permuting machine is a closed class.

Furthermore, for every closed class X we may define its *basis* to be the set $B(X)$ of minimal pairs not in the class. Then, obviously, from $B(X)$ we can reconstruct X itself since

$$X = \{(\sigma, \tau) \mid (\alpha, \beta) \not\preceq (\sigma, \tau) \text{ for all } (\alpha, \beta) \in B(X)\}.$$

Example 2. Consider the set of pairs $\{(12, 12), (21, 21)\}$. This is the basis of the set of pairs

$$\{(\sigma, \tau) \mid \tau \text{ is the reverse of } \sigma\}.$$

There are many avenues that may now be explored. For example, if we are given a closed class X (possibly arising from a permuting machine, or possibly defined by its basis) we can ask

1. Is there an efficient algorithm to decide whether a pair (σ, τ) lies in X? Is this decision question in the class P of problems soluble in polynomial time?
2. Does X have a finite basis? (If so, its membership problem would lie in P.)
3. How many pairs of length n lie in X?
4. Given a permutation σ, how many permutations τ are there with $(\sigma, \tau) \in X$? There is an obvious dual question also.

The first three of these questions all have analogues in the theory of subpermutations. Pursuing the analogy a little further we can define, for every pair $(a_1a_2 \ldots a_n, b_1b_2 \ldots b_n)$ of permutations, a $(0-1)$ $n \times n \times n$ tensor τ_{ijk} where

$$\tau_{ijk} = 1 \quad \text{if and only if } j = a_i \quad \text{and} \quad k = b_i.$$

This tensor is the analogue of the matrix of a permutation and describes a pattern of n points in 3-space just as a permutation does in two dimensions. With this viewpoint involvement of pairs corresponds to containment of one spatial layout within another.

The remainder of this paper is laid out as follows. In Section 2 we develop some general results about closed classes that we rely on in the rest of the paper. In Section 3 we briefly look at some closed classes defined by simple bases; it will be seen that the questions are significantly harder than their subpermutation counterparts. Then, in Section 4, we return to priority queues. We study permuting machines formed by connecting several priority queues in series or in parallel and the closed classes they define. We observe that two priority queues in parallel define a class with an infinite basis but mainly we focus on series connections. In the case of one or two priority queues in series we give an efficient solution to the membership question. We prove that the class associated with a single priority queue is finitely based but that the classes associated with two or more priority queues in series are not finitely based. Indeed we also prove that, even when the priority queues are severely restricted in the number of items they can store, the basis is infinite. The final section has some suggestions for follow-up work.

2. General results

Every set of pairs of permutations (closed or not) defines a binary relation on the set of permutations. Two binary relations $\mathcal{X}, \mathcal{Y}$ may be composed in the usual way:

$$\mathcal{X} \circ \mathcal{Y} = \{(\sigma, \tau) \mid \text{ there exists } \rho \text{ such that } (\sigma, \rho) \in \mathcal{X} \text{ and } (\rho, \tau) \in \mathcal{Y}\}.$$

Composition of relations is associative and so the set of all binary relations is a semigroup under composition.

Proposition 3. *The set of closed classes is a subsemigroup of the semigroup of all binary relations on permutations.*

Proof. We have to prove that, if $\mathcal{X}, \mathcal{Y}$ are closed so also is $\mathcal{X} \circ \mathcal{Y}$. Let $(\sigma, \tau) \in \mathcal{X} \circ \mathcal{Y}$ by virtue of a permutation ρ with $(\sigma, \rho) \in \mathcal{X}$ and $(\rho, \tau) \in \mathcal{Y}$. Suppose $(\alpha, \beta) \preceq (\sigma, \tau)$. By Lemma 2 we have monotonic increasing maps λ, μ, ν such that $\alpha\lambda = \mu\sigma$ and $\beta\lambda = \nu\tau$. The first of these equations expresses that $\alpha\lambda$ is a certain sequence order isomorphic to α

via the order isomorphism λ and that it occurs at a particular set of positions within σ determined by μ. Now the set of values in $\alpha\lambda$ occurs somewhere within ρ at a set of positions defined by a monotonic increasing map ω but permuted as a sequence order isomorphic to a permutation γ. In other words we have an equation $\gamma\lambda = \omega\rho$.

From the equations $\alpha\lambda = \mu\sigma$ and $\gamma\lambda = \omega\rho$ we see that $(\alpha, \gamma) \preccurlyeq (\sigma, \rho)$ and from the equations $\gamma\lambda = \omega\rho$ and $\beta\lambda = \nu\tau$ that $(\gamma, \beta) \preccurlyeq (\rho, \tau)$. The closure of $\mathcal{X}$ and $\mathcal{Y}$ tells us that $(\alpha, \gamma) \in \mathcal{X}$ and $(\gamma, \beta) \in \mathcal{Y}$. Therefore, $(\alpha, \beta) \in \mathcal{X} \circ \mathcal{Y}$ proving that $\mathcal{X} \circ \mathcal{Y}$ is closed. □

This result has an interpretation for permuting machines. For, if $\mathcal{X}_1, \mathcal{X}_2$ are the allowable sets for the machines M_1, M_2, then $\mathcal{X}_1 \circ \mathcal{X}_2$ is the allowable set for the serial composition of M_1 and M_2 (in which the output of M_1 is fed as input to M_2).

The subpermutation relation has 8 symmetry operations (generated by reversal, complementation, and inversion). For involvement we have a richer set of symmetries.

Lemma 4. *If* $(\alpha, \beta) \preccurlyeq (\sigma, \tau)$ *then*

1. $(\alpha^R, \beta) \preccurlyeq (\sigma^R, \tau)$ (*reverse first components*).
2. $(\alpha, \beta^R) \preccurlyeq (\sigma, \tau^R)$ (*reverse second components*).
3. $(\alpha^C, \beta^C) \preccurlyeq (\sigma^C, \tau^C)$ (*complement both components—in other words, for each k, replace the kth smallest symbol in both permutations by the kth largest*).
4. $(\beta, \alpha) \preccurlyeq (\tau, \sigma)$ (*swap components*).
5. $(\beta\alpha^{-1}, \alpha^{-1}) \preccurlyeq (\tau\sigma^{-1}, \sigma^{-1})$.

Proof. The first four of these are readily checked. For the last, let λ, μ, ν be the monotonic maps guaranteed by Lemma 2. Then from $\alpha\lambda = \mu\sigma$ and $\beta\lambda = \nu\tau$ we have $\beta\alpha^{-1}\mu = \nu\tau\sigma^{-1}$ and $\alpha^{-1}\mu = \lambda\sigma^{-1}$ giving the result. □

These symmetries and their compositions give, in all, 48 symmetries which is considerably more than the 8 symmetries of the subpermutation relation. On the other hand, these symmetries act on pairs rather than on single permutations. Although we shall not need this fact we note in passing that the symmetries are realised by the 48 symmetries of the cube defined by the corresponding tensors.

3. Small bases

The 48 symmetries arising out of Lemma 4 allow us to begin the study of closed classes of pairs with a small basis (just as Simion and Schmidt [11] did for the subpermutation relation). It is straightforward to verify the following:

Proposition 5. *Every pair of permutations of length* 2 *is equivalent to* (12, 21). *Every pair of permutations of length* 3 *is equivalent to one of*

(123, 123), (123, 132), (132, 213).

Let $\mathcal{W}$ be the class of pairs that avoid (12, 21). Pairs $(\alpha, \beta) \in \mathcal{W}$ are those for which, whenever a pair of elements a, b with $a < b$ occurs in β in the order $b..a$ then they occur in α also in the order $b..a$. In other words, the set of inversions of α contains the set of inversions of β. In other words β is below α in the weak order on permutations [5]. So even this simple basis defines a complex class. Indeed, so far as we are aware, no enumeration of the weakly ordered pairs of length n has been found (see sequence A007767 in [12]). When we ask about any class that avoids a pair of length 3 we obviously have an even harder question.

Since classes defined by a single basis pair seem to be out of reach we shall, instead, look at classes with two basis pairs of lengths 2 and 3, respectively. Again, it is straightforward to verify

Proposition 6. *Under the symmetry operations defined above every two pairs of permutations of lengths* 2 *and* 3 (*and incomparable under involvement*) *are equivalent to one of*

1. {(12, 21), (123, 123)}.

2. {(12, 21), (231, 213)}.
3. {(12, 21), (321, 132)}.

The reason for requiring the two pairs to be incomparable is, of course, so that they genuinely are the basis for the class of permutations that avoids them (if the shorter permutation was involved in the longer then we might as well discard the longer).

We shall denote by $\mathcal{A}, \mathcal{B}, \mathcal{P}$ the three classes defined by taking the three sets (respectively) in this proposition as basis. In the next two lemmas we shall see that $\mathcal{A}, \mathcal{B}$ define relations that are partial orders on permutations: in other words $\mathcal{A}^2 = \mathcal{A}$ and $\mathcal{B}^2 = B$. This is not true for the class $\mathcal{P}$ whose study we defer until Section 4 since it defines the priority queue class.

Lemma 7. *$\mathcal{A}$ is the set of pairs $(\alpha, \beta) \in \mathcal{W}$ such that 123 is not a subpermutation of α. In particular $\mathcal{A}$ is a partial order on permutations.*

Proof. If $(\alpha, \beta) \in \mathcal{A}$ then certainly $(\alpha, \beta) \in \mathcal{W}$. Furthermore, no three values of α can occur in increasing order for, by the avoidance of (12, 21), these would occur in increasing order in β and this contradicts the avoided pair (123, 123). The converse is obvious.

To prove that $\mathcal{A}$ is a partial order we must verify that it is transitive. However, if $(\alpha, \beta) \in \mathcal{A}$ and $(\beta, \gamma) \in \mathcal{A}$ then, as $\mathcal{A} \subseteq \mathcal{W}$, we have $(\alpha, \gamma) \in \mathcal{W}$. But 123 is not a subpermutation of α and so $(\alpha, \gamma) \in \mathcal{A}$ as required. □

Lemma 8. *The binary relation $\mathcal{B}$ is a self-dual partial order on permutations.*

Proof. To check the transitive property let (α, β) and (β, γ) be pairs that each avoid {(12, 21), (231, 213)}. Because $\mathcal{W}$ is a partial order we know that there cannot be two values $x < y$ that occur in the order $x..y$ in α and in the order $y..x$ in γ.

Suppose next that there are 3 values $x < y < z$ that occur in α in the order $y..z..x$ and in γ in the order $y..x..z$. In β the values y, z must occur in the order $y..z$ and so in β we have either a subsequence xyz, yxz or yzx. The first causes (β, γ) to involve (12, 21), the second causes (α, β) to involve (231, 213), and the third causes (β, γ) to involve (231, 213).

It is easily checked that $(\alpha, \beta) \in \mathcal{B}$ if and only if $(\beta^C, \alpha^C) \in \mathcal{B}$ and so $\mathcal{B}$ is self-dual. □

Although we have been unable to prove any enumeration results for A and B we have computed the number of pairs of small length.

Length	1	2	3	4	5	6	7	8
$\mathcal{A}$	1	3	16	124	1262	15898	238572	4152172
$\mathcal{B}$	1	3	16	122	1188	13844	185448	2781348

4. Priority queues

The class $\mathcal{P}$ is easier to study than either of $\mathcal{A}$ or $\mathcal{B}$ because of the following result.

Proposition 9. *The pairs of $\mathcal{P}$ are precisely the input–output pairs allowable for a priority queue.*

Proof. Suppose that (α, β) is a priority queue input–output pair. Then it must avoid each of (12, 21) and (321, 132) for these are easily seen to be disallowable input–output pairs.

For the converse we shall use induction on the length of permutations. Let (σ, τ) be a pair of permutations of length n that avoids both (12, 21) and (321, 132). We wish to show that a priority queue can transform σ into τ. Put $\sigma = \sigma_1 n \sigma_2$ and $\tau = \tau_1 n \tau_2$. We have

1. All symbols of σ_1 occur in τ_1. For, if we had $x \in \sigma_1$ and $x \in \tau_2$, then (σ, τ) would contain the pair (xn, nx) which is order-isomorphic to (12, 21).

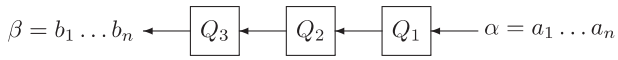

Fig. 1. Three priority queues in series.

2. The symbols of τ_1 that do not occur in σ_1 all occur in some initial segment of σ_2. If this were not true we could find symbols $s_i, s_j \in \sigma_2$ with $i < j$, $s_i \in \tau_2$ and $s_j \in \tau_1$. Then, either $s_i < s_j$ and $(s_i s_j, s_j s_i)$ (isomorphic to $(12, 21)$ occurs in (σ, τ), or $s_i > s_j$ and $(ns_i s_j, s_j n s_i)$ (isomorphic to $(321, 132)$) occurs in (σ, τ).

Now, by induction, there is a sequence of priority queue operations that can transform $\sigma_1\sigma_2$ into $\tau_1\tau_2$. Because of the last two statements we can assume that the priority queue is empty at the point that τ_1 has just been output. Therefore, we can modify this sequence of priority queue operations to incorporate the value n occurring between σ_1 and σ_2 in the input and between τ_1 and τ_2 in the output. We simply insert it in the priority queue when we have to (its presence in the priority queue does not disturb any output operations because n is maximal); and at the point when we would like to output it we can do so because it is then the only item in the priority queue. □

Now we can use the main result of [3].

Theorem 10. *$\mathcal{P}$ has $(n+1)^{n-1}$ pairs of length n.*

Unlike the classes $\mathcal{A}$, $\mathcal{B}$, the class $\mathcal{P}$ is not a partial order, i.e. $\mathcal{P} \neq \mathcal{P}^2$. Indeed we have the following result proved in [2].

Theorem 11. *For all $k \geqslant 1$, $\mathcal{P}^k$ is strictly contained in $\mathcal{P}^{k+1}$ and the transitive closure of $\mathcal{P}$ is the weak order.*

From the remark following Proposition 3 the set $\mathcal{P}^k$ is the set of allowable pairs for the permuting machine defined by a series arrangement of k priority queues $Q_1, Q_2, \dots, Q_k$ which we view as shown in Fig. 1. In such a machine the fundamental operation is a "move" which either

1. transfers the next input symbol into the first priority queue,
2. transfers the smallest item of one priority queue into the next,
3. transfers the smallest item from the last priority queue to the output.

If during a sequence of moves two elements x, y with $x < y$ ever reside in the same priority queue then x will be output before y. Therefore, when seeking a sequence of moves to produce an output β we may restrict our search to *β-viable* moves; such moves only move an element x into a priority queue Q if there is no element y presently in Q such that $x < y$ and y precedes x in β.

During a sequence of moves that transforms α into β there may be a choice of viable moves at any point. We define the *greedy* algorithm as the one that always makes the leftmost viable move. In other words, if it is possible to move the next symbol expected in the output out of Q_k we make that move. Otherwise we make a viable move from Q_{i-1} into Q_i (or from the input stream into Q_1) for the largest possible i.

Theorem 12. *If $k = 1$ or 2 and if $(\alpha, \beta) \in \mathcal{P}^k$ then the greedy algorithm with k priority queues transforms α into β.*

Proof. Consider some intermediate point in a sequence A of viable moves that transforms α into β. Assume that symbols $b_1 b_2 \dots b_{i-1}$ of β have already been output and that b_i is currently the smallest symbol of the final priority queue. If the next move of A is not a move of b_i to output we define a different sequence A' that is just like A except that the move of b_i to output is promoted so that it takes place at this point. Obviously, this variation does not affect the validity of the subsequent moves of A and so A' also transforms α into β.

This argument settles the case $k = 1$ and, for $k = 2$, shows that we need consider only a situation where A is scheduled to move a symbol from input into the first priority queue whereas the greedy algorithm would have moved a symbol s from the first priority queue to the second. Again we define another sequence A' which is the same as A except that the move of s takes place at this point rather than subsequently. Now we must check that the subsequent moves of A are all valid. Obviously, any move from input and any move out of the first priority queue can still be made. However, the second priority queue now has an extra element s and we must argue that any move demanded by A out

of the second priority queue is still valid. Consider such a move of an element t (that takes place while s is resident in the second priority queue, in which case t precedes s in β). If $t \leqslant s$ this move is still valid so we may assume that $t > s$. In this case, however, the element t cannot have arrived in the second priority queue after s (because in the original sequence A it cannot have been moved from the first priority queue as s was present there). Therefore, t was already in the second priority queue at the point that the move of s was promoted and this contradicts the viability of that greedy move. □

Corollary 13. *There is a polynomial time recognition algorithm for* $\mathcal{P}^2$.

Theorem 12 is false for higher values of k as the following pair shows:

(4 8 3 12 7 2 11 1 6 10 5 9, 1 4 3 2 5 8 7 6 9 12 11 10).

It is easy to check that the greedy algorithm fails when $k = 3$. However, the pair is allowable for $\mathcal{P}^3$ as demonstrated by the sequence which moves, in turn, the elements

4,4,4,8,8,8,3,3,12,12,7,2,11,1,1,1,1,4,3,3,2,2,2,12,7,
11,6,10,5,5,5,5,8,7,7,6,6,6,9,9,9,9,12,11,11,10,10,10

Theorem 11 tells us that the permutational power of k priority queues in series is strictly greater than the power of $k-1$ priority queues in series. Nevertheless, since $\mathcal{P}$ and its transitive closure both have simple bases, one might hope that the various powers of $\mathcal{P}$ were at least finitely based. We shall show that this is false.

Let α_n, β_n (for all $n > 1$) be permutations of length $3n + 2$ defined as

$$\alpha_n = 3\ 6\ 9 \ldots 3n\ 2\ 3n+2\ 5\ 1\ 8\ 4\ 11\ 7 \ldots 3n-1\ 3n-5\ 3n+1\ 3n-2,$$
$$\beta_n = 1\ 3\ 2\ 4\ 6\ 5\ 7\ 9\ 8 \ldots 3n-2\ 3n\ 3n-1\ 3n+2\ 3n+1.$$

Theorem 14. *The pairs* (α_n, β_n) *are part of the basis of* $\mathcal{P}^2$. *In particular,* $\mathcal{P}^2$ *is not finitely based.*

Proof. We shall prove that $(\alpha_n, \beta_n) \notin \mathcal{P}^2$ and then observe that the pair that results from deleting any symbol from α_n and β_n results in a member of $\mathcal{P}^2$.

If there was a sequence of (viable) moves that transformed α_n into β_n it would begin by moving $3, 6, 9, \ldots, 3n$ from the input to the priority queues. We shall initially make the assumption that they are all moved via the first priority queue into the second and comment later on why no generality is lost. Once this has happened the next moves are of 2 and $3n+2$ into the first priority queue; these moves are forced since they are the only viable ones available. At this point we have reached configuration $\mathcal{C}_i$ with $i = 0$ where, in general, the configuration $\mathcal{C}_i$ is one in which $3i$ symbols have already been output, the second priority queue contains $3i+3, \ldots, 3n$, the first priority queue contains $3n+2, 3i+2$ and the next two symbols of the remaining input are $3i+5, 3i+1$ (so long as $i < n$).

In general suppose we are in configuration $\mathcal{C}_i$. From here there is again a unique sequence of viable moves: the symbol $3i+5$ is moved into the first priority queue, then the symbol $3i+1$ is moved to the second priority queue via the first and then output. After that the symbol $3i+3$ is moved from the second priority queue to the output, and the symbol $3i+2$ is moved to the second priority queue and then to output. All this results in reaching configuration $\mathcal{C}_{i+1}$.

Thus, the procedure passes though configurations $\mathcal{C}_0, \mathcal{C}_1, \ldots$. When it reaches configuration $\mathcal{C}_{n-1}$ the second priority queue contains only $3n$, the first contains $3n+2, 3n-1$, and the remaining input symbols are $3n+1, 3n-2$. At this point no viable move is possible and we have failed to transform α_n into β_n.

It is not difficult to confirm that if we had started with disposing $3, 6, 9, \ldots, 3n$ in the two priority queues in a different way then we would have reached a failure condition even earlier. Thus, we now know that $(\alpha_n, \beta_n) \notin \mathcal{P}^2$.

Now suppose that an arbitrary symbol is deleted from α_n and β_n. It can easily be verified that the procedure followed above is able at some point to move the symbol $3n+2$ from the first priority queue into the second. Then when we reach the stage that corresponds to configuration $\mathcal{C}_{n-1}$ we find that the procedure is not blocked and it can complete the transformation. □

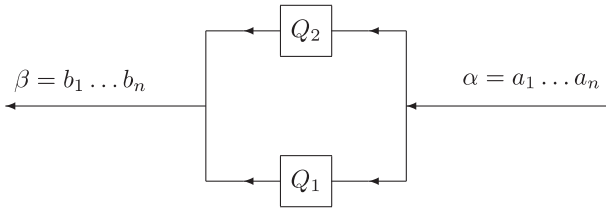

Fig. 2. Two priority queues in parallel.

It seems more difficult to find the complete basis of $\mathcal{P}^2$. By computationally intensive calculation we have found all the basis elements of length up to 9. For lengths up to 7 they are (12, 21), (4321, 1432), (32541, 13254), (42531, 14253). There are 54 of length 8 and none of length 9.

Similar examples can be constructed to show that $\mathcal{P}^k$ is not finitely based for any $k > 2$. These examples depend crucially on the last priority queue having unrestricted size. It might be supposed that if we imposed upper bounds on the number of elements that each of the priority queues could hold we would then have a finitely based system. However, that is not the case, as we see in the next result. To state it we define $\mathcal{P}_t$ to be the set of input–output pairs (α, β) that are allowed by a priority queue which is not permitted to contain more than t elements. Using this notation $\mathcal{P}_s \circ \mathcal{P}_t$ denotes the set of allowable pairs for the serial combination where the input symbols move first into a priority queue of size s, then to a priority queue of size t, and finally to the output stream.

Proposition 15. *$\mathcal{P}_4 \circ \mathcal{P}_2$ is not finitely based.*

Proof. We consider the pairs (θ_n, ϕ_n) for $n > 1$ where

$$\theta_n = 2\ 2n+2\ 4\ 1\ 6\ 3\ 8\ 5\ \dots 2n\ 2n-3\ 2n+1\ 2n-1,$$
$$\phi_n = 1\ 2\ 3\ \dots 2n\ 2n+2\ 2n+1.$$

We now find a situation similar to that in the proof of Theorem 14. When we attempt to transform θ_n into ϕ_n there is a unique viable choice at each step (subject to the storage constraints) during which time the element $2n+2$ remains in the first priority queue. But, at the point that $2n+1$ should enter the first priority queue, there is no viable move available. Furthermore, we see that if any element is omitted from (θ_n, ϕ_n) then it would be possible to move the element $2n+2$ into the second priority queue at some stage and the computation would succeed. □

We close this section with a remark about k priority queues in parallel (see Fig. 2 for the case $k = 2$). If the input to such a permuting machine was in decreasing order then each of the priority queues would behave like a stack (since its smallest symbol would be the most recently inserted). We may therefore appeal to a result of [8] and conclude that, even when $k = 2$, the set of allowable pairs is not finitely based since all of the pairs (ξ_n, ψ_n) where

$$\xi_n = 4n, 4n-1, 4n-2, \dots, 2, 1,$$
$$\psi_n = 4n-3, 4n, 4n-5, 4n-2, 4n-7, 4n-4, \dots, 1, 4, 4n-1, 2$$

are in the basis.

5. Conclusions and future work

We have seen that permuting machines whose behaviour is sensitive to their input values should be studied using the notion of involvement of permutation pairs. This relation is more complex than the subpermutation relation used when the values in the input are immaterial. Nevertheless the two relations have much in common; for example, it is easy to see that, when $(\alpha, \beta) \preccurlyeq (\sigma, \tau)$, $\beta\alpha^{-1}$ is a subpermutation of $\tau\sigma^{-1}$. So it would seem that a general theory of pair involvement will resemble the theory of subpermutations. In this paper the beginnings of such a theory have emerged in results concerned with avoided pairs and recognition procedures.

Perhaps the next step will be to carry out some enumerations for classes other than $\mathcal{P}$, such as the classes $\mathcal{W}$, $\mathcal{A}$, $\mathcal{B}$. One such result is

Proposition 16. *Let q_n be the number of pairs that avoid the set*

$$\{(12, 21), (321, 132), (321, 123), (231, 123)\}.$$

Then

$$\sum q_n x^n/n! = \frac{1}{1 + \log(1 - x)}.$$

Proof. By a simple extension of the proof of Proposition 9 the pairs in question are precisely the pairs of $\mathcal{P}_2$. The result now follows from Theorem 2.1 of [4]. □

Other enumerative results flow from enumerations of sets of permutations that avoid one or more subpermutations. Let T be a set of permutations and let X_T denote the set of permutations that do not have any member of T as a subpermutation. Then it is not hard to verify that the set of pairs

$$Y_T = \{(\sigma, \sigma\tau) \mid \tau \in X_T\}$$

is precisely the closed class that avoids all the pairs of

$$\{\alpha, \alpha\beta \mid \beta \in B\}$$

(in these expressions juxtaposition denotes composition of permutations as mappings). Furthermore $y_n = n!x_n$ where y_n and x_n denote the number of pairs and permutations of length n in Y_T and X_T, respectively. In particular, it follows from the recently proved Wilf-Stanley conjecture [10] that there is some constant c depending only on T such that $x_n \leqslant c^n$, and therefore $y_n \leqslant n!c^n$.

All of these enumerative results are evidence for the following.

Conjecture 1. *The number of pairs of permutations of length n in a closed class that does not consist of all pairs is bounded by $n!c^n$ for some constant c depending only on the class.*

References

[1] M.D. Atkinson, M.H. Albert, N. Ruškuc, Regular closed classes of permutations, Theoret. Comput. Sci. 306 (2003) 85–100.
[2] M.D. Atkinson, R. Beals, Priority queues and permutations, SIAM J. Comput. 23 (1994) 1225–1230.
[3] M.D. Atkinson, M. Thiyagarajah, The permutational power of a priority queue, BIT 33 (1993) 2–6.
[4] M.D. Atkinson, D. Tulley, Bounded capacity priority queues, Theoret. Comput. Sci. 182 (1997) 145–157.
[5] A. Björner, Orderings of Coxeter Groups, Contemporary Mathematics, Vol. 34, American Mathematical Society, Providence, RI, 1984, pp. 175–195.
[6] M. Bóna, A survey of stack-sorting disciplines, Electron. J. Combin. 9 (2) (2003) Paper A1.
[7] Electron. J. Combin. 9 (2) (2003), Special issue on permutation patterns.
[8] S. Even, A. Itai, Queues, stacks and graphs, in: Z. Kohavi, A. Paz (Eds.), Theory of Machines and Computations, Proc. Internat. Symp. on the Theory of Machines and Computations, Technion, Israel Institute of Technology, Haifa, Israel, August 1971, Academic Press, New York, 1971, pp. 71–86.
[9] D.E. Knuth, Fundamental Algorithms, The Art of Computer Programming, second ed., Vol. 1, Addison-Wesley, Reading, MA, 1973.
[10] A. Marcus, G Tardos, A linear bound on excluded permutations, preprint available at <www.renyi.hu/∼tardos/submatrix.ps>, November 2003.
[11] R. Simion, F.W. Schmidt, Restricted permutations, European J. Combin. 6 (1985) 383–406.
[12] N.J.A. Sloane, The on-line encyclopaedia of integer sequences, <http://www.research.att.com/∼njas/sequences/Seis.html>.